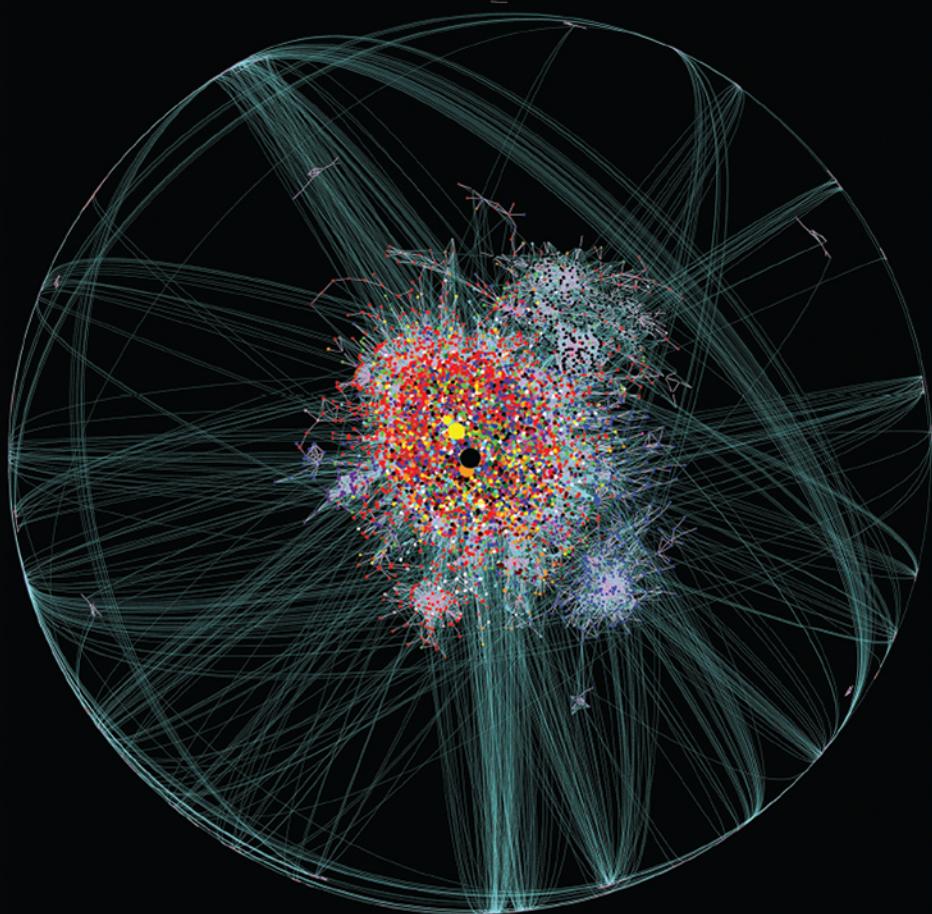


DOUGLAS E. COMER



Redes de computadoras e Internet

Sexta edición

Redes de computadoras e Internet

Redes de computadoras e Internet

Sexta edición

Douglas E. Comer

*Departament of Computer Sciences
Purdue University
West Lafayette, IN 47907*

TRADUCCIÓN

Alfonso Vidal Romero Elizondo

*Ingeniero en computación
Tecnológico de Monterrey, Campus Monterrey*

REVISIÓN TÉCNICA

Cyntia E. Enríquez Ortiz

*Academia de Telemática
Unidad Profesional Interdisciplinaria en Ingeniería
y Tecnologías Avanzadas (UPIITA)
Instituto Politécnico Nacional, México*

PEARSON

Datos de catalogación bibliográfica

COMER, DOUGLAS E.

Redes de computadoras e Internet

Sexta edición

PEARSON EDUCACIÓN, México, 2015

ISBN: 978-607-32-3324-8

Área: Computación

Formato: 20 × 25.5 cm

Páginas: 544

Authorized translation from the English language edition entitled *Computer Networks and Internet 6th Edition*, by *Douglas E. Comer*, published by Pearson Education, Inc., Copyright © 2015. All rights reserved.

ISBN 9780133587937

Traducción autorizada de la edición en idioma inglés titulada *Computer Networks and Internet 6th edición*, por *Douglas E. Comer*, publicada por Pearson Education, Inc., Copyright © 2015. Todos los derechos reservados.

Edición en español

Director General:	Sergio Fonseca Garza
Director de Contenidos y Servicios Digitales:	Alan David Palau
Editor Sponsor:	Luis M. Cruz Castillo e-mail: luis.cruz@pearson.com
Editor de Desarrollo:	Bernardino Gutiérrez Hernández
Supervisor de Producción:	Gustavo Rivas Romero
Gerente de Contenidos Educación Superior:	Marisa de Anta

SEXTA EDICIÓN, 2015

D.R. © 2015 por Pearson Educación de México, S.A. de C.V.

Antonio Dovalí Jaime núm. 70,
Torre B, Piso 6, Col. Zedec, Ed Plaza Santa Fe,
Delegación Álvaro Obregón, C.P. 01210, México, D.F.

Cámara Nacional de la Industria Editorial Mexicana. Reg. núm. 1031.

Reservados todos los derechos. Ni la totalidad ni parte de esta publicación pueden reproducirse, registrarse o transmitirse, por un sistema de recuperación de información, en ninguna forma ni por ningún medio, sea electrónico, mecánico, fotoquímico, magnético o electroóptico, por fotocopia, grabación o cualquier otro, sin permiso previo por escrito del editor.

El préstamo, alquiler o cualquier otra forma de cesión de uso de este ejemplar requerirá también la autorización del editor o de sus representantes.

ISBN LIBRO IMPRESO: 978-607-32-3324-8

ISBN E-BOOK: 978-607-32-3327-9

Impreso en México. *Printed in Mexico.*

1 2 3 4 5 6 7 8 9 0 - 18 17 16 15

PEARSON

www.pearsonenespañol.com

A los paquetes, en todas partes

Contenido

Prefacio	xxiii
Acerca del autor	xxvii

PARTE I Introducción a las redes y las aplicaciones de Internet

Capítulo 1 Introducción y descripción general	1
--	----------

1.1 <i>Crecimiento de las redes de computadoras</i>	1
1.2 <i>Por qué las redes parecen complejas</i>	2
1.3 <i>Los cinco aspectos clave de las redes</i>	2
1.4 <i>La Internet pública y la Internet privada</i>	6
1.5 <i>Redes, interoperabilidad y estándares</i>	8
1.6 <i>Suites de protocolos y modelos de distribución por capas</i>	9
1.7 <i>Cómo pasan los datos a través de las capas</i>	11
1.8 <i>Encabezados y capas</i>	12
1.9 <i>ISO y el modelo de referencia OSI de siete capas</i>	13
1.10 <i>El resto del libro</i>	14
1.11 <i>Resumen</i>	14

Capítulo 2 Tendencias de Internet	17
--	-----------

2.1 <i>Introducción</i>	17
2.2 <i>Compartición de recursos</i>	17
2.3 <i>Crecimiento de Internet</i>	18
2.4 <i>De la compartición de recursos a la comunicación</i>	21
2.5 <i>Del texto al multimedia</i>	21
2.6 <i>Tendencias recientes</i>	22
2.7 <i>De las computadoras personales a la computación en la nube</i>	23
2.8 <i>Resumen</i>	24

Capítulo 3 Aplicaciones de Internet y programación de redes	27
--	-----------

3.1 <i>Introducción</i>	27
3.2 <i>Dos paradigmas esenciales de la comunicación en Internet</i>	28

3.3	<i>Comunicación orientada a la conexión</i>	29
3.4	<i>El modelo de interacción cliente-servidor</i>	30
3.5	<i>Características de los clientes y los servidores</i>	31
3.6	<i>Programas servidor y computadoras tipo servidor</i>	31
3.7	<i>Solicitudes, respuestas y dirección del flujo de datos</i>	32
3.8	<i>Múltiples clientes y múltiples servidores</i>	32
3.9	<i>Identificación de servidores y demultiplexación</i>	33
3.10	<i>Servidores concurrentes</i>	34
3.11	<i>Dependencias circulares entre servidores</i>	35
3.12	<i>Interacciones de igual a igual</i>	35
3.13	<i>Programación de redes y la API de sockets</i>	36
3.14	<i>Sockets, descriptores y entrada y salida de red</i>	36
3.15	<i>Parámetros y la API de sockets</i>	37
3.16	<i>Llamadas de sockets en un cliente y un servidor</i>	38
3.17	<i>Funciones de sockets utilizadas por el cliente y el servidor</i>	38
3.18	<i>La función connect utilizada sólo por un cliente</i>	40
3.19	<i>Funciones de sockets utilizadas sólo por un servidor</i>	40
3.20	<i>Funciones de sockets utilizadas con el paradigma de mensajes</i>	43
3.21	<i>Otras funciones de sockets</i>	44
3.22	<i>Sockets, hilos y herencia</i>	45
3.23	<i>Resumen</i>	45

Capítulo 4 Aplicaciones tradicionales de Internet

49

4.1	<i>Introducción</i>	49
4.2	<i>Protocolos de la capa de aplicación</i>	49
4.3	<i>Representación y transferencia</i>	50
4.4	<i>Protocolos Web</i>	51
4.5	<i>Representación de documentos con HTML</i>	52
4.6	<i>Localizadores uniformes de recursos e hipervínculos</i>	54
4.7	<i>Transferencia de documentos Web con HTTP</i>	55
4.8	<i>Uso de la memoria caché en los navegadores</i>	57
4.9	<i>Arquitectura de los navegadores</i>	59
4.10	<i>Protocolo de transferencia de archivos (FTP)</i>	59
4.11	<i>Paradigma de comunicación de FTP</i>	60
4.12	<i>Correo electrónico</i>	63
4.13	<i>El protocolo simple de transferencia de correo (SMTP)</i>	64
4.14	<i>Proveedores de servicios de Internet (ISP), servidores de correo y acceso al correo</i>	66
4.15	<i>Protocolos de acceso al correo (POP, IMAP)</i>	67
4.16	<i>Estándares de representación de correo electrónico (RFC2822, MIME)</i>	67
4.17	<i>Sistema de nombres de dominio (DNS)</i>	69
4.18	<i>Nombres de dominio que comienzan con un nombre de servicio</i>	71
4.19	<i>La jerarquía del DNS y el modelo servidor</i>	72
4.20	<i>Resolución de nombres</i>	72
4.21	<i>Uso de la memoria caché en servidores del DNS</i>	74

4.22	<i>Tipos de entradas del DNS</i>	75
4.23	<i>Alias y registros de recursos CNAME</i>	76
4.24	<i>Abreviaciones y el DNS</i>	76
4.25	<i>Nombres de dominio internacionalizados</i>	77
4.26	<i>Representaciones extensibles (XML)</i>	78
4.27	<i>Resumen</i>	79

PARTE II Comunicaciones de datos

Capítulo 5 Generalidades de las comunicaciones de datos	85
--	-----------

5.1	<i>Introducción</i>	85
5.2	<i>La esencia de las comunicaciones de datos</i>	86
5.3	<i>Motivación y alcance del tema</i>	87
5.4	<i>Las partes conceptuales de un sistema de comunicaciones</i>	87
5.5	<i>Los subtemas de las comunicaciones de datos</i>	90
5.6	<i>Resumen</i>	91

Capítulo 6 Fuentes de información y señales	93
--	-----------

6.1	<i>Introducción</i>	93
6.2	<i>Fuentes de información</i>	93
6.3	<i>Señales analógicas y digitales</i>	94
6.4	<i>Señales periódicas y aperiódicas</i>	94
6.5	<i>Ondas senoidales y características de las señales</i>	95
6.6	<i>Señales compuestas</i>	97
6.7	<i>La importancia de las señales compuestas y las funciones senoidales</i>	97
6.8	<i>Representaciones en los dominios de tiempo y de frecuencia</i>	98
6.9	<i>Ancho de banda de una señal analógica</i>	99
6.10	<i>Señales digitales y niveles de señal</i>	100
6.11	<i>Baudios y bits por segundo</i>	101
6.12	<i>Conversión de una señal digital en analógica</i>	102
6.13	<i>Ancho de banda de una señal digital</i>	103
6.14	<i>Sincronización y acuerdo sobre señales</i>	103
6.15	<i>Codificación de línea</i>	104
6.16	<i>Codificación Manchester utilizada en redes de computadoras</i>	106
6.17	<i>Conversión de una señal analógica en digital</i>	107
6.18	<i>El teorema de Nyquist y la tasa de muestreo</i>	108
6.19	<i>Teorema de Nyquist y transmisión del sistema telefónico</i>	108
6.20	<i>Codificación no lineal</i>	109
6.21	<i>Codificación y compresión de datos</i>	109
6.22	<i>Resumen</i>	110

Capítulo 7 Medios de transmisión**113**

- 7.1 *Introducción* 113
- 7.2 *Transmisión guiada y no guiada* 113
- 7.3 *Una taxonomía mediante formas de energía* 114
- 7.4 *Radiación de fondo y ruido eléctrico* 115
- 7.5 *Cable de cobre de par trenzado* 115
- 7.6 *Blindaje: cable coaxial y par trenzado blindado* 117
- 7.7 *Categorías de cable de par trenzado* 118
- 7.8 *Medios que usan energía luminosa y fibra óptica* 119
- 7.9 *Tipos de fibra y transmisión de luz* 120
- 7.10 *Comparación entre fibra óptica y cable de cobre* 121
- 7.11 *Tecnologías de comunicación infrarrojas* 122
- 7.12 *Comunicación láser de punto a punto* 122
- 7.13 *Comunicación electromagnética (radio)* 123
- 7.14 *Propagación de señales* 124
- 7.15 *Tipos de satélites* 125
- 7.16 *Satélites en órbita terrestre geoestacionaria (GEO)* 126
- 7.17 *Cobertura GEO de la Tierra* 127
- 7.18 *Satélites en órbita terrestre baja (LEO) y grupos de satélites* 128
- 7.19 *Ventajas y desventajas entre los tipos de medios* 128
- 7.20 *Medición de los medios de transmisión* 129
- 7.21 *El efecto del ruido en la comunicación* 129
- 7.22 *El significado de la capacidad de un canal* 130
- 7.23 *Resumen* 131

Capítulo 8 Confiabilidad y codificación de canales**135**

- 8.1 *Introducción* 135
- 8.2 *Las tres principales fuentes de errores de transmisión* 135
- 8.3 *Efecto de los errores de transmisión sobre los datos* 136
- 8.4 *Dos estrategias para manejar los errores de canal* 137
- 8.5 *Códigos de errores de bloque y convolucionales* 138
- 8.6 *Ejemplo de un código de error de bloque: comprobación de paridad simple* 139
- 8.7 *Las matemáticas de los códigos de errores de bloque y la notación (n,k)* 140
- 8.8 *Distancia de Hamming: la medición de la fuerza de un código* 140
- 8.9 *La distancia de Hamming entre cadenas de un libro de códigos* 141
- 8.10 *Concesión entre la detección de errores y la sobrecarga* 142
- 8.11 *Corrección de errores con paridad de fila y de columna (RAC)* 142
- 8.12 *La suma de verificación de 16 bits que se utiliza en Internet* 144
- 8.13 *Códigos de redundancia cíclica (CRC)* 145
- 8.14 *Una implementación eficiente de hardware de CRC* 148
- 8.15 *Mecanismos de solicitud de repetición automática (ARQ)* 148
- 8.16 *Resumen* 149

Capítulo 9 Modos de transmisión **153**

- 9.1 *Introducción* 153
- 9.2 *Una clasificación de los modos de transmisión* 153
- 9.3 *Transmisión en paralelo* 154
- 9.4 *Transmisión en serie* 155
- 9.5 *Orden de transmisión: bits y bytes* 156
- 9.6 *Sincronización de la transmisión en serie* 156
- 9.7 *Transmisión asíncrona* 157
- 9.8 *Transmisión de caracteres asíncronos RS-232* 157
- 9.9 *Transmisión síncrona* 158
- 9.10 *Bytes, bloques y tramas* 159
- 9.11 *Transmisión isócrona* 160
- 9.12 *Transmisión simplex, semidúplex y dúplex* 160
- 9.13 *Equipo DCE y DTE* 162
- 9.14 *Resumen* 162

Capítulo 10 Modulación y módems **165**

- 10.1 *Introducción* 165
- 10.2 *Portadoras, frecuencia y propagación* 165
- 10.3 *Esquemas analógicos de modulación* 166
- 10.4 *Modulación de amplitud* 166
- 10.5 *Modulación de frecuencia* 167
- 10.6 *Modulación por desplazamiento de fase* 168
- 10.7 *Modulación de amplitud y el teorema de Shannon* 168
- 10.8 *Modulación, entrada digital y codificación por desplazamiento* 168
- 10.9 *Codificación por desplazamiento de fase* 169
- 10.10 *Cambio de fase y diagramas de constelación* 171
- 10.11 *Modulación de amplitud en cuadratura* 173
- 10.12 *Hardware para modulación y demodulación* 174
- 10.13 *Módems ópticos y de radiofrecuencias* 174
- 10.14 *Módems de marcación telefónica* 175
- 10.15 *QAM aplicada a la marcación telefónica* 175
- 10.16 *Módems de marcación telefónica V.32 y V.32bis* 176
- 10.17 *Resumen* 177

Capítulo 11 Multiplexación y demultiplexación (canalización) **181**

- 11.1 *Introducción* 181
- 11.2 *El concepto de multiplexación* 181
- 11.3 *Los tipos básicos de multiplexación* 182
- 11.4 *Multiplexación por división de frecuencias (FDM)* 183

<i>11.5 Uso de un rango de frecuencias por canal</i>	185
<i>11.6 FDM jerárquica</i>	186
<i>11.7 Multiplexación por división de longitud de onda (WDM)</i>	187
<i>11.8 Multiplexación por división de tiempo (TDM)</i>	187
<i>11.9 TDM síncrona</i>	188
<i>11.10 Entramado utilizado en la versión de TDM del sistema telefónico</i>	189
<i>11.11 TDM jerárquica</i>	190
<i>11.12 Intervalos sin completar en la TDM síncrona</i>	190
<i>11.13 TDM estadística</i>	191
<i>11.14 Multiplexación inversa</i>	192
<i>11.15 Multiplexación por división de código</i>	193
<i>11.16 Resumen</i>	195

Capítulo 12 Tecnologías de acceso e interconexión 199

<i>12.1 Introducción</i>	199
<i>12.2 Tecnologías de acceso a Internet: flujo ascendente y descendente</i>	199
<i>12.3 Tecnologías de acceso de banda estrecha y banda ancha</i>	200
<i>12.4 El bucle local y la ISDN</i>	202
<i>12.5 Tecnologías de la línea de suscriptor digital (DSL)</i>	202
<i>12.6 Características del bucle local y adaptación</i>	203
<i>12.7 La velocidad de datos de las líneas ADSL</i>	204
<i>12.8 Instalación de ADSL y filtros divisores</i>	205
<i>12.9 Tecnologías de módems de cable</i>	205
<i>12.10 La velocidad de datos de los módems de cable</i>	206
<i>12.11 Instalación de un módem de cable</i>	206
<i>12.12 Fibra híbrida coaxial</i>	207
<i>12.13 Tecnologías de acceso que emplean la fibra óptica</i>	208
<i>12.14 Terminología de módems de extremo cercano y extremo lejano</i>	208
<i>12.15 Tecnologías de acceso inalámbricas</i>	209
<i>12.16 Conexiones de alta capacidad en el núcleo de Internet</i>	209
<i>12.17 Terminación de circuitos, DSU/CSU y NIU</i>	210
<i>12.18 Estándares de telefonía para circuitos digitales</i>	211
<i>12.19 Terminología DS y velocidades de datos</i>	212
<i>12.20 Circuitos de mayor capacidad (estándares STS)</i>	212
<i>12.21 Estándares de portadora óptica</i>	213
<i>12.22 El sufijo C</i>	213
<i>12.23 Red óptica síncrona (SONET)</i>	214
<i>12.24 Resumen</i>	215

PARTE III Comutación de paquetes y tecnologías de redes

Capítulo 13 Redes de área local: paquetes, tramas y topologías 219

- 13.1 *Introducción* 219
- 13.2 *Comutación de circuitos y comunicación analógica* 220
- 13.3 *Comutación de paquetes* 221
- 13.4 *Redes de paquetes de área local y amplia* 222
- 13.5 *Estándares para formato e identificación de paquetes* 223
- 13.6 *El modelo IEEE 802 y los estándares* 224
- 13.7 *Redes punto a punto y multiacceso* 225
- 13.8 *Topologías de LAN* 227
- 13.9 *Identificación de paquetes, demultiplexación, direcciones MAC* 229
- 13.10 *Direcciones de unidifusión, difusión y multidifusión* 230
- 13.11 *Difusión, multidifusión y entrega multipunto eficiente* 231
- 13.12 *Tramas y entramado* 232
- 13.13 *Relleno de bytes y bits* 233
- 13.14 *Resumen* 234

Capítulo 14 La subcapa MAC del IEEE 239

- 14.1 *Introducción* 239
- 14.2 *Una taxonomía de los mecanismos para acceso compartido* 239
- 14.3 *Asignación estática y dinámica de canales* 240
- 14.4 *Protocolos de canalización* 241
- 14.5 *Protocolos de acceso controlado* 242
- 14.6 *Protocolos de acceso aleatorio* 244
- 14.7 *Resumen* 250

Capítulo 15 Tecnología alámbrica de LAN (Ethernet y 802.3) 253

- 15.1 *Introducción* 253
- 15.2 *La venerable Ethernet* 253
- 15.3 *Formato de tramas de Ethernet* 254
- 15.4 *Campo tipo de la trama de Ethernet y demultiplexación* 254
- 15.5 *Versión del IEEE de Ethernet (802.3)* 255
- 15.6 *Conexiones de LAN y tarjetas de interfaz de red* 256
- 15.7 *Evolución de Ethernet y cableado de Thicknet* 256
- 15.8 *Cableado de Thinnet* 257
- 15.9 *Cableado de Ethernet de par trenzado y concentradores* 258
- 15.10 *Topología física y lógica de Ethernet* 259
- 15.11 *Cableado en un edificio de oficinas* 259

15.12	<i>Velocidades de datos y tipos de cables de Ethernet</i>	261
15.13	<i>Conectores y cables de par trenzado</i>	261
15.14	<i>Resumen</i>	262

Capítulo 16 Tecnologías de redes inalámbricas**265**

16.1	<i>Introducción</i>	265
16.2	<i>Una taxonomía de las redes inalámbricas</i>	265
16.3	<i>Redes de área personal (PAN)</i>	266
16.4	<i>Bandas inalámbricas ISM utilizadas por redes LAN y PAN</i>	267
16.5	<i>Tecnologías de redes LAN inalámbricas y Wi-Fi</i>	267
16.6	<i>Tecnologías de espectro amplio</i>	268
16.7	<i>Otros estándares de redes LAN inalámbricas</i>	269
16.8	<i>Arquitectura de redes LAN inalámbricas</i>	270
16.9	<i>Superposición, asociación y formato de trama 802.11</i>	271
16.10	<i>Coordinación entre puntos de acceso</i>	272
16.11	<i>Acceso con y sin contención</i>	272
16.12	<i>Tecnología MAN inalámbrica y WiMax</i>	274
16.13	<i>Tecnologías y estándares de redes PAN</i>	276
16.14	<i>Otras tecnologías de comunicación de corta distancia</i>	277
16.15	<i>Tecnologías de redes WAN inalámbricas</i>	278
16.16	<i>Microceldas</i>	280
16.17	<i>Grupos de celdas y reutilización de frecuencias</i>	280
16.18	<i>Generaciones de tecnologías celulares</i>	282
16.19	<i>Tecnología de satélite VSAT</i>	284
16.20	<i>Satélites GPS</i>	285
16.21	<i>Radio definido por software y el futuro de la tecnología inalámbrica</i>	286
16.22	<i>Resumen</i>	287

Capítulo 17 Repetidores, puentes y conmutadores**291**

17.1	<i>Introducción</i>	291
17.2	<i>Limitación de distancia y diseño de redes LAN</i>	291
17.3	<i>Extensiones de módem de fibra óptica</i>	292
17.4	<i>Repetidores</i>	293
17.5	<i>Puentes y uso de puentes</i>	293
17.6	<i>Puentes con capacidad de aprendizaje y filtrado de tramas</i>	294
17.7	<i>Por qué es bueno usar puentes</i>	295
17.8	<i>Árbol de expansión distribuido</i>	296
17.9	<i>Comutación y conmutadores de la capa 2</i>	297
17.10	<i>Conmutadores de redes VLAN</i>	299
17.11	<i>Múltiples conmutadores y redes VLAN compartidas</i>	300
17.12	<i>La importancia de los puentes</i>	301
17.13	<i>Resumen</i>	302

Capítulo 18 Tecnologías de WAN y enrutamiento dinámico 305

- 18.1 *Introducción* 305
- 18.2 *Grandes extensiones y redes de área amplia* 305
- 18.3 *Arquitectura de WAN tradicional* 306
- 18.4 *Formación de una WAN* 308
- 18.5 *Paradigma de almacenamiento y reenvío* 309
- 18.6 *Direccionamiento en una WAN* 309
- 18.7 *Reenvío del siguiente salto* 310
- 18.8 *Independencia del origen* 313
- 18.9 *Actualizaciones de enrutamiento dinámico en una WAN* 313
- 18.10 *Rutas predeterminadas* 314
- 18.11 *Cálculo de la tabla de reenvío* 315
- 18.12 *Cálculo de ruta distribuido* 316
- 18.13 *Rutas más cortas y pesos* 320
- 18.14 *Problemas de enrutamiento* 321
- 18.15 *Resumen* 322

Capítulo 19 Tecnologías de redes pasadas y presentes 325

- 19.1 *Introducción* 325
- 19.2 *Tecnologías de conexión y acceso* 325
- 19.3 *Tecnologías de LAN* 327
- 19.4 *Tecnologías de WAN* 328
- 19.5 *Resumen* 332

PARTE IV Interconexión de redes mediante el uso de TCP/IP**Capítulo 20 Interconexión de redes: conceptos, arquitectura y protocolos 335**

- 20.1 *Introducción* 335
- 20.2 *La motivación para la interconexión de redes* 335
- 20.3 *El concepto de servicio universal* 336
- 20.4 *Servicio universal en un mundo heterogéneo* 336
- 20.5 *Interconexión de redes* 337
- 20.6 *Conexión física de redes con enrutadores* 337
- 20.7 *Arquitectura de Internet* 338
- 20.8 *Intranet e Internet* 339
- 20.9 *Obtención de un servicio universal* 339
- 20.10 *Una red virtual* 339
- 20.11 *Protocolos para la interconexión de redes* 341
- 20.12 *Repaso de la distribución en capas de TCP/IP* 341

20.13	<i>Computadoras host, enrutadores y capas de protocolos</i>	342
20.14	<i>Resumen</i>	342

Capítulo 21 IP: direccionamiento de Internet 345

21.1	<i>Introducción</i>	345
21.2	<i>El cambio a IPv6</i>	345
21.3	<i>El modelo del reloj de arena y la dificultad del cambio</i>	346
21.4	<i>Direcciones para Internet virtual</i>	346
21.5	<i>El esquema de direccionamiento de IP</i>	348
21.6	<i>La jerarquía de direcciones IP</i>	348
21.7	<i>Clases originales de direcciones IPv4</i>	349
21.8	<i>Notación decimal con puntos de IPv4</i>	350
21.9	<i>Autoridad para las direcciones</i>	351
21.10	<i>Subred IPv4 y direccionamiento sin clases</i>	351
21.11	<i>Máscaras de direcciones</i>	353
21.12	<i>Notación CIDR utilizada con IPv4</i>	354
21.13	<i>Un ejemplo de CIDR</i>	354
21.14	<i>Direcciones de hosts de CIDR</i>	356
21.15	<i>Direcciones IPv4 especiales</i>	357
21.16	<i>Resumen de direcciones IPv4 especiales</i>	359
21.17	<i>Formato de dirección de difusión Berkeley de IPv4</i>	359
21.18	<i>Los enrutadores y el principio de direccionamiento de IPv4</i>	360
21.19	<i>Hosts multiproveedor</i>	361
21.20	<i>Multihoming de IPv6 y renumeración de red</i>	361
21.21	<i>Direccionamiento de IPv6</i>	362
21.22	<i>Notación hexadecimal de dos puntos de IPv6</i>	363
21.23	<i>Resumen</i>	364

Capítulo 22 Reenvío de datagramas 369

22.1	<i>Introducción</i>	369
22.2	<i>Servicio sin conexión</i>	369
22.3	<i>Paquetes virtuales</i>	370
22.4	<i>El datagrama IP</i>	370
22.5	<i>El formato de encabezado del datagrama IPv4</i>	371
22.6	<i>El formato de encabezado del datagrama IPv6</i>	373
22.7	<i>Formato de encabezado base IPv6</i>	373
22.8	<i>Reenvío de un datagrama IP</i>	375
22.9	<i>Extracción de prefijos de red y reenvío de datagramas</i>	376
22.10	<i>Coincidencia del prefijo más extenso</i>	377
22.11	<i>Dirección de destino y dirección del siguiente salto</i>	378
22.12	<i>Entrega del mejor esfuerzo</i>	378

22.13	<i>Encapsulamiento de IP</i>	379
22.14	<i>Transmisión a través de una interred</i>	380
22.15	<i>MTU y fragmentación de datagramas</i>	381
22.16	<i>Fragmentación de un datagrama IPv6</i>	383
22.17	<i>Reensamblaje de un datagrama IP a partir de fragmentos</i>	384
22.18	<i>Recolección de los fragmentos de un datagrama</i>	385
22.19	<i>La consecuencia de la pérdida de fragmentos</i>	386
22.20	<i>Fragmentación de un fragmento IPv4</i>	386
22.21	<i>Resumen</i>	387

Capítulo 23 Protocolos y tecnologías de soporte 391

23.1	<i>Introducción</i>	391
23.2	<i>Resolución de direcciones</i>	391
23.3	<i>Un ejemplo de direcciones IPv4</i>	393
23.4	<i>El protocolo de resolución de direcciones (ARP) IPv4</i>	393
23.5	<i>Formato de mensajes del ARP</i>	394
23.6	<i>Encapsulamiento del ARP</i>	395
23.7	<i>Uso de caché y procesamiento de mensajes del ARP</i>	396
23.8	<i>El límite conceptual de direcciones</i>	398
23.9	<i>Protocolo de mensajes de control de Internet (ICMP)</i>	399
23.10	<i>Formato de mensajes y encapsulamiento del ICMP</i>	400
23.11	<i>Vinculación de direcciones IPv6 con descubrimiento del vecindario</i>	401
23.12	<i>Software de protocolo, parámetros y configuración</i>	401
23.13	<i>Protocolo de configuración dinámica de host (DHCP)</i>	402
23.14	<i>Operación del protocolo DHCP y optimizaciones</i>	403
23.15	<i>Formato de mensajes del DHCP</i>	404
23.16	<i>Acceso indirecto a un servidor DHCP por medio de un retransmisor</i>	405
23.17	<i>Configuración automática de IPv6</i>	405
23.18	<i>Traducción de direcciones de red (NAT)</i>	406
23.19	<i>Operación de NAT y direcciones IPv4 privadas</i>	407
23.20	<i>NAT de la capa de transporte (NAPT)</i>	409
23.21	<i>NAT y los servidores</i>	410
23.22	<i>Software y sistemas NAT para usar en casa</i>	410
23.23	<i>Resumen</i>	411

Capítulo 24 UDP: servicio de transporte de datagramas 415

24.1	<i>Introducción</i>	415
24.2	<i>Protocolos de transporte y comunicación de extremo a extremo</i>	415
24.3	<i>El protocolo de datagramas de usuario</i>	416
24.4	<i>El paradigma sin conexión</i>	417
24.5	<i>Interfaz orientada a mensajes</i>	417

24.6	<i>Semántica de la comunicación UDP</i>	418
24.7	<i>Modos de interacción y entrega por multidifusión</i>	419
24.8	<i>Identificación del punto final con números de puerto de protocolo</i>	419
24.9	<i>Formato de los datagramas UDP</i>	420
24.10	<i>La suma de verificación UDP y el seudoencabezado</i>	421
24.11	<i>Encapsulamiento UDP</i>	421
24.12	<i>Resumen</i>	422

Capítulo 25 TCP: servicio de transporte confiable 425

25.1	<i>Introducción</i>	425
25.2	<i>El protocolo de control de transmisión</i>	425
25.3	<i>El servicio que TCP ofrece a las aplicaciones</i>	426
25.4	<i>Servicio de extremo a extremo y conexiones virtuales</i>	427
25.5	<i>Técnicas que utilizan los protocolos de transporte</i>	428
25.6	<i>Técnicas para evitar la congestión</i>	432
25.7	<i>El arte del diseño de protocolos</i>	433
25.8	<i>Técnicas utilizadas en TCP para manejar la pérdida de paquetes</i>	434
25.9	<i>Retransmisión adaptativa</i>	435
25.10	<i>Comparación de los tiempos de retransmisión</i>	436
25.11	<i>Búferes, control de flujo y ventanas</i>	437
25.12	<i>La negociación en tres pasos de TCP</i>	438
25.13	<i>Control de congestión de TCP</i>	440
25.14	<i>Versiones del control de congestión de TCP</i>	441
25.15	<i>Otras variaciones: SACK y ECN</i>	441
25.16	<i>Formato del segmento de TCP</i>	442
25.17	<i>Resumen</i>	443

Capítulo 26 Enrutamiento de Internet y protocolos de enrutamiento 447

26.1	<i>Introducción</i>	447
26.2	<i>Comparación entre enrutamiento estático y dinámico</i>	447
26.3	<i>Enrutamiento estático en hosts y una ruta predeterminada</i>	448
26.4	<i>Enrutamiento dinámico y enrutadores</i>	449
26.5	<i>Enrutamiento en la red Internet global</i>	450
26.6	<i>Concepto de sistema autónomo</i>	451
26.7	<i>Los dos tipos de protocolos de enrutamiento de Internet</i>	451
26.8	<i>Rutas y tráfico de datos</i>	454
26.9	<i>El protocolo de puerta de enlace límite (BGP)</i>	454
26.10	<i>El protocolo de información de enrutamiento (RIP)</i>	456
26.11	<i>Formato de paquetes de RIP</i>	457
26.12	<i>El protocolo de la ruta más corta primero (OSPF)</i>	458
26.13	<i>Ejemplo de un gráfico de OSPF</i>	459

- 26.14 Áreas del OSPF 459
- 26.15 Sistema intermedio a sistema intermedio (IS-IS) 460
- 26.16 Enrutamiento por multidifusión 461
- 26.17 Resumen 465

Los capítulos 27 a 33 se encuentran en español en el sitio Web del libro

PARTE V Otros conceptos de tecnología de redes

Capítulo 27 Rendimiento de la red (QoS y DiffServ) 469

- 27.1 Introducción 469
- 27.2 Medidas de rendimiento 469
- 27.3 Latencia o retraso 470
- 27.4 Capacidad, velocidad de transferencia y caudal útil 472
- 27.5 Comprensión de la velocidad de transferencia y el retraso 473
- 27.6 Fluctuación 474
- 27.7 La relación entre el retraso y la velocidad de transferencia 475
- 27.8 Medición del retraso, la velocidad de transferencia y la fluctuación 476
- 27.9 Medición pasiva, paquetes pequeños y NetFlow 478
- 27.10 Calidad del servicio (QoS) 479
- 27.11 QoS de grano fino y de grano grueso 480
- 27.12 Implementación de QoS 482
- 27.13 Tecnologías de QoS en Internet 484
- 27.14 Resumen 485

Capítulo 28 Multimedia y telefonía IP (VoIP) 489

- 28.1 Introducción 489
- 28.2 Transmisión de datos en tiempo real y entrega del mejor esfuerzo 489
- 28.3 Reproducción con retraso y búferes de fluctuación 490
- 28.4 Protocolo de transporte en tiempo real (RTP) 491
- 28.5 Encapsulamiento de RTP 492
- 28.6 Telefonía IP 493
- 28.7 Señalización y estándares de señalización de VoIP 494
- 28.8 Componentes de un sistema telefónico IP 495
- 28.9 Resumen de protocolos y distribución en capas 498
- 28.10 Características de H.323 499
- 28.11 Distribución en capas de H.323 499
- 28.12 Características y métodos de SIP 500
- 28.13 Ejemplo de una sesión de SIP 501
- 28.14 Asignación y enrutamiento de números telefónicos 502
- 28.15 Resumen 503

Capítulo 29 Seguridad de redes **507**

- 29.1 *Introducción* 507
- 29.2 *Abusos y ataques criminales* 507
- 29.3 *Políticas de seguridad* 511
- 29.4 *Responsabilidad y control* 512
- 29.5 *Tecnologías de seguridad* 513
- 29.6 *Hashing: un mecanismo de integridad y autenticación* 513
- 29.7 *Control de acceso y contraseñas* 514
- 29.8 *Cifrado: una técnica de seguridad fundamental* 514
- 29.9 *Cifrado de clave privada* 515
- 29.10 *Cifrado de clave pública* 515
- 29.11 *Autenticación con firmas digitales* 516
- 29.12 *Autoridades de claves y certificados digitales* 517
- 29.13 *Firewall* 519
- 29.14 *Implementación de un firewall con un filtro de paquetes* 520
- 29.15 *Sistemas de detección de intrusos* 522
- 29.16 *Exploración de contenido e inspección detallada de paquetes* 522
- 29.17 *Redes privadas virtuales (VPN)* 523
- 29.18 *El uso de la tecnología VPN para el trabajo a distancia* 525
- 29.19 *Comparación entre cifrado de paquetes y uso de túneles* 526
- 29.20 *Tecnologías de seguridad* 528
- 29.21 *Resumen* 529

Capítulo 30 Administración de redes (SNMP) **533**

- 30.1 *Introducción* 533
- 30.2 *Administración de una intranet* 533
- 30.3 *FCAPS: el modelo estándar de la industria* 534
- 30.4 *Ejemplos de elementos de red* 536
- 30.5 *Herramientas de administración de redes* 536
- 30.6 *Aplicaciones de administración de redes* 538
- 30.7 *Protocolo simple de administración de redes* 539
- 30.8 *Paradigma de obtener y almacenar de SNMP* 539
- 30.9 *La MIB de SNMP y los nombres de objetos* 540
- 30.10 *Las variables de MIB* 541
- 30.11 *Variables de MIB que corresponden a arreglos* 541
- 30.12 *Resumen* 542

Capítulo 31 Redes definidas por software (SDN) **545**

- 31.1 *Introducción* 545
- 31.2 *El despliegue publicitario y la realidad* 545

31.3	<i>Motivación para un nuevo enfoque</i>	546
31.4	<i>Organización conceptual de un elemento de red</i>	548
31.5	<i>Módulos del plano de control y la interfaz de hardware</i>	549
31.6	<i>Un nuevo paradigma: las redes definidas por software</i>	550
31.7	<i>Preguntas sin responder</i>	551
31.8	<i>Controladores compartidos y conexiones de red</i>	552
31.9	<i>Comunicación SDN</i>	553
31.10	<i>OpenFlow: un protocolo de controlador a elemento</i>	554
31.11	<i>Motores de clasificación en los conmutadores</i>	555
31.12	<i>TCAM y clasificación de alta velocidad</i>	556
31.13	<i>Clasificación entre varias capas de protocolos</i>	557
31.14	<i>Tamaño de TCAM y la necesidad de patrones múltiples</i>	557
31.15	<i>Elementos que OpenFlow puede especificar</i>	558
31.16	<i>Reenvío de IP tradicional y extendido</i>	559
31.17	<i>Ruta de extremo a extremo con MPLS usando la capa 2</i>	560
31.18	<i>Creación de reglas dinámicas y control de flujos</i>	561
31.19	<i>Un modelo de canalización para tablas de flujo</i>	562
31.20	<i>Efecto potencial de SDN en los fabricantes de red</i>	563
31.21	<i>Resumen</i>	564

Capítulo 32 La Internet de las cosas 567

32.1	<i>Introducción</i>	567
32.2	<i>Sistemas integrados</i>	567
32.3	<i>Elección de una tecnología de red</i>	569
32.4	<i>Recolección de energía</i>	570
32.5	<i>Comunicación inalámbrica de baja potencia</i>	570
32.6	<i>Topología de malla</i>	571
32.7	<i>La alianza ZigBee</i>	571
32.8	<i>Radios 802.15.4 y redes de malla inalámbricas</i>	572
32.9	<i>Conectividad de Internet y enrutamiento de malla</i>	573
32.10	<i>IPv6 en una red de malla ZigBee</i>	574
32.11	<i>El paradigma de reenvío de ZigBee</i>	575
32.12	<i>Otros protocolos en la pila ZigBee</i>	576
32.13	<i>Resumen</i>	577

Capítulo 33 Tendencias en tecnologías de redes y usos 579

33.1	<i>Introducción</i>	579
33.2	<i>La necesidad de servicios de Internet escalables</i>	579
33.3	<i>Almacenamiento de contenido en caché (Akamai)</i>	580
33.4	<i>Balanceadores de carga web</i>	580
33.5	<i>Virtualización de servidores</i>	581

33.6	<i>Comunicación de igual a igual</i>	581
33.7	<i>Centros de datos distribuidos y replicación</i>	582
33.8	<i>Representación universal (XML)</i>	582
33.9	<i>Redes sociales</i>	583
33.10	<i>Movilidad y redes inalámbricas</i>	583
33.11	<i>Video digital</i>	583
33.12	<i>Acceso y conmutación de mayor velocidad</i>	584
33.13	<i>Computación en la nube</i>	584
33.14	<i>Redes superpuestas</i>	584
33.15	<i>Middleware</i>	586
33.16	<i>Implementación generalizada de IPv6</i>	586
33.17	<i>Resumen</i>	587

Apéndice 1 Una interfaz de programación de aplicaciones simplificada

A-1

Índice

I-1

Prefacio

Agradezco a los muchos lectores que se tomaron el tiempo de enviarme comentarios de las ediciones anteriores de *Redes de computadoras e Internet*. Las revisiones han sido increíblemente positivas y la audiencia es sorprendentemente extensa. Además de los estudiantes que usan el libro, los profesionales de redes también escribieron para elogiar su claridad y describir cómo les ayudó a aprobar los exámenes de certificación profesional. También me han llegado muchos comentarios entusiastas de usuarios de todo el mundo; algunos sobre la versión en inglés y otros sobre traducciones en su idioma. El éxito es especialmente satisfactorio tomando en cuenta que hay un mercado lleno de libros sobre redes. Este libro sobresale debido a su amplia cobertura, organización lógica, explicación de los conceptos y su enfoque en Internet, que lo hacen atractivo tanto para profesores como para estudiantes.

Novedades en esta edición

En respuesta a las sugerencias de los lectores y los cambios recientes en las redes, esta edición se revisó y actualizó por completo. Como siempre, el material sobre las tecnologías anteriores se redujo de manera considerable y fue reemplazado por material nuevo. Los cambios más importantes incluyen:

- Actualizaciones a lo largo de cada capítulo
- Figuras adicionales para mejorar las explicaciones
- Integración de IPv4 e IPv6 en todos los capítulos
- Cobertura mejorada de MPLS y túneles
- Nuevo capítulo sobre redes definidas por software y OpenFlow
- Nuevo capítulo sobre Internet de las cosas y ZigBee

Metodología utilizada

¿Deben los cursos aplicar una metodología descendente o ascendente con respecto al tema? En una metodología descendente, se comienza con la transmisión de bits a través de un solo cable y luego se aprende cómo es que las capas sucesivas de protocolos expanden la funcionalidad. En una metodología descendente, comenzamos con las aplicaciones de alto nivel, aprendiendo en un principio sólo lo suficiente para comprender la forma en que operan dichas aplicaciones. Posteriormente aprendemos sobre los detalles subyacentes.

Este libro combina lo mejor de ambas metodologías. Comienza con una explicación sobre las aplicaciones de red y los paradigmas de comunicación que ofrece Internet. Permite a los estudiantes comprender la estructura que brinda Internet a las aplicaciones antes de estudiar las tecnologías que implementan dicha estructura. Después de la explicación sobre las aplicaciones, el texto presenta las redes de una manera lógica, de modo que el lector comprenda cómo es que cada nueva tecnología se basa en las tecnologías de las capas inferiores.

Público objetivo

El libro responde a la pregunta básica: ¿cómo es que operan las redes de computadoras y la Internet? Ofrece un paseo exhaustivo a través de las teorías de redes que describen las aplicaciones, los protocolos de Internet, las tecnologías de red (como LAN y WAN) y los detalles de bajo nivel (como la transmisión de datos y el cableado). Muestra cómo es que los protocolos usan el hardware subyacente y cómo las aplicaciones usan la pila de protocolos para ofrecer funcionalidad a los usuarios.

Como está diseñado para estudiantes universitarios o del primer año de posgrado que tienen poca o ninguna experiencia en redes, el texto no usa expresiones matemáticas sofisticadas ni da por hecho que existe un conocimiento detallado de los sistemas operativos; define los conceptos con claridad, usa ejemplos y figuras para ilustrar cómo opera la tecnología e indica los resultados del análisis sin ofrecer pruebas matemáticas.

Organización del material

El texto está dividido en cinco partes. La primera (capítulos 1 a 4) se concentra en los usos de Internet y las aplicaciones de red. Describe la distribución de los protocolos en capas, el modelo de interacción cliente-servidor y la API de sockets; además, da ejemplos de los protocolos de la capa de aplicación que se usan en Internet.

La segunda parte (capítulos 5 a 12) explica las comunicaciones de datos y presenta los antecedentes sobre el hardware, el vocabulario básico y los conceptos fundamentales utilizados a lo largo de las redes, como el ancho de banda, la modulación y la multiplexación. El capítulo 12 presenta las tecnologías de acceso e interconexión que se utilizan en Internet, y usa conceptos de capítulos anteriores para explicar cada tecnología.

La tercera parte (capítulos 13 a 19) se concentra en la conmutación de paquetes y las tecnologías de redes de conmutación de paquetes. Proporciona la justificación para el uso de paquetes, introduce el modelo del IEEE para los protocolos de la capa 2 y considera las tecnologías de redes alámbricas e inalámbricas, como Ethernet y Wi-Fi. También introduce las cuatro categorías básicas de tecnologías de redes: LAN, MAN, PAN y WAN, y habla sobre el enrutamiento de redes WAN. El último capítulo de esta parte presenta ejemplos de tecnologías de redes que se han utilizado en Internet.

La cuarta parte (capítulos 20 a 26) se concentra en los protocolos de Internet. Después de exponer las causas para la interconexión de redes, el texto describe la arquitectura de Internet, los enrutadores, el direccionamiento de Internet, la vinculación de direcciones y la suite de protocolos TCP/IP. Los protocolos como IPv4, IPv6, TCP, UDP, ICMP, ICMPv6 y ARP se revisan con detalle para permitir a los estudiantes comprender cómo se relacionan los conceptos con la práctica. Debido a que IPv6 (finalmente) comenzó a implementarse, se integró material sobre este tema. Cada capítulo presenta los conceptos generales y luego explica cómo se implementan esos conceptos en IPv4 e IPv6. El capítulo 25 sobre TCP cubre el importante tema de la confiabilidad en los protocolos de transporte.

La quinta parte (capítulos 27 a 33, en el sitio web del libro) considera temas que cruzan varias capas de una pila de protocolos, incluyendo el desempeño, la seguridad y la administración de la red, así como el arranque o *bootstrapping*, el soporte multimedia e Internet de las cosas. El capítulo 31 presenta las redes definidas por software, uno de los nuevos y más emocionantes desarrollos en el trabajo en red. Cada capítulo se basa en los temas de partes anteriores del libro.

Uso en los cursos

El libro se adapta perfectamente a un curso de un semestre sobre redes a nivel introductorio, impartido ya sea el primero o último año. Al estar diseñado para un curso intensivo, cubre todos los temas, desde cableado hasta aplicaciones. Aunque muchos profesores optan por omitir el material sobre comunicaciones de datos, yo recomiendo que extraigan los conceptos y la terminología clave, que serán importantes para los últimos capítulos. Sin importar cómo se organicen los cursos, recomiendo a los profesores que involucren a los estudiantes en actividades prácticas. Por ejemplo, en el curso para estudiantes universitarios en la Universidad Purdue, los estudiantes reciben actividades de laboratorio semanales que abarcan una amplia variedad de temas: desde la medición de redes y el análisis de paquetes, hasta la programación de redes. Cuando termine de leer este libro, cada estudiante debe conocer cómo es que un enrutador IP usa una tabla de rutas para elegir el siguiente salto de un datagrama IP; describir cómo un datagrama cruza Internet; identificar y explicar los campos en una trama de Ethernet; saber cómo TCP identifica una conexión y por qué un servidor Web concurrente puede manejar varias conexiones al puerto 80; calcular la longitud de un bit individual a medida que se propaga a través de un cable a la velocidad de la luz; explicar por qué TCP se clasifica como de extremo a extremo; saber por qué es importante la comunicación de máquina a máquina para la Internet de las cosas y comprender la motivación para la tecnología SDN.

El objetivo de un curso individual es la amplitud, no la profundidad. Para cubrir el tema no podemos enfocarnos en unas cuantas tecnologías o en unos cuantos conceptos. Por lo tanto, la clave de un curso exitoso está en mantener un ritmo rápido. Para cubrir los temas más importantes en un semestre, el material de las capas inferiores en la segunda parte puede condensarse y las secciones sobre redes e interconexión de redes pueden dividirse en cuatro semanas cada una, dejando dos semanas para el material introductorio sobre aplicaciones y temas como la administración de redes y la seguridad. Los detalles de la programación de sockets pueden cubrirse con ejercicios de programación, ya sea en laboratorios o como problemas de tarea.

Los profesores deben concientizar a sus estudiantes sobre la importancia de los conceptos y principios. Las tecnologías específicas pueden volverse obsoletas en unos cuantos años, pero los principios prevalecerán. Además, los profesores deben ayudar a los estudiantes a que sientan la emoción que impregna a las redes. Esta emoción continúa debido a que las redes siguen cambiando, como lo ilustra la nueva era de las redes definidas por software.

Aunque ningún tema en sí es difícil, tal vez a los estudiantes les parezca que la cantidad de material es demasiada y que se enfrentan a una gran cantidad de términos nuevos. Los acrónimos y la jerga de redes pueden serles especialmente desafiantes y tal vez sientan que deben invertir gran parte de su tiempo en acostumbrarse a usar los términos correctos. En las clases en Purdue, recomendamos a los estudiantes que tengan una lista de términos, y hemos descubierto que un examen de vocabulario semanal ayuda a que se aprendan la terminología a medida que avanza el semestre, y no tengan que esperar hasta un examen final.

Puesto que la programación y la experimentación son cruciales para ayudar a los estudiantes a que aprendan sobre redes, la experiencia práctica es una parte esencial de cualquier curso sobre este tema. En Purdue, comenzamos el semestre haciendo que los estudiantes construyan software cliente para acceder a Web y extraer datos (por ejemplo, escribir un programa para visitar un sitio Web e imprimir la temperatura actual). El apéndice 1 es extremadamente útil para iniciar, ya que explica una API simplificada. La API, que está disponible en el sitio Web del autor permite a los estudiantes escribir código funcional antes de que aprendan sobre protocolos, direcciones, sockets o la (algo tediosa) API de sockets. Desde luego que, más adelante en el semestre, los estudiantes aprenden programación de sockets. Finalmente logran escribir un servidor Web concurrente. El soporte para secuencias de comandos del lado servidor es opcional, pero la mayoría lo completa. Además de la programación de aplicaciones, los estudiantes usan nuestras instalaciones de laboratorio para capturar paquetes de una red en funcionamiento, escriben programas que decodifican encabezados de paquetes (como Ethernet, IP y TCP) y observan las conexiones TCP. Si no hay instalaciones de laboratorio avanzadas disponibles, pueden experimentar con algún software gratuito analizador de paquetes, como *Wireshark*.

Además del código de la API simplificada, en el sitio Web del libro encontrará material adicional en inglés, para profesores; visítenos en:

www.pearsonenespanol.com/comer

Agradezco a todas las personas que contribuyeron a la edición de este libro. Muchos estudiantes de posgrado en Purdue contribuyeron con sugerencias y críticas. Baijian (Justin) Yang y Bo Sang recomendaron la incorporación de texto y figuras para ayudar a sus estudiantes a comprender mejor el material. Fred Baker, Ralph Droms y Dave Oran de Cisco contribuyeron en las ediciones previas. Lami Kaya sugirió cómo podían organizarse los capítulos sobre comunicaciones de datos e hizo muchas otras sugerencias valiosas. Agradezco especialmente a mi esposa y socia, Christine, cuyo cuidadoso proceso de edición y sus útiles sugerencias resultaron en muchas mejoras en todo el libro.

Douglas E. Comer

Acerca del autor

El doctor Douglas E. Comer es un experto en redes de computadoras, protocolos TCP/IP e Internet, reconocido a nivel internacional. Como uno de los investigadores que contribuyeron cuando se estaba formando Internet a finales de la década de 1970 y a principios de la de 1980, fue miembro del Consejo de arquitectura de Internet, el grupo responsable de guiar el desarrollo de Internet. También fue presidente del comité técnico de CSNET, miembro del comité ejecutivo de CSNET y presidente del Consejo de arquitectura de sistemas distribuidos de DARPA.

El doctor Comer se desempeña como consultor en la industria sobre el diseño de redes computacionales. Además de dar pláticas en universidades de Estados Unidos, cada año imparte conferencias para académicos y profesionales de redes en todo el mundo. El sistema operativo del doctor Comer, Xinu, junto con la implementación de los protocolos TCP/IP (ambos documentados en otros de sus libros), se han utilizado en productos comerciales.

Es profesor distinguido de ciencias computacionales en la Purdue University. Antes fungió como vicepresidente de investigación en Cisco Systems. Imparte cursos sobre redes, interconexión de redes, arquitectura de computadoras y sistemas operativos. En Purdue ha desarrollado innovadores laboratorios que ofrecen a los estudiantes la oportunidad de obtener experiencia práctica con sistemas operativos, redes y protocolos. Además de escribir una serie de exitosos libros técnicos que se han traducido a dieciséis idiomas, se desempeñó durante 20 años como editor para Norteamérica de la publicación científica *Software – Practice and Experience*. El doctor Comer es miembro del ACM.

Encontrará información adicional en la página web del autor en:

www.cs.purdue.edu/people/comer

PARTE I

Introducción a las redes y las aplicaciones de Internet

**Una descripción general
de las redes y la interfaz
que usan los programas
de aplicaciones para
comunicarse por Internet**

Capítulos

- 1 Introducción y descripción general**
- 2 Tendencias de Internet**
- 3 Aplicaciones de Internet y programación de redes**
- 4 Aplicaciones tradicionales de Internet**

Contenido del capítulo

- 1.1 Crecimiento de las redes de computadoras, 1
- 1.2 Por qué las redes parecen complejas, 2
- 1.3 Los cinco aspectos clave de las redes, 2
- 1.4 La Internet pública y la Internet privada, 6
- 1.5 Redes, interoperabilidad y estándares, 8
- 1.6 Suites de protocolos y modelos de distribución por capas, 9
- 1.7 Cómo pasan los datos a través de las capas, 11
- 1.8 Encabezados y capas, 12
- 1.9 ISO y el modelo de referencia OSI de siete capas, 13
- 1.10 El resto del libro, 14
- 1.11 Resumen, 14

1

Introducción y descripción general

1.1 Crecimiento de las redes de computadoras

Las redes de computadoras siguen creciendo de manera explosiva. Desde la década de 1970, la comunicación de las computadoras ha pasado de ser un tema de investigación esotérica a formar una parte esencial de la vida de todos. Las redes se usan en todos los ámbitos de cualquier empresa, incluyendo la publicidad, la producción, la planeación, la facturación, la distribución y la contabilidad. Por consecuencia, la mayoría de las corporaciones cuenta con varias redes tanto internas como externas. Las escuelas, de todos los niveles, desde primaria hasta posgrado, utilizan redes de computadoras para ofrecer a los estudiantes y profesores un acceso instantáneo a la información en línea. Las oficinas gubernamentales a nivel federal, estatal y municipal dependen de las redes, al igual que las organizaciones militares. En resumen, las redes de computadoras están presentes en todas partes.

El crecimiento y los usos de la Internet global[†] se encuentran entre los fenómenos más interesantes y emocionantes de las redes. En 1980, Internet era un proyecto de investigación que involucraba tan sólo unos cuantos sitios. En la actualidad, Internet ha crecido para convertirse en un sistema de comunicaciones que llega prácticamente a cualquier ciudad del mundo. Muchos usuarios tienen acceso a Internet de alta velocidad mediante módems de cable, DSL, tecnologías ópticas o inalámbricas.

La llegada y la utilización de las redes han cambiado drásticamente la economía. Las redes de datos han permitido que los individuos trabajen a distancia y han cambiado la comunicación de los negocios. Además, surgió toda una industria que desarrolla tecnologías, productos y servicios de redes. La importancia de las redes de computadoras ha hecho que cada vez más empresas busquen personas con experiencia en redes. Las empresas necesitan gente que planee, adquiera, instale, opere y administre los sistemas de hardware y software que constituyen sus redes de computadoras. La llegada de la

[†]A lo largo de este texto, seguiremos la convención de escribir *Internet* con “I” mayúscula para denotar a Internet global.

computación en la nube significa que la computación está pasando de las máquinas locales a los centros de datos remotos. Como resultado, las redes han afectado toda la programación de computadoras; los programadores ya no crean software para una sola computadora; desarrollan aplicaciones que se comunican a través de Internet.

1.2 Por qué las redes parecen complejas

Puesto que las redes de computadoras constituyen un campo que está cambiando en forma activa y rápida, el tema tiende a parecer complejo. Existen muchas tecnologías y cada una tiene características que la diferencian de las otras. Las empresas siguen creando productos y servicios basados en redes comerciales, a menudo mediante el uso de tecnologías nuevas y no convencionales. Por último, las redes parecen complejas ya que las tecnologías pueden combinarse e interconectarse de muchas formas.

Las redes de computadoras pueden ser especialmente confusas para un principiante, ya que no existe una sola teoría que explique la relación entre todas las partes. Varias organizaciones han creado estándares de redes, pero algunos son incompatibles con otros. Varias organizaciones y grupos de investigación han intentado definir modelos conceptuales que capturen la esencia y expliquen los matices entre los sistemas de hardware y software que forman parte de las redes, pero debido a que el conjunto de tecnologías es diverso y cambia con rapidez, los modelos resultan tan simples que no distinguen unos detalles de otros, o tan complejos que no ayudan a simplificar el tema.

La falta de consistencia en el campo produce un desafío más para los principiantes: en vez de existir una terminología uniforme para los conceptos de redes, hay varios grupos que intentan crear su propia terminología independiente. Los investigadores se aferran a la terminología científicamente precisa. Los grupos de marketing de las empresas involucradas, a menudo relacionan un producto con un término técnico que ya existe o inventan nuevos términos simplemente para diferenciar sus productos o servicios de los de la competencia. Por ende, los términos técnicos se confunden fácilmente con los nombres de productos populares. Para agregar aún más confusión, a veces los profesionales toman un término de una tecnología para referirse a una característica similar de otra tecnología. En consecuencia, además de un extenso conjunto de términos y acrónimos que contiene muchos sinónimos, la jerga de las redes contiene términos que comúnmente están abreviados, se utilizan de forma incorrecta o se asocian con productos comerciales.

1.3 Los cinco aspectos clave de las redes

Para dominar la complejidad en las redes es importante obtener una amplia formación que incluya cinco aspectos clave del tema:

- Aplicaciones de red y programación de redes
- Comunicaciones de datos
- Comutación de paquetes y tecnologías de redes
- Interconexión de redes con TCP/IP
- Otros conceptos y tecnologías de redes

1.3.1 Aplicaciones de red y programación de redes

Los servicios y las ubicaciones de red al que acceden los usuarios se entregan mediante diverso software de aplicación: un programa de aplicación en una computadora se comunica a través de una red con un programa de aplicación que se ejecuta en otra computadora. Las aplicaciones de red abarcan un amplio rango de servicios que incluye correo electrónico, envío o descarga de archivos, navegación Web, llamadas telefónicas de audio y de voz, acceso a bases de datos distribuidas y videoconferencias. Aunque cada aplicación ofrece un servicio específico con su propia interfaz de usuario, todas las aplicaciones pueden comunicarse a través de una sola red compartida. La disponibilidad de una red unificada que soporte todas las aplicaciones facilita en gran medida el trabajo de un programador, ya que éste sólo necesita aprender cómo funciona la interfaz para la red y un conjunto básico de funciones; este mismo conjunto de funciones se utiliza en todas las aplicaciones que se comunican a través de esa red.

Como veremos, es posible entender las aplicaciones de red e incluso es posible escribir código que se comunique a través de una red, sin tener que entender las tecnologías de hardware y de software que se utilizan para transferir datos de una aplicación a otra. Parecería que una vez que un programador domine la interfaz, no será necesario aprender más sobre redes. Sin embargo, la programación de redes es similar a la programación convencional. Aunque un programador convencional puede crear aplicaciones sin tener que entender el funcionamiento de los compiladores, los sistemas operativos o la arquitectura de la computadora, el conocimiento de los sistemas involucrados puede ayudarle a crear programas más confiables, precisos y eficientes. De manera similar, el conocimiento del sistema de redes involucrado permite a un programador escribir mejor código. En conclusión:

Un programador que comprende los mecanismos y las tecnologías de redes involucradas puede escribir aplicaciones de redes que sean más rápidas, más confiables y menos vulnerables.

1.3.2 Comunicaciones de datos

El término *comunicaciones de datos* se refiere al estudio de los mecanismos y tecnologías de bajo nivel que se utilizan para enviar información a través de un medio de comunicación físico, como un cable, una onda de radio o un haz de luz. Las comunicaciones de datos, que tienen que ver con las formas de usar los fenómenos físicos para transferir información, son principalmente el dominio de la ingeniería eléctrica. Los ingenieros diseñan y construyen un amplio rango de sistemas de comunicaciones. Muchas de las ideas básicas que necesitan los ingenieros se han derivado de las propiedades de la materia y la energía descubiertas por los físicos. Por ejemplo, más adelante veremos que las fibras ópticas utilizadas para la transferencia de datos de alta velocidad dependen de las propiedades de la luz y de su reflexión en un límite entre dos tipos de materia.

Como tratan con los conceptos físicos, las comunicaciones de datos pueden parecer algo irrelevantes a nuestra comprensión de las redes. En especial y debido a que muchos de los términos y conceptos se refieren a fenómenos físicos, el tema puede parecer útil sólo para los ingenieros que diseñan instalaciones de transmisión de bajo nivel. Por ejemplo, las técnicas de modulación que usan formas

físicas de energía (como la radiación electromagnética) para transportar información, parecen irrelevantes para el diseño y uso de los protocolos. Sin embargo, veremos que varios conceptos clave que surgen de las comunicaciones de datos influyen en el diseño de los protocolos de comunicación. En el caso de la modulación, el concepto del ancho de banda se relaciona directamente con el rendimiento de la red.

Como caso específico, las comunicaciones de datos introducen la noción de multiplexación, que permite combinar la información de varias fuentes para transmitirla a través de un medio compartido, para después separarla y entregarla en varios destinos. Veremos que la multiplexación no se limita a la transmisión física: la mayoría de los protocolos incorporan cierta forma de multiplexación. De igual forma, el concepto de la encriptación introducido en las comunicaciones de datos forma la base de la mayoría de la seguridad de redes. Podemos resumir su importancia de la siguiente forma:

Aunque tratan con muchos detalles de bajo nivel, las comunicaciones de datos ofrecen una base de conceptos sobre los que se construyen el resto de las redes.

1.3.3 Comutación de paquetes y tecnologías de redes

En la década de 1960, un nuevo concepto revolucionó las comunicaciones de datos: la comutación de paquetes. Las primeras redes de comunicaciones habían evolucionado del telégrafo y los sistemas telefónicos que conectaban un par de cables entre dos partes para formar un circuito de comunicación. Aunque la conexión mecánica de cables se estaba reemplazando por interruptores electrónicos, el paradigma a resolver seguía siendo el mismo: formar un circuito y luego enviar información a través de éste. La comutación de paquetes cambió las redes de una manera fundamental y proporcionó las bases para la red Internet moderna: en vez de formar un circuito especializado, la comutación de paquetes permite que varios emisores transmitan datos sobre una red compartida. La comutación de paquetes se basa en los mismos mecanismos de comunicaciones de datos fundamentales del sistema telefónico, pero usa estos mecanismos de una nueva manera. La comutación de paquetes divide los datos en bloques pequeños, conocidos como paquetes, e incluye una identificación del receptor al que va destinado cada paquete. Cada uno de los dispositivos ubicados a lo largo de la red tiene información sobre cómo llegar a cada destino posible. Cuando un paquete llega a uno de los dispositivos, éste selecciona una ruta a través de la cual va a reenviar el paquete para que finalmente llegue al destino correcto.

En teoría, la comutación de paquetes es sencilla. Sin embargo, hay muchos diseños posibles, dependiendo de las respuestas a las preguntas básicas. ¿Cómo debe identificarse un destino y cómo puede un emisor encontrar la identificación de ese destino? ¿Qué tan grande debe ser un paquete? ¿Cómo puede una red reconocer dónde termina un paquete y dónde comienza otro? Si muchas computadoras están enviando información a través de una red, ¿cómo pueden coordinarse para asegurar que cada una reciba una oportunidad igual para enviar? ¿Cómo puede la comutación de paquetes adaptarse a las redes inalámbricas? ¿Cómo pueden designarse las tecnologías de comutación de paquetes para cumplir varios requerimientos de velocidad, distancia y costos? Se han propuesto muchas respuestas y se han

creado muchas tecnologías de conmutación de paquetes. De hecho, cuando estudiamos las redes de conmutación de paquetes, es posible llegar a una conclusión fundamental:

Como cada tecnología de red se crea para cumplir con varios requerimientos de velocidad, distancia y costo económico específicos, existen muchas tecnologías de conmutación de paquetes. Las tecnologías difieren en cuanto a los detalles, como el tamaño de los paquetes y el método utilizado para identificar un receptor.

1.3.4 Interconexión de redes con TCP/IP

En la década de 1970, surgió otra revolución en las redes de computadoras: el concepto de una Internet. Muchos investigadores buscaron una sola tecnología de conmutación de paquetes que pudiera manejar todas las necesidades. En 1973, Vinton Cerf y Robert Kahn observaron que ninguna tecnología de conmutación de paquetes podría satisfacer todas las necesidades, en particular ante la posibilidad de crear tecnologías de baja capacidad para hogares u oficinas a un costo bastante bajo. La solución era dejar de intentar encontrar una única solución y en su lugar probar interconectando muchas tecnologías de conmutación de paquetes como un todo funcional. Propusieron desarrollar un conjunto de estándares para dicha interconexión; los estándares resultantes se conocieron como *suite de protocolos TCP/IP* (por lo general abreviada como TCP/IP). El concepto, que ahora se conoce como interconexión de redes, es en extremo poderoso. Proporciona la base de la Internet global y forma una parte importante del estudio de las redes de computadoras.

Una de las principales razones del éxito de los estándares TCP/IP recae en su tolerancia a la heterogeneidad. En vez de tratar de dictar los detalles sobre las tecnologías de conmutación de paquetes, como los tamaños de los paquetes o el método usado para identificar un destino, TCP/IP toma un enfoque de virtualización que define paquetes y esquemas de identificación independientes de la red y luego especifica cómo se asignan los paquetes virtuales a cada red subyacente.

Resulta interesante que la habilidad de TCP/IP para tolerar las nuevas redes de conmutación de paquetes sea una de las principales motivaciones para seguir evolucionando las tecnologías de conmutación de paquetes. A medida que Internet crece, las computadoras se vuelven más poderosas y las aplicaciones envían más datos, en especial fotos y video. Para dar cabida a los nuevos usos, los ingenieros inventan tecnologías que pueden transmitir más datos y procesar más paquetes en un tiempo dado. A medida que se inventan, las nuevas tecnologías se incorporan a Internet junto con las tecnologías existentes. Puesto que Internet tolera la heterogeneidad, los ingenieros de Internet pueden experimentar con nuevas tecnologías sin perturbar las redes existentes. En conclusión:

Internet se forma mediante la interconexión de varias redes de conmutación de paquetes. La interconexión de redes es mucho más poderosa que una sola tecnología de redes, ya que la metodología permite incorporar nuevas tecnologías en cualquier momento sin requerir el reemplazo general de las viejas tecnologías.

1.3.5 Otros conceptos y tecnologías de redes

Además del hardware y los protocolos utilizados para crear redes, hay muchas otras tecnologías que ofrecen posibilidades importantes. Por ejemplo, hay tecnologías que analizan el rendimiento de la red, que permiten que los multimedios y la telefonía IP se lleven a cabo sobre una estructura de conmutación de paquetes y que mantienen las redes seguras. La infraestructura convencional para la administración de redes y las *redes definidas por software* (SDN) permiten a los administradores configurar y controlar las redes, por otra parte, la *Internet de las cosas* hace posible que los sistemas integrados se comuniquen a través de Internet.

Las redes definidas por software y la Internet de las cosas sobresalen debido a que son conceptos nuevos y han ganado una considerable atención con rapidez. SDN propone un paradigma completamente nuevo para el control y la administración de sistemas de redes. El diseño tiene implicaciones económicas y podría promover un cambio considerable en la forma en que operan las redes.

Otro cambio en Internet tiene que ver con el paso de la comunicación que involucra a uno o más humanos, a la Internet de cosas que permite que dispositivos autónomos se comuniquen sin que haya un humano de por medio. Por ejemplo, las tecnologías de automatización de hogares permitirán a los aparatos optimizar los costos de energía al programarse para operar en horarios en que las tarifas son bajas (por ejemplo, en la noche). Como resultado, el número de dispositivos en Internet se expandirá de manera considerable.

1.4 La Internet pública y la Internet privada

Aunque funciona como un solo sistema de comunicaciones, Internet consta de varias partes que pertenecen a y son operadas por individuos u organizaciones. Para ayudar a aclarar la propiedad y el propósito de Internet, la industria de las redes usa los términos *red pública* y *red privada*.

1.4.1 Red pública

Una *red pública* opera como un servicio disponible para suscriptores. Cualquier individuo o corporación que pague la cuota de suscripción puede usar la red. Una compañía que ofrece servicios de comunicaciones se conoce como *proveedor de servicios*. El concepto de un proveedor de servicios es bastante amplio y se extiende más allá de los *proveedores de servicios de Internet (ISP)*. De hecho, la terminología se originó con empresas que ofrecían el servicio telefónico de voz analógico. En conclusión:

Una red pública es propiedad de un proveedor de servicios, y ofrece el servicio a cualquier individuo u organización que pague la cuota de suscripción.

Es importante comprender que el término *público* se refiere a la disponibilidad general del servicio y no a los datos transferidos. En especial, muchas redes públicas siguen estrictas regulaciones gubernamentales que exigen que el proveedor proteja la comunicación de intromisiones imprevistas. En conclusión:

El término público significa que hay un servicio disponible para el público en general; los datos que se transfieren a través de una red pública no necesariamente se revelan a los usuarios externos.

1.4.2 Red privada

Una *red privada* es controlada por un grupo específico. Aunque parece algo simple, la distinción entre las partes públicas y privadas de Internet puede ser sutil debido a que el control no siempre implica la propiedad. Por ejemplo, si una empresa renta un circuito de datos a un proveedor y luego restringe el uso del circuito al tráfico de la empresa, el circuito se vuelve parte de la red privada de la empresa. En conclusión:

Se dice que una red es privada si el uso de ésta se restringe a un solo grupo. Una red privada puede incluir circuitos que se rentan a un proveedor de servicios.

Los distribuidores de equipos de red dividen las redes privadas en cuatro categorías:

- Consumidor
- Pequeña oficina en casa (SOHO)
- Pequeña y mediana empresa (SMB)
- Grandes empresas

Como las categorías se relacionan con las ventas y el marketing, la definición de la terminología es imprecisa. Aunque es posible dar una descripción cualitativa de cada tipo, no podemos encontrar una definición exacta. Por ende, los siguientes párrafos ofrecen una caracterización global de tamaño y propósito, en vez de medidas detalladas.

Consumidor. Una de las formas menos costosas de una red privada consiste en una red que pertenece a un individuo; si el individuo compra un conmutador de red económico y lo usa para conectar una impresora a una computadora de escritorio, el individuo ha creado una red privada. De manera similar, un consumidor podría comprar e instalar un *enrutador inalámbrico* para habilitar conexiones tipo Wi-Fi en su hogar. Dicha instalación constituye una red privada.

Pequeña oficina en casa (SOHO). Una red SOHO es ligeramente más grande que una red de consumidor. Una red SOHO común conecta dos o más computadoras, una o más impresoras, un enrutador que conecta la oficina a Internet y posiblemente otros dispositivos, como una caja registradora o máquina de verificación de tarjetas de crédito. La mayoría de las instalaciones SOHO incluyen una fuente de poder con batería de respaldo y otros mecanismos que permiten a la red funcionar sin interrupción.

Pequeña y mediana empresa (SMB). Una red SMB puede conectar muchas computadoras en varias oficinas de un edificio, y también puede incluir a las computadoras en una instalación de producción

(por ejemplo, en un almacén de embarques). A menudo una red SMB contiene varios commutadores de red interconectados por enruteadores, usa una conexión a Internet de banda ancha de mayor capacidad y puede incluir varios dispositivos inalámbricos que proporcionen conexiones Wi-Fi.

Grandes empresas. Una red empresarial grande proporciona la infraestructura de TI necesaria para una corporación importante. Por lo general, una red empresarial grande conecta varios sitios geográficos con varios edificios en cada sitio, usa muchos commutadores y enruteadores, y tiene dos o más conexiones a Internet de alta velocidad a nivel mundial. Comúnmente las redes empresariales incluyen tecnologías tanto alámbricas como inalámbricas.

En conclusión:

Una red privada puede servir a un solo consumidor, a una oficina casera, a un negocio de pequeño a mediano, o a una gran empresa.

1.5 Redes, interoperabilidad y estándares

La comunicación siempre implica al menos dos entidades, una que envía información y otra que la recibe. De hecho, podemos ver que la mayoría de los sistemas de comunicaciones de commutación de paquetes contienen entidades intermedias (es decir, dispositivos que reenvían paquetes). El punto importante es tener en cuenta que, para que la comunicación sea exitosa, todas las entidades en una red deben estar de acuerdo en cómo se va a representar y comunicar la información. Los acuerdos de comunicación implican muchos detalles. Por ejemplo, cuando dos entidades se comunican a través de una red alámbrica, ambos lados deben estar de acuerdo en los voltajes que se van a usar, en la forma exacta en que se deben usar las señales eléctricas para representar los datos, en los procedimientos utilizados para iniciar y llevar a cabo la comunicación, y en el formato de los mensajes.

Usamos el término *interoperabilidad* para referirnos a la habilidad de dos entidades de comunicarse, y decimos que si dos entidades pueden comunicarse sin que haya malos entendidos están *interoperando* correctamente. Para asegurar que todas las partes de la comunicación estén de acuerdo con los detalles y sigan el mismo conjunto de reglas, es necesario que se establezca un conjunto exacto de especificaciones. En conclusión:

La comunicación implica varias entidades que deben acordar los detalles, que van desde el voltaje eléctrico utilizado para el formato y significado de los mensajes. Para asegurar que las entidades puedan interoperar correctamente, se escriben las reglas para todos los aspectos de la comunicación.

Siguiendo la terminología de la diplomacia, usamos el término *protocolo de comunicaciones, protocolo de redes* o *protocolo* en general, para referirnos a una especificación para la comunicación en red. Un protocolo dado puede especificar detalles de bajo nivel, como el tipo de transmisión de radio que se utiliza en una red inalámbrica, o describir un mecanismo de alto nivel como los mensajes que intercambian dos programas de aplicaciones. Dijimos que un protocolo puede definir el procedimiento a

seguir durante un intercambio. Uno de los aspectos más importantes de un protocolo incluye situaciones en las que ocurre un error o una condición inesperada. Por ende, lo común es que un protocolo explique la acción apropiada a seguir para cada condición anormal posible (por ejemplo, se espera una respuesta pero no llega). En conclusión:

Un protocolo de comunicación determina los detalles para un aspecto de la comunicación entre computadoras, incluyendo las acciones a seguir cuando surjan errores o situaciones inesperadas. Un protocolo dado puede especificar detalles de bajo nivel, como el voltaje y las señales a utilizar, o los elementos de alto nivel, como el formato de los mensajes que intercambian los programas de aplicaciones.

1.6 Suites de protocolos y modelos de distribución por capas

Para asegurar que el sistema de comunicaciones resultante sea completo y eficiente, es necesario construir cuidadosamente el conjunto de protocolos. Para evitar duplicar el esfuerzo, cada protocolo debe encargarse de una parte de la comunicación que no manejan los otros protocolos. ¿Cómo podemos garantizar que los protocolos funcionarán bien en su conjunto? La respuesta recae en un plan de diseño general: en vez de crear cada protocolo por separado, los protocolos son diseñados en conjuntos cooperativos conocidos como *suites* o *familias*. Cada protocolo de una suite maneja un aspecto de la comunicación; en conjunto, los protocolos de la suite cubren todos los aspectos de la comunicación, incluyendo las fallas de hardware y otras condiciones excepcionales. Además, toda la suite está diseñada para permitir que los protocolos trabajen juntos en forma eficiente.

La abstracción fundamental utilizada para recolectar protocolos en un todo unificado se conoce como *modelo de distribución por capas*. En esencia, este modelo describe cómo pueden particionarse todos los aspectos de un problema de comunicación en piezas que funcionen juntas. Cada pieza se conoce como *capa*; la terminología surge debido a que los protocolos de una suite están organizados en una secuencia lineal. Al dividir los protocolos en capas es más fácil para los diseñadores y los implementadores del protocolo manejar la complejidad, ya que les permite concentrarse en un aspecto específico de la comunicación.

La figura 1.1 ilustra el concepto al mostrar el modelo de distribución por capas utilizado con los protocolos de Internet. La apariencia visual de las figuras utilizadas para ilustrar la distribución por capas condujo al término coloquial *pila*. El término se utiliza para referirse al software de protocolo de una computadora, como en la pregunta “¿Esa computadora ejecuta la pila de protocolos TCP/IP?”.

Los capítulos posteriores nos ayudarán a comprender la distribución por capas, explicando los protocolos en detalle. Por ahora basta con aprender sobre el propósito general de cada capa y cómo es que se utilizan los protocolos para la comunicación. Las siguientes secciones sintetizan el rol de las capas; una sección posterior examina cómo pasan los datos a través de las capas cuando las computadoras se comunican.

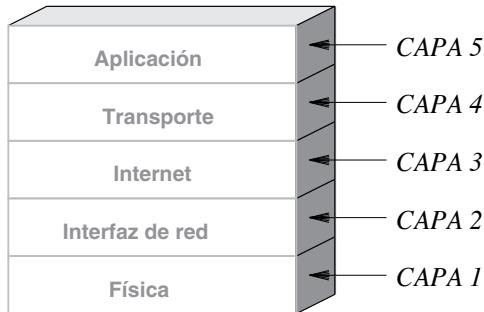


Figura 1.1 El modelo de distribución por capas usado con los protocolos de Internet (TCP/IP).

Capa 1: física

Los protocolos en la capa *física* especifican los detalles sobre el medio de transmisión a utilizar y el hardware asociado. Todas las especificaciones relacionadas con las propiedades eléctricas, las frecuencias de radio y las señales pertenecen a la capa 1.

Capa 2: interfaz de red[†] o MAC

Los protocolos en la capa *MAC* especifican los detalles de la comunicación a través de una sola red y la interfaz entre el hardware de red y la capa 3, que por lo general se implementa en el software. También pertenecen a la capa 2 las especificaciones sobre las direcciones de red y el tamaño máximo del paquete que puede soportar una red, los protocolos utilizados para acceder al medio que se usa y el direccionamiento de hardware.

Capa 3: Internet

Los protocolos de la capa 3 forman la base fundamental de Internet. Los protocolos de esta capa especifican la comunicación entre dos computadoras a través de Internet (es decir, a través de varias redes interconectadas). La estructura de direccionamiento de Internet, el formato de los paquetes de Internet, el método para dividir un paquete grande de Internet en paquetes más pequeños para la transmisión, y los mecanismos para reportar errores, pertenecen a la capa 3.

Capa 4: transporte

Los protocolos en la capa de *transporte* proporcionan la comunicación entre un programa de aplicación de una computadora y un programa de aplicación en otra computadora. Las especificaciones que controlan la velocidad máxima con la que un receptor puede aceptar datos, los mecanismos para evitar la congestión de la red y las técnicas para asegurar que todos los datos se reciban en el orden correcto pertenecen a la capa 4.

[†] Aunque el diseñador de TCP/IP usó el término *interfaz de red* y algunas organizaciones de estándares prefirieron el término *enlace de datos* para la capa 2, el término *MAC* se ha vuelto muy popular en la industria.

Capa 5: aplicación

Los protocolos de la capa superior de la pila TCP/IP especifican cómo interactúan un par de aplicaciones cuando se comunican. Los protocolos de la capa 5 brindan los detalles sobre el formato y el significado de los mensajes que las aplicaciones pueden intercambiar, así como los procedimientos a seguir durante la comunicación. En esencia, cuando un programador crea una aplicación que se comunica a través de una red, lo que está haciendo es crear un protocolo de la capa 5. Las especificaciones de intercambio de correo electrónico, la transferencia de archivos, la navegación Web, el servicio telefónico de voz, las apps de teléfonos inteligentes y la videoconferencia pertenecen a la capa 5.

1.7 Cómo pasan los datos a través de las capas

La distribución por capas no es simplemente un concepto abstracto que nos ayuda a comprender los protocolos. Las implementaciones de los protocolos siguen el modelo de distribución por capas, al pasar de la salida de un protocolo en una capa a la entrada de un protocolo en la siguiente capa. Además, para lograr una mayor eficiencia, en vez de copiar todo un paquete, algunos protocolos de las capas adyacentes pasan un apuntador sobre el paquete. Así, los datos se pasan entre las capas de manera eficiente.

Para comprender cómo operan los protocolos, considere dos computadoras conectadas a una red. La figura 1.2 ilustra los protocolos distribuidos por capas en las dos computadoras. Como se muestra en la figura, cada computadora contiene un conjunto de protocolos distribuidos por capas.

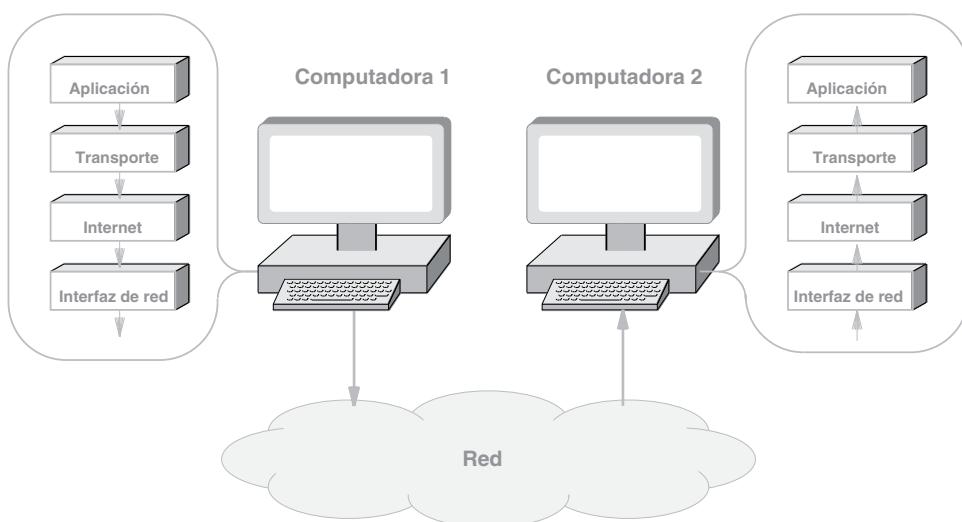


Figura 1.2 Ilustración sobre cómo pasan los datos entre las capas de protocolos cuando las computadoras se comunican a través de una red. Cada computadora tiene un conjunto de protocolos distribuidos por capas, y los datos pasan a través de cada capa.

Cuando una aplicación envía datos, éstos se colocan en un paquete, y el paquete saliente desciende a través de cada capa de protocolos. Una vez que pasa por todas las capas de protocolos en la computadora emisora, el paquete sale de ésta y se transmite a través de la red física.[†] Cuando llega a la computadora receptora, el paquete asciende a través de las capas de protocolos. Si la aplicación en la computadora emisora envía una respuesta, se invierte el proceso. Es decir, una respuesta desciende a través de las capas para salir de la computadora y luego asciende por las capas de la computadora que recibe la respuesta.

1.8 Encabezados y capas

Aprenderemos que cada capa del software de protocolo realiza cálculos para asegurar que los mensajes lleguen según lo esperado. Para realizar dicho cálculo, el software de protocolo de ambas máquinas debe intercambiar información. Para ello, cada capa en la computadora emisora antepone información adicional en el paquete; la capa del protocolo correspondiente de la computadora receptora toma y usa la información adicional.

La información adicional que un protocolo agrega a un paquete se conoce como *encabezado*. Para comprender cómo aparecen los encabezados, piense en un paquete que viaja por la red entre las dos computadoras de la figura 1.2. El software de protocolo agrega los encabezados a medida que los datos descienden por las capas de la computadora emisora. Es decir, la capa de transporte antepone un encabezado y luego la capa de Internet le antepone otro encabezado, y así sucesivamente. Por consiguiente, si observamos un paquete recorriendo la red, los encabezados aparecerán en el orden que ilustra la figura 1.3.

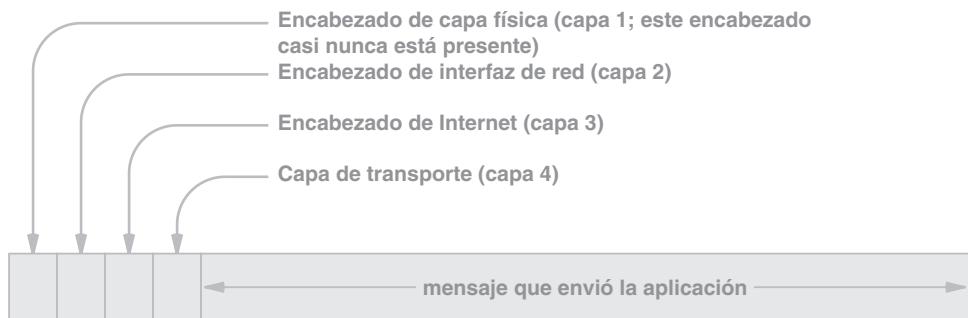


Figura 1.3 Los encabezados de protocolos anidados que aparecen en un paquete a medida que éste viaja por una red entre dos computadoras. En el diagrama, el comienzo del paquete (el primer bit que se envía a través de la red subyacente) se muestra del lado izquierdo.

Aunque la figura muestra los encabezados como del mismo tamaño, en la práctica los encabezados no son de tamaño uniforme, y el encabezado de la capa física es opcional. Comprenderemos la razón de las disparidades de tamaño cuando examinemos el contenido del encabezado. De igual forma, veremos

[†] La figura 1.2 muestra sólo una red. Cuando estudiemos la arquitectura de Internet, aprenderemos sobre los dispositivos intermedios conocidos como *enrutadores* y cómo operan los protocolos distribuidos por capas en Internet.

que la capa física por lo general especifica cómo se usan las señales para transmitir datos, lo que significa que el paquete no contiene un encabezado de capa física explícito.

1.9 ISO y el modelo de referencia OSI de siete capas

Al mismo tiempo que se desarrollaban los protocolos de Internet, dos grandes organismos de estándares formaron en conjunto un modelo de referencia alternativo. También crearon el conjunto OSI de protocolos de interconexión de redes como competidores de los protocolos de Internet. Las organizaciones son:

- Organización Internacional de Estándares (ISO)
- Sector de Estandarización de Telecomunicaciones de la Unión Internacional de Telecomunicaciones (ITU)[†]

El modelo ISO de distribución de capas se conoce como el *modelo de referencia de interconexión de sistemas abiertos de siete capas*. La confusión surge en la terminología debido a que el acrónimo de los protocolos (OSI) y el acrónimo de la organización (ISO) son similares. Es probable que encuentre referencias tanto para el *modelo OSI de siete capas* como para el *modelo ISO de siete capas*. La figura 1.4 ilustra las siete capas en el modelo.

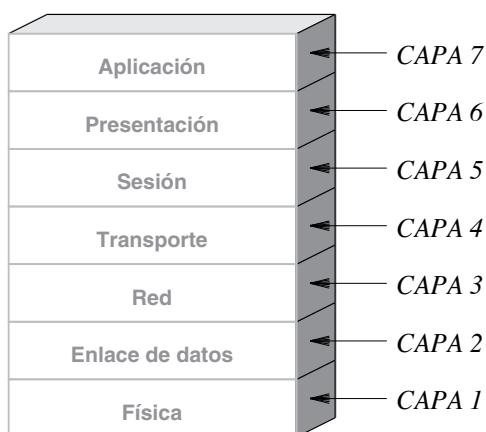


Figura 1.4 El modelo OSI de siete capas estandarizado por ISO.

Con el tiempo se volvió claro que la tecnología TCP/IP era técnicamente superior a OSI, y en cuestión de unos cuantos años se terminaron los esfuerzos por desarrollar e implementar protocolos OSI. Los organismos de estándares se quedaron con el modelo de siete capas, que no incluía una capa de Internet. En consecuencia, durante muchos años los defensores del modelo de siete capas han intentado

[†] Cuando se creó el estándar por primera vez, la ITU se conocía como *Comité Consultativo Internacional Telegráfico y Telefónico (CCITT)*.

estirar las definiciones para adaptarse a TCP/IP. Argumentan que la capa tres podría considerarse una capa de Internet y que podrían colocarse unos cuantos protocolos de apoyo en las capas cinco y seis. Tal vez la parte más irónica de la historia sea que muchos departamentos de marketing e incluso los ingenieros todavía se refieren a las aplicaciones como *protocolos de capa 7*, aun cuando saben que los protocolos de Internet sólo usan cinco capas, por lo que las capas cinco y seis de los protocolos ISO no se utilizan y son innecesarias.

1.10 El resto del libro

Este libro se divide en cinco partes principales. Después de una breve introducción, los capítulos de la primera parte presentan las aplicaciones de red y la programación de red. Recomendamos a los lectores que tengan acceso a una computadora, que creen y usen programas de aplicaciones que usen Internet mientras leen el texto. Las cuatro partes restantes explican cómo funcionan las tecnologías subyacentes. La segunda parte describe las comunicaciones de datos y la transmisión de la información. Explica cómo puede usarse la energía eléctrica y electromagnética para transportar información a través de cables o del aire, y muestra cómo se transmiten los datos.

La tercera parte del libro se enfoca en la conmutación y la tecnología de paquetes. Explica por qué las redes de computadoras usan paquetes y describe el formato general de éstos, analiza cómo se codifican para la transmisión y muestra cómo se reenvía cada paquete a través de una red hasta su destino. La tercera parte del libro también presenta las categorías básicas de redes de computadoras, como las *redes de área local* (LAN) y las *redes de área amplia* (WAN). Los capítulos describen las propiedades de cada categoría y describen las tecnologías de ejemplo.

La cuarta parte del texto cubre las interconexiones de redes y la suite de protocolos de Internet TCP/IP asociada. Esta parte describe la estructura de Internet y los protocolos TCP/IP. Explica el esquema de direccionamiento IP y la relación entre las direcciones de Internet y las direcciones del hardware implicado. También habla sobre el enrutamiento en Internet y los protocolos de enrutamiento. La cuarta parte incluye una descripción de varios conceptos fundamentales, incluyendo al encapsulamiento, la fragmentación, el control de la congestión y el flujo, las conexiones virtuales, el direccionamiento IPv4 e IPv6, la traducción de direcciones, el desarrollo de entornos de programación (*bootstrapping*) y varios protocolos de soporte.

La quinta parte del libro cubre una variedad de temas restantes que pertenecen a la red como un todo, en vez de partes individuales. Después de un capítulo sobre el rendimiento de las redes, los capítulos cubren las tecnologías emergentes, la seguridad de redes, la administración de redes y la reciente aparición de las *redes definidas por software* además de la *Internet de cosas*.

1.11 Resumen

El extenso conjunto de tecnologías, productos y esquemas de interconexión hacen de las redes un tema complejo. Hay cinco aspectos clave: aplicaciones de redes y programación de redes, comunicaciones de datos, conmutación de paquetes y tecnologías de redes, interconexión de redes con TCP/IP, y temas que se aplican a través de las capas, como la seguridad y la administración de redes.

Puesto que hay varias entidades involucradas en la comunicación, éstas deben estar de acuerdo en los detalles, incluyendo las características eléctricas como el voltaje, así como el formato y el significado de todos los mensajes. Para asegurar la interoperabilidad, cada entidad se construye para obedecer un conjunto de protocolos de comunicaciones que especifican todos los detalles necesarios para la comunicación. Para asegurar que los protocolos funcionen juntos y se encarguen de todos los aspectos de la comunicación, se diseña al mismo tiempo todo un conjunto de protocolos. La abstracción central sobre la que se construyen los protocolos se conoce como *modelo de distribución por capas*. La distribución por capas ayuda a reducir la complejidad al permitir que un ingeniero de redes se centre en un aspecto específico de la comunicación, sin preocuparse sobre los demás aspectos. Los protocolos TCP/IP que se utilizan en Internet siguen un modelo de referencia de cinco capas; las compañías telefónicas y la Organización Internacional de Estándares propusieron un modelo de referencia de siete capas.

EJERCICIOS

- 1.1** Haga una búsqueda en la Web para identificar las razones del crecimiento de Internet en años recientes.
- 1.2** Haga una lista con diez industrias que dependan de las redes de computadoras.
- 1.3** De acuerdo con el libro, ¿es posible desarrollar aplicaciones de Internet sin necesidad de comprender la arquitectura y las tecnologías de ésta? Respalde su respuesta.
- 1.4** ¿A qué aspectos de las redes se refieren las *comunicaciones de datos*?
- 1.5** ¿Qué es la commutación de paquetes y por qué es relevante para Internet?
- 1.6** Proporcione una breve historia de Internet, describiendo cómo y cuándo comenzó.
- 1.7** ¿Qué es interoperabilidad, y por qué es especialmente importante en Internet?
- 1.8** ¿Qué es un protocolo de comunicación? Conceptualmente, ¿cuáles son los dos aspectos de la comunicación que especifica un protocolo?
- 1.9** ¿Qué es una suite de protocolos y cuál es la ventaja de una suite?
- 1.10** Describa el modelo de distribución por capas de TCP/IP y explique cómo se derivó.
- 1.11** Haga una lista con las capas en el modelo TCP/IP y proporcione una breve explicación de cada una.
- 1.12** Explique cómo se agregan y quitan encabezados a medida que pasan los datos a través de una pila de protocolos distribuidos por capas.
- 1.13** Haga una lista con las principales organizaciones que crean estándares para comunicaciones de datos y redes de computadoras.

Contenido del capítulo

- 2.1 Introducción, 17
- 2.2 Compartición de recursos, 17
- 2.3 Crecimiento de Internet, 18
- 2.4 De la compartición de recursos a la comunicación, 21
- 2.5 Del texto al multimedia, 21
- 2.6 Tendencias recientes, 22
- 2.7 De las computadoras personales a la computación en la nube, 23
- 2.8 Resumen, 24

2

Tendencias de Internet

2.1 Introducción

Este capítulo trata sobre cómo han cambiado las redes de datos e Internet desde su creación. El capítulo comienza con una breve historia de Internet, resaltando algunas de las primeras motivaciones que hubo para su creación. Describe cómo el enfoque fue pasando de compartir instalaciones centralizadas a sistemas de información totalmente distribuidos.

Los capítulos posteriores de esta parte del libro profundizan en el tema mediante un análisis de las aplicaciones específicas de Internet. Además de describir los paradigmas de comunicación disponibles en Internet, los capítulos explican la interfaz de programación que usan las aplicaciones de Internet para comunicarse.

2.2 Compartición de recursos

Las primeras redes de computadora se diseñaron cuando las computadoras eran grandes y costosas, y la principal motivación era compartir recursos. Como ejemplo, tenemos que las redes se idearon como un medio para conectar a varios usuarios que contaban con una pantalla y teclado a una computadora centralizada de gran tamaño. Las redes posteriores permitían que varios usuarios compartieran dispositivos periféricos, como las impresoras. En conclusión:

Las primeras redes de computadoras se diseñaron para permitir la compartición de recursos centralizados y de alto costo.

En la década de 1960, la *Agencia de Proyectos de Investigación Avanzados (ARPA)*,[†] una agencia del Departamento de Defensa de Estados Unidos, estaba especialmente interesada en descubrir formas de compartir diversos recursos. Los investigadores necesitaban computadoras poderosas, y las computadoras eran demasiado costosas. El presupuesto de la ARPA no era suficiente para financiar muchas computadoras. Por ende, la ARPA comenzó a investigar las redes de datos: en vez de comprar una computadora para cada proyecto, la ARPA planeaba interconectar todas las computadoras con una red de datos y crear software que permitiera a un investigador usar la computadora que se adaptara mejor para realizar cierta tarea.

La ARPA reunió algunas de las mejores mentes disponibles, las concentró en la investigación de redes y contrató empresas para transformar los diseños en un sistema funcional conocido como *ARPA-NET*. La investigación resultó ser revolucionaria. El equipo de investigación optó por seguir una metodología conocida como *comutación de paquetes*, la cual se convirtió en la base de las redes de datos y de Internet.[‡] La ARPA continuó la tarea financiando el proyecto de investigación de Internet. Durante la década de 1980, Internet se expandió como un esfuerzo de investigación y durante la década de 1990 se convirtió en un éxito comercial.

2.3 Crecimiento de Internet

En menos de 40 años, Internet pasó de ser un prototipo de investigación que conectaba un grupo de sitios, a un sistema global de comunicaciones que se extiende a todos los países del mundo. La tasa de crecimiento ha sido fenomenal. El gráfico de la figura 2.1 ilustra el crecimiento del número de computadoras conectadas a Internet entre 1981 y 2012.

El eje y del gráfico muestra valores que van de cero hasta novecientos millones de computadoras. Las escalas lineales pueden ser engañosas, ya que ocultan pequeños detalles. Por ejemplo, el gráfico no muestra los detalles sobre el crecimiento preliminar de Internet, y da la apariencia de que Internet no comenzó a crecer sino hasta 1996 aproximadamente y que la mayor parte del crecimiento ocurrió durante los últimos años. De hecho, el ritmo promedio con el que se agregan nuevas computadoras a Internet llegó a más de una por segundo en 1998 y se ha acelerado desde entonces. Para 2007, se agregaban más de dos computadoras a Internet por segundo. Para comprender el ritmo preliminar de crecimiento vea la figura 2.2, donde se utiliza una escala logarítmica.

[†] En varias ocasiones la agencia ha agregado la palabra *Defensa* y utiliza el acrónimo *DARPA*.

[‡] El capítulo 13 explica con detalle la comutación de paquetes.

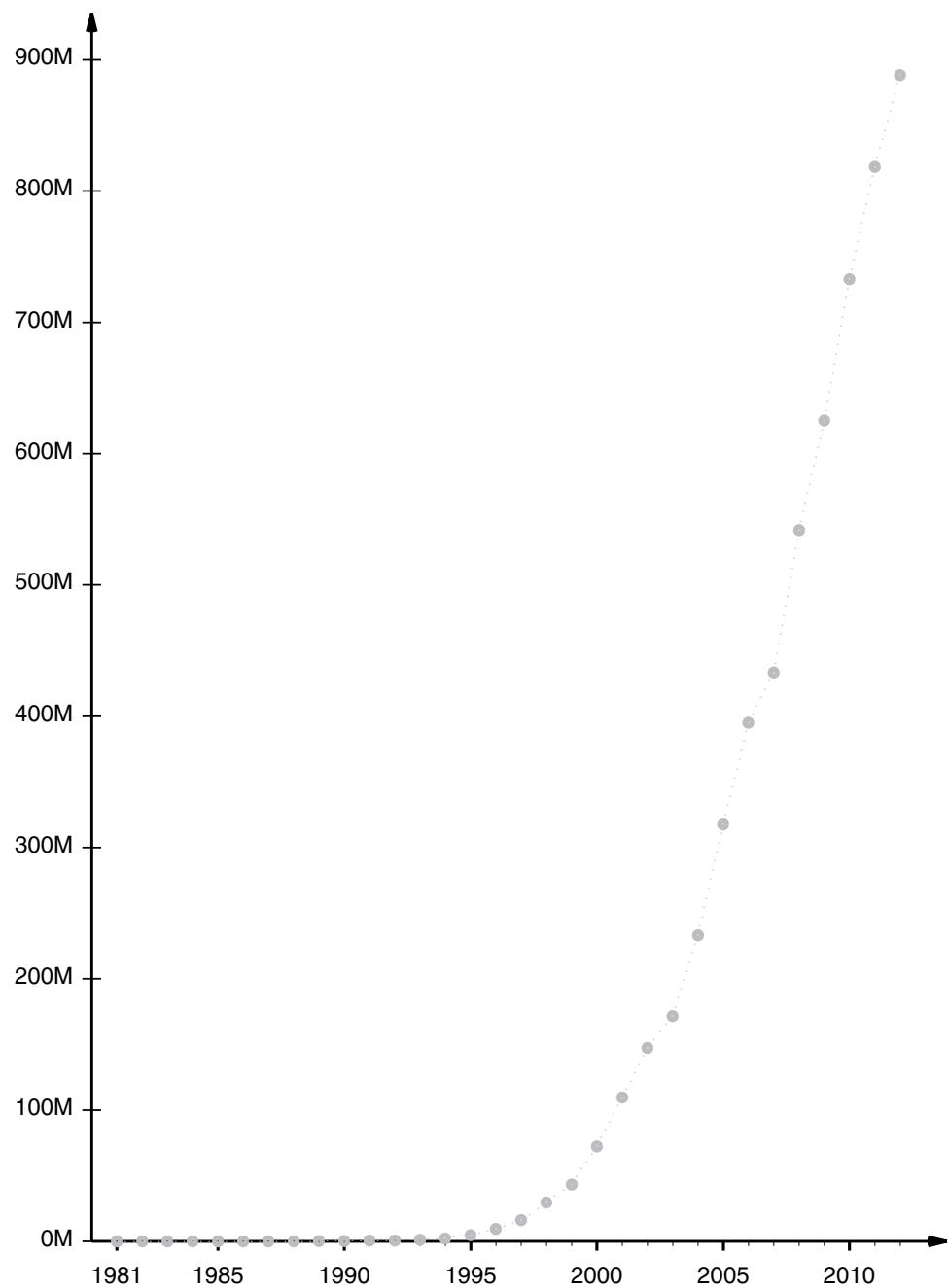


Figura 2.1 Crecimiento de Internet de 1981 a 2012, representado por el número de computadoras en Internet.

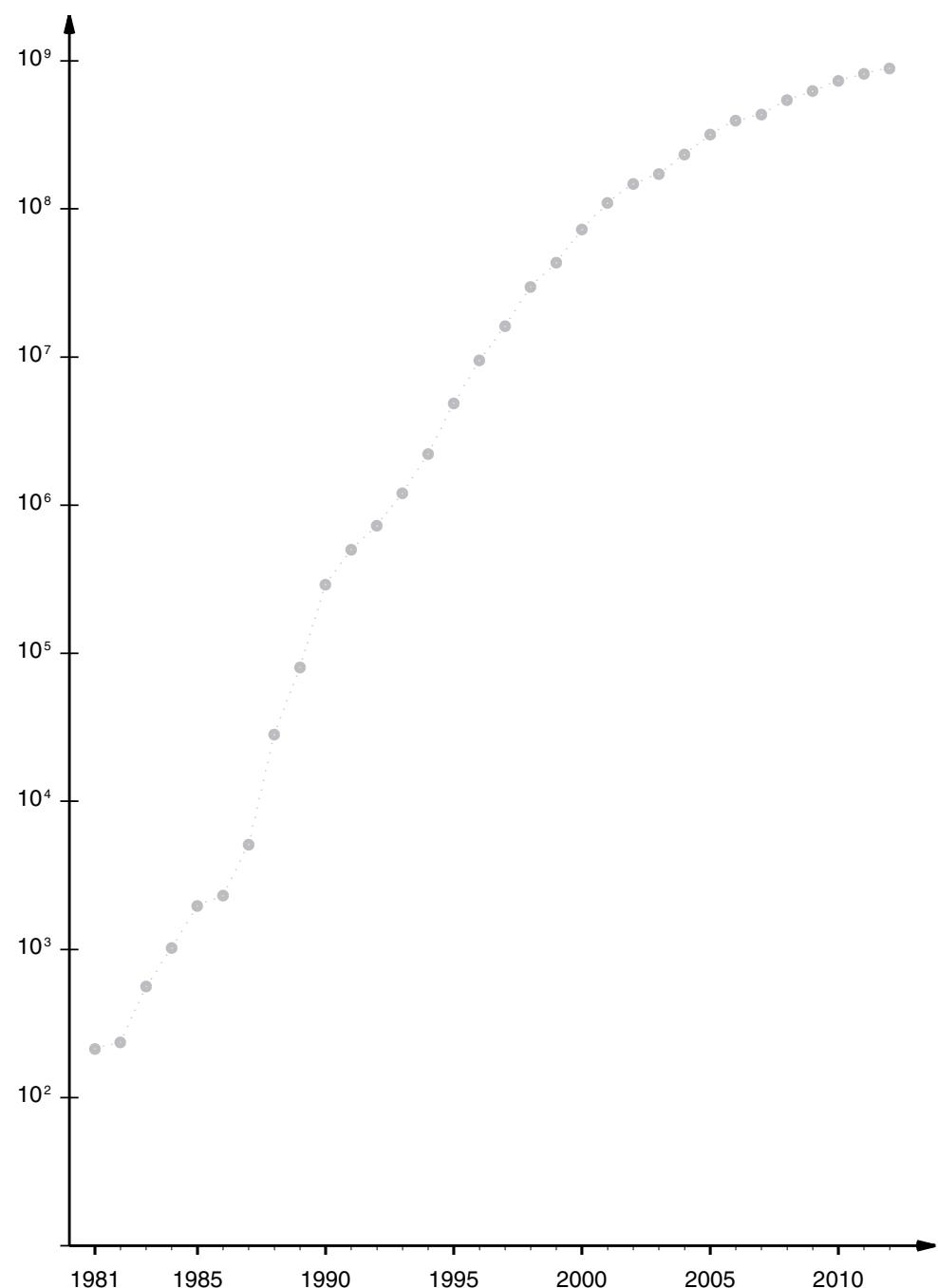


Figura 2.2 Crecimiento de Internet representado en una escala logarítmica.

La figura 2.2 revela que Internet ha experimentado un crecimiento exponencial por más de 25 años. Es decir, Internet ha estado duplicando su tamaño cada nueve a catorce meses. Resulta interesante resaltar que si se mide por el número de computadoras, la tasa de crecimiento exponencial ha disminuido ligeramente desde finales de la década de 1990. Sin embargo, considerar el número de computadoras conectadas a Internet como una medida del tamaño, puede ser engañoso debido a que muchos usuarios de todo el mundo ahora acceden a Internet a través de la red de telefonía celular.

2.4 De la compartición de recursos a la comunicación

A medida que fue creciendo, Internet cambió en dos formas importantes. Primero, las velocidades de comunicación aumentaron en forma drástica; un enlace troncal en la red Internet actual puede transportar casi 200,000 veces más bits por segundo que un enlace troncal en la red Internet original. Segundo, surgieron nuevas aplicaciones que atrajeron a una gran variedad de sectores de la sociedad. El segundo punto es obvio: Internet ya no está dominada por científicos e ingenieros, ni tiene como objetivos principales las aplicaciones científicas y el acceso a los recursos computacionales de alto costo.

Hay dos cambios tecnológicos que propiciaron un distanciamiento de la compartición de recursos hacia nuevas aplicaciones. Por una parte, las mayores velocidades de comunicación permitieron a las aplicaciones transferir grandes volúmenes de datos con rapidez. Por otra parte, la llegada de computadoras personales poderosas y accesibles dio el poder de cómputo necesario para cálculos y despliegues de gráficos complejos, eliminando la mayor parte de la demanda de recursos compartidos.

En conclusión:

La disponibilidad de las tecnologías de cómputo y de comunicaciones de alta velocidad cambió el enfoque de Internet de la compartición de recursos a la comunicación con fines generales.

2.5 Del texto al multimedia

Uno de los cambios más obvios ocurrió en los datos que se envían por Internet. La figura 2.3 ilustra un aspecto de este cambio.

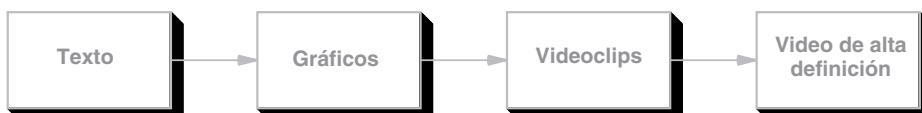


Figura 2.3 Un cambio en el tipo de datos al que los usuarios acceden a través de Internet.

Como lo indica la figura, en un principio la comunicación por Internet involucraba datos de texto. Los mensajes de correo electrónico, en particular, se limitaban a texto en un tipo de letra fijo. Para la

década de 1990, las computadoras tenían pantallas a color capaces de mostrar gráficos y surgieron aplicaciones que permitían a los usuarios transferir imágenes con facilidad. Para finales de la década de 1990, los usuarios empezaron a enviar videos cortos, aunque la descarga de videos más grandes se convirtió en algo posible. Para la década de 2000, las velocidades de Internet permitieron la descarga y reproducción de películas de alta definición. La figura 2.4 ilustra que ocurrió una transición similar en el audio.

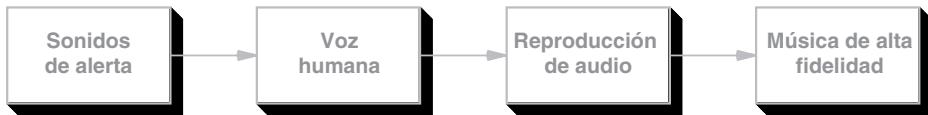


Figura 2.4 Un cambio en el tipo de audio al que los usuarios acceden a través de Internet.

Usamos el término *multimedia* para representar los datos que contienen una combinación de texto, gráficos, audio y video. Gran parte del contenido disponible en Internet consiste ahora en archivos multimedia. Lo que es más, la calidad mejoró a medida que mayores anchos de banda permitieron la comunicación mediante video de alta resolución y audio de alta fidelidad. En conclusión:

El uso de Internet pasó de la transferencia de documentos de texto estático, a la transmisión de contenido multimedia de alta calidad.

2.6 Tendencias recientes

De forma sorprendente, siguen surgiendo nuevas tecnologías de redes y nuevas aplicaciones de Internet. Algunas de las transiciones más considerables ocurrieron a medida que los sistemas de comunicaciones tradicionales, como la red de telefonía de voz y la televisión por cable, pasaron de la tecnología analógica a la digital y adoptaron la tecnología de Internet. Además, el soporte para los usuarios móviles se está acelerando en forma vertiginosa. La figura 2.5 muestra una lista con algunos de los cambios.

Tema	Transición
Sistema telefónico	De análogo a voz sobre IP (VoIP)
Televisión por cable	De la transmisión analógica al protocolo de Internet (IP)
Celular	De servicios celulares analógicos a digitales (4G)
Acceso a Internet	Del acceso alámbrico al inalámbrico (Wi-Fi)
Acceso a los datos	De servicios centralizados a distribuidos (P2P)

Figura 2.5 Ejemplos de las transiciones en las redes y en Internet.

Uno de los aspectos más interesantes de Internet surge de la forma en que han cambiado sus usos, aun cuando la tecnología involucrada ha permanecido básicamente sin grandes cambios. Como ejemplo, la figura 2.6 muestra una lista de los tipos de aplicaciones que han surgido desde que se inventó Internet.

Aplicación	Importante para
Redes sociales	Consumidores, organizaciones no lucrativas
Redes sensoriales	Medio ambiente, seguridad, rastreo de flotillas
Teleconferencias de alta velocidad	Comunicación de negocio a negocio
Banca en línea y pagos	Individuos, corporaciones, gobiernos

Figura 2.6 Ejemplos de aplicaciones populares de Internet.

Las aplicaciones de redes sociales como Facebook y YouTube son fascinantes, ya que han creado nuevas conexiones sociales: hay grupos de personas que se conocen únicamente a través de Internet. Los sociólogos sugieren que dichas aplicaciones permitirán que más personas encuentren a otras con intereses similares, e impulsarán la creación de más grupos sociales pequeños.

2.7 De las computadoras personales a la computación en la nube

Internet trajo consigo otro cambio arrollador en nuestro mundo digital: la *computación en la nube*. En 2005 las empresas se dieron cuenta que la economía de escala y las conexiones de Internet les permitirían ofrecer servicios de cómputo y de almacenamiento de datos que fueran menos costosos que los servicios implementados por un sistema en donde cada usuario tuviera su propia computadora. La idea es simple y directa: un *proveedor de servicios en la nube* construye un gran *centro de datos en la nube*, el cual contiene muchas computadoras y muchos discos duros, todos conectados a Internet. Un individuo o una empresa contrata el servicio al proveedor de la nube. En principio, un cliente de la nube sólo necesita un dispositivo de acceso (por ejemplo, un teléfono inteligente, una tableta o un dispositivo de escritorio con pantalla y teclado). Todos los archivos y las aplicaciones del usuario se ubican en el centro de datos de la nube. Cuando el cliente necesita ejecutar una aplicación, ésta se ejecuta desde una computadora que está en el centro de datos de la nube. De manera similar, cuando un cliente guarda un archivo, éste se guarda en un disco duro que se encuentra en el centro de datos de la nube. Decimos que la información del cliente se almacena “en la nube”. Algo importante es que el cliente puede acceder al centro de datos de la nube desde cualquier lugar donde haya Internet, lo que significa que un viajero no necesita llevar consigo copias de sus archivos; el entorno de cómputo siempre está disponible y siempre es el mismo.

Para los individuos, una ventaja clave de la computación en la nube surge debido a que el proveedor de servicios en la nube se hace cargo de todos los detalles del hardware, el software y la operación. El proveedor actualiza el hardware periódicamente y asegura que todo el software de aplicación se actualice

a la versión más reciente. Además, un proveedor de servicios en la nube ofrece servicios de respaldo de datos que permiten a un cliente recuperar versiones anteriores de archivos que haya perdido.

Para las empresas, la computación en la nube ofrece mayor flexibilidad a un menor costo. En vez de contratar mucho personal de TI para instalar y administrar los equipos de cómputo, la empresa puede contratar los servicios de un proveedor de la nube. El proveedor alquila el espacio físico necesario para el centro de datos, se hace cargo de la alimentación eléctrica y el enfriamiento (incluyendo generadores que funcionen durante cortes de energía), y se asegura que tanto las instalaciones como los datos se mantengan seguros. Además, un proveedor de servicios en la nube ofrece un *servicio elástico* ya que la capacidad de almacenamiento y el número de computadoras que un cliente requiere puede variar con el tiempo. Por ejemplo, muchas empresas tienen un modelo de negocios por temporadas. Una empresa agrícola realiza muchos registros de información durante la época de cosecha. Una empresa de preparación de impuestos podría necesitar de mucho poder de cálculo y almacenamiento en los meses y semanas previos al vencimiento del plazo para el pago de impuestos. Los proveedores de servicios en la nube pueden adaptarse al uso por temporadas, al permitir que un cliente adquiera los recursos cuando sea necesario y los devuelva cuando ya no sean necesarios. De esta forma, en vez de comprar instalaciones para adaptarse a la máxima demanda y dejar las computadoras inactivas durante la temporada de poca actividad, una empresa que utiliza servicios en la nube sólo paga por las instalaciones cuando las necesita. De hecho, se puede usar un enfoque híbrido en el que una empresa tenga sus propias instalaciones que sean suficientes para la mayoría de sus necesidades, y que sólo use los servicios en la nube durante la temporada alta, cuando la demanda excede la capacidad local. En conclusión:

Los servicios en la nube son elásticos, lo que significa que en vez de comprar una cantidad fija de hardware, un cliente sólo paga por los recursos que en realidad utilice.

2.8 Resumen

La Agencia de Proyectos de Investigación Avanzados (ARPA) financió gran parte de las primeras investigaciones en las redes, como una forma de compartir los recursos de cómputo entre sus investigadores. Posteriormente, la ARPA cambió su enfoque hacia las interconexiones de redes y financió la investigación sobre la Internet, la cual ha ido creciendo de manera exponencial durante las últimas décadas.

Con la llegada de las computadoras personales y las tecnologías de redes con mayor velocidad, el enfoque de Internet cambió de la compartición de recursos a la comunicación con fines generales. El tipo de datos enviados a través de Internet pasó del texto a los gráficos, los videoclips y los videos de alta definición. Ocurrió una transformación similar con el audio, lo que permitió a Internet transferir documentos multimedia.

Las tecnologías en Internet impactan a la sociedad de muchas formas. Los cambios recientes incluyen la transición de los teléfonos de voz, la televisión por cable y los servicios celulares hacia las tecnologías digitales de Internet. Además, el acceso inalámbrico a Internet y el soporte para usuarios móviles se ha vuelto esencial.

Aunque la tecnología de Internet ha permanecido prácticamente sin modificaciones, siguen surgiendo nuevas aplicaciones que proporcionan experiencias mejoradas para los usuarios. Las redes sensoriales, los mapas y los sistemas de navegación permiten el monitoreo ambiental, una mayor seguridad

y un viaje más sencillo. Las aplicaciones de redes sociales fomentan la creación de nuevos grupos y organizaciones sociales.

La llegada de la computación en la nube representa otro cambio importante. En vez de almacenar datos y ejecutar aplicaciones en una computadora local, el modelo de la nube permite a los individuos y las empresas almacenar datos y ejecutar aplicaciones en un centro de datos remoto. Los proveedores de servicios en la nube ofrecen servicios de computación y almacenamiento elásticos, lo que significa que los clientes sólo pagan por la capacidad de cómputo y almacenamiento que utilicen.

EJERCICIOS

- 2.1** ¿Por qué era importante la compartición de los recursos computacionales en la década de 1960?
- 2.2** La escala en la figura 2.1 muestra que el crecimiento de Internet no comenzó sino hasta 1995. ¿Por qué es engañosa la figura?
- 2.3** Suponga que cada año se incorporan cien millones de nuevas computadoras a Internet. Si las computadoras se incorporan a un ritmo uniforme, ¿cuánto tiempo transcurre entre una incorporación y la siguiente?
- 2.4** Amplíe la escala de la figura 2.2 para calcular cuántas computadoras estarán conectadas a Internet para 2020.
- 2.5** ¿Qué cambio en Internet ocurrió cuando apareció el servicio *World Wide Web* por primera vez?
- 2.6** Mencione los pasos en la transición en la transferencia de gráficos de la red Internet preliminar a la red Internet actual.
- 2.7** Describa la evolución que ha tenido Internet en la transmisión de audio.
- 2.8** ¿Qué impacto está teniendo la tecnología de Internet en la industria de la televisión por cable?
- 2.9** ¿Qué tecnología de Internet utiliza actualmente el sistema telefónico?
- 2.10** ¿Por qué es importante el cambio del acceso a Internet alámbrico al inalámbrico?
- 2.11** Mencione cuatro nuevos usos de Internet e indique los grupos para los que cada uno es importante.
- 2.12** Describa las aplicaciones de Internet que usa con regularidad y que no estaban disponibles para sus padres cuando tenían su edad.
- 2.13** ¿Por qué los individuos que no tienen conocimientos técnicos se sentirían atraídos a la computación en la nube?
- 2.14** Busque en Web tres empresas que ofrezcan servicios en la nube.

Contenido del capítulo

- 3.1 Introducción, 27
- 3.2 Dos paradigmas esenciales de la comunicación en Internet, 28
- 3.3 Comunicación orientada a la conexión, 29
- 3.4 El modelo de interacción cliente-servidor, 30
- 3.5 Características de los clientes y los servidores, 31
- 3.6 Programas servidor y computadoras tipo servidor, 31
- 3.7 Solicitudes, respuestas y dirección del flujo de datos, 32
- 3.8 Múltiples clientes y múltiples servidores, 32
- 3.9 Identificación de servidores y demultiplexación, 33
- 3.10 Servidores concurrentes, 34
- 3.11 Dependencias circulares entre servidores, 35
- 3.12 Interacciones de igual a igual, 35
- 3.13 Programación de redes y la API de sockets, 36
- 3.14 Sockets, descriptores y entrada y salida de red, 36
- 3.15 Parámetros y la API de sockets, 37
- 3.16 Llamadas de sockets en un cliente y un servidor, 38
- 3.17 Funciones de sockets utilizadas por el cliente y el servidor, 38
- 3.18 La función *connect* utilizada sólo por un cliente, 40
- 3.19 Funciones de sockets utilizadas sólo por un servidor, 40
- 3.20 Funciones de sockets utilizadas con el paradigma de mensajes, 43
- 3.21 Otras funciones de sockets, 44
- 3.22 Sockets, hilos y herencia, 45
- 3.23 Resumen, 45

3

Aplicaciones de Internet y programación de redes

3.1 Introducción

Internet ofrece una extensa variedad de servicios que incluyen la navegación Web, los mensajes de texto y la transmisión de video por flujo continuo o *streaming*, entre otras muchas aplicaciones. Lo sorprendente es que ninguno de los servicios forma parte de la infraestructura de comunicación con la que fue creada la Internet. En su lugar, Internet proporciona un mecanismo de comunicación de fines generales en el que se basan todos los servicios, y cada uno de ellos se lleva a cabo mediante programas de aplicación que se ejecutan en computadoras conectadas a Internet. De hecho, es posible crear servicios totalmente nuevos sin modificar la estructura de Internet.

Este capítulo cubre dos conceptos clave que explican las aplicaciones de Internet. Primero, el capítulo describe el paradigma conceptual que siguen las aplicaciones cuando se comunican a través de Internet. Segundo, el capítulo presenta los detalles de la *interfaz de programación de aplicaciones (API) de sockets* que usan las aplicaciones de Internet. El capítulo muestra que un programador no necesita comprender los detalles de los protocolos de red para escribir aplicaciones innovadoras; una vez que domina algunos conceptos básicos, el programador puede desarrollar aplicaciones de red. El siguiente capítulo continúa la explicación mediante un análisis de algunas aplicaciones de Internet de ejemplo. Las partes posteriores del texto revelan muchos de los detalles detrás de las aplicaciones de Internet haciendo énfasis en las comunicaciones de datos y los protocolos que utilizan estas aplicaciones.

3.2 Dos paradigmas esenciales de la comunicación en Internet

Internet soporta dos paradigmas básicos de comunicación: un paradigma de *flujo* y un paradigma de *mensaje*. La figura 3.1 resume las diferencias.

Paradigma de flujo	Paradigma de mensaje
Orientado a la conexión	Sin conexión
Comunicación de uno a uno	Comunicación de varios a varios
El emisor transfiere una secuencia de bytes individuales	El emisor transmite una secuencia de mensajes discretos
Transferencia de longitud variada	Cada mensaje está limitado a 64 KB
La mayoría de las aplicaciones lo utilizan	Se usa para aplicaciones multimedia
Opera sobre TCP	Opera sobre UDP

Figura 3.1 Los dos paradigmas que utilizan las aplicaciones de Internet.

3.2.1 Transporte de flujos en Internet

El término *flujo* denota un paradigma en el que una secuencia de bytes fluye de un programa de aplicación a otro. Por ejemplo, un flujo se utiliza cuando alguien descarga una película. De hecho, el mecanismo de Internet organiza dos flujos simultáneos entre un par de aplicaciones que están en comunicación, uno en cada dirección. Un navegador usa el servicio de flujos para comunicarse con un servidor Web: el navegador envía una solicitud y el servidor Web responde enviando la página. La red acepta los datos que fluyen de cada una de las dos aplicaciones y entrega los datos a la otra aplicación.

El mecanismo de flujos transfiere una secuencia de bytes sin adjuntarles ningún significado y sin agregarles límites. Una aplicación emisora puede optar por generar un byte a la vez o puede generar grandes bloques de bytes. El servicio de flujo mueve bytes a través de Internet y los va entregando a medida que llegan. Es decir, el servicio de flujo puede optar por combinar trozos más pequeños de bytes en un bloque grande, o puede dividir un bloque grande en trozos más pequeños. En conclusión:

Aunque entrega todos los bytes en secuencia, el servicio de flujo de Internet no garantiza que los trozos de bytes que se transfieran a una aplicación receptora correspondan con los trozos de bytes transferidos por la aplicación emisora.

3.2.2 Transporte de mensajes en Internet

El mecanismo alterno de comunicación de Internet sigue un *paradigma de mensaje*, en el cual la red acepta y entrega mensajes. Cada mensaje que se entrega a un receptor corresponde a un mensaje que transmitió un emisor; la red nunca entrega parte de un mensaje, ni une varios mensajes entre sí. Por consiguiente, si un emisor coloca cierta cantidad de kilobytes en un mensaje saliente, el receptor encontrará exactamente la misma cantidad de kilobytes en el mensaje entrante.

El paradigma de mensajes permite enviar un mensaje desde una aplicación que se encuentra en una computadora a una aplicación en otra computadora, e incluso se puede transmitir el mensaje a todas las computadoras de una red dada. Además, las aplicaciones de varias computadoras pueden enviar mensajes a una aplicación receptora dada. Así, el paradigma de mensajes ofrece una opción de comunicación de uno a uno, de uno a varios o de varios a uno.

Lo sorprendente es que el servicio de mensajes no ofrece ninguna garantía sobre el orden en el que se entregarán los mensajes, o si cierto mensaje llegará. El servicio permite que los mensajes:

- Se pierdan (es decir, que nunca se entreguen)
- Se dupliquen (que llegue más de una copia)
- Se retrasen (algunos paquetes pueden tardar mucho tiempo en llegar)
- Se entreguen desordenados

Los capítulos posteriores explican por qué pueden ocurrir dichos errores; por ahora basta con entender una consecuencia importante:

Un programador que selecciona el paradigma de mensajes debe asegurarse de que la aplicación opere correctamente, incluso si se pierden o reordenan paquetes.

Puesto que para ofrecer garantías se requiere de experiencia especial en el diseño de protocolos, la mayoría de los programadores seleccionan el servicio de flujo; menos del 5% de todos los paquetes en Internet usan el servicio de mensajes. Sólo se hacen excepciones para situaciones especiales (donde se necesita la difusión) o para aplicaciones en las que un receptor debe reproducir los datos a medida que van llegando (por ejemplo, una llamada de audio). En el resto del capítulo nos enfocaremos en el servicio de flujo.

3.3 Comunicación orientada a la conexión

El servicio de flujo de Internet está *orientado a la conexión*, lo que significa que opera en forma similar a una llamada telefónica: antes de poder comunicarse, dos aplicaciones deben solicitar que se establezca una *conexión* entre ellas. Una vez que se establece, la conexión permite a las aplicaciones enviar datos

en cualquier dirección. Por último, cuando terminan de comunicarse, las aplicaciones solicitan que se termine la conexión. El algoritmo 3.1 sintetiza la interacción orientada a la conexión.

Algoritmo 3.1

Propósito:

Interacción mediante el servicio de flujo de Internet

Método:

Un par de aplicaciones solicitan una conexión

El par usa la conexión para intercambiar datos

El par solicita que se termine la conexión

Algoritmo 3.1 Comunicación con el mecanismo de flujo orientado a la conexión de Internet.

3.4 El modelo de interacción cliente-servidor

El primer paso en el algoritmo 3.1 genera una pregunta: ¿cómo puede un par de aplicaciones que se ejecutan en dos computadoras independientes coordinarse para garantizar que vayan a solicitar una conexión al mismo tiempo? La respuesta está en una forma de interacción que se conoce como el *modelo cliente-servidor*. Una aplicación, conocida como *servidor*, empieza primero y espera el contacto. La otra aplicación, conocida como *cliente*, empieza después e inicia la conexión. La figura 3.2 sintetiza la interacción entre cliente y servidor.

Aplicación servidor	Aplicación cliente
Inicia primero	Inicia después
No necesita saber qué cliente la contactará	Debe saber qué servidor contactar
Espera en forma pasiva y por un tiempo indefinido a que un cliente haga contacto	Inicia un contacto cada vez que necesita la comunicación
Se comunica con un cliente mediante el envío y la recepción de los datos	Se comunica con un servidor mediante el envío y la recepción de los datos
Se mantiene en ejecución después de dar servicio a un cliente, y espera a otro	Puede terminar después de interactuar con un servidor

Figura 3.2 Resumen del modelo cliente-servidor.

Secciones posteriores describen cómo es que los servicios específicos usan el modelo cliente-servidor. Por ahora basta recordar que:

Aunque proporciona una comunicación elemental, Internet no inicia contacto con ni acepta el contacto de una computadora remota; son los programas de aplicación conocidos como clientes y servidores quienes se encargan de todos los servicios.

3.5 Características de los clientes y los servidores

Aunque existen pequeñas variaciones, la mayoría de las instancias que siguen el paradigma cliente-servidor tienen las siguientes características generales:

Software cliente

- Consiste en un programa de aplicación que se convierte temporalmente en un cliente cuando se necesita un acceso remoto
- Lo invoca un usuario en forma directa y se ejecuta sólo por una sesión
- Se ejecuta de manera local en la computadora o dispositivo de un usuario
- Inicia contacto en forma activa con un servidor
- Puede acceder a varios servicios según sea necesario, aunque generalmente sólo hace contacto con un servidor remoto a la vez
- No requiere gran capacidad de hardware

Software servidor

- Consiste en un programa con fines específicos dedicado a proveer un servicio
- Se invoca automáticamente cuando se inicia un sistema y sigue ejecutándose durante muchas sesiones
- Se ejecuta en un sistema de computadoras específico
- Espera en forma pasiva a que clientes remotos hagan contacto
- Puede aceptar conexiones de muchos clientes al mismo tiempo, aunque generalmente sólo ofrece un servicio
- Requiere de hardware poderoso y de un sofisticado sistema operativo

3.6 Programas servidor y computadoras tipo servidor

Algunas veces surge una confusión con respecto al término *servidor*. Formalmente el término se refiere a un programa que espera en forma pasiva una comunicación, y no a la computadora en la que se ejecuta dicho programa. Sin embargo, cuando una computadora se dedica a ejecutar dos o más progra-

mas servidor, el personal de TI comúnmente denomina a esta computadora como un “servidor”. Los distribuidores de hardware contribuyen a la confusión, ya que clasifican las computadoras que tienen procesadores veloces, memorias grandes y sistemas operativos poderosos como máquinas *servidor*. La figura 3.3 ilustra las definiciones.

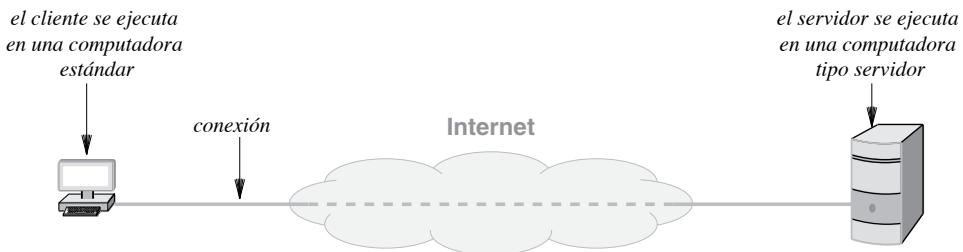


Figura 3.3 Ilustración de un cliente y un servidor.

3.7 Solicitudes, respuestas y dirección del flujo de datos

Los términos *cliente* y *servidor* surgen debido a que el lado que inicia el contacto se convierte en un *cliente*. Sin embargo, una vez que el contacto se establece se hace viable la comunicación de dos vías (es decir, los datos pueden fluir de un cliente a un servidor o de un servidor a un cliente). Por lo general, un cliente envía una solicitud a un servidor y éste devuelve una respuesta al cliente. En algunos casos, un cliente envía una serie de solicitudes y el servidor emite una serie de respuestas (por ejemplo, un cliente de base de datos podría permitir que un usuario busque más de un elemento a la vez). El concepto puede resumirse así:

La información entre un cliente y un servidor puede fluir en una o en ambas direcciones. Aunque muchos servicios acuerdan que el cliente envíe una o más solicitudes y el servidor regrese respuestas, también son posibles otras interacciones.

3.8 Múltiples clientes y múltiples servidores

Un cliente o servidor consiste en un programa de aplicación y cualquier computadora puede ejecutar varias aplicaciones simultáneamente. En consecuencia, una misma computadora puede ejecutar:

- Un solo cliente
- Un solo servidor
- Varias copias de un cliente que haga contacto con un servidor dado
- Varios clientes que hagan cada uno contacto con un servidor específico
- Varios servidores para un servicio específico cada uno

Es útil permitir que una computadora opere varios clientes, ya que podemos acceder a los servicios al mismo tiempo. Por ejemplo, un usuario puede estar ejecutando tres aplicaciones al mismo tiempo: un navegador Web, una aplicación de mensajería instantánea y una videoconferencia. Cada aplicación es un cliente que hace contacto con un servidor específico independiente de las otras aplicaciones. De hecho, la tecnología permite a un usuario tener abiertas dos copias de una sola aplicación, cada una haciendo contacto con un servidor distinto (por ejemplo, dos ventanas de navegadores Web, cada una de las cuales hace contacto con un sitio Web diferente).

Permitir que una misma computadora ejecute varios programas servidor resulta útil por dos razones. Primera, al usar sólo una computadora física en vez de varias se reducen los gastos de administración requeridos para dar mantenimiento a las instalaciones. Segunda, la experiencia ha demostrado que la demanda de un servicio es por lo general esporádica: a menudo un servidor permanece inactivo por extensos períodos y un servidor inactivo no usa la CPU. Por ende, si la demanda total de los servicios es lo bastante pequeña, al consolidar servidores en una sola computadora se puede reducir de manera drástica el costo sin reducir en gran medida el rendimiento. En conclusión:

Una sola computadora poderosa puede ofrecer varios servicios al mismo tiempo; la computadora ejecuta un programa servidor para cada servicio.

3.9 Identificación de servidores y demultiplexación

¿Cómo identifica un cliente a un servidor? Los protocolos de Internet dividen la identificación en dos piezas:

- Un identificador que determina la computadora en la que un servidor se ejecuta
- Un identificador que determina un servicio específico en la computadora

Identificación de una computadora. A cada computadora en Internet se le asigna un identificador único conocido como *dirección del protocolo de Internet (dirección IP)*[†]. Al hacer contacto con un servidor, el cliente debe especificar la dirección IP del servidor. Para que podamos identificar a cada computadora con mayor facilidad se le asigna también un nombre, y el *sistema de nombres de dominios (DNS)*, descrito en el capítulo 4, es quien se encarga de traducir un nombre en una dirección. De esta forma, un usuario sólo tiene que conocer un nombre como *www.cisco.com* en vez de una dirección compuesta únicamente por números.

Identificación de un servicio. A cada servicio disponible en Internet se le asigna un identificador único de 16 bits conocido como *número de puerto del protocolo* (que comúnmente se abrevia como *número de puerto*). Por ejemplo, al correo electrónico se le asigna el puerto número 25 y al servicio World Wide Web se le asigna el puerto número 80. Cuando un servidor comienza su ejecución, se registra con su sistema local especificando el número de puerto para el servicio que ofrece. Cuando un cliente hace contacto con un servidor remoto para solicitar un servicio, la solicitud contiene un número de puerto. Por ende, cuando llega una solicitud a un servidor, el software en el servidor usa el número de puerto contenido en la solicitud para determinar qué aplicación en la computadora servidor debe atender la solicitud.

La figura 3.4 sintetiza la explicación con una lista de los pasos básicos que realizan un cliente y un servidor para comunicarse.

[†] El capítulo 21 cubre en detalle las direcciones de Internet y explica las direcciones que se utilizan tanto con IPv4 como con IPv6.

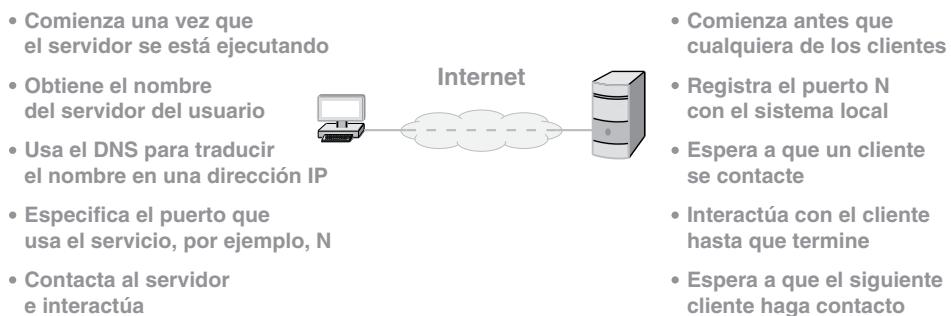


Figura 3.4 Los pasos conceptuales que realizan un cliente y un servidor para comunicarse.

3.10 Servidores concurrentes

Los pasos en la figura 3.4 implican que un servidor atiende a un cliente a la vez. Aunque una metodología *secuencial* funciona en algunos casos simples, la mayoría de los servidores son *concurrentes*. Es decir, un servidor usa más de un *hilo de control*,[†] para manejar varios clientes al mismo tiempo.

Para comprender por qué es importante el servicio simultáneo, considere lo que ocurre si un cliente descarga una película de un servidor. Si un servidor manejara una solicitud a la vez, todos los demás clientes tendrían que esperar a que el servidor terminara de transferir toda la película al primer cliente. En contraste, un servidor concurrente no obliga a un cliente a que espere. Por ende, si llega un segundo cliente y solicita una descarga corta (por ejemplo, una sola canción), la segunda solicitud comenzará de inmediato y puede incluso terminar antes de que se complete la transferencia de la película (dependiendo del tamaño de los archivos y la velocidad con la que cada cliente puede recibir datos).

Los detalles de la ejecución concurrente dependen del sistema operativo que se utilice, pero la idea es simple: el código del servidor concurrente se divide en dos piezas, un programa principal o *hilo* y un manejador. El hilo principal simplemente acepta el contacto de un cliente, y crea un hilo de control para atender a dicho cliente. Cada hilo de control interactúa con un solo cliente y ejecuta el código del manejador. Después de atender a un cliente, el hilo termina. Mientras tanto, el hilo principal mantiene al servidor activo; después de crear un hilo para atender una solicitud, el hilo principal espera a que llegue otra solicitud.

Tenga en cuenta que si N clientes utilizan de manera simultánea a un servidor concurrente, habrá $N+1$ hilos en ejecución; el hilo principal espera solicitudes adicionales y hay N hilos interactuando con un solo cliente. En conclusión:

Un servidor concurrente usa hilos de ejecución para atender solicitudes de varios clientes al mismo tiempo. Hacer esto significa que un cliente no tiene que esperar a que el cliente anterior termine.

[†] Algunos sistemas operativos usan el término *hilo de ejecución* o *proceso* para denotar un hilo de control.

3.11 Dependencias circulares entre servidores

Técnicamente, cualquier programa que contacta a otro actúa como cliente, y cualquier programa que acepta el contacto de otro actúa como servidor. En la práctica, la distinción se borra debido a que un servidor de un servicio puede actuar como cliente de otro. Por ejemplo, antes de poder llenar una página Web, un servidor Web quizás tenga que convertirse en cliente de un sistema de bases de datos específico o de un servicio de seguridad (por ejemplo, para verificar que a un cliente se le permita acceder a cierta página Web).

Desde luego que los programadores deben tener cuidado de evitar dependencias circulares entre los servidores. Por ejemplo, considere lo que puede ocurrir si un servidor del servicio X_1 se convierte en un cliente de X_2 , que a su vez se convierte en cliente del servicio X_3 , que por su parte se convierte en cliente de X_1 . La cadena de solicitudes puede continuar de manera indefinida hasta que los tres servidores agotan sus recursos. El potencial de circularidad es especialmente alto cuando los servicios se diseñan de manera independiente, ya que no hay un solo programador que controle todos los servidores.

3.12 Interacciones de igual a igual

Si un solo servidor proporciona un servicio dado, la conexión de red entre el servidor e Internet puede convertirse en un cuello de botella. La figura 3.5 ilustra el problema.

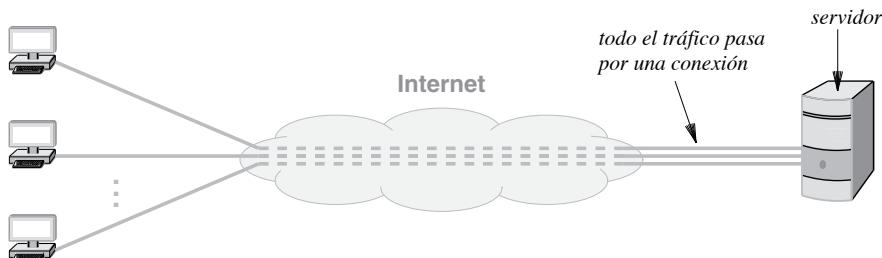


Figura 3.5 El cuello de botella en un diseño que usa un solo servidor.

Surge la pregunta: ¿pueden proveerse servicios de Internet sin crear un cuello de botella central? La forma de evitar un cuello de botella ha sido la base de las aplicaciones de compartición de archivos. Conocida como la arquitectura de *igual a igual* (*p2p*), el esquema evita colocar datos en un servidor central. Es decir, los datos se distribuyen de manera equitativa entre un conjunto de N servidores y la solicitud de cada cliente se envía al servidor apropiado. Como un servidor dado sólo proporciona I/N de los datos, la cantidad de tráfico entre un servidor e Internet es I/N , igual que la arquitectura de un solo servidor. La idea central es que el software servidor puede ejecutarse en las mismas computadoras como cliente. Si cada usuario acepta colocar I/N de los datos en su computadora, no se necesitarán servidores especiales. La figura 3.6 ilustra la arquitectura. El ejemplo sólo muestra cuatro computadoras. En un sistema *p2p* real, el tráfico en una computadora puede ser extremadamente pequeño, debido a que N puede ser muy grande (decenas de miles).

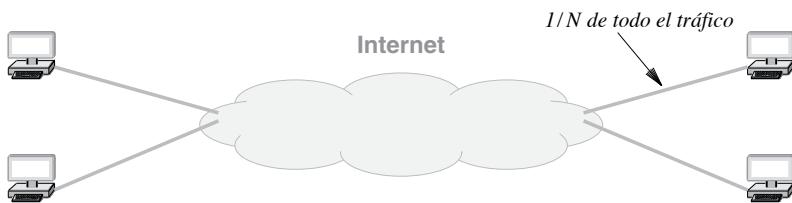


Figura 3.6 Interacción de ejemplo en un sistema de igual a igual.

3.13 Programación de redes y la API de sockets

La interfaz que una aplicación usa para especificar la comunicación con Internet se conoce como *interfaz de programación de aplicaciones (API)*.[†] Aunque los detalles exactos de cada API dependen del sistema operativo, hay una API específica que se ha establecido como el estándar para el software que se comunica a través de Internet. Se conoce como la *API de sockets* o simplemente *sockets*; la API está disponible para diversos sistemas operativos como Windows de Microsoft, OS-X de Apple, Android y varios sistemas UNIX, incluyendo Linux. En conclusión:

La API de sockets, que se ha convertido en un estándar para la comunicación en Internet, está disponible en la mayoría de los sistemas operativos.

El resto del capítulo describe las funciones en la API de sockets; los lectores que no sean programadores de computadora pueden omitir muchos de los detalles.

3.14 Sockets, descriptores y entrada y salida de red

Debido a que se desarrolló originalmente como parte del sistema operativo UNIX, la API de sockets está integrada a las operaciones de entrada y salida (E/S). En particular, cuando una aplicación crea un *socket* para usarlo en la comunicación con Internet, el sistema operativo devuelve un pequeño *descriptor* entero que identifica al socket. Cuando la aplicación llama posteriormente a las funciones para que realicen una operación con el socket (por ejemplo, para transferir datos a través de la red o recibir datos entrantes) ésta pasa el descriptor como un argumento.

En muchos sistemas operativos, los descriptores de sockets se integran con otros descriptores de E/S. Como resultado, una aplicación puede usar las operaciones de *lectura* y *escritura* para E/S de sockets o para E/S de un archivo. En conclusión:

Cuando una aplicación crea un socket, el sistema operativo devuelve un pequeño descriptor entero que la aplicación usa para hacer referencia al socket.

[†] El apéndice 1 contiene una API simplificada (con sólo siete funciones) y código de ejemplo que demuestra cómo puede usarse dicha API para crear aplicaciones de Internet, incluyendo un servidor Web funcional.

3.15 Parámetros y la API de sockets

La programación de sockets difiere de la E/S convencional debido a que una aplicación debe especificar muchos detalles, como la dirección de una computadora remota, el número de puerto de un protocolo y si la aplicación actuará como cliente o como servidor (es decir, si va a iniciar una conexión o no). Para evitar tener una sola función de socket con varios parámetros, los diseñadores de la API de sockets optaron por definir varias funciones. En esencia, una aplicación crea un socket y luego invoca funciones para especificar detalles. La ventaja de la metodología de los sockets es que la mayoría de las funciones tienen tres o menos parámetros; la desventaja es que un programador debe llamar varias funciones al usar sockets. La figura 3.7 sintetiza las funciones clave en la API de sockets.

Nombre	Utilizada por	Significado
accept	servidor	Aceptar una conexión entrante
bind	servidor	Especificar dirección IP y puerto del protocolo
close	ambos	Terminar la comunicación
connect	cliente	Conectarse a una aplicación remota
getpeername	servidor	Obtener la dirección IP del cliente
getsockopt	servidor	Obtener opciones actuales para un socket
listen	servidor	Preparar socket para que lo use un servidor
recv	ambos	Recibir datos o mensajes entrantes
recvmsg	ambos	Recibir datos (paradigma de mensaje)
recvfrom	ambos	Recibir un mensaje y la dirección del emisor
send	ambos	Enviar datos o mensaje salientes
sendmsg	ambos	Enviar un mensaje saliente
sendto	ambos	Enviar un mensaje (variante de sendmsg)
setsockopt	ambos	Cambiar opciones de socket
shutdown	ambos	Terminar una conexión
socket	ambos	Crear un socket para usarse en lo anterior

Figura 3.7 Un resumen de las principales funciones en la API de sockets.

3.16 Llamadas de sockets en un cliente y un servidor

La figura 3.8 muestra la secuencia de las llamadas de sockets realizadas por un cliente y un servidor ordinarios que usan una conexión de flujo. En la figura, el cliente envía los datos primero y el servidor espera a recibirlos. En la práctica, algunas aplicaciones acuerdan que el servidor envíe primero (es decir, se llama a *send* y a *recv* en el orden inverso).

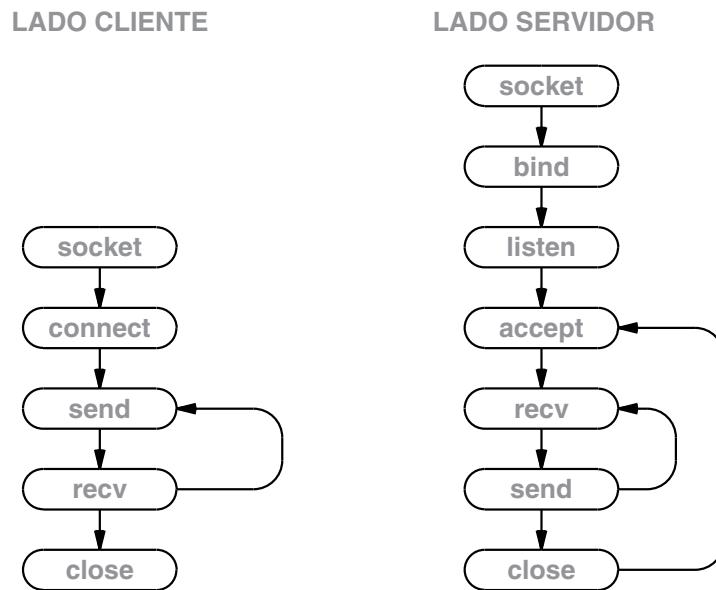


Figura 3.8 Ilustración de la secuencia de funciones de sockets llamadas por un cliente y un servidor, usando el paradigma de flujo.

3.17 Funciones de sockets utilizadas por el cliente y el servidor

3.17.1 La función `socket`

La función `socket` crea un socket y devuelve un descriptor entero.

```
descriptor = socket(dominio, tipo, protocolo)
```

El argumento *dominio* especifica la familia de direcciones a usar con el socket. El identificador *AF_INET* especifica la versión 4 de los protocolos de Internet y el identificador *AF_INET6* especifica la versión 6. El argumento *tipo* especifica el tipo de comunicación que usará el socket: la transferencia de flujo se especifica con el valor *SOCK_STREAM* y la transferencia de mensaje sin conexión se especifica con el valor *SOCK_DGRAM*.

El argumento *protocolo* especifica un protocolo de transporte específico que el socket utiliza. Al tener un argumento *protocolo* además de un argumento *tipo*, una sola suite de protocolos puede incluir dos o más protocolos que proporcionen el mismo servicio. Los valores que pueden usarse con el argumento *protocolo* dependen de la familia de protocolos. Por lo general, *IPPROTO_TCP* se usa con *SOCK_STREAM*, e *IPPROTO_UDP* se usa con *SOCK_DGRAM*.

3.17.2 La función *send*

Tanto clientes como servidores usan la función *send* para transmitir datos. Por lo general, un cliente envía una solicitud y un servidor envía una respuesta. La función *send* tiene cuatro argumentos:

```
send(socket, datos, longitud, banderas)
```

El argumento *socket* es el descriptor del socket que se va a usar, el argumento *datos* es la dirección en la memoria de los datos a enviar, el argumento *longitud* es un entero que especifica el número de bytes de datos y el argumento *banderas* contiene bits que solicitan opciones especiales.[†]

3.17.3 La función *recv*

Tanto el cliente como el servidor usan *recv* para obtener los datos enviados por el otro. La función tiene la forma:

```
recv(socket, bufer, longitud, banderas)
```

El argumento *socket* es el descriptor del socket del cual se van a recibir los datos, el argumento *bufer* especifica la dirección en memoria en la que debe colocarse el mensaje entrante y el argumento *longitud* especifica el tamaño del bufer; por último, el argumento *banderas* permite que quien hizo la llamada pueda controlar los detalles (por ejemplo, permitir que una aplicación extraiga una copia de un mensaje entrante sin quitar el mensaje del socket). La función *recv* se bloquea hasta que llegan los datos y luego coloca en el bufer tantos bytes de datos como se especifique en el parámetro *longitud* (el valor de retorno de la llamada a la función especifica el número de bytes que se extrajeron).

3.17.4 *read* y *write* con sockets

En algunos sistemas operativos como Linux, es posible usar las funciones *read* y *write* del sistema operativo en vez de *recv* y *send*. La función *read* recibe tres argumentos que son idénticos a los primeros tres argumentos de *recv*, mientras que *write* recibe tres argumentos que son idénticos a los primeros tres argumentos de *send*.

La principal ventaja de usar *read* y *write* es la generalidad: es posible crear una aplicación que transfiera los datos hacia o desde un descriptor sin saber si el descriptor corresponde a un archivo o a un socket. De esta forma, un programador puede usar un archivo en disco local para probar un cliente o un servidor antes de intentar comunicarse a través de una red. La principal desventaja de usar *read* y *write* es que tal vez haya que modificar algún programa para poder utilizarlas en otro sistema.

[†] Muchas opciones están diseñadas para la depuración de sistemas, por lo que no están disponibles para aplicaciones convencionales de cliente y servidor.

3.17.5 La función *close*

La función *close* indica al sistema operativo que debe terminar de usar un socket.[†] Tiene la forma:

```
close(socket)
```

donde *socket* es el descriptor del socket que se va a cerrar. Si hay una conexión abierta, *close* termina la conexión (es decir, informa al otro lado). El proceso de cerrar un socket termina el uso de inmediato: se libera el descriptor, evitando que la aplicación envíe o reciba datos.

3.18 La función *connect* utilizada sólo por un cliente

Los clientes llaman a *connect* para establecer una conexión con un servidor específico. La forma es:

```
connect(socket, saddr, saddrsize)
```

El argumento *socket* es el descriptor del socket que se va a usar para la conexión. El argumento *saddr* es una estructura de *sockaddr* que especifica la dirección del servidor y el número de puerto del protocolo,[‡] y el argumento *saddrsize* especifica la longitud de la dirección del servidor medida en bytes.

Para un socket que utiliza el paradigma de flujo, *connect* inicia una conexión a nivel de transporte con el servidor especificado. El servidor debe esperar una conexión (vea la función *accept* que se describe a continuación).

3.19 Funciones de sockets utilizadas sólo por un servidor

3.19.1 La función *bind*

Cuando se crea un socket, éste no contiene información sobre la dirección local o remota o el número de puerto de protocolo. Un servidor llama a *bind* para suministrar el número de puerto del protocolo en el que el servidor esperará para hacer contacto. *Bind* recibe tres argumentos:

```
bind(socket, localaddr, addrlen)
```

El argumento *socket* es el descriptor de un socket a utilizar. El argumento *localaddr* es una estructura que especifica la dirección local que se va a asignar al socket, y el argumento *addrlen* es un entero que especifica la longitud de la dirección.

Puesto que un socket puede usarse con un protocolo indefinido, el formato de una dirección depende del protocolo que se vaya a usar. La API de sockets define la forma genérica a utilizar para represen-

[†] La interfaz Windows Sockets de Microsoft usa el nombre *closesocket* en vez de *close*.

[‡] La combinación de una dirección IP y un número de puerto de protocolo se conoce algunas veces como *dirección de punto final*.

tar direcciones y luego requiere que cada familia de protocolos especifique cómo es que sus direcciones de protocolos van a utilizar la forma genérica. El formato genérico para representar una dirección se define como estructura *sockaddr*. Aunque se liberaron varias versiones, la mayoría de los sistemas definen una estructura *sockaddr* que tiene tres campos:

```
struct sockaddr {
    u_char    sa_len;           /* longitud total de la dirección */
    u_char    sa_family;        /* familia de la dirección          */
    char     sa_data[14];       /* la dirección en sí               */
};
```

El campo *sa_len* consiste en un solo octeto que especifica la longitud de la dirección. El campo *sa_family* especifica la familia a la que pertenece una dirección (se usa la constante simbólica *AF_INET* para direcciones de Internet IPv4, y *AF_INET6* para direcciones IPv6). Por último, el campo *sa_data* contiene la dirección.

Cada familia de protocolos define el formato exacto de las direcciones a utilizar con el campo *sa_data* de una estructura *sockaddr*. Por ejemplo, IPv4 usa la estructura *sockaddr_in* para definir una dirección:

```
struct sockaddr_in {
    u_char    sin_len;           /* longitud total de la dirección */
    u_char    sin_family;        /* familia de la dirección          */
    u_short   sin_port;          /* número de puerto de protocolo */
    struct   in_addr sin_addr;  /* dirección IPv4 de computadora */
    char     sin_zero[8];        /* no se usa (se deja en cero)      */
};
```

Los primeros dos campos de la estructura *sockaddr_in* corresponden exactamente a los primeros dos campos de la estructura *sockaddr* genérica. Los últimos tres campos definen la forma exacta de una dirección de Internet. Hay que recalcar dos puntos. Primero, cada dirección identifica a una computadora y a un puerto de protocolo en esa computadora. El campo *sin_addr* contiene la dirección IP de la computadora y el campo *sin_port* contiene el número de puerto del protocolo. En segundo lugar, aunque sólo se necesitan seis bytes para almacenar una dirección de punto final IPv4 completa, la estructura *sockaddr* genérica reserva catorce bytes. Por lo tanto, el campo final en la estructura *sockaddr_in* define un campo de 8 bytes con ceros, que completan la estructura para que tenga el mismo tamaño que *sockaddr*.

Dijimos que un servidor llama a *bind* para especificar el número de puerto de protocolo en el que el servidor aceptará hacer contacto. Sin embargo, además de un número de puerto de protocolo, la estructura *sockaddr_in* contiene un campo para una dirección. Aunque el servidor puede optar por llenar una dirección específica, esto provocaría problemas cuando una computadora es multiproveedor (es decir, que tiene varias conexiones de red) puesto que tiene varias direcciones. Para que un servidor pueda operar en un *host* multiproveedor, la API de sockets incluye una constante simbólica especial, *INADDR_ANY*, que permite a un servidor especificar un número de puerto y permite al mismo tiempo el contacto con cualquiera de las direcciones de la computadora. En conclusión:

Aunque la estructura `sockaddr_in` incluye un campo para una dirección, la API de sockets proporciona una constante simbólica que permite a un servidor especificar un puerto de protocolo en cualquiera de las direcciones de la computadora.

3.19.2 La función `listen`

Después de usar `bind` para especificar un puerto de protocolo, un servidor llama a `listen` para colocar el socket en modo pasivo, con lo cual el socket está listo para esperar a hacer contacto con los clientes. La función `listen` recibe dos argumentos:

```
listen(socket, queuesize)
```

El argumento `socket` es el descriptor de un socket, mientras que el argumento `queuesize` especifica una longitud para la cola de solicitudes del socket. Un sistema operativo crea una cola de solicitudes independiente para cada socket. Al principio la cola está vacía, pero a medida que las solicitudes de clientes van llegando, cada una se coloca en la cola. Cuando el servidor pide recuperar una solicitud entrante del socket, el sistema extrae la solicitud siguiente de la cola. La longitud de la cola es importante: si está llena cuando llegue una nueva solicitud, el sistema la rechazará.

3.19.3 La función `accept`

Un servidor llama a `accept` para establecer una conexión con un cliente. Si hay una solicitud presente en la cola, `accept` regresa de inmediato; si no han llegado solicitudes, el sistema bloquea el servidor hasta que un cliente inicia una solicitud. Una vez que se acepta una conexión, el servidor usa esa conexión para interactuar con un cliente. Después de que termina la comunicación, el servidor cierra la conexión.

La función `accept` tiene la forma:

```
newsock = accept(socket, caddress, caddresslen)
```

El argumento `socket` es el descriptor del socket que el servidor creó y está vinculado a un puerto de protocolo específico. El argumento `caddress` es la dirección de una estructura de tipo `sockaddr`, mientras que `caddresslen` es un apuntador a un entero. La función `accept` llena los campos del argumento `caddress` con la dirección del cliente que formó la conexión y determina a `caddresslen` según la longitud de la dirección. Finalmente, `accept` crea un nuevo socket para la conexión y devuelve el descriptor del nuevo socket a quien hizo la llamada. El servidor usa el nuevo socket para comunicarse con el cliente y luego cierra el socket cuando termine. Mientras tanto, el socket original del servidor permanece sin cambios; después de comunicarse con un cliente, el servidor usa el socket original para aceptar la siguiente conexión de un cliente. En consecuencia, el socket original sólo se usa para aceptar solicitudes y toda la comunicación con un cliente ocurre sobre el nuevo socket creado por `accept`.

3.20 Funciones de sockets utilizadas con el paradigma de mensajes

Las funciones de sockets que se utilizan para enviar y recibir mensajes son más complicadas que las que se utilizan con el paradigma de flujo, ya que hay muchas opciones disponibles. Por ejemplo, un emisor puede elegir si desea almacenar la dirección del receptor en el socket y simplemente enviar datos, o especificar la dirección del receptor cada vez que se transmite un mensaje. Además, una función permite a un emisor colocar tanto la dirección como el mensaje en la estructura y pasar la dirección de ésta como un argumento, y otra función permite a un emisor pasar la dirección y el mensaje como argumentos independientes.

3.20.1 Funciones de sockets *sendto* y *sendmsg*

Las funciones *sendto* y *sendmsg* permiten a un cliente o servidor enviar un mensaje usando un socket desconectado; ambas requieren que quien hace la llamada especifique un destino. La función *sendto* usa argumentos separados para el mensaje y la dirección de destino:

```
sendto(socket, datos, longitud, banderas, direcdestino, longdirec)
```

Los primeros cuatro argumentos corresponden a los cuatro argumentos de la función *send*; los dos finales especifican la dirección de un destino y la longitud de esa dirección. El argumento *direcdestino* corresponde a una estructura *sockaddr* (específicamente, *sockaddr_in*).

La función *sendmsg* realiza la misma operación que *sendto*, pero abrevia los argumentos mediante la definición de una estructura. La lista de argumentos más corta puede hacer que los programas que usan *sendmsg* sean más fáciles de leer:

```
sendmsg(socket, msgstruct, banderas)
```

El argumento *msgstruct* es una estructura que contiene información sobre la dirección de destino, la longitud de la dirección, el mensaje a enviar y la longitud del mensaje:

```
struct msgstruct { /* estructura usada por sendmsg */
    struct sockaddr *m_saddr; /* apunta direcc destino */
    struct datavec *m_dvec; /* apunta mensaje (vector) */
    int m_dvlength; /* núm. elementos en vector */
    struct access *m_rights; /* apunta lista derechos acceso */
    int m_alength; /* núm. elementos en la lista */
};
```

Los detalles de la estructura de mensajes no son importantes; debe verse como una forma de combinar muchos argumentos en una sola estructura. La mayoría de las aplicaciones usan sólo los primeros tres campos, que especifican una dirección de protocolo de destino, una lista de elementos de datos que constituyen el mensaje y el número de elementos en la lista.

3.20.2 Funciones *recvfrom* y *recvmsg*

Un socket desconectado puede usarse para recibir mensajes de un conjunto indeterminado de clientes. En dichos casos el sistema devuelve la dirección del emisor junto con cada mensaje entrante (el receptor usa la dirección para enviar una respuesta). La función *recvfrom* tiene argumentos que especifican una ubicación para el siguiente mensaje entrante y la dirección del emisor:

```
recvfrom(socket, bufer, longitud, banderas, sndraddr, saddrlen)
```

Los primeros cuatro argumentos corresponden a los argumentos de *recv*; los dos argumentos adicionales (*sndraddr* y *saddrlen*) se usan para registrar la dirección de Internet del emisor y su longitud. El argumento *sndraddr* es un apuntador a una estructura *sockaddr* en la que el sistema escribe la dirección del emisor, mientras que el argumento *saddrlen* es un apuntador a un entero que el sistema usa para registrar la longitud de la dirección. Tenga en cuenta que *recvfrom* registra la dirección del emisor exactamente de la misma forma que *sendto* espera, lo que facilita la transmisión de una respuesta.

La función *recvmsg*, que es la contraparte de *sendmsg*, opera como *recvfrom* pero requiere menos argumentos. Tiene la forma:

```
recvmsg(socket, msgstruct, banderas)
```

Aquí, el argumento *msgstruct* proporciona la dirección de una estructura que contiene la dirección de un mensaje entrante, así como las ubicaciones de la dirección de Internet del emisor. La dirección *msgstruct* registrada por *recvmsg* usa exactamente el mismo formato que la estructura requerida por *sendmsg*, con lo que se facilita la acción de recibir una solicitud, registrar la dirección del emisor y luego usar la dirección registrada para enviar una respuesta.

3.21 Otras funciones de sockets

La API de sockets contiene una variedad de funciones menores de soporte que no se describen anteriormente. Por ejemplo, una vez que un servidor acepta una solicitud de conexión entrante, el servidor puede llamar a *getpeername* para obtener la dirección del cliente remoto que inició la conexión. Un cliente o servidor también pueden llamar a *gethostname* para obtener información sobre la computadora en la que se está ejecutando.

Se usan dos funciones genéricas para manipular las opciones de sockets. La función *setsockopt* almacena valores en las opciones de un socket y la función *getsockopt* obtiene los valores de las opciones actuales. Las opciones se usan primordialmente para manejar casos especiales (por ejemplo, para incrementar el tamaño del bufer interno).

Hay dos funciones que hacen una traducción entre las direcciones de Internet y los nombres de computadoras. La función *gethostbyname* devuelve la dirección de Internet para una computadora a partir de su nombre. Los clientes a menudo llaman a *gethostbyname* para traducir el nombre que escribe un usuario en la dirección IP correspondiente. La función *gethostbyaddr* proporciona una asignación

inversa: a partir de la dirección IP de una computadora devuelve el nombre de ésta. Los clientes y servidores pueden usar `gethostbyaddr` para traducir una dirección en un nombre que el usuario pueda entender.

3.22 Sockets, hilos y herencia

La API de sockets funciona bien con servidores concurrentes. Aunque los detalles dependen del sistema operativo que se use, las implementaciones de la API de sockets se apegan al siguiente principio de herencia:

Cada nuevo hilo que se crea hereda una copia de todos los sockets abiertos del hilo que lo creó.

La implementación de sockets usa un mecanismo de *conteo de referencia* para controlar cada socket. Cuando se crea un socket por primera vez, el sistema establece el conteo de referencia del socket en 1, y el socket existe mientras el conteo de referencias sea positivo. Cuando un programa crea un hilo adicional, el hilo hereda el apuntador para cada socket abierto que el programa posee, y el sistema incrementa el conteo de referencias de cada socket en 1. Cuando un hilo llama a *close*, el sistema disminuye el conteo de referencia del socket; si el conteo de referencias llega a cero, el socket se elimina.

En términos de un servidor concurrente, el hilo principal posee el socket que se utiliza para aceptar conexiones entrantes. Cuando llega una solicitud de conexión, el sistema crea un nuevo socket para la nueva conexión y el hilo principal crea un nuevo hilo para manejar la conexión. Justo después de la creación de un hilo, ambos hilos tienen acceso al socket original y al nuevo socket, y el conteo de referencias de cada socket es 2. El hilo principal llama a *close* para el nuevo socket y el hilo de servicio llama a *close* para el socket original, reduciendo el conteo de referencias de cada uno a 1. Por último, cuando termina de interactuar con un cliente, el hilo de servicio llama a *close* sobre el nuevo socket y reduce el conteo de referencias a cero, provocando la eliminación del socket. Por consiguiente, podemos resumir el tiempo de vida de los sockets en un servidor concurrente:

El socket original utilizado para aceptar conexiones existe mientras que se ejecute el hilo servidor principal; un socket utilizado para una conexión específica existe mientras que exista el hilo para manejar esa conexión.

3.23 Resumen

En Internet, todos los servicios se suministran mediante aplicaciones, las cuales usan un paradigma de flujo o un paradigma de mensajes para comunicarse. El paradigma de flujo garantiza la entrega de una secuencia de bytes en orden, pero puede elegir cuántos bytes se deben pasar a cada receptor en cada lote. El paradigma de mensajes mantiene los límites pero permite que los mensajes se pierdan, se dupliquen, se retrasen o se entreguen desordenados.

El modelo básico de comunicación que utilizan las aplicaciones de red se conoce como modelo cliente-servidor. Un programa que espera un contacto en forma pasiva se denomina servidor, y un programa que inicia en forma activa el contacto con un servidor se denomina cliente.

A cada computadora se le asigna una dirección única y a cada servicio (como el correo electrónico o el acceso Web) se le asigna un identificador único conocido como número de puerto de protocolo. Cuando un servidor inicia, especifica un número de puerto de protocolo; cuando un cliente hace contacto con un servidor, especifica la dirección de la computadora en la que se ejecuta el servidor, así como el número de puerto de protocolo que usa el mismo.

Un solo cliente puede acceder a más de un servicio y puede acceder a servidores en varias máquinas, mientras que un servidor para un servicio puede convertirse en cliente para otros servicios. Los diseñadores y programadores deben tener cuidado de evitar dependencias circulares entre los servidores.

Una interfaz de programación de aplicaciones (API) especifica los detalles sobre cómo interactúa el programa de aplicación con el software de protocolo. Aunque los detalles dependen del sistema operativo, la API de sockets es un estándar. Un programa crea un socket y luego invoca a una serie de funciones para que usen el socket. Un servidor que utiliza el paradigma de flujo llama funciones de sockets como *socket*, *bind*, *listen*, *accept*, *recv*, *send* y *close*, mientras que un cliente llama a *socket*, *connect*, *send*, *recv* y *close*.

Puesto que muchos servidores son concurrentes, los sockets están diseñados para trabajar con aplicaciones concurrentes. Cuando se crea un nuevo hilo, éste hereda el acceso a todos los sockets que pertenecían al hilo creador.

EJERCICIOS

- 3.1** ¿Cuáles son los dos paradigmas básicos de comunicación que se utilizan en Internet?
- 3.2** Proporcione seis características de la comunicación por flujos en Internet.
- 3.3** Proporcione seis características de la comunicación por mensajes en Internet.
- 3.4** Si un emisor usa el paradigma de flujo y siempre envía 1024 bytes a la vez, ¿de qué tamaño pueden ser los bloques que Internet entrega a un receptor?
- 3.5** Si un emisor desea tener copias de cada bloque de datos que se envía a tres receptores, ¿qué paradigma debería elegir?
- 3.6** ¿Cuáles son los cuatro aspectos sorprendentes de la semántica de entrega de mensajes de Internet?
- 3.7** Proporcione el algoritmo general utilizado por un sistema orientado a la conexión.
- 3.8** Cuando dos aplicaciones se comunican a través de Internet, ¿cuál es el servidor?
- 3.9** Compare y contraste una aplicación cliente y una aplicación servidor, sintetizando las características de cada una.
- 3.10** ¿Cuál es la diferencia entre un servidor y una computadora tipo servidor?
- 3.11** ¿Pueden fluir los datos de un cliente a un servidor? Explique.
- 3.12** Enumere las posibles combinaciones de clientes y servidores que puede ejecutar una misma computadora.
- 3.13** ¿Pueden todas las computadoras ejecutar varios servicios de manera efectiva? ¿Por qué sí o por qué no?

- 3.14** ¿Cuáles son los dos identificadores que se utilizan para especificar un servidor en particular?
- 3.15** Enliste los pasos que un cliente utiliza para contactar con un servidor después de que un usuario especifica un nombre de dominio para el servidor.
- 3.16** ¿Qué función básica del sistema operativo utiliza un servidor concurrente para manejar las solicitudes simultáneas de múltiples clientes?
- 3.17** ¿Qué problema de rendimiento motiva la comunicación de igual a igual?
- 3.18** Mencione dos sistemas operativos que ofrezcan la API de sockets.
- 3.19** Una vez que se crea un socket, ¿cómo es que una aplicación hace referencia al mismo?
- 3.20** ¿Cuáles son las principales funciones de una API de sockets?
- 3.21** Proporcione una secuencia típica de llamadas de sockets usada por un cliente y una secuencia típica usada por un servidor.
- 3.22** ¿A qué funciones de sockets corresponden *read* y *write*?
- 3.23** ¿Alguna vez un cliente usará la función *bind*? Explique.
- 3.24** ¿Por qué se utiliza la constante simbólica *INADDR_ANY*?
- 3.25** ¿*Sendto* se utiliza con un paradigma de flujo o con un paradigma de mensaje?
- 3.26** Suponga que un socket está abierto y que se crea un nuevo hilo. ¿Podrá el nuevo hilo utilizar el socket?
- 3.27** Analice el servidor web del Apéndice 1 y construya un servidor equivalente utilizando la API de sockets.
- 3.28** Implemente la API simplificada del Apéndice 1 utilizando funciones de sockets.

Contenido del capítulo

- 4.1 Introducción, 49
- 4.2 Protocolos de la capa de aplicación, 49
- 4.3 Representación y transferencia, 50
- 4.4 Protocolos Web, 51
- 4.5 Representación de documentos con HTML, 52
- 4.6 Localizadores uniformes de recursos e hipervínculos, 54
- 4.7 Transferencia de documentos Web con HTTP, 55
- 4.8 Uso de la memoria caché en los navegadores, 57
- 4.9 Arquitectura de los navegadores, 59
- 4.10 Protocolo de transferencia de archivos (FTP), 59
- 4.11 Paradigma de comunicación de FTP, 60
- 4.12 Correo electrónico, 63
- 4.13 El protocolo simple de transferencia de correo (SMTP), 64
- 4.14 Proveedores de servicios de Internet (ISP), servidores de correo y acceso al correo, 66
- 4.15 Protocolos de acceso al correo (POP, IMAP), 67
- 4.16 Estándares de representación de correo electrónico (RFC2822, MIME), 67
- 4.17 Sistema de nombres de dominio (DNS), 69
- 4.18 Nombres de dominio que comienzan con un nombre de servicio, 71
- 4.19 La jerarquía del DNS y el modelo servidor, 72
- 4.20 Resolución de nombres, 72
- 4.21 Uso de la memoria caché en servidores del DNS, 74
- 4.22 Tipos de entradas del DNS, 75
- 4.23 Alias y registros de recursos CNAME, 76
- 4.24 Abreviaciones y el DNS, 76
- 4.25 Nombres de dominio internacionalizados, 77
- 4.26 Representaciones extensibles (XML), 78
- 4.27 Resumen, 79

4

Aplicaciones tradicionales de Internet

4.1 Introducción

El capítulo anterior presenta los temas de las aplicaciones de Internet y la programación de redes. Explica que los servicios de Internet se definen mediante programas de aplicación y caracteriza el modelo cliente-servidor que usan dichos programas para interactuar. El capítulo también cubre la API de sockets.

Este capítulo continúa explicando las aplicaciones de Internet. Define el concepto de un protocolo de transferencia y explica cómo es que las aplicaciones implementan este tipo de protocolos. Por último, el capítulo considera ejemplos de aplicaciones de Internet que se han estandarizado y describe el protocolo de transferencia que usa cada una.

4.2 Protocolos de la capa de aplicación

Cada vez que un programador crea dos aplicaciones que se comunican a través de una red, debe especificar detalles como:

- La sintaxis y semántica de los mensajes que pueden intercambiarse
- Si el cliente o el servidor inicia la interacción
- Las acciones a realizar si surge un error
- Cómo saben los dos lados cuándo terminar la comunicación

Para especificar los detalles de la comunicación, un programador define un *protocolo de capa de aplicación*. Hay dos tipos principales de protocolos de capa de aplicación que dependen del uso que se les vaya a dar:

- *Servicio privado.* Un programador o una empresa crea un par de aplicaciones que se comunican a través de Internet con la intención de que nadie más tenga permitido crear software cliente o servidor para ese servicio. No hay necesidad de publicar ni distribuir una especificación de protocolo formal para definir la interacción, ya que nadie externo necesita entender los detalles. De hecho, si la interacción entre las dos aplicaciones es lo bastante directa, tal vez no haya un documento de protocolo interno.
- *Servicio estandarizado.* Un servicio de Internet que se crea con la expectativa de que muchos programadores crearán software servidor para ofrecer el servicio o un software cliente para acceder a éste. En tales casos, el protocolo de capa de aplicación debe documentarse en forma independiente de cualquier implementación. Además, la especificación debe ser precisa e inequívoca, de modo que puedan construirse aplicaciones cliente y servidor que *interoperen* correctamente.

El tamaño de una especificación de protocolo depende de la complejidad del servicio; la especificación de un servicio sencillo puede caber en una sola página de texto. Por ejemplo, los protocolos de Internet incluyen un servicio de aplicación estandarizado conocido como *DAYTIME*, el cual permite a un cliente buscar la fecha y hora local en la ubicación del servidor. El protocolo es simple: un cliente forma una conexión a un servidor, el servidor envía una representación ASCII de la fecha y hora, y el servidor cierra la conexión. Por ejemplo, un servidor podría enviar una cadena como:

Lun Sep 9 20:18:37 2014

El cliente lee los datos de la conexión hasta encontrar un *fin de archivo*.
En conclusión:

Para que las aplicaciones para los servicios estandarizados puedan interesar, se crea un estándar de protocolo de capa de aplicación en forma independiente de cualquier implementación.

4.3 Representación y transferencia

Los protocolos de la capa de aplicación especifican dos aspectos de interacción: la representación y la transferencia. La figura 4.1 explica la distinción entre ambos.

Aspecto	Descripción
Representación de datos	Sintaxis de los elementos de datos que se intercambian; forma específica usada durante la transferencia; traducción de enteros, caracteres y archivos que se envían entre computadoras
Transferencia de datos	Interacción entre cliente y servidor; sintaxis del mensaje y semántica; manejo de errores de intercambio válido e inválido; terminación de la interacción

Figura 4.1 Dos aspectos clave de un protocolo de capa de aplicación.

Para un servicio básico, un solo estándar de protocolo puede especificar ambos aspectos; los servicios más complejos usan estándares de protocolos independientes para especificar cada aspecto. Por ejemplo, el protocolo DAYTIME descrito anteriormente usa un solo estándar para especificar que una fecha y hora se representan como una cadena ASCII, y que la transferencia consiste en un servidor que envía la cadena y luego cierra la conexión. La siguiente sección explica que los servicios más complejos definen protocolos independientes para describir la sintaxis de los objetos y la transferencia de éstos. Los diseñadores de protocolos hacen la distinción clara entre los dos aspectos:

Como una convención, la palabra transferencia en el título de un protocolo de capa de aplicación significa que el protocolo especifica el aspecto de transferencia de datos de la comunicación.

4.4 Protocolos Web

World Wide Web es uno de los servicios más utilizados en Internet. Debido a la complejidad de la Web, se idearon muchos estándares de protocolos para especificar varios aspectos y detalles de ésta. La figura 4.2 muestra una lista de los tres estándares principales.

Estándar	Propósito
Lenguaje de marcación de hipertexto (HTML)	Un estándar de representación utilizado para especificar el contenido y la distribución de una página Web
Localizador uniforme de recursos (URL)	Un estándar de representación que especifica el formato y el significado de los identificadores de una página Web
Protocolo de transferencia de hipertexto (HTTP)	Un protocolo de transferencia que especifica cómo interactúa un navegador con un servidor Web para transferir datos

Figura 4.2 Tres estándares clave utilizados por el servicio World Wide Web.

4.5 Representación de documentos con HTML

El *lenguaje de marcación de hipertexto* o *HTML* es un estándar de representación que especifica la sintaxis de una página Web. HTML tiene las siguientes características generales:

- Está representado por texto
- Describe páginas Web que contienen elementos multimedia
- Sigue un paradigma declarativo en vez de un paradigma por procedimientos
- Brinda especificaciones de las marcas en vez de un formato
- Permite incrustar un hipervínculo en un objeto cualquiera
- Permite que un documento incluya metadatos

Aunque un documento de HTML consiste en un archivo de texto, el lenguaje permite a un programador especificar una página Web compleja que contenga gráficos, audio y video, además de texto. De hecho, para ser precisos, los diseñadores debieron haber usado *hipermedios* en el nombre en vez de *hipertexto*, debido a que HTML permite que un objeto cualquiera (como una imagen) contenga un vínculo hacia otra página Web, lo que se conoce algunas veces como *hipervínculo*.

HTML se clasifica como *declarativo* debido a que el lenguaje sólo permite especificar lo que debe hacerse y no cómo hacerlo. HTML se clasifica como *lenguaje de marcación* debido a que sólo proporciona los lineamientos generales o *marcas* para la visualización y no incluye instrucciones detalladas de formato. Por ejemplo, HTML permite que una página especifique el nivel de importancia de un encabezado, pero HTML no requiere que el autor especifique los detalles de tipografía (como la fuente, el tamaño de letra en puntos y el espaciado entre líneas) exactos que deben usarse para el encabezado.[†] En esencia, un navegador es libre de elegir la mayoría de los detalles de visualización. El uso de un lenguaje de marcación es importante ya que permite que un navegador adapte la página al hardware de visualización donde se va a desplegar. Por ejemplo, se puede dar formato a una página para una pantalla de alta o baja resolución, para una ventana con una relación de aspecto específica, para una pantalla grande o para un pequeño dispositivo portátil, como un teléfono inteligente o una tableta digital.

En conclusión:

El lenguaje de marcación de hipertexto (HTML) es un estándar de representación para las páginas Web. Para que una página pueda desplegarse en un servicio cualquiera, HTML otorga lineamientos generales para la visualización y permite que un navegador seleccione los detalles.

Para especificar la marcación, HTML usa *etiquetas* incrustadas en el documento. Las etiquetas, que constan de un término encerrado entre los símbolos *menor que* (<) y *mayor que* (>), proporcionan la estructura para el documento, así como algunas sugerencias de formato. Las etiquetas controlan toda la visualización; es posible incluso que existan espacios en blanco (como líneas adicionales y caracteres vacíos) en cualquier punto del documento de HTML sin que esto afecte la versión final que despliega el navegador.

Un documento de HTML comienza con la etiqueta <HTML> y termina con la etiqueta </HTML>. El par de etiquetas <HEAD> y </HEAD> encierran el encabezado, mientras que el par de etiquetas <BODY>

[†] HTML incluye extensiones que permiten la especificación de dichos detalles de tipografía, pero no son imprescindibles.

y </BODY> encierran el cuerpo de la página. En el encabezado, las etiquetas <TITLE> y </TITLE> encierran el texto que forma el título del documento. La figura 4.3 ilustra la forma general de un documento de HTML.[†]

```
<HTML>
  <HEAD>
    <TITLE>
      texto que forma el título del documento
    </TITLE>
  </HEAD>
  <BODY>
    el cuerpo del documento aparece aquí
  </BODY>
</HTML>
```

Figura 4.3 La forma general de un documento de HTML.

HTML usa la etiqueta *IMG* para codificar una referencia hacia una imagen externa. Por ejemplo, la etiqueta:

```
<IMG SRC="icono_casa.jpg">
```

especifica que el archivo *icono_casa.jpg* contiene una imagen que el navegador debe insertar en esa sección del documento. Pueden especificarse parámetros adicionales en una etiqueta *IMG* para determinar la alineación de la figura en relación con el texto circundante. Por ejemplo, la figura 4.4 ilustra la salida del siguiente código HTML, que alinea el texto con la parte media de la figura:

Éste es el ícono de una casa.

Un navegador posiciona la imagen en sentido vertical, de modo que el texto se alinee con la parte media de la imagen.



Figura 4.4 Ilustración de la alineación de figuras en HTML.

[†] HTML no distingue letras minúsculas de mayúsculas en las etiquetas; en los ejemplos usamos mayúsculas para hacer énfasis.

4.6 Localizadores uniformes de recursos e hipervínculos

El servicio Web usa una forma sintáctica conocida como *localizador uniforme de recursos (URL)* para especificar una página Web. La forma general de un URL es:

protocolo://nombre_computadora:puerto/nombre_documento?parámetros

donde *protocolo* se refiere al nombre del protocolo utilizado para acceder al documento, *nombre_computadora* es el nombre de dominio de la computadora en la que reside el documento, *:puerto* es un número opcional de puerto de protocolo en el que el servidor está esperando la comunicación, *nombre_documento* es el nombre opcional del documento dentro de la computadora especificada y *?parámetros* son los parámetros opcionales de la página.

Por ejemplo, el URL

`http://www.netbook.cs.purdue.edu/ejemplo.html`

especifica el protocolo *http*, una computadora llamada *www.netbook.cs.purdue.edu* y un archivo llamado *ejemplo.html*.

Los URL comunes que escribe un usuario omiten muchas de las partes. Por ejemplo, el URL

`www.netbook.cs.purdue.edu`

omite el protocolo (se entiende que es *http*), el puerto (se entiende que es el 80), el nombre del documento (se entiende que es *index.html*) y los parámetros (se entiende que no hay).

Un URL contiene la información que necesita un navegador para obtener una página. El navegador usa los caracteres separadores dos puntos, barra diagonal y signo de interrogación para dividir el URL en cinco componentes: un protocolo, un nombre de computadora, un número de puerto de protocolo, un nombre de documento y los parámetros adicionales. El navegador usa el nombre de la computadora y el número de puerto del protocolo para establecer una conexión con el servidor en el que reside la página, y usa el nombre del documento junto con los parámetros para solicitar una página específica.

En HTML, una etiqueta de *ancla* usa los URL para ofrecer la posibilidad de incluir hipervínculos (es decir, vincular un documento Web con otro). El siguiente ejemplo muestra un documento de HTML con un ancla que rodea el nombre *Pearson*:

```
Este libro es una publicación de  
<A HREF="http://www.pearson.com">  
Pearson</A>, una de las editoriales  
de libros de computación más grandes.
```

El ancla hace referencia al URL *http://www.pearson.com*. Cuando se visualiza en una pantalla, la entrada de HTML produce:

Este libro es una publicación de Pearson, una de las editoriales
de libros de computación más grandes.

4.7 Transferencia de documentos Web con HTTP

El *protocolo de transferencia de hipertexto* o *HTTP* es el principal protocolo de transferencia que un navegador usa para interactuar con un servidor Web. En términos del modelo cliente-servidor, un navegador es un cliente que extrae un nombre de servidor de un URL y se contacta con ese servidor. La mayoría de los URL contienen una referencia de protocolo explícita para *http://* u omiten por completo el protocolo, en cuyo caso se entiende que es HTTP.

Podemos caracterizar al protocolo HTTP de la siguiente manera:

- Usa mensajes de control basados en texto
- Transfiere archivos de datos binarios
- Puede descargar o enviar datos
- Incorpora el uso de la memoria caché

Una vez que establece una conexión, un navegador envía una *solicitud* HTTP al servidor. La figura 4.5 muestra una lista con los cuatro tipos principales de solicitud:

Solicitud	Descripción
GET	Solicita un documento; el servidor responde enviando la información de estado, seguida de una copia del documento.
HEAD	Solicita la información de estado; el servidor responde enviando la información de estado, pero no envía una copia del documento
POST	Envía datos a un servidor; el servidor adjunta los datos a un elemento especificado (por ejemplo, adjunta un mensaje a una lista)
PUT	Envía datos a un servidor; el servidor usa los datos para reemplazar por completo el elemento especificado (es decir, sobrescribe los datos anteriores)

Figura 4.5 Los cuatro tipos principales de solicitud de HTTP.

La forma más común de interacción comienza cuando un navegador solicita una página del servidor. El navegador envía una solicitud *GET* a través de la conexión y el servidor responde enviando un encabezado, una línea en blanco y el documento solicitado. En HTTP, tanto la solicitud como el encabezado que se utilizan en una respuesta constan cada uno de información de texto. Por ejemplo, una solicitud *GET* tiene la siguiente forma:

GET /elemento versión CRLF

donde *elemento* proporciona el URL del elemento solicitado, *versión* especifica la versión del protocolo (por lo general HTTP/1.0 o HTTP/1.1) y *CRLF* denota dos caracteres ASCII, *retorno* y *salto de línea*, que se usan para indicar el final de una línea de texto.

La información de versión es importante en HTTP, ya que permite cambiar el protocolo sin perder la compatibilidad con versiones anteriores. Por ejemplo, cuando un navegador que usa la versión 1.0 del protocolo interactúa con un servidor que usa una versión posterior, el servidor se revierte a la versión anterior del protocolo y formula una respuesta apropiada. En conclusión:

Al usar HTTP, un navegador envía la información de versión, la cual permite a un servidor elegir la versión más reciente del protocolo que tanto el navegador como el servidor puedan entender.

La primera línea de un encabezado de respuesta contiene un código de estado que indica al navegador si el servidor atendió la solicitud o no. Si la solicitud no se atendió correctamente o el elemento solicitado no estaba disponible, el código de estado señala el problema. Por ejemplo, si un servidor no puede encontrar el elemento solicitado, devuelve el conocido código de estado 404. Si la solicitud fue atendida, el servidor devuelve el código de estado 200; las líneas adicionales del encabezado brindan más información sobre el elemento, como su longitud, cuándo se modificó por última vez y el tipo de contenido que incluye. La figura 4.6 muestra el formato general de las líneas en el encabezado de una respuesta básica.

```
HTTP/1.0 código_estado cadena_estado CRLF
Server: identificación_servidor CRLF
Last-Modified: fecha_en_que_se_modificó_el_documento CRLF
Content-Length: tamañodatos CRLF
Content-Type: tipo_documento CRLF
CRLF
```

Figura 4.6 Formato general de las líneas en el encabezado de una respuesta básica.

El campo *código_estado* es un valor numérico que se representa como cadena de caracteres de dígitos decimales que denota un estado, y *cadena_estado* es la correspondiente explicación legible para un ser humano. La figura 4.7 muestra una lista de ejemplos de códigos y cadenas de estado de uso común. El campo *identificación_servidor* contiene una cadena descriptiva que ofrece una descripción del servidor legible para el ser humano, y que posiblemente incluye el nombre de dominio del servidor. El campo *tamañodatos* en el encabezado *Content-Length* especifica el tamaño del elemento de datos que le sigue, medido en bytes. El campo *tipo_documento* contiene una cadena que informa al navegador sobre el contenido del documento. La cadena contiene dos elementos separados por una barra diagonal: el tipo del documento y su representación. Por ejemplo, cuando un servidor devuelve un documento HTML, el *tipo_documento* es *text/html*, y cuando el servidor devuelve un archivo jpeg, el tipo es *image/jpeg*.

Código de estado	Cadena de estado correspondiente
200	OK
400	Solicitud incorrecta
404	Elemento no encontrado

Figura 4.7 Ejemplos de códigos de estado que se utilizan en HTTP.

La figura 4.8 muestra un ejemplo de salida de un servidor Web Apache. El elemento solicitado es un archivo de texto que contiene dieciséis caracteres (es decir, el texto *Ésta es una prueba*, más un carácter para *salto de línea*). Aunque la solicitud GET especifica que se trata de HTTP versión 1.0, el servidor ejecuta la versión 1.1. El servidor devuelve nueve líneas de encabezado, una línea en blanco y el contenido del archivo.

```
HTTP/1.1 200 OK
Date: Sat, 1 Aug 2013 10:30:17 GMT
Server: Apache/1.3.37 (Unix)
Last-Modified: Thu, 15 Mar 2012 07:35:25 GMT
ETag: "78595-81-3883bbe9"
Accept-Ranges: bytes
Content-Length: 16
Connection: close
Content-Type: text/plain

Ésta es una prueba.
```

Figura 4.8 Ejemplo de respuesta HTTP de un servidor Web Apache.

4.8 Uso de la memoria caché en los navegadores

Debido a que muchos de los usuarios de Web tienden a visitar los mismos sitios en forma recurrente, el uso de la memoria caché brinda una optimización importante para el acceso a Web. Gran parte del contenido en un sitio consiste en imágenes grandes que usan los estándares *formato de imágenes de gráficos (GIF)* o del *Grupo unido de expertos en fotografía (JPEG)*. Dichas imágenes incluyen a menudo fondos o letreros que no cambian muy seguido. En conclusión:

Un navegador Web puede reducir considerablemente los tiempos de descarga, guardando una copia de cada imagen en una memoria caché del disco duro del usuario y usando esta copia la siguiente vez que se solicita esa misma página.

Ahora surge una pregunta: ¿qué ocurre si el archivo del servidor Web cambia después de que un navegador almacenó una copia en su caché? Es decir, ¿cómo puede un navegador saber cuando su copia en caché está *obsoleta*? El código de la figura 4.8 nos da una pista de la respuesta: el encabezado *Last-Modified*. Cada vez que un navegador obtiene un documento de un servidor Web, el encabezado especifica la última vez que se modificó el documento. El navegador guarda la información de la fecha *Last-Modified* junto con la copia en caché. Antes de usar un documento de la caché local, un navegador envía una solicitud *HEAD* al servidor y compara la fecha *Last-Modified* de la copia del servidor con la fecha *Last-Modified* de la copia en caché. Si la versión en caché es obsoleta, el navegador descarga la nueva versión del archivo. El algoritmo 4.1 resume la caché.

Algoritmo 4.1

Dado:

Un URL de un elemento en una página Web

Se obtiene:

Una copia de la página

Método:

```
if (elemento no está en la caché local) {
    Emitir solicitud GET y colocar una copia en la caché;
} else {
    Emitir solicitud HEAD para el servidor;
    if (elemento en caché está actualizado) {
        usar elemento en caché;
    } else {
        emitir solicitud GET y colocar una copia en la caché;
    }
}
```

Algoritmo 4.1 La caché en un navegador se utiliza para reducir tiempos de descarga.

El algoritmo omite varios pequeños detalles. Por ejemplo, HTTP permite que un sitio Web incluya un encabezado *No-cache* que especifique que un elemento dado no debe colocarse en caché. Además, los navegadores no ponen elementos pequeños en caché debido a que si se ponen muchos elementos pequeños en la misma es posible que se incrementen los tiempos de búsqueda y el tiempo para descargar un elemento pequeño con una solicitud GET es aproximadamente el mismo que el tiempo para realizar una solicitud HEAD.

4.9 Arquitectura de los navegadores

Puesto que no sólo proporciona servicios generales, sino que también soporta una interfaz gráfica, un navegador Web requiere de cierta complejidad. Desde luego que un navegador debe entender HTTP, pero también debe brindar soporte para otros protocolos. En especial y puesto que un URL puede especificar un protocolo cualquiera, un navegador debe contener código cliente para cada uno de los protocolos utilizados. Para cada servicio, el navegador debe saber cómo interactuar con un servidor y cómo interpretar respuestas. Por ejemplo, un navegador debe saber cómo acceder al servicio FTP que describiremos en la siguiente sección. La figura 4.9 ilustra los componentes que incluye un navegador.

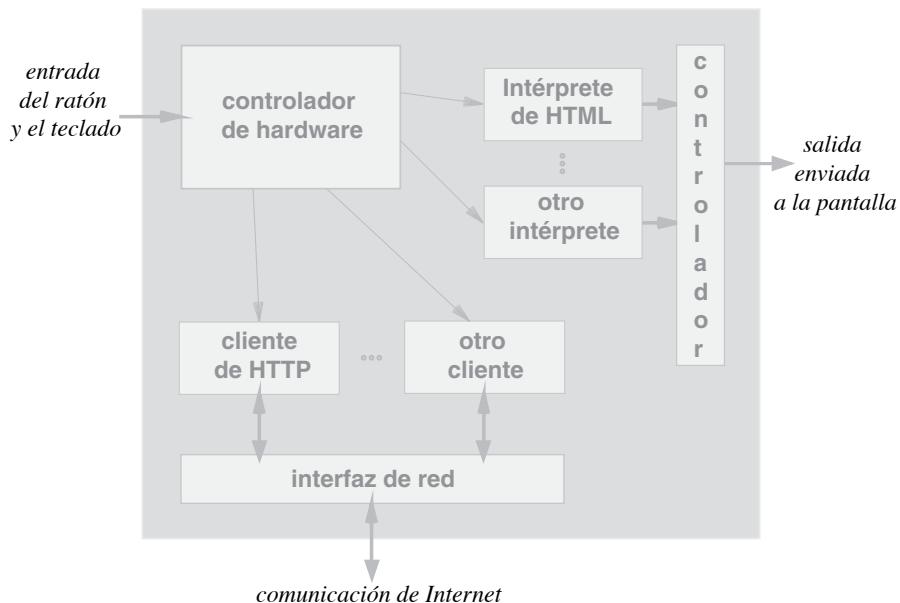


Figura 4.9 Arquitectura de un navegador que puede acceder a múltiples servicios.

4.10 Protocolo de transferencia de archivos (FTP)

Un *archivo* es la abstracción fundamental de almacenamiento. Puesto que un archivo puede contener cualquier clase de objeto (un documento, un programa de computadora, una imagen o un clip de video), un servicio que envía una copia de un archivo de una computadora a otra representa un poderoso mecanismo para el intercambio de datos. Usamos el término *transferencia de archivos* para dicho servicio.

La transferencia de archivos a través de Internet es complicada, ya que las computadoras son heterogéneas, lo que significa que cada sistema de cómputo establece sus propias representaciones de archivos, información sobre los tipos de archivos, convención para los nombres y mecanismos de acceso a los archivos. En algunas computadoras, la extensión *.jpg* se utiliza para una imagen JPEG y, en otras,

la extensión para el mismo tipo de archivo es *.jpeg*. En algunos sistemas, cada línea de un archivo de texto se termina mediante un carácter *salto de línea*, mientras que otros sistemas requieren *retorno* seguido de *salto de línea*. Algunos sistemas usan la barra diagonal (/) como separador en los nombres de archivos, mientras que otros usan una diagonal inversa (\). Además, un sistema operativo puede definir un conjunto de cuentas de usuario, cada una de las cuales recibe el derecho de acceder únicamente a ciertos archivos. Sin embargo, la información de las cuentas difiere entre una computadora y otra, por lo que el usuario *X* en una computadora no es el mismo que el usuario *X* en otra.

El servicio de transferencia de archivos estándar en Internet usa el *protocolo de transferencia de archivos (FTP)*. FTP puede caracterizarse como:

- *Cualquier contenido de archivo.* FTP puede transferir cualquier tipo de datos, incluyendo documentos, imágenes, música o video almacenado.
- *Transferencia bidireccional.* FTP puede usarse para descargar archivos (transferir de servidor a cliente) o enviar archivos (transferir de cliente a servidor).
- *Soporte para autenticación y propiedad.* FTP permite que cada archivo tenga restricciones de propiedad y acceso, y respeta esas restricciones.
- *Habilidad de navegar por carpetas.* FTP permite que un cliente obtenga el contenido de un directorio determinado (por ejemplo, una carpeta).
- *Mensajes de control basados en texto.* Al igual que muchos otros servicios de aplicaciones de Internet, los mensajes de control que se intercambian entre un cliente FTP y un servidor se envían como texto ASCII.
- *Incluye heterogeneidad.* FTP oculta los detalles de los sistemas operativos individuales, y puede transferir una copia de un archivo entre cualquier par de computadoras.

Como pocos usuarios inician una aplicación FTP, el protocolo suele ser invisible. Sin embargo, es común que el navegador invoque automáticamente al servicio FTP cuando un usuario solicita la *descarga* de un archivo.

4.11 Paradigma de comunicación de FTP

Uno de los aspectos más interesantes de FTP se relaciona con la forma en que el cliente y el servidor interactúan. En general, el método parece simple: un cliente establece una conexión a un servidor FTP y envía una serie de solicitudes, a las que el servidor responde. A diferencia de HTTP, un servidor FTP no envía las repuestas a través de la misma conexión en la que el cliente envía las solicitudes. En su lugar, la conexión que crea el cliente, conocida como *conexión de control*, se reserva para los comandos. Cada vez que el servidor necesita descargar o enviar un archivo, éste (no el cliente) abre una nueva conexión. Para diferenciarlas de la conexión de control, las conexiones que se utilizan para transferir archivos se conocen como *conexiones de datos*.

Lo sorprendente es que FTP invierte la relación cliente-servidor para las conexiones de datos. Es decir, al abrir una conexión de datos, el cliente actúa como servidor (espera la conexión de datos) y el servidor actúa como cliente (inicia la conexión de datos). Una vez que termina la transferencia, la conexión de datos se cierra. Si el cliente envía otra solicitud, el servidor abre una nueva conexión de datos. La figura 4.10 ilustra la interacción.



Figura 4.10 La ilustración de las conexiones de FTP durante una sesión común.

La figura omite varios detalles importantes. Por ejemplo, después de crear la conexión de control, un cliente debe iniciar sesión en el servidor enviando un ID de inicio de sesión y una contraseña; para obtener archivos que son públicos, se utiliza un *inicio de sesión anónimo* que tiene la contraseña *guest*. En respuesta a cada solicitud, el servidor envía a través de la conexión de control un estado numérico, que incluye un inicio de sesión; la respuesta permite al cliente saber si la solicitud era válida.

Otro detalle interesante tiene que ver con los números de puerto de protocolo que se utilizan. En especial, surge la pregunta: ¿qué número de puerto de protocolo debe especificar un servidor al conectarse a un cliente? FTP permite al cliente decidir: antes de hacer una solicitud al servidor, un cliente asigna un puerto de protocolo en su sistema operativo local y envía el número de puerto al servidor. Es decir, el cliente se vincula al puerto para esperar una conexión y luego transmite el número de puerto a través de la conexión de control, como una cadena de dígitos decimales. El servidor lee el número y sigue los pasos especificados en el algoritmo 4.2.

Algoritmo 4.2

Dada:

Una conexión de control de FTP

Se logra:

Transferir un archivo a través de una conexión TCP

Método

El cliente envía una solicitud de un archivo específico a través de la conexión de control;

El cliente asigna un puerto de protocolo local, lo llama X y se vincula a él;

El cliente envía “PORT X” al servidor a través de la conexión de control;

El cliente espera para aceptar una conexión de datos en el puerto X;

El servidor recibe el comando PORT y extrae el número, X;

Asumiendo temporalmente el rol de un cliente, el servidor crea una conexión TCP a un puerto X en la computadora del cliente;

Asumiendo temporalmente el rol de un servidor, el cliente acepta la conexión TCP (conocida como “conexión de datos”);

El servidor envía el archivo solicitado a través de la conexión de datos;

El servidor cierra la conexión de datos;

Algoritmo 4.2 Los pasos que asumen un cliente y un servidor de FTP para transferir un archivo.

La transmisión de la información de puerto entre un par de aplicaciones puede parecer simple, pero no lo es, y la técnica no funciona bien en todos los casos. En particular, si uno de los dos puntos finales se encuentra detrás de un dispositivo de *traducción de dirección de red* (NAT), como un enrutador inalámbrico utilizado en una casa u oficina pequeña, la transmisión de un número de puerto de protocolo fallará. El capítulo 23 explica que FTP es una excepción; para soportar FTP, un dispositivo NAT reconoce una conexión de control de FTP, inspecciona el contenido de la conexión y reescribe los valores en un comando PORT.

4.12 Correo electrónico

Aunque los servicios como la mensajería instantánea se han vuelto populares, el correo electrónico sigue siendo uno de las aplicaciones de Internet más utilizados. Como se concibió antes de que estuvieran disponibles las computadoras personales y los dispositivos portátiles, el correo electrónico se diseñó para permitir que un usuario en una computadora enviara un mensaje directamente a un usuario en otra computadora. La figura 4.11 ilustra la arquitectura original y el algoritmo 4.3 enumera los pasos que se llevan a cabo.

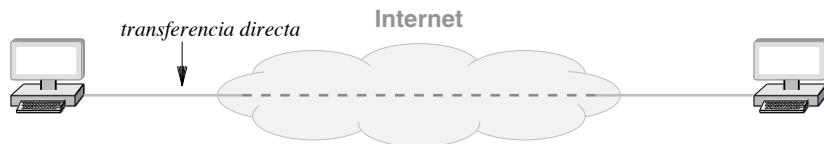


Figura 4.11 La configuración original del correo electrónico con transferencia directa de la computadora de un emisor a la computadora de un receptor.

Algoritmo 4.3

Dada:

La comunicación de correo electrónico de un usuario a otro

Se proporciona:

La transmisión de un mensaje al destinatario deseado

Método:

El usuario invoca a la aplicación de interfaz de correo y genera
un mensaje de correo electrónico para el usuario *x@destino.com*;

La interfaz de correo electrónico del usuario pasa el mensaje
a la aplicación de transferencia de correo;

La aplicación de transferencia de correo se convierte en un cliente
y abre una conexión TCP a *destino.com*;

La aplicación de transferencia de correo usa el protocolo SMTP
para transferir el mensaje y luego cierra la conexión;

El servidor de correo en *destino.com* recibe el mensaje y coloca
una copia en la bandeja de entrada del usuario *x*;

El usuario *x* en *destino.com* ejecuta la aplicación de interfaz de correo
para visualizar el mensaje;

Algoritmo 4.3 Pasos realizados para enviar un correo electrónico siguiendo el paradigma original.

Como indica el algoritmo 4.3, incluso el software original de correo electrónico se dividió en dos piezas conceptualmente separadas:

- Una aplicación de interfaz de correo electrónico
- Una aplicación de transferencia de correo

Un usuario invoca directamente a una *aplicación de interfaz de correo electrónico*. La interfaz proporciona mecanismos que permiten a un usuario redactar y editar mensajes salientes, así como leer y procesar el correo electrónico entrante. Una aplicación de interfaz de correo electrónico no actúa como cliente ni servidor y no transfiere mensajes a otros usuarios. En vez de ello, la aplicación de interfaz lee los mensajes del *buzón de correo* del usuario (un archivo en la computadora de éste) y pasa los mensajes salientes a una *aplicación de transferencia de correo*. La aplicación de transferencia de correo actúa como un cliente para enviar cada mensaje de correo electrónico a su destino. Además, la aplicación de transferencia de correo también actúa como servidor para aceptar los mensajes entrantes y deposita cada uno en el buzón de entrada del usuario apropiado.

Los estándares de los protocolos que se utilizan para el correo electrónico de Internet pueden dividirse en tres amplias categorías, como lo describe la figura 4.12.

Tipo	Descripción
Transferencia	Un protocolo que se utiliza para mover una copia de un mensaje de correo electrónico de una computadora a otra
Acceso	Un protocolo que permite a un usuario acceder a su buzón de correo y ver o enviar mensajes de correo electrónico
Representación	Un protocolo que especifica el formato de un mensaje de correo electrónico cuando se almacena en disco

Figura 4.12 Los tres tipos de protocolos que se utilizan con el correo electrónico.

4.13 El protocolo simple de transferencia de correo (SMTP)

El *protocolo simple de transferencia de correo* o *SMTP* es el protocolo estándar que un programa de transferencia de correo usa para transferir un mensaje de correo a través de Internet a un servidor. SMTP puede caracterizarse de la siguiente forma:

- Sigue un paradigma de flujo
- Usa mensajes de control basados en texto
- Sólo transfiere mensajes de texto
- Permite a un emisor especificar los nombres de los destinatarios y verificar cada nombre
- Envía una copia de un mensaje dado

El aspecto más inesperado de SMTP surge de su restricción a mensajes de texto. Una sección posterior explica el estándar MIME que permite al correo electrónico adjuntar otros objetos como imágenes o archivos binarios, pero el mecanismo SMTP se limita a texto.

El segundo aspecto de SMTP se enfoca en su habilidad de enviar un solo mensaje a varios destinatarios en una computadora. El protocolo permite a un cliente enlistar a cada uno de los usuarios y luego enviar una copia del mensaje a todos los usuarios de la lista. Por ejemplo, un cliente envía el mensaje “Tengo un mensaje de correo para el usuario A” y el servidor responde “OK” o “No existe dicho usuario”. De hecho, cada mensaje de un servidor SMTP comienza con un código numérico; por consiguiente, las respuestas son de la forma “250 OK” o “550 No existe dicho usuario”. La figura 4.13 proporciona una sesión SMTP de ejemplo que ocurre cuando se transfiere un mensaje de correo del usuario *John_Q_Smith* desde la computadora *ejemplo.edu* a dos usuarios en la computadora *algunaparte.com*.

```

Servidor: 220 algunaparte.com Simple Mail Transfer Service Ready
Cliente: HELO ejemplo.edu
Servidor: 250 OK

Cliente: MAIL FROM:<John_Q_Smith@ejemplo.edu>
Servidor: 250 OK

Cliente: RCPT TO:<Matthew_Doe@algunaparte.com>
Servidor: 550 No such user here

Cliente: RCPT TO:<Paul_Jones@algunaparte.com>
Servidor: 250 OK

Cliente: DATA
Servidor: 354 Start mail input; end with <CR><LF>.<CR><LF>
Cliente: ...envía el cuerpo del mensaje de correo, que puede contener
Cliente: ...muchas líneas de texto en cualquier forma
<CR><LF>.<CR><LF>
Servidor: 250 OK

Cliente: QUIT
Servidor: 221 algunaparte.com closing transmission channel

```

Figura 4.13 Una sesión SMTP de ejemplo.

En la figura, cada línea se etiqueta como *Cliente:* o *Servidor:* para indicar si el servidor o el cliente envían la línea pero el protocolo no incluye estas etiquetas en cursiva. El comando *HELO* permite al cliente autenticarse a sí mismo al enviar su nombre de dominio. Por último, la notación *<CR><LF>* denota los caracteres de retorno y de salto de línea. De esta forma, el cuerpo de un mensaje de correo electrónico se termina mediante una línea que consiste en un punto sin ningún otro texto o espacio.

El término *simple* dentro del nombre del protocolo implica que el SMTP está simplificado. Puesto que las versiones anteriores de SMTP eran extremadamente complejas, los diseñadores eliminaron las funciones innecesarias y se concentraron en los elementos básicos.

4.14 Proveedores de servicios de Internet (ISP), servidores de correo y acceso al correo

A medida que Internet se expandió para incluir a los consumidores, surgió un nuevo paradigma para el correo electrónico. Como la mayoría de los usuarios no dejan su computadora funcionando en forma continua y no saben cómo configurar y administrar un servidor de correo electrónico, los ISP comenzaron a ofrecer servicios de correo electrónico. En esencia, un ISP opera un servidor de correo electrónico y proporciona un buzón de correo para cada suscriptor. En vez del software de correo electrónico tradicional, cada ISP proporciona software de interfaz que permite a un usuario acceder a su buzón de correo. La figura 4.14 ilustra el arreglo.

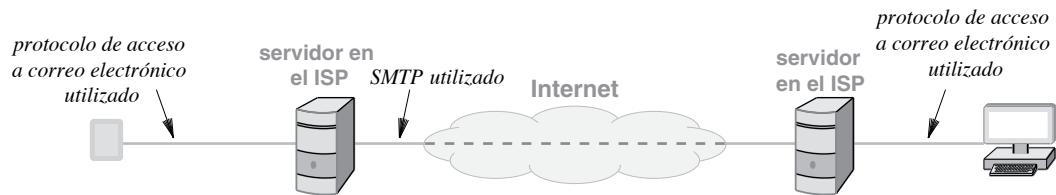


Figura 4.14 Una configuración de correo electrónico donde un ISP opera un servidor de correo electrónico y proporciona a un usuario el acceso a su buzón de correo.

El acceso al correo electrónico sigue una de estas dos formas:

- Una aplicación de interfaz de correo electrónico de propósito general
- Un navegador que accede a una página Web de correo electrónico

Las aplicaciones de interfaz específicas son muy comunes en dispositivos móviles, como tabletas o teléfonos inteligentes. Como conoce el tamaño de pantalla y la capacidad del dispositivo, la aplicación puede mostrar los mensajes de correo en un formato que se adapte al dispositivo. Otra ventaja de usar una aplicación de correo especial recae en la habilidad de descargar en el dispositivo todo un buzón de correo. La descarga es particularmente importante cuando el usuario móvil va a estar desconectado, ya que le permite administrar el correo electrónico aun cuando el dispositivo esté desconectado de Internet (por ejemplo, al viajar en un avión). Una vez que logra conectarse de nuevo a Internet, la aplicación se comunica con el servidor en el ISP del usuario para enviar el correo electrónico que éste creó y descargar los correos nuevos que puedan haber llegado al buzón mientras estuvo desconectado.

Usar un navegador como interfaz de correo electrónico es un proceso simple: un ISP proporciona una página Web especial que despliega los mensajes del buzón de correo de un usuario. De esta forma, el usuario inicia su navegador Web y accede al servicio de correo electrónico en el ISP. La página Web inicial invoca un mecanismo de autenticación que pide al usuario un ID de inicio de sesión y una contraseña; el servidor Web usa el inicio de sesión del usuario para seleccionar un buzón de correo. El servidor Web recupera los mensajes del buzón de correo, genera una página HTML que enumera los mensajes y devuelve esa página al navegador del usuario. La ventaja clave de usar una página Web para el correo electrónico surge de la habilidad de leer el correo electrónico desde cualquier computadora o dispositivo; un usuario no necesita un dispositivo específico, ni necesita ejecutar una aplicación de interfaz de correo especial. Por consiguiente, un usuario que viaja puede acceder a su correo electrónico desde una computadora que se encuentra en el centro de negocios del hotel.

4.15 Protocolos de acceso al correo (POP, IMAP)

Se han creado algunos protocolos que proporcionan *acceso* al correo electrónico. Un protocolo de acceso es distinto de un protocolo de transferencia, ya que el acceso sólo implica que un usuario interactúa con su buzón de correo, mientras que un protocolo de transferencia puede enviar correo desde un usuario cualquiera en una computadora a un buzón de correo en otra computadora. Los protocolos de acceso tienen las siguientes características:

- Proporcionan acceso al buzón de correo de un usuario
- Permiten a un usuario ver encabezados, descargar, eliminar o enviar mensajes individuales
- El cliente se ejecuta en la computadora personal o dispositivo del usuario
- El servidor se ejecuta en la computadora en donde se almacena el buzón de correo del usuario

La posibilidad de ver una lista de mensajes sin necesidad de descargar el contenido de éstos, es particularmente útil en casos donde el enlace entre un usuario y un servidor de correo es limitado. Por ejemplo, un usuario que navega en un teléfono celular puede analizar los encabezados y eliminar el correo basura o *spam* sin tener que descargar el contenido del mensaje.

Se han propuesto varios mecanismos para el acceso al correo electrónico. Algunos ISP ofrecen software de acceso al correo electrónico gratuito para sus suscriptores. Además, se crearon dos protocolos estándar de acceso al correo electrónico; la figura 4.15 enumera los estándares.

Acrónimo	Expansión
POP3	Protocolo de oficina de correos versión 3
IMAP	Protocolo de acceso a correo de Internet

Figura 4.15 Los dos protocolos estándar de acceso al correo electrónico.

Aunque ofrecen los mismos servicios básicos, los dos protocolos difieren en muchos detalles. En especial, cada uno proporciona su propio mecanismo de autenticación que el usuario sigue para identificarse. La autenticación es necesaria para asegurar que un usuario no acceda al buzón de correo de otro usuario.

4.16 Estándares de representación de correo electrónico (RFC2822, MIME)

Se estandarizaron dos representaciones de correo electrónico:

- Formato de mensaje de correo RFC2822
- Extensiones multipropósito de correo de Internet (MIME)

Formato de mensajes de correo RFC2822. El estándar de formato de mensajes de correo toma su nombre del documento de estándares del IETF *Request For Comments 2822* (Solicitud de comentarios 2822). El formato es simple: un mensaje de correo se representa como un archivo de texto y consta de una sección de *encabezado*, una línea en blanco y un *cuerpo*. Cada una de las líneas de encabezado tiene la siguiente forma:

Palabra clave: información

Hay un conjunto definido de palabras clave que incluyen *De:*, *Para:*, *Asunto:*, *Cc:* y así en lo sucesivo. Además, pueden agregarse líneas de encabezado que comienzan con X mayúscula sin afectar el procesamiento del correo. Por consiguiente, un mensaje de correo electrónico puede incluir una línea de encabezado aleatoria, tal como:

X-Peores-programas-TV: cualquier reality show

Extensiones multipropósito de correo de Internet (MIME). Recuerde que SMTP sólo puede transferir mensajes de texto. El estándar MIME extiende la funcionalidad del correo electrónico para permitir la transferencia de datos no textuales en un mensaje. MIME especifica la forma en que un archivo binario puede ser codificado en caracteres imprimibles, incluidos en un mensaje, y luego ser descodificado por el receptor.

Aunque MIME introdujo un estándar de codificación *Base64* que se ha vuelto popular, no restringe la codificación a un formato específico. En su lugar, MIME permite que un emisor y un receptor seleccionen una codificación que sea conveniente. Para especificar una codificación, el emisor incluye líneas adicionales en el encabezado del mensaje. Además, MIME permite a un emisor dividir un mensaje en varias partes y especificar una codificación independiente para cada una. De esta forma, con MIME un usuario puede enviar un mensaje de texto simple y adjuntar una imagen, una hoja de cálculo y un clip de audio, cada uno con su propia codificación. El sistema de correo electrónico receptor puede decidir cómo procesar los archivos adjuntos (por ejemplo, guardar una copia en disco o visualizarlos en pantalla).

De hecho, MIME agrega dos líneas a un encabezado de correo electrónico: una para declarar que se usó MIME para crear el mensaje y otra para especificar cómo se incluye la información de MIME dentro del cuerpo. Por ejemplo, las líneas de encabezado:

```
MIME-Version: 1.0
Content-Type: Multipart/Mixed; Boundary=Mime_separator
```

especifican que el mensaje se compuso usando la versión 1.0 de MIME y que una línea que contiene *Mime_separator* aparecerá en el cuerpo antes de cada parte del mensaje. Cuando se usa MIME para enviar un mensaje de texto estándar, la segunda línea se convierte en lo siguiente:

Content-Type: text/plain

MIME es incluso compatible con sistemas de correo electrónico anteriores que no comprenden su estándar o codificación. Desde luego que dichos sistemas no tienen forma de extraer archivos adjuntos del mensaje: tratan al cuerpo como un solo bloque de texto. En conclusión:

El estándar de MIME inserta líneas de encabezados adicionales para permitir que se envíen archivos adjuntos que no constan de texto dentro de un mensaje de correo electrónico. Un adjunto se codifica como letras que pueden imprimirse, y aparece una línea de separación antes de cada adjunto.

4.17 Sistema de nombres de dominio (DNS)

El sistema de nombres de dominio (DNS) proporciona un servicio que asigna nombres simbólicos (legibles para el ser humano) para las direcciones de las computadoras. Los navegadores, el software de correo y la mayoría de las demás aplicaciones usan el DNS. El sistema proporciona un ejemplo interesante de interacción entre cliente y servidor, debido a que la asignación no la realiza un solo servidor. En su lugar, la información de asignación de nombres se distribuye entre un gran conjunto de servidores ubicados en distintos sitios a lo largo de Internet. Cada vez que un programa de aplicación necesita traducir un nombre, la aplicación se vuelve cliente del sistema de asignación de nombres. El cliente envía un mensaje de solicitud a un servidor de nombres, el cual busca la dirección correspondiente y envía un mensaje de respuesta. Si no puede responder a una solicitud, un servidor de nombres se convierte temporalmente en el cliente de otro servidor de nombres hasta encontrar un servidor que pueda dar respuesta a la solicitud.

En sentido sintáctico, cada nombre consta de una secuencia de segmentos alfanuméricos separados por puntos. Por ejemplo, una computadora en la Purdue University tiene el siguiente nombre de dominio:

micorreo.purdue.edu

y una computadora en Google, Inc. tiene el siguiente nombre de dominio:

gmail.google.com

Los nombres de dominio son jerárquicos; la parte más importante del nombre está a la derecha. El segmento de la izquierda de un nombre (*micorreo* y *gmail*, en los ejemplos anteriores) representa el nombre de una computadora individual. Otros segmentos en un nombre de dominio identifican el grupo que posee el nombre. Por ejemplo, el segmento *purdue* proporciona el nombre de una universidad, mientras que *google* proporciona el nombre de una empresa. El DNS no restringe el número de segmentos en un nombre. En su lugar, cada organización puede elegir cuántos segmentos usar para las computadoras dentro de la organización y qué representan esos segmentos.

El sistema de nombres de dominio especifica valores para el segmento más importante, el cual se conoce como *dominio de nivel superior (TLD)*. Los dominios de nivel superior son controlados por la *Corporación de asignación de nombres y números de Internet (ICANN)*, que designa a uno o más *registradores de dominios* para administrar un dominio de nivel superior y aprobar nombres específicos. Algunos TLD son *genéricos*, lo que significa que están generalmente disponibles. Otros están restringidos a grupos o agencias gubernamentales específicas. La figura 4.16 muestra una lista de ejemplos de dominios DNS de nivel superior.

Nombre de dominio	Asignado a
aero	Industria del transporte aéreo
arpa	Dominio de infraestructura
asia	Para o sobre Asia
biz	Negocios
com	Organizaciones comerciales
coop	Asociaciones cooperativas
edu	Instituciones educativas
gov	Gobierno (de Estados Unidos)
info	Información
int	Organizaciones de tratados internacionales
jobs	Administradores de recursos humanos
mil	Ejército de Estados Unidos
mobi	Proveedores de contenido móvil
museum	Museos
name	Individuos
net	Centros importantes de soporte de redes
org	Organizaciones no comerciales
pro	Profesionales con credenciales
travel	Viaje y turismo
código de país	Una nación soberana

Figura 4.16 Ejemplos de dominios DNS de nivel superior y el grupo al que se asigna cada uno.

Cualquier organización que desee utilizar un nombre de dominio de nivel superior, debe solicitarlo. Por ejemplo, la mayoría de las corporaciones optan por registrarse bajo el dominio *com*. Por consiguiente, una corporación llamada *Empresa* podría solicitar que se le asigne el dominio *empresa* bajo el dominio de nivel superior *com*. Una vez que se aprueba la solicitud, a la corporación *Empresa* se le asigna el dominio:

empresa.com

Una vez que se asigna el nombre, otra organización también llamada *Empresa* podría solicitar *empresa.biz* o *empresa.org*, pero no *empresa.com*. Además, una vez que se asigna *empresa.com*, la corporación

Una empresa puede elegir cuántos niveles adicionales agregar y el significado de cada uno. De tal forma, si la empresa tiene ubicaciones en las zonas este y oeste de un país, podríamos encontrar nombres tales como:

computadora1.oeste.empresia.com

O podría elegir una jerarquía de nombres relativamente plana para todas las computadoras identificadas por el nombre y el nombre de dominio de la compañía:

computadora1.empresia.com

Además de la estructura organizacional familiar, el DNS permite que las organizaciones usen un registro geográfico. Por ejemplo, la Corporación de iniciativas para la investigación nacional registró el dominio:

cnri.reston.va.us

Esto es debido a que la corporación se encuentra en la ciudad de Reston, Virginia en Estados Unidos. En este caso, los nombres de las computadoras de esta organización, terminan en *.us* en vez de *.com*.

Otro país adoptaron una combinación de nombres de dominio geográficos y organizacionales. Por ejemplo, las universidades en el Reino Unido se registran bajo el dominio:

ac.uk

Donde *ac* es una abreviación de *académico* y *uk* es el código de país oficial del Reino Unido.

4.18 Nombres de dominio que comienzan con un nombre de servicio

Muchas organizaciones asignan nombres de dominio que reflejan el servicio que proporciona una computadora. Por ejemplo, una computadora que ejecuta un servidor para el protocolo de transferencia de archivos podría llamarse:

ftp.empresia.com

De manera similar, una computadora que ejecuta un servidor Web podría llamarse:

www.empresia.com

Dichos nombres son fáciles de recordar, aunque no son obligatorios. En especial, el uso de *www* para nombrar computadoras que ejecutan un servidor Web es simplemente una convención; una computadora cualquiera puede ejecutar un servidor Web, incluso aunque el nombre de dominio de la computadora no contenga *www*. De esta forma, una computadora que tenga un nombre de dominio que comience con *www* no tiene la obligación de ejecutar un servidor Web. En conclusión:

Usar la primera etiqueta en un nombre de dominio para denotar un servicio (por ejemplo, www) es simplemente una convención para ayudar al ser humano a identificar el tipo de servicio.

4.19 La jerarquía del DNS y el modelo servidor

Una de las principales características del sistema de nombres de dominio es la autonomía: el sistema está diseñado para permitir a cada organización asignar nombres a las computadoras o modificar esos nombres sin necesidad de informarlo a una autoridad central. Para lograr la autonomía, cada organización tiene permitido operar servidores DNS para administrar su parte de la jerarquía. Así, la Purdue University opera un servidor de nombres que termine en *purdue.edu*, e IBM Corporation opera un servidor de nombres que termina en *ibm.com*. Cada servidor DNS contiene información que vincula al servidor con otros servidores de nombres de dominio en niveles superiores e inferiores de la jerarquía. Además, un servidor dado puede *replicarse*, de modo que existan varias copias físicas del mismo. La replicación es especialmente útil para servidores que se utilizan mucho, como los *servidores raíz* que proporcionan información sobre los dominios de nivel superior, ya que un solo servidor no podría manejar la carga. En tales casos, los administradores deben garantizar que todas las copias se coordinen para que proporcionen exactamente la misma información.

Cada organización es libre de elegir los detalles de sus servidores. Una pequeña organización que sólo cuenta con algunas computadoras, puede contratar un ISP para que opere un servidor DNS para ella. Una organización grande que opera su propio servidor puede optar por colocar todos los nombres de la organización en un solo servidor físico, o puede optar por dividir sus nombres entre varios servidores. La división puede coincidir con la estructura organizacional (por ejemplo, los nombres de un subsidiario pueden estar en un servidor independiente) o una estructura geográfica (por ejemplo, un servidor independiente para cada sede de la compañía). La figura 4.17 ilustra cómo la corporación hipotética Empresa podría elegir estructurar los servidores si ésta tuviera una división de dulces y una división de jabones.

4.20 Resolución de nombres

El proceso de traducir un nombre de dominio en una dirección se conoce como *resolución de nombres*, y se dice que se *resuelve* una dirección. El software para realizar la traducción se conoce como *solucionador de nombres* (o simplemente *solucionador*). En la API de sockets, por ejemplo, el solucionador se invoca llamando a la función *gethostbyname*. El solucionador se convierte en un cliente, hace contacto con un servidor DNS y devuelve una respuesta a quien hizo la llamada.

Cada solucionador se configura con la dirección de uno o más servidores de nombres de dominio *locales*.[†] El solucionador forma un mensaje de *solicitud del DNS*, envía el mensaje al servidor local y espera a que éste envíe un mensaje de *respuesta del DNS*. Un solucionador puede optar por usar el paradigma de flujo o de mensajes al comunicarse con un servidor DNS; la mayoría de los solucionadores se configuran para usar un paradigma de mensajes, debido a que representa una menor sobrecarga para una solicitud pequeña.

Como ejemplo de una resolución de nombres, considere la jerarquía de servidores que ilustra la figura 4.17(a) y suponga que una computadora en la división de jabones genera una solicitud para el nombre *chocolate.dulces.empresa.com*. El solucionador estará configurado para enviar la solicitud al servidor DNS local (es decir, el servidor para *empresa.com*). Aunque no puede responder a la solicitud, el servidor sabe cómo contactar al servidor de *dulces.empresa.com*, que puede generar una respuesta.

[†] Cuando hablamos sobre el uso de la caché tendrá un mayor significado el hecho de contactar primero a un servidor local.

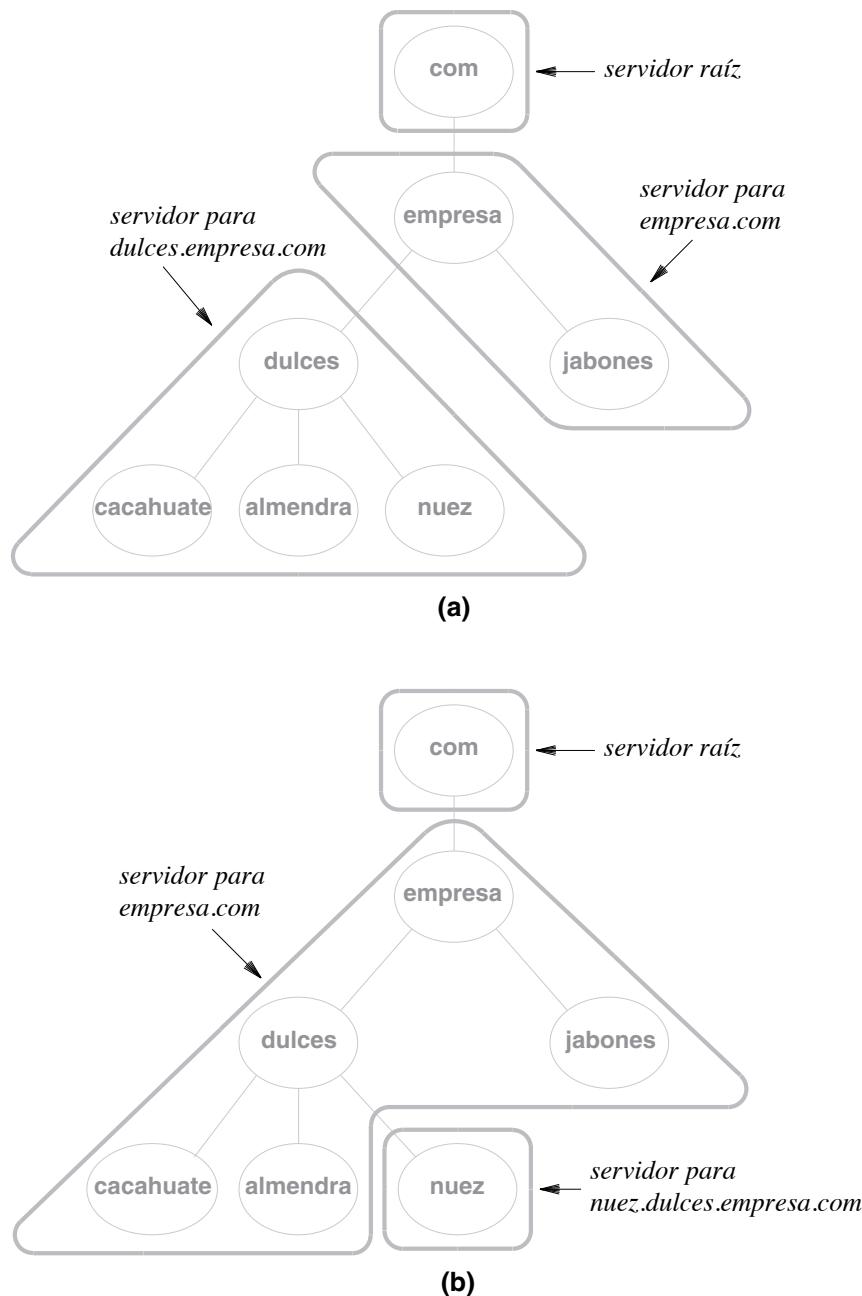


Figura 4.17 Una jerarquía hipotética de DNS y dos posibles asignaciones de nombres para los servidores.

4.21 Uso de la memoria caché en servidores del DNS

El principio de *localidad de referencia*, que forma la base para el uso de la caché, se aplica al sistema de nombres de dominio en dos formas:

- Espacial: un usuario tiende a buscar los nombres de computadoras locales con más frecuencia que los nombres de computadoras remotas
- Temporal: un usuario tiende a buscar el mismo conjunto de nombres de dominio en forma recurrente

Ya vimos cómo el DNS explota la localidad espacial: un solucionador de nombres contacta primero a un servidor local. Para explotar la localidad temporal, un servidor DNS coloca todas las búsquedas en la memoria caché. El algoritmo 4.4 sintetiza el proceso.

Algoritmo 4.4

Dado:

Un mensaje de solicitud de un solucionador de nombres DNS

Se proporciona:

Un mensaje de respuesta que contiene la dirección

Método:

Extraer el nombre, N , de la solicitud;

if (el servidor es autoridad para N) {

 Formar y enviar una respuesta al solicitante;

else if (la respuesta para N está en la caché) {

 Formar y enviar una respuesta al solicitante;

else { /* Necesita buscar una respuesta */

 if (se conoce el servidor de autoridad para N) {

 Enviar solicitud al servidor de autoridad;

 } else {

 Enviar solicitud al servidor raíz;

 }

 Recibir respuesta y colocarla en la caché;

 Formar y enviar una respuesta al solicitante;

}

Algoritmo 4.4 Pasos que realiza un servidor DNS para resolver un nombre.

De acuerdo con el algoritmo, cuando llega una solicitud de un nombre fuera del conjunto en el que el servidor es una autoridad, se produce una interacción adicional entre cliente y servidor. El servidor se convierte temporalmente en cliente de otro servidor. Cuando el otro servidor devuelve una respuesta, el servidor original coloca la respuesta en caché y regresa una copia de la respuesta al solucionador del cual llegó la solicitud. Por consiguiente, además de conocer la dirección de todos los servidores inferiores de la jerarquía, cada servidor DNS debe conocer la dirección de un servidor raíz.

La cuestión fundamental en el uso de la memoria caché tiene que ver con el tiempo que deben colocarse los elementos en la misma: si un elemento se coloca demasiado tiempo, el elemento se volverá *obsoleto*. El DNS resuelve el problema organizando que un servidor de autoridad especifique un tiempo de vigencia en caché para cada elemento. De esta forma, cuando un servidor local busca un nombre, la respuesta consiste en un *registro de recurso* que especifica la vigencia en caché junto con una respuesta. Cada vez que un servidor coloca una respuesta en caché, el servidor cumple con el tiempo especificado en el registro de recurso. En conclusión:

Como cada registro de recurso de DNS generado por un servidor de autoridad especifica un tiempo de vigencia en caché, un servidor DNS nunca devuelve una respuesta obsoleta.

El uso de caché para DNS no se limita a los servidores: un solucionador también puede colocar elementos en la caché. De hecho, el software solucionador en la mayoría de los sistemas de computadoras almacena en caché las respuestas de las búsquedas del DNS, lo que significa que solicitudes sucesivas del mismo nombre no necesitan usar la red, ya que el solucionador puede satisfacer la solicitud desde la caché del disco local de la computadora.

4.22 Tipos de entradas del DNS

Cada entrada en una base de datos del DNS consta de tres elementos: un nombre de dominio, un *tipo* de registro y un valor. El tipo de registro especifica cómo se va a interpretar el valor (por ejemplo, que el valor sea una dirección IPv4). Lo que es más importante, una consulta enviada a un servidor DNS especifica tanto un nombre de dominio como un tipo; el servidor sólo devuelve aquello que coincide con el tipo de la consulta.

Cuando una aplicación necesita una dirección IP, el navegador especifica el tipo A (IPv4) o el tipo AAAA (IPv6). Un programa de correo electrónico que usa SMTP para buscar un nombre de dominio especifica el tipo MX, al cual solicita un *intercambiador de correos* (*Mail eXchanger* en inglés). La respuesta que devuelve un servidor coincide con el tipo solicitado. De esta forma, un sistema de correo electrónico recibirá una respuesta que coincide con el tipo MX, mientras que el navegador recibirá una respuesta que coincide con el tipo A o AAAA. En conclusión:

Cada entrada en un servidor DNS tiene un tipo. Cuando un solucionador busca un nombre, tiene que especificar el tipo deseado, ya que el servidor DNS sólo devuelve entradas que coinciden con el tipo especificado.

El sistema de tipos de DNS puede producir resultados inesperados, ya que la dirección devuelta puede depender del tipo. Por ejemplo, una corporación puede optar por usar el nombre *corporacion.com* para servicios tanto de Web como de correo electrónico. Con el DNS, es posible que la corporación divida la carga de trabajo entre computadoras independientes, asignando las búsquedas de tipo A a una computadora y las búsquedas de tipo MX a otra. La desventaja de un esquema así es que parece contradictoria para el ser humano, ya que sería posible enviar un correo electrónico a *corporacion.com* aun cuando no sea posible acceder al servidor Web o enviar una prueba *ping* a la computadora.

4.23 Alias y registros de recursos CNAME

El DNS ofrece un tipo *CNAME* que es similar a un enlace simbólico en un sistema de archivos: la entrada proporciona un *alias* para otra entrada del DNS. Para comprender la utilidad de los alias, suponga que la corporación Empresa tiene dos computadoras llamadas *carlos.empresa.com* y *lucy.empresa.com*. Suponga además que Empresa decide ejecutar un servidor Web en la computadora *lucy* y desea seguir la convención de usar el nombre *www* para la computadora que ejecuta el servidor Web de la organización. Aunque la organización podría optar por cambiar el nombre de la computadora *lucy*, existe una solución mucho más sencilla: la organización puede crear una entrada *CNAME* para *www.empresa.com* que apunte hacia *lucy*. Cada vez que un solucionador envía una solicitud de *www.empresa.com*, el servidor devuelve la dirección de la computadora *lucy*.

El uso de alias es muy conveniente debido a que permite a una organización cambiar la computadora utilizada para un servicio específico sin alterar los nombres ni las direcciones de las computadoras. Por ejemplo, la corporación Empresa puede trasladar su servicio Web de la computadora *lucy* a la computadora *carlos*, moviendo el servidor y cambiando el registro *CNAME* en el servidor DNS: las dos computadoras conservan sus nombres y direcciones IP originales. El uso de alias también permite a una organización asociar múltiples alias con una sola computadora. De esta forma, la corporación Empresa puede ejecutar un servidor FTP y un servidor Web en la misma computadora, y crear los registros CNAME:

www.empresa.com

ftp.empresa.com

4.24 Abreviaciones y el DNS

El DNS no incorpora abreviaciones: un servidor sólo responde a un nombre completo. Sin embargo, la mayoría de los solucionadores pueden configurarse con un conjunto de sufijos que permiten a un usuario abreviar nombres. Por ejemplo, cada solucionador en la corporación Empresa podría programarse para buscar un nombre dos veces: uno sin cambio y otro agregando el sufijo *empresa.com*. Si un usuario escribe un nombre de dominio completo, el servidor local devolverá la dirección y continuará el procesamiento. Por el contrario, si un usuario escribe un nombre abreviado, el solucionador intentará

primero resolver el nombre y recibirá un error debido a que no existe dicho nombre. A continuación el solucionador intentará adjuntar un sufijo y buscar el nombre resultante. Puesto que un solucionador se ejecuta en la computadora personal de un usuario, el método permite a cada usuario elegir el orden en el que se prueban los sufijos.

Desde luego que permitir que cada usuario configure su solucionador para manejar abreviaciones tiene una desventaja: el nombre que escriba un usuario puede diferir del que escriba otro. Por consiguiente, si los usuarios se comunican nombres entre sí (por ejemplo, enviando un nombre de dominio en un mensaje de correo electrónico), cada uno debe tener cuidado de especificar nombres completos y no abreviaciones.

4.25 Nombres de dominio internacionalizados

Puesto que usa el conjunto de caracteres ASCII, el DNS no puede almacenar nombres en alfabetos que no estén representados en ASCII. En especial, los idiomas como ruso, griego, chino y japonés contienen caracteres para los que no existe una representación en ASCII. Muchos idiomas, como el español y otros idiomas europeos, usan acentos y otras marcas que no pueden representarse en ASCII.

Durante años, la IETF debatió las modificaciones y extensiones del DNS para adaptar los nombres de dominios internacionales. Después de considerar muchas propuestas, la IETF eligió un método conocido como *internacionalización de nombres de dominios en aplicaciones (IDNA)*. En vez de modificar el DNS, IDNA usa ASCII para almacenar todos los nombres. Es decir, cuando recibe un nombre de dominio que contiene caracteres que no pertenecen a ASCII, IDNA traduce el nombre en una secuencia de caracteres ASCII y almacena el resultado en el DNS. Cuando un usuario busca el nombre, se aplica la misma traducción para convertir el nombre en una cadena ASCII y la cadena ASCII resultante se coloca en una consulta de DNS. En esencia, IDNA depende de las aplicaciones para traducir el conjunto de caracteres internacionales que ve un usuario en la forma ASCII interna que se utiliza en el DNS.

Las reglas para traducir nombres de dominio internacionales son complejas y usan *Unicode*.[†] En esencia, la traducción se aplica a cada etiqueta en el nombre de dominio y produce etiquetas de la forma:

$$xn--\alpha-\beta$$

Donde *xn--* es una cadena reservada de cuatro caracteres para indicar que la etiqueta es un nombre internacional, α es el subconjunto de caracteres de la etiqueta original que puede representarse en ASCII y β es una cadena de caracteres ASCII adicionales para indicar a una aplicación de IDNA cómo insertar caracteres que no sean de ASCII en α y formar la versión imprimible de la etiqueta.

Las versiones más recientes de los navegadores más utilizados, como Firefox e Internet Explorer, pueden aceptar y visualizar nombres de dominio que incluyan caracteres de ASCII debido a que implementan el método IDNA. Si una aplicación no implementa IDNA, la salida puede parecer extraña al usuario. Es decir, cuando una aplicación que no implementa IDNA visualiza un nombre de dominio internacional, el usuario verá la forma interna que se muestra anteriormente y que incluye la cadena inicial *xn--* y las partes α y β que le siguen.

[†] El algoritmo de traducción utilizado para codificar etiquetas que no sean de ASCII se conoce como algoritmo *Punycode*, y la cadena resultante se conoce como *Punycode*.

En conclusión:

El estándar IDNA para nombres de dominio internacionales codifica cada etiqueta como una cadena ASCII, y depende de aplicaciones para traducir entre el conjunto de caracteres que espera un usuario y la versión codificada almacenada en el DNS.

4.26 Representaciones extensibles (XML)

Todos los protocolos de aplicaciones tradicionales cubiertos en este capítulo emplean una representación fija. Es decir, el protocolo de aplicación especifica un conjunto exacto de mensajes que un cliente y un servidor pueden intercambiar, así como la forma exacta de los datos que acompañan al mensaje. La desventaja principal de un método fijo surge de la dificultad implicada en la realización de las modificaciones. Por ejemplo, como los estándares de correo electrónico restringen el contenido de un mensaje a texto, se necesitaba una modificación importante para agregar extensiones MIME.

La alternativa a una representación fija es un sistema extensible que permita a un usuario especificar el formato de los datos. Hay un estándar de representación extensible que se ha vuelto ampliamente aceptado: el *lenguaje de marcación extensible* o XML. XML se parece a HTML ya que ambos lenguajes incrustan etiquetas en un documento de texto. A diferencia de HTML, las etiquetas de XML no se especifican por anticipado y no corresponden a los comandos de formato. En su lugar, XML describe la estructura de los datos y proporciona nombres para cada campo. Las etiquetas en XML están bien equilibradas, ya que cada vez que se utiliza una etiqueta de apertura <X> debe ir acompañada de una etiqueta de cierre </X>. Además, puesto que XML no asigna ningún significado a las etiquetas, es posible crear nombres de etiquetas según sea necesario. En especial, pueden elegirse nombres de etiquetas para facilitar el análisis o acceso de los datos. Por ejemplo, si dos compañías aceptan intercambiar directorios telefónicos corporativos, pueden definir un formato de XML que tenga elementos de datos como: nombre de un empleado, número telefónico y oficina. Las compañías pueden optar por dividir aún más un nombre en apellido y primer nombre. La figura 4.18 contiene un ejemplo.

```
<DIRECCION>
  <NOMBRE>
    <PRIMEROMBRE> Juan </PRIMEROMBRE>
    <APELIDO> Pérez </APELIDO>
  </NOMBRE>
  <OFICINA> Oficina 320 </OFICINA>
  <TELEFONO> 765-555-1234 </TELEFONO>
</DIRECCION>
```

Figura 4.18 Un ejemplo de XML para una libreta telefónica corporativa.

4.27 Resumen

Los protocolos de la capa de aplicación, necesarios para los servicios estandarizados, definen los aspectos de representación y transferencia de datos de la comunicación. Los protocolos de representación que se utilizan con World Wide Web incluyen el *lenguaje de marcación de hipertexto* o *HTML* y el estándar URL. El protocolo de transferencia Web, conocido como *protocolo de transferencia de hipertexto* (HTTP), determina la forma en que un navegador se comunica con un servidor Web para descargar o enviar contenido. Para agilizar las descargas, un navegador coloca el contenido de la página en la memoria caché y usa el comando *HEAD* de HTTP para solicitar información de estado sobre la página. Si la versión en caché sigue vigente, el navegador usa esta versión; de lo contrario, el navegador emite una solicitud *GET* para descargar una copia actualizada.

HTTP usa mensajes de texto. Cada respuesta de un servidor comienza con un encabezado que la describe. Las líneas del encabezado comienzan con un valor numérico, representado como dígitos ASCII, que indica el estado (por ejemplo, si una solicitud produce o no un error). Los datos que siguen al encabezado pueden contener valores binarios arbitrarios.

El *protocolo de transferencia de archivos* (FTP) permite la descarga de archivos grandes. FTP requiere que un cliente inicie sesión en el sistema del servidor; FTP soporta un inicio de sesión anónimo (*anonymous*) y una contraseña genérica de invitado (*guest*) para dar acceso a los archivos públicos. El aspecto más interesante de FTP surge debido a su uso inusual de las conexiones. Un cliente establece una conexión de control que se utiliza para enviar una serie de comandos. Cada vez que un servidor necesita enviar datos (por ejemplo, la descarga de un archivo o el listado de un directorio), actúa como un cliente y el cliente actúa como un servidor. Es decir, el servidor inicia una nueva conexión de datos hacia el cliente. Una vez que se envía un archivo, se cierra la conexión de datos.

Se utilizan tres tipos de protocolos de capa de aplicación con el correo electrónico: el de transferencia, el de representación y el de acceso. El *protocolo simple de transferencia de correo* (SMTP) sirve como el estándar de transferencia clave; SMTP sólo puede transferir un mensaje basado en texto. Hay dos estándares de representación para el correo electrónico: RFC 2822 define el formato del mensaje de correo como un encabezado y un cuerpo, separados por una línea en blanco. El estándar *extensiones multipropósito de correo de Internet* (MIME) define un mecanismo para enviar archivos binarios adjuntos en un mensaje de correo electrónico. MIME inserta líneas de encabezado adicionales que indican al receptor cómo interpretar el mensaje. MIME requiere que un emisor codifique un archivo como texto imprimible.

Los protocolos de acceso al correo electrónico, como POP3 e IMAP, permiten que un usuario acceda a un buzón de correo. Este tipo de acceso se ha vuelto popular, ya que un suscriptor puede permitir que un ISP opere un servidor de correo electrónico y mantenga el buzón de correo del usuario.

El *sistema de nombres de dominio* (DNS) ofrece la conversión automatizada de los nombres legibles para el ser humano en direcciones de computadora. El DNS consta de muchos servidores, cada uno de los cuales controla una parte del espacio de los nombres. Los servidores se organizan en una jerarquía y un servidor conoce las ubicaciones de los otros servidores dentro de la jerarquía.

El DNS usa la memoria caché para mantener la eficiencia: cuando un servidor de autoridad proporciona una respuesta, cada servidor que transfiere la respuesta también coloca una copia en su caché. Para evitar que las copias en caché se vuelvan obsoletas, la autoridad de un nombre especifica cuánto tiempo puede colocarse en la caché.

EJERCICIOS

- 4.1** ¿Qué detalles especifica un protocolo de aplicación?
- 4.2** ¿Por qué un protocolo de un servicio estandarizado se documenta de manera independiente a una implementación?
- 4.3** ¿Cuáles son los dos aspectos clave de los protocolos de aplicación y qué incluye cada uno?
- 4.4** Proporcione ejemplos de protocolos Web que ilustren cada uno de los dos aspectos de un protocolo de aplicación.
- 4.5** Sintetice las características de HTML.
- 4.6** ¿Cuáles son las cuatro partes de un URL y qué signos se usan para separarlas?
- 4.7** ¿Cuáles son los cuatro tipos de solicitudes de HTPP y cuándo se utiliza cada una?
- 4.8** ¿Sabe un navegador si una solicitud de HTTP es sintácticamente incorrecta o si el elemento referenciado no existe? Explique por qué.
- 4.9** ¿Qué objetos de datos coloca un navegador en caché y por qué se usa la caché?
- 4.10** Describa los pasos que realiza un navegador para determinar si debe o no usar un elemento de su caché.
- 4.11** ¿Puede un navegador usar otros protocolos de transferencia además de HTTP? Explique.
- 4.12** Cuando un usuario solicita un listado de directorios de FTP, ¿cuántas conexiones TCP se forman? Explique.
- 4.13** Verdadero o falso: cuando un usuario ejecuta una aplicación FTP, la aplicación actúa como cliente y como servidor. Explique su respuesta.
- 4.14** ¿Cómo sabe un servidor FTP qué número de puerto usar para una conexión de datos?
- 4.15** De acuerdo con el paradigma de correo electrónico original, ¿podría un usuario recibir correo electrónico si la computadora del usuario no ejecutara un servidor de correo? Explique.
- 4.16** Enliste los tres tipos de protocolos utilizados con el correo electrónico y describa cada uno de ellos.
- 4.17** ¿Cuáles son las características del SMTP?
- 4.18** ¿Puede SMTP transferir por sí solo un mensaje de correo electrónico que contenga un punto en una línea? ¿Por qué sí o por qué no?
- 4.19** ¿Dónde se usa un protocolo de acceso a correo electrónico?
- 4.20** ¿Cuáles son los dos principales protocolos de acceso a correo electrónico?
- 4.21** ¿Por qué se inventó MIME?
- 4.22** ¿Cuál es el objetivo principal del sistema de nombres de dominio?
- 4.23** Suponiendo que ISO asignó N códigos de países, ¿cuántos dominios de nivel superior existen?
- 4.24** Verdadero o falso: un servidor Web debe tener un nombre de dominio que comience con www. Explique.
- 4.25** Verdadero o falso: una compañía multinacional puede optar por dividir su jerarquía de nombres en tal forma que tenga un servidor de nombres de dominio en Europa, uno en Asia y uno en Norteamérica. Explique.
- 4.26** ¿Cuándo envía un servidor de nombres de dominio una solicitud a un servidor de autoridad y cuándo responde a la solicitud sin enviarla a éste?

- 4.27** Verdadero o falso: si una empresa mueve su servidor Web de la computadora x a la computadora y , es necesario cambiar los nombres de las dos computadoras. Explique.
- 4.28** Verdadero o falso: un servidor DNS puede devolver una dirección IP diferente para un nombre dado, dependiendo de si la búsqueda especifica el servicio de correo electrónico o Web. Explique.
- 4.29** ¿El estándar IDNA requiere cambiar los servidores DNS? ¿Y los clientes DNS? Explique.
- 4.30** Haga una búsqueda en Web para averiguar sobre la búsqueda iterativa de DNS. ¿Bajo qué circunstancias se utiliza la búsqueda iterativa?
- 4.31** ¿Cómo permite XML que una aplicación especifique campos, como el nombre y la dirección?

PARTE II

Comunicaciones de datos

**Una introducción a los medios,
la codificación, la transmisión,
la modulación, la multiplexación,
las conexiones y el acceso remoto**

Capítulos

- 5 Generalidades de las comunicaciones de datos**
- 6 Fuentes de información y señales**
- 7 Medios de transmisión**
- 8 Confiabilidad y codificación de canales**
- 9 Modos de transmisión**
- 10 Modulación y módems**
- 11 Multiplexación y demultiplexación**
- 12 Acceso y tecnologías de interconexión**

Contenido del capítulo

- 5.1 Introducción, 85
- 5.2 La esencia de las comunicaciones de datos, 86
- 5.3 Motivación y alcance del tema, 87
- 5.4 Las partes conceptuales de un sistema de comunicaciones, 87
- 5.5 Los subtemas de las comunicaciones de datos, 90
- 5.6 Resumen, 91

5

Generalidades de las comunicaciones de datos

5.1 Introducción

La primera parte del libro habla sobre la programación de redes y analiza las aplicaciones de Internet. El capítulo sobre programación de sockets explica la API que los sistemas operativos ofrecen al software de aplicación y muestra que un programador puede crear aplicaciones que usen Internet sin necesidad de comprender los mecanismos inherentes. En el resto del libro aprenderemos sobre los complejos protocolos y tecnologías que soportan la comunicación, y veremos que entender esta complejidad puede ayudar a los programadores a escribir mejor un código.

Esta parte del libro explora la transmisión de información a través de medios físicos, como cables, fibras ópticas y ondas de radio. Veremos que, aunque los detalles varían, las ideas básicas sobre información y comunicación se aplican a todas las formas de transmisión. Comprenderemos que las comunicaciones de datos ofrecen herramientas conceptuales y de análisis que brindan una explicación general sobre cómo operan los sistemas de comunicaciones. Lo que es más importante, las comunicaciones de datos nos dicen qué transferencias son teóricamente posibles y cómo en la realidad el mundo físico limita los sistemas de transmisión.

Este capítulo ofrece una descripción general de las comunicaciones de datos y explica cómo es que las partes conceptuales forman un sistema de comunicaciones completo. Los capítulos siguientes explican cada concepto con detalle.

5.2 La esencia de las comunicaciones de datos

¿Qué conllevan las comunicaciones de datos? Como se ilustra en la figura 5.1, el tema implica una combinación de ideas y metodologías provenientes de tres disciplinas.

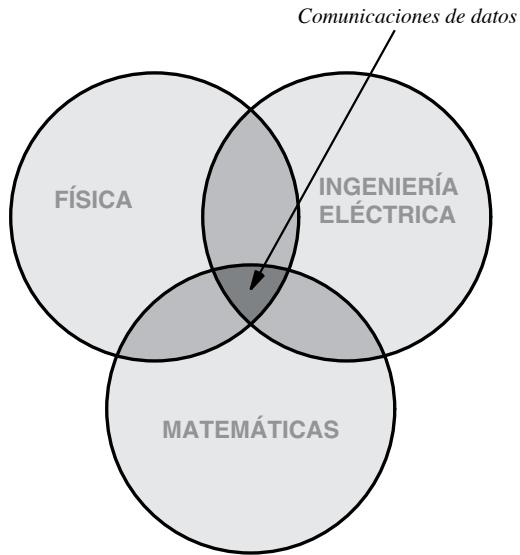


Figura 5.1 El tema de las comunicaciones de datos recae en la intersección entre la física, las matemáticas y la ingeniería eléctrica.

Puesto que implican la transmisión de información sobre medios físicos, las comunicaciones de datos abarcan algo de física. El tema se basa en conceptos sobre corriente eléctrica, luz, ondas de radio y otras formas de radiación electromagnética. Puesto que la información está digitalizada y se transmiten datos digitales, las comunicaciones de datos usan matemáticas e incluyen teorías matemáticas y varias formas de análisis matemático. Por último, como el objetivo máximo es desarrollar formas prácticas de diseñar y construir sistemas de transmisión, las comunicaciones de datos se enfocan en desarrollar técnicas que puedan ser usadas por ingenieros eléctricos.

En conclusión:

Aunque incluyen conceptos de física y matemáticas, las comunicaciones de datos no simplemente ofrecen teorías abstractas, sino que proporcionan una base que se utiliza para construir sistemas prácticos de comunicaciones.

5.3 Motivación y alcance del tema

Hay tres ideas principales que proporcionan gran parte de la motivación para las comunicaciones de datos y ayudan a definir su alcance.

- Las fuentes de información pueden ser de cualquier tipo
- La transmisión usa un sistema físico
- Varias fuentes de información pueden compartir el mismo medio

Considerando la popularidad de las aplicaciones multimedia, el primer punto es especialmente relevante: la información no se limita a los bits almacenados en una computadora. En su lugar, la información puede provenir del mundo físico, como el sonido que proviene de un micrófono y el video que se captura con una cámara. Por consiguiente, es importante comprender las posibles fuentes y formas de información, además de las maneras en que una forma puede transformarse en otra.

El segundo punto sugiere que debemos usar fenómenos naturales, como la electricidad y la radiación electromagnética, para transmitir información. Por lo tanto, es importante comprender los tipos de medios disponibles, además de las propiedades de cada uno. Debemos entender también cómo pueden usarse los fenómenos físicos para transmitir información a través de cada medio, además de la relación entre las comunicaciones de datos y la consecuente transmisión. Por último, debemos comprender los límites de los sistemas físicos, los problemas que pueden surgir durante la transmisión y las técnicas que pueden usarse para detectar o resolver los problemas.

El tercer punto sugiere que es fundamental compartir. Sin duda, veremos que la compartición desempeña un papel fundamental en las redes de computadoras. Es decir, por lo general una red de computadoras permite que varias entidades de comunicación se comuniquen a través de un medio físico dado. Por lo tanto, es importante comprender las formas en que pueden compartirse los servicios, las ventajas y desventajas de cada una, así como las formas de comunicación resultantes.

5.4 Las partes conceptuales de un sistema de comunicaciones

Para comprender las comunicaciones de datos, imagine un sistema de comunicaciones funcional que cuenta con varias fuentes de información y permite enviar cada fuente a un destino independiente. Puede parecer que la comunicación en dicho sistema es simple. Cada fuente necesita un mecanismo para recopilar la información, prepararla para la transmisión y transmitirla a través del medio físico compartido. De manera similar, se necesita un mecanismo que extraiga y entregue la información en cada destino. La figura 5.2 ilustra esta vista simplista.

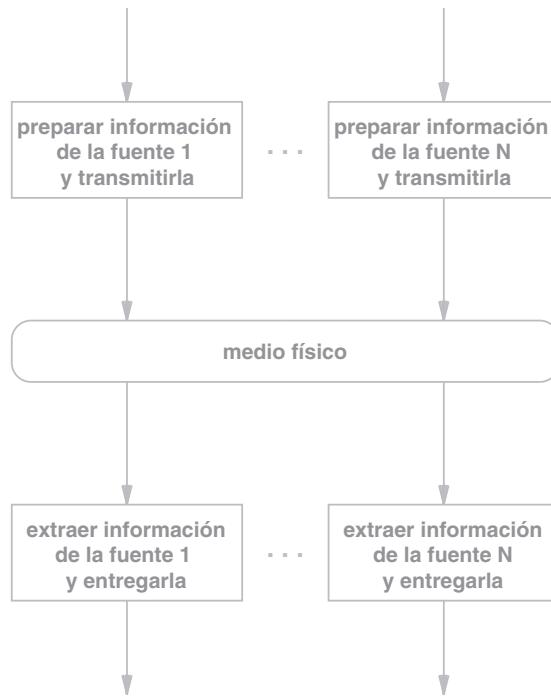


Figura 5.2 Una vista simplista de las comunicaciones de datos con un conjunto de fuentes que envían información a varios destinos, a través de un medio compartido.

En la práctica, las comunicaciones de datos son mucho más complejas de lo que el diagrama de la figura 5.2 sugiere. Puesto que la información puede provenir de muchos tipos de fuentes, las técnicas que se utilizan para manejarlas varían. Antes de poder enviar la información hay que digitalizarla, y es necesario agregar datos adicionales como la protección contra errores. Si la privacidad nos preocupa, tal vez haya que cifrar la información. Para enviar varios flujos de información a través de un sistema de comunicación compartido, hay que identificar la información de cada fuente y entremezclar los datos de todas las fuentes para la transmisión. Por consiguiente, se necesita un mecanismo que identifique cada fuente y garantice que la información de una fuente no se confunda inadvertidamente con la información de otra fuente.

Para explicar los principales aspectos de las comunicaciones de datos, los ingenieros idearon un marco conceptual que muestra cómo se adapta cada subtema a un sistema de comunicaciones. La idea es que cada elemento pueda estudiarse de manera independiente y, una vez que se hayan examinado todas las partes, se comprenderá el tema en su conjunto. La figura 5.3 ilustra esta aproximación y muestra cómo se adaptan los aspectos conceptuales a la organización en general de un sistema de comunicaciones de datos.

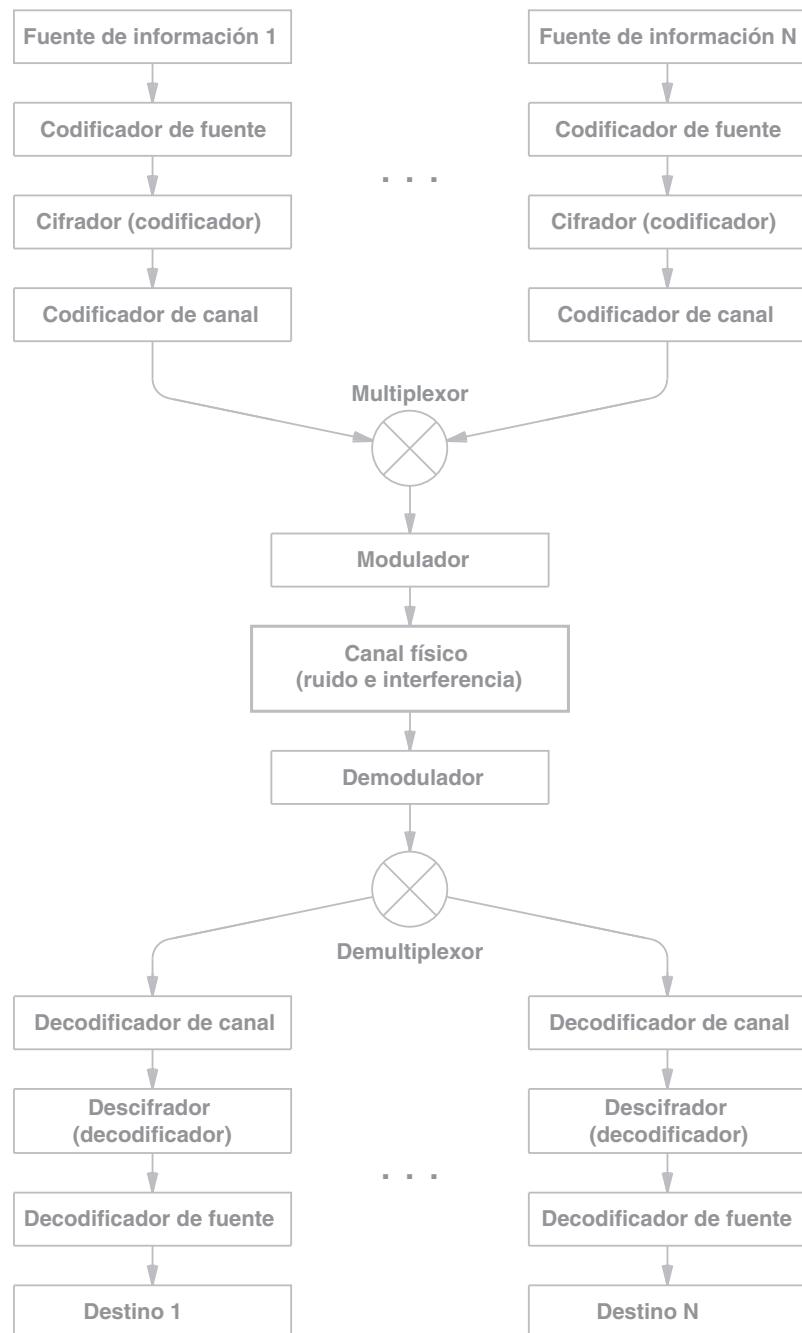


Figura 5.3 Un marco conceptual para un sistema de comunicaciones de datos. Varias fuentes envían a múltiples destinos por medio de un canal físico subyacente.

5.5 Los subtemas de las comunicaciones de datos

Cada uno de los cuadros de la figura 5.3 corresponde a un subtema de las comunicaciones de datos. Los siguientes párrafos explican la terminología. Cada uno de los siguientes capítulos analiza uno de los subtemas conceptuales.

- *Fuentes de información.* Una fuente de información puede ser analógica o digital. Los conceptos importantes incluyen características de las señales, como amplitud, frecuencia y fase. La clasificación puede ser periódica (que ocurre con regularidad) o aperiódica (que ocurre en forma irregular). Además, el subtema se enfoca en la conversión entre las representaciones analógica y digital de la información.
- *Codificador y decodificador de fuente.* Una vez que se digitaliza la información, es posible transformar y convertir las representaciones digitales. Los conceptos importantes incluyen la compresión de datos y sus consecuencias para las comunicaciones.
- *Cifrador y descifrador.* Para proteger la información y mantenerla confidencial, puede cifrarse (es decir, codificarse) antes de la transmisión y descifrarse al momento de recibirla. Los conceptos importantes incluyen técnicas y algoritmos criptográficos.
- *Codificador y decodificador de canal.* La codificación de canal se usa para detectar y corregir errores de transmisión. Entre los conceptos más importantes se encuentran los métodos para detectar y limitar los errores, así como algunas técnicas prácticas como la verificación de paridad, las sumas de verificación y los códigos de redundancia cíclica que se emplean en las redes de computadoras.
- *Multiplexor y demultiplexor.* La multiplexación se refiere a la forma en que se combina la información de varias fuentes para transmitirse a través de un medio compartido. Los conceptos importantes incluyen las técnicas de compartición simultánea, así como técnicas que permiten a las fuentes tomar turnos al momento de usar el medio.
- *Modulador y demodulador.* La modulación se refiere a la forma en que se utiliza la radiación electromagnética para enviar la información. Los conceptos incluyen los esquemas de modulación analógico y digital, y los dispositivos conocidos como módems, que se encargan de realizar los procesos de modulación y demodulación.
- *Canal físico y transmisión.* Este subtema incluye los medios y los modos de transmisión. Los conceptos más importantes incluyen el ancho de banda, el ruido eléctrico y la interferencia, así como la capacidad de canal además de las formas de transmisión, como la transmisión en serie y en paralelo.

5.6 Resumen

Puesto que se encargan de la transmisión a través de medios físicos y de la información digital, las comunicaciones de datos se basan en la física y en las matemáticas. El enfoque se centra en las técnicas que permiten a los ingenieros eléctricos diseñar mecanismos prácticos de comunicación.

Para simplificar la comprensión, los ingenieros idearon un marco conceptual para los sistemas de comunicaciones de datos. Este marco divide todo el tema en un conjunto de subtemas. Cada capítulo subsecuente de esta parte del libro habla sobre uno de los subtemas.

EJERCICIOS

- 5.1** ¿Cuáles son las tres disciplinas involucradas en las comunicaciones de datos?
- 5.2** ¿Cuáles son las motivaciones para las comunicaciones de datos?
- 5.3** ¿Cuáles son las partes conceptuales de un sistema de comunicaciones de datos?
- 5.4** ¿Qué parte de un sistema de comunicaciones de datos se encarga de la entrada analógica?
- 5.5** ¿Qué parte de un sistema de comunicaciones de datos evita que los errores de transmisión corrompan los datos?

Contenido del capítulo

- 6.1 Introducción, 93
- 6.2 Fuentes de información, 93
- 6.3 Señales analógicas y digitales, 94
- 6.4 Señales periódicas y aperiódicas, 94
- 6.5 Ondas senoidales y características de las señales, 95
- 6.6 Señales compuestas, 97
- 6.7 La importancia de las señales compuestas y las funciones senoidales, 97
- 6.8 Representaciones en los dominios de tiempo y de frecuencia, 98
- 6.9 Ancho de banda de una señal analógica, 99
- 6.10 Señales digitales y niveles de señal, 100
- 6.11 Baudios y bits por segundo, 101
- 6.12 Conversión de una señal digital en analógica, 102
- 6.13 Ancho de banda de una señal digital, 103
- 6.14 Sincronización y acuerdo sobre señales, 103
- 6.15 Codificación de línea, 104
- 6.16 Codificación Manchester utilizada en redes de computadoras, 106
- 6.17 Conversión de una señal analógica en digital, 107
- 6.18 El teorema de Nyquist y la tasa de muestreo, 108
- 6.19 Teorema de Nyquist y transmisión del sistema telefónico, 108
- 6.20 Codificación no lineal, 109
- 6.21 Codificación y compresión de datos, 109
- 6.22 Resumen, 110

6

Fuentes de información y señales

6.1 Introducción

El capítulo anterior brinda una descripción general de las comunicaciones de datos, las cuales son la base de todas las redes. El capítulo presenta el tema, proporciona un marco conceptual para las comunicaciones de datos, identifica los aspectos más importantes y explica cómo se adaptan entre sí. El capítulo también ofrece una breve descripción de cada parte conceptual.

Este capítulo comienza una exploración con mayor detalle de las comunicaciones de datos. Analiza los temas de las fuentes de información y las características de las señales que transportan la información. Los capítulos subsiguientes continúan la exploración de las comunicaciones de datos, explicando aspectos adicionales del tema.

6.2 Fuentes de información

Recuerde que un sistema de comunicaciones acepta la entrada de una o más *fuentes* y entrega la información que proviene de una fuente dada hasta un *destino* específico. Para una red, como la Internet global, la fuente y el destino de la información son un par de programas de aplicación que generan y consumen datos. Sin embargo, la teoría de las comunicaciones de datos se concentra en los sistemas de comunicaciones de bajo nivel y se aplica a cualquiera de las fuentes de información. Por ejemplo, además de los periféricos de computadora convencionales, como los teclados y ratones, las fuentes de información pueden incluir micrófonos, cámaras de video, sensores y dispositivos de medición, como termómetros y pesas. De manera similar, los destinos pueden incluir dispositivos de salida de audio, como audífonos y bocinas.

de alta potencia, así como dispositivos tales como radios (por ejemplo, un radio Wi-Fi) o motores eléctricos. En conclusión:

A lo largo del estudio de las comunicaciones de datos, es importante recordar que la fuente de información puede incluir cualquier dispositivo además de las computadoras.

6.3 Señales analógicas y digitales

Las comunicaciones de datos tratan con dos tipos de información: analógica y digital. Una señal analógica se caracteriza por una función matemática continua: cuando la entrada cambia de un valor al siguiente, pasa por todos los valores intermedios posibles. Por el contrario, una señal digital tiene un conjunto fijo de niveles válidos, y cada cambio consiste en un movimiento instantáneo de un nivel válido a otro. La figura 6.1 ilustra el concepto, mostrando ejemplos de cómo las señales de una fuente analógica y una fuente digital varían con el tiempo. En la figura, la señal analógica podría resultar de la medición de salida de un micrófono, mientras que la señal digital podría resultar de la medición de salida de un teclado de computadora.

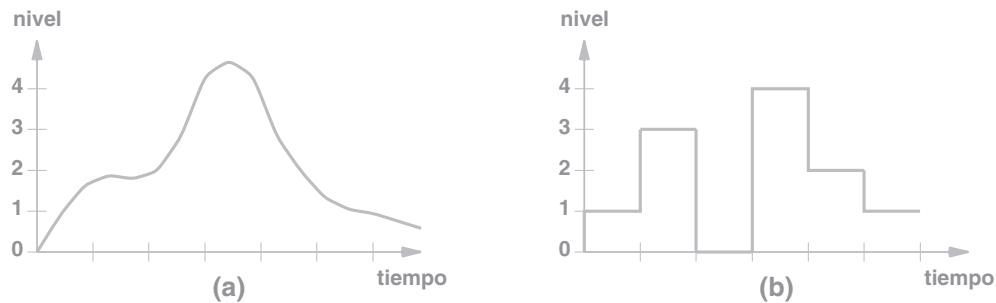


Figura 6.1 Ilustración de (a) una señal analógica y (b) una señal digital.

6.4 Señales periódicas y aperiódicas

Las señales se clasifican ampliamente como *periódicas* si se repiten o *aperiódicas* (a veces conocidas como *no periódicas*) si no lo hacen. Por ejemplo, la señal analógica de la figura 6.1(a) es aperiódica durante el intervalo de tiempo mostrado, ya que la señal no se repite. La figura 6.2 ilustra una señal periódica (es decir, que se repite).

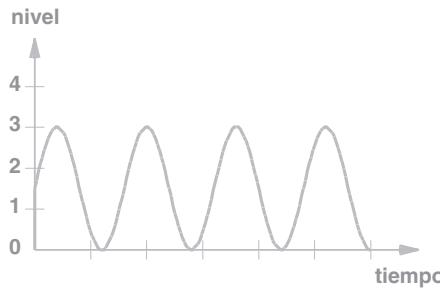


Figura 6.2 Una señal periódica se repite.

6.5 Ondas senoidales y características de las señales

Más adelante veremos que gran parte del análisis de las comunicaciones de datos implica el uso de funciones trigonométricas sinusoidales, en especial la función *seno* que por lo general se abrevia como *sen*. Las ondas senoidales son especialmente importantes en las fuentes de información, ya que los fenómenos naturales producen a menudo este tipo de ondas. Por ejemplo, cuando un micrófono capta un tono audible, la salida es una onda senoidal. De manera similar, la radiación electromagnética puede representarse como una onda senoidal. Nos interesan específicamente las ondas senoidales que corresponden a una señal que oscila en el tiempo, como la onda que ilustra la figura 6.2. En conclusión:

Las ondas senoidales son fundamentales para el procesamiento de entradas debido a que muchos fenómenos naturales producen una señal que corresponde a una onda senoidal como una función del tiempo.

Hay cuatro características importantes de las señales que se relacionan con las ondas senoidales:

- Frecuencia: el número de oscilaciones por unidad de tiempo (por lo general, segundos)
- Amplitud: la diferencia entre las alturas máxima y mínima de la señal
- Fase: qué tanto se desplaza el inicio de la onda senoidal de un tiempo de referencia
- Longitud de onda: la longitud de un ciclo a medida que una señal se propaga a través de un medio

La longitud de onda se determina mediante la velocidad con la que se propaga una señal (es decir, es una función del medio subyacente). Podemos usar una expresión matemática para especificar las otras tres características. La amplitud es más fácil de comprender. Recuerde que $\text{sen}(\omega t)$ produce valores entre -1 y $+1$, y tiene una amplitud de 1 . Si la función *sen* se multiplica por A , la amplitud de la onda resultante es A . En sentido matemático, la fase es una compensación que se agrega a t para desplazar la onda senoidal a la derecha o izquierda a lo largo del eje x . Por consiguiente, $\text{sen}(\omega t + \varphi)$ tiene una fase de φ . La frecuencia de una señal se mide en el número de ciclos por segundo de la onda senoidal o *Hertz*. Una onda senoidal completa requiere 2π radianes. Por lo tanto, si t es un tiempo en segundos y $\omega = 2\pi$, $\text{sen}(\omega t)$ tiene una frecuencia de 1 Hertz. La figura 6.3 ilustra las tres características matemáticas.

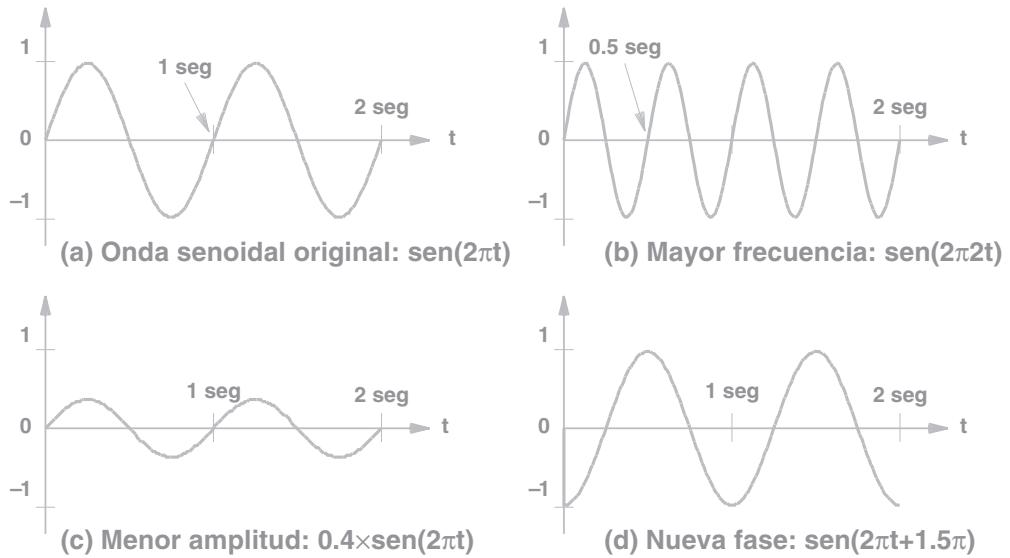


Figura 6.3 Ilustración de las características de frecuencia, amplitud y fase.

La frecuencia puede calcularse como el inverso del tiempo requerido para un ciclo, lo que se conoce como *periodo*. La onda senoidal de la figura 6.3(a) tiene un periodo de $T = 1$ segundos y una frecuencia de $1/T$ o 1 Hertz. El ejemplo de la figura 6.3(b) tiene un periodo de $T = 0.5$ segundos, por lo que su frecuencia es de 2 Hertz; ambas se consideran frecuencias extremadamente *bajas*. Los sistemas de comunicación comunes usan frecuencias *altas*, que a menudo se miden en millones de ciclos por segundo. Para aclarar las altas frecuencias, los ingenieros expresan el tiempo en fracciones de segundo o la frecuencia en unidades como *megahertz*. La figura 6.4 enlista las unidades de tiempo y los prefijos comunes que se utilizan con la frecuencia.

Unidad de tiempo	Valor	Unidad de frecuencia	Valor
Segundos (s)	10^0 segundos	Hertz (Hz)	10^0 Hz
Milisegundos (ms)	10^{-3} segundos	Kilohertz (KHz)	10^3 Hz
Microsegundos (μ s)	10^{-6} segundos	Megahertz (MHz)	10^6 Hz
Nanosegundos (ns)	10^{-9} segundos	Gigahertz (GHz)	10^9 Hz
Picosegundos (ps)	10^{-12} segundos	Terahertz (THz)	10^{12} Hz

Figura 6.4 Prefijos y abreviaciones para unidades de tiempo y frecuencia.

6.6 Señales compuestas

Las señales como la que se ilustra en la figura 6.3 se clasifican como *simples*, ya que consisten en una sola onda senoidal que no puede descomponerse más. En la práctica, la mayoría de las señales se clasifican como *compuestas* debido a que la señal puede descomponerse en un conjunto de ondas senoidales simples. Por ejemplo, la figura 6.4 ilustra una señal compuesta formada por la suma de dos ondas senoidales simples.

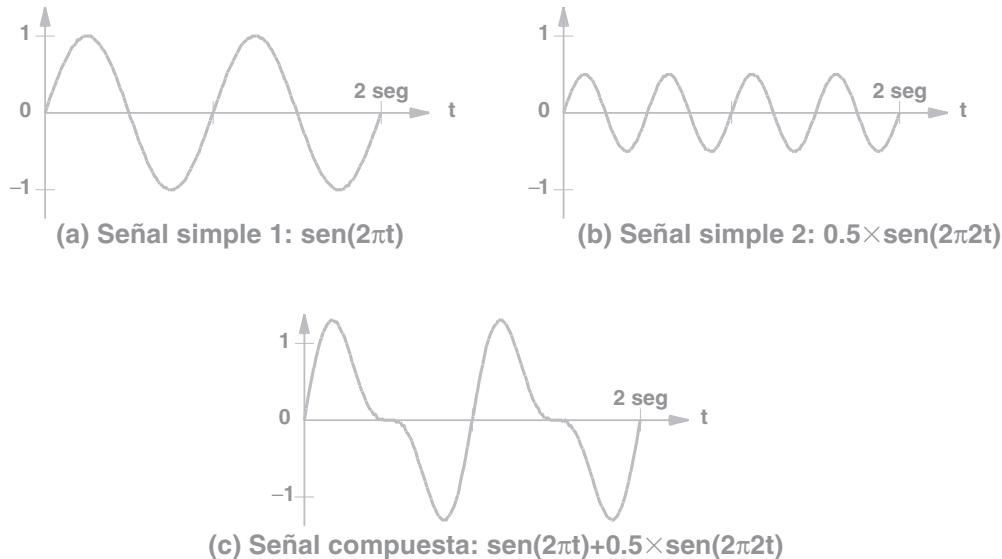


Figura 6.5 Ilustración de una señal compuesta formada por dos señales simples.

6.7 La importancia de las señales compuestas y las funciones senoidales

¿Por qué las comunicaciones de datos se centran en funciones senoidales y señales compuestas? Cuando hablemos sobre modulación y demodulación, comprenderemos una de las razones principales: las señales que resultan de la modulación son por lo general señales compuestas. Por ahora sólo importa comprender lo siguiente:

- Por lo general, la modulación forma una señal compuesta.
- Un matemático llamado Fourier descubrió que es posible descomponer una señal compuesta en las partes que la constituyen: un conjunto de funciones senoidales, cada una con frecuencia, amplitud y fase.

El análisis de Fourier muestra que si la señal compuesta es periódica, las partes que la constituyen también serán periódicas. Por ende, veremos que la mayoría de los sistemas de comunicaciones de datos

usan las señales compuestas para transportar información. Una señal compuesta se crea en el extremo emisor y el receptor la descompone en los componentes simples originales. En conclusión:

Un método matemático descubierto por Fourier permite que un receptor descomponga una señal compuesta en las partes que la componen.

6.8 Representaciones en los dominios de tiempo y de frecuencia

Debido a su importancia, las señales compuestas han sido muy estudiadas y se han inventado varios métodos para representarlas. Ya vimos una representación en las figuras anteriores: un gráfico de una señal como función del tiempo. Los ingenieros dicen que dicho gráfico representa la señal en el *dominio del tiempo*.

La alternativa principal para una representación en el dominio del tiempo se conoce como representación en el *dominio de la frecuencia*. Un gráfico del dominio de la frecuencia muestra un conjunto de ondas senoidales simples que constituyen una función compuesta. El eje y proporciona la amplitud y el eje x proporciona la frecuencia. Así, la función $A \operatorname{sen}(2\pi t)$ se representa mediante una sola línea de altura A que se posiciona en $x = t$. Por ejemplo, el gráfico del dominio de la frecuencia en la figura 6.6 representa una función compuesta de la figura 6.5(c).[†]

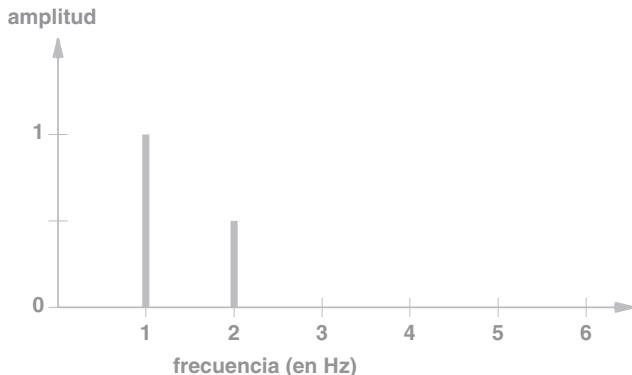


Figura 6.6 Representación de $\operatorname{sen}(2\pi t)$ y $0.5 \operatorname{sen}(2\pi 2t)$ en el dominio de la frecuencia.

La figura muestra un conjunto de señales periódicas simples. También puede usarse una representación del dominio de la frecuencia con señales no periódicas, pero la representación aperiódica no es esencial para una comprensión del tema.

Una de las ventajas de la representación del dominio de la frecuencia surge de su capacidad de compactarse. En comparación con una representación del dominio del tiempo, una representación del dominio de la frecuencia es tanto pequeña como fácil de leer, ya que cada onda senoidal ocupa un solo

[†] Los diagramas del dominio de la frecuencia que se utilizan con los sistemas reales de comunicaciones de datos tienen un eje x que se extiende hasta miles de millones de Hertz.

punto a lo largo del eje x . La ventaja se vuelve clara cuando una señal compuesta contiene muchas señales simples.

6.9 Ancho de banda de una señal analógica

La mayoría de los usuarios han escuchado sobre el “ancho de banda de red” y entienden que es conveniente tener una red con un ancho de banda elevado. Más tarde hablaremos sobre la definición del ancho de banda de una red. Por ahora exploraremos un concepto relacionado, el *ancho de banda analógico*.

Definimos el ancho de banda de una señal analógica como la diferencia entre las frecuencias más alta y más baja de las partes que la componen (es decir, la mayor y menor frecuencias que se obtienen mediante el análisis de Fourier). En el ejemplo simple de la figura 6.5(c), el análisis de Fourier produce señales de 1 y 2 Hertz, lo que significa que el ancho de banda analógico es la diferencia entre ambas, o sea 1 Hertz. La ventaja del gráfico del dominio de la frecuencia es más clara cuando se calcula el ancho de banda analógico, ya que la frecuencia más alta y la más baja son obvias. Por ejemplo, el gráfico en la figura 6.6 hace evidente que el ancho de banda analógico es 1.

La figura 6.7 muestra un gráfico en el dominio de la frecuencia, con frecuencias medidas en Kihertz (KHz). Dichas frecuencias están en el rango audible para el oído humano. En la figura, el ancho de banda es la diferencia entre la frecuencia más alta y la más baja ($5 \text{ KHz} - 1 \text{ KHz} = 4 \text{ KHz}$).

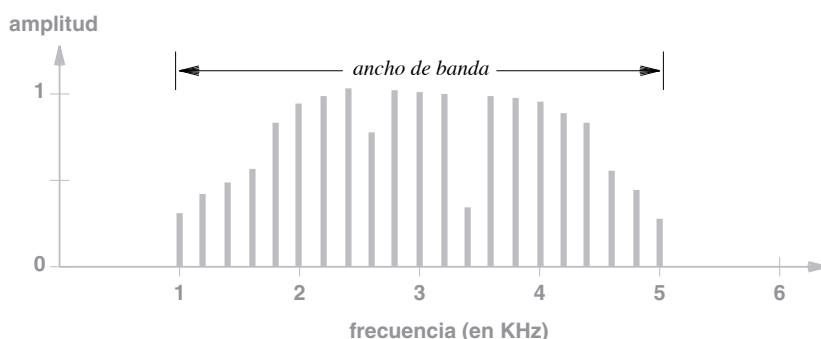


Figura 6.7 Un gráfico del dominio de la frecuencia de una señal analógica con un ancho de banda de 4 KHz.

En conclusión:

El ancho de banda de una señal analógica es la diferencia entre la frecuencia más alta y la más baja de sus componentes. Si la señal se grafica en el dominio de la frecuencia, el ancho de banda se puede calcular de forma simple.

6.10 Señales digitales y niveles de señal

Dijimos que además de representarse mediante una señal analógica, la información también puede representarse mediante una señal *digital*. También definimos que una señal es digital si se eligió un conjunto de niveles válidos y en cualquier momento la señal está en uno de esos niveles válidos. Algunos sistemas usan el voltaje para representar valores digitales al hacer que un valor positivo corresponda con un 1 lógico, y que el valor cero corresponda con un 0 lógico. Por ejemplo, +5 volts pueden usarse para un valor lógico de 1 y 0 volts para un 0 lógico.

Si sólo se usan dos niveles de voltaje, cada nivel corresponde a un bit de datos (0 o 1). Sin embargo, algunos mecanismos de transmisión física pueden soportar más de dos niveles de señal. Cuando hay varios niveles digitales disponibles, cada nivel puede representar varios bits. Por ejemplo, considere un sistema que usa cuatro niveles de voltaje: -5 volts, -2 volts, +2 volts y +5 volts. Cada nivel puede corresponder a dos bits de datos, como lo indica la figura 6.8(b).

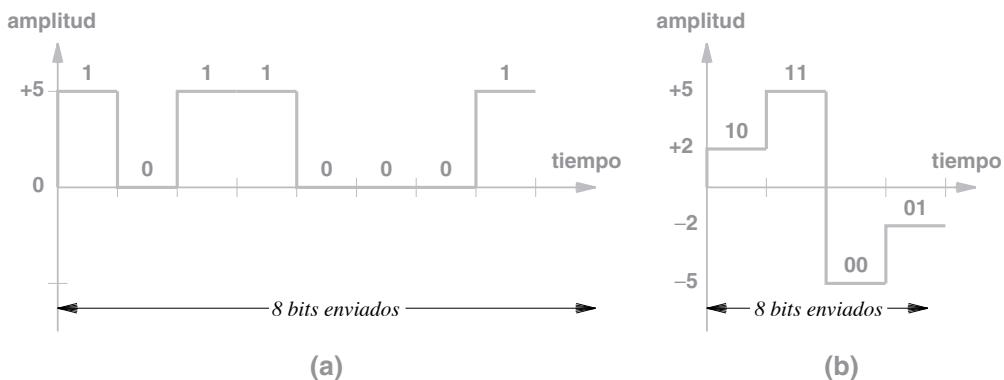


Figura 6.8 (a) Una señal digital que usa dos niveles y (b) la misma señal digital que usa cuatro niveles.

Como lo muestra la figura, la principal ventaja de usar varios niveles de señal surge de la habilidad de representar más de un bit a la vez. Por ejemplo, en la figura 6.8(b) -5 volts representan la secuencia de dos bits 00, -2 volts representan 01, +2 volts representan 10 y +5 volts representan 11. Puesto que se usan varios niveles de señal, cada ranura de tiempo puede transferir dos bits, lo que significa que la representación de cuatro niveles en la figura 6.8(b) tarda la mitad del tiempo en transferir los bits, en comparación con la representación de dos niveles de la figura 6.8(a). Por consiguiente, la velocidad de datos (bits por segundo) se duplica.

La relación entre el número de niveles requeridos y el número de bits a enviar es directa. Debe haber un nivel de señal para cada combinación posible de bits. Puesto que hay 2^n combinaciones posibles con n bits, un sistema de comunicaciones debe usar 2^n niveles para representar n bits. En conclusión:

Un sistema de comunicaciones que usa dos niveles de señales puede enviar un solo bit a la vez; un sistema que soporta 2^n niveles de señales puede enviar n bits a la vez.

Podría parecer que el voltaje es fácilmente medible en cualquier cantidad, y que podríamos obtener cualquier número de niveles al dividir arbitrariamente el voltaje en incrementos pequeños. En sentido matemático, podríamos crear un millón de niveles entre 0 y 1 volts si usamos 0.0000001 volts para un nivel, 0.0000002 para el siguiente nivel y así en lo sucesivo. Por desgracia, los sistemas electrónicos prácticos no pueden distinguir entre señales que difieran por cantidades arbitrarias tan pequeñas. Por consiguiente, en la práctica los sistemas se limitan a unos cuantos niveles de señal.

6.11 Baudios y bits por segundo

¿Cuántos datos pueden enviarse en un momento dado? La respuesta depende de dos aspectos del sistema de comunicación. Como hemos visto, la velocidad a la que pueden enviarse los datos depende del número de niveles de la señal. También es importante un segundo factor: la cantidad de tiempo que el sistema permanece en cierto nivel antes de pasar al siguiente. Por ejemplo, el diagrama de la figura 6.8(a) muestra el tiempo a lo largo del eje x , dividido en ocho segmentos, cada uno con la capacidad de enviar un bit. Si el sistema de comunicaciones se modifica para usar la mitad del tiempo para un bit dado, se enviarán el doble de bits en la misma cantidad de tiempo. En conclusión:

Un método alternativo para incrementar la cantidad de datos transferidos en un momento dado, consiste en reducir el tiempo que el sistema mantiene una señal en un cierto nivel.

Al igual que sucede con los niveles de señal, en la práctica el hardware de un sistema impone límites en cuanto al tiempo mínimo que una señal puede permanecer en un nivel. Si ésta no permanece en un nivel dado el tiempo suficiente, el hardware receptor no podrá detectarla. Lo interesante es que la medida aceptada de un sistema de comunicaciones no especifica una longitud de tiempo. En su lugar, los ingenieros miden el factor inverso: cuántas veces puede cambiar la señal por segundo, lo que se define como *baudio*. Por ejemplo, si un sistema requiere que la señal permanezca en cierto nivel por .001 segundos, decimos que el sistema opera a 1000 baudios.

La idea clave es que tanto los baudios como el número de niveles de señal controlen la velocidad de bits. Si un sistema con dos niveles de señal opera a 1000 baudios, puede transferir exactamente 1000 bits por segundo. Pero si un sistema que opera a 1000 baudios tiene cuatro niveles de señal, puede transferir 2000 bits por segundo (puesto que cuatro niveles de señal pueden representar dos bits). La ecuación 6.1 expresa la relación entre baudios, niveles de señal y velocidad de bits.

$$\text{bits por segundo} = \text{baudios} \times \left[\log_2(\text{niveles}) \right] \quad (6.1)$$

6.12 Conversión de una señal digital en analógica

¿Cómo puede convertirse una señal digital en una señal analógica equivalente? Recuerde que de acuerdo con Fourier, una curva cualquiera puede representarse como un conjunto de ondas senoidales, donde cada onda del conjunto tiene una amplitud, una frecuencia y una fase específicas. Como esto aplica para cualquier curva, el teorema de Fourier también se aplica a una señal digital. Desde una perspectiva de ingeniería, el resultado de Fourier es poco práctico para las señales digitales debido a que la representación precisa de una señal digital requiere de un conjunto infinito de ondas senoidales.

Los ingenieros parten del supuesto que la conversión de una señal digital en una analógica es *aproximada*, por lo que fabrican equipo para generar ondas analógicas que se aproximan mucho a la señal digital. La aproximación implica crear una señal compuesta a partir de unas cuantas ondas senoidales. Al elegir ondas senoidales que son múltiplos exactos de la frecuencia de la señal digital, es posible usar sólo tres ondas. Los detalles precisos están más allá del alcance de este libro, pero la figura 6.9 ilustra la aproximación al mostrar (a) una señal digital con sus aproximaciones, con (b) una sola onda senoidal, (c) un conjunto compuesto por la onda senoidal original más una onda senoidal de 3 veces la frecuencia, y (d) un conjunto compuesto por esta última onda más una onda senoidal adicional de 5 veces la frecuencia original.

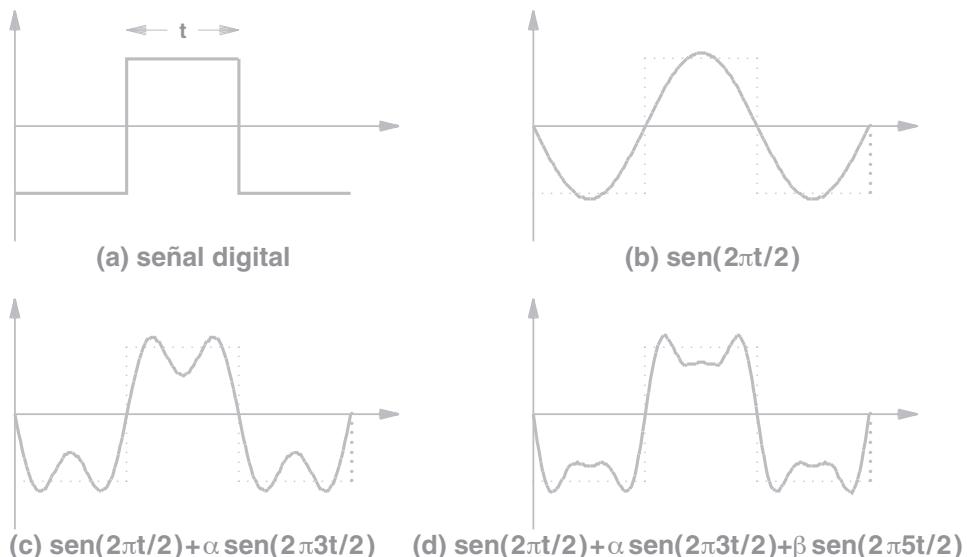


Figura 6.9 Aproximación de una señal digital con ondas senoidales.

6.13 Ancho de banda de una señal digital

¿Cuál es el ancho de banda de una señal digital? Recuerde que el ancho de banda de una señal es la diferencia entre las ondas de frecuencia más alta y la más baja que forman la señal. Por lo tanto, una forma de calcular el ancho de banda consiste en aplicar el análisis de Fourier para buscar las ondas senoidales que constituyen la señal y luego examinar las frecuencias.

En sentido matemático, cuando se aplica el análisis de Fourier a una onda cuadrada, como la señal que se ilustra en la figura 6.9(a), el análisis produce un conjunto infinito de ondas senoidales. Además, las frecuencias en el conjunto continúan hasta el infinito. De esta forma, cuando se trazan en el dominio de la frecuencia, el conjunto continúa a lo largo del eje x hasta el infinito. La consecuencia importante de esto es que:

De acuerdo con la definición del ancho de banda, una señal digital tiene un ancho de banda infinito debido a que el análisis de Fourier de una señal digital produce un conjunto infinito de ondas senoidales con frecuencias que crecen hasta el infinito.

6.14 Sincronización y acuerdo sobre señales

Nuestros ejemplos omiten muchos de los detalles involucrados en la creación de un sistema de comunicaciones viable. Por ejemplo, para garantizar que el emisor y el receptor acepten el tiempo asignado a cada elemento de una señal, los componentes electrónicos en ambos extremos de un medio físico deben tener circuitos para medir el tiempo en forma precisa. Es decir, si un extremo transmite una señal con 10^9 elementos por segundo, el otro extremo debe esperar exactamente 10^9 elementos por segundo. A velocidades bajas, es fácil hacer que ambos extremos estén de acuerdo. Sin embargo, fabricar sistemas electrónicos que coincidan usando las altas velocidades de las redes modernas es extremadamente difícil.

Debido a la forma en que se representan los datos en las señales, surge un problema más fundamental. El problema se relaciona con la *sincronización* del emisor y del receptor. Por ejemplo, suponga que un receptor pierde el primer bit que llega y comienza a interpretar los datos a partir del segundo bit. O considere lo que ocurriría si un receptor espera que los datos lleguen a una velocidad mayor de la que usa el emisor para transmitir los datos. La figura 6.10 ilustra cómo una incongruencia en la interpretación puede producir errores. En la figura, tanto el emisor como el receptor comienzan y terminan en el mismo punto de la señal, pero como el receptor asigna un poco menos de tiempo por bit, el receptor malinterpreta la señal y considera que contiene más bits de los que se enviaron.

En la práctica, los errores de sincronización pueden ser en extremo sutiles. Por ejemplo, suponga que el hardware de un receptor tiene un error de sincronización de 1×10^{-8} . El error podría no aparecer sino hasta que se transmitan diez millones de bits en una secuencia. Puesto que los sistemas de comunicaciones de alta velocidad transfieren gigabits por segundo, dichos errores pequeños pueden salir rápidamente a la superficie y volverse considerables.

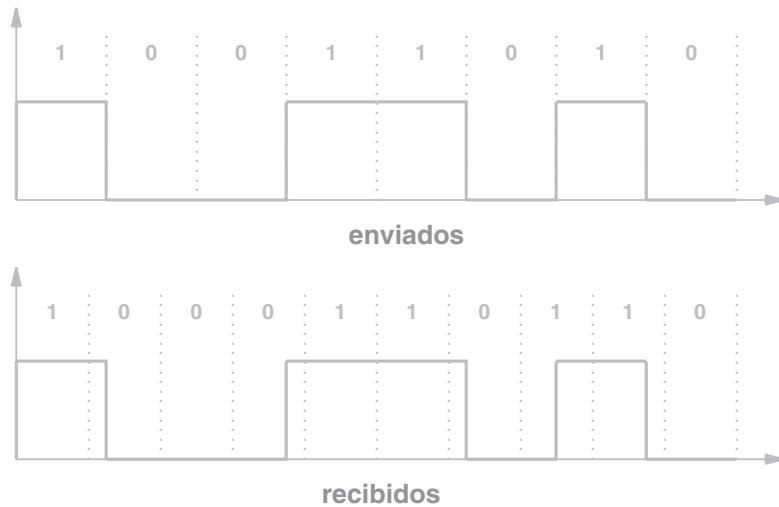


Figura 6.10 Ilustración de un error de sincronización en el que el receptor permite una cantidad mucho menor de tiempo por bit que el emisor.

6.15 Codificación de línea

Se han inventado varias técnicas que pueden ayudar a evitar los errores de sincronización. En general, hay dos metodologías ampliamente utilizadas. En una de ellas, antes de transmitir datos el emisor transmite un patrón conocido de bits (por lo general, un conjunto de dígitos 0 y 1 alternantes) que permite al receptor sincronizarse. En la otra metodología, los datos se representan mediante la señal de tal forma que no pueda haber confusión en cuanto al significado. Usamos el término *codificación de línea* para describir la forma en que se codifican los datos en una señal.

Como ejemplo de una codificación de línea que elimina la ambigüedad, considere cómo podemos usar un mecanismo de transmisión que soporta tres niveles de señal discretos. Para garantizar la sincronización, reserve uno de los niveles de señal para comenzar cada bit. Por ejemplo, si los tres posibles niveles corresponden a -5 , 0 y $+5$ volts, reserve -5 para comenzar cada bit. El 0 lógico puede representarse mediante la secuencia $-5\ 0$, y el 1 lógico puede representarse mediante la secuencia $-5\ +5$. Si especificamos que no hay otras combinaciones válidas, la ocurrencia de -5 volts siempre comienza un bit y un receptor puede usar una ocurrencia de -5 volts para sincronizarse correctamente con el emisor. La figura 6.11 ilustra la representación.

Desde luego que usar varios elementos de señal para representar un solo bit significa que se pueden transmitir menos bits por unidad de tiempo. Por ende, los diseñadores prefieren esquemas que transmitan varios bits por elemento de señal, como el que ilustra la figura 6.8(b).[†]

[†] La figura 6.8 está en la página 100.

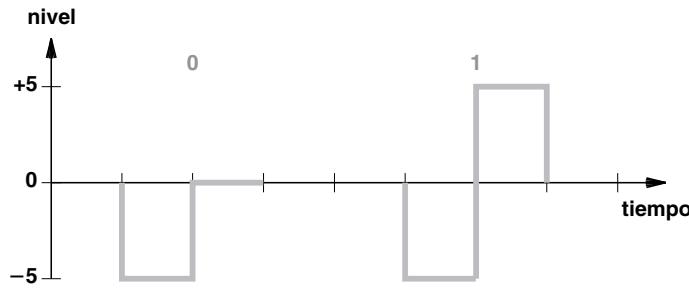


Figura 6.11 Ejemplo de dos elementos de señal utilizados para representar cada bit.

La figura 6.12 enumera los nombres de las técnicas de codificación de línea de uso común, y las agrupa en categorías relacionadas. Aunque los detalles están más allá del alcance de este libro, basta con saber que la opción depende de las necesidades específicas de un sistema de comunicaciones dado.

Categoría	Esquema	Sincronización
Unipolar	NRZ NRZ-L NRZ-I Bifásica	No, si se repiten muchos dígitos 0 y 1 No, si se repiten muchos dígitos 0 y 1 No, si se repiten muchos dígitos 0 y 1 Sí
Bipolar	AMI	No, si se repiten muchos dígitos 0
Multinivel	2B1Q 8B6T 4D-PAM5	No, si se repiten muchos bits dobles Sí Sí
Multilínea	MLT-3	No, si se repiten muchos dígitos 0

Figura 6.12 Nombres de las técnicas de codificación de línea de uso común.

En conclusión:

Hay una variedad de técnicas de codificación de línea disponibles que difieren en la forma en que manejan la sincronización además de otras propiedades, como el ancho de banda utilizado.

6.16 Codificación Manchester utilizada en redes de computadoras

Además de la lista de la figura 6.12, hay un estándar específico para la codificación de línea que es especialmente importante para las redes de computadoras: la *codificación Manchester* que se utiliza con Ethernet.[†]

Para comprender la codificación Manchester, es importante saber que es más fácil detectar una transición en el nivel de la señal que medir el nivel en sí. El hecho, que surge de la forma en que funciona el hardware, explica por qué la codificación Manchester usa transiciones en vez de niveles para definir los bits. Es decir, en vez de especificar que 1 corresponde a un nivel (por ejemplo, +5 volts), la codificación Manchester especifica que un 1 corresponde a una transición que va de 0 volts a un nivel de voltaje positivo. A su vez, un 0 corresponde a una transición que va de un nivel de voltaje positivo a cero. Además, las transiciones ocurren en “medio” de la ranura de tiempo asignada para un bit, lo cual permite que la señal regrese al nivel anterior en caso de que los datos contengan dos dígitos 0 o dos dígitos 1 repetidos. La figura 6.13(a) ilustra el concepto.

Una variación conocida como *codificación Manchester diferencial* (también conocida como *codificación condicional de desfase*) usa transiciones relativas en vez de absolutas. Es decir, la representación de un bit depende del bit anterior. Cada ranura de tiempo de un bit contiene una o dos transiciones. Una transición *siempre* ocurre a mitad del tiempo de un bit. El valor lógico del bit se representa mediante la presencia o ausencia de una transición al principio del tiempo de un bit: el 0 lógico se representa mediante una transición y el 1 lógico se representa mediante ninguna transición. La figura 6.13(b) ilustra la codificación Manchester diferencial. Tal vez la propiedad más importante de la codificación diferencial surge de una consideración práctica: la codificación funciona correctamente aun cuando los dos cables que transportan la señal se inviertan de manera accidental.

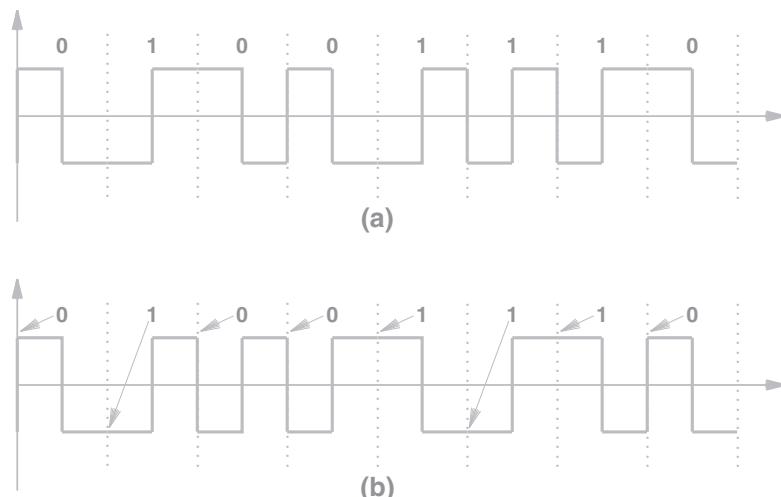


Figura 6.13 Codificaciones (a) Manchester y (b) Manchester diferencial; cada una asume que el bit anterior terminó con un nivel de señal bajo.

[†] En el capítulo 15 hablaremos sobre Ethernet.

6.17 Conversión de una señal analógica en digital

Muchas fuentes de información son analógicas, lo que significa que deben convertirse a la forma digital para poder ser procesadas (por ejemplo, antes de cifrarlas). Hay dos metodologías básicas:

- Modulación por código de pulso
- Modulación delta

La *modulación por código de pulso (PCM)*[†] se refiere a una técnica donde el nivel de una señal analógica se mide de manera repetida en intervalos de tiempo fijos y se convierte en formato digital. La figura 6.14 ilustra los pasos.

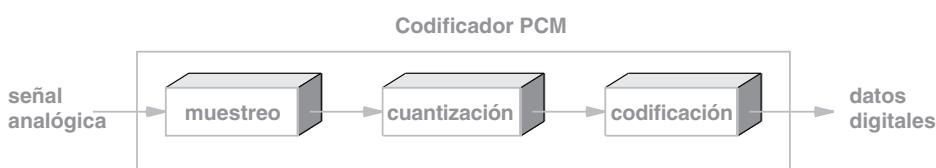


Figura 6.14 Los tres pasos utilizados en la modulación por código de pulso.

Cada medición se conoce como una *muestra*, lo que explica por qué la primera etapa se conoce como *muestreo*. Una vez que se registra, la muestra se *cuantiza* al convertirse en un valor entero pequeño, que posteriormente se *codifica* en un formato específico. El valor cuantizado no es una medida de voltaje ni de cualquier otra propiedad de la señal. En su lugar, el rango de la señal de los niveles mínimo a máximo se divide en un conjunto de ranuras, por lo general una potencia de 2. La figura 6.15 ilustra el concepto, para lo cual muestra una señal cuantizada en ocho ranuras.

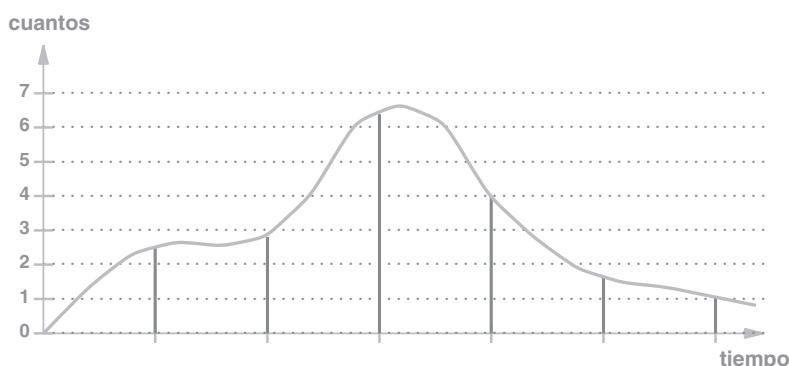


Figura 6.15 Una ilustración del muestreo y la cuantización que se utilizan en la modulación por código de pulso.

[†] El acrónimo PCM es ambiguo, ya que puede referirse a la idea general o a una forma específica de modulación por código de pulso utilizada por el sistema telefónico. En una sección posterior hablaremos sobre esto último.

En la figura, las seis muestras se representan mediante líneas grises verticales. Cada muestra se cuantiza eligiendo el intervalo del cuanto más cercano. Por ejemplo, a la tercera muestra, que se toma cerca del pico de la curva, se le asigna un valor cuantizado de 6.

En la práctica se han inventado ligeras variaciones en el muestreo. Por ejemplo, para evitar la imprecisión provocada por un breve pico o valle en la señal, puede usarse un promedio. Es decir, en vez de depender de una sola medición para cada muestra, pueden tomarse tres muestras cercanas y se calcula una media aritmética.

La principal alternativa para la modulación por código de pulso se conoce como *modulación delta*. Esta modulación también toma muestras. Sin embargo, en vez de enviar una cuantización para cada muestra, la modulación delta envía un valor de cuantización seguido de una cadena de valores que proporcionan la diferencia entre el valor anterior y el actual. La ventaja es que para transmitir las diferencias se requieren menos bits que para transmitir valores completos, en especial si la señal no varía rápidamente. La principal desventaja con la modulación delta surge debido al efecto de un error: si se pierde o daña uno de los elementos en la secuencia, se interpretarán de manera errónea todos los valores sucesivos. Por consiguiente, los sistemas de comunicaciones cuyos valores de datos sean factibles de pérdida o modificación durante la transmisión, usan por lo general la modulación por código de pulso (PCM).

6.18 El teorema de Nyquist y la tasa de muestreo

Ya sea que se utilice la modulación por código de pulso o delta, hay que muestrear la señal analógica. ¿Con qué frecuencia debe muestrearse una señal analógica? Si se toman muy pocas muestras (lo que se conoce como *submuestreo*) significa que los valores digitales sólo darán una cruda aproximación de la señal original. Si se toman demasiadas muestras (lo que se conoce como *sobremuestreo*) significa que se generarán más datos digitales, para lo cual se requiere de un ancho de banda adicional.

Un matemático llamado Nyquist descubrió la respuesta a la pregunta de cuánto muestreo se requiere:

$$\text{velocidad de muestreo} = 2 \times f_{\text{máx}} \quad (6.2)$$

donde $f_{\text{máx}}$ es la frecuencia más alta en la señal compuesta. El resultado, que se conoce como *teorema de Nyquist*, ofrece una solución práctica al problema: hay que muestrear una señal al menos dos veces más rápido que la frecuencia más alta que deseé preservarse.

6.19 Teorema de Nyquist y transmisión del sistema telefónico

Como un ejemplo específico del teorema de Nyquist, considere el sistema telefónico que se diseñó originalmente para transferir voz. Las mediciones de la voz humana han mostrado que si se conservan las frecuencias entre 0 y 4000 Hz se puede obtener una calidad de audio aceptable. Por consiguiente, el teorema de Nyquist especifica que, al convertir una señal de voz de analógica a digital, hay que muestrear la señal a una velocidad de 8000 muestras por segundo.

Para ofrecer una reproducción de calidad razonable, el estándar PCM que utiliza el sistema telefónico cuantifica cada muestra en un valor de 8 bits. Es decir, el rango de la entrada se divide en 256 niveles posibles, de modo que cada muestra tenga un valor entre 0 y 255. Como consecuencia, la velocidad a la que se generan los datos digitales para una sola llamada telefónica es:

$$\text{llamada de voz digitalizada} = 8000 \frac{\text{muestras}}{\text{segundo}} \times 8 \frac{\text{bits}}{\text{muestra}} = 64,000 \frac{\text{bits}}{\text{segundo}} \quad (6.3)$$

Como veremos en capítulos posteriores, el sistema telefónico usa la velocidad de 64,000 bits por segundo (64 Kbps) como la base para la comunicación digital. Más adelante veremos que Internet utiliza circuitos telefónicos digitales para abarcar largas distancias.

6.20 Codificación no lineal

Cuando cada muestra sólo tiene ocho bits, la codificación PCM lineal que se ilustra en la figura 6.15 no funciona bien para la voz. Los investigadores han ideado alternativas no lineales que pueden reproducir sonidos a los que el oído humano es más sensible. Se crearon dos estándares telefónicos digitales no lineales, y se utilizan mucho en la actualidad:

- a -law, un estándar utilizado en Europa
- μ -law, un estándar utilizado en América del Norte y Japón

Ambos estándares usan muestras de 8 bits y generan 8000 muestras por segundo. La diferencia entre los dos surge de un compromiso entre el rango total y la sensibilidad al ruido. El algoritmo μ -law tiene la ventaja de que cubre un rango dinámico más extenso (es decir, tiene la posibilidad de reproducir sonidos más fuertes), pero tiene la desventaja de presentar más distorsión en las señales débiles. El algoritmo a -law ofrece una menor distorsión de las señales débiles pero tiene un rango dinámico menor. Para llamadas internacionales, hay que realizar una conversión a la codificación a -law si uno de los lados usa a -law y el otro usa μ -law.

6.21 Codificación y compresión de datos

Usamos el término *compresión de datos* para referirnos a una técnica que reduce el número de bits requeridos para representar datos. La compresión de datos es especialmente relevante para un sistema de comunicaciones, ya que al reducir el número de bits utilizados para representar datos, se reduce el tiempo requerido para la transmisión. Es decir, podemos optimizar un sistema de comunicaciones al comprimir los datos antes de la transmisión.

El capítulo 28 considera la compresión en las aplicaciones multimedia. En este momento, sólo necesitamos comprender las definiciones básicas de los dos tipos de compresión:

- Con pérdidas: se pierde información durante la compresión
- Sin pérdidas: toda la información se conserva en la versión comprimida

Por lo general, la compresión *con pérdidas* se utiliza para datos que son percibidos por el ser humano, como una imagen, un clip de video o un archivo de audio. La idea clave es que la compresión sólo necesita conservar los detalles para el nivel de la percepción humana. Es decir, un cambio es aceptable si el ser humano no puede detectarlo. Más adelante veremos que los esquemas de compresión reconocidos, como JPEG (se utiliza para imágenes) o MPEG-3 (se abrevia como MP3 y se utiliza para grabaciones de audio) emplean la compresión con pérdidas.

La compresión *sin pérdidas* conserva los datos originales sin ninguna modificación. Por consiguiente, la compresión sin pérdidas puede usarse para documentos o en cualquier situación donde deban conservarse los datos con exactitud. Cuando se utilizan para la comunicación, un emisor comprime los datos antes de la transmisión, y el receptor descomprime el resultado. Puesto que la compresión es sin pérdidas un emisor puede comprimir los datos y un receptor los puede descomprimir para recibir una copia exacta del original.

La mayoría de la compresión sin pérdidas usa un enfoque de *diccionario*. La compresión busca las cadenas que se repiten en los datos y forma un *diccionario* de éstas. Para comprimir los datos, cada ocurrencia de una cadena se sustituye por una referencia hacia el diccionario. El emisor debe transmitir el diccionario junto con los datos comprimidos. Si los datos contienen cadenas que se repiten muchas veces, la combinación del diccionario más los datos comprimidos es de menor tamaño que los datos originales.

6.22 Resumen

Una fuente de información puede proporcionar datos analógicos o digitales. Una señal analógica tiene la propiedad de ser aperiódica o periódica; una señal periódica tiene propiedades de amplitud, frecuencia y fase. Fourier descubrió que es posible formar una curva cualquiera a partir de una suma de ondas senoidales; una sola onda senoidal se clasifica como muestra y una señal que puede descomponerse en varias ondas senoidales se clasifica como señal compuesta.

Los ingenieros usan dos representaciones principales de las señales compuestas. Una representación en el dominio del tiempo muestra cómo varía la señal con el tiempo. Una representación en el dominio de la frecuencia muestra la amplitud y la frecuencia de cada componente de la señal. El ancho de banda, que es la diferencia entre las frecuencias más alta y más baja de una señal, se ve con mucha claridad en un gráfico del dominio de la frecuencia.

La velocidad en baudios de una señal es el número de veces que ésta puede cambiar por segundo. Una señal digital que usa varios niveles de señal puede representar más de un bit por cambio, lo cual hace que la velocidad de transmisión efectiva sea el número de niveles multiplicado por la velocidad en baudios. Aunque tiene ancho de banda infinito, una señal digital puede ser aproximada sólo con tres ondas senoidales.

Existen varias técnicas de codificación de línea. La codificación Manchester que se utiliza con redes Ethernet, resulta especialmente importante. En vez de usar niveles de señal absolutos para representar bits, la codificación Manchester usa transiciones en el nivel de la señal. La codificación Manchester diferencial usa transiciones relativas y tiene la propiedad que funciona incluso aunque se inviertan los dos cables.

La modulación por código de pulso y la modulación delta se usan para convertir una señal analógica en digital. El esquema PCM que utiliza el sistema telefónico emplea una cuantización de 8 bits y toma 8000 muestras por segundo, lo que produce una velocidad de 64 Kbps.

La compresión puede ser con o sin pérdidas. La compresión con pérdidas es más apropiada para imágenes, audio o video que puedan percibir los seres humanos, ya que la pérdida puede ser controlada para mantener los cambios debajo del umbral de la percepción humana. La compresión sin pérdidas es más apropiada para documentos o datos que deben conservarse con exactitud.

EJERCICIOS

- 6.1** Mencione tres ejemplos de fuentes de información aparte de las computadoras.
- 6.2** Nombre un dispositivo común en el hogar que emita una señal aperiódica.
- 6.3** ¿Por qué son las ondas senoidales fundamentales para las comunicaciones de datos?
- 6.4** Indique y describa las cuatro características fundamentales de una onda senoidal.
- 6.5** Cuando se muestra un gráfico de una onda senoidal, ¿cuál es la forma más rápida de determinar si la fase es cero?
- 6.6** ¿Cuándo se clasifica una onda como *simple*?
- 6.7** ¿Qué produce el análisis de Fourier de una onda compuesta?
- 6.8** En un gráfico del dominio de la frecuencia, ¿qué representa el eje y?
- 6.9** ¿Qué es el ancho de banda analógico de una señal?
- 6.10** ¿Es más fácil calcular el ancho de banda a partir de una representación del dominio del tiempo o de la frecuencia? ¿Por qué?
- 6.11** Suponga que un ingeniero aumenta el número de niveles de señal posibles de dos a cuatro. ¿Cuántos bits más pueden enviarse en la misma cantidad de tiempo? Explique.
- 6.12** ¿Cuál es la definición de *baudio*?
- 6.13** ¿Por qué se usa una señal analógica para aproximarse a una señal digital?
- 6.14** ¿Qué es el ancho de banda de una señal digital? Explique.
- 6.15** ¿Qué es un error de sincronización?
- 6.16** ¿Por qué algunas técnicas de codificación usan varios elementos de señal para representar un solo bit?
- 6.17** ¿Qué aspecto de una señal usa la codificación Manchester para representar un bit?
- 6.18** ¿Cuál es la principal ventaja de una codificación Manchester diferencial?
- 6.19** Al convertir una señal analógica a digital, ¿cuál es el paso que sigue al muestreo?
- 6.20** Si la máxima frecuencia audible para el oído humano es de 20,000 Hz, ¿a qué velocidad debe muestrearse la señal analógica de un micrófono al convertirla en digital?
- 6.21** ¿Qué tiempo transcurre entre cada muestra para la codificación PCM que se utiliza en el sistema telefónico?
- 6.22** Describa la diferencia entre las compresiones con y sin pérdidas, e indique cuándo podría usarse cada una de ellas.

Contenido del capítulo

- 7.1 Introducción, 113
- 7.2 Transmisión guiada y no guiada, 113
- 7.3 Una taxonomía mediante formas de energía, 114
- 7.4 Radiación de fondo y ruido eléctrico, 115
- 7.5 Cable de cobre de par trenzado, 115
- 7.6 Blindaje: cable coaxial y par trenzado blindado, 117
- 7.7 Categorías de cable de par trenzado, 118
- 7.8 Medios que usan energía luminosa y fibra óptica, 119
- 7.9 Tipos de fibra y transmisión de luz, 120
- 7.10 Comparación entre fibra óptica y cable de cobre, 121
- 7.11 Tecnologías de comunicación infrarrojas, 122
- 7.12 Comunicación láser de punto a punto, 122
- 7.13 Comunicación electromagnética (radio), 123
- 7.14 Propagación de señales, 124
- 7.15 Tipos de satélites, 125
- 7.16 Satélites en órbita terrestre geoestacionaria (GEO), 126
- 7.17 Cobertura GEO de la Tierra, 127
- 7.18 Satélites en órbita terrestre baja (LEO)
y grupos de satélites, 128
- 7.19 Ventajas y desventajas entre los tipos de medios, 128
- 7.20 Medición de los medios de transmisión, 129
- 7.21 El efecto del ruido en la comunicación, 129
- 7.22 El significado de la capacidad de un canal, 130
- 7.23 Resumen, 131

7

Medios de transmisión

7.1 Introducción

El capítulo 5 presenta una descripción general de las comunicaciones de datos. El capítulo anterior considera el tema de las fuentes de información, cubre los sistemas de información analógicos y digitales, y explica las codificaciones.

Este capítulo continúa la explicación de las comunicaciones de datos considerando los medios de transmisión, incluyendo los medios alámbricos, inalámbricos y ópticos. El capítulo proporciona una taxonomía de los tipos de medios, introduce los conceptos básicos de la propagación electromagnética y explica cómo el blindaje puede reducir o evitar la interferencia y el ruido. Por último, el capítulo explica el concepto de *capacidad de un canal*. Los capítulos subsiguientes continúan la explicación sobre las comunicaciones de datos.

7.2 Transmisión guiada y no guiada

¿En qué clases deben dividirse los medios de transmisión? Hay dos enfoques principales:

- Por tipo de ruta: la comunicación puede seguir una ruta exacta como un cable, o puede no tener una ruta específica, como una transmisión por radio.
- Por tipo de energía: la energía eléctrica se usa en los cables, la transmisión por radio se usa para medios inalámbricos y la luz se usa para la fibra óptica.

Usamos los términos transmisión *guiada* y *no guiada* para diferenciar entre los medios físicos como el cable de cobre y la fibra óptica, que proporcionan una ruta específica, y una transmisión de radio que viaja en todas direcciones a través del espacio libre. De manera informal, los ingenieros usan los términos *alámbrica* e *inalámbrica*. Tenga en cuenta que la información puede ser algo confusa debido a que es probable que escuchemos el término *alámbrica* incluso aunque el medio físico sea una fibra óptica.

7.3 Una taxonomía mediante formas de energía

La figura 7.1 ilustra cómo pueden clasificarse los medios físicos de acuerdo con la forma de energía utilizada para transmitir los datos. Las siguientes secciones describen cada uno de los tipos de medios.

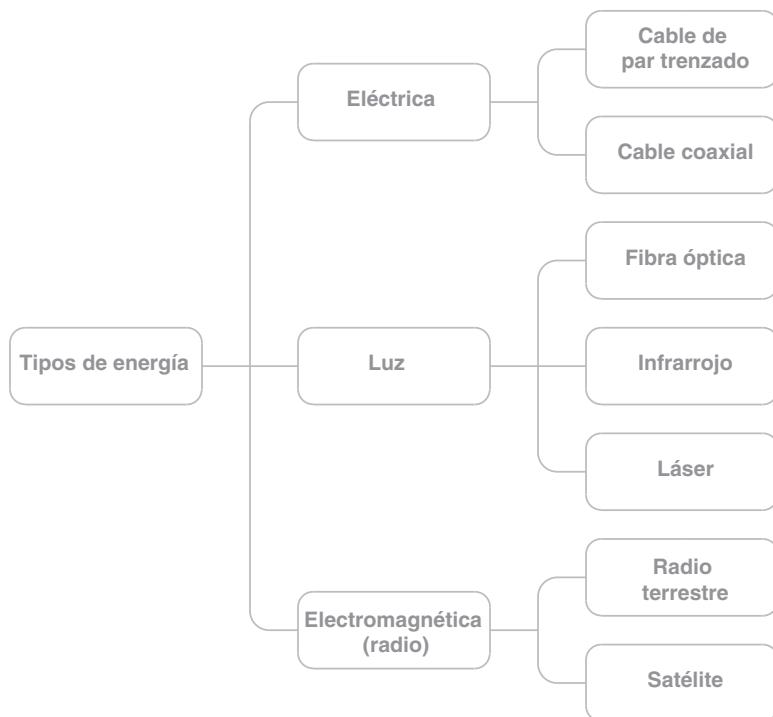


Figura 7.1 Una taxonomía de los tipos de medios, de acuerdo con la forma de energía que se utilice.

Al igual que la mayoría de las taxonomías, las categorías no son perfectas y existen excepciones. Por ejemplo, una estación espacial en órbita alrededor de la Tierra podría emplear una comunicación no terrestre que no involucre a un satélite. Sin embargo, nuestra taxonomía cubre la mayoría de las comunicaciones.

7.4 Radiación de fondo y ruido eléctrico

Recordemos de la física básica que la corriente fluye a lo largo de un circuito completo. Por consiguiente, todas las transmisiones de energía eléctrica necesitan dos alambres para formar un circuito: un alambre que va al receptor y otro alambre de vuelta al emisor. La forma más simple de cableado consiste en un cable que contiene dos alambres de cobre. Cada alambre va envuelto en un recubrimiento de plástico que lo aísla eléctricamente. Un recubrimiento exterior en el cable mantiene unidos los alambres para facilitar a los humanos el proceso de conectar el equipo.

Las redes de computadoras usan una forma alternativa de cableado. Para comprender por qué, debemos conocer tres puntos.

- La radiación electromagnética aleatoria, conocida como *ruido*, se infiltra en el entorno. De hecho, los sistemas de comunicaciones generan pequeñas cantidades de ruido eléctrico como efecto secundario de la operación normal.
- Cuando choca con metal, la radiación electromagnética induce una pequeña señal, lo que significa que el ruido aleatorio puede interferir con las señales utilizadas para la comunicación.
- Como absorbe la radiación, el metal actúa como *blindaje*. Entonces, si colocamos suficiente metal entre una fuente de ruido y un medio de comunicación, podemos evitar que el ruido interfiera con la comunicación.

Los primeros dos puntos señalan un problema fundamental inherente en los medios de comunicación que usan energía eléctrica o de radio. El problema es especialmente severo cerca de una fuente que emite radiación aleatoria. Por ejemplo, las bombillas de luz fluorescente y los motores eléctricos emiten radiación, en especial los motores poderosos como los que se utilizan para operar elevadores, aires acondicionados y refrigeradores. Lo sorprendente es que los dispositivos más pequeños, como las trituradoras de papel o las herramientas eléctricas, pueden también emitir suficiente radiación para interferir con la comunicación. En conclusión:

La radiación electromagnética aleatoria generada por dispositivos tales como los motores eléctricos puede interferir con la comunicación que usa transmisión de radio o energía eléctrica enviada a través de alambres.

7.5 Cable de cobre de par trenzado

El tercer punto en la sección anterior explica el cableado que se utiliza con los sistemas de comunicaciones. Hay tres formas de cableado que ayudan a reducir la interferencia del ruido eléctrico:

- Par trenzado sin blindaje (UTP)
- Cable coaxial
- Par trenzado blindado (STP)

La primera forma, que se conoce como cableado de *par trenzado* o cableado de *par trenzado sin blindaje*,[†] se usa mucho en las comunicaciones. Como el nombre implica, el cableado de par trenzado consiste en dos cables trenzados entre sí. Desde luego que cada cable tiene un recubrimiento de plástico que aísla los dos cables y evita que la corriente eléctrica fluya a través de ellos.

Lo sorprendente es que al trenzar dos cables se hacen menos susceptibles al ruido eléctrico que si se dejan en paralelo. La figura 7.2 ilustra por qué.

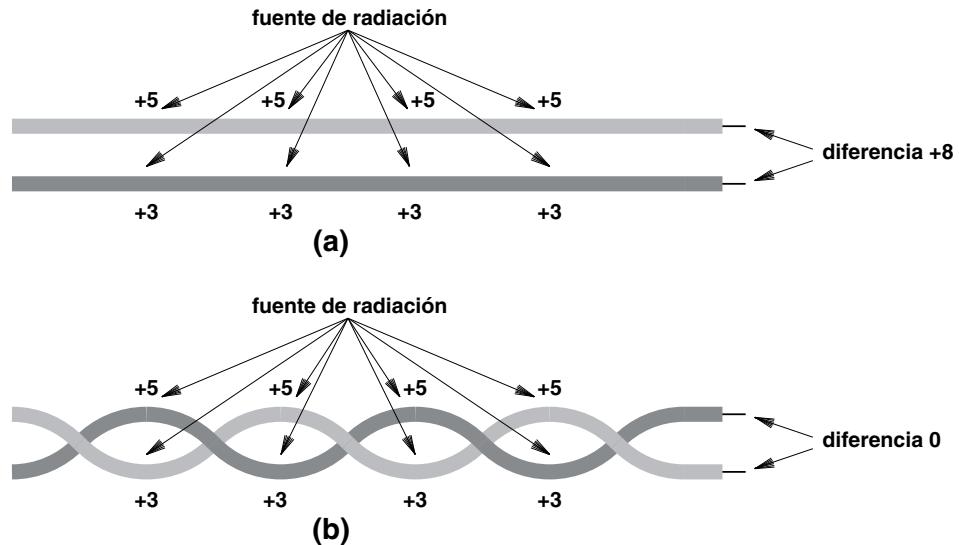


Figura 7.2 Radiación electromagnética no deseada que afecta a (a) dos cables paralelos y (b) un cable de par trenzado.

Como se muestra en la figura, cuando hay dos cables en paralelo, existe una elevada probabilidad de que uno de ellos esté más cerca de la fuente de radiación electromagnética que el otro. De hecho, un cable tiende a actuar como un blindaje que absorbe parte de la radiación electromagnética. Por lo tanto, puesto que está oculto detrás del primer cable, el segundo cable recibe menos energía. En la figura, un total de 32 unidades de radiación golpean cada uno de los dos ejemplos. En la figura 7.2(a), el cable superior absorbe 20 unidades y el inferior absorbe 12, produciendo una diferencia de 8. En la figura 7.2(b), cada uno de los dos cables está sobre el otro la mitad del tiempo, lo que significa que cada cable absorbe la misma cantidad de radiación.

¿Por qué importa la absorción equitativa? La respuesta es que si la interferencia induce exactamente la misma cantidad de energía eléctrica en cada cable, no fluirá corriente adicional. Por lo tanto, la señal original no se perturbará. En conclusión:

[†] En una sección posterior explicamos el término *blindado*.

Para reducir la interferencia provocada por la radiación electromagnética aleatoria, los sistemas de comunicaciones usan el cable de par trenzado en vez de cables paralelos.

7.6 Blindaje: cable coaxial y par trenzado blindado

Aunque es inmune a la mayor parte de la radiación de fondo, el cable de par trenzado no resuelve todos los problemas. Este tipo de cable tiende a tener problemas con:

- Ruido eléctrico muy fuerte
- Una cercanía extrema a la fuente del ruido
- Altas frecuencias utilizadas para la comunicación

Si la intensidad es alta (por ejemplo, en una fábrica que use equipo de soldadura con arco eléctrico) o los cables de comunicación se tienden cerca de la fuente del ruido eléctrico, incluso el par trenzado tal vez no sea suficiente. Por ejemplo, si un par trenzado se tiende sobre el techo de un edificio de oficinas justo sobre un balastro de luz fluorescente, puede haber interferencia. Además, es difícil construir equipos que puedan distinguir entre las señales válidas de frecuencia alta y el ruido, lo que significa que incluso una cantidad pequeña de ruido puede provocar interferencia cuando se usan frecuencias elevadas.

Para manejar situaciones en las que no basta con el par trenzado, hay tipos de cableado que tienen un blindaje metálico adicional. La forma más familiar es el cableado que se utiliza para la televisión por cable. Conocido como *cable coaxial*, el cableado tiene un blindaje metálico grueso, formado a partir de alambres trenzados, el cual rodea por completo un alambre central que transporta las señales. La figura 7.3 ilustra el concepto.

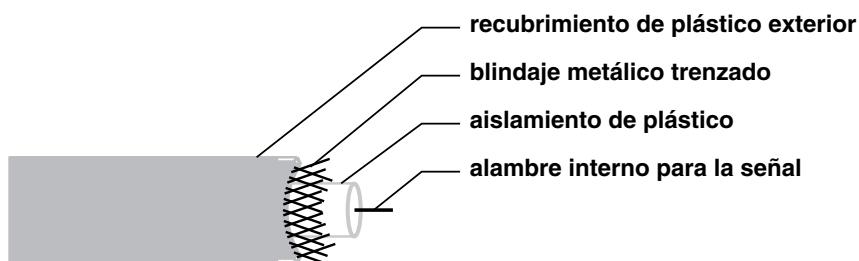


Figura 7.3 Ilustración de cable coaxial con un blindaje que rodea el cable de señal.

El blindaje en un cable coaxial forma un cilindro flexible alrededor del cable interior que proporciona una barrera contra la radiación electromagnética proveniente de cualquier dirección. La barrera también evita que las señales en el alambre interior radien energía electromagnética que pudiera afectar

a otros alambres. En consecuencia, para altas frecuencias se puede colocar un cable coaxial cerca de fuentes de ruido eléctrico y otros cables. En conclusión:

El blindaje grueso y la simetría permiten que el cable coaxial sea inmune al ruido, pueda transportar altas frecuencias y evite que las señales en el cable emitan ruido hacia los cables circundantes.

Al usar cable trenzado en vez de un blindaje metálico sólido el cable coaxial se mantiene flexible, pero el blindaje grueso hace que el cable coaxial sea menos flexible que el cable de par trenzado. Hay otros tipos de blindaje que tienen la siguiente contradicción: el cable es más flexible pero es un poco menos inmune al ruido eléctrico. Una variante común es la que se conoce como *par trenzado blindado* o *STP*. Un cable STP tiene un blindaje metálico más delgado y flexible alrededor de uno o más pares trenzados de alambres. En la mayoría de las versiones del cable STP, el blindaje consiste en una hoja metálica similar a la hoja de papel aluminio que se utiliza en una cocina. El cable STP tiene las ventajas de ser más flexible que un cable coaxial y menos susceptible a la interferencia eléctrica que el *par trenzado sin blindaje* (*UTP*).

7.7 Categorías de cable de par trenzado

En un principio, las compañías telefónicas especificaron estándares para el cableado de par trenzado que se utilizaba en la red telefónica. Hace poco, tres organizaciones de estándares trabajaron en conjunto para desarrollar estándares para los cables de par trenzado que se utilizan en las redes de computadoras. El *Instituto nacional estadounidense de estándares (ANSI)*, la *Asociación de la industria de las telecomunicaciones (TIA)* y la *Alianza de industrias electrónicas (EIA)* crearon una lista de categorías de cableado, con estrictas especificaciones para cada una. La figura 7.4 sintetiza las categorías principales.

Categoría	Descripción	Velocidad de datos (en Mbps)
CAT 1	Par trenzado sin blindaje utilizado para teléfonos	< 0.1
CAT 2	Par trenzado sin blindaje utilizado para datos de T1	2
CAT 3	CAT 2 mejorado, utilizado para redes de computadoras	10
CAT 4	CAT 3 mejorado, utilizado para redes Token Ring	20
CAT 5	Par trenzado sin blindaje, utilizado para redes	100
CAT 5E	CAT 5 extendido para mayor inmunidad al ruido	125
CAT 6	Par trenzado sin blindaje, probado para 200 Mbps	200
CAT 7	Par trenzado con un blindaje de hoja metálica alrededor de todo el cable, más un blindaje alrededor de cada par trenzado	600

Figura 7.4 Categorías de cables de par trenzado y una descripción de cada una.

7.8 Medios que usan energía luminosa y fibra óptica

De acuerdo con la taxonomía de la figura 7.1, tres formas de medios usan energía luminosa para transportar información:

- Fibras ópticas
- Transmisión infrarroja
- Rayos láser de punto a punto

El tipo de medio de transporte de luz más importante es la *fibra óptica*. Cada fibra consiste de una hebra fina de vidrio o plástico transparente encerrado en una cubierta de plástico. Una fibra óptica común se usa para la comunicación en una sola dirección: un extremo de la fibra se conecta a un láser o LED que se utiliza para transmitir luz, y el otro extremo de la fibra se conecta a un dispositivo fotosensible que se utiliza para detectar la luz entrante. Para brindar comunicación de dos vías se utilizan dos fibras, una para transportar la información en cada dirección. Es por ello que generalmente las fibras ópticas se conjuntan en un cable que envuelve una cubierta de plástico a su alrededor; un cable tiene al menos dos fibras, y un cable que se utilice en sitios grandes con varios dispositivos de red puede contener muchas fibras.

Aunque no puede doblarse en un ángulo recto, una fibra óptica es lo bastante flexible como para formar un círculo con un diámetro menor a dos pulgadas sin quebrarse. Surge la pregunta: ¿por qué la luz viaja alrededor de un doblez en la fibra? La respuesta se basa en la física: cuando la luz encuentra el límite entre dos sustancias, su comportamiento depende de la densidad de las dos sustancias y del ángulo en el que la luz golpea el límite. Para un par dado de sustancias existe un *ángulo crítico* (θ) que se mide con respecto a una línea perpendicular al límite. Si el ángulo de incidencia es exactamente igual al ángulo crítico, la luz viaja a lo largo del límite. Cuando el ángulo es menor a θ grados, la luz cruza el límite y se *refracta*, y cuando el ángulo es mayor a θ grados, la luz se refleja como si el límite fuera un espejo. La figura 7.5 ilustra el concepto.

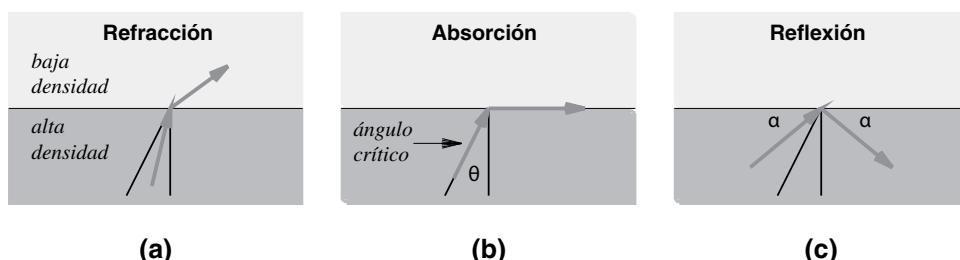


Figura 7.5 Comportamiento de la luz en un límite de densidad cuando el ángulo de incidencia es (a) menor que el ángulo crítico θ , (b) igual al ángulo crítico y (c) mayor que el ángulo crítico.

La figura 7.5(c) explica por qué la luz permanece dentro de una fibra óptica: una sustancia conocida como *revestimiento* se une a la fibra para formar un límite. A medida que viaja, la luz se refleja más allá del límite.

Por desgracia, la reflexión en una fibra óptica no es perfecta. La reflexión absorbe una pequeña cantidad de energía. Además, si un fotón sigue una ruta de zigzag en la que se refleja en las paredes de la fibra muchas veces, el fotón viajará una distancia ligeramente mayor que un fotón que siga una ruta recta. El resultado es que un pulso de luz que se envía en un extremo de una fibra emerge con menos energía y se *dispersa* (es decir, se estira) con el tiempo, como se muestra en la figura 7.6.

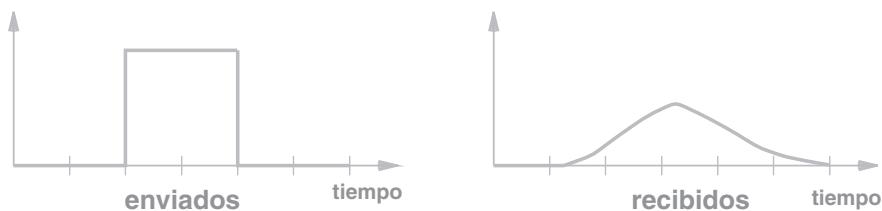


Figura 7.6 Un pulso de luz que se envía y se recibe a través de una fibra óptica.

7.9 Tipos de fibra y transmisión de luz

Aunque no es un problema para las fibras ópticas que se utilizan para conectar una computadora a un dispositivo cercano, la dispersión se vuelve un problema grave para las fibras ópticas muy extensas, como las que se utilizan entre dos ciudades o bajo un océano. En consecuencia, se inventaron tres formas de fibra óptica que permiten elegir entre rendimiento y costo:

- La *fibra multimodal de índice escalonado* es la menos costosa y se usa cuando el rendimiento no es importante. El límite entre la fibra y el revestimiento es abrupto, lo cual provoca que la luz se refleje con frecuencia. Por lo tanto, la dispersión es elevada.
- La *fibra multimodal de índice gradual* es un poco más costosa que la fibra de índice escalonado. Sin embargo, tiene la ventaja de hacer que la densidad de la fibra aumente cerca del borde, lo que reduce la reflexión y disminuye la dispersión.
- La *fibra unimodal* es la más costosa y ofrece la mayor dispersión. Esta fibra tiene un diámetro más pequeño y otras propiedades que ayudan a reducir la reflexión. La fibra unimodal se utiliza para largas distancias y mayores tasas de bits.

La fibra unimodal y el equipo que se utiliza en cada extremo están diseñados para concentrar la luz. Como resultado, un pulso de luz puede viajar miles de kilómetros sin dispersarse. La dispersión mínima ayuda a incrementar la velocidad a la que pueden enviarse los bits, ya que un pulso correspondiente a un bit no se dispersa en el pulso que corresponde a un bit sucesivo.

¿Cómo se envía y recibe la luz en una fibra? La clave es que los dispositivos utilizados para la transmisión deben adaptarse a la fibra. Los mecanismos disponibles son:

- Transmisión: diodo emisor de luz (LED) o diodo de inyección láser (ILD)
- Recepción: celda fotosensible o fotodiodo

En general, los LED y las celdas fotosensibles se usan para distancias cortas y tasas de bits más lentas comunes con la fibra multimodal. La fibra unimodal, que se utiliza en largas distancias con mayores tasas de bits, requiere por lo general de diodos ILD y fotodiodos.

7.10 Comparación entre fibra óptica y cable de cobre

La fibra óptica tiene varias propiedades que la hacen más conveniente que el cableado de cobre: es inmune al ruido eléctrico, tiene mayor ancho de banda y la luz que viaja a través de ella no se atenúa tanto como las señales eléctricas que viajan a través del cobre, sin embargo, sus extremos deben pulirse antes de usarla; por su parte, el cable de cobre es menos costoso y su instalación no requiere tanto equipo especial ni experiencia. Además, al ser más fuerte, el cable de cobre tiene menos probabilidades de romperse si se jala o dobla de manera accidental. La figura 7.7 sintetiza las ventajas de cada tipo de medio.

Fibra óptica
<ul style="list-style-type: none">• Inmune al ruido eléctrico• Menos atenuación de la señal• Mayor ancho de banda
Cableado de cobre
<ul style="list-style-type: none">• Menor costo total• Se requiere menos experiencia/equipo• Se rompe con menos facilidad

Figura 7.7 Ventajas de la fibra óptica y del cableado de cobre.

7.11 Tecnologías de comunicación infrarrojas

Las tecnologías de comunicación *infrarrojas (IR)* usan el mismo tipo de energía que un control remoto de televisión ordinario: una forma de radiación electromagnética que se comporta como la luz visible pero que está fuera del rango visible para el ojo humano. Al igual que la luz visible, la infrarroja se dispersa con rapidez. Las señales infrarrojas pueden reflejarse desde una superficie lisa y dura, y un objeto opaco tan delgado como una hoja de papel puede bloquear la señal, al igual que la humedad en la atmósfera.

En conclusión:

Las tecnologías de comunicación infrarrojas se adaptan mejor para usarse en interiores, en situaciones en las que la ruta entre el emisor y el receptor es corta y sin obstrucción.

La tecnología infrarroja que se utiliza con más frecuencia está diseñada para conectar una computadora a un periférico cercano, como una impresora. Una interfaz en la computadora y una interfaz en la impresora pueden enviar una señal infrarroja que cubre un arco de aproximadamente 30 grados. Siempre y cuando estén alineados los dos dispositivos, cada uno puede recibir la señal del otro. El aspecto inalámbrico de la tecnología infrarroja es muy atractivo para las computadoras tipo laptop, ya que un usuario puede desplazarse por una habitación sin perder el acceso a una impresora. La figura 7.8 enumera las tres tecnologías infrarrojas de uso común, junto con la velocidad de datos que soporta cada una.

Nombre	Expansión	Velocidad
IrDA-SIR	Infrarroja de baja velocidad	0.115 Mbps
IrDA-MIR	Infrarroja de velocidad media	1.150 Mbps
IrDA-FIR	Infrarroja de alta velocidad	4.000 Mbps

Figura 7.8 Tres tecnologías infrarrojas comunes y la velocidad de datos de cada una.

7.12 Comunicación láser de punto a punto

Puesto que conectan un par de dispositivos con un haz que sigue la línea de visión, las tecnologías infrarrojas antes descritas pueden clasificarse como proveedoras de comunicación de *punto a punto*. Además de la infrarroja, existen otras tecnologías de comunicación de punto a punto. Una forma de comunicación de punto a punto usa un haz de luz coherente producido por un *láser*.

Al igual que la tecnología infrarroja, la comunicación por láser sigue la línea de visión y requiere una ruta libre de obstrucciones entre los sitios que se comunican. Sin embargo, a diferencia de un trans-

misor infrarrojo, un haz láser no cubre un área extensa. En su lugar, el haz tiene sólo unos centímetros de ancho. En consecuencia, el equipo emisor y receptor debe alinearse en forma precisa para asegurar que el haz del emisor choque con el sensor en el equipo receptor. En un sistema de comunicaciones común, se necesita la comunicación de dos vías. Por lo mismo, cada lado debe tener tanto un transmisor como un receptor, y ambos transmisores deben alinearse con cuidado. Puesto que la alineación es crucial, es común que el equipo láser de punto a punto se monte de manera permanente.

Los haces láser tienen la ventaja de ser adecuados para usarse en exteriores y pueden abarcar mayores distancias que la tecnología infrarroja. Como resultado, la tecnología láser es especialmente útil en ciudades para transmitir de un edificio a otro. Por ejemplo, imagine una gran corporación con oficinas en dos edificios paralelos. Suponga que la corporación no tiene permitido tender cables que atraviesen la calle que está entre ambos edificios. Sin embargo, puede comprar equipo de comunicación láser y montarlo en forma permanente, ya sea a los lados de los dos edificios o en los techos. Una vez que se compra e instala el equipo, los costos de operación son relativamente bajos.

Para resumir:

La tecnología láser puede usarse para crear un sistema de comunicaciones de punto a punto. Puesto que un láser emite un haz estrecho de luz, el transmisor y el receptor deben estar alineados con precisión; las instalaciones comunes fijan el equipo a una estructura permanente, como el techo de un edificio.

7.13 Comunicación electromagnética (radio)

Recuerde que el término *no guiada* se utiliza para caracterizar a las tecnologías de comunicación que pueden propagar energía sin requerir un medio como un cable o fibra óptica. La forma más común de mecanismos de comunicación no guiados consiste en las tecnologías de redes *inalámbricas* que usan energía electromagnética en el rango de la *radiofrecuencia (RF)*. La transmisión RF tiene una ventaja distinta sobre la luz, ya que la energía RF puede recorrer grandes distancias y penetrar objetos como las paredes de un edificio.

Las propiedades exactas de la energía electromagnética dependen de la frecuencia. Usamos el término *espectro* para referirnos al rango de posibles frecuencias; los gobiernos de todo el mundo asignan frecuencias para propósitos específicos. En Estados Unidos, la *Comisión federal de comunicaciones* establece las reglas en cuanto a la forma de asignar las frecuencias y asigna límites a la cantidad de energía que puede emitir el equipo de comunicación en cada frecuencia. La figura 7.9 muestra el espectro electromagnético en general y las características generales de cada parte. Como muestra la figura, una parte del espectro corresponde a la luz infrarroja antes descrita. El espectro utilizado para las comunicaciones de RF abarca frecuencias de aproximadamente 3 KHz a 300 GHz, e incluye las frecuencias asignadas a las transmisiones de radio y televisión, así como las comunicaciones por satélite y microondas.[†]

[†] Google cuenta con un sistema interesante que muestra la disponibilidad del espectro en diversos puntos en Estados Unidos, en el URL: <https://www.google.com/get/spectrumdatabase/>

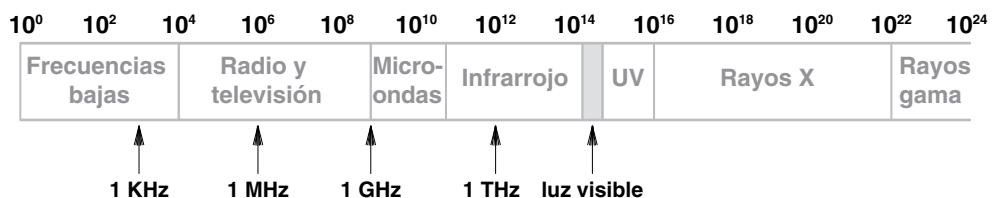


Figura 7.9 Partes principales del espectro electromagnético con la frecuencia en Hz, mostrada en una escala logarítmica.

7.14 Propagación de señales

El capítulo 6 explica que la cantidad de información que puede representar una onda electromagnética depende de su frecuencia. La frecuencia de una onda electromagnética también determina la forma en que se *propaga* esta onda. La figura 7.10 describe los tres extensos tipos de propagación de ondas.

Clasificación	Rango	Tipo de propagación
Frecuencia baja	<2 MHz	La onda sigue la curvatura de la Tierra, pero puede bloquearse por un terreno desnivelado
Frecuencia media	2 a 30 MHz	La onda puede reflejarse por las capas de la atmósfera, especialmente de la ionosfera
Frecuencia alta	> 30 MHz	La onda viaja en una línea directa y será bloqueada por obstrucciones

Figura 7.10 La propagación de una onda electromagnética a varias frecuencias.

De acuerdo con la figura, las frecuencias más bajas de radiación electromagnética siguen la superficie de la Tierra, lo que significa que si el terreno es relativamente plano, será posible colocar un receptor más allá del horizonte de un transmisor. Con frecuencias medias, un transmisor y un receptor pueden estar más alejados ya que la señal puede rebotar por la ionosfera para viajar entre éstos. Por último, las frecuencias más altas de transmisión de radio se comportan como la luz: la señal se propaga en línea recta del transmisor al receptor y la ruta debe estar libre de obstrucciones. En conclusión:

Las frecuencias que se utilizan para las tecnologías de redes inalámbricas no pueden elegirse de manera arbitraria, debido a que los gobiernos controlan el uso del espectro y cada frecuencia tiene características como la propagación de la onda, los requerimientos de potencia y la susceptibilidad al ruido.

Las tecnologías inalámbricas se clasifican en dos amplias categorías:

- *Terrestre*. La comunicación usa equipo como transmisores de radio o microondas, que está relativamente cerca de la superficie de la Tierra. Las ubicaciones comunes de las antenas y demás equipo incluyen las cimas de las colinas, torres hechas por el hombre y edificios altos.
- *No terrestre*. Parte del equipo que se utiliza en las comunicaciones está fuera de la atmósfera de la Tierra (por ejemplo, un satélite que circunda nuestra órbita terrestre).

El capítulo 16 presenta las tecnologías inalámbricas específicas y describe las características de cada una de ellas. Por ahora, basta con entender que la frecuencia y la cantidad de potencia utilizada pueden afectar la velocidad a la que se envían los datos, la distancia máxima a través de la cual puede ocurrir la comunicación y algunas características tales como si la señal puede penetrar o no objetos sólidos.

7.15 Tipos de satélites

Las leyes de la física (en particular la *ley de Kepler*) rigen el movimiento de los objetos que orbitan la Tierra, como es el caso de los satélites. El periodo (es decir, el tiempo requerido para una órbita completa) depende de la distancia a la que se encuentra el satélite de la superficie terrestre. En consecuencia, los satélites de comunicaciones se clasifican en tres amplias categorías, dependiendo de su distancia desde la Tierra. La figura 7.11 enumera las categorías y describe cada una de ellas.

Tipo de órbita	Descripción
Órbita terrestre baja (LEO)	Tiene la ventaja de poco retraso, pero la desventaja es que desde el punto de vista de un observador en la Tierra el satélite parece moverse por el cielo.
Órbita terrestre media (MEO)	Una órbita elíptica (en vez de circular) que se utiliza para brindar comunicación en los polos Norte y Sur. [†]
Órbita terrestre geoestacionaria (GEO)	Tiene la ventaja de que el satélite permanece en una posición fija con respecto a una ubicación en la superficie terrestre, pero tiene la desventaja de estar más alejado.

Figura 7.11 Las tres categorías básicas de satélites de comunicación.

[†] En 2013, una compañía comercial (O3b) anunció que crearía el primer grupo de satélites MEO con la intención de proveer servicio de Internet a la población que actualmente no cuenta con ese servicio (los otros tres mil millones).

7.16 Satélites en órbita terrestre geoestacionaria (GEO)

Como lo explica la figura 7.11, la principal concesión en los satélites de comunicación está entre la altura y el periodo orbital. La principal ventaja de un satélite en órbita *terrestre geoestacionaria (GEO)* surge debido a que el periodo de órbita es exactamente igual a la velocidad a la que gira la Tierra. Si se posiciona sobre el ecuador, un satélite GEO permanece en todo momento en la misma ubicación exacta sobre la superficie de la Tierra. Una posición de satélite estacionaria significa que, una vez que se alinee una *estación terrestre* con el satélite, el equipo nunca tendrá que moverse. La figura 7.12 ilustra el concepto.

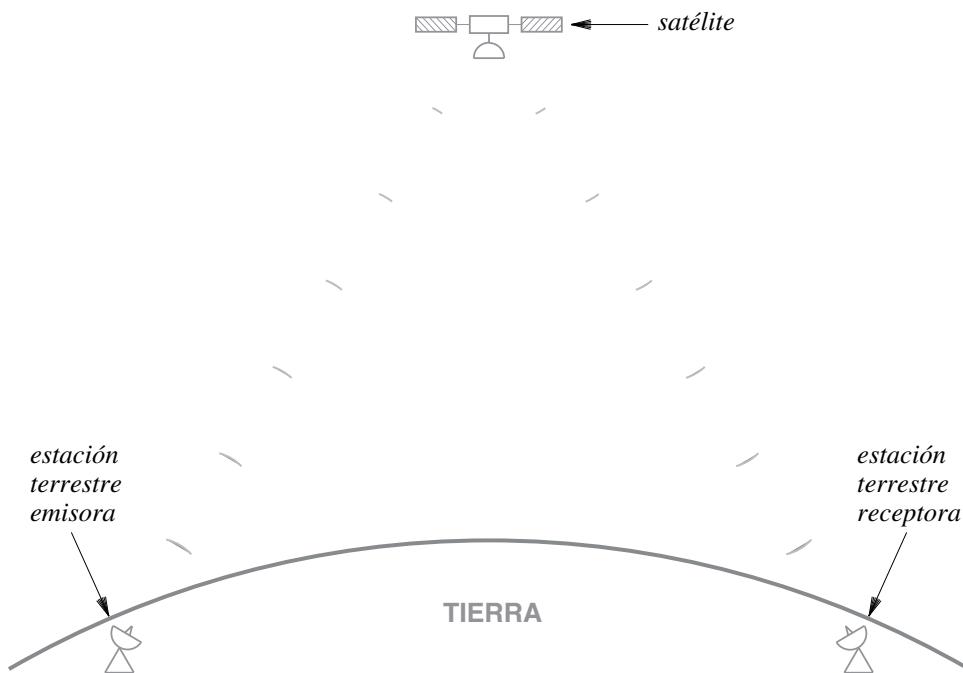


Figura 7.12 Un satélite GEO y las estaciones terrestres alineados de manera permanente.

Por desgracia, la distancia requerida para una órbita geoestacionaria es de 33,785 kilómetros o 22,236 millas, lo cual es aproximadamente una décima parte de la distancia que hay hasta la Luna. Para entender lo que significa dicha distancia para la comunicación, considere una onda de radio que viaja hacia un satélite GEO y regresa. A la velocidad de la luz, de 3×10^8 metros por segundo, el viaje tarda:

$$\frac{2 \times 35.8 \times 10^6 \text{ metros}}{3 \times 10^8 \text{ metros/seg}} = 0.238 \text{ seg} \quad (7.1)$$

Aunque podría parecer que no es importante, un retraso aproximado de 0.2 segundos puede ser considerable para algunas aplicaciones. En una llamada telefónica o teleconferencia de video, un humano puede detectar un retraso de 0.2 segundos. Para las transacciones electrónicas como una bolsa de valores que ofrezca un conjunto limitado de bonos, retrasar una oferta por 0.2 segundos puede significar la diferencia entre una oferta exitosa y no exitosa. Para resumir:

Incluso a la velocidad de la luz, una señal tarda más de 0.2 segundos para viajar desde una estación terrestre hasta un satélite GEO y regresar a otra estación terrestre.

7.17 Cobertura GEO de la Tierra

¿Cuántos satélites de comunicación GEO pueden existir? Lo interesante es que hay una cantidad limitada de “espacio” disponible en la órbita geosincrónica del ecuador, ya que los satélites de comunicación que usan una frecuencia determinada deben estar separados unos de otros para evitar la interferencia. La separación mínima depende del poder de los transmisores, pero puede requerir una separación angular de entre 4 y 8 grados. De esta forma, sin ningún ajuste adicional, la circunferencia de 360 grados sobre el ecuador sólo puede contener de 45 a 90 satélites.

¿Cuál es el número mínimo de satélites necesarios para cubrir la Tierra? Tres. Para ver por qué, analice la figura 7.13 que muestra a la Tierra con tres satélites GEO posicionados alrededor del ecuador con una separación de 120°. La figura ilustra cómo es que las señales de los tres satélites cubren la circunferencia. En la figura, el tamaño de la Tierra y la distancia de los satélites están dibujados a escala.

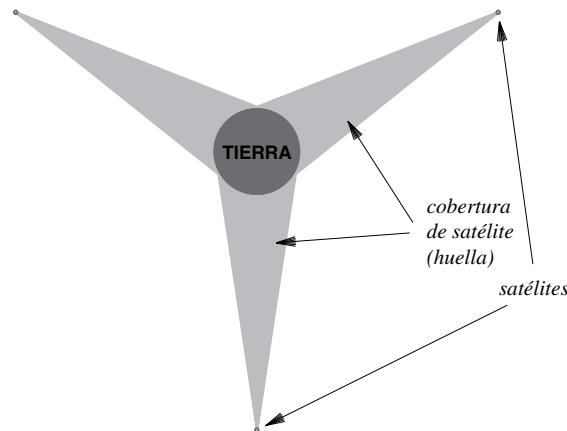


Figura 7.13 Las señales de tres satélites GEO son suficientes para cubrir la Tierra.

7.18 Satélites en órbita terrestre baja (LEO) y grupos de satélites

Para la comunicación, la alternativa primaria de GEO se conoce como órbita *terrestre baja (LEO)*, la cual se define con altitudes de hasta 2000 kilómetros. Por una cuestión práctica, un satélite debe colocarse sobre el margen de la atmósfera para evitar el arrastre producido por el enfrentamiento de gases. Por lo tanto, los satélites LEO se colocan comúnmente en altitudes de 500 kilómetros o más. LEO ofrece la ventaja de retrasos cortos (por lo general de 1 a 4 milisegundos), pero tiene la desventaja que la órbita de un satélite no coincide con la rotación de la Tierra. Por consiguiente, desde el punto de vista de un observador en la Tierra, un satélite LEO parece moverse por el cielo, lo que significa que una estación terrestre debe tener una antena que pueda girar para rastrearlo. El rastreo es difícil, ya que los satélites se mueven con rapidez. Los satélites LEO de menor altitud orbitan la Tierra en aproximadamente 90 minutos; los satélites LEO de mayor altitud requieren de varias horas.

La técnica general que se utiliza con los satélites LEO se conoce como *clúster* o *despliegue*. Se puede diseñar un grupo grande de satélites LEO para trabajar en conjunto. Además de comunicarse con estaciones terrestres, un satélite también puede comunicarse con otros satélites de su mismo grupo. Los miembros del grupo permanecen comunicados y aceptan reenviar los mensajes según sea necesario. Por ejemplo, considere lo que ocurre cuando un usuario en Europa envía un mensaje a un usuario en Norteamérica. Una estación terrestre que se encuentra en Europa transmite el mensaje al satélite que en ese momento está por encima de ésta. Los satélites se comunican para reenviar el mensaje al satélite del grupo que se encuentre en ese momento sobre una estación terrestre de Norteamérica. Por último, el satélite que está sobre Norteamérica transmite el mensaje a una estación terrestre. Para resumir:

Un clúster o grupo de satélites LEO trabajan juntos para reenviar mensajes. Los miembros del grupo deben saber qué satélite se encuentra actualmente sobre cierta área de la Tierra, y reenviar los mensajes al miembro mejor posicionado para que lo transmita a una estación terrestre.

7.19 Ventajas y desventajas entre los tipos de medios

La elección del medio es algo complejo e implica la evaluación de varios factores. Los elementos a considerar son:

- Costo: materiales, instalación, operación y mantenimiento
- Velocidad de datos: número de bits por segundo que pueden enviarse
- Retraso: tiempo requerido para propagar o procesar la señal
- Efecto sobre la señal: atenuación y distorsión
- Entorno: susceptibilidad a interferencia y ruido eléctrico
- Seguridad: susceptibilidad al espionaje

7.20 Medición de los medios de transmisión

Ya mencionamos las dos medidas más importantes de rendimiento que se utilizan para evaluar un medio de transmisión:

- *Retraso de propagación*: el tiempo requerido para que una señal recorra el medio
- *Capacidad de canal*: la máxima velocidad de datos que el medio puede soportar

El capítulo 6 explica que en 1920 un investigador llamado Nyquist descubrió una relación fundamental entre el ancho de banda de un sistema de transmisión y su capacidad de transferir datos. Conocida como el *teorema de Nyquist*, la relación ofrece un soporte teórico sobre la velocidad máxima a la que pueden enviarse los datos sin considerar el efecto del ruido. Si un sistema de transmisión usa K posibles niveles de señal y tiene un ancho de banda B , el teorema de Nyquist establece que la velocidad máxima de datos en bits por segundo, o D , es:

$$D = 2 B \log_2 K \quad (7.2)$$

7.21 El efecto del ruido en la comunicación

El teorema de Nyquist proporciona un máximo absoluto que no puede obtenerse en la práctica. En especial, los ingenieros han observado que un sistema de comunicaciones real está sujeto a pequeñas cantidades de *ruido* eléctrico y que dicho ruido hace imposible lograr la máxima tasa de transmisión teórica. En 1948, Claude Shannon extendió la obra de Nyquist para especificar la velocidad máxima de datos que podría obtenerse a través de un sistema de transmisión que experimente ruido. El resultado, conocido como *teorema de Shannon*,[†] puede indicarse como:

$$C = B \log_2(1 + S/N) \quad (7.3)$$

donde C es el límite efectivo sobre la capacidad del canal en bits por segundo, B es el ancho de banda del hardware y S/N es la *relación entre la señal y el ruido*, la relación de la potencia promedio de la señal dividida entre la potencia promedio del ruido.

Como ejemplo del teorema de Shannon, considere un medio de transmisión que tiene un ancho de banda de 1 KHz, una potencia promedio de la señal de 70 unidades y una potencia promedio del ruido de 10 unidades. La capacidad del canal es de:

$$C = 10^3 \times \log_2(1 + 7) = 10^3 \times 3 = 3,000 \text{ bits por segundo}$$

[†] El resultado también se conoce como *ley de Shannon-Hartley*.

A menudo la relación señal-ruido se proporciona en *decibeles* (se abrevian como *dB*), donde un decibel se define como una medida de la diferencia entre dos niveles de potencia. La figura 7.14 ilustra la medición.

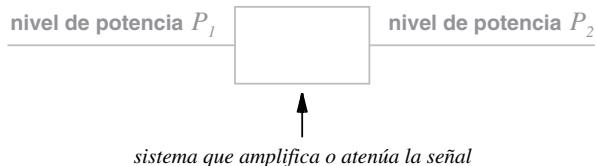


Figura 7.14 Niveles medidos de potencia en cualquier lado de un sistema.

Una vez que se miden dos niveles de potencia, la diferencia se expresa en decibeles y se define así:

$$dB = 10 \log_{10} \left[\frac{P_2}{P_1} \right] \quad (7.4)$$

Usar dB como medición puede parecer algo normal, pero tiene dos ventajas interesantes. Primero, un valor negativo de dB significa que la señal se *atenuó* o redujo y un valor positivo de dB significa que la señal se *amplificó*. Segundo, si un sistema de comunicaciones tiene varias partes ordenadas en una secuencia, pueden sumarse las medidas en decibeles de cada parte para producir una medida general del sistema.

El sistema telefónico de voz tiene una relación señal-ruido aproximada de 30 dB y un ancho de banda de aproximadamente 3000 Hz. Para convertir los dB de la relación señal-ruido en una simple fracción, divida entre 10 y use el resultado como una potencia de 10 (por ejemplo, $30/10 = 3$ y $10^3 = 1000$, por lo que la relación señal-ruido es de 1000). El teorema de Shannon puede aplicarse para determinar el número máximo de bits por segundo que pueden transmitirse a través de la red telefónica:

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

o aproximadamente 30,000 bps. Los ingenieros reconocen esto como un límite fundamental: sólo será posible lograr velocidades de transmisión mayores si puede mejorarse la relación señal-ruido.

7.22 El significado de la capacidad de un canal

Los teoremas de Nyquist y de Shannon descritos anteriormente, tienen consecuencias para los ingenieros que diseñan redes de comunicaciones de datos. El trabajo de Nyquist ha proporcionado un incentivo para explorar formas complejas de codificar los bits en las señales:

El teorema de Nyquist anima a los ingenieros a explorar nuevas formas de codificar los bits de una señal, ya que una codificación inteligente permite transmitir más bits por unidad de tiempo.

En cierto sentido, el teorema de Shannon es más básico, ya que representa un límite absoluto derivado de las leyes de la física. Por ejemplo, gran parte del ruido de una línea de transmisión puede atribuirse a la radiación que existe en el universo debido a la explosión conocida como Big Bang. Por lo tanto:

El teorema de Shannon informa a los ingenieros que ninguna cantidad de codificación inteligente puede sobreponer las leyes de la física que imponen un límite fundamental sobre el número de bits por segundo que pueden transmitirse en un sistema de comunicaciones real.

7.23 Resumen

Existe una variedad de medios de transmisión que pueden clasificarse como guiados y no guiados, o dividirse de acuerdo con la forma de energía que utilizan (eléctrica, luminosa o transmisión de radio). La energía eléctrica se utiliza a través de alambres y cables. Para protegerse contra la interferencia eléctrica, el cableado de cobre puede consistir en pares trenzados o puede estar recubierto por un blindaje.

La energía luminosa puede usarse a través de la fibra óptica o para la comunicación de punto a punto, usando rayos infrarrojos o láser. Como se refleja en los límites entre la fibra y el revestimiento, la luz permanece en una fibra óptica siempre y cuando el ángulo de incidencia sea mayor que el ángulo crítico. Al pasar a través de una fibra, un pulso de luz se dispersa; la dispersión es mayor en la fibra multimodal y menor en la fibra unimodal. La fibra unimodal es más costosa.

La comunicación inalámbrica usa energía electromagnética. La frecuencia utilizada determina el ancho de banda y el comportamiento de propagación; las frecuencias bajas siguen la superficie de la Tierra, las frecuencias mayores se reflejan de la ionosfera y las frecuencias más altas se comportan como la luz visible al requerir una ruta directa sin obstrucciones del transmisor al receptor.

La principal tecnología de comunicación no terrestre se basa en los satélites. La órbita de un satélite GEO coincide con la rotación de la Tierra, pero la elevada altitud incurre en un retraso que se mide en décimas de segundos. Los satélites LEO tienen poco retraso y se desplazan rápidamente por el cielo; se usan grupos o clústeres para transmitir los mensajes.

El teorema de Nyquist proporciona un soporte teórico sobre la capacidad de los medios de transmisión cuando no hay ruido presente; el teorema de Shannon especifica la capacidad de un canal en situaciones realistas donde hay ruido presente. La relación señal-ruido, un término en el teorema de Shannon, se mide comúnmente en decibeles.

EJERCICIOS

- 7.1** ¿Cuál es la diferencia entre transmisión guiada y no guiada?
- 7.2** ¿Cuáles son los tres tipos de energía que se utilizan al clasificar los medios físicos, de acuerdo con la energía utilizada?
- 7.3** ¿Qué ocurre cuando el ruido se encuentra con un objeto metálico?
- 7.4** ¿Cuáles son los tres tipos de cableado que se utilizan para reducir la interferencia del ruido?
- 7.5** Explique cómo es que el cable de par trenzado reduce el efecto del ruido.
- 7.6** Dibuje un diagrama que ilustre la sección transversal de un cable coaxial.
- 7.7** Si va a instalar el cableado de una red de computadoras en una nueva casa, ¿qué categoría de cable de par trenzado elegiría? ¿Por qué?
- 7.8** Explique por qué la luz no sale de una fibra óptica cuando ésta se dobla en un arco.
- 7.9** ¿Qué es la dispersión?
- 7.10** Enliste las tres formas de fibra óptica y proporcione las propiedades generales de cada una.
- 7.11** ¿Qué fuentes de luz y sensores se usan con las fibras ópticas?
- 7.12** ¿Cuál es la principal desventaja de la fibra óptica en comparación con el cable de cobre?
- 7.13** ¿Cuál es el ángulo cónico aproximado que puede usarse con la tecnología infrarroja?
- 7.14** ¿Puede usarse la comunicación por láser desde un vehículo en movimiento? Explique.
- 7.15** ¿Por qué podría usarse la radiación electromagnética de baja frecuencia para las comunicaciones? Explique.
- 7.16** ¿Cuáles son las dos categorías amplias de comunicaciones inalámbricas?
- 7.17** Enliste los tres tipos de satélites de comunicaciones y proporcione las características de cada uno.
- 7.18** Si se envían mensajes de Europa a Estados Unidos usando un satélite GEO, ¿cuánto tiempo se requerirá para enviar un mensaje y recibir una respuesta?
- 7.19** ¿Cuántos satélites GEO se necesitan para llegar a todas las áreas pobladas de la Tierra?
- 7.20** ¿Qué es el retraso de propagación?
- 7.21** ¿Cuál es la relación entre el ancho de banda, los niveles de señal y la velocidad de datos?
- 7.22** Si se usan dos niveles de señal, ¿cuál es la velocidad de datos que puede enviarse a través de un cable coaxial que tenga un ancho de banda de 6.2 MHz?
- 7.23** Si un sistema tiene un nivel de potencia promedio de 100, un nivel de ruido promedio de 33.33 y un ancho de banda de 100 MHz, ¿cuál es el límite efectivo en la capacidad del canal?
- 7.24** Si un sistema tiene un nivel de potencia de entrada de 9000 y un nivel de potencia de salida de 3000, ¿cuál es la diferencia cuando se expresa en dB?
- 7.25** Si puede crearse un sistema telefónico con una relación de señal a ruido de 40 dB y un ancho de banda analógico de 3000 Hz, ¿cuántos bits por segundo podrían transmitirse?

Contenido del capítulo

- 8.1 Introducción, 135
- 8.2 Las tres principales fuentes de errores de transmisión, 135
- 8.3 Efecto de los errores de transmisión sobre los datos, 136
- 8.4 Dos estrategias para manejar los errores de canal, 137
- 8.5 Códigos de errores de bloque y convolucionales, 138
- 8.6 Ejemplo de un código de error de bloque: comprobación de paridad simple, 139
- 8.7 Las matemáticas de los códigos de errores de bloque y la notación (n,k) , 140
- 8.8 Distancia de Hamming: la medición de la fuerza de un código, 140
- 8.9 La distancia de Hamming entre cadenas de un libro de códigos, 141
- 8.10 Concesión entre la detección de errores y la sobrecarga, 142
- 8.11 Corrección de errores con paridad de fila y de columna (RAC), 142
- 8.12 La suma de verificación de 16 bits que se utiliza en Internet, 144
- 8.13 Códigos de redundancia cíclica (CRC), 145
- 8.14 Una implementación eficiente de hardware de CRC, 148
- 8.15 Mecanismos de solicitud de repetición automática (ARQ), 148
- 8.16 Resumen, 149

8

Confiabilidad y codificación de canales

8.1 Introducción

Cada uno de los capítulos de esta parte del libro representa un aspecto de las comunicaciones de datos, que son la base de todas las redes de computadoras. En el capítulo anterior hablamos sobre los medios de transmisión y señalamos el problema del ruido electromagnético. En este capítulo continuaremos la explicación analizando los errores que pueden ocurrir durante la transmisión y las técnicas que pueden usarse para controlarlos.

Los conceptos que se presentan aquí son fundamentales para las redes de computadoras y se usan en muchas capas de la pila de los protocolos de comunicación. En especial, las metodologías y técnicas para el control de errores aparecen a lo largo de los protocolos de Internet que se describen en la cuarta parte del libro.

8.2 Las tres principales fuentes de errores de transmisión

Todos los sistemas de comunicaciones de datos son susceptibles de errores. Algunos de los problemas son propios de la física del universo, y otros son resultado ya sea de dispositivos que fallan o de equipo que no cumple con los estándares de ingeniería. Las pruebas extensas pueden eliminar muchos de los problemas que surgen de una mala ingeniería, y el monitoreo cuidadoso puede identificar un equipo que falla. Sin embargo, los pequeños errores que ocurren durante la transmisión son más difíciles de detectar que las fallas mayores, y gran parte de la tecnología de redes de computadoras se concentra en métodos para controlar y recuperarse de dichos errores. Hay tres categorías principales de errores de transmisión:

- *Interferencia.* Como se explica en el capítulo 7, la radiación electromagnética que se emite de dispositivos como motores eléctricos y la radiación cósmica de fondo provocan ruido que puede perturbar las transmisiones de radio y las señales que viajan a través de cables.
- *Distorsión.* Todos los sistemas físicos distorsionan las señales. A medida que un pulso viaja a lo largo de una fibra óptica, el pulso se dispersa. Los alambres tienen propiedades de capacitancia e inductancia que bloquean las señales en ciertas frecuencias, mientras que admiten señales en otras frecuencias. Con sólo colocar un alambre cerca de un objeto metálico grande, es posible modificar el conjunto de frecuencias que pueden pasar a través del mismo. De manera similar, los objetos metálicos pueden bloquear ciertas frecuencias de ondas de radio, mientras que dejan pasar otras.
- *Atenuación.* A medida que una señal pasa a través de un medio, se vuelve más débil. Los ingenieros dicen que la señal se *atenúa*. Por consecuencia, las señales en los alambres o fibras ópticas se debilitan al recorrer grandes distancias, al igual que una señal de radio se debilita con la distancia.

El teorema de Shannon sugiere una forma de reducir los errores: incrementar la relación señal-ruido (ya sea aumentando la señal o reduciendo el ruido). Aun cuando los mecanismos como los cables blindados pueden ayudar a reducir el ruido, un sistema de transmisión físico siempre es susceptible de errores, por lo que tal vez no sea posible incrementar la relación señal-ruido.

Aunque los errores no pueden eliminarse por completo, muchos errores de transmisión pueden detectarse. En algunos casos, los errores pueden corregirse de manera automática. Más adelante veremos que la detección de errores agrega sobrecarga. Por consiguiente, todo el manejo de errores es una concesión en la que un diseñador de sistemas tiene que decidir si es probable que ocurra cierto error y, de ser así, cuáles serán las consecuencias (por ejemplo, un error de un solo bit en una transferencia bancaria puede hacer una diferencia de más de un millón de dólares, pero un error de un bit en una imagen es menos importante). En conclusión:

Aunque los errores de transmisión son inevitables, los mecanismos de detección de errores agregan sobrecarga. Por lo tanto, un diseñador debe elegir exactamente qué mecanismos de detección y compensación de errores se utilizarán.

8.3 Efecto de los errores de transmisión sobre los datos

En vez de examinar la física y la causa exacta de los errores de transmisión, las comunicaciones de datos se enfocan en el efecto que tienen estos errores sobre los datos. La figura 8.1 enlista las tres principales formas en que los errores de transmisión afectan los datos.

Aunque cualquier error de transmisión puede provocar cada uno de los posibles errores de datos, la figura señala que a menudo un error de transmisión se manifiesta a sí mismo como un error de datos específico. Por ejemplo, la interferencia de una duración extremadamente corta, conocida como *pico*, es a menudo la causa de un error de un solo bit. Una interferencia o distorsión de mayor duración puede producir ráfagas de errores. Algunas veces no está claro si una señal es 1 o 0, sino que cae en una región ambigua, lo que se conoce como *supresión*.

Tipo de error	Descripción
Error de un solo bit	Un solo bit en un bloque de bits cambia y todos los demás bits en el bloque no cambian (a menudo como resultado de una interferencia de muy corta duración)
Ráfaga de errores	Varios bits en un bloque de bits cambian (a menudo como resultado de una interferencia de mayor duración)
Supresión (ambigüedad)	La señal que llega a un receptor es ambigua y no queda claro si corresponde a un 1 o a un 0 (puede ser resultado de distorsión o interferencia)

Figura 8.1 Los tres tipos de errores de datos en un sistema de comunicaciones de datos.

En una ráfaga de errores, el *tamaño de la ráfaga* (o *longitud*) se define como el número de bits que transcurrieron desde el inicio hasta el final de la corrupción. La figura 8.2 ilustra la definición.

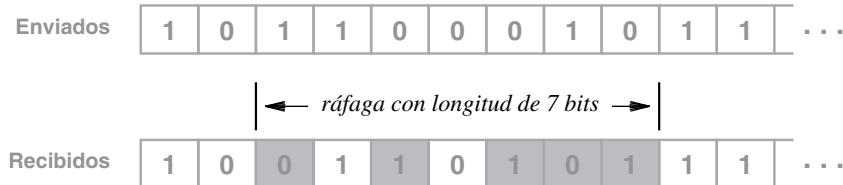


Figura 8.2 Ilustración de una ráfaga de errores con los bits modificados marcados en gris.

8.4 Dos estrategias para manejar los errores de canal

Se han desarrollado una variedad de técnicas matemáticas que solucionan los errores de datos y aumentan la confiabilidad. Conocidas en forma colectiva como *codificación de canal*, las técnicas pueden dividirse en dos amplias categorías:

- Mecanismos de corrección de errores en la recepción (FEC)
- Mecanismos de solicitud de repetición automática (ARQ)

La idea básica de la corrección de errores en la recepción es simple: agregar información adicional a los datos que permita a un receptor verificar que éstos lleguen correctamente y de ser posible corregir los errores. La figura 8.3 ilustra la organización conceptual de un mecanismo de corrección de errores en la recepción.

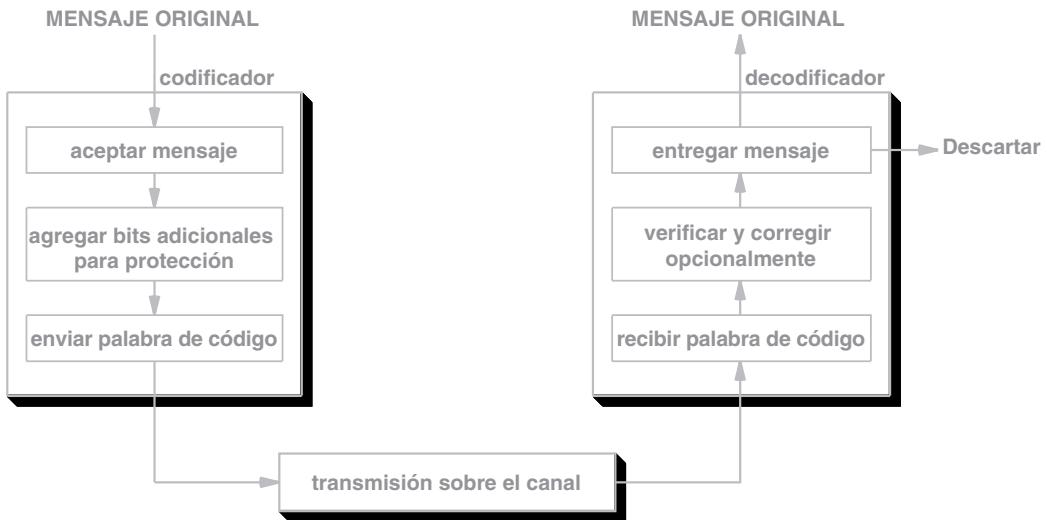


Figura 8.3 La organización conceptual de un mecanismo de corrección de errores en la recepción.

Los *mecanismos básicos de detección de errores* permiten a un receptor detectar cuando ocurre un error; los mecanismos de corrección de errores en la recepción permiten a un receptor determinar exactamente cuáles bits cambiaron y calcular los valores correctos. La segunda metodología para la codificación de canal, conocida como ARQ,[†] requiere la cooperación de un emisor; un emisor y un receptor intercambian mensajes para asegurar que todos los datos lleguen en forma correcta.

8.5 Códigos de errores de bloque y convolucionales

Los dos tipos de técnicas de corrección de errores en la recepción son:

- *Códigos de errores de bloque.* Un código de bloque divide los datos a enviar en un conjunto de bloques y agrega información adicional conocida como *redundancia* a cada bloque. La codificación de un bloque dado de bits depende sólo de esos mismos bits, y no de los bits que se enviaron antes. Los códigos de errores de bloque *no tienen memoria* en cuanto a que el mecanismo de codificación no pasa la información del estado de un bloque de datos al siguiente.
- *Códigos de errores convolucionales.* Un código convolucional trata los datos como una serie de bits y calcula un código a través de una serie continua. Por consiguiente, el código calculado para un conjunto de bits depende de la entrada actual y de algunos de los bits anteriores del flujo. Se dice que los códigos convolucionales son códigos *con memoria*.

[†] En la sección 8.15 presentaremos el concepto de ARQ.

Cuando se implementan en el software, los códigos de errores convolucionales requieren generalmente más cálculos que los códigos de errores de bloque. Sin embargo, los códigos convolucionales tienen comúnmente una mayor probabilidad de detectar problemas.

8.6 Ejemplo de un código de error de bloque: comprobación de paridad simple

Para comprender cómo puede usarse la información adicional para detectar errores, considere un mecanismo de *comprobación de paridad simple (SPC)*. Una forma de SPC define a un bloque como una unidad de datos de 8 bits (es decir, un solo byte). Del lado emisor, un codificador agrega un bit adicional a cada byte, conocido como *bit de paridad*, antes de la transmisión; un receptor elimina el bit de paridad y lo usa para verificar si los bits en el byte son correctos.

Antes de poder usar la paridad, el emisor y el receptor deben estar configurados para *paridad par* o *paridad impar*. Cuando se utiliza paridad par, el emisor selecciona un bit de paridad 0 si el byte tiene un número par de bits 1, o 1 si el byte tiene un número impar de bits 1. La forma de recordar la definición es: la paridad par o impar especifica si los 9 bits enviados a través de un canal tienen un número par o impar de bits 1. La figura 8.4 enumera ejemplos de bytes de datos y el valor del bit de paridad que se envía al usar paridad par o impar.

Para resumir:

La comprobación de paridad simple (SPC) es una forma básica de codificación de canal en la que un emisor agrega un bit adicional a cada byte para generar un número par (o impar) de bits 1 y un receptor verifica que los datos entrantes tengan el número correcto de bits 1.

Datos originales	Paridad par	Paridad impar
0 0 0 0 0 0 0 0	0	1
0 1 0 1 1 0 1 1	1	0
0 1 0 1 0 1 0 1	0	1
1 1 1 1 1 1 1 1	0	1
1 0 0 0 0 0 0 0	1	0
0 1 0 0 1 0 0 1	1	0

Figura 8.4 Los bytes de datos y el valor correspondiente de un bit de paridad simple cuando se usa paridad par o impar.

La comprobación de paridad simple es una forma débil de codificación de canal que puede detectar errores, pero no puede corregirlos. Además, los mecanismos de paridad sólo pueden manejar errores cuando se cambia un número impar de bits. Si uno de los nueve bits (incluyendo el bit de paridad) cambia durante la transmisión, el receptor declarará que el byte entrante es inválido.

No obstante, si ocurre una ráfaga de errores en la que dos, cuatro, seis u ocho bits cambien de valor, el receptor clasificará equivocadamente como válido el byte entrante.

8.7 Las matemáticas de los códigos de errores de bloque y la notación (n,k)

Observe que la corrección de errores en la recepción toma como entrada un conjunto de mensajes e inserta bits adicionales para producir una versión codificada. Matemáticamente, definimos el conjunto de todos los mensajes posibles como un conjunto de *palabras de código*. Si una palabra de datos contiene k bits y se agregan r bits adicionales para formar una palabra de código, decimos que el resultado es un

esquema de codificación (n,k)

donde $n = k + r$. La clave de la detección exitosa de errores reside en elegir un subconjunto de las 2^n posibles combinaciones que son palabras de código válidas. El subconjunto válido se conoce como *libro de códigos*.

Como ejemplo, considere la comprobación de paridad simple. El conjunto de palabras de datos consiste en cualquier combinación posible de ocho bits. Por lo tanto, $k = 8$ y hay 2^8 o 256 posibles palabras de datos. Los datos enviados consisten en $n = 9$ bits, por lo que hay 2^9 o 512 posibilidades. Sin embargo, sólo la mitad de los 512 valores forman palabras de código válidas.

Piense en el conjunto de todos los posibles valores de n bits y el subconjunto válido que forma el libro de códigos. Si ocurre un error durante la transmisión, uno o más de los bits de una palabra de código cambiarán, lo que producirá otra palabra de código válida o una combinación inválida. Por ejemplo, en el esquema de paridad simple que describimos antes, un cambio de un solo bit en una palabra de código válida produce una combinación inválida, pero al cambiar dos bits se produce otra palabra de código válida. Es obvio que deseamos una codificación en la que un error produzca una combinación inválida. Para generalizar:

Un esquema de codificación de canal ideal es uno en el que cualquier cambio en los bits de una palabra de código válida produce una combinación inválida.

8.8 Distancia de Hamming: la medición de la fuerza de un código

Ningún esquema de codificación de canal es ideal: si se cambian los suficientes bits siempre se transformará en una palabra de código válida. Entonces, para un esquema práctico, tenemos la siguiente cuestión: ¿cuál es el mínimo número de bits de una palabra de código válida que deben cambiarse para producir otra palabra de código válida?

Para responder a la pregunta los ingenieros usan una medida que se conoce como *distancia de Hamming*, nombrada en honor de un teórico de Bell Laboratories que fue pionero en el campo de la teoría de la información y la codificación de canales. Dadas dos cadenas de n bits cada una, la distancia de Hamming se define como el número de diferencias (es decir, el número de bits que deben cambiarse para transformar una cadena de bits en la siguiente). La figura 8.5 ilustra la definición.

$d(000,001) = 1$	$d(000,101) = 2$
$d(101,100) = 1$	$d(001,010) = 2$
$d(110,001) = 3$	$d(111,000) = 3$

Figura 8.5 Ejemplos de la distancia de Hamming para varios pares de cadenas de 3 bits.

Una forma de calcular la distancia de Hamming consiste en aplicar la disyunción exclusiva conocida como *o exclusivo* (*xor*) entre dos cadenas y contar el número de bits 1 en la respuesta. Por ejemplo, considere la distancia de Hamming entre las cadenas 110 y 011. El *xor* de las dos cadenas es:

$$\begin{array}{r} 110 \\ + 011 \\ \hline 101 \end{array}$$

que contiene dos bits 1. Por lo tanto, la distancia de Hamming entre 011 y 101 es 2.

8.9 La distancia de Hamming entre cadenas de un libro de códigos

Recuerde que nos interesa saber si los errores pueden transformar una palabra de código válida en otra palabra de código válida. Para medir dichas transformaciones, calculamos la distancia de Hamming entre todos los pares de palabras de código que se encuentran en un libro de códigos dado. Como un ejemplo simple, considere que se aplica una paridad impar a palabras de datos de 2 bits. La figura 8.6 enumera las cuatro posibles palabras de datos, las cuatro posibles palabras de código que resultan de adjuntar un bit de paridad y las distancias de Hamming para los pares de palabras de código.

Palabra de datos	Palabra de código
0 0	0 0 1
0 1	0 1 0
1 0	1 0 0
1 1	1 1 1

(a)

$d(001,010) = 2$	$d(010,100) = 2$
$d(001,100) = 2$	$d(010,111) = 2$
$d(001,111) = 2$	$d(100,111) = 2$

(b)

Figura 8.6 (a) Las palabras de datos y palabras de código para una codificación de paridad simple de dos cadenas de datos de 2 bits, usando paridad impar y (b) la distancia de Hamming para todos los pares de palabras de código.

Un conjunto completo de palabras de código se conoce como *libro de códigos*. Usamos d_{\min} para denotar la *mínima distancia de Hamming* entre los pares de un libro de códigos. El concepto da una respuesta precisa a la pregunta de cuántos errores de bits pueden provocar una transformación de una palabra de código válida en otra palabra de código válida. En el ejemplo de paridad simple de la figura 8.6, el conjunto consiste en la distancia de Hamming entre cada par de palabras de código y $d_{\min} = 2$. La definición significa que hay al menos una palabra de código válida que puede transformarse en otra palabra de código válida si ocurren dos errores de bits durante la transmisión. En conclusión:

Para encontrar el número mínimo de cambios de bits que pueden transformar una palabra de código válida en otra palabra de código válida, hay que calcular la distancia de Hamming mínima entre todos los pares del libro de códigos.

8.10 Concesión entre la detección de errores y la sobrecarga

Para un conjunto de palabras de código, es conveniente un valor grande de d_{\min} , ya que el código es inmune a un mayor número de errores de bits; si cambian menos de d_{\min} bits, el código puede detectar que ocurrieron uno o más errores. La ecuación (8.1) especifica la relación entre d_{\min} y e , el máximo número de errores de bits que pueden detectarse:

$$e = d_{\min} - 1 \quad (8.1)$$

La elección del código de error es una concesión, ya que aunque detecta más errores, un código con un valor d_{\min} mayor envía más información redundante que un código de error con un valor d_{\min} menor. Para medir la cantidad de sobrecarga, los ingenieros definen una *tasa de código* que proporciona la relación entre el tamaño de una palabra de datos y el tamaño de la palabra de código. La ecuación (8.2) define la tasa de código, R , para un esquema de codificación de errores (n,k) :

$$R = \frac{k}{n} \quad (8.2)$$

8.11 Corrección de errores con paridad de fila y de columna (RAC)

Ya vimos cómo un esquema de codificación de canal puede detectar errores. Para comprender cómo puede usarse un código para corregir errores, considere el siguiente ejemplo. Suponga que una palabra de datos consiste en $k = 12$ bits. En vez de pensar en los bits como una sola cadena, imagine ordenarlos en un arreglo de tres filas y cuatro columnas, agregando un bit de paridad para cada fila y para cada columna. La figura 8.7 ilustra el arreglo, que se conoce como codificación de *filas y columnas* (RAC). La codificación RAC del ejemplo tiene $n = 20$, lo que significa que es una codificación $(20, 12)$.



Figura 8.7 Un ejemplo de codificación de filas y columnas con los bits de datos ordenados en un arreglo de 3×4 y agregando un bit de paridad par para cada fila y cada columna.

Para entender cómo funciona la corrección de errores, suponga que cuando se transmiten los bits de datos de la figura 8.7 hay un bit corrupto. El receptor ordena en un arreglo los bits que llegaron, luego vuelve a calcular la paridad de cada fila y de cada columna, y compara el resultado con el valor recibido. El bit que cambió hace que dos de las verificaciones de paridad fallen, tal como lo ilustra la figura 8.8.

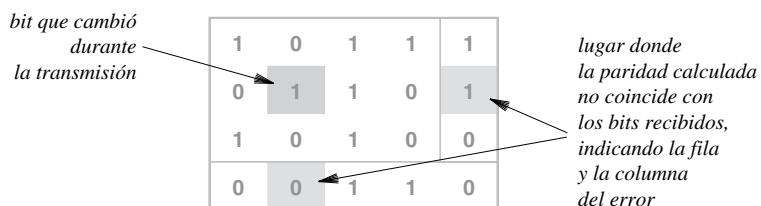


Figura 8.8 Ejemplo de cómo puede corregirse un error de un solo bit usando codificación de fila y columna.

Como se ilustra en la figura, un solo error de un bit provocará que dos bits de paridad calculados no coincidan con el bit de paridad recibido. Los dos desacuerdos corresponden a la fila y la columna del error. Un receptor usa los bits de paridad calculados para determinar exactamente qué bit de datos tiene el error y luego corrige ese bit de datos. Por ende, una codificación RAC puede corregir cualquier error que modifique un solo bit de datos.

¿Qué ocurre con un código RAC si un error cambia más de un bit en un bloque dado? La codificación RAC sólo puede corregir errores de un solo bit. En caso que haya errores en varios bits, donde cambie un número impar de bits, una codificación RAC podrá detectar pero no corregir el problema.

Para resumir:

Una codificación de filas y columnas (RAC) permite a un receptor corregir cualquier error de un solo bit y detectar errores en los que se cambia un número impar de bits.

8.12 La suma de verificación de 16 bits que se utiliza en Internet

Hay un esquema de codificación de canal particular que desempeña un papel clave en Internet. Conocido como *suma de verificación de Internet*, el código consiste en una suma de verificación de complementos a 1 de 16 bits. La suma de verificación de Internet no impone un tamaño fijo sobre una palabra de datos. En su lugar, el algoritmo permite que un mensaje sea arbitrariamente largo, y calcula una suma de verificación sobre todo el mensaje. En esencia, la suma de verificación de Internet considera los datos de un mensaje como una serie de enteros de 16 bits, como se ilustra en la figura 8.9.

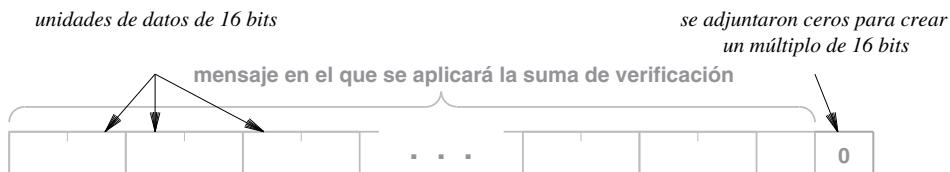


Figura 8.9 La suma de verificación de Internet divide los datos en unidades de 16 bits, adjuntando ceros si los datos no son un múltiplo exacto de 16 bits.

Para calcular una suma de verificación, un emisor suma los valores numéricos de los enteros de 16 bits y transmite el resultado. Para validar el mensaje, un receptor realiza el mismo cálculo. El algoritmo 8.1 proporciona los detalles del cálculo.

Algoritmo 8.1

Dado:

Un mensaje M de longitud arbitraria

Se calcula:

Una suma de verificación de complementos a 1 de 16 bits, C, usando aritmética de 32 bits

Método:

Rellenar M con bits cero para crear un múltiplo exacto de 16 bits

Establecer un entero de suma de verificación de 32 bits, C, en 0; para (cada grupo de 16 bits en M) {

Tratar los 16 bits como un entero y sumarlo a C;

}

Extraer los 16 bits de orden superior de C y sumarlos a C;

La inversa de los 16 bits de orden inferior de C es la suma de verificación;

Si la suma de verificación es cero, sustituir la forma de todos unos que hay en cero.

Algoritmo 8.1 El algoritmo de suma de verificación de 16 bits que se utiliza en los protocolos de Internet.

La clave para comprender el algoritmo es tener en cuenta que la suma de verificación se calcula en aritmética de complementos a 1 en vez de la aritmética de complementos a 2 que se encuentra en la mayoría de las computadoras, y usa enteros de 16 bits en vez de enteros de 32 o 64 bits. Así, el algoritmo se escribe para usar aritmética de complementos a 2 de 32 bits con el fin de realizar un cálculo de complementos a 1. Durante el ciclo *for*, la suma puede desbordarse. Por consiguiente, después del ciclo el algoritmo vuelve a sumar el desbordamiento (los bits de orden superior) a la suma. La figura 8.10 ilustra el cálculo.

$$\begin{array}{r}
 & 0100\ 1000\ 0110\ 0101 \\
 & 0110\ 1100\ 0110\ 1100 \\
 + & 0110\ 1111\ 0010\ 0001 \\
 \hline
 & 1\ 0010\ 0011\ 1111\ 0010
 \end{array}
 \quad \left. \begin{array}{l} \text{sumar valores de 16 bits} \\ \text{desbordamiento} \quad \rightarrow \\ (\text{arriba de 16}) \end{array} \right\}$$

$$\begin{array}{r}
 & 0010\ 0011\ 1111\ 0010 \\
 + & \quad \quad \quad \quad 1 \\
 \hline
 & 0010\ 0011\ 1111\ 0011
 \end{array}
 \quad \left. \begin{array}{l} \text{sumar desbordamiento} \\ \quad \quad \quad \quad 1 \end{array} \right\}$$

$$1101\ 1100\ 0000\ 1100 \quad \text{invertir el resultado}$$

Figura 8.10 Un ejemplo del algoritmo 8.1 aplicado a seis octetos de datos.

¿Por qué una suma de verificación se calcula como el inverso aritmético de la suma en vez de la propia suma? La respuesta es la eficiencia: un receptor puede aplicar el mismo algoritmo de suma de verificación que el emisor, pero puede incluir la misma suma de verificación. Como contiene el inverso aritmético del total, al sumar la suma de verificación al total se obtendrá cero como resultado. Así, un receptor incluye la suma de verificación en el cálculo y luego evalúa si la suma resultante es cero.

Hay un detalle final de la aritmética de complementos a 1 en el último paso del algoritmo. La aritmética de complementos a uno tiene dos formas de cero: todos ceros y todos unos. La suma de verificación de Internet usa la forma de todos unos para indicar que se calculó una suma de verificación y que el valor de esa suma de verificación es cero; los protocolos de Internet usan la forma de todos ceros para indicar que no se calculó una suma de verificación.

8.13 Códigos de redundancia cíclica (CRC)

Hay una forma de codificación de canal conocida como *código de redundancia cíclica (CRC)* que se utiliza en las redes de datos de alta velocidad. Los códigos CRC tienen tres propiedades clave que los hacen importantes, como se sintetiza en la figura 8.11.

Mensaje de longitud arbitraria	Al igual que una suma de verificación, el tamaño de una palabra de datos no es fijo, lo que significa que puede aplicarse un CRC a un mensaje de cualquier longitud
Excelente detección de errores	Puesto que el valor calculado depende de la secuencia de bits en un mensaje, un CRC ofrece una excelente capacidad de detección de errores
Implementación rápida en hardware	A pesar de su fundamento matemático sofisticado, un cálculo de CRC puede realizarse con extrema rapidez mediante hardware

Figura 8.11 Los tres aspectos clave de un CRC que lo hacen importante en las redes de datos.

El término *cíclico* se deriva de una propiedad de las palabras de código: un desplazamiento circular de los bits de cualquier palabra de código produce otra palabra de código. La figura 8.12 ilustra un código de redundancia cíclica (7, 4) que fue introducido por Hamming.

Palabra de datos	Palabra de código	Palabra de datos	Palabra de código
0000	0000 000	1000	1000 101
0001	0001 011	1001	1001 110
0010	0010 110	1010	1010 011
0011	0011 101	1011	1011 000
0100	0100 111	1100	1100 010
0101	0101 100	1101	1101 001
0110	0110 001	1110	1110 100
0111	0111 010	1111	1111 111

Figura 8.12 Un ejemplo de código de redundancia cíclica (7, 4).

Los códigos CRC han sido estudiados de manera exhaustiva y se ha producido una variedad de explicaciones matemáticas y técnicas computacionales. Las descripciones parecen tan dispares que es difícil entender cómo pueden todas referirse al mismo concepto. Éstas son las opiniones principales:

- Los *matemáticos* explican un cálculo de CRC como el residuo de una división de dos polinomios con coeficientes binarios, uno de los cuales representa el mensaje y el otro representa un divisor fijo.
 - Los *teóricos computacionales* explican un cálculo de CRC como el residuo de una división de dos números binarios, uno de los cuales representa el mensaje y el otro representa un divisor fijo.
 - Los *criptógrafos* explican un cálculo de CRC como una operación matemática en un campo de Galois de orden 2, lo que se escribe como GF(2).
 - Los *programadores de computadoras* explican un cálculo de CRC como un algoritmo que itera a través de un mensaje y usa la búsqueda en tablas para obtener un valor aditivo para cada paso.
 - Los *arquitectos de hardware* explican un cálculo de CRC como una pequeña unidad de canalización de hardware que recibe como entrada una secuencia de bits de un mensaje y produce un CRC sin usar división o iteración.

Como ejemplo de las opiniones anteriores, considere la división de los números binarios bajo la suposición de que no hay acarreos. Como no se realizan acarreos, la resta produce un módulo de dos y podemos reemplazarla por un *o exclusivo*. La figura 8.13 ilustra el cálculo mostrando la división de 1010, que representa un mensaje, por una constante elegida para un CRC específico, 1011.

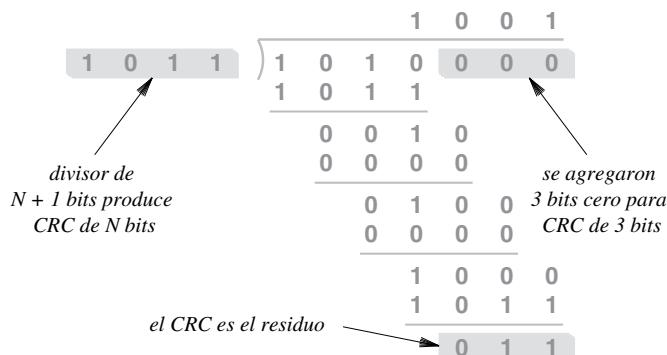


Figura 8.13 Ilustración del cálculo de un CRC visto como el residuo de una división binaria sin acarreos (es decir, donde la resta se convierte en un *o exclusivo*).

Para comprender cómo es que los matemáticos pueden considerar lo anterior como división de polinomios, piense en cada bit de un número binario como el coeficiente de un término en un polinomio. Por ejemplo, podemos considerar los dígitos del divisor en la figura 8.13, 1011 , como coeficientes del siguiente polinomio:

$$1 \times x^3 + 0 \times x^2 + 1 \times x^1 + 1 \times x^0 = x^3 + x + 1$$

De manera similar, el dividendo de la figura 8.13, 1010000 , representa el siguiente polinomio:

$$x^6 + x^4$$

Usamos el término *polinomio generador* para describir a un polinomio que corresponde a un divisor. La selección de un polinomio generador es clave para crear un CRC con buenas propiedades de detección de errores. Por lo tanto, se han realizado muchos análisis matemáticos sobre los polinomios generadores. Sabemos, por ejemplo, que un polinomio ideal es irreducible (es decir, sólo puede dividirse uniformemente entre sí mismo y entre 1), y que un polinomio con más de un coeficiente distinto de cero puede detectar todos los errores de un solo bit.

8.14 Una implementación eficiente de hardware de CRC

El hardware necesario para calcular un CRC es sorprendentemente simple. El hardware de CRC se dispone como un registro de desplazamiento con compuertas de *o exclusivo* (*xor*) entre algunos de los bits. Al calcular un CRC, el hardware se inicializa de modo que todos los bits del registro de desplazamiento sean cero. Posteriormente, los bits de datos se desplazan hacia dentro, uno a la vez. Una vez que se desplaza el último bit de datos, el valor en el registro de desplazamiento es el CRC.

El registro de desplazamiento opera una vez por cada bit de entrada y todas las partes operan al mismo tiempo, como la línea de producción en una fábrica. Durante un ciclo, cada etapa del registro acepta el bit directamente de la etapa anterior o acepta la salida de una operación *xor*. El *xor* siempre involucra el bit de la etapa anterior y un bit de retroalimentación de una etapa posterior.

La figura 8.14 ilustra el hardware necesario para el cálculo del CRC de 3 bits de la figura 8.13. Como es posible realizar una operación *xor* y un *desplazamiento* a velocidad alta, el arreglo puede usarse para redes de computadoras de alta velocidad.

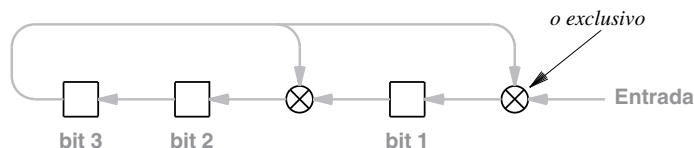


Figura 8.14 Una unidad de hardware para calcular un CRC de 3 bits para $x^3 + x^1 + 1$.

8.15 Mecanismos de solicitud de repetición automática (ARQ)

Una metodología de solicitud de repetición automática (ARQ) para la corrección de errores requiere que un emisor y un receptor se envíen información adicional. Es decir, cada vez que un lado envíe un mensaje al otro, el lado receptor debe enviar de regreso un mensaje corto de *confirmación*. Por ejemplo, si A envía un mensaje a B, B envía una confirmación de regreso a A. Una vez que reciba una confirmación,

A sabrá que el mensaje llegó correctamente. Si no se recibe una confirmación después de T unidades de tiempo, A asume que se perdió el mensaje y *vuelve a transmitir* una copia.

El mecanismo ARQ es especialmente útil en casos en los que el sistema proporciona detección de errores pero no corrección de éstos. Por ejemplo, muchas redes de computadoras usan un CRC para detectar errores de transmisión. En tales casos, es posible agregar un esquema ARQ para garantizar la entrega: si ocurre un error de transmisión, el receptor descarta el mensaje y el emisor vuelve a transmitir otra copia.

En el capítulo 25 hablaremos sobre los detalles de un protocolo de Internet que usa la metodología ARQ. Además de mostrar cómo funciona en la práctica el paradigma de *tiempo excedido* y *retransmisión*, el capítulo explica cómo el emisor y el receptor identifican los datos que se van a confirmar y habla sobre cuánto tiempo debe esperar un emisor antes de retransmitir.

8.16 Resumen

Los sistemas de transmisión físicos son susceptibles de interferencia, distorsión y atenuación, todo lo cual puede provocar errores. Los errores de transmisión pueden provocar errores de un solo bit o ráfagas de errores, y pueden existir supresiones cuando una señal recibida sea ambigua (es decir, que no se sepa con claridad si es un 1 o un 0). Para controlar los errores, los sistemas de comunicaciones de datos emplean un mecanismo de corrección de errores en la recepción o usan una técnica de solicitud de repetición automática (ARQ).

En la corrección de errores en la recepción, un emisor agrega bits redundantes a los datos y codifica el resultado antes de la transmisión a través de un canal, y luego se encarga de que el receptor decodifique y compruebe los datos entrantes. Un esquema de codificación es (n,k) si una palabra de datos contiene k bits y una palabra de código contiene n bits.

Una medida de una codificación evalúa la probabilidad de que un error cambie una palabra de código válido en otra palabra de código válido. La distancia de Hamming mínima proporciona una medida precisa.

Los códigos de bloque simples, como agregar un solo bit de paridad a cada byte, pueden detectar un número impar de errores de bits, pero no pueden detectar un número par de cambios de bits. Un código de filas y columnas (RAC) puede corregir los errores de un solo bit, y puede detectar cualquier error de varios bits en el que un número impar de bits se cambian en un bloque.

La suma de verificación de 16 bits utilizada en Internet puede usarse con un mensaje de cualquier tamaño. El algoritmo de suma de verificación divide un mensaje en bloques de 16 bits y calcula el inverso aritmético de la suma de complementos a 1 de los bloques; el desbordamiento se suma de nuevo a la suma de verificación.

Los códigos de redundancia cíclica (CRC) se usan en redes de alta velocidad debido a que un CRC acepta un mensaje de cualquier longitud, proporciona una detección de errores extremadamente buena y tiene una implementación de hardware eficiente. Las técnicas de CRC tienen una base matemática y se han estudiado de manera exhaustiva. Podemos considerar un cálculo de CRC como la obtención del residuo de una división binaria, la obtención del residuo de una división de polinomios o como una operación que usa la teoría del campo de Galois. El hardware para realizar un cálculo de CRC usa un registro de desplazamiento y operaciones del tipo *o exclusivo*.

EJERCICIOS

- 8.1** Mencione y explique las tres fuentes principales de errores de transmisión.
- 8.2** ¿Cómo afectan los errores de transmisión a los datos?
- 8.3** En una ráfaga de errores, ¿cómo se mide la longitud de la ráfaga?
- 8.4** ¿Qué es una palabra de código y cómo se usa en la corrección de errores en la recepción?
- 8.5** Proporcione un ejemplo de un código de errores de bloque utilizado con datos tipo carácter.
- 8.6** ¿Qué logra un esquema de codificación de canal ideal?
- 8.7** Defina el concepto de *distancia de Hamming*.
- 8.8** Calcule la distancia de Hamming para los siguientes pares: (0000, 0001), (0101, 0001), (1111, 1001) y (0001, 1110).
- 8.9** ¿Cómo se calcula el número mínimo de cambios de bits que pueden transformar una palabra de código válida en otra palabra de código válida?
- 8.10** Explique el concepto de *tasa de código*. ¿Es conveniente una tasa de código alta o baja?
- 8.11** Genere una matriz de paridad RAC para una codificación (20, 12) de la palabra de datos 100011011111.
- 8.12** ¿Qué puede lograr un esquema RAC que un esquema de bits de paridad simple no pueda?
- 8.13** Escriba un programa de computadora que calcule una suma de verificación de Internet de 16 bits.
- 8.14** ¿Cuáles son las características de un CRC?
- 8.15** Muestre la división de 100101010 entre 10101.
- 8.16** Exprese los dos valores del ejercicio anterior como polinomios.
- 8.17** Escriba un programa que implemente el código de redundancia cíclica (7,4) de la figura 8.12.
- 8.18** Mencione y explique la función de cada uno de los dos bloques de construcción de hardware utilizados para implementar el cálculo del CRC.

Contenido del capítulo

- 9.1 Introducción, 153
- 9.2 Una clasificación de los modos de transmisión, 153
- 9.3 Transmisión en paralelo, 154
- 9.4 Transmisión en serie, 155
- 9.5 Orden de transmisión: bits y bytes, 156
- 9.6 Sincronización de la transmisión en serie, 156
- 9.7 Transmisión asíncrona, 157
- 9.8 Transmisión de caracteres asíncronos RS-232, 157
- 9.9 Transmisión síncrona, 158
- 9.10 Bytes, bloques y tramas, 159
- 9.11 Transmisión isócrona, 160
- 9.12 Transmisión simplex, semidúplex y dúplex, 160
- 9.13 Equipo DCE y DTE, 162
- 9.14 Resumen, 162

9

Modos de transmisión

9.1 Introducción

Los capítulos de esta parte del libro cubren conceptos fundamentales que forman la base de las comunicaciones de datos. Este capítulo continúa la explicación haciendo énfasis en las formas en que se transmiten los datos. El capítulo presenta la terminología común, explica las ventajas y desventajas de la transmisión en serie y en paralelo, y describe los conceptos importantes de la comunicación síncrona y asíncrona. Los capítulos posteriores muestran cómo se utilizan las ideas aquí presentadas en las redes que se encuentran a lo largo de Internet.

9.2 Una clasificación de los modos de transmisión

Utilizamos el término *modo de transmisión* para referirnos a la forma en que se envían los datos a través del medio seleccionado. Los modos de transmisión pueden dividirse en dos categorías fundamentales:

- *En serie*: se envía un bit a la vez
- *En paralelo*: se envían varios bits a la vez

Como veremos más adelante, la transmisión en serie se puede categorizar con más detalle, de acuerdo con la sincronización de las transmisiones. La figura 9.1 muestra una clasificación general de los modos de transmisión descritos en el capítulo.



Figura 9.1 Una clasificación de los modos de transmisión.

9.3 Transmisión en paralelo

El término *transmisión en paralelo* se refiere a un mecanismo de transmisión que transfiere varios bits de datos al mismo tiempo sobre medios independientes. En general, la transmisión en paralelo se utiliza con un medio alámbrico en el que se usan varios alambres independientes. Además, las señales en todos los alambres se sincronizan de modo que cada bit viaje exactamente al mismo tiempo a través de cada uno de los cables. La figura 9.2 ilustra el concepto y muestra por qué los ingenieros usan el término *paralelo* para caracterizar el cableado.

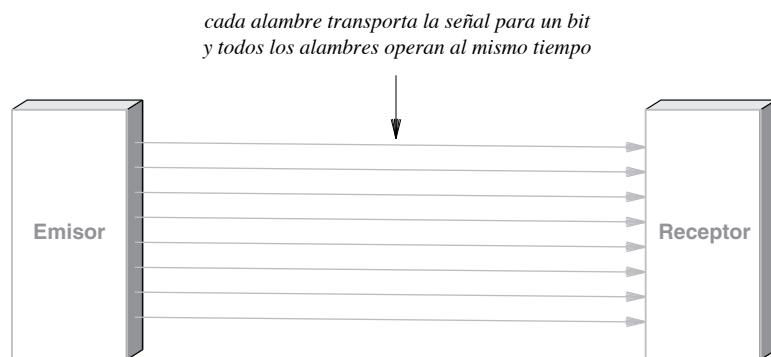


Figura 9.2 Ilustración de la transmisión en paralelo que usa 8 cables para enviar 8 bits al mismo tiempo.

La figura omite dos detalles importantes. Primero, además de cada uno de los alambres paralelos que transporta datos, una interfaz paralela por lo general contiene otros alambres que permiten la coordinación entre el emisor y al receptor. Segundo, para facilitar la instalación y el diagnóstico de fallas, los alambres de un sistema de transmisión en paralelo se colocan en un solo cable físico. Así, lo común es ver un solo cable grande que conecta a un emisor y un receptor, en vez de un conjunto de alambres físicamente independientes.

El modo de transmisión en paralelo tiene dos ventajas importantes:

- Alto rendimiento. Gracias a su capacidad para enviar simultáneamente N cantidad de bits, una interfaz en paralelo puede enviar varios bits en el mismo tiempo que tarda una interfaz en serie en enviar uno solo.
- Se adapta al hardware. En su interior, el hardware de computadora y de comunicaciones usa circuitos en paralelo. Por lo tanto, una interfaz en paralelo se adapta muy bien al hardware interno.

9.4 Transmisión en serie

La alternativa a la transmisión en paralelo, conocida como *transmisión en serie*, envía un bit a la vez. Si consideramos la velocidad, tal vez parezca lógico que quien fuera a diseñar un sistema de comunicaciones de datos elegiría la transmisión en paralelo. Sin embargo, la mayoría de los sistemas de comunicaciones usan el modo en serie. Existen tres razones principales para ello. Primero, un sistema de transmisión en serie cuesta menos, ya que se necesitan menos alambres físicos y los componentes electrónicos intermedios son menos costosos. Segundo, los sistemas en paralelo requieren que cada cable tenga exactamente la misma longitud (incluso una diferencia de milímetros podría provocar problemas). Tercero, a velocidades de datos extremadamente altas las señales de los alambres en paralelo pueden provocar ruido electromagnético que interfiere con las señales de otros alambres.

Para usar la transmisión en serie, el emisor y el receptor deben contener una pequeña pieza de hardware que convierta los datos de la forma paralela que se utiliza en el dispositivo a la forma serial que se utiliza en el alambre. La figura 9.3 ilustra la configuración.

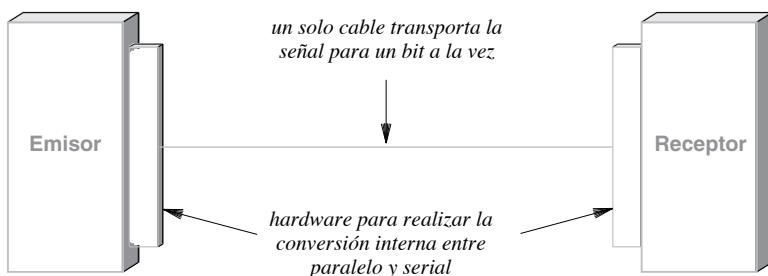


Figura 9.3 Ilustración de un modo de transmisión en serie.

El hardware necesario para convertir datos internamente entre una forma paralela y una forma serial puede ser simple o complejo, dependiendo del tipo de mecanismo de comunicación en serie que se utilice. En el caso más simple, un solo chip conocido como *receptor y transmisor asíncrono universal (UART)* realiza la conversión. Un chip similar, conocido como *receptor y transmisor síncrono-asíncrono universal (USART)*, se encarga de la conversión en las redes síncronas.

9.5 Orden de transmisión: bits y bytes

El modo de transmisión en serie presenta una cuestión interesante: al enviar bits, ¿cuál bit debe enviarse primero a través del medio? Por ejemplo, considere un entero. ¿Debería un emisor transmitir primero el *bit más significativo (MSB)* o el *bit menos significativo (LSB)*?

Los ingenieros usan el término *little-endian* para describir un sistema que envía primero el LSB y el término *big-endian* para describir un sistema que envía primero el MSB. Se puede usar cualquier forma, pero el emisor y el receptor deben estar de acuerdo.

Lo interesante es que el orden en el que se transmiten los bits no resuelve toda la cuestión sobre el orden de transmisión. Los datos en una computadora se dividen en bytes y cada byte puede dividirse aún más, en bits (por lo general, 8 bits por cada byte). Por consiguiente, es posible elegir un orden de bytes y un orden de bits, uno independiente del otro. Por ejemplo, la tecnología Ethernet especifica que los datos se envían de la siguiente forma: bytes como *big-endian* y bits como *little-endian*. La figura 9.4 ilustra el orden en el que Ethernet envía los bits a partir de una cantidad de 32 bits.

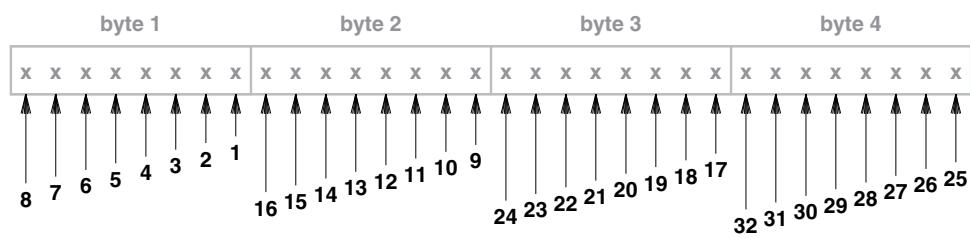


Figura 9.4 Ilustración del orden de bytes como *big-endian* y bits como *little-endian*, donde se envía primero el bit menos significativo del byte más significativo.

9.6 Sincronización de la transmisión en serie

Los mecanismos de transmisión en serie pueden dividirse en tres amplias categorías, dependiendo de la forma en que se distribuyan las transmisiones en el tiempo:

- La transmisión *asíncrona* puede ocurrir en cualquier momento, con un retraso arbitrario entre la transmisión de dos elementos de datos.
- La transmisión *síncrona* ocurre en forma continua, sin espacios vacíos entre la transmisión de dos elementos de datos.
- La transmisión *isócrona* ocurre a intervalos regulares con un espacio fijo entre la transmisión de dos elementos de datos.

9.7 Transmisión asíncrona

Un sistema de transmisión se clasifica como *asíncrono* si permite que el medio físico esté inactivo por un tiempo arbitrario entre dos transmisiones. El estilo asíncrono de comunicación se adapta bien a las aplicaciones que generan datos al azar (por ejemplo, un usuario que escribe en un teclado, que hace clic en un vínculo para obtener una página Web, que lee un rato y luego hace clic en otro vínculo para obtener otra página Web).

La desventaja de la asincronía se debe a la falta de coordinación entre el emisor y el receptor; mientras el medio está inactivo, un receptor no puede saber cuánto tiempo permanecerá el medio inactivo antes de que lleguen más datos. Por lo tanto, lo común es que las tecnologías asíncronas requieren que un emisor transmita unos cuantos bits adicionales antes de cada elemento de datos para informar al receptor que va a comenzar la transferencia de datos. Los bits adicionales permiten que el hardware del receptor se sincronice con la señal entrante. En algunos sistemas asíncronos, los bits adicionales se conocen como *preámbulo*; en otros, los bits adicionales se conocen como *bits de inicio*. Para resumir:

Puesto que permite a un emisor permanecer inactivo durante un tiempo indeterminado entre cada transmisión, un mecanismo de transmisión asíncrona envía información adicional antes de cada transmisión para que un receptor pueda sincronizarse con la señal.

9.8 Transmisión de caracteres asíncronos RS-232

Como ejemplo de comunicación asíncrona, considere la transferencia de caracteres a través de alambres de cobre entre una computadora y un dispositivo conocido como teclado. La tecnología de comunicación asíncrona estandarizada por la *Alianza de industrias electrónicas* (EIA) se ha convertido en la más ampliamente aceptada para la comunicación de caracteres. Conocido como *RS-232-C* y abreviado comúnmente como *RS-232*,[†] el estándar EIA especifica los detalles de la conexión física (por ejemplo, que la conexión no exceda los 50 pies), los detalles eléctricos (por ejemplo, que el voltaje oscile entre -15 volts y +15 volts) y la codificación de línea (por ejemplo, que el voltaje negativo corresponda al 1 lógico y el positivo al 0 lógico).

Debido a que está diseñado para usarse con dispositivos como los teclados, el estándar RS-232 especifica que cada elemento de datos representa un carácter. El hardware puede configurarse para controlar el número exacto de bits por segundo y enviar caracteres de siete u ocho bits. Aunque un emisor puede generar un retraso de una longitud arbitraria antes de enviar un carácter, una vez que comienza la transmisión, éste transmite todos los bits del carácter uno después de otro, sin retraso entre éstos. Al terminar la transmisión, el emisor deja el cable con un voltaje negativo (que corresponde a un 1 lógico) hasta que haya otro carácter listo para ser transmitido.

¿Cómo sabe un receptor dónde comienza un nuevo carácter? El estándar RS-232 especifica que un emisor debe transmitir un bit 0 adicional (conocido como *bit de inicio*) antes de transmitir los bits de

[†]Aunque el estándar RS-449 posterior ofrece un poco más de funcionalidad, la mayoría de los ingenieros siguen usando el nombre original.

un carácter. Además, el RS-232 especifica que un emisor debe dejar la línea inactiva entre un carácter y otro durante al menos el tiempo requerido para enviar un bit. En consecuencia, podemos pensar en un bit *I fantasma* que se adjunta a cada carácter. La figura 9.5 ilustra cómo varía el voltaje cuando se envían un bit de inicio, ocho bits de un carácter y un bit de parada.

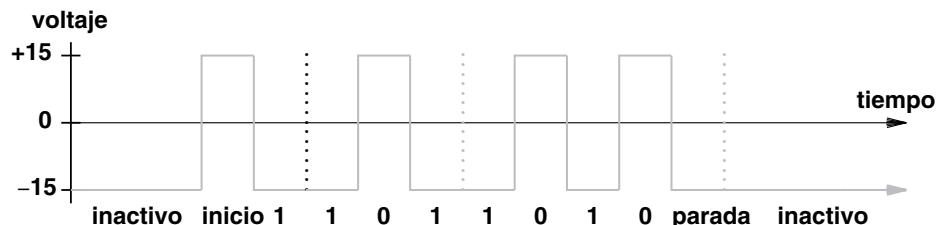


Figura 9.5 Ilustración del voltaje durante la transmisión de un carácter de 8 bits al usar RS-232.

Para resumir:

El estándar RS-232 utilizado en distancias cortas para la comunicación en serie asíncrona, antepone a cada carácter un bit de inicio, envía cada bit del carácter y después de cada carácter deja un periodo inactivo de al menos un bit de longitud (bit de parada).

9.9 Transmisión síncrona

La principal alternativa para la transmisión asíncrona se conoce como *transmisión síncrona*. En el nivel más elemental, un mecanismo síncrono transmite bits de datos en forma continua, sin tiempo de inactividad entre un bit y el siguiente. Es decir, después de transmitir el último bit de un byte de datos, el emisor transmite el primer bit del siguiente byte de datos.

La principal ventaja de un mecanismo síncrono se debe a que el emisor y el receptor permanecen en sincronización constante, lo que implica una menor sobrecarga de sincronización. Para comprender la sobrecarga, compare la transmisión de caracteres de 8 bits en un sistema asíncrono, como se ilustra en la figura 9.5, con la del sistema síncrono que se ilustra en la figura 9.6. Cada carácter enviado mediante RS-232 requiere adicionalmente de un bit de inicio y un bit de parada, lo que significa que cada carácter de 8 bits requiere un mínimo de tiempos de 10 bits, aun cuando no se inserte un tiempo inactivo. En un sistema síncrono, cada carácter se envía sin bits de inicio ni parada.

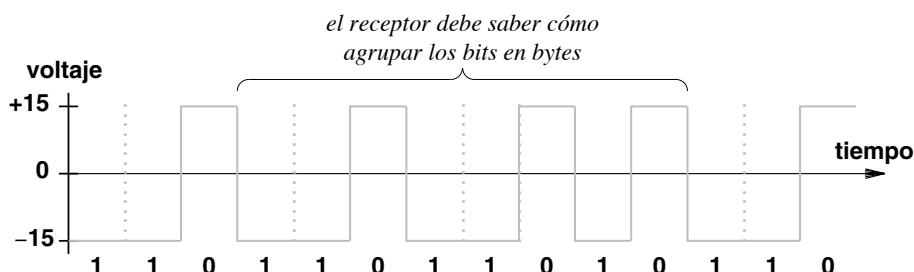


Figura 9.6 Ilustración de la transmisión síncrona, donde el primer bit de un byte sigue justo después del último bit del byte anterior.

En conclusión:

Si se compara con la transmisión síncrona, un mecanismo RS-232 asíncrono tiene un 25% más de sobrecarga por cada carácter.

9.10 Bytes, bloques y tramas

Si el mecanismo síncrono a utilizar debe enviar bits en forma continua, ¿qué ocurre si un emisor no tiene datos listos para enviarse en todo momento? La respuesta está en una técnica conocida como *entramado*: se agrega una interfaz a un mecanismo asíncrono que acepte y transmita un *bloque* de bytes conocido como *trama*. Para asegurar que el emisor y el receptor permanezcan sincronizados, una trama comienza con una secuencia especial de bits. Además, la mayoría de los sistemas síncronos incluyen una secuencia *inactiva* (o *byte inactivo*) especial que se transmite cuando el emisor no tiene datos para enviar. La figura 9.7 ilustra el concepto.

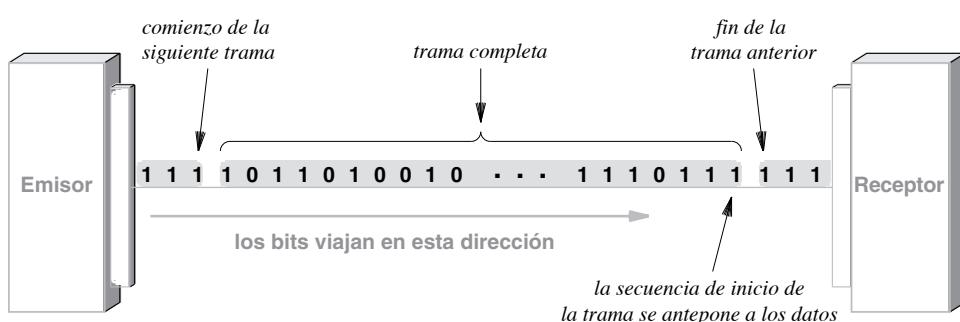


Figura 9.7 Ilustración del entramado en un sistema de transmisión síncrona.

Podemos resumir así la consecuencia del entramado:

Aunque el mecanismo involucrado transmite bits en forma continua, el uso de una secuencia inactiva y de un entramado permite que un mecanismo de transmisión asíncrona proporcione una interfaz enfocada en los bytes y que haya espacios inactivos entre bloques de datos.

9.11 Transmisión isócrona

El tercer tipo de sistema de transmisión no propone en realidad un nuevo mecanismo. En su lugar puede considerarse como una forma importante de usar la transmisión síncrona. Conocido como *transmisión isócrona*, el sistema está diseñado para brindar un flujo estable de bits para las aplicaciones multimedia que contengan voz o video. Es imprescindible entregar dichos datos a una velocidad estable, ya que las variaciones en el retraso (conocidas como *fluctuación*) pueden afectar la recepción (puede provocar crujidos o tronidos en el audio, o hacer que el video se congele por breves instantes).

En vez de usar la presencia de datos para controlar la transmisión, una red isócrona está diseñada para aceptar y enviar datos a una velocidad fija, R . De hecho, la interfaz para la red es tal que los datos *deben* entregarse a la red para transmitirse a exactamente R bits por segundo. Por ejemplo, un mecanismo isócrono diseñado para transmitir voz opera a una velocidad de 64,000 bits por segundo. Un emisor debe generar audio digitalizado en forma continua y un receptor debe ser capaz de aceptar y reproducir el flujo.

La red a utilizar puede usar el entramado y optar por transmitir información adicional junto con los datos. Sin embargo, para ser isócrono, un sistema debe diseñarse de modo que el emisor y el receptor vean un flujo continuo de datos, sin retrasos adicionales al comienzo de una trama. Por consiguiente, una red isócrona que proporciona una velocidad de datos de R bits por segundo incluye por lo general un mecanismo síncrono que opera a un poco más que R bits por segundo.

9.12 Transmisión simplex, semidúplex y dúplex

Los canales de comunicaciones se pueden clasificar en uno de los tres tipos siguientes, dependiendo de la dirección de la transferencia.

- Simplex
- Dúplex (full-duplex)
- Semidúplex (half-duplex)

Símplex. Un mecanismo *símplex* es el más fácil de entender. Como su nombre lo indica, un mecanismo simplex puede transferir datos en una sola dirección. Por ejemplo, una fibra óptica individual actúa como un mecanismo de transmisión simplex, ya que la fibra tiene un dispositivo de transmisión (un LED o láser) en un extremo y un dispositivo receptor fotosensible en el otro. La transmisión simplex es similar a la difusión de radio o televisión. La figura 9.8(a) ilustra la comunicación simplex.

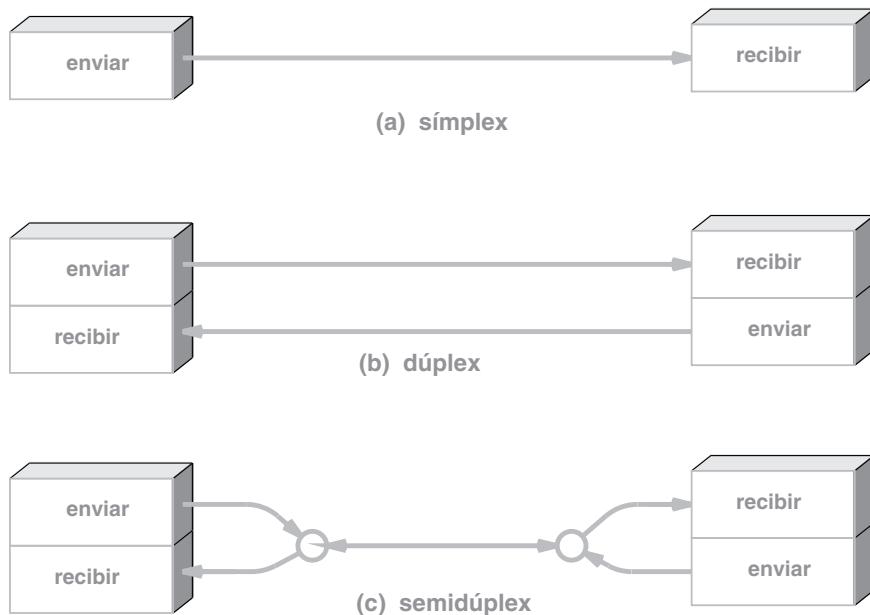


Figura 9.8 Ilustración de los tres modos de operación.

Dúplex. El mecanismo *dúplex* también es sencillo: el sistema involucrado permite la transmisión simultánea en dos direcciones. Por lo general, un mecanismo dúplex consiste de dos mecanismos *simplex*, cada uno de los cuales transmite la información en cada sentido, como se ilustra en la figura 9.8(b). Por ejemplo, es posible usar un par de fibras ópticas para brindar comunicación dúplex si se tienden las dos en paralelo y se hacen los arreglos para enviar datos en sentidos opuestos. La comunicación dúplex es similar a una conversación telefónica de voz en la que un participante puede hablar y a su vez escuchar la música de fondo proveniente del otro extremo.

Semidúplex. Un mecanismo *semidúplex* implica un medio de transmisión compartido. El medio compartido puede usarse para la comunicación en cada dirección, pero ésta no puede realizarse de manera simultánea. Por consiguiente, la comunicación semidúplex es similar al uso de los radios tipo “walkie-talkie”, donde sólo uno de los lados puede transmitir a la vez. Se necesita un mecanismo adicional en cada extremo de una comunicación semidúplex que coordine la transmisión para asegurar que sólo un lado transmita en un momento dado. La figura 9.8(c) ilustra la comunicación semidúplex.

9.13 Equipo DCE y DTE

Los términos *equipo de comunicación de datos (DCE)* y *equipo terminal de datos (DTE)* fueron creados originalmente por AT&T para distinguir entre el equipo de comunicaciones que pertenecía a la compañía telefónica y el equipo *terminal* que pertenecía a un suscriptor.

La terminología se mantiene: si un negocio renta un circuito de datos a una compañía telefónica, la compañía instala el equipo DCE en el negocio y éste compra un equipo DTE que conecta al equipo de la compañía telefónica.

Desde un punto de vista académico, la importancia de distinguir entre DCE y DTE no es quién es el propietario del equipo, sino entender que se tiene la habilidad de definir una interfaz cualquiera para un usuario. Por ejemplo, si la red empleada usa transmisión síncrona, el equipo DCE puede brindar una interfaz síncrona o isócrona para el equipo del usuario. La figura 9.9 ilustra la estructura conceptual.[†]

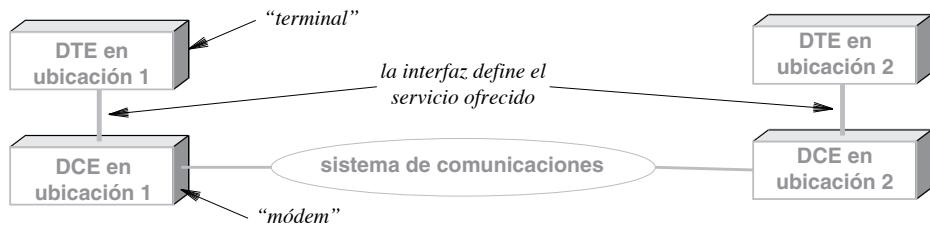


Figura 9.9 Ilustración del equipo de comunicación de datos y del equipo terminal de datos, que proporcionan un servicio de comunicación entre dos ubicaciones.

Existen varios estándares que especifican una posible interfaz entre DCE y DTE. Por ejemplo, es posible usar el estándar RS-232 descrito en este capítulo y el estándar RS-449 diseñado como reemplazo. Además, hay un estándar disponible, conocido como X.21.

9.14 Resumen

Los sistemas de comunicaciones usan la transmisión en paralelo o en serie. Un sistema paralelo tiene varios alambres y cada alambre transmite la señal para un bit en un momento dado. Por otra parte, un sistema de transmisión en paralelo con K cables puede enviar K bits al mismo tiempo. Aunque la comunicación en paralelo ofrece una mayor velocidad, casi todos los sistemas de comunicaciones usan mecanismos en serie de menor costo que envían un bit a la vez.

[†] Nota: los términos DCE y DTE también se usan para distinguir entre dos tipos de conectores, aun cuando el equipo no pertenezca a una compañía telefónica (por ejemplo, el conector en una PC y el conector en un módem externo).

La comunicación en serie requiere que el emisor y el receptor estén de acuerdo en la sincronización y el orden en el que se enviarán los bits. El orden de transmisión define si se envía primero el bit más significativo o el menos significativo, y si se envía primero el byte más significativo o el menos significativo.

Los tres tipos de sincronización son: a) asíncrona, donde la transmisión puede ocurrir en cualquier momento y el sistema de comunicaciones puede permanecer inactivo entre transmisiones, b) síncrona, donde los bits se transmiten en forma continua y los datos se agrupan en tramas, y c) isócrona, donde la transmisión ocurre a intervalos regulares sin retrasos adicionales en los límites de las tramas.

Un sistema de comunicaciones puede ser simplex, dúplex o semidúplex. Un mecanismo simplex envía datos en una sola dirección. Un mecanismo dúplex transfiere datos en dos direcciones al mismo tiempo, y un mecanismo semidúplex permite la transferencia en dos sentidos, pero sólo permite transferir en un sentido a la vez.

La distinción entre un equipo de comunicación de datos y un equipo terminal de datos se ideó originalmente para denotar si un proveedor o un suscriptor eran propietarios del equipo. El concepto clave surge de la habilidad de definir una interfaz para el usuario que ofrece un servicio diferente al del sistema de comunicaciones que se usa.

EJERCICIOS

- 9.1** Describa la diferencia entre transmisión en serie y transmisión en paralelo.
- 9.2** ¿Cuáles son las ventajas de la transmisión en paralelo? ¿Cuál es la principal desventaja?
- 9.3** Al transmitir un entero de 32 bits en complementos a 2 usando un orden *big-endian*, ¿en qué momento se transmite el bit de signo?
- 9.4** ¿Cuál es la principal característica de la transmisión asíncrona?
- 9.5** ¿Qué tipo o tipos de transmisión en serie son apropiados para transmitir video? ¿Para conectar un teclado a una computadora?
- 9.6** ¿Qué es un bit de inicio y con qué tipo de transmisión en serie se usa?
- 9.7** Al usar un esquema de transmisión asíncrona, ¿qué ocurre cuando un emisor no tiene datos para enviar?
- 9.8** Cuando dos humanos sostienen una conversación, ¿usan el tipo de transmisión simplex, semidúplex o dúplex?
- 9.9** ¿Un módem se clasifica como DTE o DCE?
- 9.10** Busque en Web las especificaciones de los pines de un conector DB-25 (nota: los pinos 2 y 3 son para transmitir y recibir). En un conector tipo DCE, ¿el pin 2 transmite o recibe?

Contenido del capítulo

- 10.1 Introducción, 165
- 10.2 Portadoras, frecuencia y propagación, 165
- 10.3 Esquemas analógicos de modulación, 166
- 10.4 Modulación de amplitud, 166
- 10.5 Modulación de frecuencia, 167
- 10.6 Modulación por desplazamiento de fase, 168
- 10.7 Modulación de amplitud y el teorema de Shannon, 168
- 10.8 Modulación, entrada digital y codificación por desplazamiento, 168
- 10.9 Codificación por desplazamiento de fase, 169
- 10.10 Cambio de fase y diagramas de constelación, 171
- 10.11 Modulación de amplitud en cuadratura, 173
- 10.12 Hardware para modulación y demodulación, 174
- 10.13 Módems ópticos y de radiofrecuencias, 174
- 10.14 Módems de marcación telefónica, 175
- 10.15 QAM aplicada a la marcación telefónica, 175
- 10.16 Módems de marcación telefónica V.32 y V.32bis, 176
- 10.17 Resumen 177

10

Modulación y módems

10.1 Introducción

Cada uno de los capítulos de esta parte del libro cubre un aspecto de las comunicaciones de datos. Los capítulos anteriores hablan sobre las fuentes de información, explican cómo es que una señal puede representar la información y describen las formas de energía utilizadas por diversos medios de transmisión.

Este capítulo continúa la explicación sobre las comunicaciones de datos enfocándose en el uso de las señales de alta frecuencia para transportar información. El capítulo habla sobre cómo se usa la información para cambiar una onda electromagnética de alta frecuencia, explica también por qué es importante la técnica que se utiliza y describe cómo se usan las entadas analógicas y digitales. Los capítulos posteriores amplían la explicación mostrando cómo puede crearse un sistema de comunicaciones que transfiera al mismo tiempo varios flujos independientes de datos sobre un medio de transmisión compartido.

10.2 Portadoras, frecuencia y propagación

Muchos sistemas de comunicaciones de larga distancia usan una onda electromagnética que oscila en forma continua, conocida como *portadora* o *carrier*. El sistema realiza pequeños cambios en la onda portadora los cuales representan la información que se va a enviar. Para entender por qué son importantes las portadoras, recuerde que en el capítulo 7 comentamos que la frecuencia de la energía electromagnética determina cómo se propaga la energía. Una razón para el uso de las portadoras surge del deseo de seleccionar una frecuencia que se propague bien, sin importar la velocidad con la que se envíen los datos.

10.3 Esquemas analógicos de modulación

Usamos el término *modulación* para referirnos a los cambios realizados en una portadora, de acuerdo con la información que se va a enviar. En teoría, la modulación recibe dos entradas, una portadora y una señal, y genera una portadora modulada como salida, tal como se ilustra en la figura 10.1.

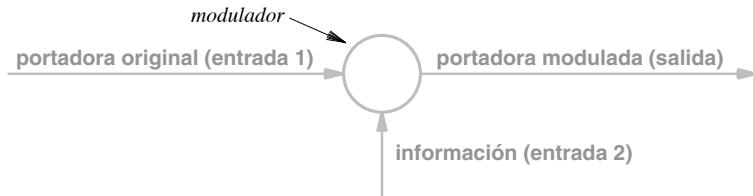


Figura 10.1 El concepto de modulación con dos entradas.

Básicamente, un emisor debe cambiar una de las características fundamentales de la onda. Por consiguiente, hay tres técnicas principales que modulan una portadora electromagnética de acuerdo con una señal:

- Modulación de amplitud
- Modulación de frecuencia
- Modulación por desplazamiento de fase

Los primeros dos métodos de modulación son los más conocidos y se han usado muchas veces. En definitiva no se originaron con las redes de computadoras; se idearon y usaron para la radiodifusión y también se utilizan para la transmisión televisiva.

10.4 Modulación de amplitud

Una técnica conocida como *modulación de amplitud* varía la amplitud de una portadora en proporción con la información que se envía (por ejemplo, de acuerdo con una señal). La portadora sigue oscilando a una frecuencia fija, pero la amplitud de la onda varía. La figura 10.2 ilustra una onda portadora sin modular, una señal de información analógica y la portadora resultante con la amplitud modulada.

La modulación de amplitud es fácil de entender, ya que sólo se modifica el tamaño de la onda senoidal. Además, si analizamos un gráfico del tiempo de una portadora modulada veremos que tiene una forma similar a la señal que se utilizó. Por ejemplo, si pensamos en una *envoltura curva* que conecta los picos de la onda senoidal en la figura 10.2(c), la curva resultante tiene la misma forma que la señal de información de la figura 10.2(b).

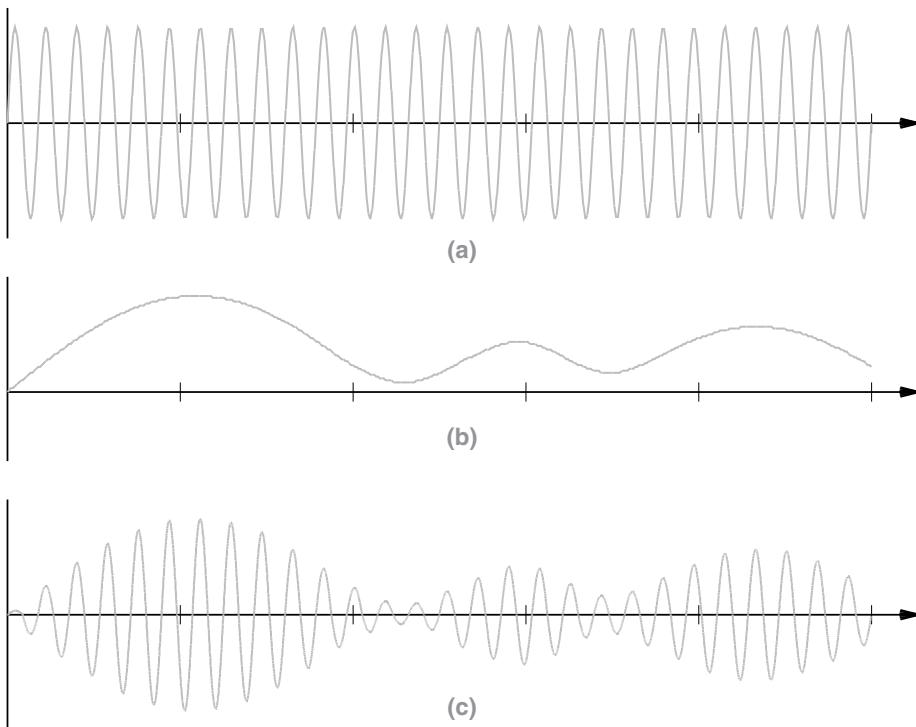


Figura 10.2 Ilustración de (a) una onda portadora sin modular, (b) una señal de información analógica y (c) una portadora con la amplitud modulada.

10.5 Modulación de frecuencia

Una alternativa a la modulación de amplitud se conoce como *modulación de frecuencia*. Cuando se emplea este tipo de modulación, la amplitud de la portadora permanece fija pero la frecuencia cambia de acuerdo con la señal: cuando ésta es más fuerte, la frecuencia de la portadora aumenta ligeramente y cuando la señal es más débil, la frecuencia de la portadora disminuye ligeramente. La figura 10.3 ilustra una onda portadora con la frecuencia modulada, a partir de la señal de la figura 10.2(b).

Como muestra la figura, la modulación de frecuencia es más difícil de visualizar debido a que los cambios ligeros en la frecuencia no son tan claramente visibles. Sin embargo, podemos observar que la onda modulada tiene mayores frecuencias cuando la señal utilizada para la modulación es más fuerte.

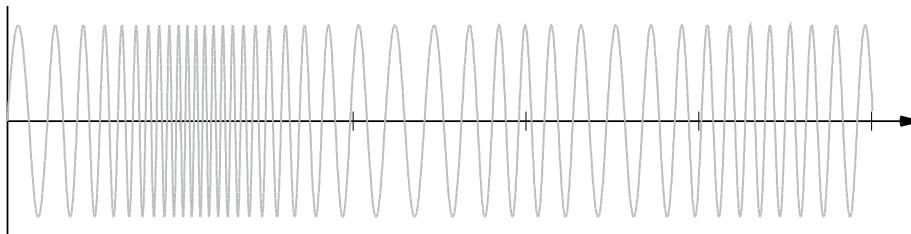


Figura 10.3 Ilustración de una onda portadora con la frecuencia modulada, de acuerdo con la señal de la figura 10.2(b).

10.6 Modulación por desplazamiento de fase

La tercera propiedad de una onda senoidal es su *fase*, la compensación de un tiempo de referencia en el que comienza la onda senoidal. Es posible aplicar cambios a la fase para representar una señal. Usamos el término *desplazamiento de fase* para representar dichos cambios.

Aunque en teoría es posible modular la fase, esta técnica se utiliza raras veces con una señal analógica. Para entender por qué, observe que si la fase cambia después del ciclo k , la siguiente onda senoidal comenzará ligeramente más tarde que el tiempo en el que se completa el ciclo k . Un ligero retraso se asemeja a un cambio en la frecuencia. De esta forma, para la entrada analógica, podemos considerar la modulación por desplazamiento de fase como una forma especial de modulación de frecuencia. Sin embargo, veremos que los desplazamientos de fase son importantes cuando se usa una señal digital para modular una portadora.

10.7 Modulación de amplitud y el teorema de Shannon

La ilustración en la figura 10.2(c) muestra que la amplitud varía de un valor máximo hasta casi cero. Aunque es fácil de entender para un ser humano, la figura es un poco engañosa ya que en la práctica la modulación sólo cambia ligeramente la amplitud de una portadora, dependiendo de una constante conocida como el *índice de modulación*.

Para comprender por qué los sistemas en la práctica no permiten que una señal modulada se aproxime a cero, considere el teorema de Shannon. Suponiendo que la cantidad de ruido es constante, la relación señal-ruido se aproximará a cero conforme la señal también se aproxime a cero. Por consiguiente, si se mantiene la onda portadora cerca del máximo se asegura que la relación señal-ruido sea lo más grande posible, lo cual permite la transferencia de más bits por segundo.

10.8 Modulación, entrada digital y codificación por desplazamiento

La anterior descripción de la modulación muestra cómo se usa una señal de información analógica para modular una portadora. Surge la pregunta: ¿cómo puede usarse una entrada digital? La respuesta se basa en modificaciones simples de los esquemas de modulación antes descritos: en vez de una modulación

que sea proporcional a una señal continua, los esquemas digitales usan valores discretos. Además, para distinguir entre modulación analógica y digital, usamos el término *codificación por desplazamiento* en vez de modulación.

En esencia, la codificación por desplazamiento opera de manera similar a la modulación analógica. En vez de un continuo de valores posibles, la codificación por desplazamiento digital tiene un conjunto fijo. Por ejemplo, la modulación de amplitud permite que la amplitud de una portadora varíe por cantidades arbitrariamente pequeñas en respuesta a un cambio en la señal que se va a usar. En contraste, la codificación por desplazamiento de amplitud usa un conjunto fijo de amplitudes posibles. En el caso más simple una amplitud total puede corresponder a un 1 lógico y una amplitud mucho más pequeña puede corresponder a un 0 lógico. De manera similar, la codificación por desplazamiento de frecuencia usa dos frecuencias básicas. La figura 10.4 ilustra una onda portadora, una señal de entrada digital y las formas de onda resultantes para la *codificación por desplazamiento de amplitud (ASK)* y la *codificación por desplazamiento de frecuencia (FSK)*.

10.9 Codificación por desplazamiento de fase

Aunque los cambios de amplitud y frecuencia funcionan bien para el audio, ambos requieren al menos un ciclo de una onda portadora para enviar un solo bit, a menos que se utilice un esquema de codificación especial (por ejemplo, si las partes positiva y negativa de la señal cambian de manera independiente). El teorema de Nyquist descrito en el capítulo 6 sugiere que el número de bits enviados por unidad de tiempo puede aumentar si el esquema de codificación permite codificar varios bits en un solo ciclo de la portadora. Por lo tanto, los sistemas de comunicaciones de datos usan comúnmente técnicas que pueden enviar más bits. En especial, la *codificación por desplazamiento de fase* cambia en forma abrupta la fase de la onda portadora para codificar datos. Cada uno de estos cambios se conoce como *desplazamiento de fase*. Después de un desplazamiento de fase, la portadora sigue oscilando, pero salta de inmediato a un nuevo punto en el ciclo de la onda senoidal. La figura 10.5 ilustra cómo afecta un desplazamiento de fase a una onda senoidal.

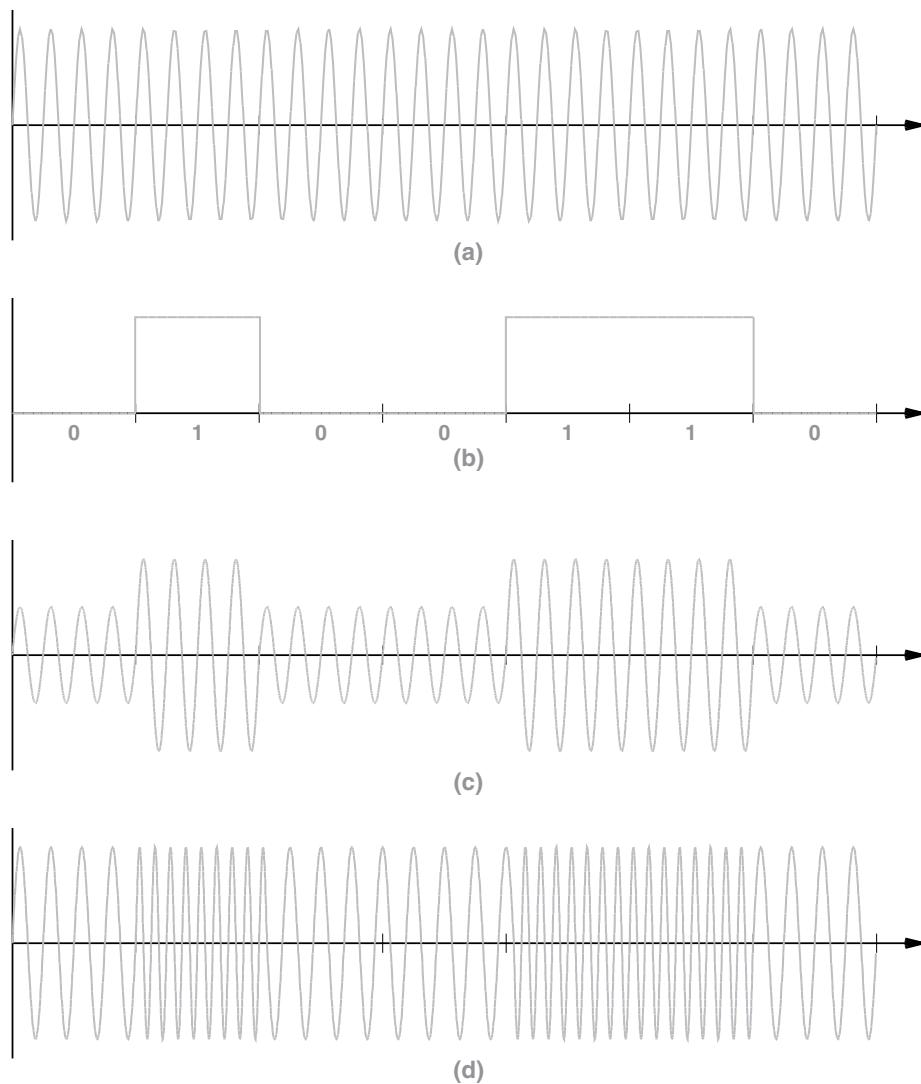


Figura 10.4 Ilustración de (a) una onda portadora, (b) una señal de entrada digital, (c) codificación por desplazamiento de amplitud y (d) codificación por desplazamiento de frecuencia.

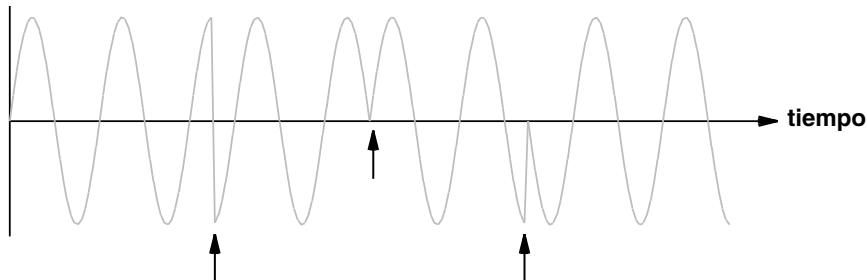


Figura 10.5 Una ilustración de la modulación por desplazamiento de fase, con flechas que indican los tiempos en que la portadora salta de manera abrupta a un nuevo punto en el ciclo de la onda senoidal.

Un desplazamiento de fase se mide por el ángulo del cambio. Por ejemplo, el desplazamiento del extremo izquierdo de la figura 10.5 cambia el ángulo por $\pi/2$ radianes o 180° . El segundo cambio de fase de la figura también corresponde a un desplazamiento de 180° . El tercer cambio de fase corresponde a un desplazamiento de -90° (lo que equivale a 270°).

10.10 Cambio de fase y diagramas de constelación

¿Cómo pueden codificarse los datos usando desplazamientos de fase? En el caso más simple, un emisor y un receptor pueden estar de acuerdo en el número de bits por segundo, y pueden usar un desplazamiento de fase nulo para denotar un 0 lógico, y en la presencia de un desplazamiento de fase, denotar un 1 lógico. Por ejemplo, un sistema podría usar un desplazamiento de fase de 180° . Para expresar la asignación exacta de bits de datos a los cambios de fase específicos, se usan los *diagramas de constelación*. La figura 10.6 ilustra el concepto.

El hardware puede hacer algo más que detectar la presencia de un desplazamiento de fase; un receptor puede medir la cantidad de desplazamiento de una portadora durante un cambio de fase. De esta forma, es posible idear un sistema de comunicaciones que reconozca un conjunto de desplazamientos de fase y use cada desplazamiento de fase para representar valores específicos de datos. Por lo general, los sistemas están diseñados para usar una potencia de dos posibles desplazamientos, lo que significa que un emisor puede usar bits de datos para seleccionar entre estos desplazamientos.

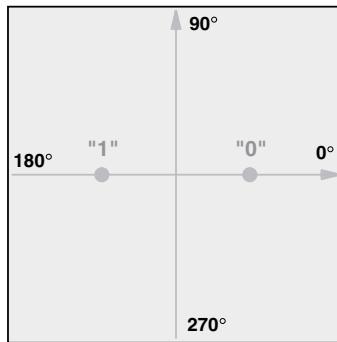


Figura 10.6 Un diagrama de constelación que muestra un 0 lógico como un desplazamiento de fase de 0° y un 1 lógico como desplazamiento de fase de 180°.

La figura 10.7 muestra el diagrama de constelación de un sistema que usa cuatro desplazamientos de fase posibles (es decir, 2^2). En cada etapa de la transmisión, un emisor usa dos bits de datos para seleccionar entre los cuatro posibles valores de desplazamiento.

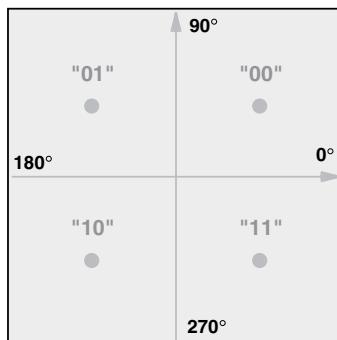


Figura 10.7 Un diagrama de constelación de un sistema que usa cuatro posibles desplazamientos de fase, cada uno de los cuales representa dos bits de datos.

Para resumir:

La principal ventaja de los mecanismos como la codificación por desplazamiento de fase se debe a la habilidad de representar más de un bit de datos en un cambio dado. Un diagrama de constelación muestra la asignación de los bits de datos a los cambios de fase.

En la práctica existen muchas variantes de la codificación por desplazamiento de fase que se utilizan en las redes. Por ejemplo, un mecanismo de desplazamiento de fase como el que se ilustra en la figura 10.6, que permite a un emisor transferir un bit a la vez, se clasifica como mecanismo de *codificación por desplazamiento de fase binaria (BPSK)*. La notación 2-PSK se usa para denotar los dos posibles valores. De manera similar, la variación que se ilustra en la figura 10.7 se conoce como mecanismo 4-PSK.

En teoría es posible incrementar la velocidad de datos si se incrementa el rango de desplazamientos de fase. Por lo tanto, un mecanismo 16-PSK puede enviar el doble de bits por segundo que un mecanismo 4-PSK. Sin embargo, en la práctica el ruido y la distorsión limitan la capacidad del hardware para distinguir las pequeñas diferencias en los desplazamientos de fase. En conclusión:

Aunque existen muchas variaciones de la codificación por desplazamiento de fase, en la práctica el ruido y la distorsión limitan la capacidad de los sistemas para distinguir las diferencias arbitrariamente pequeñas en los cambios de fase.

10.11 Modulación de amplitud en cuadratura

Si el hardware es incapaz de detectar los cambios arbitrarios de fase, ¿cómo puede incrementarse aún más la velocidad de los datos? La respuesta está en una combinación de técnicas de modulación que cambian dos características de una portadora al mismo tiempo. La tecnología más sofisticada combina la modulación de fase y la codificación por desplazamiento de fase. Conocida como *modulación de amplitud en cuadratura (QAM)*,[†] esta metodología usa tanto el cambio en fase como el cambio en amplitud para representar valores.

Para representar la QAM en un diagrama de constelación, usamos la distancia desde el origen como una medida de amplitud. Por ejemplo, la figura 10.8 muestra el diagrama de constelación para una variante conocida como *16QAM*, con áreas en color gris oscuro que indican las amplitudes.

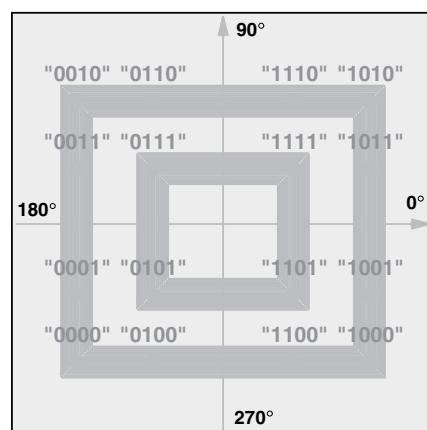


Figura 10.8 Un diagrama de constelación para 16QAM, donde la distancia desde el origen refleja la amplitud.

[†] Por lo general, los textos y la industria de las redes usan el término *modulación de amplitud en cuadratura*, aun cuando el término *codificación por desplazamiento de amplitud en cuadratura* sería más preciso.

10.12 Hardware para modulación y demodulación

Un mecanismo de hardware que acepta una secuencia de bits de datos y modula una onda portadora de acuerdo con los bits, se conoce como *modulador*; un mecanismo de hardware que acepta una onda portadora modulada y recrea la secuencia de bits de datos que se usó para modularla, se conoce como *demodulador*. De esta forma, la transmisión de datos requiere un modulador en un extremo del medio de transmisión y un demodulador en el otro. En la práctica, la mayoría de los sistemas de comunicaciones son dúplex completo, lo que significa que cada ubicación necesita tanto un modulador (que se usa para enviar datos) como un demodulador (que se usa para recibir los datos). Para mantener bajo el costo y facilitar tanto la instalación como la operación del par de dispositivos, los fabricantes combinan los mecanismos de modulación y demodulación en un solo dispositivo, conocido como *módem* (*modulador* y *demodulador*). La figura 10.9 ilustra cómo es que un par de módems usan una conexión de 4 alambres para comunicarse.

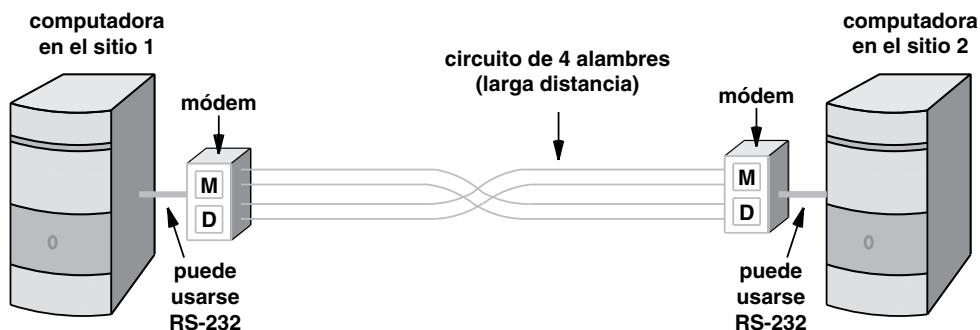


Figura 10.9 Ilustración de dos módems que usan una conexión de 4 alambres.

Como se indica en la figura, los módems están diseñados para ofrecer comunicación a través de largas distancias. Un circuito de 4 alambres que conecta dos módems puede extenderse dentro de un edificio, a lo largo de un campus corporativo, entre edificios o incluso entre ciudades.[†]

10.13 Módems ópticos y de radiofrecuencias

Además de alambres dedicados, los módems también se usan con otros medios, incluyendo la transmisión de *radiofrecuencias* (RF) y las fibras ópticas. Por ejemplo, es posible usar un par de módems de radiofrecuencia para enviar datos a través de un radio (por ejemplo, en una red Wi-Fi), y podemos usar un par de *módems ópticos* para enviar datos a través de un par de fibras ópticas. Aunque dichos módems usan medios totalmente distintos a los módems que operan a través de alambres dedicados, el principio es el mismo: en el extremo emisor, un módem modula una portadora, mientras que en el extremo receptor, los datos se extraen de la portadora modulada.

[†] Un circuito que atraviesa por una propiedad pública debe rentarse a un proveedor de servicios, por lo general una compañía telefónica.

10.14 Módems de marcación telefónica

Otra aplicación interesante de los módems involucra al sistema telefónico de voz. En vez de usar una señal eléctrica como una portadora, un *módem de marcación telefónica* usa un tono de audio. Al igual que con los módems convencionales, la portadora se modula en el extremo emisor y se demodula en el extremo receptor. Así, además de la capacidad para hacer y recibir llamadas telefónicas, la principal diferencia entre los módems de marcación telefónica y los convencionales está en el ancho de banda más bajo que tienen los tonos audibles.

Cuando se diseñaron por primera vez los módems de marcación telefónica, el método parecía muy razonable: un módem de marcación telefónica convertía los datos en una portadora analógica modulada, debido a que el sistema telefónico transportaba señales analógicas. Lo irónico es que el interior de un sistema telefónico moderno es digital. Por ende, del lado emisor un módem de marcación telefónica usa datos para modular una portadora audible, la cual se transmite al sistema telefónico. Este sistema digitaliza el audio entrante, transporta una forma digital internamente y convierte la versión digitalizada de nuevo a audio analógico para su entrega. El módem receptor demodula la portadora analógica y extrae los datos digitales originales. La figura 10.10 ilustra el uso irónico de las señales analógicas y digitales por parte de los módems de marcación telefónica.



Figura 10.10 Ilustración de señales digitales y analógicas (representadas por una onda cuadrada y una onda senoidal) que ocurren cuando se usa un módem de marcación telefónica para enviar datos de una computadora a otra.

Como se indica en la figura, por lo general un módem de marcación telefónica va integrado a una computadora. Usamos el término *módem interno* para indicar un dispositivo integrado y el término *módem externo* para indicar un dispositivo separado físicamente.

10.15 QAM aplicada a la marcación telefónica

La modulación de amplitud en cuadratura también se usa con módems de marcación telefónica como una forma de maximizar la velocidad a la que se pueden enviar los datos. Para comprender cómo se logra esto, considere la figura 10.11 que muestra el ancho de banda disponible en una conexión de marcación telefónica. Como se indica en la figura, la mayoría de las conexiones telefónicas transfieren frecuencias entre 300 y 3000 Hz, pero una conexión dada tal vez no maneje bien los valores extremos. Por lo tanto, para garantizar una mejor reproducción y un menor ruido, los módems de marcación telefónica usan frecuencias entre 600 y 3000 Hz, lo cual significa que el ancho disponible es de 2400 Hz. Un esquema QAM puede incrementar de manera considerable la velocidad de los datos.

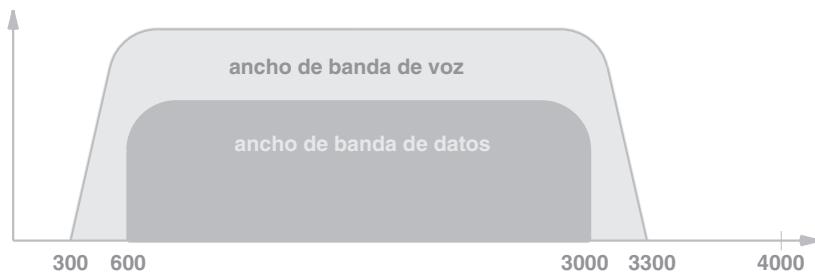


Figura 10.11 Ilustración del ancho de banda de voz y de datos en una conexión de marcación telefónica.

10.16 Módems de marcación telefónica V.32 y V.32bis

Como ejemplo de módems de marcación telefónica que usan la QAM, considere los estándares V.32 y V.32bis. La figura 10.12 ilustra la constelación de QAM para un módem V.32 que usa 32 combinaciones de desplazamiento de amplitud y desplazamiento de fase para lograr una velocidad de datos de 9600 bps en cada dirección.

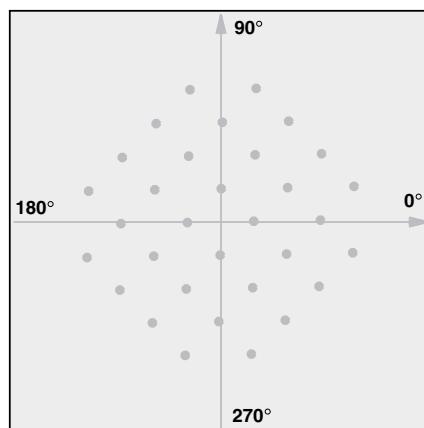


Figura 10.12 Ilustración de la constelación de QAM para un módem de marcación telefónica V.32.

Un módem V.32bis usa 128 combinaciones de desplazamiento de amplitud y de desplazamiento de fase para lograr una velocidad de datos de 14,400 bps en cada dirección. La figura 10.13 ilustra la constelación. Se necesita un análisis sofisticado de señales para detectar el pequeño cambio que ocurre desde un punto en la constelación, hasta un punto cercano.

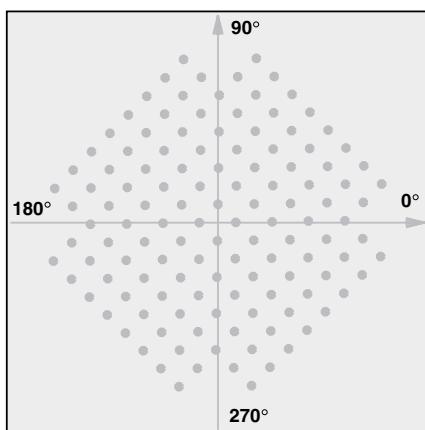


Figura 10.13 Ilustración de la constelación QAM para un módem de marcación telefónica V.32bis.

10.17 Resumen

Los sistemas de comunicaciones de larga distancia usan una onda portadora modulada para transmitir información. Para modular una portadora, se puede cambiar la amplitud, la frecuencia o la fase. La modulación de amplitud y la modulación de frecuencia son las formas más comunes que se utilizan con una entrada analógica.

Cuando se usa una señal digital como entrada, la modulación se conoce como codificación por desplazamiento. Al igual que con la modulación analógica, la codificación por desplazamiento cambia una portadora. Sin embargo, sólo se permite un conjunto fijo de posibilidades. Los diagramas de constelación se usan para representar las posibilidades de la codificación por desplazamiento de fase. Si el sistema permite una potencia de dos posibilidades, se pueden usar varios bits de entrada para seleccionar una posibilidad en cada punto del tiempo. La modulación de amplitud en cuadratura combina la codificación por desplazamiento de amplitud y la codificación por desplazamiento de fase para producir más posibilidades.

Un módem es un dispositivo de hardware que incluye circuitos para realizar tanto la modulación como la demodulación; para la comunicación dúplex se usa un par de módems. También existen módems ópticos de radiofrecuencia y de marcación telefónica. Como el ancho de banda es limitado, los módems de marcación telefónica usan esquemas de modulación de amplitud en cuadratura. Un módem V.32 usa 32 posibles combinaciones de desplazamientos de fase y cambios de amplitud; un módem V.32bis usa 128 combinaciones posibles.

EJERCICIOS

- 10.1** Mencione los tres tipos básicos de modulación analógica.
- 10.2** Al usar la modulación de amplitud, ¿tiene sentido que una portadora de 1 Hz sea modulada por una onda senoidal de 2 Hz? ¿Por qué sí o por qué no?
- 10.3** Use el teorema de Shannon para explicar por qué en la práctica los sistemas de modulación de amplitud mantienen la portadora cerca de la fuerza máxima.
- 10.4** ¿Cuál es la diferencia entre la codificación por desplazamiento y la modulación?
- 10.5** En la codificación por desplazamiento de fase, ¿es posible tener un desplazamiento de fase de 90° ?, ¿de 270° ?, ¿de 360° ? Dibuje un ejemplo para explicar su respuesta.
- 10.6** Busque en Web un diagrama de constelación para 32QAM. ¿Cuántos puntos se definen en cada cuadrante?
- 10.7** La figura 10.9 muestra una configuración dúplex con cuatro alambres, dos de los cuales se usan para transmitir en cada dirección. Argumente si debería ser mejor utilizar tres alambres.
- 10.8** En la pregunta anterior, ¿por qué son preferibles cuatro cables?
- 10.9** Asumiendo una relación señal-ruido de 30 dB, ¿cuál es la máxima velocidad de datos que puede lograrse para el ancho de banda de marcación telefónica ilustrado en la figura 10.11?

Contenido del capítulo

- 11.1 Introducción, 181
- 11.2 El concepto de multiplexación, 181
- 11.3 Los tipos básicos de multiplexación, 182
- 11.4 Multiplexación por división de frecuencias (FDM), 183
- 11.5 Uso de un rango de frecuencias por canal, 185
- 11.6 FDM jerárquica, 186
- 11.7 Multiplexación por división de longitud de onda (WDM), 187
- 11.8 Multiplexación por división de tiempo (TDM), 187
- 11.9 TDM síncrona, 188
- 11.10 Entramado utilizado en la versión de TDM del sistema telefónico, 189
- 11.11 TDM jerárquica, 190
- 11.12 Intervalos sin completar en la TDM síncrona, 190
- 11.13 TDM estadística, 191
- 11.14 Multiplexación inversa, 192
- 11.15 Multiplexación por división de código, 193
- 11.16 Resumen, 195

11

Multiplexación y demultiplexación (canalización)

11.1 Introducción

Los capítulos de esta parte del libro cubren los fundamentos de las comunicaciones de datos. El capítulo anterior habla sobre el concepto de modulación y explica cómo puede modularse una onda portadora para transportar información analógica o digital.

Este capítulo continúa la explicación sobre las comunicaciones de datos mediante una introducción a la multiplexación. El capítulo describe los beneficios y define los tipos básicos de multiplexación que se utilizan en las redes de computadoras y en Internet. El capítulo también explica cómo es que las portadoras moduladas proporcionan la base para muchos mecanismos de multiplexación.

11.2 El concepto de multiplexación

Utilizamos el término *multiplexación* para referirnos a la combinación de flujos de información de diversas fuentes para su transmisión sobre un medio compartido, y el de *multiplexor* para denotar un mecanismo que implementa tal combinación. De manera similar, usamos el término *demultiplexación* para referirnos a la separación de varios flujos de información combinados en flujos independientes, y *demultiplexor* para referirnos a un mecanismo que implementa dicha separación. La multiplexación

y demultiplexación no se limitan a cierto hardware o a flujos de bits individuales; en capítulos posteriores veremos que la idea de combinar y separar la comunicación es la base para muchos aspectos de las redes de computadoras. La figura 11.1 ilustra el concepto.

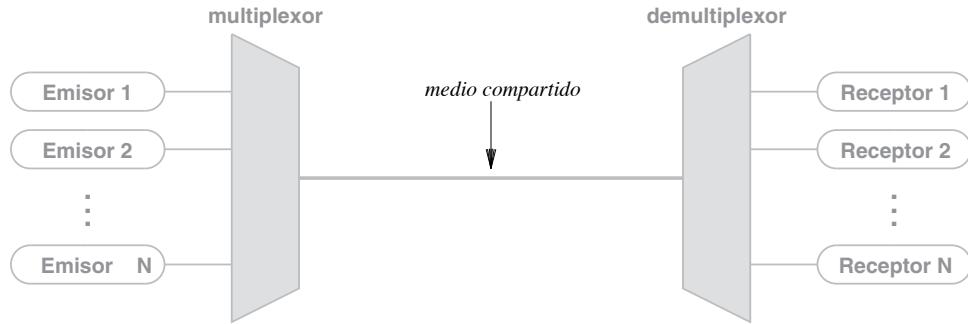


Figura 11.1 El concepto de multiplexación en el que pares independientes de emisores y receptores comparten un mismo medio de transmisión.

En la figura, cada emisor se comunica con un solo receptor. Aunque cada par mantiene una comunicación independiente, todos los pares comparten un solo medio de transmisión. El multiplexor combina la información de los emisores para su transmisión, de tal forma que el demultiplexor pueda separar la información para cada receptor. Todos los días los usuarios de redes experimentan diversas formas de multiplexación. Por ejemplo, varias computadoras de una residencia pueden usar un enrutador o *router* inalámbrico para comunicarse con un sitio de Internet. Cada computadora transporta una conversación independiente y todas las conversaciones se multiplexan a través de la conexión que hay entre la residencia y un proveedor de servicios de Internet.

11.3 Los tipos básicos de multiplexación

En la capa física, hay cuatro métodos básicos de multiplexación, cada uno de los cuales tiene un conjunto de variaciones e implementaciones.

- Multiplexación por división de frecuencias
- Multiplexación por división de longitud de onda
- Multiplexación por división de tiempo
- Multiplexación por división de código

Las multiplexaciones por división de tiempo y por división de frecuencias se usan ampliamente. La multiplexación por división de longitud de onda es una forma de multiplexación por división de frecuencias que se utiliza para la fibra óptica. La multiplexación por división de código es un método matemático que se utiliza en algunos sistemas de telefonía celular.

11.4 Multiplexación por división de frecuencias (FDM)

La *multiplexación por división de frecuencias (FDM)* es fácil de entender, ya que forma la base para la radiodifusión. El principio de ésta surge de la física de la transmisión: un conjunto de estaciones de radio pueden transmitir señales electromagnéticas en forma simultánea sin interferencia, siempre y cuando cada una de ellas usen un *canal* independiente (es decir, una frecuencia de portadora propia). Los sistemas de comunicaciones de datos aplican este principio al enviar al mismo tiempo varias ondas portadoras a través de un solo alambre de cobre, o al usar la multiplexación por división de longitud de onda para enviar múltiples frecuencias de luz a través de una fibra óptica. En el extremo receptor, un demultiplexor aplica un conjunto de filtros, cada uno de los cuales extrae un pequeño rango de frecuencias cercanas a las frecuencias portadoras. La figura 11.2 ilustra la estructura.

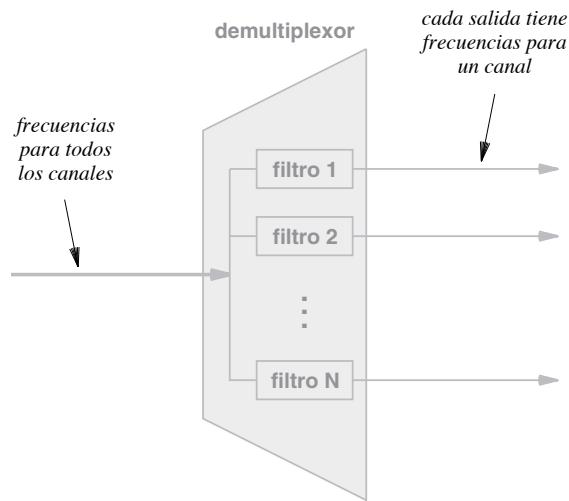


Figura 11.2 Ilustración de la demultiplexación FDM donde hay un conjunto de filtros, cada uno de los cuales selecciona las frecuencias para un canal y suprime las otras frecuencias.

Una idea clave es que los filtros utilizados en la FDM sólo analizan las frecuencias. Si a un par de emisor y receptor se le asigna una frecuencia portadora específica, el mecanismo de la FDM separará la frecuencia de las otras sin modificar la señal de ninguna otra forma. Por consiguiente, es posible usar cualquiera de las técnicas de modulación descritas en el capítulo 10 con cualquier portadora.

En conclusión:

Puesto que las ondas portadoras en frecuencias independientes no interfieren, la multiplexación por división de frecuencias proporciona a cada par de emisor y receptor un canal de comunicación privado, a través del cual puede usarse cualquier esquema de modulación.

La ventaja más importante de la FDM se debe a que varios pares de entidades en comunicación usan un medio de transmisión al mismo tiempo. Imaginamos que la FDM proporciona a cada par una ruta de transmisión privada, como si tuvieran un medio físico de transmisión independiente. La figura 11.3 ilustra este concepto.

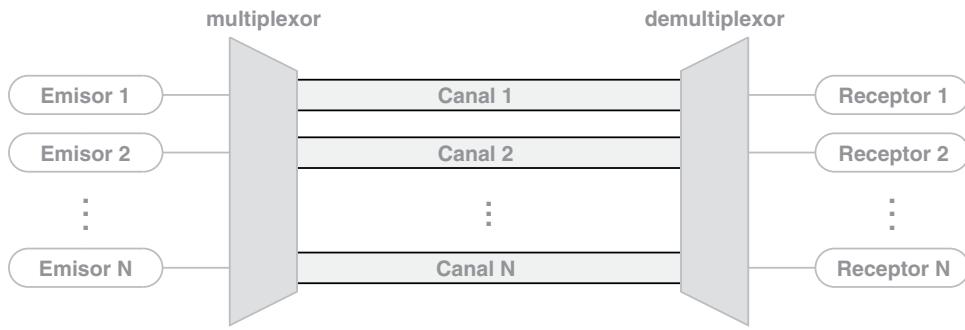


Figura 11.3 La vista conceptual de la multiplexación por división de frecuencias (FDM), que proporciona un conjunto de canales independientes.

Desde luego que en la práctica cualquier sistema de FDM impone límites en cuanto al conjunto de frecuencias que pueden usarse para los canales. Si las frecuencias de dos canales están relativamente cerca, puede haber interferencia. Además, el hardware de demultiplexación que recibe una señal combinada debe ser capaz de dividir esa señal en portadoras independientes. Para la difusión de radio en Estados Unidos, la *Comisión federal de comunicaciones (FCC)* regula las estaciones para asegurar que haya un *espacio* adecuado entre las frecuencias portadoras. Para los sistemas de comunicaciones de datos, los diseñadores siguen la misma metodología al elegir un conjunto de frecuencias portadoras con cierto espacio entre ellas. A este espacio se le conoce como *banda, o intervalo, de guarda*.

Como ejemplo de asignación de canales, considere el caso de la figura 11.4 que asigna 200 KHz a cada uno de los 6 canales con una banda de guarda de 20 KHz entre canales adyacentes.

Canal	Frecuencias utilizadas
1	100 KHz - 300 KHz
2	320 KHz - 520 KHz
3	540 KHz - 740 KHz
4	760 KHz - 960 KHz
5	980 KHz - 1180 KHz
6	1200 KHz - 1400 KHz

Figura 11.4 Ejemplo de asignación de frecuencias con una banda de guarda entre canales adyacentes.

Cuando se analiza un gráfico en el dominio de la frecuencia, la banda de guarda es claramente visible. La figura 11.5 contiene el gráfico para la asignación de la figura 11.4.

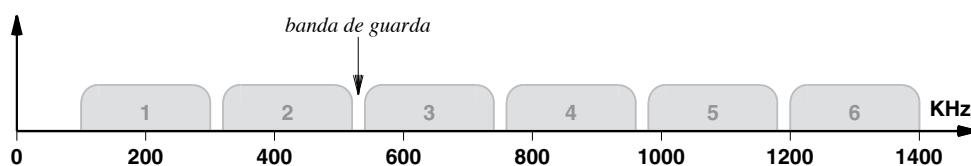


Figura 11.5 Un gráfico en el dominio de la frecuencia para la asignación de canales de la figura 11.4, con una banda de guarda visible entre los canales.

11.5 Uso de un rango de frecuencias por canal

Si una portadora usa una sola frecuencia, ¿por qué el ejemplo asigna bloques de frecuencias? Para comprender las causas, considere las características generales de la FDM:

- *Se ha usado por mucho tiempo.* La FDM es anterior a las comunicaciones de datos modernas; la idea de dividir el espectro electromagnético en canales surgió durante los primeros experimentos con las radiofrecuencias.
- *Se ha usado ampliamente.* La FDM se ha usado en difusión de radio y televisión, en televisión por cable y en el sistema telefónico celular AMPS.
- *Es analógica.* El hardware de multiplexación y demultiplexación de FDM recibe y distribuye señales analógicas. Incluso si una portadora se modula para contener información digital, el hardware de FDM trata a la portadora como una onda analógica.
- *Es muy versátil.* Puesto que filtra rangos de frecuencia sin analizar otros aspectos de las señales, la FDM resulta sumamente versátil.

La característica analógica tiene la desventaja de hacer la multiplexación por división de frecuencias susceptible al ruido y la distorsión,[†] pero tiene la ventaja de proveer flexibilidad. En particular, la mayoría de los sistemas de FDM asignan a cada par de emisor y receptor un rango de frecuencias junto con la posibilidad de elegir cómo pueden usarse esas frecuencias. Hay dos formas principales en las que los sistemas usan un rango de frecuencias.

- Incrementar la velocidad de datos
- Incrementar la inmunidad a la interferencia

Para incrementar la velocidad de datos en general, un emisor divide el rango de frecuencias del canal en K portadoras y envía $1/K$ de los datos a través de cada portadora. En esencia, un emisor realiza la multiplexación por división de frecuencias dentro del canal que se asignó. Algunos sistemas usan el término *asignación de subcanales* para referirse a la subdivisión.

[†] Los sistemas de comunicaciones de datos que usan FDM a menudo requieren un cable coaxial para ofrecer una mayor inmunidad al ruido.

Para incrementar la inmunidad a la interferencia, un emisor usa una técnica conocida como *espectro amplio*. Es posible usar varias formas de espectro amplio, pero la idea básica es dividir el rango del canal en K portadoras, transmitir los mismos datos a través de varios canales y permitir que un receptor use una copia de los datos que llegan con la menor cantidad de errores. Este esquema se utiliza mucho en las redes inalámbricas y funciona extremadamente bien en casos en los que el ruido podría interferir con algunas frecuencias en un momento dado.

11.6 FDM jerárquica

Parte de la flexibilidad de la FDM surge de la habilidad del hardware para cambiar de frecuencias. Si todo un conjunto de señales entrantes utiliza el rango de frecuencias entre 0 y 4 KHz, el hardware de multiplexación puede dejar la primera etapa como está, asignar la segunda etapa al rango de 4 KHz a 8 KHz, asignar la tercera etapa al rango de 8 KHz a 12 KHz, y así en lo sucesivo. La técnica forma la base para una jerarquía de multiplexores de FDM, cada uno de los cuales asignan sus entradas a una banda más grande y continua de frecuencias. La figura 11.6 ilustra el concepto de la *FDM jerárquica*.[†]

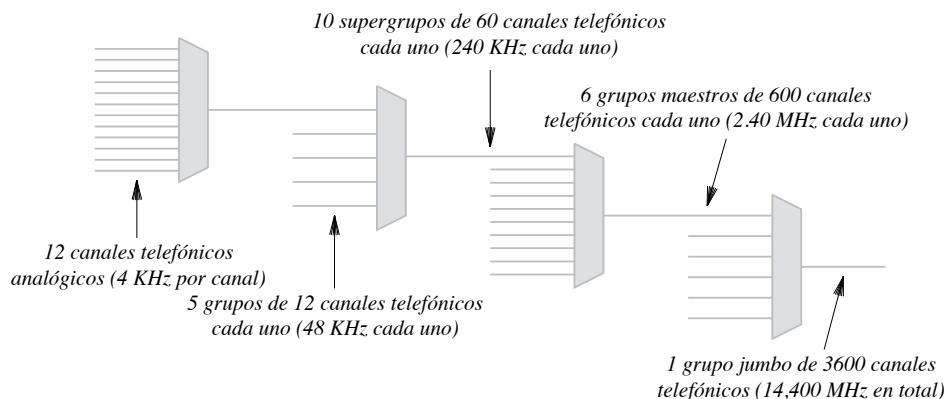


Figura 11.6 Ilustración de la jerarquía de FDM utilizada en el sistema telefónico.

Como se ilustra en la figura, la entrada primaria consiste en un conjunto de doce señales telefónicas analógicas, cada una de las cuales ocupa las frecuencias de 0 a 4 KHz. En la primera etapa, las señales se multiplexan en una sola señal conocida como *grupo*, que usa el rango de frecuencia de 0 a 48 KHz. En la siguiente etapa, cinco grupos se multiplexan en un solo *supergrupo* que usa las frecuencias de 0 a 240 KHz, y así en lo sucesivo. En la etapa final, 3600 señales telefónicas se multiplexan en una sola señal. Para resumir:

Es posible crear una jerarquía de multiplexación por división de frecuencias donde cada etapa acepte como entradas las salidas de la etapa anterior.

[†] En la práctica, se necesita un ancho de banda adicional para transportar los bits de entramado.

11.7 Multiplexación por división de longitud de onda (WDM)

El término *multiplexación por división de longitud de onda (WDM)* se refiere a la aplicación de la multiplexación por división de frecuencia en la fibra óptica; el término *multiplexación por división de longitud de onda densa (DWDM)* se usa para enfatizar que se emplean muchas longitudes de onda de luz. Las entradas y salidas de dicha multiplexación son longitudes de onda de luz, las cuales se denotan mediante la letra griega lambda (λ) y se conocen informalmente como *colores*. Para comprender cómo es que la multiplexación y la demultiplexación pueden funcionar con la luz, recordemos de la física básica que cuando la luz pasa a través de un prisma, los colores del espectro se dividen. Un prisma opera en el modo inverso también: si cada uno de los haces de colores de un conjunto se dirigen hacia un prisma en el ángulo correcto, el prisma combinará los haces para formar un solo haz de luz blanca. Por último, recuerde que lo que perciben los humanos como color es de hecho un rango de longitudes de onda de la luz.

Los prismas forman la base de la multiplexación y demultiplexación ópticas. Un multiplexor acepta haces de luz de diversas longitudes de onda y usa un prisma para combinarlos en un solo haz; un demultiplexor usa un prisma para separar las longitudes de onda. La figura 11.7 ilustra el concepto.

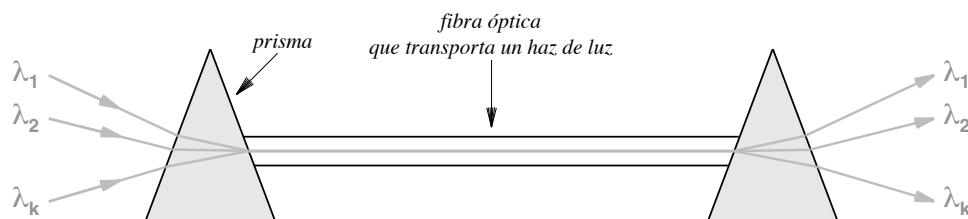


Figura 11.7 Ilustración de prismas utilizados para combinar y separar longitudes de ondas de luz en las tecnologías de multiplexación por división de longitud de onda.

En conclusión:

Cuando se aplica la multiplexación por división de frecuencia a la fibra óptica, se usan prismas para combinar o separar longitudes de ondas de luz individuales y el resultado se conoce como multiplexación por división de longitud de onda.

11.8 Multiplexación por división de tiempo (TDM)

La principal alternativa a la FDM se conoce como *multiplexación por división de tiempo (TDM)*. La TDM es menos complicada que la FDM y no depende de propiedades especiales de la energía electromagnética. En su lugar, multiplexar el tiempo simplemente significa transmitir un elemento desde una fuente, luego transmitir un elemento desde otra fuente, y así en lo sucesivo. La figura 11.8 ilustra el concepto.

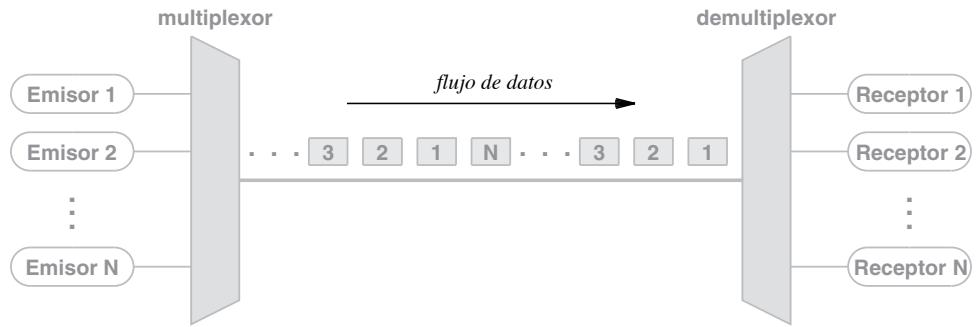


Figura 11.8 Ilustración del concepto de multiplexación por división de tiempo (TDM) con elementos de varias fuentes que se envían a través de un medio compartido.

11.9 TDM síncrona

La multiplexación por división de tiempo es un concepto amplio que se presenta de muchas formas y se usa abundantemente a través de Internet. Por consiguiente, el diagrama de la figura 11.8 es simplemente una vista conceptual; los detalles pueden variar. Por ejemplo, la figura muestra cómo se envían los elementos según el método *por turnos* (es decir, un elemento del emisor 1 seguido de un elemento del emisor 2, y así en lo sucesivo). Aunque algunos sistemas de TDM usan el orden por turnos, otros no lo hacen.

Un segundo detalle de la figura 11.8 (la pequeña laguna de tiempo que hay entre los elementos) no se aplica a todos los tipos de TDM. Vamos a extender la terminología del capítulo 9 para decir que un sistema de *TDM síncrona* envía elementos uno tras otro sin retraso. La figura 11.9 ilustra cómo funciona la TDM síncrona para un sistema de cuatro emisores.

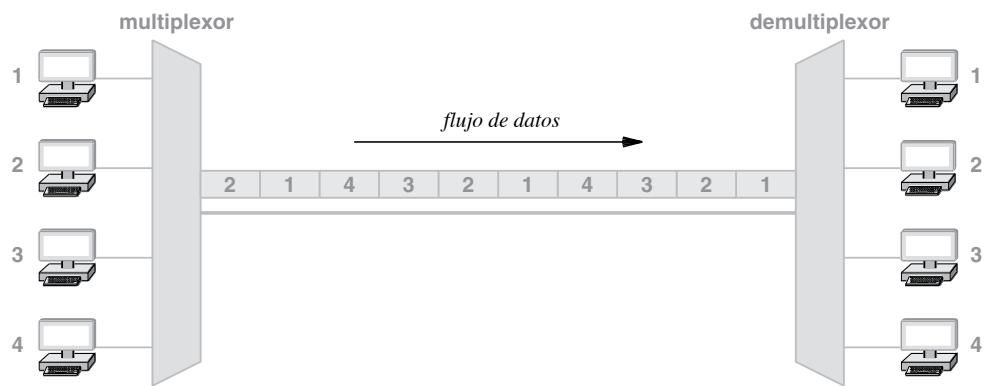


Figura 11.9 Ilustración de un sistema de TDM síncrona con cuatro emisores.

11.10 Entramado utilizado en la versión de TDM del sistema telefónico

Los sistemas telefónicos analógicos usan la TDM síncrona para multiplexar flujos digitales de varias llamadas telefónicas a través de un solo medio. De hecho, las compañías telefónicas usan el acrónimo TDM para hacer referencia a la forma específica de TDM utilizada para multiplexar llamadas telefónicas digitales.

Los estándares del sistema telefónico para la TDM incluyen una técnica interesante para asegurar que un demultiplexor permanezca sincronizado con el multiplexor. Para comprender por qué es necesaria la sincronización, observe que un sistema de TDM síncrona envía un elemento detrás de otro sin indicación de la salida hacia la que va dirigido ese elemento. Puesto que un demultiplexor no puede saber dónde comienza un elemento, una ligera diferencia en los relojes utilizados para sincronizar los bits puede provocar que un demultiplexor malinterprete el flujo de bits.

Para evitar una interpretación errónea, la versión de la TDM utilizada en el sistema telefónico incluye un *canal de entramado* adicional como entrada. En vez de tomar un intervalo completo, el entramado inserta un solo bit en el flujo de cada turno. Junto con los otros canales, un demultiplexor extrae los datos del canal de entramado y verifica que haya bits 0 y 1 alternantes. La idea es que si un error provoca que un demultiplexor pierda un bit, es muy probable que la verificación del entramado detecte el error y permita reiniciar la transmisión. La figura 11.10 ilustra el uso de los bits de entramado.

Para resumir:

El mecanismo de TDM síncrona utilizado para las llamadas digitales incluye un bit de entramado al principio de cada turno. La secuencia de entramado de los dígitos 1 y 0 alternantes asegura que un demultiplexor permanezca sincronizado o detecte el error.

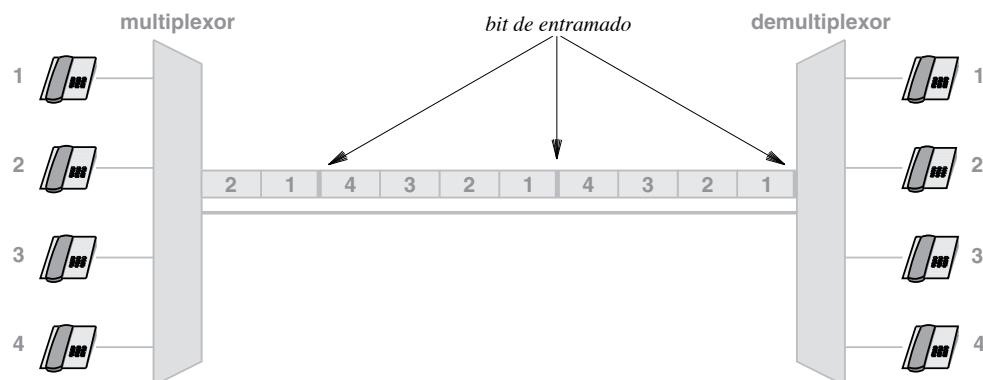


Figura 11.10 Ilustración de la TDM síncrona utilizada por el sistema telefónico, donde un bit de entramado antecede a cada turno de intervalos.

11.11 TDM jerárquica

Al igual que un sistema de multiplexación por división de frecuencia, un sistema TDM puede distribuirse en una jerarquía. La diferencia es que cada etapa sucesiva de una jerarquía de TDM usa N veces la velocidad de bits, mientras que cada etapa sucesiva de una jerarquía de FDM usa N veces las frecuencias. Se agregan a los datos bits de entramado adicionales, lo que significa que la velocidad de bits de cada capa sucesiva de la jerarquía es ligeramente mayor que el tráfico promedio de voz. Compare la jerarquía de TDM de ejemplo de la figura 11.11 con el ejemplo de FDM en la figura 11.6 de la página 186.

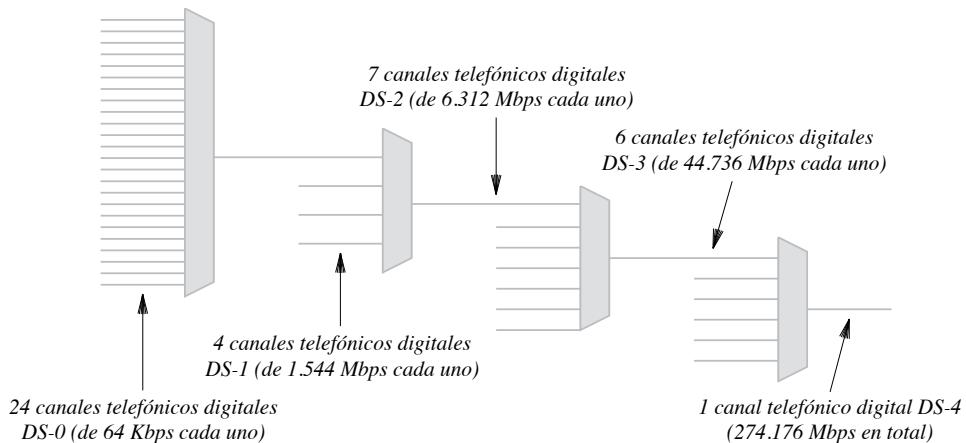


Figura 11.11 Ilustración de la jerarquía de TDM que se utiliza en el sistema telefónico.

11.12 Intervalos sin completar en la TDM síncrona

La TDM síncrona funciona bien si cada fuente produce datos a una velocidad fija y uniforme equivalente a $1/N$ de la capacidad del medio compartido. Por ejemplo, si una fuente corresponde a una llamada telefónica digital, los datos llegarán a una velocidad uniforme de 64 Kbps. Como lo señala el capítulo 9, muchas fuentes generan datos en ráfagas con un tiempo de inactividad entre ellas, pero esto no funciona bien con un sistema de TDM síncrona. Para entender por qué, considere el ejemplo en la figura 11.12.

En la figura, las fuentes de la izquierda producen elementos de datos al azar. Por consiguiente, el multiplexor síncrono deja un intervalo sin completar si la fuente correspondiente no ha producido un elemento para cuando debe enviarse ese intervalo. En la práctica, desde luego, un intervalo no puede estar vacío debido a que el sistema debe seguir transmitiendo datos. Por lo tanto, a ese intervalo se le asigna un valor (como cero) y se establece un bit adicional para indicar que el valor no es válido.

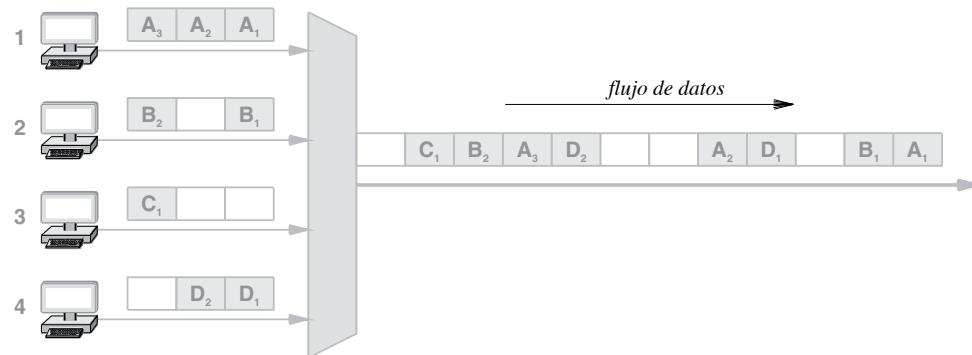


Figura 11.12 Ilustración de un sistema de TDM síncrona que deja los intervalos sin completar cuando una fuente no tiene un elemento de datos a tiempo

11.13 TDM estadística

¿Cómo puede un sistema de multiplexación TDM hacer un uso más eficiente de un medio compartido? Una técnica para incrementar la velocidad de datos en general se conoce como *TDM estadística* o *multiplexación estadística*.[†] La terminología es complicada, pero la técnica es simple: seleccionar elementos para transmitirlos por turnos, pero en vez de dejar un intervalo sin completar, omitir las fuentes que no tengan datos listos. Al eliminar los intervalos sin utilizar, la TDM estadística requiere menos tiempo para enviar la misma cantidad de datos. Por ejemplo, la figura 11.13 ilustra cómo un sistema de TDM estadística envía los datos de la figura 11.12 en sólo 8 intervalos en vez de 12.

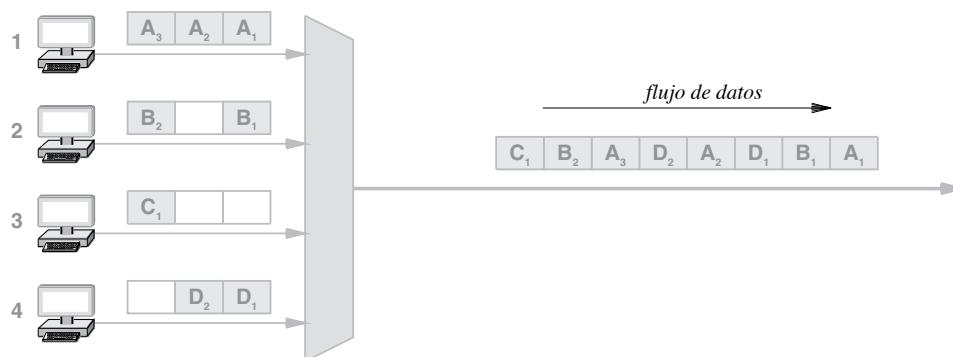


Figura 11.13 Ilustración que muestra cómo la multiplexación estadística evita intervalos sin completar y requiere menos tiempo para enviar los datos.

[†] Algunos libros usan el término *multiplexación asíncrona por división de tiempo*.

Aunque evita los intervalos sin completar, la multiplexación estadística incurre en una sobrecarga adicional. Para entender el porqué, considere la demultiplexación. En un sistema de TDM síncrona, un demultiplexor sabe que cada $N^{\text{ésimo}}$ intervalo corresponde a cierto receptor. En un sistema de multiplexación estadístico los datos en un intervalo dado pueden corresponder a cualquier receptor. Por ende, además de los datos, cada intervalo debe contener la identificación del receptor al que se le van a enviar. Los capítulos posteriores describen los mecanismos de identificación que se utilizan con la multiplexación estadística en las redes de commutación de paquetes y en Internet.

11.14 Multiplexación inversa

Un giro interesante en la multiplexación se da en casos en los que la única conexión entre dos puntos consiste en varios medios de transmisión, pero ningún medio tiene una velocidad de bits suficiente. En el entorno de Internet, los proveedores de servicios necesitan velocidades de bits mayores a las disponibles. Por ejemplo, un ISP puede necesitar una capacidad de 100 Gbps pero sólo adquiere conexiones que operen a 10 Gbps.[†] Para resolver el problema, se usa la multiplexación inversa: enviar datos a través de diez circuitos de 10 Gbps en paralelo. En general, la multiplexación inversa distribuye la entrada digital de alta velocidad a través de varios circuitos de menor velocidad y combina los resultados en el extremo receptor. La figura 11.14 ilustra el concepto.

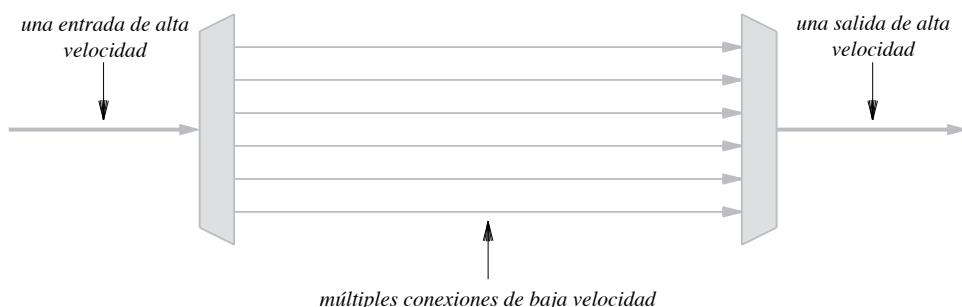


Figura 11.14 Ilustración de la multiplexación inversa, donde una sola entrada digital de alta capacidad se distribuye a través de conexiones de menor capacidad para la transmisión y luego se recombinan para formar una copia de la entrada.

En la práctica, un *multiplexor inverso* no puede construirse con sólo conectar las piezas de un multiplexor convencional al revés. En su lugar hay que diseñar hardware para que el emisor y el receptor estén de acuerdo en la forma en que se distribuirán los datos de entrada a través de las conexiones de menor velocidad. Y lo que es más importante, para asegurar que todos los datos se entreguen en el mismo orden en el que llegaron, el sistema debe estar diseñado para manejar casos donde una o más de las conexiones de baja velocidad tengan una mayor latencia que otras. A pesar de su complejidad, la multiplexación inversa se usa mucho en Internet.

[†] La multiplexación inversa también se usa por razones económicas: en algunos casos, el costo de N circuitos de capacidad inferior es menor que el costo de un solo circuito de alta capacidad.

11.15 Multiplexación por división de código

Una forma final de multiplexación que se utiliza en partes del sistema telefónico celular y para cierta comunicación entre satélites se conoce como *multiplexación por división de código* (CDM). La versión específica de CDM que se utiliza en los teléfonos celulares se conoce como *acceso múltiple por división de código* (CDMA).

A diferencia de la FDM y la TDM, la CDM no depende de las propiedades físicas, como la frecuencia o el tiempo, sino de una idea matemática interesante: los valores de los espacios vectoriales ortogonales pueden combinarse y separarse sin interferencia. La forma específica utilizada en la red telefónica es más fácil de entender. A cada emisor se le asigna un código binario único C_i , que se conoce como *secuencia de chips*. Las secuencias de chips se seleccionan como vectores ortogonales (es decir, que el producto punto de dos secuencias de chips cualesquiera sea cero). En cualquier punto del tiempo, cada emisor tiene un valor para transmitir, V_i . Cada uno de los emisores multiplica $C_i \times V_i$ y transmite los resultados. En esencia, los emisores transmiten al mismo tiempo y los valores se suman. Para extraer el valor V_i , un receptor multiplica la suma por C_i .

Para aclarar el concepto, considere el siguiente ejemplo. Para que éste sea fácil de entender, usaremos una secuencia de chips de sólo dos bits de longitud y valores de datos que tengan cuatro bits de longitud. Consideremos la secuencia de chips como si fuera un vector. La figura 11.15 muestra una lista de los valores.

Emisor	Secuencia de chips	Valor de datos
A	1 0	1 0 1 0
B	1 1	0 1 1 0

Figura 11.15 Valores de ejemplo para usar con la multiplexación por división de código.

El primer paso consiste en convertir los valores binarios en vectores de bits que usen -1 para representar 0:

$$C_1 = (1, -1) \quad V_1 = (1, -1, 1, -1) \quad C_2 = (1, 1) \quad V_2 = (-1, 1, 1, -1)$$

Al multiplicar $C_i \times V_i$ y $C_2 \times V_2$ se produce:

$$((1, -1), (-1, 1), (1, -1), (-1, 1)) \quad ((-1, -1), (1, 1), (1, 1), (-1, -1))$$

Si pensamos en los valores resultantes como una secuencia de intensidades de señal a transmitir al mismo tiempo, la señal resultante será la suma de las dos señales:

$$\begin{array}{r}
 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
 + & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\
 \hline
 0 & -2 & 0 & 2 & 2 & 0 & -2 & 0
 \end{array}$$

Un receptor trata la secuencia como un vector, calcula el producto del vector y la secuencia de chips, luego trata el resultado como una secuencia y convierte el resultado en binario mediante la interpretación de los valores positivos como 1 binario y los valores negativos como 0 binario. Así, el receptor A calcula:

$$(1, -1) \cdot ((0, -2), (0, 2), (2, 0), (-2, 0))$$

para obtener:

$$((0+2), (0-2), (2+0), (-2+0))$$

Si se interpreta el resultado como una secuencia se produce:

$$2 \quad -2 \quad 2 \quad -2$$

lo que se convierte en el valor binario:

$$1 \ 0 \ 1 \ 0$$

Cabe mencionar que 1010 es el valor correcto de V_1 . Mientras tanto, el receptor B extraerá V_2 de la misma transmisión.

Podría parecer que la CDM ofrece poca ventaja con respecto a la TDM. De hecho, la CDM resulta ser un tanto ineficiente debido a que se requiere de una secuencia de chips más grande, aun cuando sólo unos cuantos emisores transmitan durante un intervalo dado. Por consiguiente, cuando el uso es limitado, la TDM estadística funciona mejor que la CDM.

Las ventajas de la CDM surgen de su habilidad de escalarse y de ofrecer un menor retraso en una red con mucho tráfico. Para ver por qué es importante un menor retraso, considere un sistema de TDM estadística. Una vez que el emisor transmite, un multiplexor de TDM permite que otros $N - 1$ emisores transmitan antes de dar al primer emisor otro turno. Por lo tanto, si todos los emisores están activos, el retraso potencial entre transmisiones sucesivas de un emisor dado puede ser alto. Sin embargo, en un sistema con CDM, un emisor puede transmitir al mismo tiempo que otros emisores, lo que significa que el retraso es menor. La CDM es especialmente atractiva para un servicio telefónico, ya que el retraso mínimo entre transmisiones es esencial para transmitir voz de alta calidad. Para resumir:

Cuando una red tiene mucho tráfico, la CDM incurre en un retraso menor que la TDM estadística.

11.16 Resumen

La multiplexación es un concepto fundamental en las comunicaciones de datos. Un mecanismo de multiplexación permite que pares de emisores y receptores se comuniquen a través de un medio compartido. Un multiplexor envía entradas de muchos emisores a través de un medio compartido, y un demultiplexor separa y entrega los elementos.

Hay cuatro metodologías básicas para la multiplexación: división de frecuencia, división de tiempo, división de longitud de onda y división de código. La *multiplexación por división de frecuencia* (FDM) permite la comunicación simultánea a través de varios canales, cada uno de los cuales corresponde a una frecuencia de radiación electromagnética independiente. La *multiplexación por división de longitud de onda* (WDM) es una forma de multiplexación por división de frecuencia que envía frecuencias de luz, conocidas como longitudes de onda, a través de una fibra óptica.

La *multiplexación por división de tiempo* (TDM) envía un elemento a la vez a través del medio compartido. Un sistema de TDM síncrona transmite elementos sin tiempo inactivo entre ellos, por lo general usando la selección por turnos. Un sistema de TDM estadística evita intervalos vacíos al omitir cualquier emisor que no tenga listo un elemento para enviar durante su turno.

La *multiplexación por división de código* (CDM) usa una combinación matemática de códigos que permite que varios emisores transmitan al mismo tiempo sin interferencia. Las principales ventajas de la CDM surgen de la habilidad de escalar con un mínimo retraso.

EJERCICIOS

- 11.1** Mencione un ejemplo de multiplexación en un sistema de comunicaciones no electrónico.
- 11.2** ¿Cuáles son los cuatro tipos básicos de multiplexación?
- 11.3** ¿Cómo es que la FDM usa la radiación electromagnética?
- 11.4** ¿Qué es una banda de guarda?
- 11.5** Un sistema de FDM puede asignar un rango de frecuencias a cada canal. ¿Con qué tipo de modulación para cada portadora resulta imprescindible el uso de un rango?
- 11.6** Explique cómo puede usarse un rango de frecuencias para incrementar la velocidad de datos.
- 11.7** En un sistema de FDM jerárquica, explique cómo se divide un canal de alta capacidad en subcanales.
- 11.8** ¿Cuál es el mecanismo clave que se utiliza para combinar o separar longitudes de ondas de luz en un sistema WDM?
- 11.9** ¿Un sistema de TDM debe utilizar el servicio por turnos? Explique.
- 11.10** Explique por qué el entramado y la sincronización son importantes en un sistema de TDM.
- 11.11** En un sistema de TDM jerárquica, ¿a qué velocidad de bits necesita operar la salida de un nivel dado? (Exprese la respuesta en términos del número y la velocidad de bits de las entradas).
- 11.12** Suponga que N usuarios compiten usando un sistema de TDM estadística y suponga que el transporte físico utilizado puede enviar K bits por segundo. ¿Cuál es la velocidad de datos mínima y máxima que puede experimentar un solo usuario?

- 11.13** Suponga que un circuito OC-12 cuesta el veinte por ciento de lo que cuesta un circuito OC-48. ¿Qué tecnología de multiplexación puede usar un ISP para reducir el costo de enviar datos a la velocidad del OC-48? Explique.
- 11.14** Busque en la Web cuál es la longitud de una secuencia de chips utilizada en los sistemas telefónicos CDMA.
- 11.15** De las cuatro técnicas básicas de multiplexación, ¿es CDM siempre la mejor? Explique.

Contenido del capítulo

- 12.1 Introducción, 199
- 12.2 Tecnologías de acceso a Internet: flujo ascendente y descendente, 199
- 12.3 Tecnologías de acceso de banda estrecha y banda ancha, 200
- 12.4 El bucle local y la ISDN, 202
- 12.5 Tecnologías de la línea de suscriptor digital (DSL), 202
- 12.6 Características del bucle local y adaptación, 203
- 12.7 La velocidad de datos de las líneas ADSL, 204
- 12.8 Instalación de ADSL y filtros divisores, 205
- 12.9 Tecnologías de módems de cable, 205
- 12.10 La velocidad de datos de los módems de cable, 206
- 12.11 Instalación de un módem de cable, 206
- 12.12 Fibra híbrida coaxial, 207
- 12.13 Tecnologías de acceso que emplean la fibra óptica, 208
- 12.14 Terminología de módems de extremo cercano y extremo lejano, 208
- 12.15 Tecnologías de acceso inalámbricas, 209
- 12.16 Conexiones de alta capacidad en el núcleo de Internet, 209
- 12.17 Terminación de circuitos, DSU/CSU y NIU, 210
- 12.18 Estándares de telefonía para circuitos digitales, 211
- 12.19 Terminología DS y velocidades de datos, 212
- 12.20 Circuitos de mayor capacidad (estándares STS), 212
- 12.21 Estándares de portadora óptica, 213
- 12.22 El sufijo C, 213
- 12.23 Red óptica síncrona (SONET), 214
- 12.24 Resumen, 215

12

Tecnologías de acceso e interconexión

12.1 Introducción

Cada capítulo de esta sección analiza uno de los aspectos fundamentales de las comunicaciones de datos. El capítulo anterior habla sobre la multiplexación y el concepto de una jerarquía de multiplexación; describe los esquemas de multiplexación por división de tiempo y por división de frecuencias que usan las compañías telefónicas para la telefonía digital.

Este capítulo concluye la explicación sobre las comunicaciones de datos, mediante el análisis de dos características utilizadas en Internet. Primera, el capítulo habla sobre las tecnologías de acceso, como DSL, módems de cable y líneas T1 que se usan para conectar residencias y negocios a Internet. Segunda, el capítulo considera los circuitos digitales de alta capacidad que se usan en el núcleo de Internet y amplía la explicación sobre la jerarquía de multiplexación del sistema telefónico, además de dar ejemplos de circuitos que ofrecen las portadoras comunes para los negocios y los proveedores de servicios de Internet (ISP). La explicación se concentra en aquellos aspectos de las tecnologías que están relacionados con las comunicaciones de datos, incluyendo la multiplexación y las velocidades de datos.

12.2 Tecnologías de acceso a Internet: flujo ascendente y descendente

Tecnología de acceso a Internet se refiere a un sistema de comunicaciones de datos que conecta a un *suscriptor* de Internet (por lo general una residencia o un negocio privado) con un *proveedor de servicios de Internet (ISP)*, como una compañía telefónica o de televisión por cable. Para comprender cómo se diseña la tecnología de acceso, hay que saber que la mayoría de los usuarios de Internet siguen un

patrón *asimétrico*. Un suscriptor residencial común recibe más datos de Internet de los que envía. Por ejemplo, para ver una página Web, un navegador envía un URL que consiste en unos cuantos bytes de datos. En respuesta, el servidor Web envía contenido que puede consistir en miles de bytes de texto o una imagen que puede estar formada por decenas de miles de bytes. Una empresa que opera un servidor Web puede tener el patrón de tráfico opuesto: enviar más datos de los que recibe. En conclusión:

Puesto que un suscriptor residencial común recibe mucha más información de la que envía, las tecnologías de acceso a Internet están diseñadas para transferir más datos en una dirección que en otra.

La industria de las redes usa el término *flujo descendente* para referirse a los datos que viajan de un proveedor de servicios en Internet hacia un suscriptor, y *flujo ascendente* para referirse a los datos que viajan de un suscriptor hacia un proveedor de servicios. La figura 12.1 ilustra las definiciones.

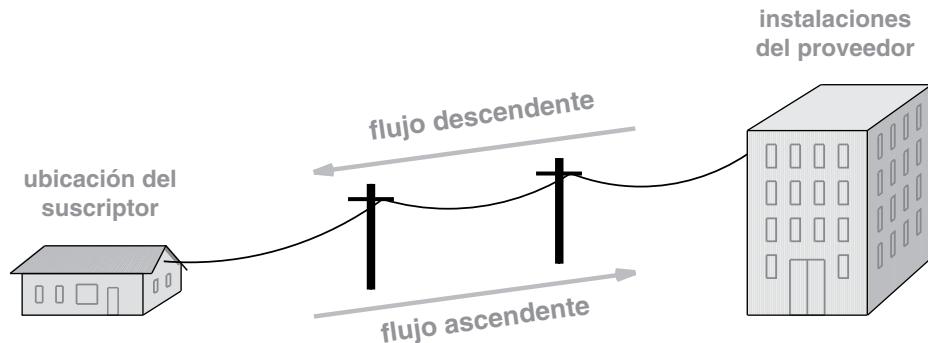


Figura 12.1 Las direcciones de los flujos ascendente y descendente usados por las tecnologías de acceso a Internet.

12.3 Tecnologías de acceso de banda estrecha y banda ancha

Existe una variedad de tecnologías que se utilizan para proveer acceso a Internet. Pueden dividirse en dos amplias categorías con base en la velocidad de datos que proporcionan:

- Banda estrecha o *narrowband*
- Banda ancha o *broadband*

Aunque el capítulo 6 explica la diferencia entre el ancho de banda de un medio de transmisión y la velocidad de datos, la terminología usada para acceder a las redes no sigue esta distinción. En su defecto, la industria de las redes por lo general usa el término *ancho de banda de red* para referirse a la velocidad de datos. De esta forma, los términos *banda estrecha* y *banda ancha* reflejan la práctica de la industria.

12.3.1 Tecnologías de banda estrecha

Por lo general, la *banda estrecha* se refiere a las tecnologías que transmiten datos a una velocidad de hasta 128 Kbps. Por ejemplo, la velocidad máxima de datos que puede lograrse a través de una conexión telefónica analógica con la tecnología de módem más sofisticada y las líneas telefónicas menos ruidosas es de 56 Kbps. Por lo tanto, la marcación telefónica se considera como una tecnología de banda estrecha. De manera similar, los circuitos analógicos que usan módems, circuitos digitales de velocidades más bajas y algunos de los servicios de datos que ofrecen las compañías telefónicas (como la *red digital de servicios integrados* o ISDN) son de banda estrecha. La figura 12.2 sintetiza las principales tecnologías de acceso de banda estrecha.

Banda estrecha
Conexiones de marcación telefónica
Círculo arrendado que usa módems
Circuitos de datos de T1 fraccionada
ISDN y otros servicios de datos de compañías de telecomunicaciones

Figura 12.2 Las principales tecnologías de banda estrecha utilizadas para acceder a Internet.

12.3.2 Tecnologías de banda ancha

Por lo general, el término *banda ancha* se refiere a las tecnologías que ofrecen velocidades de datos elevadas, pero el límite exacto entre la banda ancha y la banda estrecha no es muy claro. Muchos profesionales sugieren que las tecnologías de banda ancha deben transmitir más de 1 Mbps. Sin embargo, algunos proveedores como las compañías telefónicas usan el término *banda ancha* para referirse a cualquier servicio que ofrezca una velocidad más alta que la marcación telefónica. Por lo tanto, las compañías telefónicas afirman que cualquier tecnología que ofrezca 128 Kbps o más puede clasificarse como banda ancha. La figura 12.3 sintetiza las principales tecnologías de acceso de banda ancha.

Banda ancha
Tecnologías de DSL
Tecnologías de módem de cable
Tecnologías de acceso inalámbrico
Circuitos de datos a velocidad de T1 o mayor

Figura 12.3 Las principales tecnologías de acceso a Internet de banda ancha.

12.4 El bucle local y la ISDN

El término *línea de suscriptor local* o *bucle local* describe la conexión física que existe entre la *oficina central* (*CO*) de una compañía telefónica y la ubicación de un suscriptor. Para comprender cómo puede usarse un bucle local, es importante considerarlo como una entidad independiente del resto del sistema telefónico. Aunque el sistema telefónico en general está diseñado para proveer a cada llamada telefónica 4 KHz de ancho de banda, la parte correspondiente al bucle local consiste en un par de alambres de cobre y a menudo tiene un ancho de banda potencial mucho mayor. En particular, el bucle local para un suscriptor que está cerca de una CO puede manejar frecuencias superiores a 1 MHz.

A medida que las redes de datos se volvieron más importantes, las compañías telefónicas exploraron formas de usar el bucle local para proveer una comunicación de datos de mayor velocidad. Uno de los primeros esfuerzos de las compañías telefónicas por brindar servicios digitales de gran escala a los suscriptores, se ofrece bajo el nombre de *red digital de servicios integrados* (*ISDN*). Desde el punto de vista de un suscriptor, una ISDN ofrece tres canales digitales independientes, designados como *B*, *B* y *D* (por lo general se escribe como $2B + D$). Los dos canales *B*, cada uno de los cuales opera a una velocidad de 64 Kbps, están diseñados para transportar voz digitalizada, datos o video comprimido; el canal *D*, que opera a 16 Kbps, se usa como canal de control. En general, un suscriptor usa el canal *D* para solicitar servicios, los que luego se suministran a través de los canales *B* (por ejemplo, una llamada telefónica que usa voz digital). Ambos canales *B* pueden combinarse o *unirse* para producir un solo canal con una velocidad de datos efectiva de 128 Kbps. Cuando se propuso la tecnología ISDN por primera vez, la velocidad de 128 Kbps parecía mucho mayor que la que ofrecían los módems de marcación telefónica. Las tecnologías de bucle local recientes proporcionan velocidades de datos mayores a un menor costo, relegando a la ISDN a unos cuantos casos especiales.

12.5 Tecnologías de la línea de suscriptor digital (DSL)

La *línea de suscriptor digital* (*DSL*) es una de las principales tecnologías que se utilizan para proveer servicios de comunicación de datos de alta velocidad a través de un bucle local. La figura 12.4 enumera las variantes de la DSL. Puesto que los nombres difieren sólo en la primera palabra, el conjunto se conoce colectivamente mediante el acrónimo *xDSL*.

Nombre	Expansión	Uso general
ADSL	DSL asimétrica	Clientes residenciales
ADSL2	DSL asimétrica versión 2	Aproximadamente tres veces más rápida
SDSL	DSL simétrica	Empresas que exportan datos
HDSL	DSL de tasa de bits alta	Empresas que están hasta a 3 millas de distancia
VDSL	DSL de tasa de bits muy alta	Versión propuesta para 52 Mbps

Figura 12.4 Variantes principales de la DSL que se conocen colectivamente como *xDSL*.

ADSL es la variante más implementada a nivel mundial, y es la que utiliza la mayoría de los clientes residenciales. ADSL usa la multiplexación por división de frecuencias para dividir el ancho de banda del bucle local en tres regiones. Una de las regiones corresponde al servicio telefónico analógico tradicional, que se conoce en la industria como *servicio telefónico básico (POTS)*, y las otras dos regiones proporcionan comunicación de datos. En conclusión:

Puesto que utilizan la multiplexación por división de frecuencias, la ADSL y el servicio telefónico analógico tradicional (POTS) pueden usar los mismos alambres al mismo tiempo.

La figura 12.5 ilustra cómo la ADSL divide el ancho de banda.

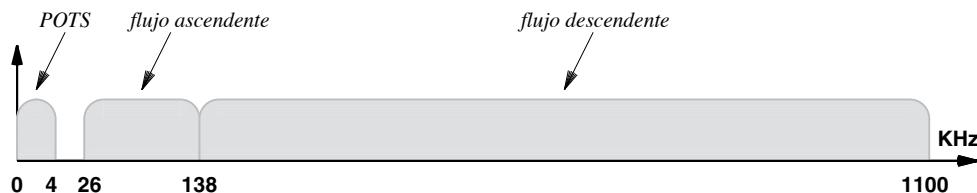


Figura 12.5 Una ilustración de cómo ADSL divide el ancho de banda disponible del bucle local.

En la figura, el eje x no es lineal. Si lo fuera, la región de 4 KHz reservada para POTS no sería visible, ni lo sería la banda de guarda de 22 KHz entre el POTS y la región de flujo ascendente.

12.6 Características del bucle local y adaptación

La tecnología ADSL es compleja debido a que no hay dos bucles locales que tengan características eléctricas idénticas. En su lugar, la habilidad de transportar señales depende de la distancia, del diámetro del alambre utilizado y del nivel de interferencia eléctrica. Por ejemplo, considere a dos suscriptores que viven en diferentes partes de una ciudad. Si la línea telefónica que llega al primer suscriptor pasa cerca de una estación de radio comercial, la señal de la estación provocará interferencia en la frecuencia utilizada por ésta. Si el segundo suscriptor no vive cerca de la misma estación de radio, la frecuencia que usa la estación de radio puede funcionar bien para los datos de la línea de ese suscriptor. Sin embargo, el segundo suscriptor podría experimentar interferencia en otra frecuencia. Por consiguiente, los diseñadores de ADSL no podrían elegir un conjunto específico de frecuencias de portadora o técnicas de modulación que funcionaran bien en todos los bucles locales.

Para adaptarse a las diferencias en las características del bucle local, la ADSL es *adaptativa*. Es decir, cuando un par de módems ADSL se encienden, sondean la línea entre ellos para averiguar sus características y luego aceptan comunicarse usando técnicas que sean óptimas para la línea. En especial, ADSL

usa un esquema conocido como *modulación por multitone discreto (DMT)* que combina las técnicas de multiplexación por división de frecuencias y multiplexación inversa.

La multiplexación por división de frecuencias de la DMT se implementa dividiendo el ancho de banda en 286 frecuencias independientes conocidas como *subcanales*;[†] se asignan 255 subcanales para la transmisión de datos de flujo descendente y 31 para la transmisión de datos de flujo ascendente. Dos de los canales de flujo ascendente se reservan para la información de control. En concepto, hay un “módem” independiente operando en cada subcanal, que tiene su propia portadora modulada. Las portadoras se separan en intervalos de 4.1325 KHz para evitar que las señales interfieran entre sí. Además, para garantizar que sus transmisiones no interfieran con las señales telefónicas analógicas, la tecnología ADSL evita usar el ancho de banda debajo de 26 KHz. Cuando la línea ADSL inicia, ambos extremos sondean las frecuencias disponibles para determinar qué frecuencias funcionan bien y cuáles experimentan interferencia. Además de seleccionar las frecuencias, los dos extremos evalúan la calidad de la señal en cada frecuencia y usan esa calidad para seleccionar un esquema de modulación. Si una frecuencia específica tiene una relación señal-ruido elevada, la ADSL selecciona un esquema de modulación que codifica muchos bits por baudio; por el contrario, si la calidad en una frecuencia dada es baja, la ADSL selecciona un esquema de modulación que codifique menos bits por baudio. Podemos resumir:

Puesto que las características eléctricas de los bucles locales varían, la ADSL usa una tecnología adaptativa en la que un par de módem sondean varias frecuencias en la línea que hay entre ellos, y seleccionan las frecuencias y las técnicas de modulación que produzcan resultados óptimos en esa línea.

12.7 La velocidad de datos de las líneas ADSL

¿Qué tan rápido puede operar una línea ADSL? Las ADSL pueden lograr una velocidad de flujo descendente de 8.448 Mbps en bucles locales cortos, además de una velocidad de flujo ascendente de 640 Kbps. Puesto que el canal de control de red obligatorio requiere 64 Kbps, la velocidad efectiva de flujo ascendente para los datos de usuario es de 576 Kbps. Bajo las mejores condiciones, la ADSL2 puede descargar a una velocidad cercana a los 20 Mbps.

Desde el punto de vista de un usuario, la adaptación tiene una propiedad interesante: ADSL no garantiza una velocidad de datos. En su lugar, sólo puede garantizar que funcionará tan bien como las condiciones de la línea se lo permitan. Los suscriptores que viven más alejados de una oficina central o cuyo bucle local pase cerca de fuentes de interferencia, experimentan menores velocidades de datos que los suscriptores que viven cerca de la oficina central y cuyo bucle local no pasa cerca de fuentes de interferencia. Por lo tanto, la velocidad de flujo descendente varía desde 32 Kbps hasta 8.448 Mbps, y la velocidad de flujo ascendente varía desde 32 hasta 640 Kbps.

Es importante entender que la velocidad de datos de ADSL sólo se aplica a la conexión del bucle local entre un suscriptor y la oficina central telefónica. Muchos otros factores afectan en general a las velocidades de datos que experimenta un usuario. Por ejemplo, cuando el usuario se contacta con un servidor Web, la velocidad de datos efectiva puede limitarse debido ya sea a la velocidad o la carga actual en el servidor, la tecnología de acceso utilizada para conectar el sitio del servidor con Internet, o las redes intermedias que hay entre la oficina central que corresponde al usuario y el proveedor que se encarga del servidor.

[†] El término *subcanal* surge debido a que algunas variantes de DSL dividen el ancho de banda en “canales” de 1.544 Mbps, cada uno de los cuales corresponde a un circuito T1, como veremos más adelante en este mismo capítulo.

12.8 Instalación de ADSL y filtros divisores

Aunque los teléfonos analógicos tradicionales operan a frecuencias inferiores a los 4 KHz, al levantar un auricular se puede generar ruido que interfiera con las señales de una DSL. Para brindar un aislamiento total, ADSL usa un dispositivo de FDM conocido como *filtro divisor* o *splitter* que divide el ancho de banda pasando las frecuencias bajas a una salida y las frecuencias altas a otra. Lo interesante es que el filtro divisor es *pasivo*, lo que significa que no requiere alimentación eléctrica. Por lo general el filtro se instala en el punto de entrada del bucle local hacia una residencia o negocio. Uno de los extremos del filtro se conecta al cableado del POTS y el otro lado se conecta a un módem ADSL. La figura 12.6 ilustra la conexión.

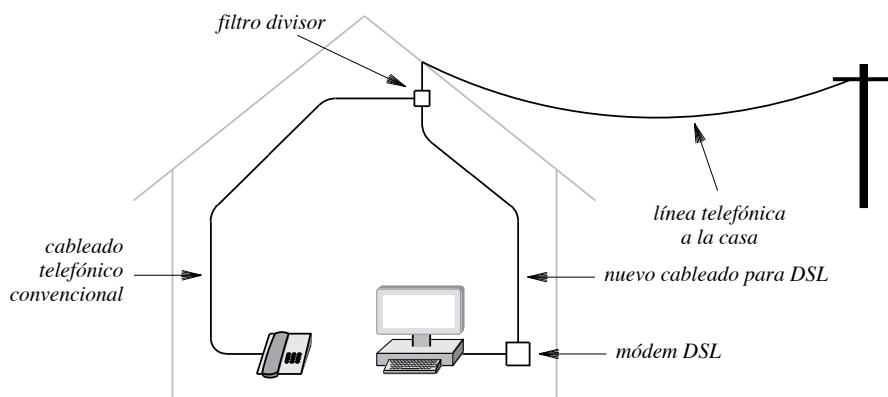


Figura 12.6 Ilustración de un filtro divisor y del cableado utilizado con ADSL.

Hay una variante interesante del cableado ADSL que se ha vuelto popular. Conocido algunas veces como *DSL lite*, este método alternativo no requiere de la instalación de un filtro divisor en la línea telefónica entrante. En su lugar, se usa el cableado para DSL que ya existe en el hogar, y se instala un filtro entre cada teléfono y este cableado. La ventaja de este método alternativo es que un suscriptor puede instalar el servicio DSL en cualquier toma de pared con tan sólo conectar un filtro divisor en la toma y luego un aparato telefónico al filtro.

12.9 Tecnologías de módems de cable

Aunque las tecnologías como ADSL ofrecen velocidades de datos mucho mayores de lo que se había pensado en un principio, el cableado del bucle local telefónico tiene limitaciones inherentes. El principal problema recae en las características eléctricas del cableado de par trenzado. La falta de protección hace al alambre susceptible a interferencias, lo cual degrada de manera considerable el rendimiento para algunos suscriptores. A medida que aumenta la demanda de mayores velocidades de bits, los esquemas de cableado alternativos se han vuelto más importantes. En consecuencia, se están desarrollando una variedad de tecnologías inalámbricas y alámbricas para usarse en el bucle local.

Una tecnología de acceso inalámbrica que resulta especialmente atractiva usa el cableado que ya está instalado para la *televisión por cable*.[†] El medio que se utiliza en los sistemas de cable es el cable coaxial, el cual tiene un ancho de banda alto y es menos susceptible a la interferencia magnética que el par trenzado. Además, los sistemas de televisión por cable usan la multiplexación por división de frecuencias (FDM) para transmitir muchos canales de entretenimiento al mismo tiempo.

Podríamos suponer que con tantos canales disponibles, un proveedor de cable podría usar un canal independiente para transmitir información digital a cada suscriptor. Es decir, configurar un par de *módems de cable* (uno en el centro CATV y el otro en la ubicación del suscriptor) para usar un canal determinado para la comunicación y multiplexar el canal dentro del cable junto con las señales de televisión.

A pesar del ancho de banda grande del que disponen los sistemas de CATV, el ancho de banda no es suficiente para manejar un esquema de multiplexación por división de frecuencias que extienda un canal a cada usuario. Para comprender por qué, observe que en un área metropolitana densa, un solo proveedor de cable puede tener millones de suscriptores. Como resultado, no es proporcional usar un canal independiente por cada suscriptor.

Para resolver el problema, los sistemas de cable combinan la FDM y la multiplexación estadística mediante la asignación de un canal de comunicación digital para un conjunto de suscriptores (por lo general, de un mismo vecindario). A cada suscriptor se le asigna una *dirección única* y cada mensaje que se envía a través del canal contiene la dirección a la que se envió. El módem de un suscriptor escucha la frecuencia asignada, pero antes de aceptar un mensaje, éste verifica que la dirección del mensaje coincida con la dirección asignada al suscriptor.

12.10 La velocidad de datos de los módems de cable

¿Qué tan rápido puede operar un módem de cable? En teoría, un sistema de cable puede soportar velocidades de datos de 52 Mbps en flujo descendente y de 512 Kbps en flujo ascendente. En la práctica, la velocidad puede ser mucho menor. Primero, la velocidad de datos de un módem de cable sólo depende de la comunicación entre la oficina de cable local y la ubicación del suscriptor. Segundo, el ancho de banda se comparte entre un conjunto de N suscriptores, donde el tamaño del conjunto se controla mediante el proveedor de cable. Desde el punto de vista de un suscriptor, compartir el ancho de banda con otros suscriptores puede ser una desventaja debido a que la velocidad de datos efectiva disponible para cada suscriptor individual varía con respecto al tiempo. En el peor de los casos, si N suscriptores comparten una sola frecuencia, la capacidad disponible para cada uno de los suscriptores será de $1/N$.

12.11 Instalación de un módem de cable

Puesto que los sistemas de cable usan la FDM, la instalación de un módem de cable es sencilla. A diferencia de las tecnologías xDSL que requieren el uso de filtros divisorios, los módems de cable se conectan directamente al cableado correspondiente. El hardware de FDM que está en las cajas y en los

[†] La televisión por cable, que se conoce formalmente como *televisión de antena comunitaria (CATV)*, usa la FDM para transmitir señales de televisión a los suscriptores a través del cable coaxial. La CATV no está disponible en todos los países.

módems de cable existentes garantiza que los canales de datos y de entretenimiento no interfieran entre sí. En conclusión:

Puesto que los sistemas de cable usan la multiplexación por división de frecuencias, es posible conectar un módem de cable directamente al cable existente sin necesidad de un filtro divisor.

12.12 Fibra híbrida coaxial

Uno de los aspectos que encarecen las tecnologías de acceso se debe al costo de tender cables de cobre u ópticos entre cada cliente y las instalaciones de un proveedor. Por lo tanto, los proveedores buscan tecnologías que les permitan ofrecer servicios de mayor velocidad, al tiempo que minimizan el número de alambres físicos que deben cambiarse. Una tecnología que ofrece velocidades de datos mayores sin reemplazar todos los alambres se conoce por el nombre general de *fibra híbrida coaxial (HFC)*. Como el nombre lo indica, un sistema de fibra híbrida coaxial usa una combinación de fibras ópticas y cables coaxiales, donde la fibra se utiliza para las instalaciones centrales y el cable coaxial para las conexiones individuales de los suscriptores. En esencia, un sistema HFC es jerárquico. Usa fibra óptica para las partes de la red que requieren el mayor ancho de banda y cable coaxial para las piezas que pueden tolerar velocidades de datos menores. Para implementar dicho sistema, un proveedor coloca dispositivos en cada vecindario que puedan realizar la conversión entre el cable óptico y el cable coaxial. Cada dispositivo se conecta de regreso con el proveedor a través de una fibra óptica y se conecta con las casas del vecindario por medio de un cable coaxial. Si ya hay cables coaxiales instalados para la TV por cable, el costo se minimiza. La figura 12.7 ilustra la arquitectura.

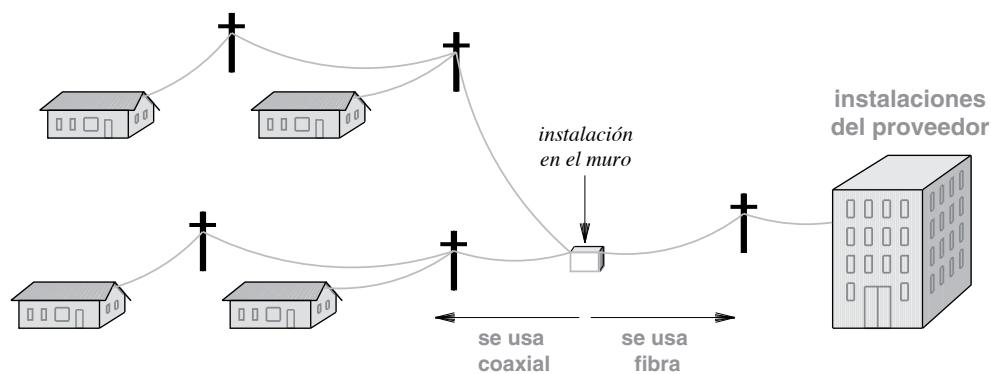


Figura 12.7 Ilustración de un sistema de acceso de fibra híbrida coaxial.

La industria del cable usa el término *troncal* para referirse a las conexiones de alta capacidad que hay entre la oficina de cable y cada vecindario, y el término *circuito alimentador* para referirse a la conexión hacia cada suscriptor individual. Las conexiones troncales pueden ser de hasta 15 millas de longitud; por lo general los circuitos alimentadores son menores a una milla.

12.13 Tecnologías de acceso que emplean la fibra óptica

Las compañías de cable han propuesto una variedad de tecnologías que emplean ya sea la fibra óptica en un sistema híbrido o despliegan la fibra óptica en toda la conexión que va hacia cada suscriptor. La figura 12.8 sintetiza los nombres de las tecnologías clave.

Nombre	Expansión
FTTC	Fibra hasta el muro
FTTB	Fibra hasta el edificio
FTTH	Fibra hasta el hogar
FTTP	Fibra hasta las instalaciones

Figura 12.8 Nombres de las tecnologías adicionales de acceso que usan fibra óptica.

Fibra hasta el muro (FTTC). Como el nombre lo implica, FTTC es similar a HFC debido a que usa fibra óptica para troncales de alta capacidad. La idea es tender fibra óptica cerca del suscriptor final y luego usar cable de cobre para los circuitos alimentadores. FTTC difiere de HFC debido a que usa dos medios en cada circuito alimentador para que el sistema de cable pueda ofrecer un servicio adicional, como la voz. La tecnología se está implementando en ciertas áreas, especialmente en Estados Unidos y Canadá.

Fibra hasta el edificio (FTTB). Hay una pregunta fundamental respecto del ancho de banda que necesitarán las empresas, y si las tecnologías de acceso que utilizan cobre (incluso el cable coaxial) bastarán. FTTB es una tecnología que usará fibra óptica para permitir altas velocidades de datos de flujo ascendente.

Fibra hasta el hogar (FTTH). La FTTH es la contraparte de FTTB y es una tecnología de acceso que usa fibra óptica para ofrecer mayores velocidades de datos de flujo descendente para los suscriptores residenciales. Aunque la FTTH también ofrece mayores velocidades de datos de flujo ascendente, el énfasis es en muchos canales de entretenimiento y de video.

Fibra hasta las instalaciones (FTTP). FTTP es un término genérico que abarca tanto a FTTB como a FTTH.

12.14 Terminología de módems de extremo cercano y extremo lejano

Cualquier tecnología de acceso (ya sea que use alambres de cobre o fibras ópticas) requiere de un par de módems: uno en el sitio del suscriptor y otro en el sitio del proveedor. La industria usa el término *módem de extremo cercano* o *head-end* para referirse a un módem utilizado en el sitio del proveedor, y *módem de extremo lejano* o *tail-end* para referirse a un módem utilizado en la ubicación del suscriptor.

Los módems de extremo cercano no son dispositivos individuales, sino que se fabrica un gran conjunto de éstos dentro de una unidad, de modo que se puedan configurar, monitorear y controlar juntos. A este conjunto de módems de extremo cercano utilizados por un proveedor de cable, se le conoce como *sistema*

de terminación de módem de cable (CMTS). El conjunto de estándares de la industria conocido como *especificaciones de la interfaz del sistema de servicio de datos sobre cable (DOCSIS)* especifica tanto el formato de los datos que pueden enviarse como los mensajes utilizados para solicitar servicios (por ejemplo, las películas sobre demanda).

12.15 Tecnologías de acceso inalámbricas

Aunque las tecnologías como ADSL o HFC pueden ofrecer servicios digitales a la mayoría de los suscriptores, no resuelven todas las situaciones. Los principales problemas surgen en las áreas rurales. Por ejemplo, imagine una granja o aldea remota a muchos kilómetros de distancia de la ciudad más cercana. El cableado de par trenzado que se utiliza para ofrecer servicio telefónico a dichas ubicaciones excede la distancia máxima permitida para tecnologías como la ADSL. Además, es poco probable que las áreas rurales tengan servicio de televisión por cable.

Incluso en áreas suburbanas, las tecnologías como ADSL pueden tener restricciones técnicas en cuanto al tipo de línea que pueden usar. Por ejemplo, tal vez sea imposible usar frecuencias altas en líneas telefónicas que contengan *bobinas de carga, puentes de conexión o repetidores*. Por consiguiente, incluso en áreas en las que la tecnología de bucle local funciona para la mayoría de los suscriptores, tal vez no funcione en todas las líneas.

Para manejar casos especiales, se exploró una variedad de tecnologías de acceso inalámbrico. La figura 12.9 lista algunos ejemplos y el capítulo 16 habla sobre varias de estas tecnologías.

Tecnología	Descripción
Servicios 3G y 4G	Servicios de datos de telefonía celular de tercera y cuarta generación (por ejemplo, EVDO y LTE)
WiMAX	Tecnología de acceso inalámbrica de hasta 155 Mbps que usa radiofrecuencias
Satélite	Varios distribuidores comerciales ofrecen servicios de acceso a Internet a través de satélite

Figura 12.9 Ejemplos de tecnologías de acceso inalámbrico.

12.16 Conexiones de alta capacidad en el núcleo de Internet

Los profesionales de redes dicen que las tecnologías de acceso se hacen cargo del *problema del último tramo*, refiriéndose a la conexión final hacia un típico suscriptor residencial o de negocio pequeño. Las tecnologías de acceso ofrecen capacidad suficiente para un suscriptor residencial o una pequeña oficina en casa. Pero las conexiones a empresas grandes o entre los propios proveedores requieren de un ancho de banda mucho mayor. Para diferenciar entre las conexiones de alta velocidad y las que se encuentran en el extremo de Internet, los profesionales usan el término *tecnologías centrales* para las tecnologías de alta velocidad.

Para comprender las velocidades de datos necesarias para el *núcleo de internet*, considere los datos que transfiere un proveedor con 5000 clientes. Suponga que el proveedor usa una tecnología de acceso que puede proveer hasta 2 Mbps por cliente, y considere lo que ocurre si todos los suscriptores intentan descargar datos al mismo tiempo. La figura 12.10 muestra el tráfico acumulado desde la Internet al proveedor.

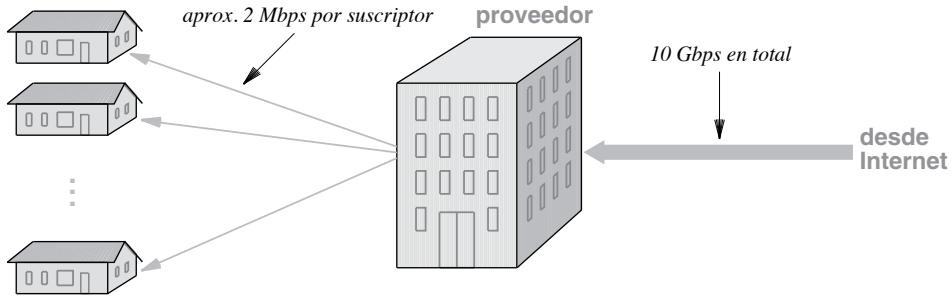


Figura 12.10 Tráfico acumulado desde Internet a un proveedor, suponiendo que el proveedor tiene 5000 clientes y cada uno de ellos descarga 2 Mbps.

Surge la pregunta: ¿qué tecnología puede ofrecer un proveedor para mover datos por largas distancias a la velocidad de 10 Gbps? La respuesta reside en un *circuito digital de punto a punto* que se renta a una compañía telefónica. Aunque los circuitos digitales de alta capacidad se diseñaron en un principio para usarse internamente en el sistema telefónico, éstos están disponibles por una cuota mensual y pueden usarse para transferir datos. Como las compañías telefónicas tienen el permiso para instalar cableado que atravesie las calles de una ciudad o población, estos circuitos pueden extenderse entre dos edificios, a través de toda la ciudad o incluso de una ciudad a otra. La cuota a cobrar depende de la velocidad de datos del circuito y de la distancia cubierta. Para resumir:

Los circuitos digitales que se rentan a portadoras comunes forman los bloques fundamentales para las comunicaciones de datos de larga distancia. El costo depende de la capacidad del circuito y la distancia.

12.17 Terminación de circuitos, DSU/CSU y NIU

Para usar un circuito digital rentado, hay que aceptar las reglas del sistema telefónico, lo que incluye seguir los estándares para la transmisión de llamadas de voz digitalizadas. Puesto que las computadoras también son digitales, tal vez parezca que seguir estos estándares no tiene sentido. Sin embargo, y debido a que la industria de la computación y la industria telefónica se desarrollaron de manera independiente, los estándares para los circuitos digitales del sistema telefónico difieren de los que se utilizan en la industria de las computadoras. Por lo tanto, se requiere una pieza especial de hardware para interconectar una computadora con un circuito digital proporcionado por una compañía telefónica. Conocido

como *unidad de servicio de datos/unidad de servicio de canal (DSU/CSU)*, el dispositivo contiene dos partes funcionales que por lo general se combinan en un solo chasis. La parte correspondiente a la CSU del dispositivo se encarga de la terminación de la línea y de los diagnósticos. Por ejemplo, una CSU contiene circuitos de diagnóstico que pueden probar si la línea se desconectó. Una CSU también contiene una herramienta de prueba de *bucle de retorno* que le permite transmitir de regreso al emisor una copia de todos los datos que llegan en el circuito, sin necesidad de un procesamiento adicional.

Una CSU brinda un servicio que los ingenieros de computadoras encuentran sorprendente: prohíbe el exceso de bits *1* consecutivos. La necesidad de evitar los bits *1* surge debido a las señales eléctricas utilizadas. Puesto que la compañía telefónica diseñó en un principio sus circuitos digitales para que trabajaran sobre cables de cobre, a los ingenieros les preocupaba que demasiados bits *1* contiguos provocaran un exceso de corriente en el cable. Para evitar estos problemas, una CSU puede usar una codificación que garantice un equilibrio (por ejemplo, una codificación diferencial), o una técnica conocida como *relleno de bits*.

La parte correspondiente a la DSU de un dispositivo DSU/CSU se encarga de los datos. Traduce los datos entre el formato digital utilizado en el circuito de la portadora y el formato digital requerido por el equipo de cómputo del cliente. El estándar de interfaz que se utiliza del lado de la computadora depende de la velocidad con la que opere el circuito. Si la velocidad de datos es menor a 56 Kbps, la computadora puede usar RS-232. Para velocidades superiores a 56 Kbps, la computadora debe usar hardware de interfaz que soporte velocidades mayores (por ejemplo, hardware que use los estándares RS-449 o V-35).

La compañía telefónica proporciona una pieza de equipo adicional, conocida como *unidad de interfaz de red (NIU)*,[†] que forma un límite entre el equipo perteneciente a la compañía telefónica y el equipo que proporciona el suscriptor. La compañía telefónica se refiere al límite como *punto de demarcación*. En conclusión:

Un circuito digital necesita un dispositivo conocido como DSU/CSU en cada extremo. El DSU/CSU hace la traducción entre la representación digital utilizada por las compañías telefónicas y la representación digital utilizada por la industria computacional.

12.18 Estándares de telefonía para circuitos digitales

Un circuito digital que se renta a una compañía telefónica cumple los mismos estándares de transmisión digital utilizados por esta compañía para transportar llamadas telefónicas digitales. En Estados Unidos, los estándares para los circuitos telefónicos digitales recibieron nombres que consisten en la letra *T* seguida de un número. Los ingenieros se refieren a ellos colectivamente como *estándares de la serie T*. Uno de los más populares se conoce como T1; muchas empresas pequeñas usan un circuito T1 para transportar datos.

Por desgracia, los estándares T no son universales. Japón adoptó una versión modificada de los estándares de la serie T, y Europa eligió un esquema ligeramente diferente. Los estándares europeos pueden diferenciarse debido a que usan la letra *E*. La figura 12.11 enumera las velocidades de datos de varios estándares de circuitos digitales.

[†]Aunque algunas veces se usa el término *smartjack* como sinónimo de NIU, éste se refiere a un tipo específico de NIU fabricado por Westell Corporation.

Nombre	Velocidad de bits	Circuitos de voz	Ubicación
velocidad básica	0.064 Mbps	1	
T1	1.544 Mbps	24	Norteamérica
T2	6.312 Mbps	96	Norteamérica
T3	44.736 Mbps	672	Norteamérica
E1	2.048 Mbps	30	Europa
E2	8.448 Mbps	120	Europa
E3	34.368 Mbps	480	Europa

Figura 12.11 Ejemplos de circuitos digitales y su capacidad.

12.19 Terminología DS y velocidades de datos

En el capítulo 11 vimos que las compañías telefónicas usan una jerarquía de multiplexación que combina varias llamadas de voz en un solo circuito digital. Por lo tanto, las velocidades de datos de los estándares T se eligieron de modo que cada uno pueda manejar varias llamadas de voz. Lo importante a observar aquí es que la capacidad de los circuitos no se incrementa en forma lineal con sus números. Por ejemplo, el estándar T3 define a un circuito de más de tres veces la capacidad de T1. Por último, cabe mencionar que las compañías telefónicas rentan circuitos con menor capacidad que los que se indican en la figura y se conocen como circuitos de línea *T1 fraccionada*.

Para ser técnicamente precisos, debemos diferenciar entre los estándares T que definen el sistema de portadora y los estándares que especifican cómo multiplexar varias llamadas telefónicas en una sola conexión. Estos últimos se conocen como *estándares de nivel de señal digital* o *estándares DS*. Sus nombres se definen con las letras *DS* seguidas de un número, de manera similar a los estándares T. Por ejemplo, DS1 denota un servicio que puede multiplexar 24 llamadas telefónicas en un solo circuito y T1 denota un estándar específico que hace eso. Como DS1 define la velocidad efectiva de datos, es técnicamente más preciso decir “un circuito que opera a una velocidad de DS1” en vez de referirse a “velocidad de T1”. En la práctica, pocos ingenieros se molestan en diferenciar entre T1 y DS1. Por lo tanto, es común escuchar que alguien se refiera a la “velocidad de T1”.

12.20 Circuitos de mayor capacidad (estándares STS)

Las compañías telefónicas usan el término *troncal* para denotar un circuito de alta capacidad y crearon una serie de estándares para los circuitos troncales digitales. Conocidos como los estándares de *señal de transporte síncrono* (STS), éstos especifican los detalles de las conexiones de alta velocidad. La figura 12.12 sintetiza las velocidades de datos asociadas con diversos estándares STS. Todas las velocidades de datos

de la tabla se dan en Mbps, lo que facilita su comparación. Hay que señalar que las velocidades de datos para STS-24 y superiores son mayores de 1 Gbps.

Nombre de cobre	Nombre óptico	Velocidad de bits	Circuitos de voz
STS-1	OC-1	51.840 Mbps	810
STS-3	OC-3	155.520 Mbps	2430
STS-12	OC-12	622.080 Mbps	9720
STS-24	OC-24	1,244.160 Mbps	19440
STS-48	OC-48	2,488.320 Mbps	38880
STS-192	OC-192	9,953.280 Mbps	155520

Figura 12.12 Velocidades de datos de los circuitos digitales de acuerdo con la jerarquía de estándares STS.

12.21 Estándares de portadora óptica

Además de los estándares STS, la compañía telefónica define un conjunto equivalente de estándares de *portadora óptica* (*OC*). La figura 12.12 anterior proporciona los nombres de los estándares ópticos, así como los estándares de cobre. Para ser precisos, hay que observar una distinción entre la terminología STS y OC: los estándares STS se refieren a las señales eléctricas que se utilizan en la interfaz del circuito digital (es decir, a través del cobre), mientras que los estándares OC se refieren a las señales ópticas que se propagan a través de la fibra óptica. Al igual que con el resto de la terminología de red, pocos profesionales hacen esta distinción. Por lo tanto, a menudo escuchamos que los profesionales de redes usan el término *OC-3* para referirse a un circuito digital que opera a 155 Mbps, sin importar que el servicio use cobre o fibra óptica.

12.22 El sufijo C

La terminología de la señal de transporte síncrono y de la portadora óptica antes descritas tiene una característica adicional que no se muestra en la figura 12.12: un sufijo opcional de la letra *C*, que significa *concatenación*. La presencia del sufijo denota un circuito sin multiplexación inversa. Es decir, un circuito OC-3 puede consistir en tres circuitos OC-1 que operan a 51.849 Mbps cada uno, o puede consistir en un solo circuito OC-3C (STS-3C) que opera a 155.520 Mbps.

¿Es un solo circuito que opera a toda velocidad mejor que varios circuitos operando a velocidades menores? La respuesta depende de cómo se utilice el circuito. En general, el hecho de que un solo circuito opere a toda su capacidad ofrece más flexibilidad y elimina la necesidad de equipo de multiplexación inversa. Más concretamente, las redes de datos no son como las redes de voz. En un sistema de voz,

los circuitos de alta capacidad se usan como forma de agregar flujos de voz más pequeños. Sin embargo, en una red de datos hay un solo flujo de tráfico de datos. Por consiguiente, si se les da a elegir, la mayoría de los diseñadores de redes prefieren un circuito OC-3C en vez de un circuito OC-3.

12.23 Red óptica síncrona (SONET)

Además de los estándares STS y OC antes descritos, las compañías telefónicas definieron un amplio conjunto de estándares para la transmisión digital. En Norteamérica los estándares se conocen por el término *red óptica síncrona (SONET)*, mientras que en Europa se conocen como *jerarquía digital síncrona (SDH)*. SONET especifica detalles como la forma en que se entraman los datos, cómo se multiplexan los circuitos de menor capacidad en un circuito de alta capacidad, y cómo se envía la información del reloj sincronizado junto con los datos. Como las portadoras usan SONET en muchos lugares, cuando alguien renta un circuito STS-1 es probable que la portadora requiera que usen la codificación SONET en el circuito. Por ejemplo, la figura 12.13 muestra el formato de trama de SONET que se utiliza en un circuito STS-1.

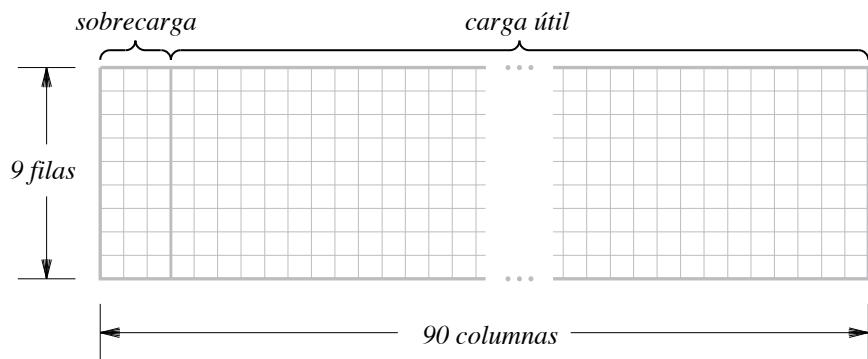


Figura 12.13 Ilustración de una trama de SONET cuando se usa a través de un circuito STS-1.

Cada trama tiene 810 octetos de largo. De acuerdo con la terminología de SONET, los octetos de la trama se dividen en 9 “filas” con 90 “columnas” en cada fila. Lo interesante es que el tamaño de una trama de SONET depende de la velocidad de bits del circuito utilizado. Cuando se utiliza en un circuito STS-3, cada trama SONET contiene 2430 octetos. ¿Cómo surgen estos números? Para comprender esta diferencia, recuerde que la telefonía digital toma 8000 muestras PCM por segundo, lo que significa que se toma una muestra cada 125μ segundos. SONET usa el tiempo para definir el tamaño de la trama. A la velocidad de transmisión de 51.840 Mbps de STS-1 se transfieren exactamente 6480 bits en 125μ segundos, lo que significa que una trama consiste en 810 octetos de 8 bits. De manera similar, a la velocidad de STS-3 es posible transmitir 2430 octetos en 125μ segundos. La principal ventaja de hacer que el tamaño de la trama dependa de la velocidad de bits del circuito es que la multiplexación síncrona se convierte en algo sencillo; mantener la sincronización al tiempo que se combinan tres flujos SONET STS-1 en un flujo SONET STS-3 es un proceso simple y directo.

Aunque la mayoría de las redes de datos usan SONET como un esquema de codificación en un solo circuito de punto a punto, el estándar ofrece más posibilidades. En especial, es posible crear una red tipo anillo de alta capacidad en sentido contrario, usando tecnología SONET que maneje fallas en un solo punto. Cada estación del anillo usa un dispositivo conocido como *multiplexor de adición/supresión*. Además de pasar los datos que recibe a lo largo del anillo, el multiplexor de adición/supresión puede configurarse para aceptar datos adicionales de un circuito local y agregarlos a las tramas que pasan a través del anillo, o puede extraer datos y entregarlos a una computadora local. Si el anillo se rompe, el hardware detecta la pérdida de información de entrampado y usa el anillo en sentido contrario para reconectarse. Para resumir:

Aunque el estándar SONET define una tecnología que puede usarse para construir una red tipo anillo de alta capacidad con varios circuitos de datos multiplexados a través de las fibras que constituyen el anillo, la mayoría de las redes de datos usan SONET sólo para definir el entrampado y la codificación en un circuito rentado.

12.24 Resumen

Las tecnologías de acceso proporcionan conexiones individuales de Internet a residencias o empresas pequeñas. Existe una variedad de tecnologías de acceso, que incluyen las conexiones de marcación telefónica, así como las conexiones inalámbricas (que utilizan radiofrecuencia o satélite) y las alámbricas. Dos de las tecnologías de acceso actuales son la línea de suscriptor digital (DSL) y los módems de cable. DSL usa técnicas de FDM para permitir que tanto la comunicación digital como una llamada de voz analógica tradicional procedan al mismo tiempo en el bucle local entre la oficina central de una compañía telefónica y un suscriptor. El servicio de módem por cable usa la FDM para multiplexar la comunicación digital a través del mismo sistema de cable coaxial que se utiliza para transportar canales de entretenimiento. Con la tecnología de módem de cable, los módems de cada vecindario emplean la multiplexación estadística para compartir un solo canal de comunicaciones de datos.

Las tecnologías como la fibra híbrida coaxial (HFC) y la fibra hasta el muro (FTTC) usan fibras ópticas para distribuir los datos en cada vecindario y usan cable coaxial para llegar a un suscriptor individual. Se han propuesto tecnologías a futuro que usarán fibra óptica para ofrecer mayores velocidades de datos en cada residencia.

Aunque son suficientes para las residencias particulares y empresas pequeñas, las tecnologías de acceso no ofrecen suficiente capacidad para usarse en el núcleo de Internet. Para obtener velocidades de datos más altas a través de largas distancias, los proveedores de servicios y las empresas grandes rentan circuitos punto a punto a las portadoras comunes. Los circuitos digitales usan los estándares de multiplexación por división de tiempo (estándares T en Norteamérica o estándares E en Europa). Los circuitos de alta velocidad se definen mediante los estándares de la *señal de transporte síncrono* (Norteamérica) o la *jerarquía digital síncrona* (Europa). Existe un conjunto paralelo de estándares de portadora óptica para usarse con la fibra óptica; muchos profesionales usan los nombres del estándar OC, sin importar que el circuito use fibra o cobre.

Un estándar de las compañías telefónicas conocido como SONET define el entramado para usarse en un circuito digital. El tamaño de una trama de SONET depende de la velocidad de bits del circuito; una trama siempre tarda 125 μ segundos en enviarse. Además de su uso en circuitos punto a punto, SONET puede configurarse en un anillo, lo cual permite al hardware determinar si el anillo está roto y reconfigurarse de manera automática alrededor de la falla.

EJERCICIOS

- 12.1** ¿Qué es una tecnología de acceso?
- 12.2** ¿Por qué los proveedores de servicio distinguen entre comunicación de flujo ascendente y de flujo descendente?
- 12.3** Mencione ejemplos de tecnologías de acceso de banda estrecha y de banda ancha.
- 12.4** Alguna vez las compañías telefónicas promovieron a ISDN como una tecnología de acceso de alta velocidad. ¿Por qué disminuyó el uso de ISDN?
- 12.5** Si un cliente desea transmitir más datos de los que recibe, ¿qué formas de DSL serían apropiadas? ¿Y si desea recibir más datos de los que transmite?
- 12.6** ¿Qué tipo de multiplexación usa ADSL?
- 12.7** Dos vecinos que viven en la misma calle utilizan el servicio ADSL, pero las mediciones muestran que un suscriptor puede descargar a una velocidad aproximada de 1.5 Mbps y el otro a 2.0 Mbps. Explique esto.
- 12.8** ¿Por qué se usa un filtro divisor para DSL?
- 12.9** Si tuviera que elegir entre DSL y el módem de cable, ¿cuál opción proporcionaría la mayor velocidad potencial de datos?
- 12.10** ¿Por qué un proveedor de servicios elegiría la fibra híbrida coaxial en vez de la fibra hasta las instalaciones?
- 12.11** ¿En dónde se encuentra un módem de extremo cercano? ¿Y un módem de extremo lejano?
- 12.12** ¿Cuál es la ventaja de la tecnología de acceso WiMAX en comparación con el satélite? ¿Cuál es la ventaja del satélite?
- 12.13** Si usted renta un circuito T1, ¿qué equipo se instalará entre el circuito y su computadora?
- 12.14** Investigue en Web el tamaño aproximado de una película en DVD. ¿Cuánto tiempo tardaría en descargar la película a través de una línea T1? ¿Y a través de una línea T3? (Ignore la sobrecarga).
- 12.15** Si alguien le muestra un cable de cobre y afirma que es un “circuito OC-12”, ¿qué error cometió? ¿Cuál es el nombre correcto que debería utilizar?
- 12.16** ¿Por qué los diseñadores de la jerarquía digital síncrona eligieron valores inusuales para las velocidades de datos en vez de potencias exactas de diez?
- 12.17** Explique cómo se calcula el tamaño de una trama de SONET.

PARTE III

Commutación de paquetes y tecnologías de redes

**Una descripción general de la commutación
de paquetes y las tecnologías de redes que
usan medios alámbricos e inalámbricos**

Capítulos

- 13 Redes de área local: paquetes, tramas y topologías**
- 14 La subcapa MAC del IEEE**
- 15 Tecnología alámbrica de LAN (Ethernet y 802.3)**
- 16 Tecnologías de redes inalámbricas**
- 17 Repetidores, puentes y conmutadores**
- 18 Tecnologías WAN y enrutamiento dinámico**
- 19 Tecnologías de redes pasadas y presentes**

Contenido del capítulo

- 13.1 Introducción, 219
- 13.2 Conmutación de circuitos y comunicación analógica, 220
- 13.3 Conmutación de paquetes, 221
- 13.4 Redes de paquetes de área local y amplia, 222
- 13.5 Estándares para formato e identificación de paquetes, 223
- 13.6 El modelo IEEE 802 y los estándares, 224
- 13.7 Redes punto a punto y multiacceso, 225
- 13.8 Topologías de LAN, 227
- 13.9 Identificación de paquetes, demultiplexación, direcciones MAC, 229
- 13.10 Direcciones de unidifusión, difusión y multidifusión, 230
- 13.11 Difusión, multidifusión y entrega multipunto eficiente, 231
- 13.12 Tramas y entramado, 232
- 13.13 Relleno de bytes y bits, 233
- 13.14 Resumen, 234

13

Redes de área local: paquetes, tramas y topologías

13.1 Introducción

La primera parte del libro cubre las aplicaciones de Internet y la programación de redes. La segunda parte explora los temas relacionados con las comunicaciones de datos. Cada capítulo cubre un concepto fundamental, como la multiplexación, que forma la base de todas las redes de computadoras.

Este capítulo comienza la parte del libro que examina la conmutación de paquetes y las tecnologías de redes de computadoras. Después de una breve descripción general, el capítulo explica el modelo de estándares de IEEE y se concentra en los conceptos de direccionamiento de hardware e identificación de tramas.

Los capítulos posteriores de esta parte amplían la explicación al considerar los paquetes tanto en redes de área local como en redes de área amplia. Además, los capítulos posteriores cubren una variedad de tecnologías de redes alámbricas e inalámbricas que aceptan y entregan paquetes.

13.2 Comutación de circuitos y comunicación analógica

El término *comutación de circuitos* se refiere a un mecanismo de comunicación que establece una ruta entre un emisor y un receptor con un aislamiento garantizado de las rutas utilizadas por otros pares de emisores y receptores. Por lo general, la comutación de circuitos se relaciona con la tecnología de telefonía analógica, ya que un sistema telefónico proporciona una conexión dedicada entre dos teléfonos. De hecho, el término se originó con las primeras redes de marcación telefónica que usaban dispositivos de comutación electromecánicos para formar un circuito físico. La figura 13.1 ilustra cómo se lleva a cabo la comunicación a través de una red de comutación de circuitos.

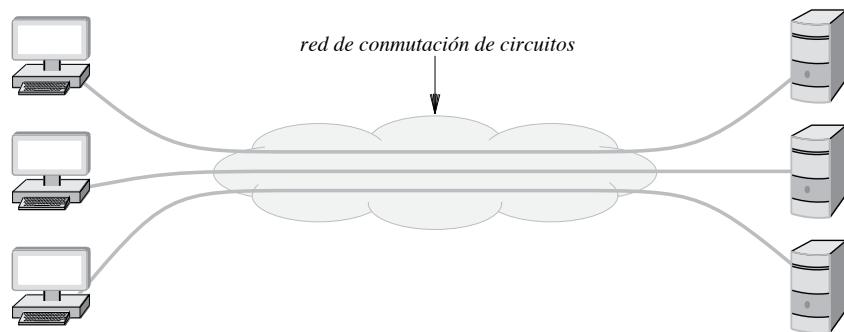


Figura 13.1 Una red de comutación de circuitos que proporciona una conexión directa entre cada par de entidades de comunicación.

Las redes de comutación de circuitos usan dispositivos electrónicos para establecer los circuitos. Además, en vez de hacer que cada circuito corresponda a una ruta física, se multiplexan varios circuitos a través de medios compartidos y el resultado se conoce como *circuito virtual*. Así, la distinción entre la comutación de circuitos y otras formas de redes ya no se define por la existencia de rutas físicas independientes. En su lugar, tres propiedades generales definen un paradigma de comutación de circuitos:

- La comunicación punto a punto
- Cada uno de los pasos para la creación, uso y terminación de circuitos
- El rendimiento equivalente a una ruta física aislada

La primera propiedad indica que se forma un circuito exactamente entre dos puntos finales, y la segunda propiedad distingue los circuitos *comutados* (es decir, que se crean cada vez que es necesario) de los circuitos *permanentes* (los que siempre están listos para usarse). Los circuitos comutados usan un proceso de tres pasos similar al de realizar una llamada telefónica. En el primer paso, cuando un humano o un programa de aplicación intentan comunicarse, se crea un circuito. En el segundo paso, las dos partes usan el circuito para comunicarse, y en el tercero las dos partes terminan su uso. De esta forma, un circuito comutado es temporal en cuanto a que sólo permanece vigente mientras se necesita; una vez que termina la comunicación, el circuito se suprime.

La tercera propiedad ofrece una distinción crucial entre las redes de comutación de circuitos y otros tipos de redes. La comutación de circuitos significa que la comunicación entre dos partes no se ve afectada de ninguna forma por la comunicación entre otras partes, aun cuando toda la comunicación se multiplexe a través de un medio común. En especial, la comutación de circuitos debe dar la ilusión de una ruta independiente para cada par de entidades en comunicación. Por consiguiente, para multiplexar circuitos a través de un medio compartido, deben usarse técnicas como la multiplexación por división de frecuencias o la multiplexación síncrona por división de tiempo.

En conclusión:

La comutación de circuitos da la ilusión de una ruta física aislada entre un par de entidades en comunicación; se crea una ruta cuando es necesario y se descontinúa después de usarla.

13.3 Comutación de paquetes

La principal alternativa a la comutación de circuitos es la *comutación de paquetes*, que forma la base para la Internet. Un sistema de comutación de paquetes usa la multiplexación estadística, donde la comunicación de varias fuentes compite por el uso de medios compartidos. La figura 13.2 ilustra el concepto.

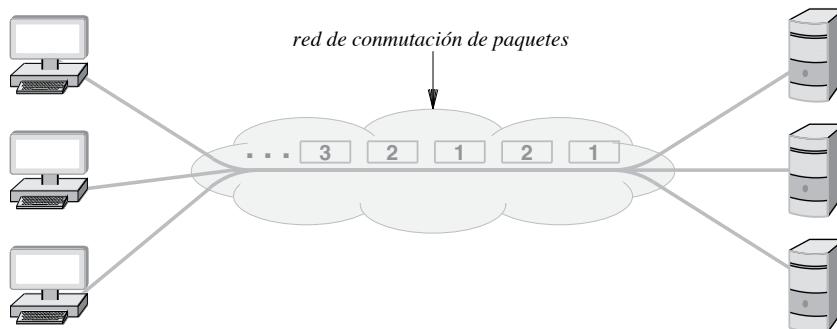


Figura 13.2 Una red de comutación de paquetes que envía un paquete a la vez, a través de un medio compartido.

La principal diferencia entre la comutación de paquetes y otras formas de multiplexación estadística surge debido a que un sistema de comutación de paquetes requiere que un emisor divida cada mensaje en pequeños bloques de datos que se conocen como *paquetes*. El tamaño de un paquete varía; cada tecnología de comutación de paquetes define un tamaño máximo del paquete.[†]

[†] Los paquetes no son grandes: un tamaño máximo común de un paquete es de 1500 bytes.

Las propiedades generales que definen un paradigma de conmutación de paquetes son:

- Comunicación asíncrona arbitraria
- No se requiere configuración antes de comenzar la comunicación
- El rendimiento varía debido a la multiplexación estadística entre paquetes

La primera propiedad significa que la conmutación de paquetes puede permitir a un emisor comunicarse con uno o varios receptores, y un receptor dado puede recibir mensajes de uno o varios emisores. Además, la comunicación puede ocurrir en cualquier momento y un emisor puede tener retrasos arbitrariamente extensos entre eventos de comunicación sucesivos. La segunda propiedad significa que a diferencia de un sistema de conmutación de circuitos, un sistema de conmutación de paquetes está listo para entregar un paquete en cualquier destino y cualquier momento. Por lo tanto, un emisor no necesita realizar una inicialización antes de comunicarse y no necesita notificar al sistema utilizado cuando termina la comunicación.

La tercera propiedad significa que la multiplexación ocurre entre paquetes y no entre bits o bytes. Es decir, una vez que un emisor obtiene acceso al canal involucrado, el emisor transmite un paquete completo y luego permite a otros emisores transmitir un paquete. Cuando no hay otros emisores listos para transmitir un paquete, un solo emisor puede transmitir en forma repetida. Pero si varios emisores comparten una red de conmutación de paquetes, la red está diseñada para dar a cada emisor una oportunidad equitativa. Es decir, si N emisores tienen cada uno un conjunto extenso de paquetes listos para ser enviados, un emisor dado podrá transmitir aproximadamente $1/N$ de todos los paquetes.

Para resumir:

La conmutación de paquetes forma la base de Internet y es una forma de multiplexación estadística que permite la comunicación de varios a varios. Un emisor debe dividir un mensaje en un conjunto de paquetes; después de que un emisor transmite un paquete, la red está diseñada para permitir que otros emisores transmitan antes que el emisor pueda transmitir su siguiente paquete.

Una de las ventajas principales de la conmutación de paquetes es el costo más bajo que se obtiene al compartir. Para brindar una comunicación entre N computadoras, una red de conmutación de circuitos debe tener una conexión para cada computadora, más $N/2$ rutas independientes cuando menos. Con la conmutación de paquetes, una red debe tener una conexión para cada computadora pero sólo requiere una ruta, la cual es compartida.

13.4 Redes de paquetes de área local y amplia

Las tecnologías de conmutación de paquetes se clasifican comúnmente de acuerdo con la distancia que abarcan. Las redes menos costosas usan tecnologías que abarcan una distancia corta (por ejemplo, dentro de un edificio) y las más costosas abarcan largas distancias (por ejemplo, a lo largo de varias ciudades). La figura 13.3 sintetiza las tres categorías principales.

Nombre	Expansión	Descripción
LAN	Red de área local	La menos costosa; abarca una sola habitación o edificio
MAN	Red de área metropolitana	De costo mediano; abarca una ciudad o área metropolitana importante
WAN	Red de área amplia	La más costosa; abarca sitios en varias ciudades

Figura 13.3 Las tres principales categorías de redes de conmutación de paquetes.

En la práctica se han creado pocas tecnologías de MAN, por lo que las redes MAN no han sido comercialmente exitosas. En consecuencia, los profesionales de redes tienden a agrupar las tecnologías de MAN en la categoría de WAN y usar sólo los términos LAN y WAN.

La terminología se ha vuelto tan extensa que a menudo los grupos proponen variantes que comienzan con “redes de área”. Por ejemplo, el capítulo 16 describe las tecnologías de *redes de área personal (PAN)* como Bluetooth, que se limitan a unos cuantos metros. Además, algunas veces los distribuidores de chips usan el término *red de área de chip (CAN)* para referirse a los mecanismos de conmutación de paquetes que conectan varios núcleos con un solo chip VLSI.

13.5 Estándares para formato e identificación de paquetes

Puesto que los sistemas de conmutación de paquetes se basan en la compartición, cada paquete que se envía a través de una red debe contener la identificación del receptor destinado. Además, para asegurar que no haya ambigüedades, todos los emisores deben estar de acuerdo en los detalles exactos sobre cómo identificar un receptor y en dónde colocar la identificación de un paquete. Las organizaciones de estándares crean documentos de protocolos que especifican todos los detalles. El conjunto de estándares más utilizado para las LAN lo creó el *Instituto de ingenieros eléctricos y electrónicos (IEEE)*.

En 1980, el IEEE organizó el *Comité de estándares LAN/MAN del proyecto 802* para producir los estándares del trabajo en red. Para entender los estándares del IEEE, es importante saber que la organización está compuesta por ingenieros que se concentran en las dos capas inferiores de la pila de protocolos. De hecho, si leemos los documentos del IEEE puede parecer que todos los demás aspectos de las redes no importan. Sin embargo, existen otras organizaciones de estándares y cada una se concentra en capas específicas de la pila. El IETF se enfoca en los protocolos de transporte y de Internet, y el consorcio World Wide Web se concentra en los estándares de la capa de aplicación. Cada grupo piensa que sus capas son las más importantes. La figura 13.4 ofrece una ilustración humorística de una pila de protocolos, según el punto de vista de cada organización de estándares.

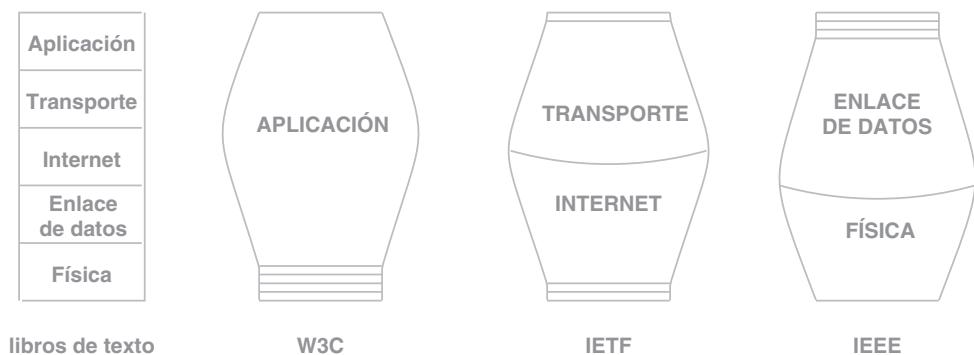


Figura 13.4 Una ilustración humorística de una pila de protocolos, según el punto de vista de varias organizaciones de estándares.

Debemos estar conscientes de que los estándares de una organización específica sólo pertenecen a ciertas capas, y de que la cantidad de publicaciones de estándares no es proporcional a la importancia de una capa específica. Para resumir:

Cada organización de estándares se concentra en algunas capas específicas de la pila de protocolos. Los estándares del IEEE se concentran en la especificación de las dos capas inferiores de la pila y las tecnologías de LAN.

13.6 El modelo IEEE 802 y los estándares

Hay una confusión adicional sobre la división en capas, ya que el IEEE divide la capa 2 de la pila de protocolos en dos tipos y usa el término *subcapas* para representar esta división. Sin embargo, la terminología puede ser engañosa debido a que los datos no pasan a través de las dos subcapas como cuando pasan a través de las capas convencionales. En su defecto veremos que las subcapas definen diversos aspectos de los protocolos de la capa 2, como el direccionamiento y la compartición de medios. La figura 13.5 enumera las dos subcapas conceptuales del IEEE y su propósito.

Subcapa	Expansión	Propósito
LLC	Control de enlace lógico	Direccionamiento y demultiplexación
MAC	Control de acceso a medios	Acceso a los medios compartidos

Figura 13.5 La división conceptual de la capa 2 en subcapas de acuerdo con el modelo del IEEE.

La subcapa de *control de enlace lógico (LLC)* especifica el direccionamiento y el uso de direcciones para la demultiplexación, como lo describiremos más adelante. La subcapa *control de acceso a medios (MAC)* especifica cómo comparten varias computadoras el medio relacionado.

En vez de usar nombres textuales para identificar el grupo de personas que trabajan en un estándar o el documento del estándar final, el IEEE asigna un identificador multiparte de la forma XXX.YYY.ZZZ. El valor numérico XXX indica la categoría del estándar, mientras que el sufijo YYY indica una subcategoría. Si una subcategoría es lo bastante grande, es posible agregar un tercer nivel para diferenciar entre estándares específicos. Por ejemplo, a las especificaciones de LAN se les asignó la categoría 802. Así, a cada grupo de trabajo que idee un estándar de LAN se le asigna un ID como 802.1, 802.2 y así en lo sucesivo. Cabe mencionar que ni el valor 802 ni los sufijos individuales transmiten un significado técnico; simplemente identifican los estándares. La figura 13.6 enlista algunos ejemplos de asignaciones del IEEE.

Como la figura indica, el IEEE ha creado muchos grupos de trabajo, cada uno de los cuales están dedicados a estandarizar un tipo específico de tecnología de red. Los grupos, que están formados por representantes de las comunidades industrial y académica, se reúnen con frecuencia para hablar sobre las metodologías e idear estándares. Cuando un grupo acuerda finalmente los detalles, escribe un documento estándar que el IEEE publica.

Cada vez que se necesita una nueva tecnología, se crea un grupo de trabajo; el grupo puede decidir si se disuelve o no después de producir un estándar. Por lo general el IEEE permite que un grupo funcional siga activo siempre y cuando progrese y la tecnología se considere todavía importante. Si un grupo de trabajo decide que la tecnología bajo investigación ya no es relevante, puede optar por disolverse sin producir un estándar. Como alternativa, el IEEE puede decidir que un estándar ya no es relevante. Por ejemplo, podría descubrirse una mejor tecnología y entonces la estandarización en curso ya no tendría sentido. En algunos casos, otra organización de estándares podría producir primero un estándar, con lo que el esfuerzo del IEEE sería redundante. De este modo, la figura 13.6 incluye temas que alguna vez fueron importantes pero que se han descartado.

13.7 Redes punto a punto y multiacceso

Cabe recordar que el término *punto a punto* se refiere a un mecanismo de comunicación que conecta precisamente dos entidades de comunicación. Las tecnologías LAN usan una alternativa en la que varias computadoras comparten un medio de tal forma que cualquier computadora en la LAN pueda comunicarse con las demás. Para describir los arreglos compartidos, usamos el término *multiacceso* y decimos que una LAN es una *red multiacceso*.

En general, las tecnologías de LAN proporcionan conexiones directas entre las entidades en comunicación. Los profesionales dicen que las LAN conectan *computadoras*, con el entendimiento de que dispositivos como una impresora también pueden conectarse a una LAN multiacceso.

ID	Tema
802.1	Protocolos de LAN de capa superior
802.2	Control de enlace lógico
802.3	Ethernet
802.4	Token bus (disuelto)
802.5	Token ring
802.6	Redes de área metropolitana (disuelto)
802.7	LAN de banda ancha que usa cable coaxial (disuelto)
802.9	LAN de servicios integrados (disuelto)
802.10	Seguridad de LAN interoperable (disuelto)
802.11	LAN inalámbrica (Wi-Fi)
802.12	Prioridad de demanda
802.13	Categoría 6 – LAN de 10Gb
802.14	Módems de cable (disuelto)
802.15	PAN inalámbrica 802.15.1 (Bluetooth) 802.15.4 (ZigBee)
802.16	Acceso inalámbrico de banda ancha 802.16e Banda ancha (móvil) inalámbrica
802.17	Anillo de paquetes resistente
802.18	TAG de regulación de radio
802.19	TAG sobre coexistencia
802.20	Acceso inalámbrico de banda ancha móvil
802.21	Entrega independiente de los medios
802.22	Red de área regional inalámbrica

Figura 13.6 Ejemplos de los identificadores que asignó el IEEE a diversos estándares de LAN.

13.8 Topologías de LAN

Como se han inventado muchas tecnologías de LAN, es importante conocer las similitudes y diferencias de cada tecnología específica. Para ayudar a entender las similitudes, cada red se clasifica en una categoría de acuerdo con su *topología* o forma general. Esta sección describe cuatro topologías básicas que se usan para construir redes LAN; en un capítulo posterior veremos las tecnologías específicas. La figura 13.7 ilustra las topologías.

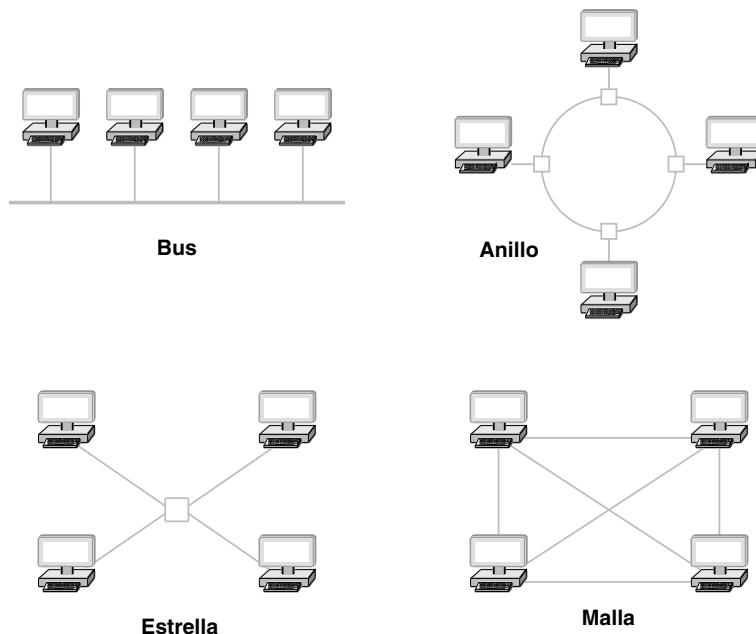


Figura 13.7 Cuatro topologías de red que se usan con redes LAN.

13.8.1 Topología de bus

El término *topología de bus* se acuñó para identificar a las redes (como la Ethernet original) que consisten en un solo cable al que se conectan las computadoras. Cualquier computadora conectada a un bus puede enviar una señal por el cable y todas las computadoras reciben la señal. Como todas las computadoras se conectan directamente al cable, cualquier computadora puede enviar datos a cualquier otra computadora. Desde luego que las computadoras conectadas a una red tipo bus deben coordinarse para asegurar que sólo una computadora envíe una señal en un momento dado. Algunas redes de bus conectan las computadoras a un dispositivo pequeño, y el bus (el cable compartido) se mantiene dentro del dispositivo.

13.8.2 Topología de anillo

Una red que usa una *topología de anillo* conecta las computadoras en un bucle cerrado: un cable conecta la primera computadora a una segunda computadora, otro cable conecta la segunda computadora a una tercera y así sucesivamente, hasta que un cable conecta la computadora final a la primera. Algunas tecnologías que usan una topología de anillo requieren que una computadora se conecte a un dispositivo pequeño y el anillo se forma dentro del dispositivo. La ventaja de usar un dispositivo independiente reside en la habilidad del anillo para continuar la operación, aun cuando algunas de las computadoras estén desconectadas. El nombre *anillo* surge debido a que podemos imaginar las computadoras (y los cables que las conectan) dispuestas en un círculo, como lo indica la figura 13.7. En la práctica, los cables de una red de anillo no forman un círculo. En su lugar, se extienden por los pasillos o se elevan verticalmente de un piso a otro de un edificio.

13.8.3 Topología de malla

Una red que usa una *topología de malla* proporciona una conexión directa entre cada par de computadoras. La principal desventaja de una malla se debe al costo: una red de malla que conecta n computadoras requiere:

$$\text{conexiones en una red de malla} = \frac{n!}{(n-2)! 2!} = \frac{n^2 - n}{2} \quad (13.1)$$

El punto importante es que el número de conexiones necesarias para una red de malla crece con más rapidez que el número de computadoras. Como las conexiones físicas son costosas, pocas redes LAN alámbricas emplean una topología de malla.

13.8.4 Topología de estrella

Una red usa una *topología de estrella* cuando todas las computadoras se conectan a un punto central. Puesto que una red en forma de estrella se asemeja a los rayos de una rueda, el centro de una red de estrella se conoce comúnmente como *concentrador* o *hub*. Un concentrador común consiste en un dispositivo electrónico que acepta datos de una computadora emisora y los entrega en el destino apropiado.

En la práctica, las redes de estrella raras veces tienen una forma simétrica en donde el concentrador se encuentra a la misma distancia de todas las computadoras. En su lugar, un concentrador reside por lo general en una ubicación separada de las computadoras conectadas a él. Por ejemplo, las computadoras pueden residir en oficinas individuales, mientras que el concentrador reside en una ubicación accesible para el personal encargado de las redes de la organización.

13.8.5 La razón de tener varias topologías

Cada topología tiene ventajas y desventajas. Una topología de anillo facilita a las computadoras coordinar el acceso y detectar si la red está operando correctamente. No obstante, sin un dispositivo externo, una red de anillo completa puede desactivarse si se corta uno de los cables. Una topología de estrella ayuda a proteger la red contra el daño de un solo cable, ya que cada cable conecta sólo una máquina. Un bus requiere menos alambres que una estrella, pero tiene la misma desventaja que un anillo: la red se desactiva si alguien corta por accidente el cable principal. Los capítulos posteriores que describen las

tecnologías específicas de red, proporcionan detalles adicionales sobre las diferencias. Por ahora basta con comprender que:

Las redes se clasifican en amplias categorías de acuerdo con su forma general. Aunque es posible usar una topología de malla, las principales topologías utilizadas con las LAN son: estrella, anillo y bus; cada una tiene sus ventajas y desventajas.

13.9 Identificación de paquetes, demultiplexación, direcciones MAC

Además de los estándares que especifican los detalles de varias tecnologías de LAN, el IEEE creó un estándar para el *direcccionamiento*. Para comprender el direccionamiento, considere los paquetes que recorren un medio compartido como en la figura 13.2.[†] En el caso más simple, cada paquete que viaja a través del medio compartido está destinado para un receptor específico, y sólo este receptor debe procesar dicho paquete. En los sistemas de comutación de paquetes, la demultiplexación usa un identificador conocido como *dirección*. A cada computadora se le asigna una dirección única y cada paquete contiene la dirección del receptor de destino.

En el esquema de direccionamiento del IEEE, cada dirección consiste en un valor binario de 48 bits. El IEEE usa el término *dirección de control de acceso al medio* (*dirección MAC*). Puesto que las direcciones de 48 bits se originaron con la tecnología de Ethernet, a menudo los profesionales de redes usan el término *dirección Ethernet*. Para garantizar que cada dirección sea única, el IEEE asigna una dirección a cada pieza de hardware de la interfaz de red. Por lo tanto, si un consumidor compra una *tarjeta de interfaz de red (NIC)* para su PC, la NIC contiene una dirección única del IEEE asignada cuando se fabricó el dispositivo.

En vez de asignar direcciones individuales, el IEEE asigna un bloque de direcciones a cada distribuidor de equipo y permite que éste asigne un valor único a cada dispositivo que fabrica. Así, una dirección de 48 bits se divide en un *identificador único de organización (OUI)* de 3 bytes que identifica al distribuidor del equipo y un bloque de 3 bytes que identifica a un *controlador de interfaz de red (NIC)* específico. La figura 13.8 ilustra la división.

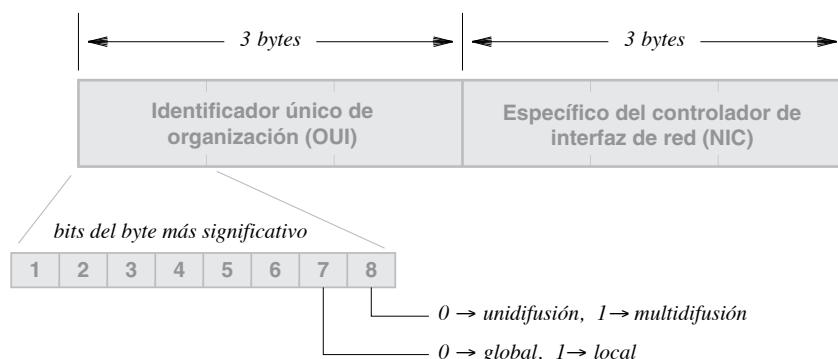


Figura 13.8 La división de una dirección MAC del IEEE de 48 bits.

[†] Encontrará la figura 13.2 en la página 221.

Lo interesante es que a los dos bits de menor orden del byte más significativo del OUI, se les asigna un significado especial, como lo indica la figura. El bit menos significativo del byte más significativo es un bit *multidifusión* que especifica si la dirección es *unidifusión* (0) o *multidifusión* (1), y el siguiente bit especifica si el OUI es único a nivel global (0) o se asigna en forma local (1). La siguiente sección explica la multidifusión. Las direcciones únicas a nivel global las asigna el IEEE; las direcciones asignadas en forma local están disponibles para el trabajo experimental o para las organizaciones que desean crear su propio espacio de direcciones.

13.10 Direcciones de unidifusión, difusión y multidifusión

El esquema de direccionamiento del IEEE soporta tres tipos de direcciones que corresponden a tres tipos de entrega de paquetes. La figura 13.9 proporciona un resumen.

Tipo de dirección	Significado y entrega de paquetes
Unidifusión (unicast)	Identifica en forma única a una computadora y especifica que sólo la computadora identificada debe recibir una copia del paquete
difusión (broadcast)	Relaciona a todas las computadoras y especifica que cada computadora en la red debe recibir una copia del paquete.
multidifusión (multicast)	Identifica un subconjunto de las computadoras en una red dada, y especifica que cada computadora del subconjunto debe recibir una copia del paquete

Figura 13.9 Los tres tipos de direcciones MAC y los significados correspondientes.

Tal vez parezca extraño que el formato de direcciones del IEEE reserve un bit para diferenciar entre la unidifusión y la multidifusión, pero no proporciona una forma de designar una dirección de difusión. El estándar especifica que una *dirección de difusión* consiste en 48 bits, todos 1. Por lo tanto, una dirección de difusión tiene establecido el bit de multidifusión. En concepto, la difusión puede considerarse una forma especial de multidifusión. Es decir, cada dirección de multidifusión corresponde a un grupo de computadoras, siendo que la dirección de difusión corresponde a un grupo amplio que incluye a todas las computadoras de la red.

13.11 Difusión, multidifusión y entrega multipunto eficiente

Las direcciones de difusión y multidifusión son especialmente útiles en las redes LAN ya que permiten una entrega eficiente a muchas computadoras. Para entender la eficiencia, recuerde que la mayoría de las tecnologías de LAN transmiten paquetes a través de un medio compartido. En una LAN común, cada computadora monitorea el medio compartido, extrae una copia de cada paquete y luego examina la dirección del paquete para determinar si éste debe procesarse o ignorarse. El algoritmo 13.1 especifica los pasos que toma una computadora para procesar un paquete entrante.

Algoritmo 13.1

Propósito:

Manejar un paquete que haya llegado a través de una LAN

Método:

```
Extraer la dirección de destino D del paquete;  
if (D coincide con mi dirección de unidifusión ) {  
    aceptar y procesar el paquete;  
} else if (D coincide con la dirección de difusión ) {  
    aceptar y procesar el paquete;  
} else if (D coincide con una de las direcciones de multidifusión para  
un grupo de multidifusión del cual sea yo miembro ) {  
    aceptar y procesar el paquete;  
} else {  
    ignorar el paquete;  
}
```

Algoritmo 13.1 Algoritmo de procesamiento de paquetes que se utiliza en una LAN.

La eficiencia debe ser clara en el algoritmo. En el caso de la difusión o la multidifusión, se transmite una sola copia del paquete a través del medio compartido y todas las computadoras reciben y procesan esa copia. Por ejemplo, considere el caso específico de la difusión: en vez de realizar N transmisiones independientes, cada una enviando una copia del paquete a cada una de las computadoras de la red, un solo emisor transmite una copia del paquete que contiene la dirección de difusión y todas las computadoras reciben la copia.

13.12 Tramas y entramado

El capítulo 9 presenta el concepto del entramado en el contexto de los sistemas de comunicación asíncronos como un mecanismo que permite a un receptor saber dónde comienza y dónde termina un mensaje. En un sentido más general, usamos el término *entramado* para referirnos a la estructura que se agrega a una secuencia de bits o bytes que permite a un emisor y un receptor ponerse de acuerdo en cuanto al formato exacto del mensaje. En una red de conmutación de paquetes, cada *trama* corresponde a un paquete. Una trama consiste en dos partes conceptuales:

- Un encabezado que contiene metadatos, como una dirección
- Una carga útil que contiene los datos a enviar

Un *encabezado* de trama contiene la información que se usa para procesar dicha trama. El encabezado contiene generalmente una dirección que especifica el receptor deseado. El área de la *carga útil* contiene el mensaje a enviar y por lo general es mucho más grande que el encabezado de la trama. En la mayoría de las tecnologías de redes, el mensaje es *opaco* en el sentido en que la red sólo examina el encabezado de la trama. De esta forma, la carga útil puede contener una secuencia arbitraria de bytes que sólo son significativos para el emisor y el receptor.

La disposición común de una trama es que el encabezado se transmita antes de la carga útil, lo cual permite a un receptor comenzar a procesar la trama a medida que vayan llegando los bits. Algunas tecnologías delimitan cada trama enviando un preludio corto antes de la trama y un colofón corto después de ésta. La figura 13.10 ilustra el concepto.

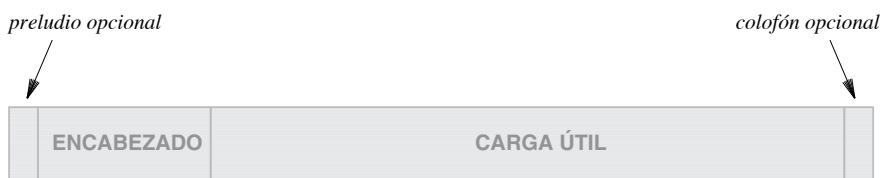


Figura 13.10 Estructura típica de una trama en una red de conmutación de paquetes.

Para comprender cómo funciona el entramado, considere un ejemplo simplificado donde se utilizan bytes.[†] Es decir, suponga que un mecanismo de comunicaciones de datos puede transferir un byte arbitrario de 8 bits de un emisor a un receptor, e imagine que el mecanismo se usa para enviar paquetes. Suponga que el encabezado de un paquete consiste en 6 bytes y que la carga útil consiste en un número arbitrario de bytes. Usaremos un solo byte para marcar el inicio de una trama y uno solo para marcar el final de la misma. En el conjunto de caracteres ASCII, el carácter *inicio de encabezado* (*SOH*) marca el inicio de una trama y el carácter *fin de la transmisión* (*EOT*) marca el final. La figura 13.11 ilustra el formato.

[†] La mayoría de las tecnologías de redes utiliza bits en lugar de bytes; en este ejemplo utilizamos bytes para hacerlo más intuitivo.



Figura 13.11 Un formato de trama de ejemplo que usa caracteres SOH y EOT para delimitar una trama.

El formato de ejemplo parece tener una sobrecarga innecesaria. Para entender por qué, considere lo que ocurre cuando un emisor transmite dos tramas sin ningún retardo entre ellas. Al final de la primera trama, el emisor transmite EOT y luego, sin ningún retardo, el emisor transmite SOH para iniciar la segunda trama. En dichas circunstancias sólo se necesita un carácter para separar dos bloques de datos; un esquema de entramado que delimita tanto el inicio como el final de cada trama parece enviar un carácter adicional innecesario entre tramas.

La ventaja de enviar un carácter al final de una trama se vuelve clara si consideramos que la transmisión de paquetes es asíncrona y pueden ocurrir errores. Para la comunicación asíncrona, al usar un carácter EOT para marcar el final de una trama, el receptor puede procesar esta trama sin tener que esperar el inicio de una trama sucesiva. En caso de un error, el uso de SOH y EOT para encerrar la trama ayuda en el proceso de recuperación y sincronización; si un emisor falla durante la transmisión de una trama, el receptor podrá determinar si llegó una trama incompleta.

13.13 Relleno de bytes y bits

En el conjunto de caracteres ASCII, SOH tiene el valor hexadecimal 0x01 y EOT tiene el valor hexadecimal 0x04. Surge la pregunta: ¿qué ocurre si la carga útil de una trama incluye uno o más bytes con el valor 0x01 o 0x04? La respuesta está en una técnica que permite la transmisión de datos sin confusión.

Para distinguir entre la información de datos y la de control (como los delimitadores de tramas), un emisor por lo general cambia los datos para sustituir cada byte de control con una secuencia, mientras que el receptor sustituye esa secuencia con el valor original. Como resultado, una trama puede transmitir cualquier tipo de datos y el sistema nunca confunde éstos con la información de control. La técnica se conoce como *relleno de bytes*; algunas veces se usan los términos *relleno de datos* y *relleno de caracteres*. Por su parte, el *relleno de bits* es una técnica similar que se usa con los sistemas que transfieren un flujo de bits.

Como ejemplo de relleno de bytes, considere una trama como la que se muestra en la figura 13.11. Puesto que se usan los caracteres SOH y EOT para delimitar la trama, esos dos bytes no deben aparecer en la carga útil. El relleno de bytes resuelve el problema al reservar un tercer carácter para marcar las ocurrencias de los caracteres reservados en los datos. Por ejemplo, suponga que se selecciona el carácter ASCII ESC (valor hexadecimal 1B) como el tercer carácter. Cuando ocurre alguno de los tres caracteres

especiales en los datos, el emisor sustituye el carácter con una secuencia de dos caracteres. La figura 13.12 enumera una posible asignación.

Byte en la carga útil	Secuencia enviada
SOH	ESC A
EOT	ESC B
ESC	ESC C

Figura 13.12 Un ejemplo de relleno de bytes en donde se asigna cada carácter especial a una secuencia de 2 caracteres.

Como se especifica en la figura, el emisor sustituye cada ocurrencia de SOH en los datos por los dos caracteres ESC seguidos de A, cada ocurrencia de EOT por los caracteres ESC seguidos de B y cada ocurrencia de ESC por los dos caracteres ESC seguidos de C. Un receptor invierte la asignación al buscar ESC seguido de un carácter A, B o C y sustituye esa combinación de 2 caracteres con el carácter individual apropiado. La figura 13.13 muestra una carga útil de ejemplo y la misma carga útil después de que ocurre el relleno de bytes. Cabe mencionar que, una vez que se realiza el relleno de bytes, no aparecen SOH ni EOT en ninguna parte de la carga útil.

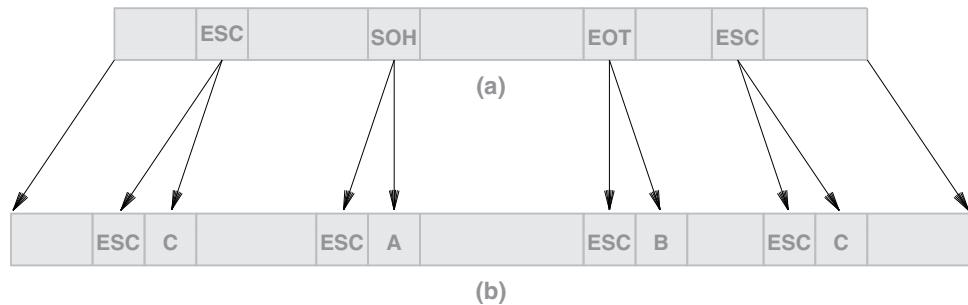


Figura 13.13 Ilustración de (a) los datos originales y (b) una versión después de realizar el relleno de bytes.

13.14 Resumen

Las redes de datos pueden clasificarse de dos formas: las que usan commutación de circuitos y las que usan commutación de paquetes. La commutación de paquetes, que constituye la base de Internet, es una forma de multiplexación estadística en la que los emisores dividen los mensajes en paquetes pequeños. Las tecnologías de redes de commutación de paquetes se clasifican como *redes de área local* (LAN),

redes de área amplia (WAN) y *redes de área metropolitana* (MAN); las LAN y las WAN son las más populares.

Una organización llamada IEEE creó estándares para las redes de datos. Los estándares del IEEE especifican principalmente los detalles para las redes LAN y se concentran en las primeras dos capas de la pila de protocolos.

Se utilizan cuatro formas o topologías básicas para caracterizar a las redes LAN: *bus*, *estrella*, *anillo* y *malla*. Las topologías de malla se usan rara vez debido a que son costosas.

Cada paquete enviado a través de una LAN contiene una dirección MAC que especifica el receptor deseado. El estándar del IEEE para las direcciones MAC especifica un valor de 48 bits dividido en dos campos: uno que identifica a la organización que asigna la dirección y otro que proporciona un valor único para la pieza específica de hardware a la que se asigna la dirección. Una dirección puede especificar *unidifusión* (una sola computadora), *difusión* (todas las computadoras en una LAN dada) o *multidifusión* (un subconjunto de computadoras en una LAN).

El término *trama* se usa para especificar el formato de un paquete en una red específica. Una trama consiste en dos partes conceptuales: un encabezado que contiene metadatos y un área de carga útil que contiene los datos que se van a enviar. Para una red que transmite caracteres, es posible formar una trama si se usa un valor de byte para indicar el inicio de la trama y otro para indicar el final de ésta.

Las técnicas de relleno de bytes o bits permiten reservar bytes (o secuencias de bits) y usarlos para marcar el inicio y el final de una trama. Para asegurar que una carga útil no contenga bytes reservados (cadenas de bits), un emisor sustituye las ocurrencias de los valores reservados antes de la transmisión y un receptor invierte el cambio para obtener los datos originales.

EJERCICIOS

- 13.1** ¿Qué es la conmutación de circuitos y cuáles son sus características principales?
- 13.2** En una red de conmutación de circuitos, ¿pueden varios circuitos compartir una sola fibra óptica? Explique.
- 13.3** En un sistema de conmutación de paquetes, ¿cómo es que un emisor transfiere un archivo grande?
- 13.4** Si alguien desea difundir una copia de una presentación en video, ¿es preferible un sistema de conmutación de circuitos o uno de conmutación de paquetes? ¿Por qué?
- 13.5** ¿Cuáles son las características de las redes LAN, MAN y WAN?
- 13.6** Nombre las dos subcapas de protocolos de la capa 2 que define el IEEE y mencione el propósito de cada una.
- 13.7** ¿Qué es una red punto a punto?
- 13.8** ¿Cuáles son las cuatro topologías básicas de las redes LAN?
- 13.9** ¿Pueden los cables de una red tipo anillo distribuirse en línea recta (por ejemplo, a lo largo de un pasillo)?
- 13.10** En una red tipo malla, ¿cuántas conexiones se requieren entre 20 computadoras?
- 13.11** Dada una dirección MAC del IEEE, ¿cómo podemos saber si la dirección se refiere a una unidifusión?
- 13.12** Defina las direcciones de unidifusión, multidifusión y difusión. Explique el significado de cada una.

- 13.13** ¿Cómo decide una computadora conectada a una LAN compartida si acepta un paquete o no?
- 13.14** ¿Qué término se usa para describir los metadatos que acompañan a un paquete?
- 13.15** Proporcione una definición del término trama.
- 13.16** ¿Por qué se necesita el relleno de bytes?
- 13.17** Escriba un par de programas de computadora: uno que acepte un archivo de datos como entrada y produzca una versión con relleno de bytes, de acuerdo con la asignación en la figura 13.12, y otro que elimine el relleno de bytes. Muestre que sus programas pueden interoperar con los que escriban otros programadores.

Contenido del capítulo

- 14.1 Introducción, 239
- 14.2 Una taxonomía de los mecanismos para acceso compartido, 239
- 14.3 Asignación estática y dinámica de canales, 240
- 14.4 Protocolos de canalización, 241
- 14.5 Protocolos de acceso controlado, 242
- 14.6 Protocolos de acceso aleatorio, 244
- 14.7 Resumen, 250

14

La subcapa MAC del IEEE

14.1 Introducción

Los capítulos de esta parte del libro cubren las redes que usan commutación de paquetes. El capítulo anterior presenta el concepto de la commutación de paquetes y define los dos tipos básicos de redes de commutación de paquetes: WAN y LAN. El capítulo también introduce el modelo IEEE para los estándares y explica que el IEEE divide la capa de enlace de datos en dos subcapas.

Este capítulo continúa la explicación mediante un análisis de la subcapa MAC del IEEE. También se refiere a los protocolos multiacceso y a la asignación de caracteres tanto estática como dinámica. Los capítulos posteriores de esta parte del libro hablan sobre las tecnologías de redes específicas que usan los mecanismos de acceso aquí explicados.

14.2 Una taxonomía de los mecanismos para acceso compartido

¿Cómo coordinan varias computadoras independientes el acceso a un medio compartido? Hay tres enfoques generales: pueden usar una técnica de multiplexación modificada, pueden participar en un algoritmo distribuido para el acceso controlado, o pueden usar una estrategia de acceso aleatorio. La figura 14.1 ilustra la taxonomía, incluyendo las formas específicas, de cada enfoque.

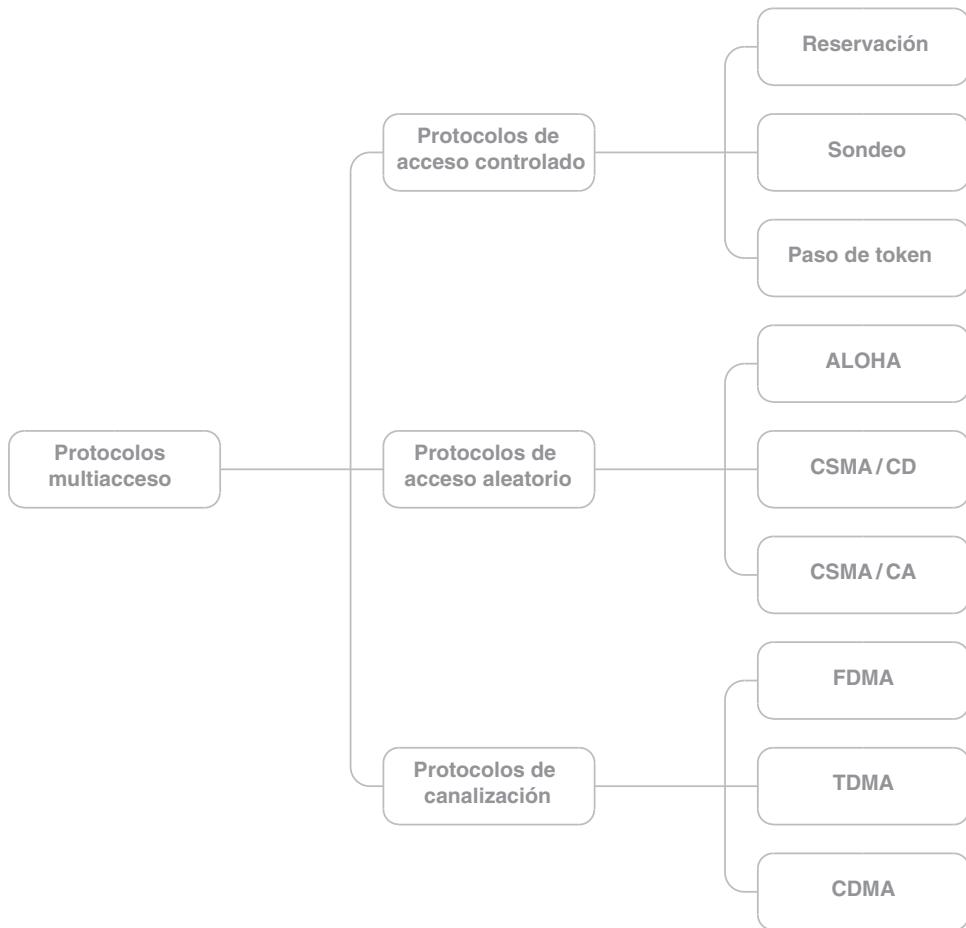


Figura 14.1 Una taxonomía de protocolos que controlan el acceso a un medio compartido.

14.3 Asignación estática y dinámica de canales

Usamos el término *canalización* para referirnos a una correlación entre una comunicación específica y un canal en el sistema de transmisión. La canalización se relaciona con las técnicas de multiplexación que se explican en el capítulo 11. Por ejemplo, considere un mecanismo de multiplexación por división de frecuencias (FDM). La mayoría de los sistemas FDM asignan a cada par de entidades de comunicación una frecuencia de portadora. Es decir, a cada par se le asigna un canal único. Además, la asignación entre un par de entidades y una frecuencia de portadora no cambia. En tales situaciones, describimos la correlación entre las entidades de comunicación y un canal como de *1 a 1 y estática*.

La asignación estática de canales funciona bien en situaciones en las que el conjunto de entidades de comunicación se conoce de antemano y no cambia. Sin embargo, en muchas redes el conjunto de entidades que usan la red varía con el tiempo. Por ejemplo, considere los teléfonos celulares de una ciudad. Los usuarios se mueven y pueden encender o apagar su teléfono en cualquier momento. Por consiguiente, el conjunto de teléfonos celulares que opera dentro del rango de una torre celular dada, varía constantemente. En tales situaciones se necesita una *asignación dinámica de canales*: la correlación puede hacerse cuando aparezca una nueva estación o teléfono celular, y puede eliminarse cuando desaparezca.

Para resumir:

La asignación estática de canales resulta suficiente cuando el conjunto de entidades de comunicación se conoce de antemano y no cambia; la mayoría de las redes requieren una forma de asignación dinámica de canales.

14.4 Protocolos de canalización

Los protocolos de canalización extienden las técnicas de multiplexación cubiertas en el capítulo 11. La figura 14.2 enumera las principales técnicas de canalización.

Protocolo	Expansión
FDMA	Acceso múltiple por división de frecuencias
TDMA	Acceso múltiple por división de tiempo
CDMA	Acceso múltiple por división de código

Figura 14.2 Los tres tipos principales de canalización.

14.4.1 FDMA

Como se indica en la figura, las técnicas de canalización emplean la multiplexación por división de frecuencias, por división de tiempo o por división de código. Por ejemplo, el *acceso múltiple por división de frecuencias (FDMA)* extiende la multiplexación por división de frecuencias. En esencia, la extensión consiste en un mecanismo que permite que estaciones independientes elijan frecuencias de portadora que no estén en conflicto con las portadoras utilizadas por otras estaciones. ¿Cómo asigna la FDMA las portadoras? En algunos sistemas, un controlador central proporciona una asignación dinámica. Cada vez que aparece una nueva estación, ésta usa un canal de control reservado para comunicarse con el controlador. La estación hace una solicitud y el controlador selecciona una frecuencia que no se utilice en ese momento para luego informar a la estación. Después del intercambio inicial, la estación usa la frecuencia de portadora asignada (es decir, el canal asignado) para toda la comunicación.

14.4.2 TDMA

La extensión de la multiplexación por división de tiempo, conocida como *acceso múltiple por división de tiempo (TDMA)*, es similar a la extensión de la multiplexación por división de frecuencias. En el caso más simple, a cada participante activo se le asigna un número de secuencia de 1 a N y las estaciones transmiten en el orden 1, 2, 3, … N . Al igual que con la FDMA, algunos sistemas TDMA ofrecen asignación dinámica: cuando la estación aparece por primera vez en la red, se le asigna un intervalo de tiempo.

14.4.3 CDMA

La multiplexación por división de código permite que varias estaciones transmitan al mismo tiempo mediante la codificación matemática de cada transmisión. El *acceso múltiple por división de código (CDMA)* que se explica en el capítulo 11, constituye la principal aplicación de la multiplexación por división de código.

14.5 Protocolos de acceso controlado

Los protocolos de acceso controlado proporcionan una versión distribuida de la multiplexación estadística. La figura 14.3 menciona las tres formas principales:

Tipo	Descripción
Sondeo	Un controlador centralizado sondea repetidamente las estaciones y permite que cada una transmita un paquete
Reservación	Las estaciones envían una solicitud para la siguiente ronda de transmisión de datos
Paso de token	Las estaciones ponen a circular una señal tipo token; cada vez que recibe la señal, una estación transmite un paquete

Figura 14.3 Los tipos principales de protocolos de acceso controlado.

14.5.1 Sondeo

Una red que emplea el *sondeo*, o *polling*, usa un controlador centralizado que recorre las estaciones que están en la red y ofrece a cada una de ellas una oportunidad de transmitir un paquete. El algoritmo 14.1 indica los pasos que sigue un controlador. El paso de la selección es importante ya que significa que un controlador puede elegir qué estación sondear en un momento dado. Hay dos políticas de sondeo generales:

- Orden por turnos
- Orden por prioridad

El orden por turnos significa que cada estación tiene la misma oportunidad de transmitir paquetes. El orden por prioridad significa que algunas estaciones tendrán mayor posibilidad de enviar. Por ejemplo, el orden por prioridad podría usarse para asignar a un teléfono IP más prioridad que a una computadora personal.

Algoritmo 14.1

Propósito:

Transmisión de control de los paquetes mediante sondeo

Método:

El controlador se repite en forma indefinida {

 Seleccionar una estación S y enviar un mensaje de sondeo a S;

 Esperar a que S responda enviando un paquete o nada;

}

Algoritmo 14.1 Acceso controlado mediante el sondeo.

14.5.2 Reservación

El sistema de *reservación* se utiliza a menudo con la transmisión por satélite y emplea un proceso de dos pasos en el que cada ronda de transmisiones de paquetes se planea por adelantado. Por lo general, los sistemas de reservación tienen un controlador central que sigue el algoritmo 14.2.

Algoritmo 14.2

Propósito:

Controlar la transmisión de paquetes mediante la reservación

Método:

El controlador se repite en forma indefinida {

 Formar una lista de estaciones que tengan un paquete para enviar;

 Dejar que las estaciones en la lista transmitan;

}

Algoritmo 14.2 Acceso controlado mediante la reservación.

En el primer paso, cada emisor potencial especifica si tiene un paquete para enviar durante la siguiente ronda, y el controlador transmite una lista de las estaciones que van a transmitir. En el segundo paso, las estaciones usan la lista para saber cuándo deben transmitir. Existen variaciones donde un

controlador usa un canal alternativo para acumular reservaciones para la siguiente ronda, mientras que en el canal principal se lleva a cabo la ronda actual de transmisiones.

14.5.3 Paso de token

El *paso de token* se ha utilizado en varias tecnologías de LAN y se asocia más comúnmente con las topologías de anillo.[†] Para comprender el paso de token, imagine un conjunto de computadoras conectadas en un anillo e imagine que, en cualquier instante, sólo una de las computadoras recibe una señal especial de control conocida como *token*. Para controlar el acceso, cada computadora sigue el algoritmo 14.3.

Algoritmo 14.3

Propósito:

Controlar la transmisión de paquetes mediante el paso de un token

Método:

Cada computadora en la red repite {

Esperar a que llegue el token;

Transmitir un paquete si hay uno esperando el envío;

Enviar el token a la siguiente estación;

}

Algoritmo 14.3 Acceso controlado mediante el paso de token.

En un sistema de paso de token, cuando ninguna estación tiene paquetes por enviar, el token circula continuamente entre todas las estaciones. Para una topología de anillo, el orden de circulación se define con base en el anillo. Es decir, si un anillo está dispuesto para enviar mensajes en el sentido de las manecillas del reloj, la *siguiente estación* indicada por el algoritmo se remite a la siguiente estación física en el sentido de las manecillas del reloj. Cuando se aplica el paso de token a otras topologías (como la de bus), a cada estación se le asigna una posición dentro de una secuencia lógica y la señal token se pasa de acuerdo con la secuencia asignada.

14.6 Protocolos de acceso aleatorio

Muchas redes, en especial las LAN, no emplean un mecanismo de acceso controlado. En su lugar, un conjunto de computadoras conectadas a un medio compartido intentan acceder a éste sin coordinación. Se usa el término *aleatorio* debido a que el acceso sólo ocurre cuando cierta estación tiene un paquete para enviar y se emplea la aleatorización para evitar que todas las computadoras de una LAN intenten usar el medio al mismo tiempo. Las descripciones de los siguientes métodos específicos aclararán el uso de la aleatorización. La figura 14.4 enumera los tres métodos de acceso aleatorio que se describen.

[†] Aunque las redes LAN más antiguas usaban la tecnología de anillo de paso de token, su popularidad ha decaído y quedan pocas de estas redes.

Tipo	Descripción
ALOHA	Protocolo histórico utilizado en una de las primeras redes de radio en Hawái; es popular en los libros de texto y fácil de analizar, pero no se usa en redes reales
CSMA/CD	Acceso múltiple por detección de portadora con detección de colisiones La base de la Ethernet original, pero ya no se usa con la Ethernet comutada
CSMA/CA	Acceso múltiple por detección de portadora con evasión de colisiones La base de las redes Wi-Fi inalámbricas

Figura 14.4 Tres protocolos de acceso aleatorios.

14.6.1 ALOHA

Una de las primeras redes en Hawái, conocida como *ALOHAnet*, fue la pionera del concepto del acceso aleatorio. Aunque esta red ya no está en uso, sus ideas se extendieron. La red consistía en un solo transmisor poderoso colocado en una ubicación geográfica central y rodeado por un conjunto de estaciones, cada una de las cuales correspondían a una computadora. Cada estación tenía un transmisor capaz de llegar al transmisor central (pero no era lo bastante poderoso como para llegar a todas las demás estaciones). Como se muestra en la figura 14.5, ALOHAnet usaba dos frecuencias de portadora: una a 413.475 MHz para el tráfico *saliente* que enviaba el transmisor central a todas las estaciones y otra a 407.305 MHz para el tráfico *entrante* que enviaban las estaciones al transmisor central.

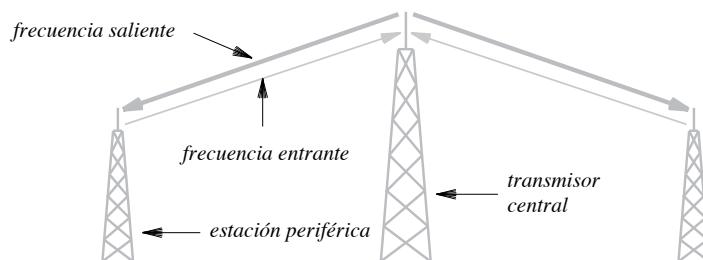


Figura 14.5 Ilustración de las frecuencias saliente y entrante en ALOHAnet.

El protocolo ALOHA es simple: cuando una estación tiene un paquete para enviar, lo transmite en la frecuencia entrante. El transmisor central repite la transmisión en la frecuencia saliente (que

todas las estaciones pueden recibir). Para asegurar que la transmisión sea exitosa, una estación emisora escucha el canal saliente. Si llega una copia de su paquete, la estación emisora se mueve al siguiente paquete; si no llega ninguna copia, la estación emisora espera por un breve lapso y luego vuelve a intentar.

¿Por qué podría un paquete no llegar? La respuesta es la interferencia: si dos estaciones intentan transmitir en la frecuencia entrante al mismo tiempo, las señales interferirán y las dos transmisiones serán ininteligibles. Usamos el término *colisión* y decimos que los dos paquetes transmitidos tuvieron una *colisión* en el medio. Para hacerse cargo de una colisión, el protocolo requiere que el emisor *retransmita* cada paquete perdido. La idea es común y aparece en muchos protocolos de red.

El tiempo de espera antes de la retransmisión debe elegirse con cuidado. De lo contrario, dos estaciones esperarán exactamente el mismo tiempo antes de volver a enviar e interferirán una con la otra de nuevo. Por lo mismo, se usa la aleatorización (es decir, cada estación selecciona un retraso aleatorio) para que la probabilidad de interferencia sea mucho menor. El análisis muestra que cuando ALOHAnet estaba muy cargada ocurrían muchas colisiones. Incluso con la aleatorización, las colisiones bajaron la transferencia exitosa de datos en ALOHAnet aproximadamente al 18% de la capacidad del canal (es decir, se utilizaba sólo el 18% del canal).

14.6.2 CSMA/CD

En 1973, los investigadores en Xerox PARC crearon una tecnología de red extremadamente exitosa que utilizaba un protocolo de acceso aleatorio. En 1978, Digital Equipment Corporation, Intel y Xerox crearon un estándar (denominado de manera informal como el *estándar DIX*). Este estándar se dio a conocer con el nombre de *Ethernet*; la tecnología de la Ethernet original consistía en un solo cable largo al que se conectaban las computadoras.[†] El cable servía como medio compartido: en vez de difundir transmisiones de radiofrecuencia a través de la atmósfera, Ethernet transmitía señales por un cable. Además, en vez de usar dos frecuencias y un transmisor central, Ethernet permitía que toda la comunicación se realizara a través del cable compartido. A pesar de sus diferencias, Ethernet y ALOHAnet tenían que resolver el mismo problema básico: si dos estaciones trataban de transmitir al mismo tiempo, las señales interferían y ocurría una colisión.

Ethernet ofreció tres innovaciones en cuanto a la forma de manejar las colisiones:

- Detección de portadora
- Detección de colisión
- Retroceso exponencial binario

Detección de portadora. En vez de permitir que una estación transmitiera cada vez que estuviera listo un paquete, Ethernet requería que cada estación monitoreara el cable para detectar si había otra transmisión en progreso. El mecanismo, conocido como *detección de portadora*, evita los problemas más obvios de colisiones y mejora de manera considerable el uso de la red.

[†] El siguiente capítulo describe el cableado de la Ethernet moderna.

Detección de colisión. Aunque se usa la detección de portadora, puede ocurrir una colisión si dos estaciones esperan a que se detenga una transmisión, encuentran que el cable está inactivo y ambas comienzan a transmitir. Parte del problema se debe a que, aun a la velocidad de la luz, se requiere cierto tiempo para que una señal recorra el cable. Por ende, una estación en un extremo del cable no puede saber al instante cuando una estación en el otro extremo comienza a transmitir.

Para manejar las colisiones, cada estación monitorea el cable durante la transmisión. Si la señal en el cable difiere de la señal que está enviando la estación, significa que ocurrió una colisión. La técnica se conoce como *detección de colisión*. Cuando se detecta una colisión, la estación emisora aborta la transmisión.

Hay muchos detalles que complican la transmisión en Ethernet. Por ejemplo, después de una colisión la transmisión no se aborta sino hasta que se haya enviado suficientes bits como para garantizar que las señales que hicieron colisión lleguen a todas las estaciones. Además, después de una transmisión las estaciones deben esperar un *espacio entre paquetes* (de 9.6μ segundos para la Ethernet original de 10 Mbps) para asegurar que todas las estaciones detecten una red inactiva y tengan la oportunidad de transmitir. Dichos detalles ilustran el cuidado con el que se diseñó la tecnología.

Retroceso exponencial binario. Ethernet hace más que sólo detectar las colisiones, también se recupera de ellas. Después de que ocurre una colisión, la computadora debe esperar a que el cable esté inactivo de nuevo antes de transmitir una trama. Al igual que con ALOHAnet, se utiliza la aleatorización para evitar que varias estaciones transmitan al mismo tiempo tan pronto como el cable esté inactivo. Es decir, el estándar especifica un retraso máximo d y requiere que cada estación seleccione un retraso aleatorio menor a d cuando ocurra una colisión. En la mayoría de los casos cuando dos estaciones seleccionan cada una un valor aleatorio, la estación que seleccione el menor retraso procederá a enviar un paquete y la red reanudará su operación normal.

En el caso en que dos o más computadoras seleccionen casi la misma cantidad de retraso, ambas comenzarán a transmitir prácticamente al mismo tiempo y se producirá una segunda colisión. Para evitar una secuencia de colisiones, Ethernet requiere que cada computadora duplique el rango a partir del cual se seleccione un retraso después de cada colisión. Una computadora selecciona un retraso aleatorio de 0 a d después de una colisión, un retraso aleatorio entre 0 y $2d$ después de una segunda colisión, entre 0 y $4d$ después de una tercera, y así en lo sucesivo. Después de unas cuantas colisiones, el rango a partir del cual se selecciona un valor se hace grande. Así, una computadora elegirá un retraso aleatorio más corto que las otras y transmitirá sin una colisión.

Al proceso de duplicar el rango del retraso aleatorio después de cada colisión se le conoce como *retroceso exponencial binario*. En esencia, esto significa que una Ethernet puede recuperarse con rapidez después de una colisión debido a que cada computadora acepta esperar tiempos más largos entre cada intento cuando el cable esté ocupado. Incluso en el poco probable suceso de que dos o más computadoras seleccionen retrasos que sean aproximadamente iguales, el retroceso exponencial garantiza que se reducirá la contención para el cable después de unas cuantas colisiones.

La combinación de las técnicas antes descritas se conoce por el nombre *acceso múltiple por detección de portadora con detección de colisiones (CSMA/CD)*. El algoritmo 14.4 sintetiza la técnica CSMA/CD.

En la figura, la computadora 1 puede comunicarse con la computadora 2 pero no puede recibir la señal de la computadora 3. Por consiguiente, si la computadora 3 transmite un paquete a la computadora 2, el mecanismo de detección de portadora de la computadora 1 no detectará la transmisión. De manera similar, si las computadoras 1 y 3 transmiten al mismo tiempo, sólo la computadora 2 detectará una colisión. El problema se conoce algunas veces como *el problema de la estación oculta* ya que algunas estaciones no pueden ver a otras.

Para asegurar que todas las estaciones comparten correctamente el mismo medio de transmisión, las LAN inalámbricas usan un protocolo modificado de acceso que se conoce como *acceso múltiple por detección de portadora con evasión de colisiones (CSMA/CA)*. En vez de depender de las demás computadoras para recibir todas las transmisiones, el protocolo CSMA/CA desencadena una transmisión breve desde el receptor deseado antes de transmitir un paquete. La idea es que si el emisor y el receptor transmiten un mensaje, todas las computadoras dentro de su rango sepan que está comenzando la transmisión de un paquete. La figura 14.7 ilustra la secuencia.

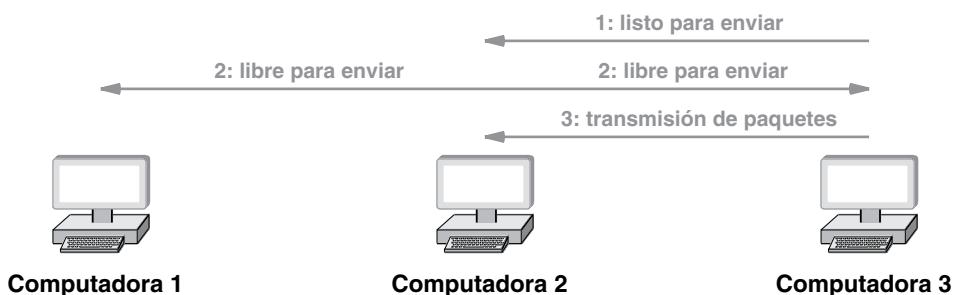


Figura 14.7 Una secuencia de mensajes que se envía cuando la computadora 3 transmite un paquete a la computadora 2.

En la figura, la computadora 3 envía un mensaje corto para anunciar que está lista para transmitir un paquete a la computadora 2, y la computadora 2 responde enviando un mensaje corto para anunciar que está lista para recibir el paquete. Todas las computadoras en el rango de la computadora 3 reciben el anuncio inicial y todas las computadoras en el rango de la computadora 2 reciben la respuesta. Como resultado, incluso aunque no pueda recibir la señal o detectar una portadora, la computadora 1 sabe que se está llevando a cabo la transmisión de un paquete.

Las colisiones de los mensajes de control pueden ocurrir al usar CSMA/CA, pero pueden manejarse fácilmente. Por ejemplo, en la figura, si las computadoras 1 y 3 intentan cada una transmitir un paquete a la computadora 2 exactamente al mismo tiempo, habrá una colisión entre sus mensajes de control. La computadora 2 detectará la colisión y no responderá. Cuando ocurre una colisión, las estaciones emisoras aplican el retroceso aleatorio antes de volver a enviar los mensajes de control. Puesto que los mensajes de control son mucho más cortos que un paquete, la probabilidad de una segunda colisión es muy baja. Finalmente uno de los dos mensajes de control llegará intacto y la computadora 2 transmitirá una respuesta.

Podemos resumir:

Como las computadoras en una LAN inalámbrica pueden abarcar distancias mayores de las que puede abarcar una señal, las LAN inalámbricas usan CSMA/CA, donde las computadoras emisora y receptora envían cada una un mensaje de control antes de que ocurra la transmisión de paquetes.

14.7 Resumen

La capa MAC del IEEE contiene protocolos que controlan el acceso a un medio compartido. Los protocolos de canalización consisten en extensiones de la multiplexación por división de tiempo, de frecuencias y de código; las extensiones se conocen como *acceso múltiple por división de frecuencias, por división de tiempo y por división de código*, respectivamente. Es posible la asignación estática o dinámica de canales.

Los protocolos de acceso controlado permiten que estaciones independientes se involucren en la multiplexación estadística. El sondeo usa un controlador central que revisa repetidas veces si las estaciones están listas para enviar un paquete. Un sistema de reservación, que se utiliza comúnmente con los satélites, requiere que las estaciones declaren si están listas o no para la siguiente ronda de transmisión. El paso de token, que a menudo se usa con la topología de anillo, pasa una señal de control entre las estaciones; al recibir el token una estación podrá transmitir un paquete.

Los protocolos de acceso aleatorio permiten a las estaciones competir por el acceso. El protocolo histórico ALOHA utilizaba dos frecuencias, una para transmisiones entrantes y otra para salientes; si una estación no recibía una copia de su paquete, la estación volvía a transmitir. La Ethernet original popularizó el *acceso múltiple por detección de portadora con detección de colisiones* (CSMA/CD), que se utilizó para controlar el acceso a un cable compartido. Además de evitar que una estación transmita mientras haya otra transmisión en progreso, CSMA/CD usa el *retroceso exponencial binario* para recuperarse de las colisiones.

Puesto que algunas estaciones están ocultas de otras, las LAN inalámbricas usan el *acceso múltiple por detección de portadora con evasión de colisiones* (CSMA/CA). Antes de la transmisión de un paquete de una computadora a otra, cada una de las dos computadoras envía un mensaje de control corto, el cual permite que todas las computadoras en su rango sepan que está a punto de ocurrir una transmisión.

EJERCICIOS

- 14.1** Explique las tres metodologías básicas que se usan para controlar el acceso a un medio compartido.
- 14.2** Mencione un ejemplo de una red que use la asignación dinámica de canales.
- 14.3** Enliste los tres tipos principales de canalización y las características de cada una.
- 14.4** Explique el sondeo y las dos políticas de sondeo generales.
- 14.5** En un sistema de reservación, ¿cómo determina un controlador la lista de las estaciones que transmitirán en una ronda específica?
- 14.6** ¿Qué es un token y cómo se usan para controlar el acceso a la red?

- 14.7** En el protocolo ALOHA, ¿qué ocurre si dos estaciones intentan transmitir al mismo tiempo en la frecuencia entrante y cómo se maneja el problema?
- 14.8** Desglose el acrónimo CSMA/CD y explique cada parte.
- 14.9** ¿Qué es el retroceso exponencial binario?
- 14.10** ¿Por qué CSMA/CD usa un retraso aleatorio? (Sugerencia: piense en muchas computadoras idénticas en una red).
- 14.11** ¿Por qué se necesita CSMA/CA en una red inalámbrica?

Contenido del capítulo

- 15.1 Introducción, 253
- 15.2 La venerable Ethernet, 253
- 15.3 Formato de tramas de Ethernet, 254
- 15.4 Campo tipo de la trama de Ethernet y demultiplexación, 254
- 15.5 Versión del IEEE de Ethernet (802.3), 255
- 15.6 Conexiones de LAN y tarjetas de interfaz de red, 256
- 15.7 Evolución de Ethernet y cableado de Thicknet, 256
- 15.8 Cableado de Thinnet, 257
- 15.9 Cableado de Ethernet de par trenzado y concentradores, 258
- 15.10 Topología física y lógica de Ethernet, 259
- 15.11 Cableado en un edificio de oficinas, 259
- 15.12 Velocidades de datos y tipos de cables de Ethernet, 261
- 15.13 Conectores y cables de par trenzado, 261
- 15.14 Resumen, 262

15

Tecnología alámbrica de LAN (Ethernet y 802.3)

15.1 Introducción

Esta parte del libro describe las tecnologías de redes de conmutación de paquetes. El capítulo 13 presenta el modelo IEEE 802 que se utiliza en las redes LAN y la división de la capa 2 en las subcapas de enlace lógico y MAC. Los capítulos también describen el esquema de direccionamiento de 48 bits que forma una parte considerable de la subcapa de enlace lógico. El capítulo 13 se concentra en la subcapa MAC y considera los protocolos para el acceso al medio.

Este capítulo continúa con la explicación de las redes de área local y se concentra en las tecnologías de LAN alámbricas. El capítulo muestra cómo es que los conceptos de los capítulos anteriores forman la base de Ethernet, la tecnología de LAN alámbrica que avanzó para dominar al resto.

15.2 La venerable Ethernet

En el capítulo 14 vimos que Ethernet es una tecnología de LAN que se inventó originalmente en Xerox PARC y después se estandarizó por parte de Digital Equipment Corporation, Intel y Xerox. Ethernet ha sobrevivido durante treinta años. Aunque los dispositivos de hardware, el cableado y los medios utilizados con Ethernet cambiaron de manera considerable, el formato básico de paquetes y el esquema de direccionamiento han seguido constantes. Uno de los aspectos más interesantes de la evolución de Ethernet es que las versiones más recientes de ésta siguen siendo compatibles con las versiones anteriores; una versión puede detectar una forma anterior y adaptarse de manera automática para dar cabida a la tecnología más antigua.

15.3 Formato de tramas de Ethernet

El término *trama* se refiere a un paquete de la capa 2 de enlace y el término *formato de trama* se refiere a la forma en que está organizado el paquete, incluyendo detalles como el tamaño y el significado de los campos individuales. La principal razón por la que versiones anteriores de Ethernet han permanecido compatibles con las versiones más nuevas se debe al formato de la trama, que ha permanecido constante desde que se creó el estándar DIX en la década de 1970. La figura 15.1 ilustra el formato de trama general de Ethernet y los detalles del encabezado de la trama.

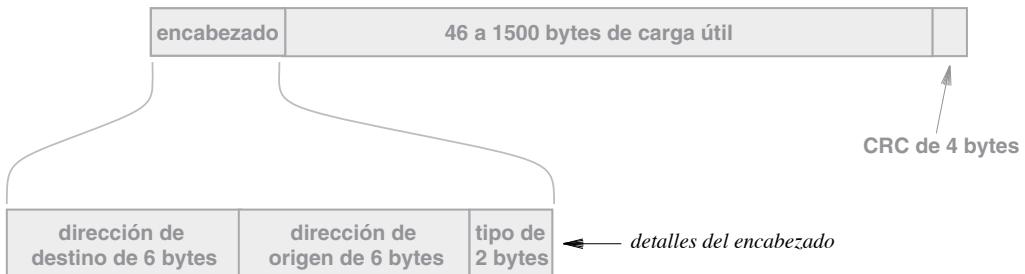


Figura 15.1 Ilustración del formato de trama de Ethernet y detalles del encabezado.

Como se muestra en la figura, una trama de Ethernet consiste en un encabezado de longitud fija, una carga útil de longitud variable y una comprobación por redundancia cíclica de longitud fija.[†] El encabezado contiene tres campos: un campo *dirección de destino* (de 48 bits) que proporciona la dirección del receptor deseado, un campo *dirección de origen* (de 48 bits) que contiene la dirección de la computadora que envió la trama y un campo *tipo* (de 16 bits).

15.4 Campo tipo de la trama de Ethernet y demultiplexación

El campo de tipo en una trama de Ethernet proporciona la multiplexación y la demultiplexación para que una computadora tenga varios protocolos operando al mismo tiempo. Por ejemplo, los capítulos posteriores explican que los protocolos utilizados en Internet envían datagramas IPv4 e IPv6 a través de Ethernet. A cada datagrama se le asigna un tipo de Ethernet único (0x0800 hexadecimal para los datagramas IPv4 y 0x08DD hexadecimal para los datagramas IPv6). De esta forma, al transmitir un datagrama IPv4 en una trama de Ethernet, el emisor asigna un tipo 0x0800. Cuando llega una trama a su destino, el receptor analiza el campo de tipo y usa el valor para determinar qué módulo de software debe procesar la trama. La figura 15.2 ilustra la demultiplexación.

[†] Cuando se envía una trama de Ethernet a través de una red, los bits se codifican usando la codificación Manchester que se describe en el capítulo 6, y antes de la trama puede ir un preámbulo de 64 bits de 0s y 1s alternantes.

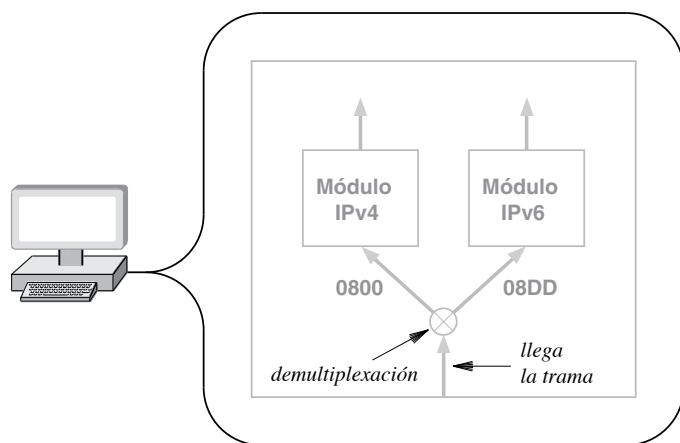


Figura 15.2 Ilustración del uso del campo de tipo para la demultiplexación.

15.5 Versión del IEEE de Ethernet (802.3)

Curiosamente, en 1983 el IEEE desarrolló un estándar para Ethernet e intentó redefinir el formato de trama de Ethernet.[†] El grupo de trabajo del IEEE que produjo el estándar se enumera como 802.3 y, para distinguir este estándar IEEE de otros, por lo general los profesionales lo conocen como *Ethernet 802.3*.

La mayor diferencia entre la Ethernet convencional y la Ethernet 802.3 surge de la interpretación del campo de tipo. El estándar 802.3 interpreta el campo de tipo original como una *longitud del paquete* y agrega un encabezado adicional de 8 bytes que contiene el tipo del paquete. El encabezado adicional se conoce como *control de enlace lógico/punto de conexión a la subred (LLC/SNAP)*; la mayoría de los profesionales lo llaman simplemente un *encabezado SNAP*. La figura 15.3 ilustra el formato.

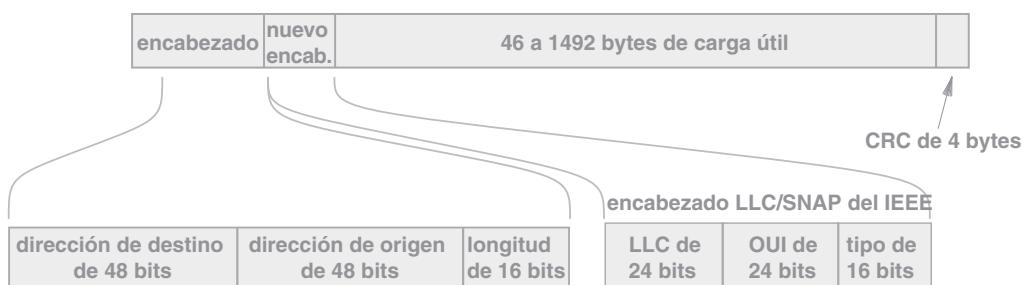


Figura 15.3 El formato de trama 802.3 del IEEE con un encabezado LLC/SNAP.

[†] La versión del IEEE no ha tenido mucho éxito; la mayoría de las instalaciones siguen usando el formato de trama original.

Como se muestra en la figura, el tamaño de trama total en la Ethernet 802.3 sigue siendo el mismo que en la Ethernet convencional: 1514 bytes. Por lo tanto, el IEEE redujo la capacidad de carga máxima de 1500 bytes a 1492 bytes. Podemos pensar en un encabezado SNAP que ocupa los primeros 8 bytes del área de carga útil. Para que ambas versiones de Ethernet sean compatibles, se usa una convención:

Si los bytes 13 y 14 de una trama de Ethernet contienen un valor numérico menor que 1500, el campo se interpreta como longitud de paquete y se aplica el estándar 802.3; de lo contrario el campo se interpreta como un campo de tipo y se aplica el estándar de Ethernet original.

15.6 Conexiones de LAN y tarjetas de interfaz de red

En términos de arquitectura computacional, una LAN parece ser un dispositivo de entrada o salida (E/S) y se conecta a la computadora de la misma forma que un dispositivo de audio o de video. Dentro de una computadora, un *controlador de interfaz de red (NIC)* se conecta a un bus de E/S. El controlador puede integrarse en la placa base o tomar la forma de una *tarjeta de interfaz de red* independiente que se conecta en un bus.

En el sentido lógico, un NIC maneja el reconocimiento de dirección, el cálculo del CRC y el reconocimiento de trama (por ejemplo, un NIC revisa la dirección de destino en una trama e ignora las tramas que no estén destinadas a esa computadora). Además, el NIC se conecta a una red y se encarga de los detalles de la comunicación de datos (es decir enviar y recibir tramas). En el sentido físico, un NIC consiste en un tablero de circuitos con un conector en un lado que coincide con el bus de la computadora y un conector del otro lado que acepta un enchufe apropiado para una LAN específica. La mayoría de las computadoras ya tienen un NIC instalado. Sin embargo, el NIC es independiente del resto de la computadora, por lo que un usuario puede optar por reemplazarlo sin tener que realizar otros cambios.

15.7 Evolución de Ethernet y cableado de Thicknet

Desde la versión original de la década de 1970, Ethernet ha pasado por varios cambios importantes, principalmente en el medio y en el cableado. El esquema de cableado original de Ethernet se conocía de manera informal como *Ethernet de alambre grueso* o *Thicknet*, ya que el medio de comunicación consistía en un cable coaxial pesado; el término formal para el cableado es *10Base5*. El hardware que se utilizaba con Thicknet se dividía en dos rutas principales. Un NIC se encargaba de los aspectos digitales de la comunicación y un dispositivo electrónico aparte conocido como *transceptor* se conectaba al cable de Ethernet y manejaba la detección de portadora, la conversión de bits en voltajes apropiados para su transmisión y la conversión de las señales entrantes a bits.

Un cable físico conocido como *interfaz de unidad de conexión (AUI)* conectaba un transceptor a un NIC en una computadora. Generalmente el transceptor estaba lejos de la computadora. Por ejemplo, en un edificio de oficinas, los transceptores de la Ethernet podían conectarse en el techo de un pasillo.

La figura 15.4 ilustra cómo el cableado Thicknet original usaba un cable AUI para conectar una computadora a un transceptor.

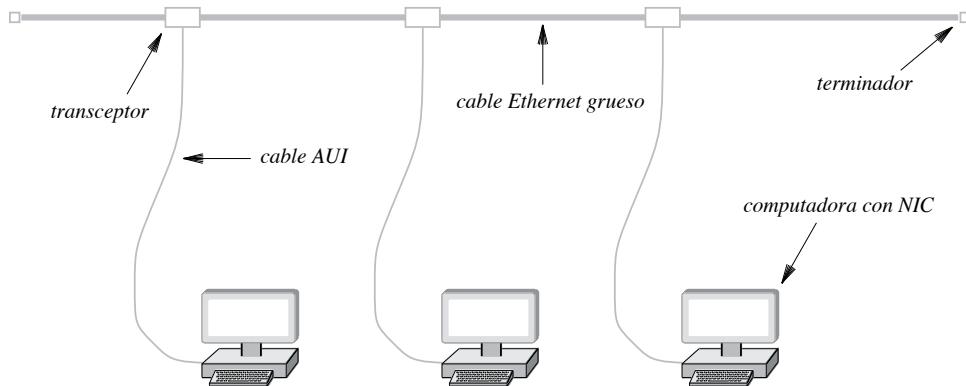


Figura 15.4 Ilustración del cableado Thicknet Ethernet original.

15.8 Cableado de Thinnet

Posteriormente se ideó una segunda generación de cableado de Ethernet que utilizaba un cable coaxial más delgado y flexible que Thicknet. Conocido de manera formal como *10base2* y de manera informal como *cable delgado de Ethernet* o *Thinnet*, el esquema de cableado difería de manera considerable de Thicknet. En vez de usar conexiones AUI entre una computadora y un transceptor, Thinnet integra un transceptor directamente en la tarjeta de interfaz de red y tiende un cable coaxial de una computadora a otra. La figura 15.5 ilustra el cableado de Thinnet.

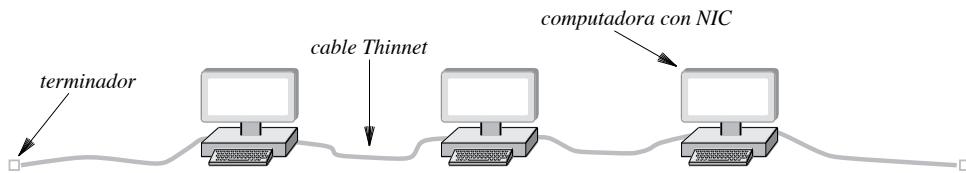


Figura 15.5 Ilustración del cableado de Ethernet de segunda generación conocido como Thinnet.

Thinnet tenía ventajas y desventajas. Las principales ventajas eran un menor costo total y facilidad de instalación. No se necesitaban transceptores externos y el cable Thinnet podía instalarse en una ruta conveniente (por ejemplo, a través de un tablero entre computadoras, bajo el piso o en una canaleta). La principal desventaja surgió debido a que toda la red era vulnerable: si un usuario desconectaba un segmento de la red para reubicar alambres o mover una computadora, la red completa dejaba de trabajar.

15.9 Cableado de Ethernet de par trenzado y concentradores

Una tercera generación de cableado de Ethernet hizo un cambio espectacular de dos formas:

- En lugar de cable coaxial, la tercera generación usa un dispositivo electrónico central separado de las computadoras conectadas a la red
- En vez de cable blindado pesado, la tercera generación usa cableado de par trenzado[†]

Puesto que no usa cable coaxial, la tecnología de tercera generación se conoce de manera informal como *Ethernet de par trenzado* y reemplaza a las otras versiones. Por lo tanto, Ethernet ya no es un cable sino un dispositivo electrónico al que se conectan las computadoras.

Para la versión original de Ethernet de par trenzado, el dispositivo electrónico que sirvió como interconexión central se conocía como *concentrador* o *hub*. Había concentradores disponibles en una variedad de tamaños, con el costo proporcional a su tamaño. Un concentrador pequeño tenía cuatro u ocho *puertos*, cada uno de ellos conectado a una computadora u otro dispositivo (por ejemplo, una impresora). Los concentradores más grandes podían contener cientos de conexiones. La figura 15.6 ilustra el esquema de cableado.

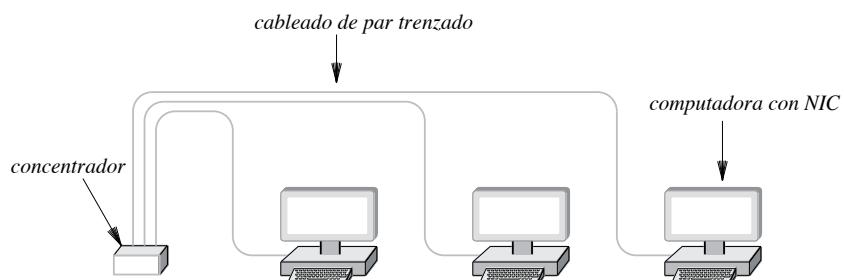


Figura 15.6 Ilustración de la Ethernet de tercera generación que usa cableado de par trenzado.

Los componentes electrónicos en un concentrador emulan un cable físico, haciendo que todo el sistema opere como una Ethernet convencional. Por ejemplo, una computadora conectada a un concentrador usa CSMA/CD para acceder a la red, recibe una copia de cada trama y usa la dirección de una trama para determinar si debe procesarla o ignorarla. Además, la tercera generación de Ethernet retuvo el mismo formato de trama que las versiones anteriores. De hecho, el software en una computadora no puede distinguir entre Ethernet, Ethernet Thinnet y Ethernet de par trenzado; la interfaz de red de una computadora se encarga de los detalles y oculta las diferencias. En conclusión:

[†] Las versiones más recientes de Ethernet que operan a una velocidad mayor a 1 gigabit por segundo requieren el uso de fibras ópticas en vez de cableado de cobre de par trenzado.

Aunque Ethernet de tercera generación usa un dispositivo electrónico en vez de un cable compartido, el formato de los paquetes que las computadoras envían y reciben ha permanecido sin cambio.

15.10 Topología física y lógica de Ethernet

Recuerde que las LAN se clasifican de acuerdo con su topología (es decir, su forma general). La figura 13.7 sintetiza las principales topologías.[†] Surge la duda: ¿qué es la topología de Ethernet? No debe sorprender que la respuesta sea compleja.

Es claro que la versión Thicknet original de Ethernet seguía una topología de bus y por ello la Ethernet original se cita a menudo como un ejemplo clásico de esta topología. Tal vez parezca que la Ethernet de tercera generación sigue una topología de estrella. De hecho, el término *concentrador* o *hub* surgió para aclarar el concepto de un punto de interconexión central. Sin embargo y como un concentrador emula un cable físico, el sistema parece funcionar como si las computadoras se conectaran a un mismo cable. De hecho, los profesionales bromeaban que un concentrador era en realidad un:

“bus en una caja”

Para entender la topología de Ethernet, debemos diferenciar entre las topologías *lógicas* y las *físicas*. Desde el punto de vista lógico, la Ethernet de tercera generación emplea una topología de bus. Sin embargo, desde el punto de vista físico, el cableado que se utiliza con la Ethernet de tercera generación constituye una topología en forma de estrella. En conclusión:

Al diferenciar entre las topologías lógicas y físicas podemos entender que la Ethernet de tercera generación usa una topología física de estrella, pero actúa lógicamente como un bus.

15.11 Cableado en un edificio de oficinas

Los estilos de cableado que se utilizan para las redes LAN presentan pocas diferencias en un cuarto de máquinas o en un laboratorio de cómputo. Sin embargo, cuando se usan en un edificio de oficinas, el tipo de cableado presenta una gran diferencia en términos del tipo y número de cables necesarios, la distancia que cubren y el costo. Las tres versiones de cableado de Ethernet ilustran las tres formas principales que usan las redes LAN. La figura 15.7 describe el cableado en el piso de un edificio de oficinas.

En la figura cabe observar que la Ethernet de tercera generación requiere que vayan muchos cables individuales entre oficinas y un punto central, lo cual se conoce como *armario de cables*. De tal forma que un armario de cables puede estar repleto de cientos de cables conectados a un dispositivo electrónico grande. Es imprescindible tener un etiquetado cuidadoso de los cables para evitar problemas.

[†] Encontrará la figura 13.7 en la página 227.

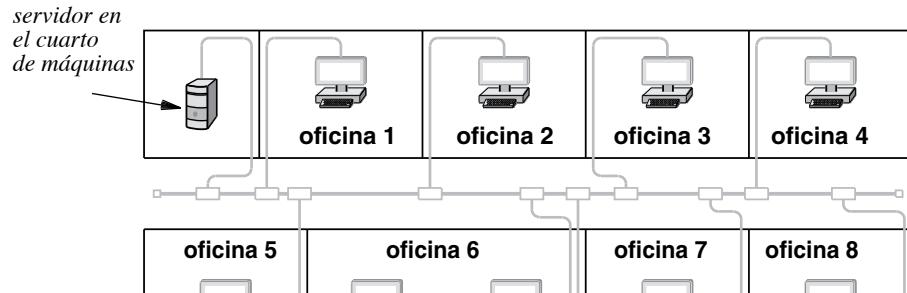


Figura 15.7 Ilustración de diversos esquemas de cableado de LAN utilizados en un edificio de oficinas.

15.12 Velocidades de datos y tipos de cables de Ethernet

Desde que la Ethernet de par trenzado surgió, se han realizado mejoras considerables en la calidad y el blindado disponibles en este tipo de cables. Como resultado, la velocidad de datos utilizada en la Ethernet de par trenzado se ha incrementado. La figura 15.8 sintetiza los tres tipos de Ethernet de par trenzado y el cable que se utiliza con cada uno.

Designación	Nombre	Velocidad de datos	Cable utilizado
10BaseT	Ethernet de par trenzado	10 Mbps	Categoría 5
100BaseT	Fast Ethernet	100 Mbps	Categoría 5E
1000BaseT	Gigabit Ethernet	1 Gbps	Categoría 6

Figura 15.8 Tres tipos de Ethernet de par trenzado, sus velocidades de datos y el cable que se utiliza con cada uno.

Como se indica en la figura, la primera versión de la Ethernet de par trenzado recibió la designación formal *10BaseT*, donde el valor 10 designa que la velocidad es de 10 Mbps. Después se introdujo una versión bajo el nombre *Fast Ethernet* que operaba a 100 Mbps y se le dio la designación formal *100BaseT*. Una tercera versión, conocida como *Gigabit Ethernet*, opera a 1 Gbps (es decir, 1000 Mbps). A menudo los profesionales abrevian el nombre como *Gig-E*.

El capítulo 17 explica que las tecnologías de Ethernet de mayor velocidad usan un dispositivo electrónico conocido como *comutador* en vez de un concentrador. Además, para seguir siendo compatible con versiones anteriores, los estándares de las versiones de mayor velocidad especifican que las interfaces deben detectar automáticamente la velocidad a la que puede operar una conexión, y disminuir su velocidad para dar cabida a otros dispositivos. De tal forma, si conectamos un cable de Ethernet entre un dispositivo anterior que use 10BaseT y un nuevo dispositivo que use 1000BaseT, el nuevo dispositivo *detectará automáticamente* la diferencia y reducirá su velocidad a 10 Mbps.

15.13 Conectores y cables de par trenzado

La Ethernet de par trenzado usa conectores *RJ45*, que son versiones más grandes de los conectores RJ11 utilizados para los teléfonos. Un conector RJ45 sólo puede conectarse de una forma en el enchufe, y hay una pieza física que mantiene el conector en su lugar. Por lo tanto, los conectores no pueden enchufarse de manera incorrecta y una vez insertados es difícil que se zafen.

Es posible comprar cables de varias longitudes con un conector RJ45 montado en cada extremo, lo que significa que la mayoría de los usuarios no tienen que armar el cable. Sin embargo, surge la confusión debido a que hay dos tipos de cables: *directos* y *cruzados*. Un cable cruzado, que se usa para

conectar dos commutadores, conecta una pata en un extremo a una pata diferente en el otro extremo. Un cable directo, que se usa entre una computadora y un conmutador, conecta cada pata del conector RJ45 que está en un extremo del cable directamente con la pata correspondiente del RJ45 en el otro extremo. Por lo tanto, la pata 1 se conecta a la pata 1 y así en lo sucesivo. Aunque el hardware de interfaz más sofisticado puede detectar un cable incorrecto y adaptarse, la mayor parte del hardware no funcionará correctamente si se usa un cable cruzado cuando se requiere un cable directo.

Para ayudar a los técnicos a realizar las conexiones correctas, cada uno de los alambres en un cable de categoría 5 o 6 está cubierto con plástico de colores. La figura 15.9 enlista los códigos de color utilizados con un cable directo.[†]

Pata del RJ45	Color de cable usado	Función
1	blanco-verde	TX_D1+
2	verde	TX_D1-
3	blanco-naranja	RX_D2+
4	azul	BI_D3+
5	blanco-azul	BI_D3-
6	naranja	RX_D2-
7	blanco-café	BI_D4+
8	café	BI_D4-

Figura 15.9 Lista de códigos de colores usados con un conector RJ45.

15.14 Resumen

La tecnología de Ethernet, que se inventó por primera vez en la década de 1970, se convirtió en el estándar para las redes de área local. Una trama de Ethernet comienza con un encabezado de 14 bytes que contiene una dirección de destino de 48 bits, una dirección de origen de 48 bits y un campo de tipo de 16 bits. Aunque el estándar 802.3 del IEEE intentó definir un nuevo formato de trama con un encabezado adicional de 8 bytes, esta versión raras veces se usa.

El campo de *tipo* de Ethernet se usa para la demultiplexación una vez que llega una trama a su destino. Al crear una trama, un emisor especifica el tipo; luego, un receptor usa el tipo para determinar qué módulo debe procesar la trama.

Aunque el formato y el direccionamiento de tramas de Ethernet han permanecido sin cambios desde el primer estándar, los cables utilizados para Ethernet y el esquema de direccionamiento han cambiado de manera considerable. Se han desarrollado tres versiones importantes de cableado de Ethernet. Thick-

[†] La abreviación en la figura especifica si se usa una pata para *transmitir* (TX), *recibir* (RX) o para comunicación *bidireccional* (BI), y para especificar una de las cuatro rutas de datos (D1 a D4) en la que se usa la pata.

net usaba un cable coaxial grande con transceptores independientes de las computadoras. Thicknet usaba un cable coaxial flexible que se tendía de una computadora a otra y la interfaz de red en cada computadora contenía un transceptor. La tercera generación sustituye el cable compartido con un dispositivo electrónico conocido como *concentrador (hub)* o *comutador (switch)* y usa el cableado de par trenzado o fibra óptica (para una mayor velocidad) entre una computadora y un concentrador. El sistema resultante tiene una topología física de estrella y una topología lógica de bus.

Al igual que las primeras versiones de Ethernet, la primera tecnología de par trenzado operaba a 10 Mbps y se designó como 10BaseT. Una versión con el nombre formal de 100BaseT opera a 100 Mbps y se conoce comercialmente como *Fast Ethernet*. Una tercera versión, llamada *Gigabit Ethernet* o *Gig-E*, opera a 1000 Mbps, lo que equivale a 1 Gbps. El hardware para Ethernet de alta velocidad detecta de manera automática cuando se conecta un dispositivo de baja velocidad y se adapta al dispositivo.

EJERCICIOS

- 15.1** ¿Qué tan grande es la trama máxima de Ethernet, incluyendo el CRC?
- 15.2** ¿Cómo se usa el campo de tipo en el encabezado de Ethernet?
- 15.3** En una trama de Ethernet 802.3, ¿cuál es el tamaño máximo de la carga útil?
- 15.4** ¿Cómo puede saber un receptor si una trama de Ethernet usa el estándar 802.3?
- 15.5** Cuando se usa el encabezado LLC/SNAP, ¿dónde se coloca?
- 15.6** ¿Cómo se conectaba una computadora a una Thicknet Ethernet?
- 15.7** ¿Cómo se conectaban las computadoras a una Thinnet Ethernet?
- 15.8** ¿Qué es un concentrador de Ethernet y qué cableado se usa con éste?
- 15.9** Busque en Web comutadores y concentradores (también aparecen como *switches* y *hubs*, respectivamente). Si le ofrecieran un comutador y concentrador que operara a la misma velocidad de bits por el mismo precio, ¿cuál de los dos seleccionaría? ¿Por qué?
- 15.10** Mencione un ejemplo de una red con diferentes topologías físicas y lógicas.
- 15.11** ¿Qué estilo de cableado de Ethernet requiere más alambres físicos en un edificio de oficinas?
- 15.12** ¿Qué categoría de cableado de par trenzado se necesita para una red de 10 Mbps? ¿De 100 Mbps? ¿De 1000 Mbps?

Contenido del capítulo

- 16.1 Introducción, 265
- 16.2 Una taxonomía de las redes inalámbricas, 265
- 16.3 Redes de área personal (PAN), 266
- 16.4 Bandas inalámbricas ISM utilizadas por redes LAN y PAN, 267
- 16.5 Tecnologías de redes LAN inalámbricas y Wi-Fi, 267
- 16.6 Tecnologías de espectro amplio, 268
- 16.7 Otros estándares de redes LAN inalámbricas, 269
- 16.8 Arquitectura de redes LAN inalámbricas, 270
- 16.9 Superposición, asociación y formato de trama 802.11, 271
- 16.10 Coordinación entre puntos de acceso, 272
- 16.11 Acceso con y sin contención, 272
- 16.12 Tecnología MAN inalámbrica y WiMax, 274
- 16.13 Tecnologías y estándares de redes PAN, 276
- 16.14 Otras tecnologías de comunicación de corta distancia, 277
- 16.15 Tecnologías de redes WAN inalámbricas, 278
- 16.16 Microceldas, 280
- 16.17 Grupos de celdas y reutilización de frecuencias, 280
- 16.18 Generaciones de tecnologías celulares, 282
- 16.19 Tecnología de satélite VSAT 284
- 16.20 Satélites GPS, 285
- 16.21 Radio definido por software y el futuro de la tecnología inalámbrica, 286
- 16.22 Resumen, 287

16

Tecnologías de redes inalámbricas

16.1 Introducción

Esta parte del libro se enfoca en las tecnologías de redes y sus aplicaciones en las redes de datos que usan conmutación de paquetes. Los capítulos 13 y 14 tratan sobre la conmutación de paquetes y el modelo del IEEE. El capítulo anterior explica las tecnologías inalámbricas que se utilizan en las redes de área local.

En este capítulo se describen las tecnologías inalámbricas. El capítulo explica que se han propuesto innumerables tecnologías inalámbricas, que la comunicación inalámbrica se usa en un amplio rango de distancias y que existen muchos sistemas comerciales. Por lo tanto, a diferencia de la situación de las redes alámbricas donde domina una sola tecnología, el trabajo en redes inalámbricas incluye varias tecnologías, muchas con características similares.

16.2 Una taxonomía de las redes inalámbricas

La comunicación inalámbrica abarca una amplia gama de tipos y tamaños de redes. Esto se debe en parte a las regulaciones gubernamentales que hacen que los rangos específicos del espectro electromagnético estén disponibles para la comunicación. Mientras que en ciertas partes del espectro se requiere de una licencia para operar el equipo de transmisión, en otras partes no se cuenta con ella. A la fecha, se han creado muchas tecnologías inalámbricas y continuamente aparecen nuevas variantes. Como se muestra en la taxonomía de la figura 16.1, las tecnologías inalámbricas pueden clasificarse ampliamente de acuerdo con el tipo de red.



Figura 16.1 Una taxonomía de las tecnologías de redes inalámbricas.

16.3 Redes de área personal (PAN)

Además de los tres tipos de redes principales descritos en el capítulo 13 (LAN, MAN y WAN), las redes inalámbricas incluyen las *redes de área personal (PAN)*. Una tecnología PAN proporciona comunicación a través de una distancia corta y está diseñada para usarse con dispositivos que pertenecen y son operados por un solo usuario. Por ejemplo, una PAN puede proveer comunicación entre un auricular inalámbrico y un teléfono celular. Las tecnologías PAN también se usan entre una computadora y un ratón o un teclado inalámbricos cercanos.

Las tecnologías PAN pueden agruparse en tres categorías amplias. La figura 16.2 enumera las categorías y proporciona una descripción breve de cada una de ellas. Las secciones posteriores de este capítulo explican con más detalle la comunicación en las redes PAN y enlistan los estándares de ésta.

Tipo	Propósito
Bluetooth	Comunicación a través de una distancia corta entre un dispositivo periférico pequeño, como un auricular o ratón, y un sistema como un teléfono celular o un computadora
Infrarroja	Comunicación dentro de la línea de visión entre un dispositivo pequeño (por lo general, un control remoto) y un sistema cercano, como una computadora o centro de entretenimiento
ZigBee	Comunicación a través de distancias aproximadamente equivalentes a una residencia, lo cual permite que los aparatos eléctricos se conecten a la red inteligente (Smart Grid).
Otras tecnologías ISM inalámbricas	Comunicación que utiliza frecuencias reservadas para dispositivos industriales, científicos y médicos, o entornos donde puede haber interferencia electromagnética

Figura 16.2 Cuatro tipos de tecnologías de redes de área personal.

16.4 Bandas inalámbricas ISM utilizadas por redes LAN y PAN

Los gobiernos reservaron tres áreas del espectro electromagnético para uso de los grupos *industriales, científicos y médicos*. Conocidas como *frecuencias ISM inalámbricas*, estas frecuencias no están licenciadas a portadoras comerciales, sino que están disponibles ampliamente para este tipo de propósitos y se utilizan en redes LAN y PAN. La figura 16.3 ilustra los rangos de frecuencias ISM.

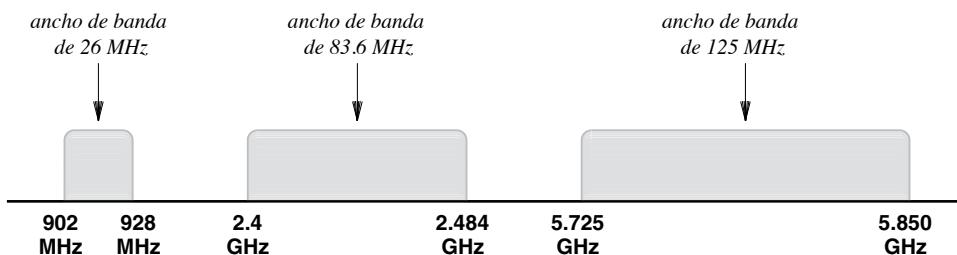


Figura 16.3 Bloques de frecuencias que constituyen las bandas ISM y el ancho de banda de cada una.

16.5 Tecnologías de redes LAN inalámbricas y Wi-Fi

Existe una variedad de tecnologías de LAN inalámbricas que usan diversas frecuencias, técnicas de modulación y velocidades de datos. El IEEE proporciona la mayoría de los estándares, categorizados como *IEEE 802.11*. En 1999, un grupo de distribuidores que fabrican equipo inalámbrico formaron la Alianza Wi-Fi, una organización sin fines de lucro que evalúa y certifica equipo inalámbrico utilizando los estándares 802.11. Como la alianza recibió una gran publicidad, la mayoría de los consumidores asocian las LAN inalámbricas con el término *Wi-Fi*.[†] La figura 16.4 enumera los estándares clave del IEEE que se clasifican bajo la Alianza Wi-Fi.

Estándar del IEEE	Banda de frecuencia	Velocidad de datos	Técnicas de modulación y multiplexación
802.11 original	2.4 GHz	1 o 2 Mbps	DSSS, FHSS
	2.4 GHz	1 o 2 Mbps	FHSS
	infrarrojo	1 o 2 Mbps	PPM
802.11b	2.4 GHz	5.5 a 11 Mbps	DSSS
802.11g	2.4 GHz	22 a 54 Mbps	OFDM, DSSS
802.11n	2.4 GHz	54 a 600 Mbps	OFDM

Figura 16.4 Estándares inalámbricos clave certificados por la Alianza Wi-Fi.

[†]Aunque la frase *fidelidad inalámbrica* aparecía originalmente en su publicidad, la alianza retiró la frase y no da ninguna explicación en cuanto al nombre.

16.6 Tecnologías de espectro amplio

El capítulo 11 introduce el término *espectro amplio* y explica que la transmisión del espectro amplio usa varias frecuencias para enviar datos. Es decir, el emisor esparce los datos a través de diversas frecuencias y el receptor combina la información que obtiene de esas frecuencias para reproducir los datos originales.

En general, el espectro amplio puede usarse para lograr una de las siguientes dos metas:

- Aumentar el rendimiento general
- Hacer que la transmisión sea más inmune al ruido

La tabla en la figura 16.5 sintetiza las tres técnicas de multiplexación clave que se usan en las redes inalámbricas Wi-Fi.

Nombre	Expansión	Descripción
DSSS	Espectro amplio de secuencia directa	Similar a CDMA, donde un emisor multiplica los datos salientes por una secuencia para formar varias frecuencias y el receptor multiplica por la misma secuencia para decodificar
FHSS	Espectro ensanchado por salto de frecuencias	Un emisor usa una secuencia de frecuencias para transmitir datos y un receptor usa la misma secuencia para extraerlos
OFDM	Multiplexación por división de frecuencias ortogonales	Un esquema de multiplexación por división de frecuencias donde la banda de transmisión se divide en muchas portadoras de tal forma que éstas no interfieran

Figura 16.5 Estándares inalámbricos clave certificados por la Alianza Wi-Fi.

Cada técnica tiene sus ventajas. OFDM ofrece la mayor flexibilidad. DSSS tiene un buen rendimiento y FHSS hace a una transmisión más inmune al ruido. Así, cuando se define una tecnología inalámbrica, los diseñadores seleccionan una técnica de multiplexación apropiada. Por ejemplo, se crearon dos versiones del estándar 802.11 original para adaptarlos a DSSS y FHSS. Para resumir:

Las técnicas del espectro amplio pueden ayudar a las redes LAN inalámbricas a funcionar en entornos donde hay ruido.

16.7 Otros estándares de redes LAN inalámbricas

El IEEE ha creado muchos estándares de redes inalámbricas que manejan varios tipos de comunicación. Cada estándar especifica el rango de frecuencias, la modulación y la multiplexación a usar, así como las velocidades de datos. La figura 16.6 enlista los principales estándares que se crearon o propusieron, con una breve descripción de cada uno.

Estándar	Propósito
802.11a	La primera variante de 802.11 que se creó para mejorar la velocidad; ya no es popular
802.11e	Calidad de servicio mejorada como una garantía de baja fluctuación
802.11h	Igual que 802.11a, pero agrega control de espectro y potencia (diseñado principalmente para usarse en Europa)
802.11i	Seguridad mejorada, incluyendo el estándar de cifrado avanzado (AES); la versión completa se conoce como WPA2
802.11k	Proveerá administración de recursos de radio, incluyendo la potencia de transmisión
802.11p	Comunicación dedicada de corto rango (DSRC) entre los vehículos en una carretera, y de un vehículo a un lado de la carretera.
802.11r	Habilidad mejorada de desplazarse entre puntos de acceso sin perder conectividad.
802.11s	Se propone para una red de malla donde un conjunto de nodos forman automáticamente una red y pasan paquetes

Figura 16.6 Principales estándares 802.11 y el propósito de cada uno.

En 2007, el IEEE combinó muchos de los estándares 802.11 existentes en un solo documento conocido como 802.11-2007. El IEEE usa el término “acumulado” para describir esta combinación de estándares. El documento combinado contiene un cuerpo principal que describe los requerimientos básicos que se aplican a todas las variaciones, y además tiene un apéndice para cada variante que proporciona los detalles de la misma.

En conclusión:

Se crearon o propusieron muchas variantes de 802.11; cada una ofrece ventajas para ciertas situaciones.

16.8 Arquitectura de redes LAN inalámbricas

Los tres bloques fundamentales de una LAN inalámbrica son: en primer lugar los *puntos de acceso*, que se conocen de manera informal como *estaciones base*; en segundo lugar un mecanismo de interconexión, como un comutador o enrutador utilizado para conectar los puntos de acceso, y en tercer lugar un conjunto de *hosts* inalámbricos, también conocidos como *nodos* inalámbricos o *estaciones* inalámbricas. En principio hay dos arquitecturas posibles para las redes LAN inalámbricas:

- Arquitectura *ad hoc*: los hosts inalámbricos se comunican entre sí, sin una estación base.
- Arquitectura de *infraestructura*: un host inalámbrico sólo se comunica con un punto de acceso y éste transmite todos los paquetes.

En la práctica existen pocas redes *ad hoc*. En su lugar, una organización o un proveedor de servicios implementa un conjunto de puntos de acceso, y cada host inalámbrico se comunica a través de uno de ellos. Por ejemplo, una compañía privada o una universidad podrían implementar puntos de acceso en todos sus edificios. La figura 16.7 ilustra esta arquitectura.

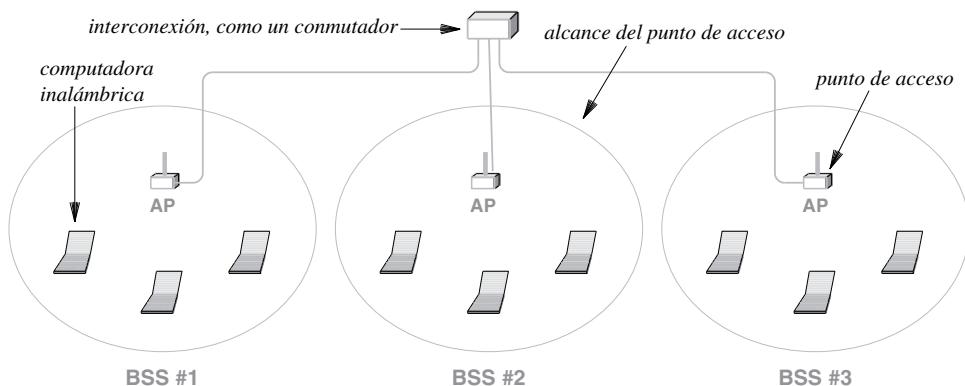


Figura 16.7 Ilustración de la infraestructura para una LAN inalámbrica.

Las conexiones cableadas que se extienden hasta los puntos de acceso, consisten por lo general en una Ethernet de par trenzado. El conjunto de computadoras dentro del alcance de un punto de acceso dado se conoce como *conjunto de servicio básico (BSS)*.[†] En la figura existen tres conjuntos de servicios básicos, uno para cada punto de acceso.

Para resumir:

La mayoría de las LAN inalámbricas usan una arquitectura del tipo infraestructura, en la que una computadora inalámbrica se comunica a través de un punto de acceso (estación base).

[†] De manera similar al sistema telefónico celular, la región que abarca un punto de acceso se conoce informalmente como *celda*.

16.9 Superposición, asociación y formato de trama 802.11

En la práctica, muchos detalles complican la arquitectura de infraestructura. Por una parte, si un par de puntos de acceso están demasiado separados, existirá una *zona muerta* entre ellos (es decir, una ubicación física sin conectividad inalámbrica). Por otra parte, si un par de puntos de acceso están demasiado cerca, existirá una superposición en la que un host inalámbrico podría alcanzar ambos puntos de acceso. Además, la mayoría de las LAN inalámbricas se conectan a Internet. Por ende, el mecanismo de interconexión generalmente incluye una conexión alámbrica adicional a un enrutador de Internet. La figura 16.8 ilustra esta arquitectura.

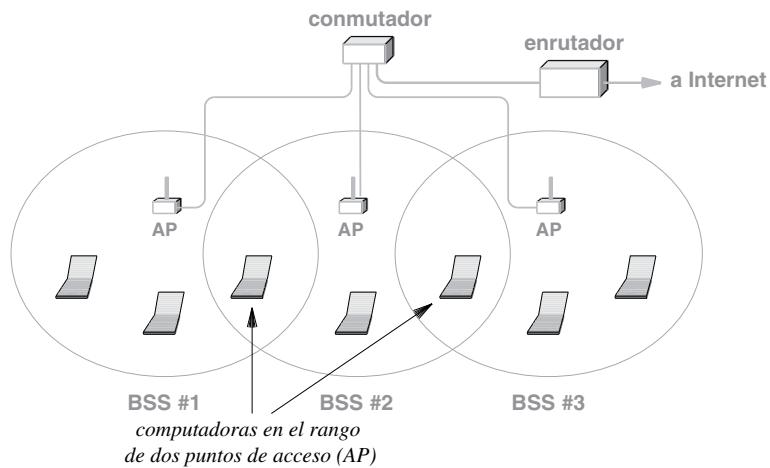


Figura 16.8 Ilustración de una arquitectura de infraestructura con regiones superpuestas.

Para lidiar con la superposición, las redes 802.11 requieren que un host inalámbrico se *asocie* con un solo punto de acceso. Es decir, el host inalámbrico envía tramas a un punto de acceso específico, el cual a su vez reenvía esas tramas a través de la red. La figura 16.9 ilustra el formato de tramas 802.11 y muestra que cuando se usa con una arquitectura de infraestructura, la trama transporta la dirección MAC de un punto de acceso, así como la dirección de un enrutador de Internet.

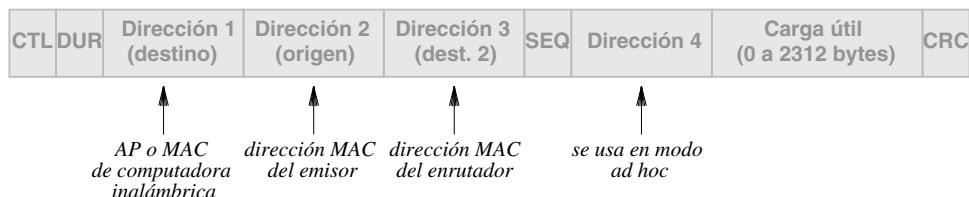


Figura 16.9 El formato de trama utilizado con una LAN inalámbrica de 802.11.

16.10 Coordinación entre puntos de acceso

Surge una pregunta interesante: ¿hasta qué grado necesitan coordinarse los puntos de acceso? Muchos de los primeros diseños de puntos de acceso eran complejos. Los puntos de acceso se coordinaban para brindar una movilidad constante similar a la del sistema telefónico celular. Es decir, los puntos de acceso se comunicaban entre sí para asegurar un traspaso uniforme a medida que una computadora inalámbrica se desplazaba de la región cubierta por un punto de acceso a la región de otro. Por ejemplo, ciertos diseños medían la fuerza de la señal e intentaban desplazar un nodo inalámbrico a un nuevo punto de acceso cuando la señal recibida en el nuevo punto de acceso excedía la fuerza de la señal en el punto de acceso anterior.

Como alternativa, algunos distribuidores comenzaron a ofrecer puntos de acceso menos complejos y de menor costo que no se coordinan. Los distribuidores argumentan que la fuerza de la señal no provee una medida válida de movilidad, que una computadora móvil puede encargarse de cambiar de un punto de acceso a otro y que la infraestructura alámbrica que conecta los puntos de acceso tiene suficiente capacidad para permitir una coordinación más centralizada. En situaciones en las que una instalación consiste en un solo punto de acceso, un diseño de punto de acceso menos complejo es especialmente apropiado.

Para resumir:

Existen dos metodologías básicas: los puntos de acceso complejos se coordinan para asegurar un traspaso uniforme, mientras que los puntos de acceso de menor costo operan de manera independiente y dejan a las computadoras inalámbricas la responsabilidad de cambiar su asociación de un punto de acceso a otro.

16.11 Acceso con y sin contención

El estándar 802.11 original definía dos metodologías generales para el acceso a los canales. Éstas pueden caracterizarse como:

- Función coordinada por puntos (PCF) para un servicio sin contención
- Función coordinada distribuida (DCF) para un servicio basado en la contención

El servicio coordinado por puntos significa que un punto de acceso controla las estaciones en el conjunto de servicio básico (BSS) para asegurar que las transmisiones no interfieran entre sí. Por ejemplo, un punto de acceso podría asignar a cada estación una frecuencia independiente. En la práctica, la PCF nunca se usa.

La función coordinada distribuida dispone que cada estación de un BSS debe ejecutar un protocolo de acceso aleatorio. En el capítulo 14 vimos que las redes inalámbricas pueden experimentar un *problema de estaciones ocultas*, donde dos estaciones pueden comunicarse pero una tercera estación puede recibir la señal sólo de una de ellas. Recuerde además que para resolver el problema, las redes 802.11 usan el *acceso múltiple por detección de portadora con evasión de colisiones* (CSMA/CA) que requiere un par para intercambiar mensajes de los tipos *listo para enviar* (RTS) y *libre para enviar* (CTS) antes de

transmitir un paquete. El estándar 802.11 incluye varios detalles que el capítulo 14 omite. Por ejemplo, el estándar define tres parámetros de sincronización como se muestra a continuación:

- SIFS: espacio corto entre tramas de 10μ segundos
- DIFS: espacio distribuido entre tramas de 50μ segundos
- Tiempo de intervalo de 20μ segundos

De manera intuitiva, el parámetro SIFS define cuánto tiempo debe esperar una estación receptora antes de enviar una respuesta ACK u otro tipo de respuesta; el parámetro DIFS, que es igual a SIFS más dos tiempos de intervalo, define cuánto tiempo debe estar un canal inactivo antes de que una estación pueda intentar transmitir. La figura 16.10 ilustra cómo se usan los parámetros en una transmisión de paquetes.

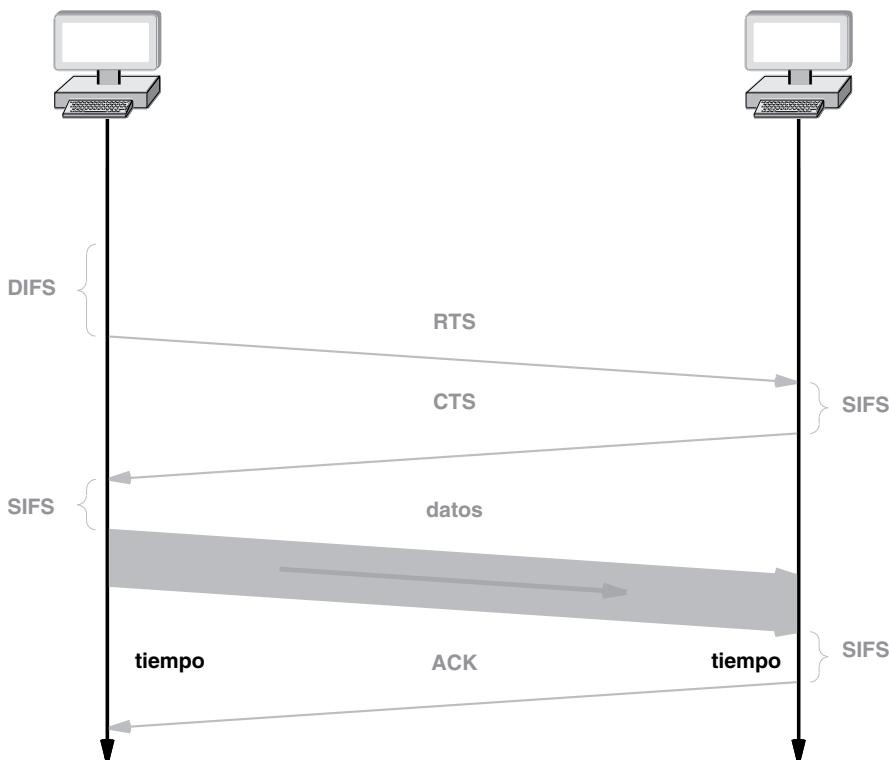


Figura 16.10 Ilustración de CSMA/CA con sincronización SIFS y DIFS.

En conclusión:

La técnica CSMA/CA que se usa en las redes Wi-Fi incluye parámetros de sincronización que especifican cuánto debe esperar una estación antes de enviar un paquete inicial y cuánto debe esperar antes de enviar una respuesta.

La separación física entre estaciones y el ruido eléctrico dificulta la acción de distinguir entre señales débiles, interferencia y colisiones. Por lo tanto, las redes Wi-Fi no emplean detección de colisiones. Es decir, el hardware no intenta detectar la interferencia durante una transmisión. En su lugar, un emisor espera un mensaje de *acuse de recibo (ACK)*. Si no llega un ACK, el emisor asume que se perdió la transmisión y emplea una estrategia de *retroceso* similar a la que se utiliza para la Ethernet alámbrica. En la práctica, las redes 802.11 que tienen pocos usuarios y no experimentan interferencia eléctrica raras veces necesitan la retransmisión. Sin embargo, otras redes 802.11 experimentan con frecuencia una pérdida de paquetes y dependen de la retransmisión.

16.12 Tecnología MAN inalámbrica y WiMax

En general, las tecnologías MAN no han tenido un éxito comercial, pero hay una tecnología MAN inalámbrica que sobresale debido a su potencial de éxito. El IEEE estandarizó esta tecnología bajo la categoría 802.16. Un grupo de compañías acuñaron el término *WiMAX*, que se interpreta como *interoperabilidad inalámbrica para el acceso a través de microondas*, y formaron el *Foro WiMAX* para promover el uso de la tecnología.

Se están desarrollando dos versiones principales de WiMAX, las cuales difieren en su metodología general. Las dos se conocen comúnmente como:

- WiMAX fija
- WiMAX móvil

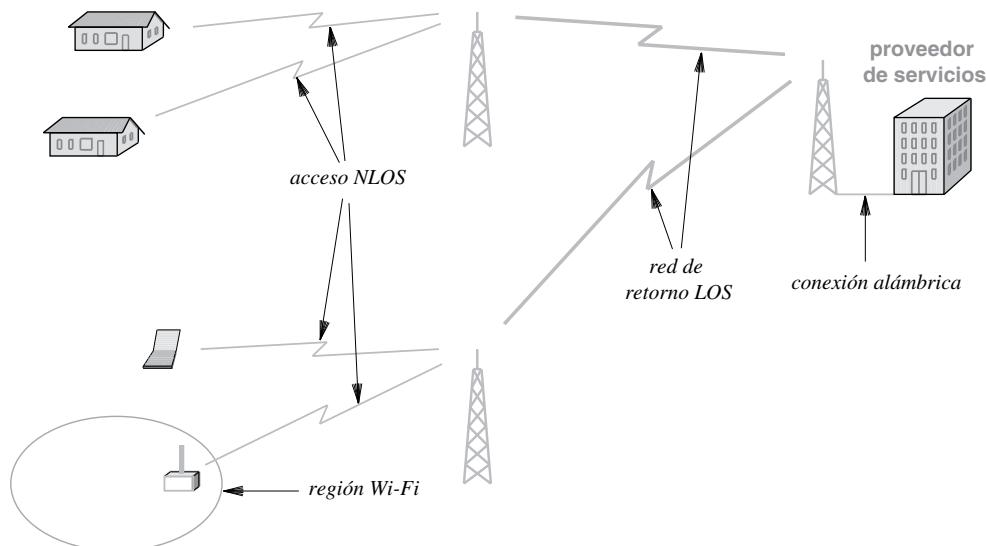
WiMAX fija se refiere a los sistemas creados con base en el estándar 802.16-2004 del IEEE, al que se le conoce de manera informal como 802.16d. El término *fija* surge debido a que la tecnología no ofrece un traspaso entre puntos de acceso. Por lo tanto, está diseñada para proveer conexiones entre un proveedor de servicios y una ubicación fija, como una residencia o edificio de oficinas, en vez de hacerlo entre un proveedor y un teléfono celular.

WiMAX móvil se refiere a los sistemas creados con base en el estándar 802.16e-2005, al que se le conoce de manera informal como 802.16e. Como el término *móvil* implica, la tecnología ofrece traspaso entre puntos de acceso, lo que significa que es posible usar un sistema WiMAX móvil con dispositivos portátiles, como computadoras laptop y teléfonos celulares.

WiMAX ofrece una comunicación de banda ancha que puede usarse de varias formas. Algunos proveedores de servicios planean usar WiMAX como una tecnología de acceso a Internet que abarca el último tramo. Otros ven el potencial de WiMAX para ofrecer una interconexión de propósito general entre sitios físicos, en especial en una ciudad. Hay otro tipo de interconexión que se conoce como *red de retorno o backhaul* y que se refiere a la conexión entre la ubicación de la red central de un proveedor de servicios y las instalaciones remotas, como las torres celulares. La figura 16.11 enumera algunos de los usos propuestos de WiMAX.

**Figura 16.11** Usos potenciales de la tecnología WiMAX.

En general, las implementaciones de WiMAX que se usen para la red de retorno tendrán las velocidades de datos más altas y usarán frecuencias que requieran de una *línea de visión (LOS)* entre dos entidades en comunicación. Por lo general, las estaciones LOS se montan en torres o en los techos de los edificios. Aunque las implementaciones utilizadas para el acceso a Internet pueden usar WiMAX fija o móvil, éstas usan comúnmente frecuencias que no requieren LOS. Por los tanto, se clasifican como aplicaciones *sin línea de vista (NLOS)*. La figura 16.12 ilustra las dos implementaciones.

**Figura 16.12** Ilustración del uso de WiMAX para acceso y red de retorno.

Las características clave de WiMAX pueden resumirse de la siguiente manera:

- Usa el espectro con licencia (es decir, el que ofrecen las portadoras)
- Cada celda puede cubrir un radio de 3 a 10 Km
- Usa FDM ortogonal escalable
- Garantiza la calidad de los servicios (para voz o video)
- Puede transportar 70 Mbps en cada dirección a distancias cortas
- Proporciona 10Mbps a través de una distancia larga (10 Km)

Para resumir:

WiMAX es una tecnología de LAN inalámbrica que puede usarse para una red de retorno, o para un acceso fijo o móvil; las implementaciones para el acceso no requieren de una línea de visión clara.

16.13 Tecnologías y estándares de redes PAN

El IEEE asignó el número 802.15 a los estándares de PAN. Para cada una de las tecnologías clave de las redes PAN se formaron varios grupos de trabajo y consorcios industriales. La figura 16.13 muestra una lista de los principales estándares de redes PAN.

Estándar	Propósito
802.15.1a	Tecnología Bluetooth (1 Mbps; 2.4 GHz)
802.15.2	Coexistencia entre redes PAN (sin interferencia)
802.15.3	PAN de alta velocidad (55 Mbps; 2.4 GHz)
802.15.3a	PAN de alta velocidad con banda ultra ancha (UWB) (110 Mbps; 2.4 GHz)
802.15.4	Tecnología ZigBee: PAN de baja velocidad de datos para control remoto
802.15.4a	PAN alternativa con velocidad baja de datos que usa poca energía

Figura 16.13 Estándares de PAN del IEEE.

Bluetooth. El estándar 802.15.1a del IEEE evolucionó luego de que los distribuidores crearan la tecnología Bluetooth como una tecnología de conexión inalámbrica de corta distancia. Las características de la tecnología Bluetooth son:

- Reemplazo inalámbrico para los cables (por ejemplo, audífonos o ratón)
- Usa la banda de frecuencia de 2.4 GHz
- Corta distancia (hasta 5 metros, con variaciones que extienden el rango a 10 o 50 metros)
- El dispositivo es *maestro* o *esclavo*
- El maestro otorga permiso al esclavo
- La velocidad de datos es de hasta 721 Kbps

Banda ultra ancha (UWB). La idea detrás de la comunicación mediante UWB es que para esparcir los datos a través de muchas frecuencias se requiere menos potencia para alcanzar la misma distancia. Las características clave de la UWB son:

- Usa un espectro amplio de frecuencias
- Consumo muy poca energía
- Corta distancia (2 a 10 metros)
- La señal penetra en obstáculos como las paredes
- Velocidad de datos de 110 Mbps a 10 metros y de hasta 500 Mbps a 2 metros
- El IEEE no puede resolver disputas y formar un solo estándar

ZigBee. El estándar ZigBee (802.15.4) surgió de un deseo de estandarizar la tecnología de controles remotos inalámbricos, en especial para el equipo industrial. Puesto que las unidades de control remoto sólo envían comandos cortos, no se requieren velocidades de datos altas. Las principales características de ZigBee son:

- Estándar inalámbrico para control remoto (velocidad de datos baja)
- El objetivo es la automatización tanto en la industria como en el hogar
- Se usan tres bandas de frecuencias (868 MHz, 915 MHz y 2.4 GHz)
- Velocidad de datos de 20, 40 o 250 Kbps, dependiendo de la banda de frecuencia
- Bajo consumo de energía
- Se están definiendo varios niveles de seguridad

16.14 Otras tecnologías de comunicación de corta distancia

Aunque por lo general no se agrupan con redes PAN inalámbricas, hay otras dos tecnologías inalámbricas que brindan comunicación a través de distancias cortas. Las tecnologías infrarrojas proporcionan comunicaciones de control y de datos de baja velocidad, y las tecnologías RFID se usan con sensores.

Infrarroja. La tecnología infrarroja se usa mucho en los controles remotos y puede usarse como reemplazo de un cable (por ejemplo, para un ratón inalámbrico). La Asociación de datos infrarrojos

(IrDA) produjo un conjunto de estándares ampliamente aceptados. Las principales características de la tecnología IrDA son:

- Familia de estándares para varias velocidades y propósitos
- Los sistemas en la práctica tienen un alcance de uno a varios metros
- Transmisión direccional en forma de un cono que cubre 30 grados
- Velocidades de datos entre 2.4 Kbps (control) y 16 Mbps (datos)
- Por lo general tienen un consumo bajo de energía, con versiones de muy poco consumo de energía
- La señal puede reflejarse de las superficies pero no puede penetrar objetos sólidos

Identificación por radiofrecuencia (RFID). La tecnología RFID usa una forma interesante de comunicación inalámbrica para crear un mecanismo en donde una *etiqueta* pequeña contiene información de identificación que un receptor puede *extraer*.

- Existen más de 140 estándares de RFID para una variedad de aplicaciones
- Las RFID pasivas consumen energía de la señal que envía el lector
- Las RFID activas contienen una batería, que puede durar hasta 10 años
- Distancia limitada, aunque las RFID activas se extienden más que las pasivas
- Pueden usar frecuencias desde menos de 100 MHz hasta 868-954 MHz
- Se usan para control de inventario, sensores, pasaportes y otras aplicaciones

16.15 Tecnologías de redes WAN inalámbricas

Las tecnologías de redes WAN inalámbricas pueden dividirse en dos categorías:

- Sistemas de comunicaciones celulares
- Sistemas de comunicaciones satelitales

En un principio, los sistemas de comunicaciones celulares se diseñaron para ofrecer servicios de voz a los clientes móviles. Por lo tanto, el sistema se diseñó para interconectar *celdas* o *células* a la red de telefonía pública. Los sistemas celulares se están usando cada vez más para proporcionar servicios de datos y conectividad a Internet.

En términos de arquitectura, cada celda contiene una torre, y un grupo de celdas (por lo general adyacentes) se conecta a un *centro de conmutación móvil*. El centro rastrea a un usuario móvil y administra el traspaso a medida que el usuario se mueve de una celda a otra. La figura 16.14 ilustra cómo podrían distribuirse las celdas a lo largo de una carretera.

Cuando un usuario se mueve entre dos celdas que se conectan al mismo centro de conmutación móvil, el centro se encarga del cambio. Cuando un usuario pasa de una región geográfica a otra, hay dos centros de conmutación móviles involucrados en el traspaso.

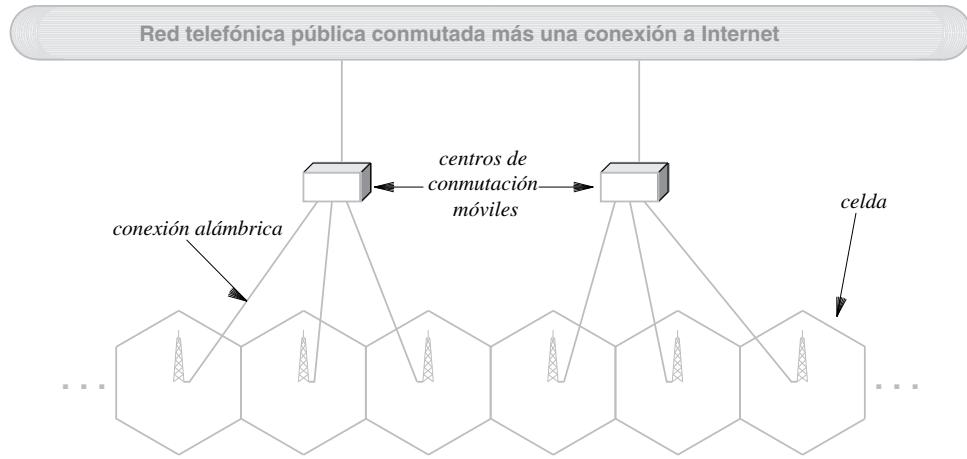


Figura 16.14 Ilustración de la arquitectura celular.

En teoría, una cobertura celular perfecta ocurre si cada celda forma un hexágono, ya que éstas pueden distribuirse en forma de panal. En la práctica, la cobertura celular es imperfecta. La mayoría de las torres de celdas usan antenas *omnidireccionales* que se transmiten en un patrón circular. Como resultado, en algunos casos las celdas se traslanan y en otros existen espacios sin cobertura. La figura 16.15 muestra la cobertura ideal y la real.

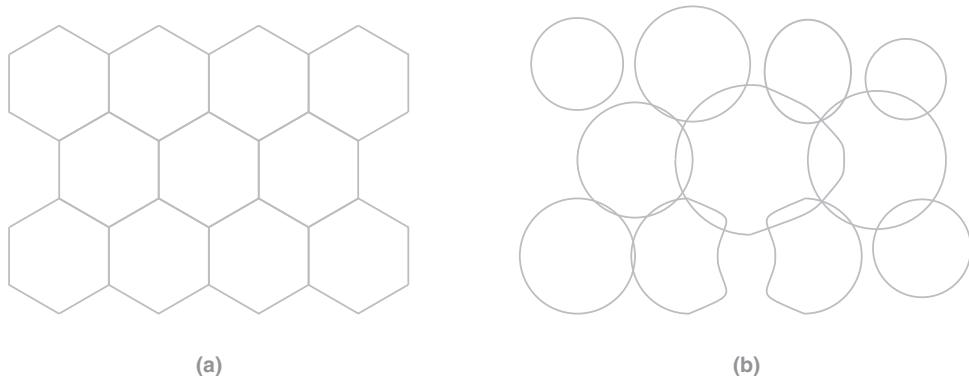


Figura 16.15 Ilustración de (a) una cobertura celular ideal y (b) una versión realista con superposiciones y espacios vacíos.

Otro aspecto práctico de la tecnología celular surge debido a la variabilidad en la densidad de las celdas. En áreas rurales donde la densidad esperada de teléfonos celulares es baja, las celdas son de gran tamaño; una sola torre es adecuada para un área geográfica grande. Sin embargo, en un entorno urbano, muchos teléfonos celulares se concentran en un área específica. Por ejemplo, considere una zona financiera de un área metropolitana grande. Además de los peatones y la gente que viaja en vehículos,

dicha área puede contener edificios de oficinas o de apartamentos con muchos ocupantes. Para manejar más teléfonos celulares, los diseñadores dividen una región en muchas celdas. Así, a diferencia de la estructura ideal de la figura 16.15(a) que tiene un solo tamaño de celda, una implementación práctica usa celdas de varios tamaños, donde las más pequeñas son usadas para cubrir áreas metropolitanas. En conclusión:

Aunque es fácil visualizar las celdas como un panal uniforme, los sistemas prácticos varían el tamaño de las mismas de acuerdo con la densidad de teléfonos celulares y obstrucciones que provocan que la cobertura sea irregular, lo cual produce superposiciones y espacios vacíos.

16.16 Microceldas

La variabilidad de las celdas es más aparente en áreas donde la densidad poblacional es especialmente alta (por ejemplo, un edificio de apartamentos de gran altura en una ciudad). En vez de una sola celda que abarque toda la cuadra de la ciudad, tal vez los proveedores celulares tengan que crear celdas que sólo den servicio a un subconjunto de los pisos de un edificio. Usamos el término *microcelda* para captar la idea. Para evitar interferencia, una microcelda usa mucho menor potencia que una celda normal.

Como caso especial, algunos proveedores ofrecen dispositivos de microceldas a clientes individuales. Aunque los distribuidores usan una variedad de términos de marketing como *pico celda* o *celda personal*, la idea básica es siempre la misma. Un cliente individual compra o renta un dispositivo de microcelda que da servicio a su residencia. En el sentido físico, la microcelda consiste en un dispositivo electrónico de unos cuantos centímetros de tamaño. En el sentido lógico, una microcelda se conecta a la red alámbrica de Internet y proporciona servicio de celda dentro de la residencia del propietario (algunos dispositivos están restringidos a un apartamento pequeño). El propietario configura el dispositivo de microcelda para reconocer un conjunto específico de teléfonos celulares. Cuando uno de los teléfonos reconocidos entra al rango de la microcelda, el teléfono se conecta y la usa para todos los servicios. La idea clave es que el propietario puede acceder a los servicios de datos sin pagar por minutos de celular.

16.17 Grupos de celdas y reutilización de frecuencias

La comunicación celular sigue un principio clave:

La interferencia puede minimizarse si un par adyacente de celdas no usan la misma frecuencia.

Para implementar el principio, los planificadores de redes celulares emplean una metodología de *grupo* o *clúster* de celdas, en la que se replica un patrón pequeño de celdas. A cada celda en un grupo se le asigna una frecuencia única. La figura 16.16 ilustra los grupos de tamaños 3, 4, 7 y 12 que se usan comúnmente.

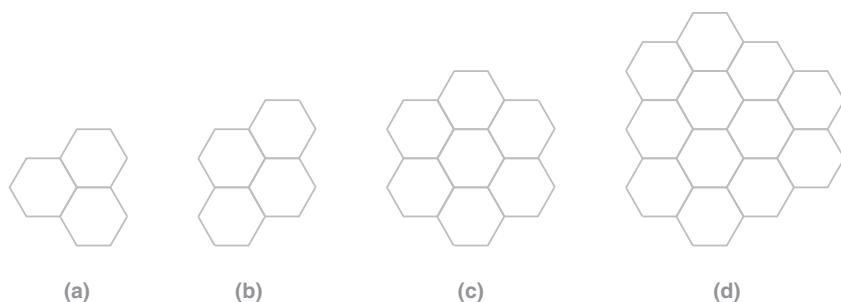


Figura 16.16 Ilustración de grupos de celdas comunes.

En términos geométricos, cada una de las formas en la figura puede usarse para cubrir un plano. Es decir, al replicar la misma forma es posible cubrir toda un área sin dejar espacios libres. Además, si a cada celda en una forma dada se le asigna una frecuencia única, el patrón repetido no asignará la misma frecuencia a ningún par de celdas adyacentes. Por ejemplo, la figura 16.17 ilustra una replicación del grupo de 7 celdas con una letra en cada celda para denotar la frecuencia asignada a la misma.

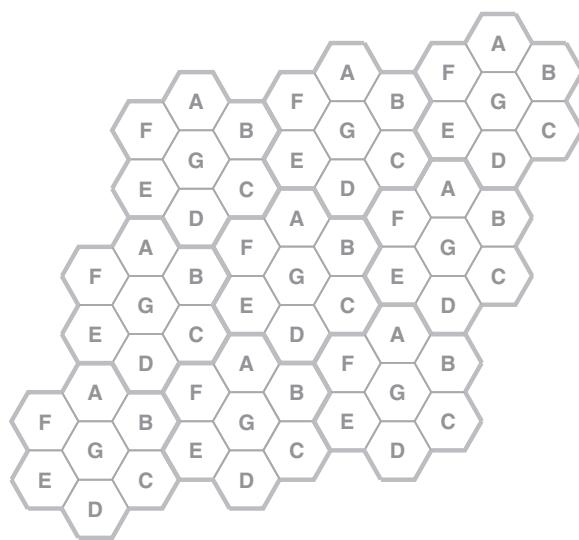


Figura 16.17 Ilustración de la asignación de frecuencias cuando se replica un grupo de 7 celdas.

En la figura, cada letra corresponde a una frecuencia específica y a cada celda dentro del grupo se le asigna una frecuencia diferente. Como indica la figura, cuando el patrón de grupo se replica, ninguna celda adyacente comparte una frecuencia común.

16.18 Generaciones de tecnologías celulares

La industria de las telecomunicaciones divide las tecnologías celulares en cuatro generaciones denominadas *1G*, *2G*, *3G* y *4G*. Las versiones intermedias se denominan *2.5G* y *3.5G*. Las generaciones pueden caracterizarse de las siguientes formas:

- *1G*. La primera generación comenzó a finales de la década de 1970 y se extendió hasta la década de 1980. Los sistemas, que en un principio se llamaban *radioteléfonos móviles celulares*, usaban señales analógicas para transmitir voz.
- *2G* y *2.5G*. La segunda generación comenzó a principios de la década de 1990 y sigue en uso. La principal distinción entre *1G* y *2G* surge debido a que *2G* utiliza señales digitales para transmitir voz. La etiqueta *2.5G* se usa para los sistemas que extienden un sistema *2G* para incluir algunas funciones de *3G*.
- *3G* y *3.5G*. La tercera generación comenzó en la década de 2000 y se enfoca en la incorporación de servicios de datos de mayor velocidad. Un sistema *3G* ofrece velocidades de descarga de 400 Kbps a 2 Mbps y está diseñado para soportar aplicaciones como navegación Web y compartición de fotos. *3G* permite que un solo teléfono deambule por toda Norteamérica, Japón y Europa.
- *4G*. La cuarta generación comenzó alrededor del año 2008 y se concentra en el soporte para multimedia de tiempo real, como programas de televisión o descarga de videos de alta velocidad. Además, los teléfonos *4G* incluyen varias tecnologías de conexión como Wi-Fi y satelital; en un momento específico, el teléfono puede seleccionar automáticamente la mejor tecnología de conexión disponible.

Se ha desarrollado una amplia variedad de tecnologías y estándares celulares. Cuando surgió *2G*, muchos grupos intentaron elegir por su cuenta un método y crear un estándar. La *Conferencia europea de administradores postales y de telecomunicaciones* seleccionó una tecnología de TDMA conocida como *sistema global de comunicaciones móviles (GSM)* y creó un sistema diseñado como estándar a nivel mundial. En Estados Unidos, cada portadora creó una red con su propia tecnología. Motorola inventó un sistema TDMA conocido como *iDEN*. La mayoría de las portadoras estadounidenses y asiáticas adoptaron una metodología de CDMA que se estandarizó como *IS-95A*. Japón creó una tecnología TDMA conocida como *PDC*. Con el tiempo se desarrollaron tecnologías digitales que usan técnicas de modulación y multiplexación más sofisticadas para incrementar las velocidades de datos. Una tecnología, conocida como *velocidades de datos mejoradas para la evolución de GSM (EDGE)* o *GPRS mejorado (EGPRS)*, ofrece una velocidad de transferencia de hasta 473.6 Kbps; un sucesor conocido como *evolución de EDGE* ofrece una velocidad de datos máxima de 1 Mbps.

Para cuando los proveedores comenzaron a pensar en las tecnologías de tercera generación, era aparente que los clientes querían un servicio de telefonía celular que funcionara a nivel mundial. Como resultado, los proveedores presionaron para que las tecnologías fueran interoperables y la industria consolidó muchas de las metodologías de *2G* en unos cuantos estándares clave. *IS-136*, *PDC*, *IS-95A*

y EDGE influyeron en el diseño de *UMTS*, una tecnología que usa *CDMA de banda ancha (WCDMA)*. Mientras tanto, IS-95B se extendió para producir *CDMA 2000*.

Varios estándares competidores evolucionaron para los servicios de datos de tercera generación. *EVDO* y *EVDV* surgieron aproximadamente al mismo tiempo. Cada uno de los dos combina las técnicas de CDMA y de multiplexación por división de frecuencias para incrementar el rendimiento en general. EVDO, que podría leerse como *evolución con datos optimizados* o como *evolución de sólo datos*, es el que se implementa en forma más extensa. EVDO cuenta con dos versiones que difieren en la velocidad a la que se transmiten los datos: 2.4 Mbps o 3.1 Mbps. Una alternativa llamada *acceso a paquetes de alta velocidad por enlace descendente (HSDPA)* ofrece velocidades de descarga de 14 Mbps.[†] Desde luego que las portadoras cobran más por los servicios que ofrecen una velocidad de datos más alta. La figura 16.18 sintetiza los principales estándares celulares 2G y 3G.

Método	Estándar
GSM	GSM, GPRS, EDGE, (EGPRS), EDGE Evolution, HSCSD
CDMA	IS-95A, IS-95B
TDMA	IDEN, IS-136, POC
WCDMA	UMTS, HSDPA
CDMA	1xRTT, EVDO, EVDV

Figura 16.18 Principales tecnologías celulares de segunda y tercera generación.

Para cuando se estaban diseñando las tecnologías celulares de cuarta generación habían aparecido los teléfonos inteligentes, y estaba claro que los datos dominarían el uso del teléfono celular. Además de descargar datos y ver transmisiones de video de flujo continuo, los usuarios comenzaron a enviar archivos, imágenes y videos. Para adaptarse a los incrementos esperados en el uso de datos, la UIT publicó una especificación para los sistemas celulares 4G conocida como *telecomunicaciones móviles internacionales avanzadas (IMT-Advanced)*. IMT-Advanced especifica velocidades de datos de 100 Mbps al moverse rápidamente (por ejemplo, en un tren o auto) y de 1 Gbps al moverse lentamente (por ejemplo, un peatón caminando).

Los distribuidores trabajaron para crear e implementar estándares para 4G. Cuatro de los primeros estándares fueron *HSPA+*, *HTC Evo 4G*, *WiMAX* (descrito anteriormente) y una tecnología conocida como *evolución a largo plazo (LTE)*. Ninguno de los primeros estándares cumplieron los criterios especificados por IMT-Advanced (por ejemplo, LTE sólo puede transmitir 300 Mbps en el enlace descendente y 75 Mbps en el enlace ascendente). Sin embargo, la UIT decidió que a los distribuidores se les permitiría anunciar los sistemas como 4G. Por lo tanto, los distribuidores comercializaron los sistemas LTE bajo el nombre *4G LTE*. Mientras tanto, los distribuidores desarrollaron dos estándares que la UIT clasifica como “4G real”: *LTE Advanced* y *WiMAX Advanced*. La figura 16.19 sintetiza los estándares 4G.

[†] También se definió un protocolo de *acceso a paquetes de alta velocidad por enlace descendente (HSUPA)*, pero recibió menos interés que HSDPA.

Clasificación	Estándar
Puede anunciarse como 4G	HSPA+, HTC Evo 4G, LTE, WiMAX
Se adhiere a IMT-Advanced	LTE Advanced, WiMAX Advanced

Figura 16.19 Estándares inalámbricos celulares de cuarta generación.

La principal diferencia entre los estándares 3G y 4G surge de la tecnología que está detrás. Todos los sistemas 3G se diseñaron usando estándares telefónicos de voz heredados del viejo sistema telefónico analógico. Los datos se agregaron como una característica adicional de una llamada de voz, en vez de agregarse como una parte principal del diseño. Los sistemas 4G están diseñados para usar el protocolo de Internet (que describiremos en la parte IV del libro) como la base de toda comunicación.[†] De esta forma, un sistema 4G usa la conmutación de paquetes y la transmisión de voz es una aplicación específica. En conclusión:

Como usan conmutación de paquetes, los sistemas celulares 4G facilitan el hecho de que un teléfono celular se comunique con cualquier sitio en Internet.

16.19 Tecnología de satélite VSAT

El capítulo 7 describe los tres tipos de satélites de comunicación (LEO, MEO y GEO) y el capítulo 14 describe los mecanismos de acceso a los canales, incluyendo los mecanismos de reservación que se usan para proveer TDMA a través de un satélite. Esta sección concluye la explicación de los satélites mediante una descripción de las tecnologías satelitales específicas.

La clave para la comunicación satelital es un diseño de antena parabólica que se conoce de manera informal como *plato*. La forma parabólica significa que la energía electromagnética que llega de un satélite distante se refleja hacia un solo punto de enfoque. Al orientar el plato hacia un satélite y colocar un receptor en el punto de enfoque, un diseñador puede garantizar que se reciba una señal fuerte. La figura 16.20 ilustra el diseño y muestra cómo se refleja la energía entrante de la superficie del plato hacia el receptor.

Para maximizar la señal recibida, las primeras comunicaciones satelitales usaban estaciones terrestres con grandes antenas de plato de más de tres metros de diámetro. Aunque son apropiadas para situaciones como el enlace transatlántico que utiliza una compañía telefónica, los consumidores y los negocios pequeños no pueden colocar dichas estaciones terrestres en su propiedad. Por lo tanto, ocurrió un cambio importante con el surgimiento de una tecnología conocida como *terminal de apertura muy pequeña* (VSAT) que usa platos de menos de tres metros de diámetro. Una antena VSAT común mide menos de un metro de diámetro.

Muchas empresas usan tecnología VSAT para enlazar todas sus tiendas. Por ejemplo, las farmacias como Walgreens y CVS, muy comunes en Estados Unidos, emplean comunicación VSAT, al igual que las cadenas de comida rápida como Pizza Hut y Taco Bell, y los almacenes minoristas como Walmart. Además, los servicios VSAT están disponibles para los consumidores que acceden tanto a entretenimiento como a Internet.

[†]En la práctica, muchos proveedores celulares siguen enviando las llamadas de voz a través de sistemas 3G y usan los protocolos 4G sólo para datos.

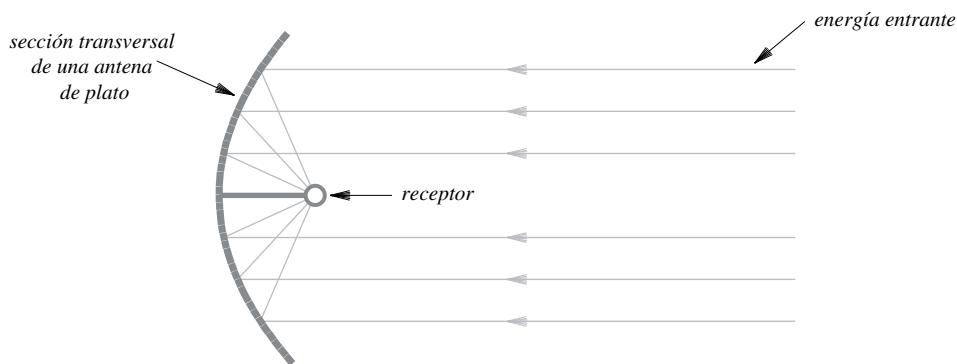


Figura 16.20 Ilustración de una antena de plato parabólica que refleja las ondas electromagnéticas hacia un punto de enfoque.

Los satélites VSAT usan tres rangos de frecuencia que difieren en cuanto a la fuerza de la señal transmitida, la sensibilidad a la lluvia y otras condiciones atmosféricas, y el área de la superficie terrestre cubierta (lo que se conoce como *huella* del satélite). La figura 16.21 describe las características de cada banda de frecuencia.

Banda	Frecuencia	Huella	Fuerza de la señal	Efecto de la lluvia
Banda C	3 a 7 GHz	Grande	Baja	Medio
Ku	10 a 18 GHz	Medianas	Media	Moderado
Ka	18 a 31 GHz	Pequeñas	Alta	Severo

Figura 16.21 Bandas de frecuencia utilizadas por la tecnología VSAT y las características de cada una.

16.20 Satélites GPS

Los satélites del *sistema de posicionamiento global (GPS)* brindan información precisa sobre tiempo y ubicación. Aunque no es parte de la comunicación de las computadoras, la información relacionada con la ubicación se usa cada vez más en las redes móviles. Las características clave son:

- Precisión entre 20 y 2 metros (las versiones militares tienen una mayor precisión)
- Hay en total 24 satélites orbitando la Tierra
- Los satélites están distribuidos en seis planos orbitales
- Proporciona sincronización de tiempo usada en algunas redes de comunicaciones

En un sentido, la técnica que se utiliza para obtener la información sobre la posición es simple: como todos los satélites GPS orbitan en posiciones bien conocidas, un receptor puede determinar una ubicación única en la superficie de la Tierra si encuentra la distancia que tiene en relación con tres satélites. Para entender por qué, considere los puntos que están a la distancia D_1 del satélite 1; el conjunto define una esfera. De manera similar, el conjunto de puntos que están a la distancia D_2 del satélite 2 define otra esfera. Un sistema GPS que esté al mismo tiempo a la distancia D_1 del satélite 1 y a D_2 del satélite 2, se encuentra en el círculo formado por la intersección de las dos esferas. Si el sistema GPS también está a la distancia D_3 del satélite 3, el sistema GPS estará en la intersección de una tercera esfera con el círculo, lo cual produce dos puntos posibles. Los satélites se acomodan de modo que sólo uno de los puntos se encuentre en la superficie de la Tierra y el otro esté en el espacio, con lo cual se facilita la acción de elegir el punto exacto.

Para calcular la distancia, un sistema GPS aplica la forma de la física newtoniana que especifica que la distancia equivale a la velocidad multiplicada por el tiempo. La velocidad es constante (la velocidad de la luz, 3×10^8 metros por segundo). El tiempo se calcula haciendo que cada sistema GPS calcule la hora local y cada satélite debe tener un reloj preciso que incluya una *etiqueta de hora y fecha* en la información que envía. Así, un receptor puede extraer la etiqueta de fecha a la hora local para determinar el tiempo que ha estado la información en tránsito.

16.21 Radio definido por software y el futuro de la tecnología inalámbrica

Cada una de la amplia variedad de tecnologías inalámbricas descritas en el capítulo utiliza hardware de radio de propósito específico. La antena, el transmisor y el receptor en un dispositivo dado están diseñados para operar en frecuencias predeterminadas, usando formas específicas de modulación y multiplexación. Un teléfono celular que pueda usar redes GSM, Wi-Fi y CDMA debe tener tres sistemas de radio totalmente independientes, y debe elegir entre ellos.

Los radios tradicionales se están sustituyendo por radios que sigan un paradigma *programable* en el que las funciones se controlen mediante un software que se ejecute en un procesador. La figura 16.22 enumera las principales funciones de radio que pueden controlarse en un *radio definido por software (SDR)*.

Las tecnologías clave que permiten los radios definidos por software son los filtros analógicos sintonizables y la administración de antenas múltiples. En la actualidad hay chips disponibles que brindan filtros analógicos sintonizables. De esta forma, es posible seleccionar las frecuencias y controlar la potencia. Hay también *procesadores de señales digitales (DSP)* para manejar la codificación y la modulación de las señales. El aspecto más interesante de los radios definidos por software se refiere al uso de varias antenas. En vez de elegir una sola antena para usarla en un momento dado, un radio definido por software puede usar varias antenas al mismo tiempo para proporcionar *multiplexación espacial*, una técnica que permite transmitir o recibir una señal de una dirección dada. Usamos el término *múltiple entrada, múltiple salida (MIMO)* para denotar un sistema que emplea varias antenas para transmisión y recepción (es decir, que puede dirigir la transmisión o la recepción).

Función	Descripción
Frecuencia	El conjunto exacto de frecuencias utilizadas en un momento dado
Potencia	La cantidad de potencia que emite el transmisor
Modulación	La codificación y modulación de la señal y del canal
Multiplexación	Cualquier combinación de CDMA, TDMA, FDMA y otras
Dirección de señal	Las antenas pueden sintonizarse para una dirección específica
Protocolo MAC	Todos los aspectos del entramado y del direccionamiento MAC

Figura 16.22 Funciones bajo el control del software en un radio definido por software.

El ejército estadounidense ya está implementando radios definidos por software, y hay kits disponibles para los experimentadores. Uno de los mayores obstáculos para el despliegue generalizado se relaciona con la necesidad de políticas que regulen el uso del espectro. Tradicionalmente, los dispositivos que transmiten energía electromagnética están certificados para asegurar que se encuentren dentro de ciertos límites de potencia especificados y no interfieran con otras comunicaciones (por ejemplo, un teléfono celular no interfiere con las comunicaciones de la policía o de emergencia). Si un usuario puede controlar un radio definido por software, sería posible que el consumidor configure el radio para operar en frecuencias que ya están otorgadas a estaciones de radio o a operadores de telecomunicaciones. Lo que es más importante, un usuario podría descargar inadvertidamente un virus que use el radio para interferir los canales de comunicación existentes (incluyendo los canales que utilizan los servicios de emergencia). Por lo tanto, se están investigando técnicas que controlen la potencia que un radio de software pueda usar y el conjunto de frecuencias sobre las que puede operar.

16.22 Resumen

Existen muchas tecnologías de comunicación inalámbricas y se usan para crear redes LAN, PAN, MAN y WAN inalámbricas. El IEEE estandarizó varias tecnologías de redes LAN y MAN. Wi-Fi usa los estándares 802.11 del IEEE, con variantes a las que se les asigna un sufijo, como 802.11b o 802.11g. Las redes LAN inalámbricas pueden ser *ad hoc* o usar una arquitectura de infraestructura con puntos de acceso; el formato de trama incluye una dirección MAC para un punto de acceso, así como una dirección MAC para un enrutador más allá del punto de acceso.

Además de las redes LAN, las tecnologías inalámbricas se usan para redes MAN y PAN. La principal tecnología MAN se conoce como WiMAX, que puede usarse para redes de retorno o para acceder a datos. Existe una variedad de tecnologías de redes PAN, incluyendo Bluetooth, banda ultra ancha (UWB), ZigBee e IrDA. Las etiquetas RFID proporcionan otra forma de comunicación inalámbrica que se utiliza principalmente para inventario y embarques.

Las redes WAN inalámbricas usan tecnologías celulares y satelitales. Las tecnologías celulares evolucionaron a través de cuatro generaciones, desde un sistema analógico diseñado para transportar voz hasta una red de computadoras basada en paquetes (conocida como 4G inalámbrica) que se enfoca en los datos. Existen muchas tecnologías y estándares. Las tecnologías de satélite VSAT hacen posible que los negocios y consumidores tengan antenas de plato en su propiedad.

Los sistemas inalámbricos emergentes usan radios definidos por software que permiten al software controlar todos los aspectos de la transmisión de radio. Los radios definidos por software están disponibles para usos militares y especiales; hay kits disponibles para experimentación.

EJERCICIOS

- 16.1** Nombre las tres tecnologías de redes PAN inalámbricas e incluya una descripción breve de cada una.
- 16.2** ¿Cuáles son los tres bloques de frecuencias utilizados por las redes LAN y PAN?
- 16.3** ¿Qué es la Alianza Wi-Fi?
- 16.4** Mencione el prefijo numérico que usan los estándares del IEEE para las redes Wi-Fi.
- 16.5** Enliste tres técnicas de espectro amplio y proporcione una descripción general de cada una.
- 16.6** Busque OFDM en Web y proporcione una descripción de un párrafo.
- 16.7** Enliste los estándares del IEEE que se propusieron o crearon para redes LAN inalámbricas.
- 16.8** ¿Por qué la mayoría de las redes LAN inalámbricas usan una metodología de infraestructura en vez de una *ad hoc*?
- 16.9** ¿Por qué una computadora inalámbrica debe asociarse con una estación base específica?
- 16.10** Un encabezado 802.11 contiene dos direcciones de destino. Explique el propósito de cada una.
- 16.11** ¿Qué son SIFS y DIFS, y por qué se necesitan?
- 16.12** Nombre los dos tipos de tecnologías WiMAX y describa el propósito de cada una.
- 16.13** ¿Qué es ZigBee y dónde se usa?
- 16.14** Proporcione las características de la tecnología UWB.
- 16.15** ¿Tiene sentido usar IrDA para aplicaciones como la transferencia de archivos? ¿Por qué sí o por qué no?
- 16.16** ¿Qué es RFID y dónde se usa?
- 16.17** ¿A qué se conecta una torre de celdas?
- 16.18** ¿Qué es un grupo de celdas y cómo usa un diseñador los grupos?
- 16.19** Nombre las cuatro generaciones de tecnología celular y describa cada una de ellas.
- 16.20** ¿Qué es GSM y qué estándares incluye?
- 16.21** ¿Cuáles son las tecnologías celulares de tercera generación que usan la multiplexación por división de código?
- 16.22** ¿Qué es un satélite VSAT?
- 16.23** ¿Por qué un plato de satélite tiene la forma de una parábola?
- 16.24** Nombre las tres principales bandas de frecuencia que usan los satélites de comunicación y describa el efecto del clima en cada una.
- 16.25** ¿Cuántos satélites se usan en GPS y qué tan preciso es este sistema?
- 16.26** Además de la posición, ¿qué proporciona el sistema GPS?
- 16.27** ¿Qué funciones son controlables en un radio definido por software?

Contenido del capítulo

- 17.1 Introducción, 291
- 17.2 Limitación de distancia y diseño de redes LAN, 291
- 17.3 Extensiones de módem de fibra óptica, 292
- 17.4 Repetidores, 293
- 17.5 Puentes y uso de puentes, 293
- 17.6 Puentes con capacidad de aprendizaje y filtrado de tramas, 294
- 17.7 Por qué es bueno usar puentes, 295
- 17.8 Árbol de expansión distribuido, 296
- 17.9 Comutación y comutadores de la capa 2, 297
- 17.10 Comutadores de redes VLAN, 299
- 17.11 Múltiples comutadores y redes VLAN compartidas, 300
- 17.12 La importancia de los puentes, 301
- 17.13 Resumen, 302

17

Repetidores, puentes y conmutadores

17.1 Introducción

Los capítulos anteriores describen las topologías de las redes LAN y los esquemas de cableado. Una LAN común está diseñada para abarcar algunos cientos de metros, lo que significa que la tecnología de LAN funciona bien dentro de un solo edificio o en un campus pequeño.

En este capítulo veremos dos ideas clave que revolucionaron las redes LAN. En primer lugar están los mecanismos que permiten extender una LAN a través de una distancia extensa, y en segundo, la conmutación. El capítulo presenta dos tecnologías relacionadas con la extensión de la red: los repetidores y los puentes. También explica la idea básica del algoritmo de árbol de expansión que se utiliza para evitar el reenvío de bucles. Por último, el capítulo habla sobre los conmutadores y la relación conceptual entre un conmutador y un puente.

17.2 Limitación de distancia y diseño de redes LAN

La limitación de distancia es una parte fundamental de los diseños de redes LAN. Al diseñar una tecnología de red, los ingenieros eligen una combinación de la capacidad, el retraso máximo y la distancia que pueda alcanzarse por un costo dado. La limitación en la distancia surge debido a que el hardware está diseñado para emitir una cantidad fija de energía; si el cableado se extiende más allá de los límites de diseño, las estaciones no recibirán una señal con la fuerza necesaria y se producirán errores. En conclusión:

La especificación de una longitud máxima es una parte fundamental de la tecnología de redes LAN; el hardware de LAN no funcionará correctamente a través de cables que excedan el límite.

17.3 Extensiones de módem de fibra óptica

Los ingenieros desarrollaron varias formas de extender la conectividad de las redes LAN. Como regla general, los mecanismos de extensión no incrementan la intensidad de las señales; tampoco son para extender los cables en el sentido estricto. En su lugar, la mayoría de los mecanismos de extensión insertan componentes de hardware adicionales que pueden transmitir señales a través de distancias más largas.

El mecanismo de extensión LAN más simple consiste en una fibra óptica y en un par de *módems de fibra*, los cuales se utilizan para conectar una computadora a una Ethernet remota. La figura 17.1 ilustra la interconexión.

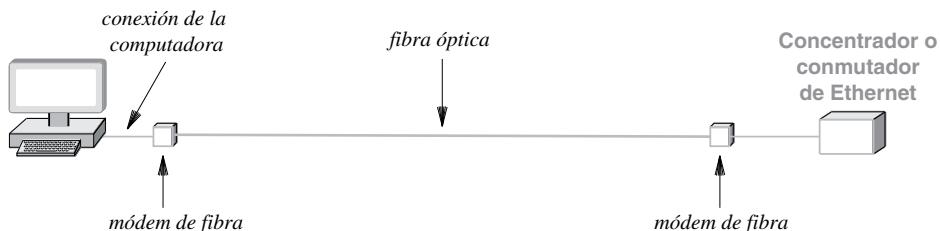


Figura 17.1 Ilustración de los módems de fibra utilizados para ofrecer una conexión entre una computadora y una Ethernet remota.

Cada uno de los módems de fibra contiene hardware para realizar dos tareas: aceptar paquetes de la computadora y enviarlos a través de la fibra óptica hacia el conmutador de Ethernet, así como aceptar paquetes que lleguen del conmutador de Ethernet y enviarlos a la computadora.[†] Si los módems ofrecen una interfaz estándar en cada extremo, ni la computadora ni el conmutador de Ethernet necesitarán hardware especial. La computadora está conectada a un módem de fibra exactamente como se conectaría a un conmutador de Ethernet, y el conmutador de Ethernet está conectado a un módem de fibra exactamente como se conectaría a una computadora.

En conclusión:

Para ofrecer una conexión entre una computadora y una LAN remota, como una Ethernet, es posible usar un par de módems de fibra y cableado de fibra óptica.

[†] En la práctica, las implementaciones usan un par de fibras para permitir la transmisión simultánea en ambas direcciones.

17.4 Repetidores

Un *repetidor* es un dispositivo analógico que se utiliza para propagar las señales de una LAN a través de largas distancias. Un repetidor no entiende paquetes ni bits. En su lugar, el repetidor simplemente amplifica la señal recibida y transmite la versión amplificada como salida.

Los repetidores se utilizaban de manera extensa con la Ethernet original, y se han usado con otras tecnologías de red. Hace poco se introdujeron repetidores con sistemas de control remoto infrarrojos para poder colocar un repetidor a una distancia lejana de un transmisor. Por ejemplo, imagine tener un receptor infrarrojo para el control remoto de su televisor ubicado en un cuarto distinto al del aparato. Un repetidor puede extender la conexión, como se ilustra en la figura 17.2. El repetidor no necesita entender los comandos; simplemente transmite una copia de las señales que llegan al sensor remoto.

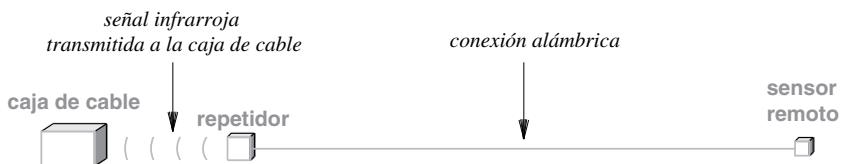


Figura 17.2 Ilustración de un sensor infrarrojo extendido con un repetidor.

Para resumir:

Un repetidor es un dispositivo de hardware analógico que se utiliza para extender una LAN. El repetidor amplifica y envía todas las señales entrantes hacia el otro lado.

17.5 Puentes y uso de puentes

Un *punto* es un mecanismo que conecta dos redes LAN (por ejemplo, dos concentradores) y transfiere paquetes entre ellas. El puente escucha cada segmento en *modo promiscuo* (es decir, recibe todos los paquetes enviados en el segmento). Cuando recibe una trama válida de un segmento, el puente reenvía una copia de la trama al otro segmento. De esta forma, los segmentos de dos redes LAN conectadas por un puente parecen comportarse como una sola LAN; una computadora conectada a uno de los dos segmentos puede enviar una trama a cualquier computadora de los dos segmentos. Además se entrega una trama por difusión a todas las computadoras de los dos segmentos. Por lo tanto, las computadoras no saben si están conectadas a un solo segmento de la LAN o a dos redes LAN conectadas por un puente.

En un principio los puentes se vendían como dispositivos de hardware independientes, cada uno de los cuales tenía dos conexiones de red. En la actualidad, la tecnología de los puentes se incorpora a otros dispositivos, como el módem de cable o DSL. El módem de cable usa un puente para transferir copias de los paquetes de las computadoras de un suscriptor hacia la oficina de la compañía de cable, y viceversa.

Para resumir:

Un puente es un mecanismo que se usa para conectar dos segmentos de redes LAN y reenviar tramas de un segmento a otro; las computadoras no pueden saber si están en un solo segmento o en dos redes LAN conectadas por un puente.

17.6 Puentes con capacidad de aprendizaje y filtrado de tramas

Los puentes no envían a ciegas una copia de cada trama de una LAN a otra, sino que usan direcciones MAC para realizar un *filtrado*. Es decir, un puente analiza la dirección de destino de una trama y no reenvía la trama al otro segmento de LAN a menos que sea necesario. Desde luego que si la LAN soporta la difusión o multidifusión, el puente debe reenviar una copia de cada trama de difusión o multidifusión para que las dos LAN conectadas por el puente operen como una sola.

¿Cómo puede un puente saber qué computadoras están conectadas a cada segmento? La mayoría de los puentes se conocen como puentes *adaptativos* o *con capacidad de aprendizaje*, ya que automáticamente se aprenden las ubicaciones de las computadoras. Para ello, un puente usa direcciones de origen. Cuando llega una trama de un segmento dado, el puente extrae la dirección de origen del encabezado y agrega esa dirección a una lista de computadoras conectadas al segmento. Desde luego que el puente debe entonces extraer la dirección MAC de destino de la trama y usar esa dirección para determinar si debe reenviar o no la trama. De esta forma, tan pronto como la computadora transmite una trama, el puente aprende que esa computadora está presente en un segmento.

Para facilitar la forma en que trabajan los puentes, vamos a considerar una situación en la que un puente está separado de otros dispositivos. La figura 17.3 ilustra la arquitectura conceptual.

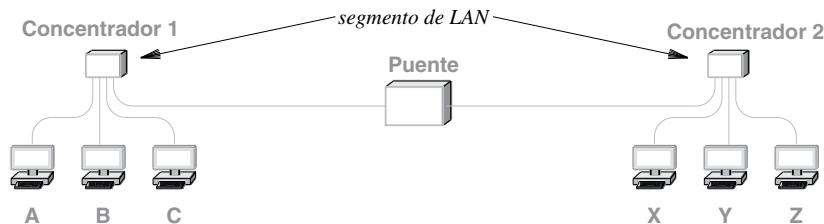


Figura 17.3 Ilustración de seis computadoras conectadas por un puente a un par de segmentos de LAN.

Para ver cómo un puente aprende las ubicaciones de las computadoras a medida que se envían tramas, considere que las computadoras de la figura 17.3 se están comunicando. La figura 17.4 enumera una secuencia de transmisiones de paquetes, la información de la ubicación que acumuló el puente en cada etapa y la disposición del paquete (es decir, los segmentos a través de los cuales se envía el mismo).

Evento	Segmento 1	Segmento 2	Recorrido de la trama
Se inicia el puente	—	—	—
A envía a B	A	—	Ambos segmentos
B envía a A	A, B	—	Sólo segmento 1
X transmite por difusión	A, B	X	Ambos segmentos
Y envía a A	A, B	X, Y	Ambos segmentos
Y envía a X	A, B	X, Y	Sólo segmento 2
C envía a Z	A, B, C	X, Y	Ambos segmentos
Z envía a X	A, B, C	X, Y, Z	Sólo segmento 2

Figura 17.4 Ejemplo de un puente con capacidad de aprendizaje, con las computadoras A, B y C en un segmento y las computadoras X, Y y Z en otro.

Podemos resumir:

Un puente adaptativo usa la dirección MAC de origen de un paquete para registrar la ubicación del emisor, y usa la dirección MAC de destino para determinar a dónde debe reenviar la trama.

17.7 Por qué es bueno usar puentes

Es importante saber que una vez que un puente aprende las ubicaciones de todas las computadoras, una red con puente puede mostrar un rendimiento general mayor que una sola LAN. Para entender por qué, es importante saber que un puente permite la transmisión simultánea en cada segmento. Por ejemplo, en la figura 17.3 la computadora A puede enviar un paquete a la computadora B al mismo tiempo que la computadora X envía un paquete a la computadora Y. Aunque recibe una copia de cada paquete, el puente no reenviará ninguno de ellos porque cada paquete se envió a un destino dentro del mismo segmento que el origen. Por lo tanto, el puente sólo descarta las dos tramas sin reenviarlas. Podemos resumir:

Puesto que un puente permite la actividad simultánea en los segmentos conectados, un par de computadoras en un segmento puede comunicarse al mismo tiempo que un par de computadoras en otro segmento.

La habilidad de ubicar localmente la comunicación permite usar puentes entre distintos edificios en un campus, o entre una residencia y un ISP. La mayor parte de la comunicación es local (por ejemplo,

una computadora se comunica con una impresora en la misma ubicación con más frecuencia de la que se comunica con una impresora en una ubicación remota). Así, un puente puede proveer comunicación entre edificios de un campus cuando sea necesario, pero no envía paquetes si no hay necesidad. Para los módems DSL y de cable que contienen un puente, éste aísla una red local de la red del ISP basándose en la premisa del suscriptor. Por lo tanto, si una computadora que está en la ubicación del suscriptor se comunica con una impresora local, no se reenvían paquetes al ISP. Sin embargo, los paquetes fluyen hacia el ISP cada vez que se necesitan (por ejemplo, cuando el suscriptor usa una computadora para navegar en Web).

17.8 Árbol de expansión distribuido

Considere la figura 17.5 que muestra cuatro segmentos de LAN conectados por tres puentes y un cuarto puente que está a punto de insertarse. Asumimos que las computadoras (que no se muestran en el diagrama) también están conectadas a cada uno de los concentradores.

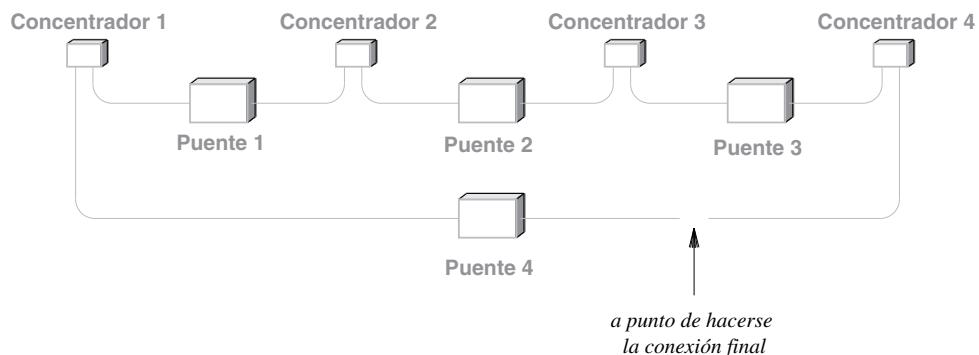


Figura 17.5 Ilustración de una red con puentes, en la que está a punto de insertarse un cuarto puente.

Antes de que se inserte el cuarto puente, la red opera según lo esperado; cualquier computadora puede enviar una trama de unidifusión a otra computadora, o enviar una trama de difusión o multidifusión a todas las computadoras. La difusión y la multidifusión funcionan debido a que un puente siempre reenvía una copia de una trama que se envía a una dirección de difusión o multidifusión. Si se inserta un cuarto puente, surge un problema debido a que existirá un bucle. A menos que se evite que por lo menos un puente reenvíe las difusiones, las copias de una trama de difusión seguirán fluyendo indefinidamente alrededor del ciclo, y las computadoras conectadas a los concentradores recibirán una cantidad interminable de copias.

Para evitar que un bucle provoque un ciclo interminable, los puentes implementan un algoritmo que calcula un árbol *de expansión distribuido* (*DST*). Es decir, el algoritmo ve los puentes como nodos de un gráfico y los ordena a manera de árbol (en matemáticas, un árbol es un gráfico que no contiene ciclos). El método original, que se desarrolló en Digital Equipment Corporation en 1985, se diseñó para redes Ethernet y se conoce como el *protocolo de árbol de expansión* (*STP*). El STP consiste en tres pasos:

- Elección de la raíz
- Cálculo de la ruta más corta
- Reenvío

Para usar el STP, los puentes de Ethernet se comunican entre sí mediante una dirección de multidifusión que se reserva para un árbol de expansión:[†]

01:80:C2:00:00:00

El primer paso consiste en la elección de una raíz. La elección es simple: los puentes transmiten por multidifusión un paquete que contiene su *identificación de puente*, y se selecciona el puente con el identificador más pequeño. Para permitir que un administrador controle la elección, un identificador de puente consiste en dos partes: un *número de prioridad* configurable de 16 bits y una dirección MAC de 48 bits. Al comparar los identificadores, el puente compara primero la porción de la prioridad y usa la porción de la dirección MAC para romper un empate. De esta forma, un administrador puede asegurar que un puente se convierta en la raíz al asignar una prioridad que sea menor que la prioridad de cualquier otro puente.

El segundo paso es el cálculo de la ruta más corta. Cada puente calcula la ruta más corta al puente raíz. El resultado es que los enlaces incluidos en las rutas más cortas de todos los puentes forman el árbol de expansión.

Una vez que se calcula un árbol de expansión, los puentes comienzan a reenviar paquetes. Una interfaz conectada a la ruta más corta está habilitada para reenviar paquetes; una interfaz que no depende de la ruta más corta se bloquea, lo que significa que no pueden enviarse paquetes de usuario a través de la interfaz.

Se han diseñado y estandarizado muchas variaciones de un árbol de expansión. En 1990 el IEEE creó un estándar llamado 802.1d; el estándar se actualizó en 1998. El estándar 802.1q del IEEE proporciona la forma de ejecutar el árbol de expansión en un conjunto de redes lógicamente independientes que comparten un medio físico sin confusión o interferencia entre las redes lógicas. Cisco creó una versión propietaria de un árbol de expansión, el árbol *de expansión por VLAN (PVST)* para usarlo en un conmutador de VLAN,[‡] y más tarde actualizó el protocolo a *PVST+*, haciéndolo compatible con el estándar 802.1q. En 1998, el estándar 802.1w del IEEE introdujo el *protocolo de árbol de expansión rápido* para reducir el tiempo requerido para la convergencia después de un cambio de topología. El árbol de expansión rápido se incorporó en el estándar 801.1d-2004 y ahora reemplaza al STP. Se definieron versiones conocidas como el *protocolo de árbol de expansión de múltiples instancias (MISTP)* y *protocolo de árbol de expansión múltiple (MSTP)* para manejar conmutadores VLAN más complejos; el MSTP se incorporó al estándar 802.1q-2003 del IEEE.

17.9 Comutación y conmutadores de la capa 2

El concepto de los puentes ayuda a explicar un mecanismo que forma la base de las redes Ethernet modernas: la *comutación*. Un *conmutador de Ethernet*, algunas veces conocido como *conmutador de capa 2*, es un dispositivo electrónico parecido a un concentrador. Al igual que un concentrador, un conmutador provee varios *puertos* que se conectan cada uno a una sola computadora, y éste permite que las computadoras se envíen tramas entre sí. La diferencia entre un concentrador y un conmutador

[†] Las direcciones de Ethernet se escriben en hexadecimal con símbolos de dos puntos que separan cada par de dígitos hexagonales.

[‡] Las siguientes secciones describen la comutación y los conmutadores de VLAN.

se debe a la forma en que los dispositivos operan: un concentrador opera como un dispositivo analógico que reenvía señales entre computadoras, mientras que un conmutador es un dispositivo digital que reenvía paquetes. Podemos considerar un concentrador como la simulación de un medio de transmisión compartido, y a un conmutador como la simulación de una red con puentes que tiene una computadora por cada segmento de la LAN. La figura 17.6 ilustra el uso conceptual de los puentes en un conmutador.

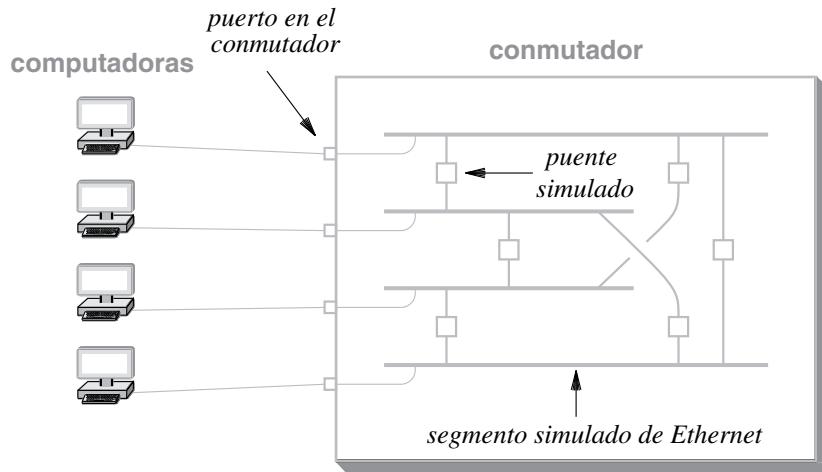


Figura 17.6 Organización conceptual de una LAN conmutada.

Aunque la figura proporciona una vista conceptual, un conmutador no contiene puentes separados. En su lugar, un conmutador consiste en una *interfaz inteligente* conectada a cada puerto y una *fábrica* central que brinda una transferencia simultánea entre pares de interfaces. La interfaz contiene un procesador, una memoria y demás hardware necesario para aceptar un paquete entrante, consultar una tabla de reenvío y enviar el paquete a través de la fábrica hacia el puerto de salida correcto. La interfaz también acepta paquetes de la fábrica y los transmite hacia fuera por el puerto. Lo que es más importante, como contiene memoria, una interfaz puede colocar en el búfer los paquetes entrantes cuando un puerto de salida esté ocupado. Por lo tanto, si la computadora 1 y la computadora 2 envían al mismo tiempo paquetes a la computadora 3, la interfaz 1 o la interfaz 2 contendrán un paquete mientras que la otra interfaz transmite. La figura 17.7 ilustra la arquitectura.

En el sentido físico, los conmutadores están disponibles en muchos tamaños. Los más pequeños consisten en un dispositivo económico independiente que proporciona cuatro conexiones, las cuales son suficientes para conectar una computadora, una impresora y otros dos dispositivos (por ejemplo, un escáner y un disco duro de respaldo). Las empresas usan los conmutadores más grandes para conectar decenas de miles de computadoras y otros dispositivos a lo largo de una compañía.

La principal ventaja de usar una LAN conmutada en vez de un concentrador es el paralelismo. Mientras un concentrador sólo puede soportar una transmisión a la vez, un conmutador permite que ocurran varias transferencias al mismo tiempo, siempre y cuando sean independientes (es decir, que sólo se transfiera un paquete a un puerto específico en un momento dado). Por consiguiente, si un conmutador tiene N puertos conectados a N computadoras, pueden ocurrir $N/2$ transferencias al mismo tiempo. En conclusión:

Puesto que maneja paquetes en vez de señales y usa una fábrica para proporcionar rutas internas paralelas, un comutador con N puertos puede transferir hasta $N/2$ paquetes al mismo tiempo.

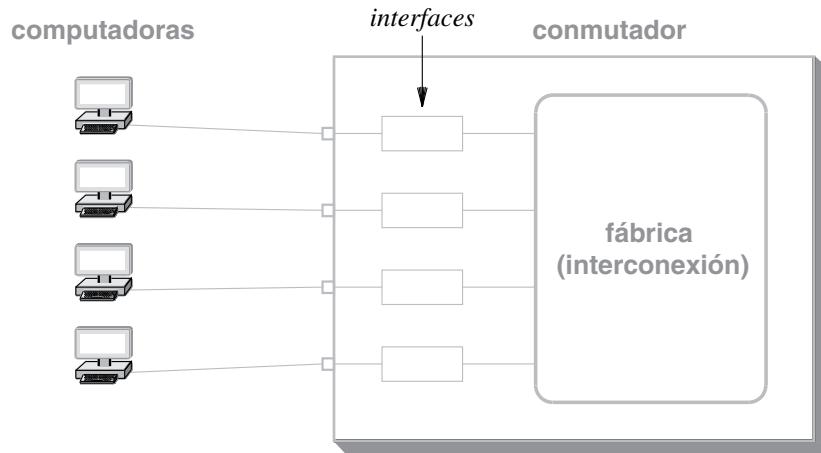


Figura 17.7 Ilustración de la arquitectura de un comutador.

17.10 Comutadores de redes VLAN

Los comutadores se ampliaron al agregar una virtualización, lo que se conoce como *comutador de red de área local virtual (comutador de VLAN)*. El concepto es simple: permitir que un administrador configure un solo comutador para emular varios comutadores independientes. Es decir, un administrador especifica un conjunto de puertos en el comutador y los designa para que estén en la LAN 1 virtual, designa otro conjunto de puertos para que estén en la LAN 2 virtual, y así sucesivamente. Cuando una computadora en la LAN 2 virtual difunde un paquete, únicamente las computadoras de la misma LAN virtual reciben una copia (es decir, una vez configurado, un comutador de VLAN hace parecer que hay varios comutadores). En esencia, el comutador de VLAN actúa como un conjunto de comutadores más pequeños.

Dividir las computadoras en *dominios de difusión* independientes no parece importante sino hasta que consideramos una empresa o proveedor de servicios de gran tamaño. En cada caso, tal vez sea importante garantizar que un conjunto de computadoras pueden comunicarse sin que otros reciban los paquetes y sin recibir paquetes de fuera. Por ejemplo, tal vez una empresa decida agregar un sistema *cortafuegos* o *firewall* entre las computadoras de la oficina del CEO y las demás computadoras de la empresa.[†] Al configurar una VLAN independiente para las computadoras del CEO, es posible la instalación de un firewall.

[†] En el capítulo 29 se describen los firewalls.

17.11 Múltiples conmutadores y redes VLAN compartidas

Por lo general, los conmutadores se colocan en proximidad física a las computadoras. Por ejemplo, una organización puede optar por colocar un conmutador en cada piso de un edificio, aun cuando se ocupen algunas oficinas de varios pisos. Al ubicar un conmutador cerca de un conjunto de computadoras, se reduce el número de alambres que deben tenderse. Además, los conmutadores convencionales pueden interconectarse para formar una sola red extensa. De esta forma, sólo hay que tender un cable entre conmutadores.

¿Pueden las redes VLAN operar a través de varios conmutadores? No; por lo menos no sin soporte adicional. Para entender por qué, considere el hardware. Puesto que es posible pasar información de configuración a cada componente de hardware, es fácil imaginar redes VLAN que operan en un solo conmutador de VLAN. Así, si un administrador de red especifica que los puertos 1, 12, 18 y 46 están en una VLAN, es posible configurar cada una de las cuatro interfaces de hardware para enviar paquetes de difusión entrantes a las otras tres. Pero si se interconecta un par de conmutadores, el puerto 1 en el primer conmutador y el puerto 1 del segundo podrían estar en redes VLAN diferentes, lo cual significa que un número de puerto no es suficiente para identificar una VLAN.

El IEEE extendió el estándar de Ethernet para que fuera posible configurar redes VLAN que atravesen varios conmutadores. A cada VLAN se le asigna un número único. En vez de enviar tramas normales de Ethernet a través de la conexión entre los conmutadores, se agrega un campo adicional al encabezado de cada paquete. El campo adicional de la trama es un entero de 16 bits conocido como *etiqueta de VLAN*. Una etiqueta de VLAN identifica la VLAN a la que se asignó la computadora emisora. Es decir, si un administrador configuró el puerto 5 de un conmutador para que sea parte de la red VLAN 17, al recibir la trama el conmutador inserta una etiqueta en el encabezado con el valor 17. El interruptor sólo mantiene la etiqueta para uso interno. Antes de entregar una trama a una de las computadoras conectadas, el conmutador retira la etiqueta.

El estándar 802.1Q del IEEE especifica el formato de una trama de Ethernet que contiene una etiqueta de VLAN. Lo interesante es que el campo de la etiqueta no se coloca al principio ni al final de la trama. En su lugar, se inserta una etiqueta entre la dirección de origen y los campos de tipo de Ethernet. La figura 17.8 ilustra el formato de una trama 802.1Q; compare el formato con el de una trama de Ethernet estándar que se muestra en la figura 15.1.[†]

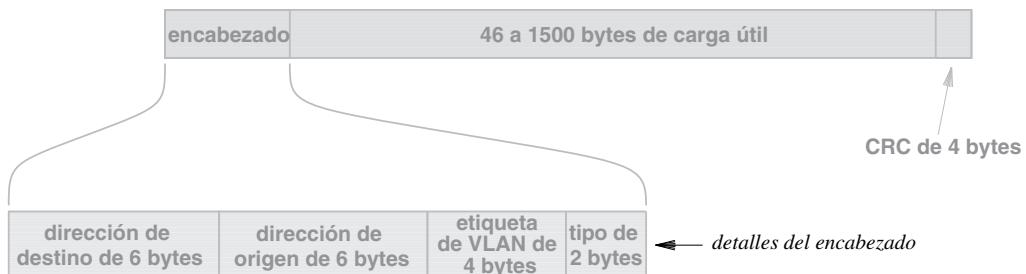


Figura 17.8 El formato de una trama de Ethernet con una etiqueta VLAN del estándar 802.1Q.

[†] Encontrará la figura 15.1 en la página 254.

El estándar 802.1Q permite interconectar varios comutadores de VLAN para que operen como un comutador de VLAN gigante. Para ello, se configura un conjunto de comutadores que usarán el formato 802.1Q para las tramas que se envían en los enlaces entre comutadores. Puesto que sólo los comutadores entienden el formato 802.1Q, en los enlaces hacia las computadoras individuales se usa el formato de trama Ethernet estándar. La figura 17.9 muestra un ejemplo de comutadores interconectados que usan el estándar 802.1Q.

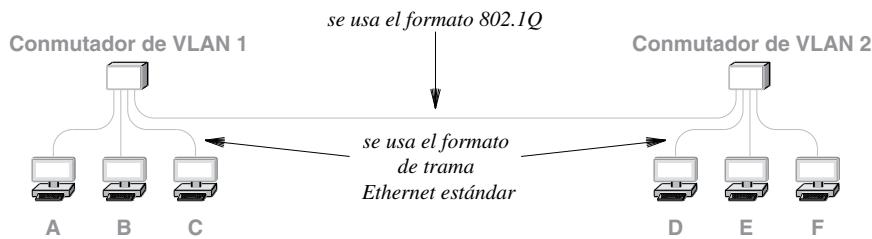


Figura 17.9 Un ejemplo de cómo se usa el formato de trama 802.1Q con los comutadores de VLAN interconectados.

Cuando una computadora transmite una trama por difusión, el comutador al que está conectada entrega una copia a cada puerto local que forma parte de la VLAN, luego inserta la etiqueta de VLAN y envía el resultado a través del enlace entre comutadores. El segundo comutador recibe la trama, extrae el número de VLAN y entrega una copia a cada computadora que forma parte de la VLAN.

Para que los comutadores de redes VLAN interconectadas operen correctamente, el administrador de la red debe asignar los números de VLAN con cuidado y debe configurar cada comutador para que use los números de VLAN de manera consistente. Por ejemplo, en la figura 17.9 un administrador podría asignar las computadoras A, B y E a la VLAN 10, las computadoras C y D a la VLAN 11 y la computadora F a la VLAN 12. Para configurar la red, el administrador tendría que configurar cada comutador por separado.

17.12 La importancia de los puentes

Aunque nuestra descripción se concentra en un puente como dispositivo independiente, el uso de puentes es un concepto fundamental que se ha incorporado a muchos sistemas de redes. Dijimos que un módem DSL o de cable funciona como una especie de puente en donde los paquetes se envían al ISP sólo cuando éstos no se envían a un destino local de la red del suscriptor. Además, muchos enrutadores inalámbricos incorporan un comutador de Ethernet que proporciona funcionalidad de puente. El comutador de Ethernet en un enrutador inalámbrico tiene varios puertos a los que pueden conectarse las computadoras (por lo general, los puertos del comutador se etiquetan con la palabra *local* para diferenciarlos del puerto de *Internet* que se conecta al módem del ISP). Si dos computadoras locales se comunican, el enrutador inalámbrico enviará paquetes entre ellos sin enviar los paquetes al ISP. En conclusión:

Aunque los distribuidores ya no venden dispositivos tipo puente independientes, el concepto de los puentes se incorporó a los dispositivos de red, como los módems y los enrutadores inalámbricos.

17.13 Resumen

Se crearon varios mecanismos para extender las redes LAN a través de una distancia geográfica más grande. Es posible usar un par de módems de fibra para extender la conexión entre una computadora y una LAN. Un repetidor es un dispositivo analógico que amplifica las señales eléctricas de un segmento de LAN y transmite una copia en el otro, y viceversa. Un puente es un dispositivo digital que conecta dos segmentos de LAN y transfiere paquetes entre ellos.

Para optimizar el reenvío, un puente examina las direcciones MAC en el encabezado de cada trama y memoriza qué computadoras están conectadas a cada segmento. Una vez que un puente aprende la ubicación de una computadora, no reenvía a las demás computadoras del segmento las tramas que están destinadas a esa computadora.

Un conmutador de Ethernet conecta varias computadoras y reenvía tramas entre ellas. En teoría, un conmutador funciona como un conjunto de segmentos de LAN interconectados por puentes. En la práctica, un conmutador contiene un conjunto de interfaces inteligentes que usan un mecanismo de interconexión de hardware de alta velocidad, conocido como *fábrica*. La principal ventaja de un conmutador en comparación con un concentrador es que el primero puede transferir varios paquetes al mismo tiempo, siempre y cuando sólo se destine un paquete a la vez para un puerto de salida específico. Un conmutador de VLAN permite a un administrador configurar un conmutador para que actúe como un conjunto de conmutadores independientes.

EJERCICIOS

- 17.1** Cuando se usa una fibra óptica para extender una conexión a una LAN, ¿qué dispositivos adicionales se necesitan?
- 17.2** Si un televisor proporciona una extensión alámbrica para un sensor infrarrojo remoto, ¿qué tecnología está probablemente siendo usada?
- 17.3** Si se conectan dos computadoras en una red con un puente, ¿se requieren cambios en el direccionamiento o en las aplicaciones? Explique.
- 17.4** Proporcione una declaración precisa de las condiciones bajo las que un puente adaptativo reenviará un paquete.
- 17.5** Considere un paquete que se envía a una dirección inexistente en una LAN con puentes. ¿A través de cuántos segmentos los puentes reenviarán el paquete?
- 17.6** Suponga que una red contiene tres segmentos de Ethernet que operan a 100 Mbps conectados por dos puentes y que cada segmento contiene una computadora. Si dos computadoras envían a una tercera, ¿cuál es la velocidad máxima de datos que puede lograr un emisor? Y la mínima?

- 17.7** Busque en Web una descripción del algoritmo del árbol de expansión y escriba un programa de computadora que simule puentes que formen un árbol de expansión.
- 17.8** ¿Las computadoras en una Ethernet con puentes reciben paquetes de un árbol de expansión? Explique.
- 17.9** Use un analizador de red para observar el tráfico en una Ethernet con puentes. ¿Qué observa después del reinicio de un puente?
- 17.10** Cuando se usan puentes con un enlace satelital, por lo general se usan dos puentes: uno en cada lado. Explique por qué.
- 17.11** De acuerdo con la figura 17.6, ¿pueden dos computadoras conectadas a una LAN conmutada transmitir paquetes al mismo tiempo? Explique.
- 17.12** Amplíe la figura 17.6 para tener cinco puertos. Haga el diagrama.
- 17.13** En el ejercicio anterior, escriba una ecuación que proporcione el número de puentes simulados necesarios como una función del número de puertos.
- 17.14** Escriba un programa de computadora que simule una función de puente. Deje que dos archivos de datos simulen las tramas transmitidas en dos segmentos a los que se conecta el puente. Suponga que cada trama simulada contiene una dirección de origen y de destino. Para realizar la simulación, lea una trama del primer archivo, luego una trama del segundo archivo y así sucesivamente. Para cada trama muestre si el puente reenviará una copia de la trama al otro segmento de LAN.
- 17.15** Amplíe el programa del ejercicio anterior para simular un interruptor de VLAN. Haga que el programa comience leyendo la información de configuración que especifica un conjunto de hosts y un conjunto de redes LAN virtuales a las que deben conectarse. Cree un archivo de tramas en donde cada una especifique qué computadora envía la trama (es decir, el puerto del conmutador a través del cual llega la misma) y una dirección de destino. Muestre cómo se reenvía cada trama.
- 17.16** ¿Puede un puente conectar una red Wi-Fi a una Ethernet? ¿Puede hacerlo un conmutador? ¿Por qué sí o por qué no?

Contenido del capítulo

- 18.1 Introducción, 305
- 18.2 Grandes extensiones y redes de área amplia, 305
- 18.3 Arquitectura de WAN tradicional, 306
- 18.4 Formación de una WAN, 308
- 18.5 Paradigma de almacenamiento y reenvío, 309
- 18.6 Direcciónamiento en una WAN, 309
- 18.7 Reenvío del siguiente salto, 310
- 18.8 Independencia del origen, 313
- 18.9 Actualizaciones de enrutamiento dinámico en una WAN, 313
- 18.10 Rutas predeterminadas, 314
- 18.11 Cálculo de la tabla de reenvío, 315
- 18.12 Cálculo de la ruta distribuido, 316
- 18.13 Rutas más cortas y pesos, 320
- 18.14 Problemas de enrutamiento, 321
- 18.15 Resumen, 322

18

Tecnologías de WAN y enrutamiento dinámico

18.1 Introducción

Los capítulos de esta parte del libro describen una variedad de tecnologías de conmutación de paquetes tanto alámbricas como inalámbricas. En el capítulo anterior vimos las extensiones, los puentes y la conmutación de las LAN. En este capítulo consideraremos la estructura de una sola red que se extiende por un área amplia. El capítulo describe los componentes básicos utilizados para crear un sistema de conmutación de paquetes y explica el concepto fundamental del enrutamiento. También presenta los dos algoritmos de enrutamiento básicos y explica las ventajas de cada uno. En un capítulo posterior ampliaremos la explicación del enrutamiento hacia Internet y presentaremos los protocolos de enrutamiento que usan los algoritmos aquí descritos.

18.2 Grandes extensiones y redes de área amplia

Anteriormente dijimos que las tecnologías de redes pueden clasificarse de acuerdo con la extensión que cubren, en:

- PAN: se extiende alrededor de un individuo
- LAN: se extiende por un edificio o campus
- MAN: se extiende por un área metropolitana grande
- WAN: se extiende a través de varias ciudades o países

Considere una empresa que usa un puente satelital para conectar las redes LAN de dos sedes. ¿Debería clasificarse la red como WAN o como LAN extendida? ¿Cambia la respuesta si la empresa sólo tiene una PC y una impresora en cada sitio? Claro que sí. La cuestión clave que separa las tecnologías WAN de las tecnologías LAN es la *escalabilidad*. Una WAN debe ser capaz de crecer lo necesario para conectar muchos sitios a través de distancias extensas, con muchas computadoras en cada sitio. Por ejemplo, una WAN debe ser capaz de conectar todas las computadoras de una gran corporación que tenga oficinas o fábricas en varias ciudades. Además, una tecnología no se clasifica como WAN a menos que pueda ofrecer un rendimiento razonable para una red de gran escala. Es decir, una WAN no sólo se conecta a muchas computadoras en muchos sitios, sino que además debe ofrecer la capacidad suficiente para permitir que todas las computadoras se comuniquen de manera óptima. De tal forma, un puente satelital que conecta un par de computadoras e impresoras es simplemente una LAN extendida.

18.3 Arquitectura de WAN tradicional

En capítulos posteriores aprenderemos que la mayoría de los sistemas modernos de comunicaciones que abarcan grandes distancias usan la tecnología de Internet, en la que se coloca un enrutador en cada sitio y los enruteadores se interconectan mediante circuitos arrendados. Sin embargo, este capítulo considera una metodología más simple: una sola red que atraviesa distancias geográficas extensas. Los conceptos y ejemplos se obtienen de la historia. Antes de crear Internet, la investigación de redes se concentraba en encontrar nuevas formas de crear redes individuales que pudieran interconectar varios sitios. Los primeros trabajos usaban el término *redes de largo alcance*, pero después de inventar las tecnologías de LAN, el término *red de área amplia* se hizo popular.

La época durante la que se desarrollaron las tecnologías de redes WAN es importante. La investigación sobre las redes WAN fue anterior al trabajo con las LAN y a la llegada de las computadoras personales. En ese tiempo, incluso las grandes corporaciones o universidades sólo tenían una o dos computadoras grandes. Como consecuencia, cuando se diseñaron las tecnologías WAN, el objetivo era conectar muchos sitios con unas cuantas computadoras en cada sitio. Lo que es más importante, las tecnologías WAN tenían que permitir que las computadoras se conectaran a una red de largo alcance.

Como las tecnologías LAN no se habían inventado todavía, los diseñadores de las WAN eligieron un método que colocaba un dispositivo de hardware especial en cada sitio. Conocido como *comutador de paquetes*, el dispositivo proporciona conexiones locales para las computadoras del sitio, así como conexiones para los circuitos de datos que conducen a otros sitios.

En teoría, un comutador de paquetes consiste en un pequeño sistema de cómputo con un procesador, una memoria y dispositivos de E/S que se usan para enviar y recibir paquetes. Los primeros comutadores de paquetes se construyeron a partir de computadoras convencionales. Los comutadores de paquetes utilizados en las WAN de más alta velocidad requieren de un hardware especial. La figura 18.1 ilustra la arquitectura interna de un comutador de paquetes tradicional.

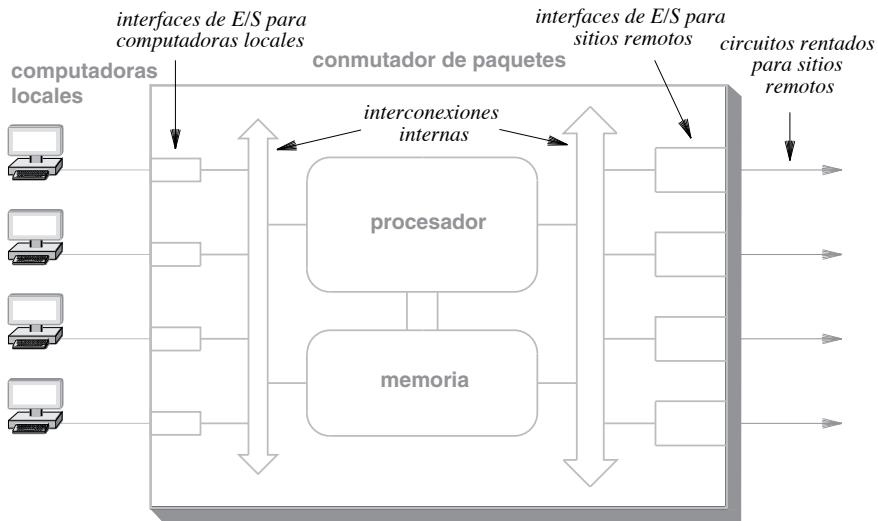


Figura 18.1 Ilustración de la arquitectura de un comutador de paquetes tradicional.

Como se muestra en la figura, un comutador de paquetes contiene dos tipos de dispositivos de E/S. El primero, que opera a velocidades altas, se usa para conectar el comutador a un circuito digital que conduce a otro comutador de paquetes. El segundo tipo de dispositivo de E/S, que opera a una velocidad más baja, se usa para conectar el comutador a una computadora individual.

Desde la llegada de la tecnología de LAN, la mayoría de las WAN separan un comutador de paquetes en dos partes: un comutador de la capa 2 que conecta las computadoras locales y un enrutador que se conecta a otros sitios. La parte IV del libro habla en detalle sobre los enrutadores de Internet y explica cómo es que los conceptos aquí cubiertos se aplican a Internet; por ahora basta con entender que la comunicación con las computadoras locales puede separarse de la transmisión a través de una WAN. La figura 18.2 ilustra la separación.

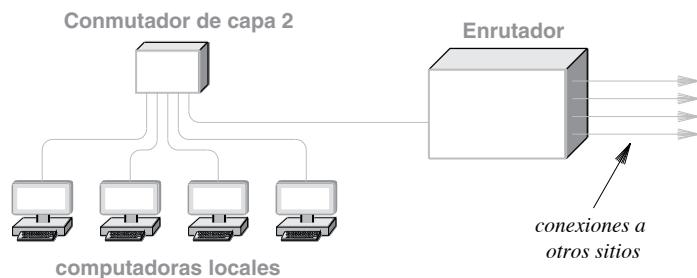


Figura 18.2 Ilustración de un sitio de WAN moderno, donde la comunicación local se maneja mediante una LAN separada.

18.4 Formación de una WAN

En teoría, una WAN puede formarse mediante la interconexión de un conjunto de sitios. Los detalles exactos de las interconexiones dependen de la velocidad de datos necesaria, la distancia cubierta y el retraso que pueda tolerarse. Como se describe en el capítulo 12, muchas WAN usan circuitos de datos rentados (por ejemplo, un circuito T3 o un circuito OC-12). Sin embargo, también están disponibles otras formas de conexión, como los canales de microondas y de satélite. Además de elegir la tecnología para una conexión, un diseñador debe elegir una topología. Para un conjunto dado de sitios, hay muchas topologías posibles. Por ejemplo, la figura 18.3 ilustra una posible manera de interconectar cuatro conmutadores de paquetes tradicionales y ocho computadoras.

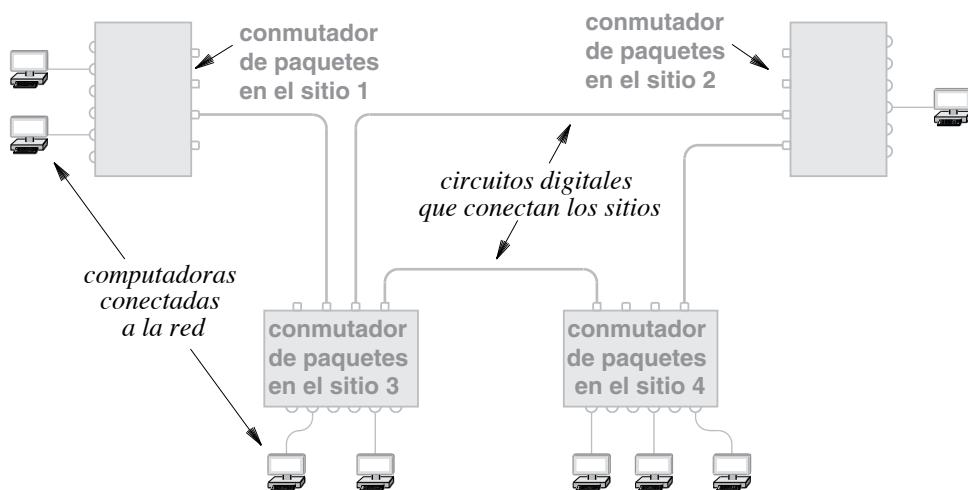


Figura 18.3 Una WAN de ejemplo formada por conmutadores de paquetes en interconexión.

Como se indica en la figura, una WAN no necesita ser simétrica: las interconexiones entre conmutadores de paquetes y la capacidad de cada conexión se eligen con el fin de soportar el tráfico esperado y proporcionar redundancia en caso de falla. En la figura, el conmutador de paquetes en el sitio 1 sólo tiene una conexión al resto de la red, mientras que los conmutadores de paquetes en los otros sitios tienen al menos dos conexiones externas. En conclusión:

Una WAN tradicional se forma mediante la interconexión de conmutadores de paquetes; un conmutador de paquetes en cada sitio se conecta a las computadoras. La topología y la capacidad de las conexiones son elegidas de acuerdo con el tráfico esperado y la necesidad de redundancia.

18.5 Paradigma de almacenamiento y reenvío

El objetivo de una WAN es permitir que tantas computadoras como sea posible puedan enviar paquetes al mismo tiempo. El paradigma fundamental utilizado para lograr una transmisión simultánea se conoce como *almacenamiento y reenvío*. Para procesar el almacenamiento y reenvío, un conmutador de paquetes coloca éstos en un *búfer* de memoria. La operación de *almacenar* ocurre cuando llega un paquete: el hardware de E/S dentro del conmutador de paquetes coloca en la memoria una copia del paquete. La operación *reenviar* ocurre una vez que llega un paquete y está esperando en la memoria. El procesador analiza el paquete, determina su destino y lo envía a través de la interfaz de E/S que conduce al destino.

Un sistema que usa el paradigma de almacenar y reenviar puede mantener cada enlace de datos ocupado y por consiguiente aumentar el rendimiento general. Lo que es más importante, si se envían varios paquetes al mismo dispositivo de salida, el conmutador de paquetes puede aceptar y mantenerlos en la memoria hasta que el dispositivo esté listo. Por ejemplo, considere la transmisión de paquetes en la red de la figura 18.3. Suponga que cada una de las computadoras del sitio 1 generan al mismo tiempo un paquete destinado para una computadora del sitio 3. Ambas computadoras pueden enviar al mismo tiempo su paquete al conmutador de paquetes. Conforme llega cada paquete, el hardware de E/S del conmutador de paquetes los coloca en memoria e informa al procesador del conmutador de paquetes. El procesador analiza el destino de cada paquete y determina que el paquete debería enviarse al sitio 3. Si la interfaz de salida que conduce al sitio 3 está inactiva cuando llega un paquete, la transmisión comienza de inmediato. Si el dispositivo de salida está ocupado, el procesador coloca el paquete saliente en una cola asociada con el dispositivo. Tan pronto como termina de enviar un paquete, el dispositivo extrae y envía el siguiente paquete de la cola.

El concepto puede resumirse así:

Los sistemas de conmutación de paquetes de área amplia usan la técnica de almacenar y reenviar, en la que los paquetes que llegan a un conmutador de paquetes se colocan en una cola hasta que éste puede reenviarlos hacia su destino. La técnica permite a un conmutador de paquetes colocar en el búfer una ráfaga corta de paquetes que llegan al mismo tiempo.

18.6 Direccionamiento en una WAN

Desde la perspectiva de una computadora conectada, una red WAN tradicional opera de manera similar a una LAN. Cada tecnología de WAN define el formato de trama exacto que una computadora usa al enviar y recibir datos. Además, a cada computadora conectada a una WAN se le asigna una dirección. Al enviar una trama a otra computadora, el emisor debe proporcionar la dirección de destino.

Aunque los detalles varían, las direcciones WAN siguen un concepto clave que se usa en Internet: el *direccionamiento jerárquico*. En concepto, el direccionamiento jerárquico divide cada dirección en dos partes:

[sitio,computadora en el sitio]

En la práctica, hay un conmutador de paquetes en cada sitio y el esquema de direccionamiento asigna un número único a cada conmutador de paquetes. De esta forma, la primera parte de una dirección identifica a un conmutador de paquetes y la segunda parte identifica a una computadora conectada al conmutador. Por ejemplo, la figura 18.4 muestra direcciones jerárquicas en dos partes que se asignan a las computadoras conectadas a un par de conmutadores de paquetes.

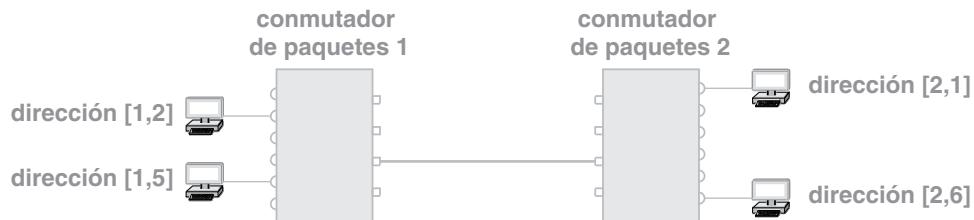


Figura 18.4 Ejemplo de una jerarquía de direcciones donde cada dirección identifica a un conmutador de paquetes y a una computadora conectada al conmutador.

La figura muestra cada dirección como un par de enteros decimales. Por ejemplo, a una computadora conectada al puerto 6 en el conmutador de paquetes 2 se le asigna la dirección [2,6]. Aunque pensamos en la dirección como si fuera un par de enteros, una dirección se representa como un solo valor binario, donde se usan los bits iniciales del valor binario para representar un número de conmutador de paquetes y el resto de los bits se usan para identificar a una computadora. En la parte IV del texto veremos que Internet usa el mismo esquema: cada dirección de Internet consiste en un número binario donde la primera porción de los bits identifica a una red específica en Internet y el resto identifica a una computadora conectada a la red.

18.7 Reenvío del siguiente salto

La importancia del direccionamiento jerárquico se vuelve clara cuando consideramos el procesamiento de los paquetes. Cuando llega un paquete, un conmutador de paquetes debe elegir una ruta saliente a través de la cual reenviarlo. Si un paquete está destinado para una computadora local (es decir, una computadora conectada al mismo conmutador), el conmutador envía el paquete directamente a la computadora de destino. De lo contrario, debe reenviar el paquete a través de una de las conexiones que conducen a otro conmutador. El software en el conmutador de paquetes usa la dirección de destino del paquete para tomar la decisión. El software extrae de la dirección el número del conmutador de paquetes. Si el valor extraído es idéntico al identificador del conmutador de paquetes, el paquete está destinado para una computadora local. Si el número no coincide, entonces el paquete está destinado para una computadora en otro conmutador de paquetes. El algoritmo 18.1 explica el cálculo.

La idea importante es que un conmutador de paquetes no necesita mantener la información completa sobre cómo llegar a todas las computadoras posibles, y tampoco necesita calcular toda la ruta que seguirá un paquete a través de la red. En su lugar, un conmutador basa el reenvío en los identificadores

de los commutadores de paquetes, lo que significa que un commutador sólo necesita saber qué enlace saliente usar para llegar a un commutador específico.

Algoritmo 18.1

Dado:

Un paquete que llega al commutador de paquetes Q

Se realiza:

El paso del reenvío del siguiente salto

Método:

Extraer la dirección de destino del paquete;

Dividir la dirección en un número de commutador de paquete P

y un identificador de computadora C;

if ($P == Q$) { /* el destino es local */

 Reenviar el paquete a la computadora local C;

} else {

 Seleccionar un enlace que conduzca a otro commutador de paquetes

 y reenviar el paquete a través del enlace;

}

Algoritmo 18.1 Los dos pasos que usa un commutador de paquetes para reenviar un paquete cuando se usa el reenvío del siguiente salto.

Decimos que un commutador sólo necesita calcular el *siguiente salto* para un paquete. El proceso se conoce como *reenvío del siguiente salto* y es similar a la forma en que las aerolíneas enlistan los vuelos. Suponga que un pasajero de una aerolínea que viaja de San Francisco a Miami descubre que el único itinerario disponible involucra tres vuelos: el primero de San Francisco a Dallas, el segundo de Dallas a Atlanta y el tercero de Atlanta a Miami. Aunque el destino final (Miami) sigue siendo el mismo en todo el viaje, el siguiente salto cambia en cada aeropuerto. Cuando el pasajero sale de San Francisco, el siguiente salto es Dallas. Cuando el pasajero está en Dallas, el siguiente salto es Atlanta y cuando el pasajero está en Atlanta, el siguiente salto es Miami.

Para que el cálculo sea eficiente, los commutadores de paquetes usan la búsqueda por tablas. Es decir, cada commutador de paquetes contiene una *tabla de reenvío*[†] que enlista todos los posibles commutadores de paquetes y proporciona el siguiente salto para cada uno. La figura 18.5 ilustra el reenvío del siguiente salto con un ejemplo trivial.

[†]Aunque los puristas insisten en el nombre *tabla de reenvío*, dichas tablas se conocían en un principio como *tablas de enrutamiento* y la terminología todavía se utiliza ampliamente en la industria de las redes.

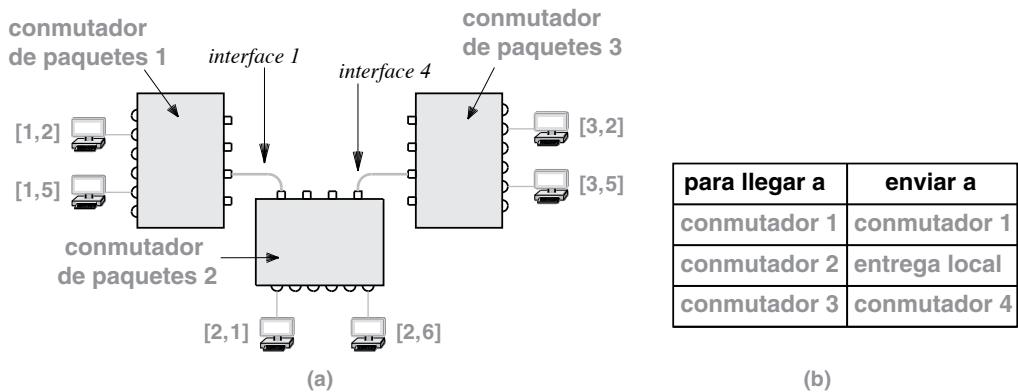


Figura 18.5 (a) Una red de tres comutadores de paquetes y (b) la tabla de reenvío del siguiente salto para el comutador 2.

Para usar una tabla de reenvío, un comutador extrae del paquete la dirección de destino y usa la parte de la dirección correspondiente al comutador de paquetes como un índice de la tabla de reenvío. Por ejemplo, considere la tabla de la figura 18.5(b). Si un paquete está destinado para [3,5], el comutador extrae el 3, consulta la tabla y reenvía el paquete a la interfaz 4, que conduce al comutador 3.

Usar sólo una de las dos partes de una dirección jerárquica para reenviar un paquete tiene dos consecuencias prácticas. Primero, el tiempo de cálculo requerido para reenviar un paquete se reduce debido a que la tabla de reenvío puede organizarse como un arreglo que use indexación en vez de búsqueda. Segundo, la tabla de reenvío contiene una entrada por cada comutador de paquetes en vez de una entrada por cada computadora de destino. La reducción en el tamaño de la tabla puede ser considerable, en especial para una WAN extensa que tenga muchas computadoras conectadas a cada comutador de paquetes.

En esencia, el esquema de direccionamiento jerárquico de dos partes permite a los comutadores de paquetes usar sólo la primera parte de la dirección de destino hasta que el paquete llegue al comutador final (es decir, el comutador al que está conectada la computadora de destino). Una vez que el paquete llega al comutador final, el comutador usa la segunda parte de la dirección para elegir una computadora específica, como se describe en el algoritmo 18.1.

Para resumir:

Al reenviar un paquete a través de una WAN, únicamente se usa la primera parte de una dirección de destino. Una vez que el paquete llega al comutador al que está conectada la computadora de destino, se usa la segunda parte de la dirección para reenviar el paquete a la computadora local correcta.

Tal vez parezca que nuestro ejemplo es demasiado simple y se requieren más detalles. En este momento es suficiente con entender este principio. En capítulos posteriores veremos que los enruteadores de Internet usan el reenvío del siguiente salto y hablaremos sobre esto con mayor detalle.

18.8 Independencia del origen

Cabe mencionar que el reenvío del siguiente salto no depende de la fuente original del paquete ni de la ruta que haya tomado éste antes de llegar a un conmutador de paquetes específico; el siguiente salto al que se envía un paquete depende sólo del destino del paquete. El concepto, que se conoce como *independencia del origen*, es una idea fundamental en las redes y será implícito en nuestras explicaciones de todo el capítulo y en capítulos posteriores que describan el reenvío en Internet.

La independencia del origen permite que el mecanismo de reenvío en una red de computadoras sea compacto y eficiente. Puesto que todos los paquetes siguen la misma ruta, sólo se necesita una tabla. Como el reenvío no usa la información del origen únicamente hay que extraer la dirección de destino de un paquete. Además, un solo mecanismo maneja el reenvío de manera uniforme. Tanto los paquetes que se originan en computadoras conectadas directamente como los paquetes que llegan de otros conmutadores de paquetes, usan el mismo mecanismo.

18.9 Actualizaciones de enrutamiento dinámico en una WAN

Para que una WAN opere correctamente, cada conmutador debe tener una tabla de reenvío y debe reenviar los paquetes. Además, los valores en la tabla de reenvío deben garantizar lo siguiente:

- Comunicación universal. La tabla de reenvío de cada conmutador debe contener una ruta válida del siguiente salto para cada posible dirección de destino.
- Rutas óptimas. En un conmutador, el valor del siguiente salto en la tabla de reenvío para un destino dado debe apuntar a la ruta más corta hacia el destino.

Las fallas de la red complican todavía más el reenvío. Por ejemplo, si existen dos rutas hacia un destino dado y una de ellas no está disponible debido a que el hardware falla (digamos, si se desconecta un circuito), hay que cambiar el reenvío para evitar la ruta no disponible. Por lo tanto, un administrador no puede simplemente configurar una tabla de reenvío para que contenga valores estáticos que no cambien. En su lugar, el software que se ejecuta en los conmutadores de paquetes realiza pruebas continuas en busca de fallas y reconfigura automáticamente las tablas de reenvío. Usamos el término *software de enrutamiento* para describir el programa que reconfigura las tablas de reenvío en forma automática.

La manera más sencilla de pensar en el cálculo de las rutas de una WAN es considerar un gráfico que modela la red e imaginar un software que usa este gráfico para calcular la ruta más corta hacia todos los posibles destinos. Cada *nodo* del gráfico corresponde a un conmutador de paquetes en la red (las computadoras individuales no son parte del gráfico). Si la red contiene una conexión directa entre un par de conmutadores de paquetes, el gráfico contiene un *extremo* o un *enlace* entre los nodos correspondientes.[†] La figura 18.6 muestra una WAN de ejemplo y el gráfico correspondiente.

[†] Como la relación entre la teoría del gráfico y las redes de computadoras es sólida, a menudo escuchamos que a un conmutador de paquetes se le llama *nodo de red* y que a un circuito de datos entre dos sitios se le llama *enlace*.

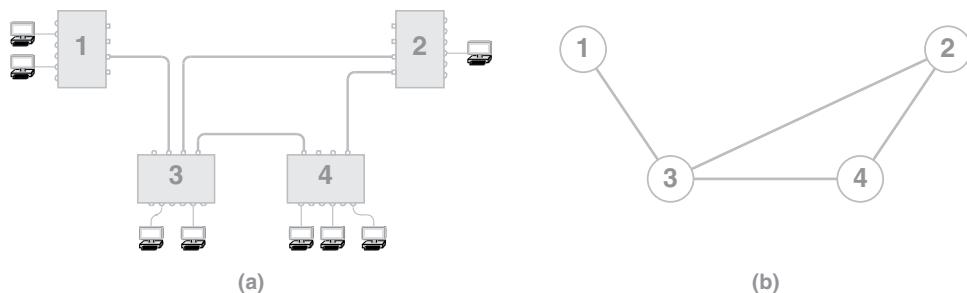


Figura 18.6 Ilustración de una WAN y el gráfico correspondiente.

Como se muestra en la figura, los nodos en el gráfico reciben una etiqueta, que es la misma que el número asignado al comutador de paquete correspondiente. Gracias a que se ha estudiado la teoría de los gráficos y se han desarrollado algoritmos eficientes, la representación de un gráfico resulta especialmente útil para calcular el reenvío del siguiente salto. Además, un gráfico abstrae los detalles y permite al software de enrute lidiar con la esencia del problema.

Cuando calcula el reenvío del siguiente salto para un gráfico, el algoritmo de enrutamiento debe identificar un enlace. Nuestros ejemplos usarán la notación (k,j) para denotar un enlace del nodo k al nodo j . De esta forma, cuando se ejecuta un algoritmo de enrutamiento en el gráfico de la figura 18.6(b), éste produce la salida como se muestra en la figura 18.7.

para llegar a	siguiente salto						
1	-	1	(2,3)	1	(3,1)	1	(4,3)
2	(1,3)	2	-	2	(3,2)	2	(4,2)
3	(1,3)	3	(2,3)	3	-	3	(4,3)
4	(1,3)	4	(2,4)	4	(3,4)	4	-
<i>Nodo 1</i>		<i>Nodo 2</i>		<i>Nodo 3</i>		<i>Nodo 4</i>	

Figura 18.7 Una tabla de reenvío para cada nodo en el gráfico de la figura 18.6(b).

18.10 Rutas predeterminadas

La tabla de reenvío para el nodo *I* de la figura 18.7 plantea una cuestión importante: una tabla de reenvío puede contener muchas entradas que apunten al mismo siguiente salto. Un análisis de la WAN en la figura 18.6(a) revela por qué todas las entradas remotas contienen el mismo siguiente salto. El commutador de paquetes tiene sólo una conexión a la red. Por lo tanto, todo el tráfico saliente debe enviarse a través de la misma conexión. En consecuencia, excepto por la entrada que corresponde al nodo en sí,

todas las entradas en la tabla de reenvío del nodo 1 tienen un siguiente salto que apunta al enlace que va del nodo 1 al nodo 3.

En nuestro ejemplo, la lista de entradas duplicadas en la tabla de reenvío es corta. Sin embargo, una WAN extensa puede contener cientos de entradas duplicadas. La mayoría de los sistemas de WAN incluyen un mecanismo que puede usarse para eliminar el caso común de entradas duplicadas. Conocido como *ruta predeterminada*, el mecanismo permite que una sola entrada de una tabla de reenvío reemplace a una larga lista de entradas que tengan el mismo valor del siguiente salto. Sólo se permite una entrada predeterminada en una tabla de reenvío, y esta entrada tiene menor prioridad que las otras entradas. Si el mecanismo de reenvío no encuentra una entrada explícita para un destino dado, usa el valor predeterminado. La figura 18.8 muestra las tablas de reenvío de la figura 18.7 revisadas para usar una ruta predeterminada.

| para siguiente
llegar a salto |
|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| 1 – | 2 – | 1 (3,1) | 2 (4,2) |
| * (1,3) | 4 (2,4) | 2 (3,2) | 4 – |
| | * | 3 – | * |
| | | 4 (3,4) | (4,3) |

Nodo 1 *Nodo 2* *Nodo 3* *Nodo 4*

Figura 18.8 Las tablas de reenvío de la figura 18.7 con rutas predeterminadas denotadas por un asterisco.

El enrutamiento predeterminado es opcional ya que una entrada predeterminada está presente sólo cuando más de un destino tiene el mismo valor del siguiente salto. Por ejemplo, la tabla de reenvío para el nodo 3 no contiene una ruta predeterminada debido a que cada entrada tiene un siguiente salto único. Sin embargo, la tabla de reenvío para el nodo 1 se beneficia de una ruta predeterminada debido a que todos los destinos remotos tienen el mismo siguiente salto.

18.11 Cálculo de la tabla de reenvío

¿Cómo se construye una tabla de reenvío? Hay dos metodologías básicas.

- *Enrutamiento estático*. Un programa calcula e instala rutas cuando se inicia un conmutador de paquetes; las rutas no cambian.
- *Enrutamiento dinámico*. Un programa crea una tabla de reenvío inicial cuando se inicia un conmutador de paquetes; después, el programa altera la tabla conforme cambian las condiciones en la red.

Cada metodología tiene ventajas y desventajas. Las principales ventajas del enrutamiento estático son la simpleza y el nivel bajo de sobrecarga. La principal desventaja es la inflexibilidad: las rutas estáticas no pueden cambiarse cuando se interrumpe la comunicación. Puesto que las redes grandes se diseñan con conexiones redundantes para manejar fallas de hardware ocasionales, la mayoría de las redes WAN usan una forma de enrutamiento dinámico.

18.12 Cálculo de ruta distribuido

Hay un algoritmo reconocido que puede calcular las rutas más cortas en un gráfico. Éste se muestra en el algoritmo 18.2. Sin embargo, en la práctica se necesita una versión distribuida de este cálculo. Es decir, en vez de que un programa centralizado calcule todas las rutas más cortas, cada commutador de paquetes debe calcular su propia tabla de reenvío en forma local. En consecuencia, se necesita un *cálculo de ruta distribuido*.

Hay que seguir dos metodologías generales para un cálculo de ruta distribuido, y veremos que se utilizan ambas:

- Enrutamiento de estado de enlace (LSR), que usa el algoritmo de Dijkstra
- Enrutamiento de vector de distancias (DVR), que usa otra metodología

Las siguientes secciones describen cada una de las dos metodologías. El capítulo 26 explica cómo se usa cada metodología para controlar las rutas en Internet.

18.12.1 Enrutamiento de estado del enlace (LSR)

Conocido formalmente como *enrutamiento de estado del enlace*, la metodología se conoce ahora como enrutamiento de la *ruta más corta primero* o *SPF*. La terminología surge debido a que caracteriza la forma en que funciona el algoritmo. Sin embargo, es algo engañoso debido a que todos los algoritmos buscan las rutas más cortas.

Para usar el enrutamiento LSR, los commutadores de paquetes envían periódicamente mensajes a través de la red, los cuales transportan el estado de un enlace entre dos commutadores de paquetes. Por ejemplo, los commutadores de paquetes 5 y 9 miden el enlace entre ambos y envían un mensaje de estado como “el enlace entre 5 y 9 está activo”. Cada mensaje de estado se difunde a todos los commutadores. Cada commutador ejecuta software que recolecta los mensajes de estado entrantes y los usa para crear un gráfico de la red. Después cada commutador usa el algoritmo 18.2 para producir una tabla de reenvío, eligiéndose a sí mismo como el origen para el cálculo.

Un algoritmo LSR puede adaptarse a fallas de hardware. Si falla un enlace entre los commutadores de paquetes, los commutadores conectados detectarán la falla y transmitirán un mensaje de estado que especifique que el enlace está inactivo. Todos los commutadores de paquetes reciben la difusión, cambian su copia del gráfico para reflejar el cambio en el estado del enlace y recalculan las rutas más cortas. De manera similar, cuando un enlace está disponible de nuevo, los commutadores de paquetes conectados al enlace detectan que está funcionando y comienzan a enviar mensajes de estado que reportan su disponibilidad.

Algoritmo 18.2

Dado:

Un gráfico con un peso no negativo asignado a cada extremo
y a un nodo de origen designado

Se calcula:

La distancia más corta desde el nodo de origen hasta cada uno
de los otros nodos y una tabla de enrutamiento del siguiente salto

Método:

Iniciar el conjunto S para que contenga todos los nodos excepto el de origen;

Iniciar el arreglo D de modo que $D[v]$ sea el peso del extremo

desde el origen hasta v, si existe dicho extremo, e *infinito* en caso contrario;

Iniciar las entradas de R, de modo que a $R[v]$ se le asigne v si

existe un extremo desde el origen hasta v, y cero en caso contrario;

while (conjunto S no está vacío) {

 elegir un nodo u de S, de tal forma que $D[u]$ sea mínimo;

 if ($D[u]$ es *infinito*) {

 error; no existe ruta hacia los nodos en S; salir;

 }

 eliminar u del conjunto S;

 for each nodo v tal que (u,v) sea un extremo {

 if (v aún está en S) {

 c = $D[u] + \text{peso}(u,v)$;

 if ($c < D[v]$) {

$R[v] = R[u]$;

$D[v] = c$;

 }

 }

 }

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

18.12.2 Enrutamiento de vector de distancias (DVR)

La principal alternativa a LSR se conoce como el método del *enrutamiento de vector de distancias* (*DVR*). Al igual que con LSR, a cada enlace de la red se le asigna un peso, y la *distancia* a un destino entre dos comutadores de paquetes se define como la suma de los pesos a lo largo de la ruta entre ambos. Al igual que LSR, el enrutamiento de vector de distancias hace que los comutadores de paquetes intercambien mensajes en forma periódica. Sin embargo y a diferencia del LSR, un esquema de vector de distancias hace que un comutador de paquetes envíe una lista completa de los destinos y el costo actual de llegar a cada uno. En esencia, cuando envía un mensaje DVR, un comutador de paquetes envía una serie de declaraciones individuales, de la forma:

“Puedo llegar al destino X y su distancia actual desde mi posición es Y.”

Los mensajes DVR no se difunden, sino que cada comutador de paquetes envía en forma periódica un mensaje DVR a sus vecinos. Cada mensaje contiene pares de (*destino, distancia*). Así, cada comutador de paquetes debe mantener una lista de destinos posibles, junto con la distancia actual hasta el destino y el siguiente salto a usar. Tanto la lista de destinos como el siguiente salto para cada uno, pueden encontrarse en la tabla de reenvío. Podemos pensar que el software DVR conserva una extensión de la tabla de reenvío que almacena la *distancia* para cada destino.

Cuando llega un mensaje a un comutador de paquetes del vecino *N*, el comutador analiza cada elemento del mensaje y, si el vecino tiene una ruta más corta hasta cierto destino que la ruta que se usa actualmente, entonces cambia su tabla de reenvío. Por ejemplo, si el vecino *N* anuncia una ruta al destino *D* con un costo de cinco y la ruta actual a través del vecino *K* tiene un costo de cien, el siguiente salto actual para *D* se reemplazará por *N*, y el costo para llegar a *D* será de cinco más el costo para llegar a *N*. El algoritmo 18.3 especifica cómo se actualizan las rutas al usar el método del vector de distancias.

Una de las diferencias clave entre los métodos LSR y DVR está en el momento en que se realiza el cálculo. En un algoritmo LSR, todos los nodos en la red se enteran al mismo tiempo del problema. Esto se debe a que como la información se difunde, el único retraso es el tiempo requerido para transmitir un mensaje. Sin embargo, en un algoritmo DVR cada comutador de paquetes realiza un cálculo antes de enviar la información a otro comutador. Por lo tanto, podemos imaginar que un par de comutadores de paquetes detectan la falla de un enlace, calculan nuevas tablas de reenvío y luego envían la información a sus vecinos. Cada vecino debe calcular una nueva tabla de reenvío antes de enviar la información al siguiente conjunto de vecinos. Por consiguiente, a un algoritmo DVR le lleva más tiempo notificar a todos los comutadores de paquetes después de que ocurre un error.

Algoritmo 18.3

Dada:

Una tabla de reenvío local con una distancia para cada entrada, una distancia para llegar a cada vecino y un mensaje DV entrante de un vecino

Se calcula:

Una tabla de reenvío actualizada

Método:

Mantener un campo *distancia* en cada entrada de la tabla de reenvío;

Inicializar la tabla de reenvío con una sola entrada que tenga el

destino igual al comutador de paquetes local, el *siguiente salto* sin usar y la *distancia* establecida en cero;

Repetir indefinidamente {

Esperar a que llegue un mensaje de enrutamiento a través de la

red proveniente de un vecino; dejar que el emisor sea el comutador *N*;

for each entrada en el mensaje {

Dejar que *V* sea el destino en la entrada y que *D* sea
la distancia;

Calcular *C* como *D* más el peso asignado al enlace a través
del cual llegó el mensaje;

Examinar y actualizar la tabla de enrutamiento local:

if (no existe ruta a *V*) {

agregar una entrada a la tabla de enrutamiento local para
el destino *V* con el siguiente salto *N* y la distancia *C*;

} else if (existe una ruta que tenga el siguiente salto *N*) {

reemplazar la distancia en la ruta existente con *C*;

} else if (existe una ruta con una distancia mayor a *C*) {

cambiar el siguiente salto a *N* y la distancia a *C*;

}

}

Algoritmo 18.3 Algoritmo de vector de distancias para el cálculo de rutas.

18.13 Rutas más cortas y pesos

Cabe recordar que un algoritmo LSR usa mensajes de estado de enlace para propagar la información sobre el estado de los enlaces, luego requiere que cada conmutador de paquetes recolecte los mensajes y cree un gráfico, y finalmente hace que cada conmutador de paquetes ejecute el *algoritmo de Dijkstra*[†] para encontrar la ruta más corta desde un nodo de origen hasta cada uno de los otros nodos del gráfico. A medida que el algoritmo calcula las rutas más cortas, se construye una tabla de reenvío del siguiente salto.

El algoritmo de Dijkstra es popular debido a que puede usarse con varias definiciones de la ruta “más corta”. En especial, el algoritmo no requiere que los valores en los extremos del gráfico representen la distancia geográfica, sino que permite asignar a cada extremo un valor no negativo conocido como *peso*, y define la distancia entre dos nodos como la suma de los pesos a lo largo de una ruta entre los nodos. El punto importante es:

Puesto que usa los pesos en los enlaces para calcular las rutas más cortas, el algoritmo de Dijkstra puede usarse con otras mediciones aparte de la distancia geográfica.

La figura 18.9 ilustra el concepto de los pesos mediante un gráfico de ejemplo, donde se asigna un peso entero a cada extremo y una ruta del menor peso entre dos nodos del gráfico.

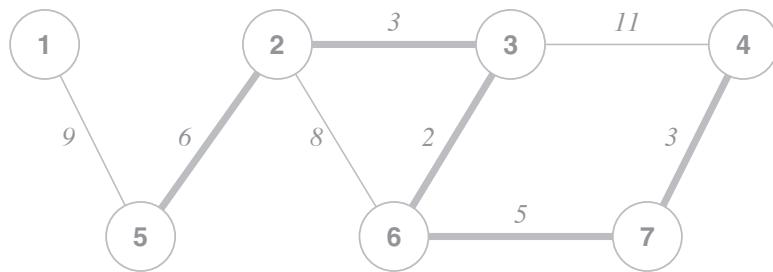


Figura 18.9 Un gráfico de ejemplo con un peso asignado a cada extremo; la ruta más corta entre los nodos 4 y 5 se muestra resaltada.

Para entender el algoritmo de Dijkstra, imagine construir las rutas más cortas un paso a la vez. El algoritmo mantiene un conjunto de nodos S para los que aún no se ha calculado la distancia mínima ni el siguiente salto. El conjunto se inicializa con todos los nodos excepto con el de origen. Después, el algoritmo itera hasta que el conjunto S esté vacío. En cada iteración el algoritmo extrae el nodo de S que tenga la distancia más corta desde el origen. Al eliminar el nodo u , el algoritmo analiza la distancia actual desde el origen hasta cada uno de los vecinos de u que aún se encuentren en el conjunto. Si una

[†] El algoritmo se nombró en honor a su inventor, E. Dijkstra; su uso en el enruteamiento se presenta en el algoritmo 18.2 de la página 317.

ruta del origen hasta el vecino pasa por u y tiene menos peso que la ruta actual, el algoritmo actualiza la distancia hasta el vecino. Una vez que se extraigan todos los nodos de S , el algoritmo habrá calculado la distancia mínima hasta cada nodo y la tabla de reenvío del siguiente salto correcta para todas las rutas posibles.

La implementación del algoritmo de Dijkstra es simple. Además de la estructura de datos utilizada para almacenar información sobre el gráfico, el algoritmo de Dijkstra necesita almacenar tres estructuras de datos: la distancia actual a cada nodo, el siguiente salto para la ruta más corta y la información sobre el conjunto de nodos restantes. Los nodos pueden enumerarse de 1 a n como se demuestra en la figura 18.9, lo cual hace a la implementación eficiente debido a que un número de nodo puede usarse como índice para una estructura de datos. En especial, el algoritmo puede usar dos arreglos D y R , cada uno de los cuales se indexa con base en el número de nodo. La $i^{\text{ésima}}$ entrada en el arreglo D almacena el valor actual de la mínima distancia desde el origen hasta el nodo i . La $i^{\text{ésima}}$ entrada en el arreglo R almacena el siguiente salto a usar para llegar al nodo i a lo largo de la ruta que se está calculando. El conjunto S puede mantenerse como una lista doblemente enlazada de números de nodo, lo cual facilita la búsqueda en todo el conjunto o la eliminación de una entrada.

En el algoritmo 18.2, $\text{peso}(u,v)$ es una función que devuelve el peso del extremo del nodo u al nodo v . Se asume que la función peso devuelve el valor *infinito* si no existe un extremo del nodo u al nodo v . En la práctica puede usarse cualquier valor para representar infinito, siempre y cuando sea mayor que la suma de los pesos a lo largo de cualquiera de las rutas en el gráfico. Una forma de generar el valor *infinito* consiste en agregar uno a la suma de todos los pesos en todos los extremos.

Permitir la asignación de pesos arbitrarios a los extremos de un gráfico significa que es posible usar un algoritmo con distintas mediciones de distancia. Por ejemplo, algunas tecnologías de WAN miden la distancia contando el número de comutadores de paquetes a lo largo de una ruta. Para usar el algoritmo para dichas tecnologías, a cada extremo del gráfico se le asigna un peso de 1 . En otras tecnologías de WAN, los pesos se asignan de acuerdo con la capacidad de las conexiones involucradas. Como alternativa, un administrador puede asignar pesos a los enlaces para controlar el enrutamiento de acuerdo con una política de prioridades específica. Por ejemplo, considere un caso en el que existen dos rutas separadas entre un par de comutadores de paquetes, con una ruta designada como la *primaria* y la otra designada como una *ruta de respaldo*. Para cumplir con esa política, un administrador puede asignar al enlace primario un peso bajo y al otro enlace un peso alto. El software de enrutamiento configurará las tablas de reenvío para usar la ruta con el menor peso a menos que no esté disponible, en cuyo caso seleccionará la ruta alternativa.

18.14 Problemas de enrutamiento

En teoría, cualquiera de los enrutamientos LSD o DVR calculará correctamente las rutas más cortas. Además, cada método *convergerá* en un momento dado, lo que significa que las tablas de reenvío en todos los comutadores de paquetes estarán de acuerdo. Sin embargo, puede haber problemas. Por ejemplo, si se pierden los mensajes LSR, dos comutadores de paquetes podrían estar en desacuerdo en cuanto a la ruta más corta. Los problemas del DVR pueden ser más graves, ya que la falla de un enlace puede provocar que dos o más comutadores de paquetes creen un *bucle de enrutamiento*, en el que cada comutador de paquetes piensa que el siguiente comutador de paquetes en el conjunto es la ruta más corta hacia un destino específico. Como resultado, el paquete puede circular indefinidamente entre ambos comutadores de paquetes.

Una de las principales razones por las que los protocolos DVR presentan problemas se debe al retrolavado (es decir, cuando un comutador de paquetes recibe la información que envía). Por ejemplo, suponga que un comutador dice a sus vecinos: “Puedo llegar al destino D1 a un costo de 3”. Si la conexión que conduce al destino D1 falla, el comutador eliminará de su tabla de reenvío toda la entrada para D1 (o marcará la entrada como inválida). Pero el comutador dijo a sus vecinos que existía una ruta. Imagine que justo después de que falla el enlace, uno de los vecinos envía un mensaje DVR que especifica: “Puedo llegar al destino D1 a un costo de 4”. Desafortunadamente, los vecinos tomarán el mensaje como cierto y se creará un bucle de enrutamiento.

La mayoría de los mecanismos de enrutamiento contienen restricciones y procedimientos para evitar problemas como los bucles de enrutamiento. Por ejemplo, los esquemas DVR emplean el horizonte dividido, que especifica que un sistema no envía información de regreso a su origen. Además, la mayoría de los sistemas de enrutamiento prácticos utilizan la histéresis para evitar que el software realice muchos cambios en poco tiempo. Sin embargo, en una red grande donde muchos enlaces fallan y se recuperan con frecuencia, pueden ocurrir problemas de enrutamiento.

18.15 Resumen

Es posible usar una tecnología de red de área amplia (WAN) para formar redes que abarquen una distancia extensa y conectar muchas computadoras de manera arbitraria. Una WAN tradicional consiste en dispositivos electrónicos conocidos como comutadores de paquetes, los cuales se interconectan mediante circuitos de datos rentados. Un comutador de paquetes contiene un procesador, una memoria e interfaces de E/S. Las interfaces se conectan a una computadora local o a otro comutador de paquetes.

Las redes de conmutación de paquetes usan la metodología de almacenar y reenviar, en la cual un paquete entrante se coloca en la memoria de un comutador de paquetes hasta que el procesador pueda reenviar el paquete a su destino. El reenvío se basa en una estructura de datos conocida como tabla de reenvío. La tabla contiene una entrada para cada destino y la entrada especifica el siguiente salto que se utilizará para llegar a ese destino. Una tabla de reenvío enumera los comutadores de paquetes como destinos en vez de computadoras individuales.

Una WAN puede representarse como un gráfico en el que cada nodo corresponde a un comutador de paquetes y cada extremo corresponde a una línea de comunicación. La representación del gráfico es útil debido a que elimina los detalles y puede usarse para calcular tablas de reenvío. Los dos métodos básicos que se usan en el software de enrutamiento son el *enrutamiento de estado del enlace* (LSR) y el *enrutamiento de vector de distancias* (DVR). El LSR establece que cada comutador de paquetes debe difundir el estado de cada enlace conectado directamente y usa el algoritmo de Dijkstra para calcular las rutas más cortas. El DVR establece que un comutador de paquetes debe enviar a sus vecinos una lista de destinos y el costo para llegar a cada una. Cada vecino analiza la lista en un mensaje DVR entrante y, si hay una ruta de menor costo, reemplaza los elementos en su tabla de reenvío.

EJERCICIOS

- 18.1** ¿Cuáles son las partes conceptuales de un conmutador de paquetes tradicional y a qué se conecta?
- 18.2** ¿Cuáles son las dos piezas en las que conceptualmente se divide un conmutador de paquetes moderno?
- 18.3** ¿Puede una computadora usar una interfaz Ethernet para comunicarse con una WAN? Explique.
- 18.4** Si una WAN conecta N sitios, ¿cuál es el número mínimo de circuitos digitales necesarios?
- 18.5** Explique el paradigma de almacenar y reenviar.
- 18.6** ¿Cuáles son las dos partes conceptuales de una dirección WAN?
- 18.7** La figura 18.4 muestra cómo pueden asignarse direcciones a las computadoras que se conectan a un conmutador de paquetes. Suponga que el hardware de una de las interfaces de un conmutador falla y que un administrador de red mueve la conexión de una computadora a una interfaz sin usar. ¿Funcionará correctamente la nueva configuración? ¿Por qué sí o por qué no?
- 18.8** Escriba un programa de computadora que reciba como entrada una tabla de reenvío y una serie de paquetes, y que genere como salida una instrucción sobre cómo debe reenviarse cada paquete. Recuerde hacerse cargo de los paquetes que tengan una dirección incorrecta.
- 18.9** Considere una WAN con dos conmutadores de paquetes. Suponga que cada conmutador tiene una entrada para cada dirección local en la tabla de reenvío (es decir, la dirección de cada computadora conectada al conmutador) más una entrada predeterminada que apunta al otro conmutador. ¿Bajo qué circunstancias funcionará el esquema? ¿Bajo qué circunstancias fallará?
- 18.10** ¿Qué beneficios ofrece el enrutamiento dinámico?
- 18.11** Escriba un programa de computadora que implemente el algoritmo de Dijkstra para encontrar las rutas más cortas en un gráfico.
- 18.12** ¿Cuáles son los dos métodos básicos que se utilizan para realizar un cálculo de ruta distribuido y cómo funciona cada uno?
- 18.13** Cuando los programas que se ejecutan en dos conmutadores de paquetes intercambian información de vectores de distancias, éstos deben acordar un formato de mensaje. Cree una especificación para un formato de mensaje sin ambigüedades. (Sugerencia: considere las diferencias en la forma en que las computadoras representan la información).
- 18.14** Extienda el ejercicio anterior implementando un programa de computadora que use el formato de mensaje especificado. Haga que otro estudiante implemente un programa a partir de la misma especificación y vea si operan correctamente entre sí.
- 18.15** Cuando un conmutador de paquetes recibe un mensaje de un vector de distancias de un vecino, ¿cambiará siempre la tabla de reenvío del conmutador? Explique.
- 18.16** ¿Qué es un bucle de enrutamiento?

Contenido del capítulo

- 19.1 Introducción, 325
- 19.2 Tecnologías de conexión y acceso, 325
- 19.3 Tecnologías de LAN, 327
- 19.4 Tecnologías de WAN, 328
- 19.5 Resumen, 332

19

Tecnologías de redes pasadas y presentes

19.1 Introducción

Los capítulos anteriores analizan las comunicaciones y las redes de datos considerando sus taxonomías básicas. Los primeros capítulos consideran la división entre las tecnologías utilizadas para el acceso a Internet y las que se usan en el núcleo de Internet. Los capítulos de esta parte del libro usan la taxonomía clásica de las redes LAN, MAN y WAN para describir las redes alámbricas e inalámbricas.

A través de los años, se han definido muchas tecnologías de redes para cada tipo básico. Algunas que tuvieron gran importancia en un momento dado ya se desvanecieron en la oscuridad y otras siguen llenando un nicho. Este breve capítulo sintetiza las tecnologías que han tenido un impacto duradero en el campo, ya sea debido a que la tecnología sigue en uso o porque los profesionales de redes usan el nombre como referencia para una idea. Este capítulo sólo brinda un resumen rápido; a lo largo del libro encontrará más información sobre muchas de estas tecnologías.

19.2 Tecnologías de conexión y acceso

Los primeros capítulos describen las tecnologías de conexión y acceso más importantes (como DSL y los módems de cable). Se ha creado una variedad de tecnologías adicionales, incluyendo una tecnología que transmite datos a través de líneas de alimentación eléctrica y mecanismos de acceso inalámbricos. El conjunto de tecnologías puede resumirse de la siguiente manera:

19.2.1 Red óptica síncrona o jerarquía digital (SONET/SDH)

SONET y la jerarquía TDM relacionada se diseñaron originalmente como un sistema para transportar llamadas telefónicas de voz digitales. La tecnología se convirtió en el estándar para los circuitos digitales que se usan en Internet. SONET permite construir un anillo físico con el objetivo de brindar redundancia. El hardware puede detectar y corregir problemas automáticamente; aun cuando se dañe una parte del anillo, los datos podrán transmitirse. Para conectar un sitio a un anillo SONET, se utiliza un dispositivo conocido como *multiplexor de extracción-inserción*. El término surge debido a que el multiplexor de extracción-inserción inserta o termina un conjunto de circuitos de datos, cada uno de los cuales se conecta a otro multiplexor de extracción-inserción del anillo. SONET usa la multiplexación por división de tiempo para multiplexar los circuitos de la fibra. SDH brinda los estándares para los circuitos ya reconocidos, tales como un circuito T3 que puede configurarse a través de un anillo SONET.

19.2.2 Circuitos de portadora óptica (OC)

Los estándares OC especifican la señalización que se utiliza en un anillo SONET de fibra óptica. Los estándares OC están asociados con velocidades de datos más altas que los estándares de la serie T proporcionados por SDH. Una compañía privada podría optar por rentar un circuito OC para conectar dos de sus sitios. Los ISP de nivel 1 usan circuitos de OC-192 (10 Mbps) y OC-768 (40 Mbps) en la red troncal de Internet.

19.2.3 Línea de suscriptor digital (DSL) y módems de cable

Estas dos tecnologías ofrecen actualmente los principales medios para proporcionar acceso a Internet de banda ancha a residencias y negocios pequeños. DSL hace uso de las líneas telefónicas fijas existentes, y la tecnología de los módems de cable usa la infraestructura existente de la televisión por cable. DSL ofrece velocidades de datos de 1 a 6 Mbps, dependiendo de la distancia entre una oficina central y un suscriptor; los módems de cable ofrecen hasta 52 Mbps pero el ancho de banda se comparte entre un conjunto de usuarios. Ambas tecnologías se consideran transitorias hasta que la fibra óptica esté disponible para llegar hasta el hogar.

19.2.4 Wi-Fi y WiMAX

Wi-Fi constituye un conjunto de tecnologías inalámbricas que se han vuelto bastante populares para ofrecer acceso a Internet en casas, cafés, aeropuertos, hoteles y otras ubicaciones. Las generaciones sucesivas de tecnologías Wi-Fi tienen mayores velocidades de datos.

WiMAX es una tecnología inalámbrica emergente que puede usarse para formar una MAN. WiMAX proporciona capacidades de acceso o de red de retorno,[†] y se definen dos versiones para dar soporte a los puntos finales fijos y móviles.

[†] La red de retorno comprende las conexiones que van de una ubicación remota o punto de acceso y de regreso a la instalación central de un proveedor.

19.2.5 Satélite de apertura muy pequeña (VSAT)

Las tecnologías de VSAT cuyo tamaño de plato es menor a 3 metros, han hecho posible el uso de satélites para proporcionar acceso a Internet a individuos o pequeños negocios. Aunque proporciona velocidades de datos elevadas, VSAT sufre de retrasos extensos.

19.2.6 Comunicación a través de la línea de alimentación (PLC)

La tecnología PLC usa altas frecuencias para enviar datos a través de las líneas de alimentación eléctrica. La idea es usar la infraestructura existente para ofrecer acceso a Internet. Aunque se ha realizado mucha investigación, la tecnología no ha disfrutado de una implementación extensa.

19.3 Tecnologías de LAN

19.3.1 Token Ring

Después de que se inventaron las redes LAN, muchos grupos propusieron diseños o crearon prototipos experimentales, y el desarrollo de las tecnologías de LAN continuó durante veinte años antes de que Ethernet llegara a dominar. Parte de los primeros trabajos en las LAN exploraron el *paso de señales token* como un mecanismo de control de acceso, e IBM optó por crear una tecnología de LAN de paso de token que se conoció como *Token Ring*. A pesar de su velocidad de datos menor (en un principio de 4 Mbps y con el paso del tiempo hasta 16 Mbps) y de su alto costo, los departamentos corporativos de tecnología de la información aceptaron ampliamente la tecnología Token Ring de IBM, y se convirtió en una tecnología de LAN importante durante muchos años.

19.3.2 Interconexión de datos distribuidos por fibra y cobre (FDDI y CDDI)

A finales de 1980, las dos tecnologías principales de LAN (Ethernet a 10 Mbps y Token Ring de IBM a 16 Mbps) tenían velocidades de datos aparentemente insuficientes para cumplir con la demanda creciente. Se creó el estándar FDDI para incrementar las velocidades de datos de LAN a 100 Mbps. En ese entonces, para lograr mayores velocidades de datos, se requería el uso de fibra óptica en vez de cableado de cobre. FDDI usaba un par de *anillos de rotación inversa* para ofrecer redundancia; si se cortaba un anillo FDDI, el hardware se encargaba automáticamente de recalcular la ruta de datos para enrutar el tráfico alrededor de la falla y mantener el anillo activo. FDDI introdujo uno de los primeros comutadores de LAN en los que cada computadora se conectaba directamente a un mecanismo FDDI central. Por lo tanto, FDDI tenía una topología física de estrella y una topología lógica de anillo.

Puesto que ofrecía la velocidad de datos más alta disponible en ese momento y la posibilidad de redundancia, FDDI se hizo popular como una interconexión de alta velocidad entre las computadoras de un centro de datos. Sin embargo, el elevado costo y la experiencia necesaria para instalar la fibra desanimó a la mayoría de las organizaciones a la hora de reemplazar el cableado de cobre. Para competir con la Ethernet más rápida se creó una versión de FDDI que operaba a través de cableado de cobre (CDDI). En última instancia Ethernet demostró tener un costo más bajo y las tecnologías FDDI desaparecieron.

19.3.3 Ethernet

En cierto sentido, Ethernet ganó la carrera; las tecnologías de Ethernet dominan completamente el mercado de las redes LAN. Sin duda se implementan más redes Ethernet que cualquier otro tipo de LAN. En otro sentido, la red Ethernet original desapareció y se reemplazó por una nueva tecnología que aún se conoce como *Ethernet*. Podemos observar, por ejemplo, que casi no hay similitud entre el pesado cable coaxial y la señalización RF utilizada en las primeras redes Ethernet, y el cableado y la señalización que se utilizan con una red Gigabit Ethernet. Además de los cambios en la velocidad de datos, las topologías física y lógica evolucionaron también. Los concentradores reemplazaron a los cables, los conmutadores de Ethernet reemplazaron a los concentradores y los conmutadores de VLAN reemplazaron a los conmutadores convencionales.

19.4 Tecnologías de WAN

Se han creado muchas tecnologías para uso experimental y de producción en las redes de área amplia. En esta sección resumimos unos cuantos ejemplos que ilustran parte de la diversidad.

19.4.1 ARPANET

Las redes WAN de conmutación de paquetes tienen menos de cincuenta años de antigüedad. A finales de la década de 1960, la *Agencia de proyectos de investigación avanzada (ARPA)* patrocinó la investigación en redes para el Departamento de defensa de Estados Unidos. Un proyecto de investigación importante de la ARPA desarrolló una red de área amplia para determinar si la tecnología de conmutación de paquetes sería valiosa para los militares. Conocida como *ARPANET*, la red fue una de las primeras WAN con conmutación de paquetes. La ARPANET conectó a los investigadores académicos y de la industria. Aunque según los estándares actuales ARPANET era lenta (se rentaban líneas de datos seriales que conectaban conmutadores de paquetes operando a tan sólo 56 Kbps), el proyecto dejó un legado de conceptos, algoritmos y terminología que aún se utilizan.

Cuando comenzó el proyecto de Internet, ARPANET era la WAN sobre la cual los investigadores se comunicaban y experimentaban. En enero de 1983, ARPA ordenó a todos los que estaban conectados a ARPANET que dejaran de usar los protocolos originales y comenzaran a usar los protocolos de Internet. De esta forma, ARPANET se convirtió en la primera red troncal de Internet.

19.4.2 X.25

La organización que establece los estándares telefónicos internacionales, la *Unión internacional de telecomunicaciones (ITU)*, desarrolló uno de los primeros estándares para la tecnología WAN que se hizo popular entre portadoras públicas. En ese tiempo la ITU se conocía como *Comité consultivo internacional telegráfico y telefónico (CCITT)*, y el estándar se sigue conociendo como *CCITT X.25*. Aunque los bancos comerciales y otros de los primeros usuarios de la red adoptaron el estándar X.25, esta tecnología disfrutó de una mayor popularidad en Europa que en Estados Unidos.

El estándar X.25 usaba un diseño de WAN tradicional. Una red X.25 consistía en dos o más commutadores de paquetes X.25 interconectados por líneas rentadas. Las computadoras se conectaban directamente a los commutadores de paquetes. X.25 usaba un paradigma orientado a la conexión similar a una llamada telefónica, en el que una computadora tenía que abrir una conexión antes de transferir los datos.

Puesto que X.25 se inventó antes de que las computadoras personales se hicieran populares, muchas de las primeras redes X.25 se diseñaron para conectar terminales ASCII a las computadoras remotas de tiempo compartido. Conforme un usuario introducía datos en un teclado, una interfaz de red X.25 capturaba las pulsaciones de tecla, colocaba cada pulsación en un paquete X.25 y transmitía los paquetes a través de la red. Cuando un programa ejecutándose en una computadora remota visualizaba los resultados, el estándar X.25 operaba en dirección inversa: la computadora pasaba el resultado a la interfaz de red X.25, la cual colocaba la información en paquetes X.25 para transmitirlos de vuelta a la pantalla del usuario. Aunque las compañías telefónicas impulsaban el uso de los servicios X.25, la tecnología era costosa para el rendimiento que ofrecía, por lo que se reemplazó por otras tecnologías de WAN.

19.4.3 Retransmisión de tramas

Los proveedores de servicios de larga distancia crearon una serie de tecnologías de red de área amplia que transportan datos. Uno de esos servicios, la *retransmisión de tramas* o *frame relay*, se diseñó para aceptar y entregar bloques de datos, donde cada bloque puede contener hasta 8K octetos de datos. Parte de la motivación para el gran tamaño de datos surge debido a que los inventores contemplaron usar el servicio de retransmisión de tramas como puente para los segmentos de LAN. Una organización con oficinas en dos ciudades podía obtener un servicio de retransmisión de tramas para cada oficina y luego usarlo para reenviar paquetes de un segmento de LAN en un sitio, a un segmento de LAN en el otro sitio. Los diseñadores eligieron un paradigma orientado a la conexión que fuera aceptable para las corporaciones que contaban con varias oficinas. Aunque hay unos cuantos sitios que aún utilizan la retransmisión de tramas, ésta se reemplazó en gran parte por alternativas de menor costo.

Como la retransmisión de tramas se diseñó para manejar datos de un segmento de LAN, los diseñadores contemplaron que la red funcionaría a velocidades entre 4 y 100 Mbps (la velocidad de las LAN cuando se creó la retransmisión de tramas). Sin embargo, en la práctica el alto costo del servicio de retransmisión de tramas condujo a muchos clientes a elegir conexiones más lentas que funcionaban a 1.5 Mbps o 56 Kbps.

19.4.4 Servicio de datos multimegabit commutados (SMDS)

Al igual que la retransmisión de tramas, SMDS es un servicio de datos de área amplia de alta velocidad que ofrecen los proveedores de servicios de larga distancia. Se basó en el estándar 802.6DQDB del IEEE y se considera precursor del modo de transferencia asíncrona (ATM). En vez del tráfico de voz, SMDS está diseñado para transportar datos. Lo que es más importante, SMDS está optimizado para operar a las velocidades más altas. Por ejemplo, la información de encabezado en los paquetes puede requerir una cantidad considerable del ancho de banda disponible. Para minimizar la sobrecarga de los encabezados, SMDS usa un encabezado pequeño y limita cada paquete para que contenga no más de 9188 octetos de datos. SMDS también define una interfaz especial de hardware que se utiliza para conectar computadoras a la red. En ese tiempo, la interfaz hizo posible entregar datos tan rápido como una computadora podía moverlos a la memoria.

Como su nombre lo implica, a menudo las redes SMDS operan a velocidades mayores de 1 Mbps (es decir, más rápido que una conexión común de retransmisión de tramas). Los dos servicios diferían en cuanto a la forma en que podían usarse. SMDS no requería conexión, lo cual le daba flexibilidad. Sin embargo, la mayoría de las compañías telefónicas se sentían más cómodas con las tecnologías orientadas a la conexión, lo que significó que SMDS no fue tan popular y se reemplazó.

19.4.5 Modo de transferencia asíncrona (ATM)

En la década de 1990, la industria de las telecomunicaciones diseñó el ATM como una alternativa para Internet y anunció su esfuerzo con bombo y platillos. Cuando surgió la tecnología ATM, tenía metas ambiciosas: los diseñadores afirmaban que reemplazaría todas las tecnologías de WAN y LAN y conduciría a un sistema de comunicaciones totalmente uniforme a nivel mundial. Además de los datos, ATM tiene funciones especiales para manejar la transmisión de video y las llamadas telefónicas de voz. También, los diseñadores aseguraban que el ATM escalaría a velocidades de datos mucho mayores que otras tecnologías de conmutación de paquetes.

La nueva idea clave introducida en ATM se conoce como *comutación de etiquetas*. ATM es una tecnología orientada a la conexión, pero los paquetes no contienen direcciones de destino como los paquetes tradicionales. En su defecto, un paquete (denominado *celda*) transporta un pequeño identificador conocido como *etiqueta*. Además, una etiqueta puede cambiarse cada vez que el paquete pasa a través de un conmutador. Cuando se establece una conexión, se elige una etiqueta única para cada enlace de la ruta y éstas se colocan en tablas contenidas en los conmutadores. Cuando llega un paquete, el conmutador busca la etiqueta actual y sustituye una etiqueta de reemplazo. En teoría, la conmutación de etiquetas puede realizarse en hardware a una mayor velocidad que el reenvío convencional.

Para dar cabida a todos los posibles usos, los diseñadores agregaron muchas características al ATM, incluyendo mecanismos para ofrecer garantías de extremo a extremo en cuanto al servicio (por ejemplo, ancho de banda garantizado y amortiguación de los retrasos). Cuando comenzaron a implementar el ATM, los ingenieros descubrieron que tantas características significaba que el hardware era complejo y costoso. Además, el mecanismo original elegido para configurar rutas conmutadas con etiquetas era tan torpe que no se usó. En consecuencia, no se aceptó el ATM y prácticamente desapareció.

19.4.6 Comutación de etiquetas multiprotocolo (MPLS) y túneles

Aunque no es un sistema de red en sí, MPLS es un resultado notable del esfuerzo del ATM en el que los ingenieros adaptaron la conmutación de etiquetas para usarla en enrutadores de Internet.[†] En vez de reemplazar por completo el hardware subyacente como ATM intentó hacerlo, MPLS se implementa como una función adicional en los enrutadores de Internet convencionales. Un enrutador MPLS acepta paquetes de Internet estándar, usa un conjunto de reglas para colocar cada paquete en una envoltura especial que contiene una etiqueta y envía el paquete resultante a través de una ruta conmutada con etiquetas. Es decir, cada uno de los enrutadores que están a lo largo de la ruta debe tener capacidad de MPLS y debe configurarse para usar la conmutación de etiquetas de modo que se seleccione un siguiente salto en vez del reenvío estándar de Internet. Una vez que un paquete llega al final de la ruta MPLS (es decir, cuando llega al último enrutador que entiende MPLS), el enrutador se configura para desenvolver el paquete y usar el reenvío de Internet para llegar al destino. La idea importante es que los enrutadores MPLS tam-

[†] El capítulo 20 describe la arquitectura y el enrutamiento de Internet.

bién tienen capacidades de reenvío de Internet, lo que significa que los enrutadores pueden configurarse para usar MPLS para cierto tráfico y reenvío normal para el resto del tráfico.

MPLS se usa mucho en el núcleo de Internet. Los ISP de nivel 1 (los ISP más grandes) usan MPLS para controlar las rutas que siguen los paquetes. De este modo, un ISP podría enviar tráfico Web a lo largo de una ruta y video de flujo continuo a través de otra ruta. Los ISP también ofrecen servicios MPLS a clientes individuales. Por ejemplo, un cliente grande podría pagar por que un ISP configure una ruta MPLS entre una oficina en la ciudad de Nueva York y una oficina en San Francisco. Al pagar por una ruta, un cliente recibe la garantía de que su tráfico tendrá prioridad y no competirá con el tráfico de los clientes que pagan menos.

MPLS popularizó el término *túnel* de red. En general, un túnel se define como la transmisión de tráfico que usa un protocolo de alto nivel sobre otro protocolo de alto nivel. En el caso de MPLS, un paquete de Internet se coloca dentro de una envoltura MPLS para la transmisión y esta envoltura se retira en el otro extremo de la ruta MPLS. La terminología surge debido a que el reenvío de Internet sólo necesita llevar el paquete al comienzo de la ruta MPLS y después desde el final de la ruta MPLS hasta el destino del paquete. Aunque MPLS puede enviar el paquete a través de muchos enrutadores, se hace cargo totalmente del reenvío a lo largo de la ruta y los detalles son invisibles para el software de reenvío de Internet. En consecuencia, los profesionales de redes dicen que el paquete de Internet entra en un “túnel” de MPLS y sale por el otro extremo.

19.4.7 Red digital de servicios integrados (ISDN)

Las compañías telefónicas crearon la tecnología ISDN para brindar servicios de red a una velocidad de datos más alta de la que podía obtenerse con un módem de marcación telefónica. Cuando se propuso por primera vez, 128 Kbps era una velocidad rápida. Para cuando estuvo disponible, la tecnología parecía lenta para el precio. En muchas partes del mundo, ISDN fue reemplazado por DSL, módems de cable o sistemas celulares 3G, todos los cuales ofrecen velocidades de datos mucho mayores.

19.4.8 Video y voz sobre IP (VoIP): SIP y H.323

Aunque se diseñó para transportar datos, Internet (en particular, el protocolo Internet) también puede usarse para transferir voz y video digitalizados. Por ejemplo, *Skype* usa software de aplicación para ofrecer llamadas de voz y video a través de la Internet estándar. La idea se conoce genéricamente como *voz sobre IP (VoIP)*; muchos distribuidores venden productos VoIP. Por ejemplo, algunos distribuidores ofrecen enrutadores que detectan el tráfico VoIP y le dan prioridad. Una empresa que use teléfonos VoIP en forma interna puede elegir esos enrutadores para asegurarse de que las llamadas telefónicas no se interrumpan debido a otros tipos de tráfico de datos.

La IETF y el ITU inventaron tecnologías que van más allá del simple hecho de transportar voz digitalizada y reemplazar el sistema de marcación telefónica con una infraestructura basada en paquetes. El sistema telefónico incluye características como reenvío de llamadas, correo de voz y contabilización de tarifas, por ello los esfuerzos son grandes y complejos. Un asunto particularmente espinoso se relaciona con la creación de una interfaz entre el sistema basado en paquetes y la red telefónica existente. Se crearon dos estándares para resolver el problema, y ambos se utilizan: el IETF creó el *protocolo de inicio de sesión (SIP)* y el ITU creó el *H.323*.

19.4.9 Redes definidas por software (SDN) y OpenFlow

El concepto de *redes definidas por software (SDN)* se creó en la década de 2000 y se ha vuelto extremadamente popular. La idea es separar el software de administración de redes de los dispositivos de red que se están administrando. Por ejemplo, una tecnología popular que se usa con las SDN, conocida como *OpenFlow*, hace que un controlador (por lo general, una PC con Linux) ejecute software de aplicación que configure el reenvío en un conmutador de Ethernet.

En esencia, en vez de depender del software de administración del distribuidor, SDN permite que los propietarios de equipo compran o creen software de administración de redes que funcione con equipos de cualquier distribuidor. Por consiguiente, el personal de TI no tendrá que aprender los comandos administrativos de cada distribuidor y la compañía podrá cambiar fácilmente de distribuidores de equipos. El capítulo 31 proporciona más información y ejemplos de esto.

19.5 Resumen

Se han creado diversas tecnologías de redes. Algunas eran demasiado complejas, otras muy costosas y algunas más carecían de características esenciales. Incluso después de lograr cierto éxito comercial, muchas fueron reemplazadas. En algunos casos la terminología persiste incluso después de que se haya desvanecido la tecnología. Lo irónico es que, aun cuando la tecnología de Ethernet ha sobrevivido por treinta años, sólo se conservó su nombre y el formato de trama; la tecnología usada actualmente ha cambiado por completo.

EJERCICIOS

- 19.1** ¿Qué es SONET?
- 19.2** ¿Con qué nombre conoce un consumidor a la tecnología DOCSIS?
- 19.3** ¿Qué tecnología esperaría que tuviera un menor retraso, la tecnología VSAT o WiMAX?
¿Por qué?
- 19.4** ¿Qué compañía es bien conocida por una tecnología Token Ring?
- 19.5** ¿Qué tecnología eclipsó y desplazó finalmente a FDDI?
- 19.6** ¿Qué tecnología reemplazó a los concentradores de Ethernet?
- 19.7** Nombre una tecnología de WAN que adoptó los protocolos de Internet en 1983.
- 19.8** ¿Qué tecnología de WAN usaron los bancos en la década de 1980?
- 19.9** ¿Qué significa ATM en el mundo de las redes?
- 19.10** Nombre una tecnología actual que surgió del ATM.
- 19.11** ¿Por qué ISDN no ganó un mercado grande?

PARTE IV

Interconexión de redes mediante el uso de TCP/IP

**Arquitectura de Internet,
direcciónamiento, vinculación,
encapsulamiento y protocolos
de la suite TCP/IP**

Capítulos

- 20 Interconexión de redes: conceptos, arquitectura y protocolos**
- 21 IP: direcciónamiento de Internet**
- 22 Reenvío de datagramas**
- 23 Protocolos y tecnologías de soporte**
- 24 UDP: servicio de transporte de datagramas**
- 25 TCP: servicio de transporte confiable**
- 26 Enrutamiento en Internet y protocolos de enrutamiento**

Contenido del capítulo

- 20.1 Introducción, 335
- 20.2 La motivación para la interconexión de redes, 335
- 20.3 El concepto de servicio universal, 336
- 20.4 Servicio universal en un mundo heterogéneo, 336
- 20.5 Interconexión de redes, 337
- 20.6 Conexión física de redes con enrutadores, 337
- 20.7 Arquitectura de Internet, 338
- 20.8 Intranet e Internet, 339
- 20.9 Obtención de un servicio universal, 339
- 20.10 Una red virtual, 339
- 20.11 Protocolos para la interconexión de redes, 341
- 20.12 Repaso de la distribución en capas de TCP/IP, 341
- 20.13 Computadoras host, enrutadores y capas de protocolos, 342
- 20.14 Resumen, 342

20

Interconexión de redes: conceptos, arquitectura y protocolos

20.1 Introducción

Los capítulos anteriores describen las redes básicas, incluyendo los componentes de hardware utilizados en las redes LAN y WAN, así como algunos conceptos generales como el direccionamiento y el enruteamiento. Este capítulo inicia el análisis de otra idea fundamental en la comunicación de computadoras: una tecnología de interconexión de redes que puede usarse para conectar varias redes físicas en un sistema de comunicaciones uniforme y extenso. El capítulo habla sobre la motivación para lograr la interconexión de redes, presenta los componentes de hardware utilizados, describe la arquitectura en la que se conectan los componentes y habla sobre la importancia del concepto. El resto de los capítulos en esta sección expanden el concepto de interconexión de redes y proporcionan detalles adicionales sobre la tecnología. Analizan los protocolos individuales y explican cómo es que cada uno utiliza técnicas de los capítulos anteriores para obtener una comunicación confiable y libre de errores.

20.2 La motivación para la interconexión de redes

Cada tecnología de red está diseñada para adaptarse a un conjunto específico de limitaciones. Por ejemplo, las tecnologías LAN están diseñadas para ofrecer una comunicación de alta velocidad a través de distancias cortas, mientras que las tecnologías de WAN están diseñadas para brindar comunicación a través de áreas extensas. En consecuencia:

Ninguna tecnología individual de red es la mejor para todas las necesidades.

Una organización de gran tamaño con diversas necesidades de trabajo en red necesita varias redes físicas. Lo que es más importante, si la organización selecciona el tipo de red que sea mejor para cada tarea, tendrá varios tipos de redes. Por ejemplo, una tecnología de LAN como Ethernet podría ser la mejor solución para conectar las computadoras que se encuentran en un sitio dado, pero para interconectar un sitio que está en una ciudad con un sitio que se encuentra en otra, podría usarse un circuito de datos arrendado.

20.3 El concepto de servicio universal

El principal problema que surge cuando se tienen varias redes debería ser obvio: una computadora conectada a cierta red sólo puede comunicarse con otras computadoras conectadas a la misma red. El problema se hizo evidente en la década de 1970, a medida que las grandes organizaciones comenzaron a adquirir varias redes. Cada red en la organización formaba una isla. En muchas de las primeras instalaciones, cada computadora se conectaba a una sola red y los empleados tenían que elegir una computadora apropiada para cada tarea. Es decir, a un empleado se le otorgaba acceso a varias pantallas y teclados, y para enviar un mensaje a través de la red apropiada se veía obligado a desplazarse de una computadora a otra.

Los usuarios nunca estarían satisfechos ni serían productivos si debieran usar una computadora independiente para cada red. En consecuencia, la mayoría de los sistemas de comunicaciones computacionales modernos permiten la comunicación entre dos computadoras cualquiera, del mismo modo que un sistema telefónico proporciona comunicación entre dos teléfonos cualquiera. Este concepto, que se conoce como *servicio universal*, se ha convertido en un dogma fundamental del trabajo en red. Con el servicio universal, un usuario en cualquier computadora de cualquier organización puede enviar mensajes o datos a cualquier otro usuario. Además, el usuario no tiene que cambiar el equipo de cómputo al cambiar las tareas que realiza; toda la información está disponible para todas las computadoras. Como resultado, los usuarios son más productivos. Para resumir:

Un sistema de comunicaciones que brinda un servicio universal permite que cualquier par de computadoras se comuniquen entre sí.

20.4 Servicio universal en un mundo heterogéneo

¿El servicio universal significa que todos necesitan adoptar una sola tecnología de red, o es posible tener un servicio universal a través de varias redes que usan múltiples tecnologías? Las incompatibilidades hacen que sea imposible formar una red extensa con sólo interconectar los cables entre las redes. Además, algunas técnicas de extensión como el uso de puentes no pueden usarse con tecnologías de redes heterogéneas, ya que cada tecnología usa su propio formato de paquetes y esquema de direccionamiento. Por lo tanto, una trama creada para cierta tecnología de red no puede transmitirse en una red que use una tecnología diferente. Podemos resumir este punto así:

Aunque el servicio universal es altamente deseable, las incompatibilidades entre el hardware de red, las tramas y las direcciones impiden que una red puenteada utilice cualquier tecnología.

20.5 Interconexión de redes

A pesar de las incompatibilidades entre las tecnologías de redes, los investigadores idearon un esquema que proporciona un servicio universal entre redes heterogéneas. Conocido como *interconexión de redes*, el esquema usa tanto hardware como software. Se usan sistemas de hardware adicionales para interconectar un conjunto de redes físicas. El software instalado en las computadoras conectadas proporciona entonces un servicio universal. El sistema resultante de redes físicas conectadas se conoce como *interconexión de redes* o *interred*.

La interconexión de redes es bastante ambigua. En particular, una interred no está limitada en tamaño; hay interredes que contienen sólo unas cuantas redes mientras que la red Internet global contiene cientos de miles de redes. De manera similar, el número de computadoras conectadas a cada red individual dentro de una interred puede variar. Algunas redes no tienen computadoras conectadas, mientras que otras tienen cientos de ellas.

20.6 Conexión física de redes con enrutadores

El componente básico de hardware utilizado para conectar redes heterogéneas es un *enrutador*. Físicamente, un enrutador es un sistema de hardware independiente dedicado a la tarea de interconectar redes. Al igual que un puente, un enrutador contiene un procesador y memoria, así como una interfaz independiente de E/S para cada red a la que se conecta. La red considera que una conexión a un enrutador es lo mismo que una conexión a cualquier otra computadora. La figura 20.1 ilustra que la conexión física de redes con un enrutador es simple.

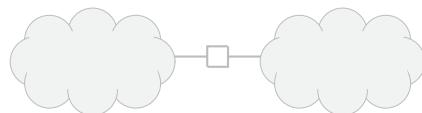


Figura 20.1 Dos redes físicas conectadas por un enrutador, el cual tiene una interfaz independiente para cada conexión de red. Se pueden conectar computadoras a cada red.

Puesto que las conexiones de los enrutadores no se limitan a una tecnología de red específica, la figura usa una nube para representar a cada red. Un enrutador puede conectar dos redes LAN, una LAN y una WAN o dos WAN. Además, cuando un enrutador conecta dos redes de la misma categoría general, no necesitan usar la misma tecnología. Por ejemplo, un enrutador puede conectar una red Ethernet a una red Wi-Fi. De esta forma, cada nube representa una tecnología cualquiera de red. Podemos resumir esto así:

Un enrutador de Internet es un sistema de hardware de propósito específico dedicado a la tarea de interconectar redes. Un enrutador puede interconectar redes que usen distintas tecnologías, incluyendo diferentes medios, esquemas de direccionamiento físicos o formatos de trama.

20.7 Arquitectura de Internet

Los enrutadores permiten que las organizaciones seleccionen las tecnologías de redes apropiadas para cada necesidad y que usen enrutadores para conectar todas las redes en una interred. Por ejemplo, la figura 20.2 ilustra cómo pueden usarse tres enrutadores para conectar cuatro redes físicas en una interred.

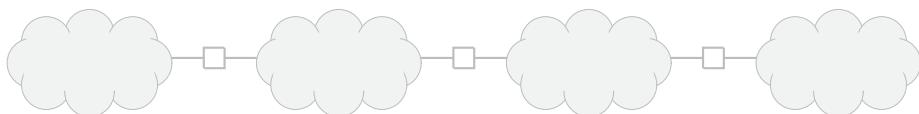


Figura 20.2 Una interred que se forma mediante el uso de tres enrutadores para interconectar cuatro redes físicas.

Aunque la figura muestra que cada enrutador sólo tiene dos conexiones, los enrutadores comerciales pueden conectar más de dos redes. Por lo tanto, un solo enrutador podría conectar las cuatro redes del ejemplo. A pesar de la viabilidad, las organizaciones raras veces usan un solo enrutador para conectar todas sus redes. Existen dos razones de ello:

- En vez de adquirir un costoso enrutador que conecte todas las redes, por lo general una organización compra varios enrutadores pequeños y así puede actualizarlos de manera independiente cuando se excede su capacidad.
- El uso de varios enrutadores mejora la confiabilidad y evita que un solo punto de falla afecte toda la red. Más adelante veremos que si existen varias rutas, la tecnología de Internet puede desviar el tráfico por una ruta alternativa cuando un enlace falla.

Por lo tanto, al planificar una interred, la organización debe elegir un diseño que cumpla con las necesidades de la organización en cuanto a confiabilidad, capacidad y costo. Los detalles exactos de la topología de la interred dependen a menudo del ancho de banda de las redes físicas, del tráfico esperado, de los requerimientos de confiabilidad de la organización, así como del costo y el rendimiento del hardware de enrutador disponible. Para resumir:

Una interred consiste en un conjunto de redes interconectadas por enrutadores. El esquema de la interred permite que cada organización seleccione el número y tipo de redes, el número de enrutadores a usar para interconectarlas y la topología de interconexión exacta.

20.8 Intranet e Internet

Utilizamos el término *red interna* o *intranet* para referirnos a una interred que pertenece a una organización privada (por ejemplo, una empresa) y está diseñada para que sólo la usen los empleados de esa organización. Por otra parte, usamos el término *Internet* para referirnos a la red Internet global y a los protocolos relacionados con ella.

La distinción entre intranets e Internet no está muy clara debido a dos razones. Primero, para crear su intranet la mayoría de las organizaciones utilizan ahora el mismo equipo y software de protocolo que usan los ISP para crear los enlaces de la red Internet. Segundo, como se conectan directamente a Internet, las intranets organizacionales pueden verse como parte de ésta en vez de verse como una entidad independiente. La situación es en especial incierta si una organización usa internamente direcciones de Internet globales. Sin embargo, la definición se concentra en la propiedad y el control, en vez de la tecnología.

20.9 Obtención de un servicio universal

El objetivo de la interconexión de redes es tener un servicio universal entre redes heterogéneas. Para brindar un servicio universal entre todas las computadoras de una interred, los enrutadores deben estar de acuerdo en reenviar la información desde un punto de origen en una red hasta un destino especificado en otra. La tarea es compleja, ya que los formatos de trama y los esquemas de direccionamiento que usan las redes involucradas pueden ser distintos. Como resultado, para que el servicio universal sea posible, hay que usar software de protocolo tanto en las computadoras como en los enrutadores.

Los capítulos posteriores describen el software de protocolo de Internet y explican cómo los protocolos de Internet solventan las diferencias en los formatos de trama y las direcciones físicas para hacer posible la comunicación entre redes que usan distintas tecnologías. Antes de considerar cómo funcionan los protocolos de Internet, es importante comprender el efecto que tiene el sistema de una interred en las computadoras conectadas.

20.10 Una red virtual

En general, el software de Internet crea la sensación de un solo sistema de comunicaciones sin fallas al que se conectan muchas computadoras. El sistema ofrece un servicio universal en el que a cada computadora se le asigna una dirección y cualquier computadora puede enviar un paquete a cualquier otra computadora. Además, el software de protocolo de Internet oculta los detalles de las conexiones físicas de red, las direcciones físicas y la información de enrutamiento. Ni los usuarios ni los programas de aplicación están conscientes de las redes físicas involucradas ni de los enrutadores que las conectan.

Decimos que una interred es un sistema de *red virtual* debido a que el sistema de comunicaciones es una abstracción. Es decir, aunque una combinación de hardware y software ofrece la ilusión de un sistema de red uniforme, no existe dicha red. La figura 20.3 ilustra el concepto de redes virtuales, así como la estructura física correspondiente.

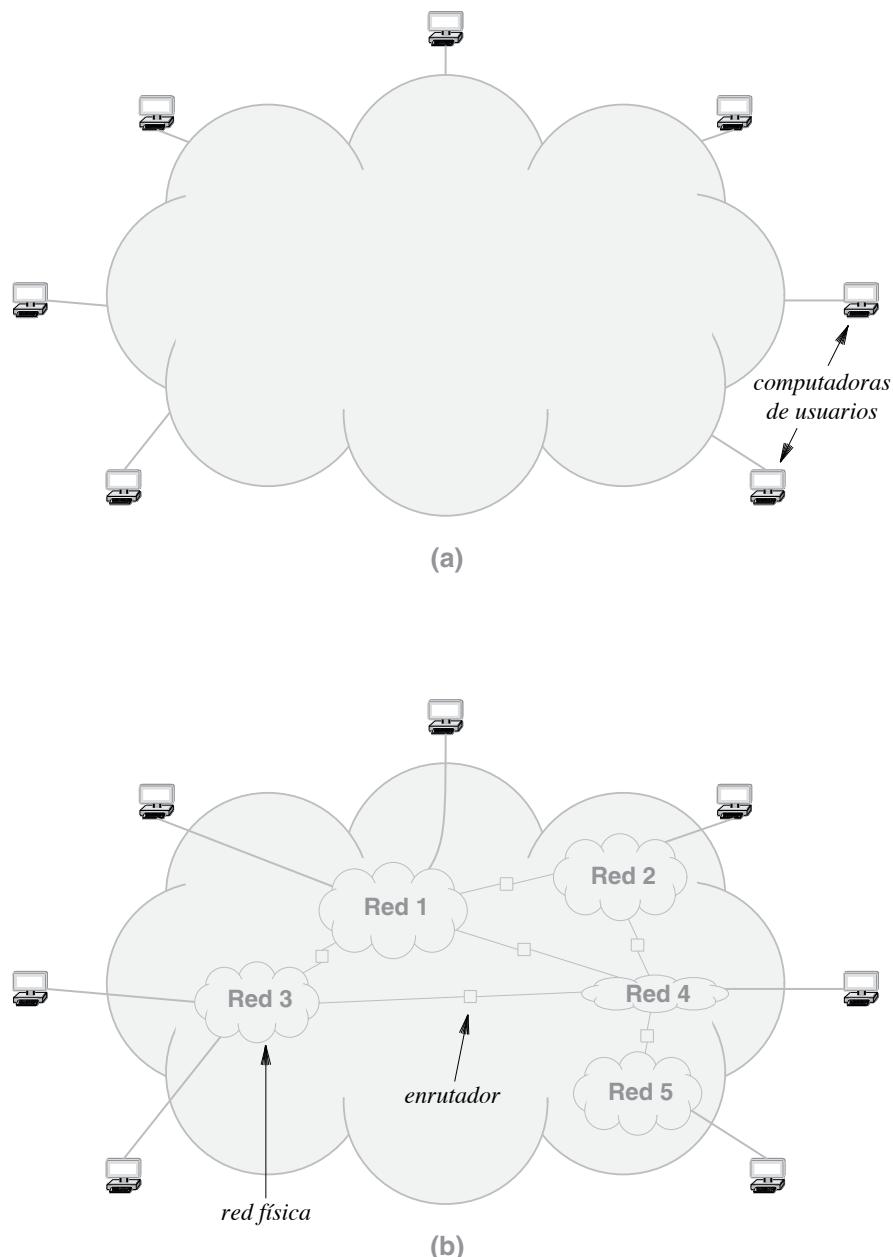


Figura 20.3 El concepto de Internet. (a) La sensación de una sola red que los usuarios y las aplicaciones perciben, y (b) la estructura física involucrada con enrutadores que interconectan a las redes.

20.11 Protocolos para la interconexión de redes

Aunque se han propuesto varios protocolos para usarse con las interredes, hay una suite de protocolos que se ha convertido en un éxito comercial. Ésta se conoce formalmente como *protocolos TCP/IP de Internet*; la mayoría de los profesionales de redes simplemente se refieren a la suite como *TCP/IP*.[†]

TCP/IP se desarrolló al mismo tiempo que la red Internet global. De hecho, los mismos investigadores que propusieron TCP/IP también propusieron la arquitectura de Internet antes descrita. El trabajo con TCP/IP comenzó en la década de 1970, aproximadamente al mismo tiempo que se desarrollaron las redes de área local, y continuó hasta principios de la década de 1990, cuando la red Internet se hizo comercial.

20.12 Repaso de la distribución en capas de TCP/IP

En el capítulo 1 vimos que los protocolos de Internet usan un modelo de referencia de cinco capas, como se ilustra en la figura 20.4.

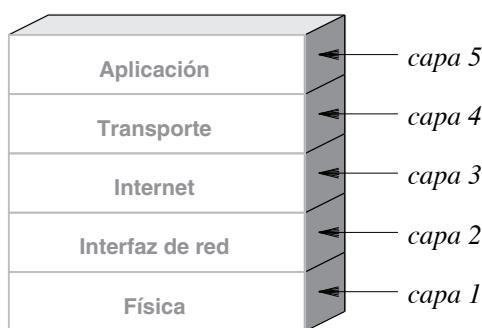


Figura 20.4 Las cinco capas del modelo de referencia TCP/IP.

Ya exploramos tres de las capas. Los capítulos de la parte I del libro consideran la capa 5 (aplicaciones), y los capítulos de las partes II y III hablan sobre los protocolos de las capas 1 y 2. Los capítulos de esta parte del libro consideran en detalle las dos capas restantes (capas 3 y 4):

- La capa 3 (IP) especifica el formato de los paquetes que se envían a través de Internet, así como los mecanismos utilizados para reenviar paquetes de una computadora hacia uno o más enruteadores, hasta un destino final.
- La capa 4 (TCP) especifica los mensajes y procedimientos que se usan para asegurar una transferencia confiable.

[†] TCP e IP son acrónimos de dos de los protocolos más importantes en la suite; el nombre del conjunto de protocolos se pronuncia deletreando T-C-P-I-P.

Para resumir:

Los protocolos de Internet se organizan en cinco capas conceptuales. El protocolo IP está en la capa 3 y el TCP en la capa 4.

20.13 Computadoras host, enrutadores y capas de protocolos

Usamos el término *host* para referirnos a un sistema final que se conecta a Internet y ejecuta aplicaciones. Un host puede ser tan pequeño como un teléfono inteligente o puede ser tan grande como una computadora *mainframe* o servidor de bases de datos. Además, el CPU de un host puede ser lento o rápido, la memoria puede ser grande o pequeña y la red a la que se conecta un host puede operar a velocidad alta o baja. Los protocolos TCP/IP hacen posible la comunicación entre cualquier par de computadoras, a pesar de las diferencias de hardware.

Tanto hosts como enrutadores necesitan software de protocolo TCP/IP. Sin embargo, los enrutadores no usan protocolos de todas las capas. En particular, un enrutador no necesita los protocolos de la capa 5 para aplicaciones como la transferencia de archivos, ya que no ejecutan aplicaciones convencionales.[†] En el siguiente capítulo hablaremos con más detalle sobre el software de protocolo TCP/IP y mostraremos cómo funciona la distribución en capas en Internet.

20.14 Resumen

En el sentido lógico, Internet parece ser un sistema de comunicaciones simple y sin fallas. Un par cualquiera de computadoras conectadas a Internet pueden comunicarse como si estuvieran conectadas a una sola red. Es decir, una computadora puede enviar un paquete a cualquier otra computadora conectada a Internet. En el sentido físico, Internet es una colección de redes interconectadas por dispositivos conocidos como *enrutadores*. Cada enrutador es un dispositivo de propósito específico que se conecta a dos o más redes y está dedicado a transferir paquetes de Internet entre las redes.

Los sistemas finales que se conectan a Internet se llaman *hosts*. Un host puede ser una computadora grande (por ejemplo, una supercomputadora) o un sistema pequeño (como un teléfono celular). Cada host se conecta a una de las redes físicas de Internet.

La ilusión de un solo sistema de comunicaciones se proporciona mediante software de protocolo de Internet. Cada host o enrutador en Internet debe ejecutar el software, el cual oculta los detalles de las conexiones físicas subyacentes y se encarga de enviar cada paquete a su destino.

Los protocolos más importantes desarrollados para la interconexión de redes se conocen como *protocolos TCP/IP de Internet*, que por lo general se abrevian como *TCP/IP*. Además de usarse en interredes privadas, TCP/IP se ha utilizado por décadas en la red Internet global.

[†] En la práctica, la mayoría de los enrutadores no ejecutan aplicaciones convencionales, pero sí ejecutan software de aplicación especial que permite a un administrador monitorear y controlar el enrutador desde una ubicación remota.

EJERCICIOS

- 20.1** ¿Será Internet sustituida por una sola tecnología de redes? ¿Por qué sí o por qué no?
- 20.2** ¿Cuál es la principal dificultad al brindar un servicio universal?
- 20.3** ¿Cuáles son dos razones por las que una organización no usa un solo enrutador para conectar todas sus redes?
- 20.4** Si un enrutador dado puede conectarse al menos a K redes, ¿cuántos enrutadores R se requieren para conectar N redes? Escriba una ecuación que obtenga R en términos de N y K .
- 20.5** Los usuarios perciben a Internet como una sola red. ¿Cuál es la realidad y a qué se conecta la computadora de un usuario?
- 20.6** En el modelo de referencia de 5 capas utilizado con los protocolos TCP/IP de Internet, ¿cuál es el propósito de cada una de las cinco capas?

Contenido del capítulo

- 21.1 Introducción, 345
- 21.2 El cambio a IPv6, 345
- 21.3 El modelo del reloj de arena y la dificultad del cambio, 346
- 21.4 Direcciones para Internet virtual, 346
- 21.5 El esquema de direccionamiento de IP, 348
- 21.6 La jerarquía de direcciones IP, 348
- 21.7 Clases originales de direcciones IPv4, 349
- 21.8 Notación decimal con puntos de IPv4, 350
- 21.9 Autoridad para las direcciones, 351
- 21.10 Subred IPv4 y direccionamiento sin clases, 351
- 21.11 Máscaras de direcciones, 353
- 21.12 Notación CIDR utilizada con IPv4, 354
- 21.13 Un ejemplo de CIDR, 354
- 21.14 Direcciones de hosts de CIDR, 356
- 21.15 Direcciones IPv4 especiales, 357
- 21.16 Resumen de direcciones IPv4 especiales, 359
- 21.17 Formato de dirección de difusión Berkeley de IPv4, 359
- 21.18 Los enrutadores y el principio de direccionamiento de IPv4, 360
- 21.19 Hosts multiproveedor, 361
- 21.20 Multihoming de IPv6 y renumeración de red, 361
- 21.21 Direccionamiento de IPv6, 362
- 21.22 Notación hexadecimal de dos puntos de IPv6, 363
- 21.23 Resumen, 364

21

IP: direccionamiento de Internet

21.1 Introducción

El capítulo anterior explica la arquitectura física de Internet, en la cual los enruteadores interconectan redes físicas. Este capítulo comienza una descripción del software de protocolo que hace que Internet parezca un solo sistema de comunicaciones sin fallas. El capítulo introduce el esquema de direccionamiento utilizado por el *protocolo de Internet* y habla sobre el uso de máscaras de direcciones. Internet está en transición entre el IP versión 4 (*IPv4*) y el IP versión 6 (*IPv6*).[†] En consecuencia, el capítulo cubre ambas versiones. El texto presenta los principios generales que se aplican a ambas versiones y luego brinda los detalles de IPv4 y de IPv6.

Los siguientes capítulos amplían la descripción de IP. Cada uno considera en detalle un aspecto del protocolo. Si se toman como grupo, los capítulos definen el protocolo IP y explican cómo el software de IP permite a las computadoras intercambiar paquetes a través de Internet.

21.2 El cambio a IPv6

Antes de considerar el direccionamiento en IPv4 e IPv6, es importante entender el cambio que está ocurriendo. IPv4 ha sido un protocolo con mucho éxito. El diseño permitió a Internet adaptarse a redes heterogéneas, a cambios drásticos en la tecnología de hardware y a un exagerado aumento de su tamaño. La versatilidad y escalabilidad de IPv4 son evidentes en las aplicaciones que lo utilizan y en el tamaño de la red Internet global. En conclusión:

[†] Por razones históricas y políticas, se omitió la versión número 5.

El éxito de IPv4 es increíble; el protocolo se adaptó a los cambios en las tecnologías de hardware, a redes heterogéneas y a un tamaño extremadamente grande de Internet.

Si IP funciona tan bien, ¿por qué cambiar? Cuando se definió el IPv4, sólo existían unas cuantas redes de computadoras. Los diseñadores decidieron usar 32 bits para una dirección IP debido a que ello permitía a Internet incluir más de un millón de redes. Sin embargo, la red Internet global sigue creciendo en forma exponencial y su tamaño se duplica en menos de un año. Ya se asignaron todas las direcciones IPv4. Por consiguiente, la principal justificación para definir una nueva versión de IP surgió debido a la limitante del espacio de direcciones, ya que se necesitaban direcciones más grandes para adaptarse al crecimiento continuo de Internet.

21.3 El modelo del reloj de arena y la dificultad del cambio

Aunque la escasez aparente de direcciones disponibles se consideraba crucial cuando se inició el trabajo en una nueva versión de IP en 1993, no ocurrió ninguna emergencia y las organizaciones se rehusaban a cambiar a una nueva versión. Para entender esto, piense en la importancia del protocolo IP y en el costo relacionado con el cambio. En términos de importancia, IP se encuentra en el centro de la comunicación de Internet. Todas las aplicaciones usan IP, e IP se ejecuta a través de todas las tecnologías de redes involucradas. Los profesionales de redes dicen que la comunicación de Internet sigue un *modelo de reloj de arena*, y que IP se encuentra en el punto en el que el reloj de arena es más delgado. La figura 21.1 muestra el concepto.

Hay un punto importante derivado de la dependencia que se tiene hacia IP y de la consecuente inercia que representa este protocolo:

Puesto que IP es crucial para toda la comunicación en Internet, cambiar este protocolo requiere un cambio en toda Internet.

21.4 Direcciones para Internet virtual

En el capítulo 20 vimos que el objetivo de la interconexión de redes es brindar un sistema de comunicación sin fallas. Para lograrlo, el software de protocolo debe ocultar los detalles de las redes físicas y ofrecer la ilusión de una sola red extensa. Desde el punto de vista de una aplicación, la red Internet virtual opera como cualquier otra red y permite a las computadoras tanto enviar como recibir paquetes. La principal diferencia entre Internet y una red física es que Internet es una abstracción imaginada por sus diseñadores y creada en su totalidad por el software de protocolo. Por lo tanto, los diseñadores seleccionan direcciones, formatos de paquetes y técnicas de entrega independientemente de las características del hardware involucrado.

El direccionamiento es un componente imprescindible en esta abstracción de Internet. Para dar la apariencia de una sola red, todas las computadoras host deben usar un esquema de direccionamiento

uniforme y cada dirección debe ser única. Aunque cada computadora tiene una dirección MAC, dichas direcciones no bastan debido a que Internet puede incluir varias tecnologías de red y cada tecnología define sus propias direcciones MAC.

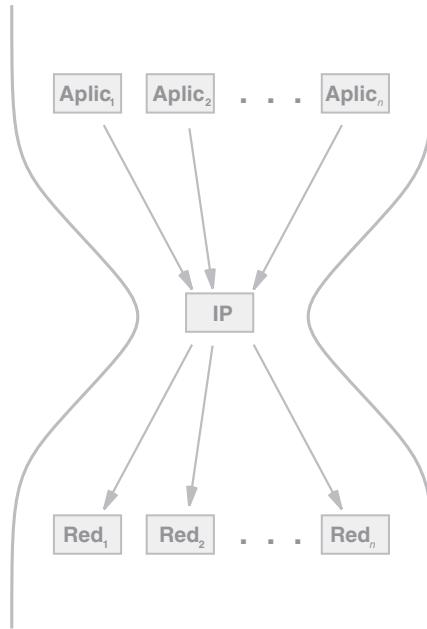


Figura 21.1 El modelo del reloj de arena de la comunicación de Internet con el protocolo IP en el centro.

Para garantizar un direccionamiento uniforme, IP define un esquema de direccionamiento independiente de las direcciones MAC. En Internet las direcciones IP se usan como destinos de manera similar a como las direcciones MAC se usan en una red LAN. Para enviar un paquete a través de Internet, el emisor coloca la dirección IP de destino en el paquete y lo pasa al software del protocolo IP para reenviarlo. Cuando reenvía el paquete a través de Internet hacia la computadora de destino, el software del protocolo IP usa la dirección IP de destino.

La ventaja del direccionamiento IP recae en la uniformidad: cualquier par de programas de aplicación pueden comunicarse sin conocer el tipo de hardware de red o direcciones MAC que se utilicen. La ilusión es tan completa que algunos usuarios se sorprenden de saber que las direcciones IP las suministra el software de protocolo y no forman parte de la red en sí. Más adelante veremos que muchas capas de software de protocolo usan direcciones IP. Para resumir:

Para brindar un direccionamiento uniforme en Internet, IP define un esquema de direccionamiento abstracto que asigna a cada host una dirección de protocolo única; las aplicaciones usan las direcciones IP para comunicarse.

21.5 El esquema de direccionamiento de IP

IP especifica que a cada host se le debe asignar un número conocido como *dirección de protocolo de Internet*, *dirección IP* o *dirección de Internet* del host.[†] IPv4 usa direcciones de 32 bits e IPv6 usa direcciones de 128 bits. Al enviar un paquete a través de Internet, el emisor debe especificar su propia dirección IP (la dirección de origen) así como la dirección del receptor deseado (la dirección de destino).

Para resumir:

Una dirección de Internet (dirección IP) es un número binario único que se asigna a un host y se usa para toda la comunicación con el host. IPv4 usa direcciones de 32 bits, mientras que IPv6 usa direcciones de 128 bits.

21.6 La jerarquía de direcciones IP

De manera similar al uso del direccionamiento jerárquico en las redes WAN, cada dirección IP se divide en dos partes: un prefijo y un sufijo. A diferencia de una WAN tradicional, Internet usa la conmutación de paquetes. En su lugar, un prefijo IP identifica a la red física a la que el host está conectado mientras que un sufijo IP identifica a una computadora específica de la red. Es decir, a cada red física de Internet se le asigna un *número de red* único. El número de red aparece como prefijo en la dirección IP de cada computadora conectada a la red y a cada computadora de una red física específica se le asigna un sufijo único.

Para garantizar la unicidad, no es posible asignar el mismo número de red a dos redes de Internet. Por lo tanto, un administrador de red debe asegurarse de no asignar el mismo sufijo a dos computadoras de una red dada. Por ejemplo, si una interred contiene tres redes, se les podrían asignar los números de red 1, 2 y 3. A tres computadoras conectadas a la red 1 se les pueden asignar los sufijos 1, 3 y 5, mientras que a tres computadoras conectadas a la red 2 se les pueden asignar los sufijos 1, 2 y 3. Los valores asignados no necesitan ser consecutivos.

El punto importante es que el esquema de direcciones IP garantiza dos propiedades:

- A cada computadora se le asigna una dirección única (es decir, nunca se asigna una sola dirección a más de una computadora).
- Aunque las asignaciones de los números de red deben coordinarse a nivel global, los sufijos pueden asignarse a nivel local sin necesidad de una coordinación global.

La primera propiedad se garantiza debido a que una dirección IP contiene tanto un prefijo como un sufijo. Si dos computadoras se conectan a distintas redes físicas, los prefijos asignados a sus direcciones serán distintos. Si dos computadoras están conectadas a la misma red física, los sufijos asignados a sus direcciones serán diferentes. Por consiguiente, la dirección que se asigna a cada computadora es única.

[†] Los tres términos se usan como sinónimos intercambiables.

21.7 Clases originales de direcciones IPv4

Una vez que seleccionaron un tamaño para las direcciones IP y decidieron dividir cada dirección en dos partes, los diseñadores de IPv4 tuvieron que determinar cuántos bits debían colocar en cada parte. El prefijo necesita suficientes bits para poder asignar un número de red único a cada red física que existe en Internet. Por su parte, el sufijo necesita suficientes bits para permitir que a cada computadora conectada a una red se le asigne un sufijo único. No era posible una elección simple debido a que agregar bits a una parte significa restar bits a la otra. Al elegir un prefijo grande se pueden alojar muchas redes, pero se limita el tamaño de cada red; por otra parte, elegir un sufijo grande significa que cada red física puede contener muchas computadoras, pero limita el número total de redes.

Puesto que Internet incluye diversas tecnologías de red, está conformada por algunas redes físicas grandes y muchas redes pequeñas. Por lo tanto, los diseñadores eligieron un esquema de direccionamiento para dar cabida a una combinación de redes grandes y pequeñas. El esquema original, que se conoce como *direccionamiento IP con clases*, divide el espacio de direcciones de IPv4 en tres *clases* primarias, donde cada clase tiene un prefijo y sufijo de distinto tamaño.

Los primeros cuatro bits de una dirección determinaban la clase a la que pertenecía una dirección y especificaban cómo se dividía el resto de la dirección en prefijo y sufijo. La figura 21.2 ilustra las cinco clases de direcciones, los bits a la izquierda que se utilizan para identificar cada clase, así como la división de prefijo y sufijo. La figura sigue la convención utilizada en los protocolos TCP/IP de enumerar bits de izquierda a derecha y usar cero para el primer bit.

	bits	0	1	2	3	4	8	16	24	31
Clase A	0									sufijo
Clase B	1	0								sufijo
Clase C	1	1	0							sufijo
Clase D	1	1	1	0						dirección de multidifusión
Clase E	1	1	1	1						reservado (no asignado)

Figura 21.2 Las cinco clases de direcciones IPv4 en el esquema original con clases.

Aunque el esquema con clases se sustituyó, las direcciones de la clase D se siguen usando para la multidifusión, lo cual permite la entrega de datos a un conjunto de computadoras. Cada dirección de multidifusión corresponde a un grupo de computadoras. Una vez que se establezca un grupo de multidifusión, a cada host del grupo se le entregará una copia de cualquier paquete que se envíe a la dirección de multidifusión.

En la práctica, la multidifusión de Internet nunca ha estado disponible a nivel global, lo cual significa que la multidifusión se limita a sitios individuales.

Podemos resumir esto así:

El esquema de direccionamiento IPv4 original dividía las direcciones en clases. Las direcciones de la clase D se siguen usando para la multidifusión, pero ésta no se encuentra disponible a través de la red Internet global.

21.8 Notación decimal con puntos de IPv4

Aunque las direcciones IPv4 son números de 32 bits, los usuarios no introducen ni leen los valores en binario. En su lugar, al interactuar con un usuario el software usa una notación que es más fácil de entender para el ser humano. Conocida como *notación decimal con puntos*, la forma expresa cada sección de 8 bits del total de 32 bits como un valor decimal y usa puntos para separar las secciones. La figura 21.3 muestra ejemplos de números binarios y la notación decimal equivalente con puntos.

Número binario de 32 bits	Decimal con puntos equivalente
10000001 00110100 00000110 00000000	129.52.6.0
11000000 00000101 00110000 00000011	192.5.48.3
00001010 00000010 00000000 00100101	10.2.0.37
10000000 00001010 00000010 00000011	128.10.2.3
10000000 10000000 11111111 00000000	128.128.255.0

Figura 21.3 Ejemplos de números binarios de 32 bits y su equivalente en la notación decimal con puntos que se usa con IPv4.

La notación decimal con puntos trata a cada *octeto* (cada valor de 8 bits) como un entero binario sin signo.[†] Como lo muestra el ejemplo final de la figura, el menor valor posible (0) ocurre cuando todos los bits de un octeto son cero y el mayor valor posible (255) ocurre cuando todos los bits de un octeto son uno. De esta forma, las direcciones decimales con puntos varían de 0.0.0.0 hasta 255.255.255.255. Las direcciones de multidifusión (clase D) ocupan el rango de 224.0.0.0 hasta 239.255.255.255.

[†] Los protocolos de Internet usan el término *octeto* en vez de *byte*, ya que el tamaño de un byte depende de la computadora. Por lo tanto, aunque 8 bytes se han convertido de facto en el estándar, el octeto es inequívoco.

Para resumir:

La notación decimal con puntos es una forma sintáctica que el software de IPv4 usa para expresar valores binarios de 32 bits al interactuar con el ser humano. Una notación decimal con puntos representa cada octeto en forma decimal y usa un punto para separarlos.

21.9 Autoridad para las direcciones

Cada prefijo asignado a una red en Internet debe ser único. Por lo tanto, es necesario que una organización central los asigne. En la actualidad, la responsabilidad recae en la *Corporación de asignación de nombres y números de Internet (ICANN)*. A medida que Internet sobrepasó los límites de sus raíces de investigación, se estableció la ICANN específicamente para manejar la asignación de direcciones y resolver disputas.

La ICANN no asigna directamente los prefijos individuales, sino que autoriza a un conjunto de *registradores* para que lo hagan. Cada región geográfica tiene un registrador (por ejemplo, hay uno para Norteamérica, otro para Europa y así en lo sucesivo). Los registradores ponen grandes bloques de direcciones a disposición de los ISP importantes quienes, a su vez, los ponen a disposición de proveedores ISP más pequeños. Los ISP conectan a los suscriptores y proporcionan a cada suscriptor un conjunto de prefijos que éste usa para sus redes. Por lo tanto, para obtener un prefijo de red, una corporación o individuo se pone en contacto con un ISP.

21.10 Subred IPv4 y direccionamiento sin clases

A medida que Internet creció, el esquema de direccionamiento con clases de IPv4 original se convirtió en una limitación. Se inventaron dos nuevos mecanismos para superar el problema:

- Direccionamiento de subredes
- Direccionamiento sin clases

Los dos mecanismos están tan estrechamente relacionados que pueden considerarse parte de un solo concepto, en el cual en vez de tener tres clases distintas de direcciones, se permite que la división entre el prefijo y el sufijo ocurra en un límite de bits ajustable. El direccionamiento de subredes se usó inicialmente dentro de las organizaciones grandes que se conectaban a la red Internet global, mientras que el direccionamiento sin clases extendió este concepto a toda la red Internet. IPv6 también adoptó la idea.

Para entender el porqué de usar un límite ajustable, considere un ISP que asigna prefijos. Suponga que un cliente del ISP solicita un prefijo para una red que contiene treinta y cinco hosts. Si se utilizara el direccionamiento con clases, el ISP asignaría un prefijo de clase C. De hecho, sólo se necesitan seis bits de sufijo de host para representar los treinta y cinco valores, lo que significa que 219 de los 254 posibles sufijos nunca se asignarían a los hosts.[†] En otras palabras, se desperdicia la mayoría del espacio de las direcciones de clase C.

[†] El número 254 surge debido a que una dirección de clase C tiene 256 posibles sufijos y los sufijos que son todos ceros y todos unos se reservan para la difusión en la subred, como se describe más adelante en el capítulo.

Por desgracia, a medida que Internet creció fue obvio que en un momento dado se necesitarían todas las direcciones y no podíamos darnos el lujo de dejar direcciones sin usar. El direccionamiento sin clases resolvió el problema al permitir a un ISP que asignara un prefijo que fuera de un tamaño más apropiado. En nuestro ejemplo, un ISP que usa el direccionamiento sin clases puede asignar un prefijo que tenga veintiséis bits de longitud. Como una dirección IPv4 contiene treinta y dos bits, el sufijo tiene seis bits de longitud, lo que significa que hay sesenta y dos sufijos posibles. Como resultado, sólo veintisiete direcciones quedarán sin usarse.

Otra forma de analizar la situación es asumir que el ISP posee un prefijo de clase C. El direccionamiento sin clases asigna todo el prefijo a una organización. Sin embargo, con el direccionamiento sin clases el ISP puede dividir el prefijo en varios prefijos más largos y asignar cada uno a un suscriptor. La figura 21.4 ilustra cómo es que el direccionamiento sin clases permite a un ISP dividir un prefijo de clase C en cuatro prefijos más largos, cada uno de los cuales puede acomodar a una red de hasta sesenta y dos hosts.

En la figura, la parte de cada prefijo correspondiente al host se muestra en gris. La dirección original de clase C tiene ocho bits de sufijo y cada una de las direcciones sin clases tiene seis bits de sufijo. Suponiendo que el prefijo original de la clase C sea único, cada uno de los prefijos sin clases también será único. Por lo tanto, en vez de desperdiciar direcciones, el ISP puede asignar cada uno de los cuatro prefijos sin clases a un suscriptor con sesenta y dos hosts o menos.

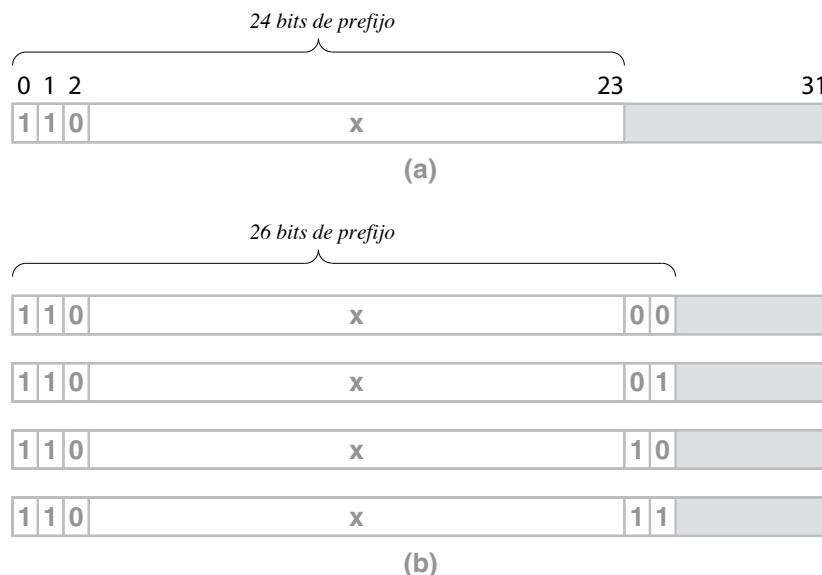


Figura 21.4 (a) Un prefijo IPv4 de la clase C y (b) el mismo prefijo dividido en cuatro prefijos sin clases.

21.11 Máscaras de direcciones

¿Cómo puede dividirse una dirección IP en un límite ajustable? Tanto el esquema de direccionamiento sin clases como el de subredes requieren de hosts y enrutadores que sean capaces de procesar direcciones que almacenen una pieza adicional de información: un valor que especifique el límite exacto entre el prefijo de red y el sufijo del host. Para marcar el límite, IPv4 usa un valor de 32 bits (e IPv6 usa un valor de 128 bits) conocido como *máscara de dirección*, conocida en un principio como *máscara de subred*. Una máscara de dirección tiene bits *uno* para marcar el prefijo de red y bits *cero* para marcar la parte correspondiente al host.

¿Por qué almacenar el tamaño del límite como una máscara de bits? Una máscara hace que el procesamiento sea eficiente. En especial, veremos que cuando los hosts y los enrutadores manejan un paquete de IP, necesitan comparar la parte de la dirección correspondiente al prefijo de red con un valor de sus tablas de reenvío. La representación de la máscara de bits hace que la comparación sea eficiente. Para entender por qué, suponga que un enrutador que usa IPv4 recibe una dirección de destino D , un prefijo de red representado como un valor de 32 bits N y una máscara de dirección de 32 bits M . Es decir, suponga que los bits superiores de N contienen un prefijo de red y que el resto de los bits se establecen en cero. Para evaluar si el destino se encuentra en la red especificada, el enrutador prueba la condición:

$$N == (D \& M)$$

Esto es, el enrutador usa la máscara con una operación “y lógica” (AND) para establecer los bits del host de la dirección D en cero y luego compara el resultado con el prefijo de red N .

Como un ejemplo mediante el uso de IPv4, considere el siguiente prefijo de red de 32 bits:

10000000	00001010	00000000	00000000
----------	----------	----------	----------

que tiene el valor decimal con puntos *128.10.0.0*. Considere también una máscara de 32 bits que tiene dieciséis bits uno seguidos por 16 bits cero, los cuales pueden indicarse en notación decimal con puntos como *255.255.0.0*:

11111111	11111111	00000000	00000000
----------	----------	----------	----------

Ahora considere una dirección de destino de 32 bits *128.10.2.3*, cuyo equivalente binario es:

10000000	00001010	00000010	00000011
----------	----------	----------	----------

Una operación y lógica entre la dirección de destino y la máscara de dirección extrae los dieciséis bits de mayor orden, para producir el siguiente resultado:

10000000	00001010	00000000	00000000
----------	----------	----------	----------

esto equivale al prefijo de red *128.10.0.0*.

21.12 Notación CIDR utilizada con IPv4

El esquema de direccionamiento sin clases se conoce formalmente como *enrutamiento entre dominios sin clase (CIDR)*. El nombre es desafortunado debido a que CIDR sólo especifica el direccionamiento y el reenvío. Cuando se creó el esquema de direccionamiento CIDR, los diseñadores querían facilitar a un ser humano la especificación de una máscara. Para comprender la dificultad que esto representa, considere la máscara necesaria para el ejemplo de la figura 21.4(b), que tiene veintiséis bits 1 seguidos de seis bits 0. En decimal con puntos, la máscara es:

255.255.255.192

Para facilitar la especificación y la interpretación de los valores de máscara, se extendió la notación decimal con puntos. En la versión extendida, que se conoce como *notación CIDR*, es posible especificar una dirección y una máscara proporcionando una dirección decimal con puntos seguida de una barra diagonal y un número decimal que especifique el número de bits uno consecutivos que hay en la máscara, alineados a la izquierda. Es decir, la forma general es:

ddd.ddd.ddd.ddd/m

donde *ddd* es el valor decimal para un octeto de la dirección y *m* es el número de bits uno en la máscara. Por lo tanto, podríamos introducir una configuración de enrutador de la siguiente forma:

192.5.48.69/26

esto especifica una máscara de 26 bits. La figura 21.5 enlista las máscaras de dirección en notación CIDR junto con el equivalente decimal con puntos de cada una. Cabe mencionar que algunas de las máscaras de dirección CIDR corresponden a las asignaciones originales con clases.

21.13 Un ejemplo de CIDR

Como un ejemplo de CIDR, suponga que un ISP tiene el siguiente bloque de direcciones disponible para asignar:

128.211.0.0/16

Suponga además que el ISP tiene dos clientes: un cliente necesita doce direcciones IP y el otro necesita nueve. El ISP puede asignar a un cliente el prefijo CIDR:

128.211.0.16/28

y puede asignar al otro cliente:

128.211.0.32/28

Longitud (CIDR)	Máscara de dirección	Notas
/0	0 . 0 . 0 . 0 . 0	Todos 0 (sin máscara)
/1	128 . 0 . 0 . 0 . 0 . 0	
/2	192 . 0 . 0 . 0 . 0 . 0	
/3	224 . 0 . 0 . 0 . 0 . 0	
/4	240 . 0 . 0 . 0 . 0 . 0	
/5	248 . 0 . 0 . 0 . 0 . 0	
/6	252 . 0 . 0 . 0 . 0 . 0	
/7	254 . 0 . 0 . 0 . 0 . 0	
/8	255 . 0 . 0 . 0 . 0 . 0	Máscara original de clase A
/9	255 . 128 . 0 . 0 . 0 . 0	
/10	255 . 192 . 0 . 0 . 0 . 0	
/11	255 . 224 . 0 . 0 . 0 . 0	
/12	255 . 240 . 0 . 0 . 0 . 0	
/13	255 . 248 . 0 . 0 . 0 . 0	
/14	255 . 252 . 0 . 0 . 0 . 0	
/15	255 . 254 . 0 . 0 . 0 . 0	
/16	255 . 255 . 0 . 0 . 0 . 0	Máscara original de clase B
/17	255 . 255 . 128 . 0 . 0 . 0	
/18	255 . 255 . 192 . 0 . 0 . 0	
/19	255 . 255 . 224 . 0 . 0 . 0	
/20	255 . 255 . 240 . 0 . 0 . 0	
/21	255 . 255 . 248 . 0 . 0 . 0	
/22	255 . 255 . 252 . 0 . 0 . 0	
/23	255 . 255 . 254 . 0 . 0 . 0	
/24	255 . 255 . 255 . 0 . 0 . 0	Máscara original de clase C
/25	255 . 255 . 255 . 128 . 0 . 0	
/26	255 . 255 . 255 . 192 . 0 . 0	
/27	255 . 255 . 255 . 224 . 0 . 0	
/28	255 . 255 . 255 . 240 . 0 . 0	
/29	255 . 255 . 255 . 248 . 0 . 0	
/30	255 . 255 . 255 . 252 . 0 . 0	
/31	255 . 255 . 255 . 254 . 0 . 0	
/32	255 . 255 . 255 . 255 . 0 . 0	Todos 1 (máscara específica del host)

Figura 21.5 Una lista de máscaras de direcciones en notación CIDR y en decimal con puntos.

Aunque ambos clientes tienen el mismo tamaño de máscara (28 bits), los prefijos son distintos. El valor binario asignado a un cliente es:

10000000	11010011	00000000	0001 0000
----------	----------	----------	-----------

y el valor binario que se asigna al otro cliente es:

10000000	11010011	00000000	0010 0000
----------	----------	----------	-----------

De esta forma no hay ambigüedad; cada cliente tiene un prefijo único y puede asignar catorce direcciones IP. Lo que es más importante, el ISP retiene la mayoría del bloque original de direcciones, que puede asignar a otros clientes.

21.14 Direcciones de hosts de CIDR

Considere calcular el rango de direcciones en un bloque CIDR de IPv4. Una vez que un ISP asigna un prefijo CIDR a un cliente, éste puede asignar direcciones de host. Por ejemplo, suponga que a una organización se le asigna *128.211.0.16/28* como se describió anteriormente. La figura 21.6 muestra que la organización tendrá cuatro bits para usar como campo de dirección de host, y muestra la dirección más alta y la más baja tanto en forma binaria como en decimal con puntos. El ejemplo evita asignar las direcciones de host con todos los bits uno y todos los bits cero.

0	Prefijo de red 128.211.0.16/28	28	31				
	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; border: 1px solid black; padding: 2px;">1 0 0 0 0 0 0 0</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">1 1 0 1 0 0 1 1</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">0 0 0 0 0 0 0 0</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">0 0 0 1 0 0 0 0</td> </tr> </table>	1 0 0 0 0 0 0 0	1 1 0 1 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 0		
1 0 0 0 0 0 0 0	1 1 0 1 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 0				

0	Máscara de dirección 255.255.255.240	28	31				
	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; border: 1px solid black; padding: 2px;">1 1 1 1 1 1 1 1</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">1 1 1 1 1 1 1 1</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">1 1 1 1 1 1 1 1</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">1 1 1 1 0 0 0 0</td> </tr> </table>	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 0 0 0 0		
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 0 0 0 0				

0	Dirección de host más baja 128.211.0.17	28	31				
	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; border: 1px solid black; padding: 2px;">1 0 0 0 0 0 0 0</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">1 1 0 1 0 0 1 1</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">0 0 0 0 0 0 0 0</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">0 0 0 1 0 0 0 1</td> </tr> </table>	1 0 0 0 0 0 0 0	1 1 0 1 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 1		
1 0 0 0 0 0 0 0	1 1 0 1 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 1				

0	Dirección de host más alta 128.211.0.30	28	31				
	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; border: 1px solid black; padding: 2px;">1 0 0 0 0 0 0 0</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">1 1 0 1 0 0 1 1</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">0 0 0 0 0 0 0 0</td> <td style="width: 25%; border: 1px solid black; padding: 2px;">0 0 0 1 1 1 1 0</td> </tr> </table>	1 0 0 0 0 0 0 0	1 1 0 1 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 1 1 1 1 0		
1 0 0 0 0 0 0 0	1 1 0 1 0 0 1 1	0 0 0 0 0 0 0 0	0 0 0 1 1 1 1 0				

Figura 21.6 Ilustración del direccionamiento CIDR de IPv4 para un prefijo de ejemplo /28.

La figura 21.6 ilustra una desventaja del direccionamiento sin clases, ya que como el sufijo de host puede iniciar en un límite cualquiera, los valores no son tan fáciles de leer en decimal con puntos. Por ejemplo, cuando se combina con el prefijo de red, los catorce posibles sufijos de host dan como resultado los valores decimales con puntos del 128.211.0.17 al 128.211.0.30.

21.15 Direcciones IPv4 especiales

Además de asignar una dirección a cada computadora, es conveniente tener direcciones que puedan usarse para denotar redes o conjuntos de computadoras. El protocolo IP define una forma especial de direcciones como *reservadas*. Es decir, direcciones especiales que nunca se asignan a los hosts. Esta sección describe tanto la sintaxis como la semántica de cada forma de dirección especial.

21.15.1 Dirección de red IPv4

Una de las motivaciones para definir formas especiales de direcciones puede verse en la figura 21.6. Es conveniente tener una dirección que pueda usarse para denotar el prefijo asignado a una red dada. IP reserva la dirección de host cero y la usa para denotar una *red*. Así, la dirección 128.211.0.16/28 denota una red porque los bits más allá del 28 son cero. Una dirección de red nunca debe aparecer como la dirección de destino en un paquete.[†]

21.15.2 Dirección de difusión dirigida IPv4

Algunas veces es conveniente enviar una copia de un paquete a todos los hosts de una red física. Para simplificar la difusión, IPv4 define una *dirección de difusión dirigida* para cada red física. Cuando se envía un paquete a una dirección de difusión dirigida de una red, una sola copia del paquete viaja a través de Internet hasta que llega a la red especificada. Después el paquete se entrega a todos los hosts en la red.

La dirección de difusión dirigida para una red se forma mediante la adición de un sufijo que consiste en todos los bits 1 para el prefijo de red. De esta forma, el sufijo de host que consiste en todos los bits 1 *queda reservado*. Si un administrador asigna involuntariamente a una computadora el sufijo con todos los bits uno, el software podría fallar.

¿Cómo funciona la difusión? Si el hardware de red soporta la difusión, una difusión dirigida se entregará usando la capacidad de difusión del hardware. Si una red específica no tiene soporte de hardware para la difusión, el software debe enviar una copia independiente del paquete a cada host de la red.

[†] La sección 21.17 habla sobre la forma de dirección de difusión Berkeley de IPv4, que es una excepción no estándar.

21.15.3 Dirección de difusión limitada IPv4

El término *difusión limitada* se refiere a una difusión en una red que está conectada en forma directa. De manera informal decimos que la difusión se limita a “un solo cable”. La difusión limitada se usa durante el arranque del sistema en una computadora que no conoce todavía el número de red.

IPv4 reserva la dirección que consiste en treinta y dos bits 1 para la difusión limitada. De esta forma, el software de IP difundirá a través de la red local cualquier paquete que se envíe a la dirección que tenga todos los bits uno.

21.15.4 Dirección de esta computadora de IPv4

Como cada paquete de Internet contiene tanto la dirección de origen como la de destino, una computadora necesita conocer su dirección IPv4 antes de poder enviar o recibir paquetes de Internet. En el capítulo 23 aprenderemos que TCP/IP contiene protocolos que una computadora puede usar para obtener automáticamente su dirección IP al iniciarse. Lo interesante es que los protocolos de arranque usan el protocolo IP para comunicarse. Al usar dichos protocolos de arranque, una computadora no puede proporcionar una dirección de origen IP correcta. Para manejar estos casos, IPv4 reserva la dirección que consiste en todos los bits cero para indicar que se trata de *esta computadora*.[†]

21.15.5 Dirección de *loopback* IPv4

IP define una *dirección de ciclo de retorno* o *loopback* que se usa para probar aplicaciones de red. A menudo los programadores usan el *loopback* para hacer una depuración preliminar después de haber creado una aplicación de red. Para realizar una prueba de *loopback*, un programador debe tener dos programas de aplicación con el fin de comunicarse a través de una red. Cada aplicación incluye el código necesario para interactuar con el software de protocolo TCP/IP. En vez de ejecutar cada programa en una computadora independiente, el programador ejecuta ambos programas en una sola computadora y las instruye para que usen una dirección de *loopback* al comunicarse. Cuando una aplicación envía datos a la otra, estos datos viajan por la pila de protocolos hasta el software IP, el cual los reenvía a través de la pila de protocolos de regreso al segundo programa. De esta forma, el programador puede probar rápidamente la lógica del programa sin necesitar dos computadoras y sin enviar paquetes a través de una red.

IP reserva el prefijo de red 127/8 para usarlo con *loopback*. La dirección de host que se use con 127 es irrelevante, ya que todas las direcciones de host se tratan igual. Por convención, a menudo los programadores usan el host número 1, con lo que 127.0.0.1 se convierte en la dirección de *loopback* más popular.

Durante la prueba de *loopback* ningún paquete sale de la computadora. El software de IP reenvía los paquetes de un programa de aplicación a otro. En consecuencia, la dirección de *loopback* nunca aparece en un paquete que viaja a través de una red.

[†] El significado especial se aplica sólo cuando la dirección aparece como una dirección de origen en un paquete; una entrada con todos los bits cero puede aparecer también como la ruta predeterminada en la tabla de reenvío.

21.16 Resumen de direcciones IPv4 especiales

La tabla de la figura 21.7 sintetiza las formas de direcciones IP especiales.

Prefijo	Sufijo	Tipo de dirección	Propósito
todos cero	todos cero	esta computadora	se usa durante el arranque (<i>bootstrap</i>)
red	todos cero	red	identifica una red
red	todos uno	difusión dirigida	difusión en red especificada
todos uno	todos uno	difusión limitada	difusión en red local
127/8	cualquiera	<i>loopback</i>	prueba

Figura 21.7 Resumen de las formas de direcciones IP especiales.

Dijimos que las direcciones especiales son reservadas y nunca deben asignarse a las computadoras host. Además, cada dirección especial está restringida a ciertos usos. Por ejemplo, una dirección de difusión nunca debe aparecer como dirección de origen y la dirección con todos los bits cero no debe usarse una vez que un host complete el procedimiento de arranque y haya obtenido una dirección IP.

21.17 Formato de dirección de difusión Berkeley de IPv4

La Universidad de California en Berkeley desarrolló y distribuyó una de las primeras implementaciones de los protocolos TCP/IP como parte de BSD UNIX.[†] La implementación BSD contenía una característica no estándar que afectó a muchas implementaciones subsecuentes. En vez de usar un sufijo de host con todos los bits uno para representar una dirección de difusión dirigida, la implementación de Berkeley usa un sufijo de host con todos sus bits cero (es decir, idéntico a la dirección de red). La forma de la dirección se conoce de manera informal como *difusión de Berkeley*.

Por desgracia, muchos fabricantes de computadoras basaron su primer software de TCP/IP en la implementación de Berkeley, por lo que algunos sitios siguen usando la difusión de Berkeley. A menudo el software TCP/IP comercial incluye un parámetro de configuración que puede seleccionar entre la forma estándar TCP/IP y la forma de difusión de Berkeley; muchas implementaciones se crean de modo que puedan aceptar ambas formas de direcciones de difusión. Por consiguiente, si se permite la difusión dirigida, un administrador de red debe elegir la forma de dirección a usar en cada red.

[†] BSD significa distribución de software de Berkeley.

21.18 Los enrutadores y el principio de direccionamiento de IPv4

Además de asignar una dirección de Internet a cada host, el protocolo de Internet especifica que también hay que asignar direcciones IP a los enrutadores. De hecho, a cada enrutador se le asignan dos o más direcciones IP: una para cada red a la que se conecta. Para entender por qué, recuerde dos hechos:

- Un enrutador tiene conexiones hacia varias redes físicas.
- Cada dirección IPv4 contiene un prefijo que especifica una red física.

Por lo tanto, una sola dirección IPv4 no basta para un enrutador, ya que cada uno se conecta a varias redes y cada red tiene un prefijo único. El esquema IPv4 puede explicarse mediante un principio fundamental:

Una dirección IPv4 no identifica a una computadora específica. En su lugar, cada dirección IP identifica una conexión entre una computadora y una red. A una computadora con varias conexiones de red (por ejemplo, un enrutador) se le debe asignar una dirección IPv4 para cada conexión.

La figura 21.8 ilustra la idea con un ejemplo de direcciones IPv4 asignadas a dos enrutadores que conectan tres redes.

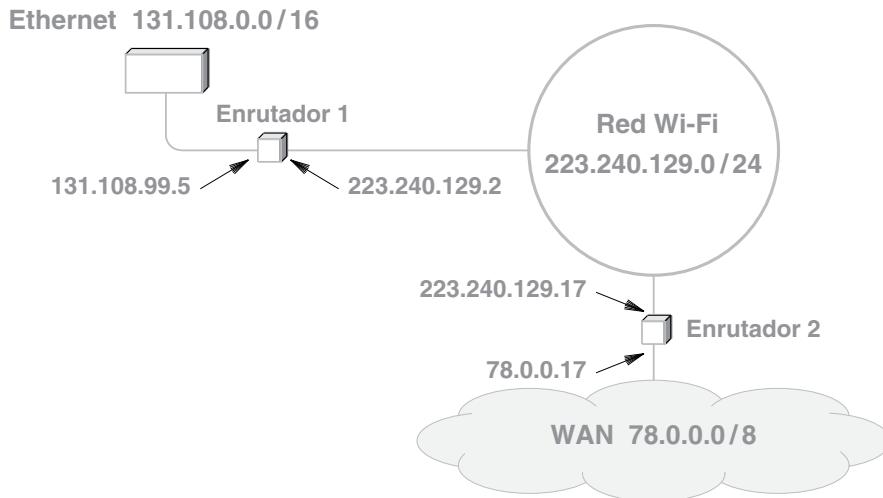


Figura 21.8 Un ejemplo de direcciones IPv4 asignadas a dos enrutadores.

El protocolo IP no requiere que se asigne el mismo sufijo a todas las interfaces de un enrutador. Por ejemplo, el enrutador que conecta a las redes Ethernet y Wi-Fi de la figura, tiene los sufijos 99.5 (conexión a Ethernet) y 2 (conexión a la red Wi-Fi). Sin embargo, IP no impide que se use el mismo sufijo para todas las conexiones. Por consiguiente, el ejemplo muestra que el administrador eligió usar el mismo sufijo 17 para ambas interfaces del enrutador que conecta la red Wi-Fi a la WAN. Como cuestión práctica, usar el mismo sufijo puede ayudar a las personas que administran las redes, ya que un solo número es más fácil de recordar.

21.19 Hosts multiproveedor

¿Puede un host conectarse a varias redes? La respuesta es “sí”. Se dice que una computadora host con varias conexiones de red es *multiproveedor*. A esta técnica se le conoce como *multihoming* y se utiliza para aumentar la confiabilidad de una red. Si una conexión falla, el host aún puede llegar a Internet a través de la segunda conexión. Como alternativa, el *multihoming* se usa para incrementar el rendimiento, ya que las conexiones a varias redes permiten enviar el tráfico en forma directa y evitar enrutadores que podrían estar congestionados. Al igual que un enrutador, un host multiproveedor tiene varias direcciones de protocolos, una para cada conexión de red.

En IPv4, el *multihoming* siempre ha sido una parte incómoda del diseño de protocolos, ya que permite a una computadora tener varias direcciones. Surgen preguntas como: si un paquete llega a través de una red pero tiene la dirección de host de otra red, ¿debe aceptarse? Una razón de dichas preguntas concierne a la seguridad ya que se podría hacer una suplantación de identidad o *spoofing* al enviar un paquete a través de una ruta no autorizada.

21.20 *Multihoming de IPv6 y renumeración de red*

Una cuestión interesante es que, en vez de prohibir el *multihoming*, IPv6 lo promueve en el sentido que asume que un host IPv6 tiene varias conexiones y varias direcciones. Más importante aún, IPv6 permite a una organización asignar varios prefijos IPv6 a cada red.

La razón para permitir que una red tenga varios prefijos viene de la necesidad de *volver a enumerar* las redes. Si una organización cambia su proveedor de servicios, el prefijo de red asignado a la organización también puede cambiar. Los diseñadores de IPv6 querían que dichos cambios se realizaran con facilidad. Por lo tanto, los protocolos se diseñaron de modo que pudiera agregarse el nuevo prefijo mientras que las aplicaciones en ejecución seguían usando el prefijo anterior. Cuando se lanzara una aplicación, ésta usaría el nuevo prefijo. Despues de un corto tiempo todas las aplicaciones estarían usando el nuevo prefijo y podría quitarse el prefijo anterior. Tristemente, a pasar de varios años de trabajo en la reenumeración de redes en IPv6, la reenumeración automática todavía no es práctica.

21.21 Direccionamiento de IPv6

Al igual que IPv4, IPv6 asigna una dirección única para cada conexión entre una computadora y una red física. Por lo tanto, si se conecta un enrutador a tres redes físicas, a éste se le asignan al menos tres direcciones IPv6 (recuerde que IPv6 permite asignar varios prefijos a una red dada). Además, y al igual que IPv4, IPv6 separa cada dirección en un prefijo que identifica a la red y un sufijo que identifica a una computadora específica en la red.

A pesar de adoptar el mismo enfoque para asignar direcciones de computadora, el direccionamiento IPv6 difiere del direccionamiento IPv4 de varias formas considerables. Primero, los detalles de la dirección son totalmente distintos. Al igual que las direcciones CIDR, la división entre prefijo y sufijo en IPv6 puede ocurrir en un límite arbitrario de bits. Sin embargo y a diferencia de IPv4, IPv6 incluye direcciones con una jerarquía de tres niveles. Un prefijo inicial de la dirección es un valor globalmente único que se usa para el enrutamiento en Internet. Consideramos que el prefijo se asigna a una sola organización. La siguiente parte de la dirección identifica a una *subred* de la organización, mientras que la tercera parte de la dirección corresponde a una computadora específica de la red.

Al igual que los prefijos en IPv4, el prefijo de una dirección IPv6 es de tamaño variable y es elegida por un ISP, dependiendo del tamaño de un suscriptor. Sin embargo, la tercera parte de una dirección IP (la parte que identifica a una computadora específica) es fija. Por convención, la tercera parte usa 64 bits. Así, el prefijo global y la subred siempre forman un prefijo /64. Es decir, si un ISP asigna a una organización un prefijo global de K bits, la parte de las direcciones IPv6 correspondiente a la subred de la organización debe ser de $64-K$ bits de longitud. La figura 21.9 ilustra la división de una dirección IPv6.

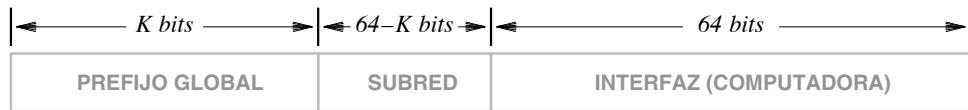


Figura 21.9 La división de una dirección IPv6 de 128 bits en secciones de prefijo, subred e interfaz. La interfaz tiene 64 bits de ancho.

Al igual que IPv4, IPv6 define un conjunto de direcciones especiales. Sin embargo, las dos versiones difieren por completo en cuanto al tipo de direcciones especiales utilizadas. Por ejemplo, IPv6 proporciona direcciones con *alcance limitado* (direcciones que sólo pueden usarse en una red y direcciones limitadas a una organización). Lo que es más importante, IPv6 no incluye direcciones especiales para difusión. En su lugar, IPv6 usa la multidifusión. Cada dirección IPv6 pertenece a uno de los tres tipos básicos que se enlistan en la figura 21.10.

Tipo	Propósito
unidifusión <i>(unicast)</i>	La dirección corresponde a una sola computadora. Un datagrama que se envía a la dirección se enruta a través de la ruta más corta hacia la computadora.
multidifusión <i>(multicast)</i>	La dirección corresponde a un conjunto de computadoras y la membresía del conjunto puede cambiar en cualquier momento. IPv6 entrega una copia del datagrama a cada miembro del grupo.
<i>anycast</i>	La dirección corresponde a un conjunto de computadoras que comparten un prefijo común. Un datagrama que se envía a la dirección se entrega precisamente a cualquiera de las computadoras (por ejemplo, la que esté más cerca del emisor).

Figura 21.10 Los tres tipos de direcciones IPv6.

Como se muestra en la figura, IPv6 retiene el direccionamiento de unidifusión y multidifusión. Para manejar una difusión limitada (difusión en la red local), IPv6 define un grupo de multidifusión especial que corresponde a todos los hosts y enruteadores en la red local.

El direccionamiento “anycast” se conocía en un principio como direccionamiento de *grupo*. La justificación de dicho direccionamiento surge de un deseo de permitir la replicación de servicios. Por ejemplo, una corporación que ofrece un servicio a través de la red asigna una dirección *anycast* a varias computadoras que proporcionan el servicio. Cuando un usuario envía un datagrama a la dirección *anycast*, IPv6 enruta el datagrama a una de las computadoras del conjunto. Si un usuario de otra ubicación envía un datagrama a la dirección *anycast*, IPv6 puede optar por enrutar el datagrama hacia un miembro diferente del grupo para permitir que ambas computadoras procesen las solicitudes al mismo tiempo.

21.22 Notación hexadecimal de dos puntos de IPv6

Puesto que una dirección IPv6 ocupa 128 bits, escribir dichos números puede convertirse en un proceso inmanejable. Por ejemplo, considere un número de 128 bits escrito en la notación decimal con puntos que usa IPv4:

105.220.136.100.255.255.255.0.0.18.128.140.10.255.255

Para ayudar a reducir el número de caracteres que se usan para escribir una dirección, los diseñadores del IPv6 eligieron una forma sintáctica más compacta conocida como *notación hexadecimal de dos puntos*, que por lo general se abrevia como *hex de dos puntos*. En la notación hexadecimal

de dos puntos, cada grupo de 16 bits se escribe en hexadecimal con un símbolo de dos puntos para separar los grupos. Por ejemplo, cuando el número anterior se escribe en hexadecimal de dos puntos, se convierte en:

69DC:8864:FFFF:FFFF:0:1280:8C0A:FFFF

Como el ejemplo ilustra, la notación hexadecimal de dos puntos requiere menos caracteres para expresar una dirección. Una optimización adicional conocida como *compresión cero* reduce aún más el tamaño. La compresión cero reemplaza las secuencias de ceros con símbolos de dos puntos. Por ejemplo, la dirección:

FF0C:0:0:0:0:0:B1

puede escribirse así:

FF0C::B1

El extenso espacio de direcciones de IPv6 y el esquema de asignación de direcciones propuesto hacen de la compresión cero algo realmente importante, ya que los diseñadores esperan que muchas direcciones IPv6 contengan cadenas de ceros. En especial, para ayudar a facilitar la transición hacia el nuevo protocolo, los diseñadores asignaron las direcciones IPv4 existentes en el espacio de direcciones de IPv6. Cualquier dirección de IPv6 que comience con 80 bits cero seguidos de 16 bits uno contiene una dirección IPv4 en los 32 bits de menor orden.

A diferencia del IPv4, el IPv6 no reserva direcciones de difusión especiales. En vez de ello, IPv6 usa un conjunto de direcciones multidifusión para manejar casos especiales. Por ejemplo, en vez de la dirección de difusión limitada de IPv4 en donde todos los bits son uno, IPv6 define una dirección multidifusión que corresponde a *todos los nodos en la red local*. IPv6 también define direcciones multidifusión que van más allá de los casos especiales de IPv4; por ejemplo, IPv6 define una dirección multidifusión para *todos los enrutadores de la red local*.

21.23 Resumen

Para dar la apariencia de una red grande sin fallas, Internet usa un esquema de direccionamiento uniforme. A cada computadora se le asigna una dirección IP única. Al comunicarse con la computadora, todas las aplicaciones de Internet usan el direccionamiento.

El *protocolo de Internet* se encarga de especificar el direccionamiento. IPv4 divide cada dirección de Internet en una jerarquía de dos niveles: un prefijo identifica la red a la que se conecta una computadora y un sufijo identifica a una computadora específica de la red; IPv6 usa una jerarquía de tres niveles para el prefijo, la subred y la computadora. Para asegurar que las direcciones sigan siendo únicas a lo largo de una interred determinada, una autoridad central asigna prefijos de red. En IPv4, una vez que se asigna un prefijo, un administrador de red local asigna a cada host de la red un sufijo único. Más adelante veremos que en IPv6, los sufijos de host únicos se pueden generar de manera automática.

Una dirección IPv4 es un número de 32 bits; una dirección IPv6 es un número de 128 bits. El esquema de direccionamiento original de IPv4 dividía las direcciones en clases. La clase de multidifusión IPv4 se sigue usando. El direccionamiento IPv4 sin clases y el de subredes permiten que el límite entre

prefijo y sufijo ocurría en un rango de bits arbitrario. Para ello, la subred y el direccionamiento sin clases (CIDR) almacenan una máscara de 32 bits junto con cada dirección. La máscara tiene el valor *1* para cada bit en el prefijo y el valor *0* para cada bit en el sufijo. IPv6 retiene el enfoque sin clases, pero usa una máscara de 128 bits.

IPv4 especifica un conjunto de direcciones reservadas que tienen un significado especial. Pueden usarse direcciones IPv4 especiales para especificar el ciclo de retorno o *loopback* (utilizado para hacer pruebas), la dirección de una red, la difusión en la red física local, así como la difusión en una red remota. IPv6 define un conjunto de direcciones de multidifusión, como podría ser una dirección para todos los nodos de una red y una dirección para todos los hosts de una red.

Aunque es conveniente pensar en una dirección IP como si especificara una computadora, cada dirección IP identifica una conexión entre una computadora y una red. Los enruteadores y los hosts multiproveedor, que tienen conexiones a varias redes físicas, tienen múltiples direcciones IP cada uno, a lo cual se le conoce como *multihoming*.

EJERCICIOS

- 21.1** ¿Podría el protocolo IP rediseñarse para usar direcciones de hardware en vez de las direcciones IP que usa actualmente? ¿Por qué sí o por qué no?
- 21.2** ¿Qué permite la jerarquía de direcciones de Internet que haga un administrador local?
- 21.3** En el esquema de direcciones IPv4 sin clases original, ¿era posible determinar la clase de una dirección a partir de la misma dirección? Explique.
- 21.4** Escriba un programa de computadora que acepte una dirección decimal con puntos como entrada y visualice una cadena de 32 bits.
- 21.5** Escriba un programa de computadora que acepte una dirección hexadecimal de dos puntos como entrada y visualice una cadena de 128 bits.
- 21.6** Escriba un programa de computadora que lea una dirección IPv4 en formato decimal con puntos y determine si la dirección es de multidifusión.
- 21.7** Escriba un programa de computadora que traduzca entre la notación CIDR con barra diagonal y un valor decimal con puntos equivalente.
- 21.8** Si un ISP le asignó un bloque de direcciones IPv4 /28, ¿a cuántas computadoras se les podría asignar una dirección de ese bloque?
- 21.9** Si un ISP le asignara un bloque de direcciones IPv6 /28, ¿a cuántas computadoras se les podría asignar una dirección de ese bloque?
- 21.10** Si un ISP ofrece un bloque de direcciones /17 por *N* dólares al mes y un bloque de direcciones /16 por 1.5 *N* dólares al mes, ¿cuál fue el costo más económico por computadora?
- 21.11** ¿Es el prefijo CIDR 1.2.3.4/29 válido? ¿Por qué sí o por qué no?
- 21.12** Suponga que usted es un ISP con un bloque de direcciones IPv4 /24. Explique si puede aceptar una solicitud de un cliente que necesita direcciones para 255 computadoras. (Sugerencia: considere las direcciones especiales).
- 21.13** Suponga que usted es un ISP que posee un bloque de direcciones IPv4 /22. Muestre la asignación de CIDR que usaría para asignar bloques de direcciones a cuatro clientes que necesitan direcciones para 60 computadoras cada uno.

- 21.14** Suponga que usted es un ISP que posee un bloque de direcciones IPv4 /22. ¿Puede aceptar solicitudes de seis clientes que necesitan direcciones para 9, 15, 20, 41, 128 y 260 computadoras, respectivamente? De ser así, ¿cómo? Si no, explique por qué.
- 21.15** Escriba un programa de computadora que lea una dirección IPv4 en una notación CIDR e imprima la dirección resultante y la máscara en binario.
- 21.16** Escriba un programa de computadora que lea como entrada un prefijo de red IPv4 en notación CIDR y una solicitud para un número de hosts. Suponga que la solicitud se dio a un ISP que posee el prefijo y asigne un prefijo de CIDR que acepte la solicitud sin desperdiciar direcciones.
- 21.17** Escriba un programa de computadora que lea una dirección de host IPv4 de 32 bits y una máscara de 32 bits en notación de CIDR, y que indique si la dirección es una de las direcciones especiales.
- 21.18** Escriba un programa de computadora que lea una dirección de host IPv6 de 128 bits y una máscara de 128 bits en notación CIDR, y que indique si la dirección es de multidifusión. (Sugerencia: la IETF publica estándares que especifican las asignaciones de direcciones IPv6).
- 21.19** ¿Qué es una dirección de difusión de Berkeley?
- 21.20** ¿Usa el IPv6 direcciones de difusión? Explique.
- 21.21** ¿Cuántas direcciones IPv4 se asignan a un enrutador que se conecta a N redes? Explique.
- 21.22** ¿Cuántas direcciones IPv6 pueden asignarse a un enrutador que se conecta a N redes? Explique.
- 21.23** ¿Puede un host tener más de una dirección IPv4? Explique.
- 21.24** Si un host IPv6 se conecta a cinco redes, ¿qué término se usa para describir al host?
- 21.25** ¿Cuándo podría ser útil una dirección *anycast*?

Contenido del capítulo

- 22.1 Introducción, 369
- 22.2 Servicio sin conexión, 369
- 22.3 Paquetes virtuales, 370
- 22.4 El datagrama IP, 370
- 22.5 El formato de encabezado del datagrama IPv4, 371
- 22.6 El formato de encabezado del datagrama IPv6, 373
- 22.7 Formato de encabezado base IPv6, 373
- 22.8 Reenvío de un datagrama IP, 375
- 22.9 Extracción de prefijos de red y reenvío de datagramas, 376
- 22.10 Coincidencia del prefijo más extenso, 377
- 22.11 Dirección de destino y dirección del siguiente salto, 378
- 22.12 Entrega del mejor esfuerzo, 378
- 22.13 Encapsulamiento de IP, 379
- 22.14 Transmisión a través de una interred, 380
- 22.15 MTU y fragmentación de datagramas, 381
- 22.16 Fragmentación de un datagrama IPv6, 383
- 22.17 Reensamblaje de un datagrama IP a partir de fragmentos, 384
- 22.18 Recolección de los fragmentos de un datagrama, 385
- 22.19 La consecuencia de la pérdida de fragmentos, 386
- 22.20 Fragmentación de un fragmento IPv4, 386
- 22.21 Resumen, 387

22

Reenvío de datagramas

22.1 Introducción

Los capítulos anteriores del libro describen la arquitectura y el direccionamiento de Internet. Este capítulo explica el servicio de comunicaciones fundamental en Internet. Describe el formato de los paquetes que se envían a través de Internet y habla sobre los conceptos clave del encapsulamiento de datos, reenvío, fragmentación y reensamblaje. Los capítulos posteriores amplían la explicación al considerar los protocolos adicionales que forman un servicio completo.

22.2 Servicio sin conexión

El objetivo de la interconexión de redes es ofrecer un sistema de comunicación de paquetes que permita a un programa que se ejecute en una computadora enviar datos a un programa que se ejecute en otra computadora. En una interred bien diseñada, los programas de aplicaciones no son conscientes de las redes físicas involucradas, sino que pueden enviar y recibir datos sin conocer los detalles de la red local a la que se conecta una computadora, la red remota a la que se conecta la computadora de destino o la interconexión entre ambas.

Una de las preguntas fundamentales que debemos considerar a la hora de diseñar una interred se relaciona con los servicios que se van a ofrecer. En especial, los diseñadores deben decidir entre ofrecer un servicio *orientado a la conexión*, un servicio *sin conexión* o ambos.

Los diseñadores de Internet eligieron incluir protocolos tanto para un servicio sin conexión como para un servicio orientado a la conexión. Ellos optaron por hacer que el servicio de entrega fundamental

sea sin conexión y agregar un servicio con conexión confiable que haga uso del servicio sin conexión. El diseño tuvo éxito y forma la base de toda la comunicación de Internet.

22.3 Paquetes virtuales

El servicio sin conexión es una extensión simple de la commutación de paquetes. El servicio permite a un emisor transmitir paquetes individuales de datos a través de Internet. Cada paquete viaja de manera independiente y contiene información que identifica al receptor deseado.

¿Cómo pasa un paquete a través de Internet? En general, la respuesta es que los enruteadores de Internet manejan la mayor parte del reenvío. Un host crea un paquete, coloca la dirección de destino en el encabezado del paquete y luego lo envía a un enruteador cercano. Cuando un enruteador recibe un paquete, usa la dirección de destino para seleccionar el siguiente enruteador disponible en la ruta hacia el destino y luego reenvía el paquete. En algún momento dado, el paquete llega a un enruteador que pueda entregarlo directamente a su destino final.

¿Qué formato se utiliza para un paquete de Internet? Como Internet consiste en redes heterogéneas que usan formatos de trama incompatibles, no es posible adoptar ninguno de los formatos de trama de hardware. Lo que es más importante, un enruteador no puede simplemente cambiar el formato del encabezado de la trama debido a que las dos redes podrían usar un direccionamiento incompatible (por ejemplo, las direcciones en una trama entrante tal vez no tengan sentido en otra red).

Para solucionar la heterogeneidad, el protocolo de Internet define un formato de paquete que es independiente del hardware utilizado. El resultado es un paquete *universal* y *virtual* que puede transferirse intacto a través del hardware involucrado. Como el término *virtual* implica, el formato de los paquetes de Internet no está enlazado directamente a ningún hardware. De hecho, el hardware no comprende ni reconoce un paquete de Internet. Como el término *universal* implica, cada host o enruteador en Internet contiene software de protocolo que reconoce los paquetes de Internet. Podemos resumir:

Puesto que incluye redes que son incompatibles, Internet no puede adoptar un formato específico de paquetes de hardware. Para adaptarse a la heterogeneidad, el protocolo de Internet define un formato de paquete que es independiente del hardware.

22.4 El datagrama IP

Los protocolos TCP/IP usan el nombre *datagrama de IP* para referirse a un paquete de Internet. Lo sorprendente es que un datagrama de IP tiene el mismo formato general que una trama de hardware: el datagrama comienza con un encabezado seguido de una *carga útil* (es decir, datos). La figura 22.1 ilustra el formato del datagrama.



Figura 22.1 La forma general de un datagrama de IP con un encabezado seguido de una carga útil.

Para resumir:

Un paquete enviado a través de una interred de TCP/IP se conoce como datagrama de IP. Cada datagrama consiste en un encabezado seguido de un área de datos, la cual se conoce como carga útil.

La cantidad de datos que se transportan en un datagrama no es fija. Un emisor selecciona una cantidad de datos apropiada para un fin específico. Por ejemplo, una aplicación que transmite pulsaciones de teclas a través de una red puede colocar cada pulsación en un datagrama independiente, mientras que una aplicación que transmite video por flujo continuo puede enviar datagramas grandes. En conclusión:

El tamaño de un datagrama se determina mediante la cantidad de datos que envía una aplicación. Al permitir que el tamaño de los datagramas varíe, el protocolo IP puede adaptarse a una variedad de aplicaciones.

En IPv4, un datagrama completo puede contener hasta 64K octetos. Sin embargo, el límite incluye los octetos del encabezado. Un datagrama IPv6 es ligeramente más grande, ya que puede transportar hasta 64K octetos de carga útil más un encabezado.

En la mayoría de los datagramas, el encabezado es mucho más pequeño que la carga útil. Incluso para las transacciones Web, una solicitud que envía un navegador es por lo general más grande que un encabezado de datagrama. Cuando un servidor Web devuelve datos, por lo general los datagramas contienen muchas veces más carga útil que encabezado. Cabe señalar que un encabezado representa sobrecarga. Si sólo se enviara un octeto de datos en cada datagrama, la mayor parte de la capacidad de la red se ocuparía en la transmisión de los encabezados. Por lo tanto, para maximizar el rendimiento de una transmisión (es decir, los bits de datos que se transfieren por segundo), las aplicaciones envían datagramas más grandes. Sin embargo, más adelante veremos que al enviar un datagrama demasiado grande también se pueden provocar problemas.

22.5 El formato de encabezado del datagrama IPv4

¿Qué contiene un encabezado de datagrama? Al igual que un encabezado de trama, un encabezado de datagrama contiene información que se utiliza para reenviar el datagrama. En especial, el encabezado contiene la dirección de origen (el emisor original), la dirección de destino (el receptor final) y un campo que especifica el tipo de datos que se van a transportar en el área de carga útil. Pero a diferencia de las tramas que se envían a través de una sola red, un datagrama no contiene direcciones MAC. En su defecto, cada dirección en el encabezado del datagrama es una dirección IP; las direcciones MAC no aparecen en el datagrama.

Cada campo de un encabezado de datagrama IPv4 tiene un tamaño fijo, lo cual hace eficiente el procesamiento del encabezado. La figura 22.2 muestra los campos de un encabezado de datagrama IPv4, y el texto que le sigue describe cada campo.



Figura 22.2 Campos en el encabezado de un datagrama de IP versión 4.

VERS. Un campo de versión de 4 bits que contiene el valor 4 para indicar el IPv4.

LON E. El campo longitud de encabezado de 4 bits especifica el número de cantidades de 32 bits que tiene el encabezado. Si no hay opciones presentes, el valor es 5.

TIPO SERVICIO. Un campo de 8 bits que transporta una clase de servicio para el datagrama (en la práctica, casi nunca se usa). El capítulo 27 proporciona los detalles.

LONGITUD TOTAL. Un entero de 16 bits que especifica el número total de bytes en el datagrama, incluyendo tanto el encabezado como la carga útil.

IDENTIFICACIÓN. Un número único de 16 bits (casi siempre secuencial) que se asigna al datagrama utilizado para recopilar todos los fragmentos para el reensamblaje.

BANDERAS. Un campo de 3 bits que especifica si el datagrama es un fragmento y, en caso de serlo, si el fragmento corresponde a la pieza del extremo derecho del datagrama original.

DESPLAZAMIENTO FRAGMENTO. Un campo de 13 bits que especifica en qué parte del fragmento del datagrama original pertenece este fragmento. El valor se multiplica por ocho para obtener un desplazamiento.

TIEMPO DE VIDA. Un entero de 8 bits que es reducido por cada enrutador que procesa el datagrama; si el valor llega a cero, el datagrama se desecha y se reporta un error.

TIPO. Un campo de 8 bits que especifica el tipo de la carga útil.

SUMA VERIF. ENCABEZADO. Una suma de verificación de 16 bits de complementos a 1 de los campos del encabezado, calculados de acuerdo con el algoritmo 8.1.[†]

DIRECCIÓN IP ORIGEN. La dirección IPv4 de 32 bits del emisor original (las direcciones de los enrutadores intermedios no aparecen en el encabezado).

[†] Encontrará el algoritmo 8.1 en la página 144.

DIRECCIÓN IP DESTINO. La dirección IPv4 de 32 bits del destino final (las direcciones de los enrutadores intermedios no aparecen en el encabezado).

Opciones IP. Las opciones IP (que pueden controlar el reenvío y el procesamiento de datagramas) casi nunca se usan, lo cual significa que la mayoría de las veces el campo se omite.

RELLENO. Si las opciones no terminan en un límite de 32 bits, se agregan cero bits para que el encabezado sea un múltiplo de 32 bits.

22.6 El formato de encabezado del datagrama IPv6

El IPv6 define un formato de encabezado de datagrama totalmente nuevo. En lugar de un encabezado fijo, el IPv6 divide la información del encabezado en un *encabezado base* y una serie de *encabezados de extensión* más pequeños que son opcionales. Un datagrama IPv6 consiste en un encabezado base, seguido de ninguno o más encabezados de extensión, seguidos a su vez de una carga útil. Por consiguiente, a diferencia de IPv4, donde el encabezado contiene campos fijos para cada pieza clave de información, IPv6 coloca muchas piezas clave de información en los encabezados de extensión, lo que significa que la mayoría de los datagramas contienen una secuencia de encabezados. La figura 22.3 ilustra la forma en que se organizan los datagramas de IPv6.



Figura 22.3 La forma general de un datagrama de IPv6.

Aunque la figura ilustra la estructura general, los campos no se dibujan a escala. En especial, algunos encabezados de extensión son más grandes que el encabezado base y otros más pequeños. En muchos datagramas, el tamaño de la carga útil es mucho mayor que el tamaño de los encabezados.

22.7 Formato de encabezado base IPv6

La figura 22.4 ilustra el formato de un encabezado base IPv6. Aunque tiene el doble de largo que un encabezado IPv4, el encabezado base IPv6 contiene menos información.

Como se indica en la figura, la mayor parte del espacio en el encabezado se dedica a los campos **DIRECCIÓN ORIGEN** y **DIRECCIÓN DESTINO**, cada uno de los cuales ocupa diecisésis octetos, cuatro veces más que una dirección IPv4. Al igual que en IPv4, la dirección de origen identifica a la fuente inicial y la dirección de destino identifica al receptor final.

Además de las direcciones de origen y de destino, el encabezado base contiene seis campos que se describen a continuación.

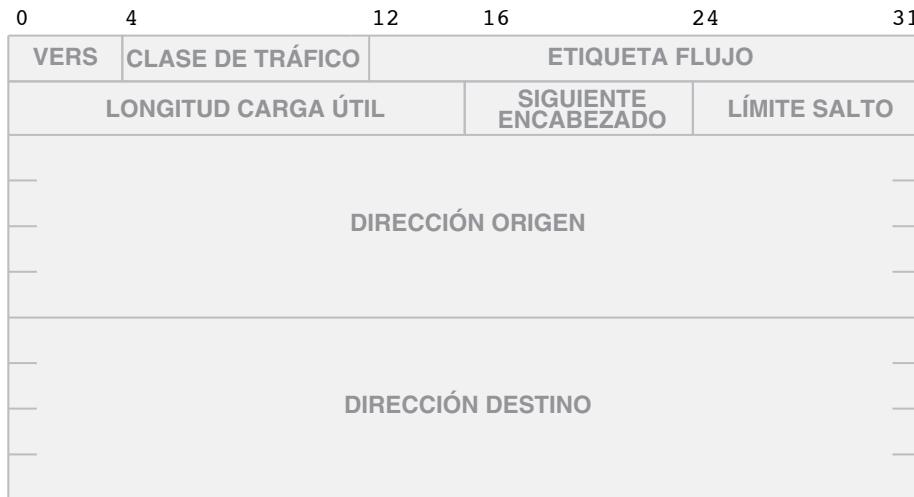


Figura 22.4 El formato del encabezado base en un datagrama IPv6.

VERS. Un campo de versión de 4 bits que contiene el valor 6 para indicar IPv6.

CLASE TRÁFICO. El campo de 8 bits especifica la clase de servicio para el datagrama, usando la misma definición que *TIPO SERVICIO* de IPv4.

ETIQUETA FLUJO. El campo de 20 bits destinado en un principio para asociar un datagrama con una ruta de conmutación de etiquetas (vea MPLS). Sin embargo, la conmutación de etiquetas ha entrado en desuso, por lo que *ETIQUETA FLUJO* se ha vuelto menos importante.

LONGITUD CARGA ÚTIL. El campo de 16 bits especifica el tamaño de la carga útil medido en octetos. A diferencia de IPv4, la *LONGITUD CARGA ÚTIL* únicamente especifica el tamaño de los datos que se van a llevar (es decir, la carga útil); se excluye el tamaño del encabezado.

SIGUIENTE ENCABEZADO. El campo de 8 bits especifica el tipo de información que sigue del encabezado actual, que puede ser un encabezado de extensión o la carga útil.

LÍMITE SALTO. Un campo de 8 bits con la misma semántica que el campo *TIEMPO DE VIDA* de IPv4; el valor es reducido por cada enrutador y el datagrama se desecha si el valor llega a cero.

DIRECCIÓN ORIGEN. La dirección IPv6 del emisor original.

DIRECCIÓN DESTINO. La dirección IPv6 del destino final.

Tanto el encabezado base como los encabezados de extensión contienen un campo *SIGUIENTE ENCABEZADO*. La figura 22.5 ilustra cómo se usan los campos para identificar piezas sucesivas de un datagrama IPv6. Algunos encabezados de extensión tienen un tamaño fijo (definido por el estándar del protocolo). Los encabezados que tienen un tamaño variable contienen un campo de longitud para especificar el tamaño de un datagrama dado. Por lo tanto, el software de una computadora receptora podrá saber con exactitud en dónde termina cada encabezado y nunca habrá ambigüedad.

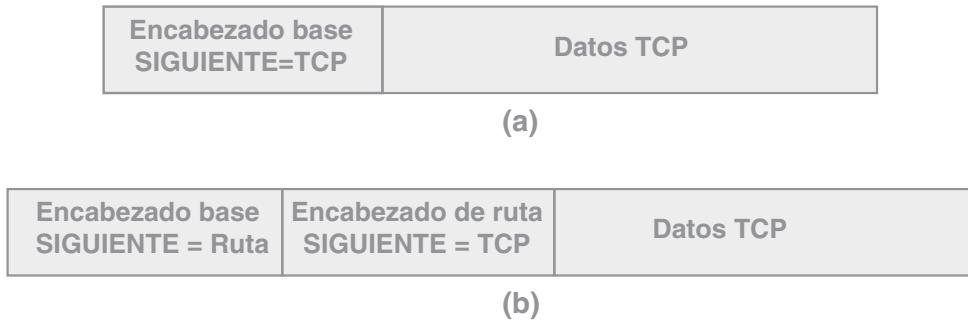


Figura 22.5 El campo SIGUIENTE ENCAZADO en (a) un datagrama IPv6 que tiene un encabezado base y carga útil de TCP, y (b) un datagrama con un encabezado base, un encabezado de ruta y una carga útil TCP.

22.8 Reenvío de un datagrama IP

Dijimos que un datagrama recorre Internet siguiendo una ruta desde su origen, pasando a través de un conjunto de enrutadores, hasta el destino final. Internet usa el reenvío del siguiente salto. Cada enrutador a lo largo de la ruta recibe el datagrama, extrae la dirección de destino del encabezado y usa la dirección de destino para determinar un siguiente salto hacia donde el datagrama debe enviarse. Después, el enrutador reenvía el datagrama al siguiente salto, ya sea el destino final u otro enrutador.

Para que la selección de un siguiente salto sea eficiente, un enrutador IP usa una *tabla de reenvío*. La tabla de reenvío se inicializa al momento del arranque del enrutador, y debe actualizarse si cambia la topología de Internet (por ejemplo, cuando una falla de hardware provoca que un enlace no pueda usarse). Por ahora consideraremos el reenvío cuando todo el hardware puede seguir funcionando; en capítulos posteriores consideraremos cómo se modifican las tablas de reenvío cuando surgen problemas.

En concepto, una tabla de reenvío contiene un conjunto de entradas, cada una de las cuales especifican un destino y el siguiente salto a usar para llegar a dicho destino. La figura 22.6 muestra una interred IPv4 de ejemplo y el contenido de una tabla de reenvío en uno de los tres enrutadores que se usan para interconectarse a las redes (las tablas IPv6 tienen direcciones más grandes). Lo importante a tener en cuenta es el tamaño de la tabla de reenvío. Cada entrada en una tabla de reenvío corresponde a una red, no a un host. Aunque hay mas de 800,000,000 de hosts en Internet, sólo hay cerca de 400,000 redes individuales. En conclusión:

Puesto que cada destino de una tabla de reenvío corresponde a una red, el número de entradas en la tabla es proporcional al número de redes en Internet, no al número de hosts.

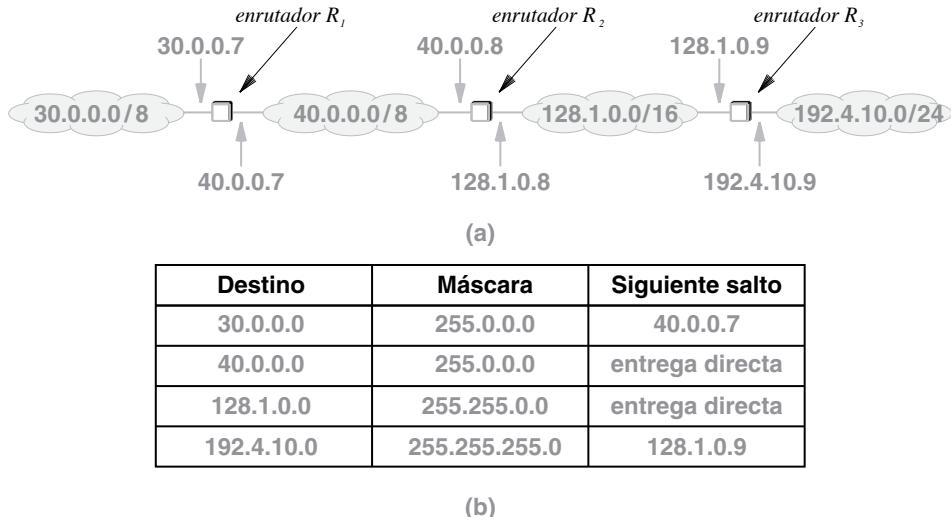


Figura 22.6 (a) Una interred IPv4 de ejemplo con cuatro redes y (b) la tabla de reenvío que se encuentra en el enrutador R_2 .

En la figura, a cada enrutador se le asignan dos direcciones IPv4, una para cada interfaz. Al enrutador R_2 que se conecta directamente a las redes $40.0.0.0/8$ y $128.1.0.0/16$, se le asignan las direcciones $40.0.0.8$ y $128.1.0.8$. Recuerde que IPv4 no requiere que el sufijo sea el mismo en todas las interfaces. Para facilitar la tarea a las personas que administran la red, el administrador de red del ejemplo eligió el mismo sufijo para cada interfaz.

22.9 Extracción de prefijos de red y reenvío de datagramas

Al proceso de usar una tabla de reenvío para seleccionar el siguiente salto para un datagrama dado, se le conoce como *reenvío*. En el capítulo 21 vimos que durante una búsqueda, se usa el campo *máscara* en la entrada de una tabla de reenvío para extraer de una dirección la parte correspondiente a la red. Cuando un enrutador encuentra un datagrama con la dirección IP de destino D , la función de reenvío debe encontrar una entrada en la tabla de reenvío que especifique un siguiente salto para D . Para ello, el software examina cada entrada en la tabla, usando la máscara en la entrada para extraer un prefijo de la dirección D y comparar el prefijo resultante con el campo *Destino* de la entrada. Si los dos son iguales, el datagrama se reenviará al *Siguiente salto* indicado en la entrada.

La representación de la máscara de bits hace eficiente a la extracción. El cálculo consiste en una operación *AND* booleana entre la máscara y la dirección de destino D . En concepto, el cálculo para examinar la i -ésima entrada en la tabla puede expresarse como:

```
if( (Máscara[i] & D) == Destino[i] ) reenviar a SiguienteSalto[i];
```

Como un ejemplo, considere un datagrama destinado para la dirección $192.4.10.3$ y suponga que el datagrama llega al enrutador del centro (R_2) de la figura 22.6. Suponga también que el procedimiento de reenvío busca en forma ordenada las entradas de la tabla. La primera entrada falla ya que $255.0.0.0$ & $192.4.10.3$ no es igual a $30.0.0.0$. Después de rechazar la segunda y tercera entradas en la tabla, el software de enrutamiento selecciona finalmente el siguiente salto $128.1.0.9$, ya que:

$$255.255.255.0 \& 192.4.10.3 == 192.4.10.0$$

22.10 Coincidencia del prefijo más extenso

La figura 22.6 contiene un ejemplo trivial. En la práctica, las tablas de reenvío de Internet pueden ser en extremo grandes, y el algoritmo de reenvío es más complejo de lo que parece. Por ejemplo, al igual que el reenvío de WAN descrito en el capítulo 18, las tablas de reenvío de Internet pueden contener una entrada *predeterminada* que proporciona una ruta para todos los destinos que no se enlistan de manera explícita. Además, el reenvío de Internet permite a un administrador dirigir el tráfico destinado a un host en particular por una *ruta específica de host* diferente a la de otros hosts en la misma red. En términos prácticos, la entrada en la tabla de reenvío para una ruta específica de host contiene una máscara que cubre toda una dirección completa (32 bits para IPv4 y 128 bits para IPv6).

Debido a que las máscaras de direcciones se pueden traslapar, entra en función una de las características importantes del reenvío de Internet. Por ejemplo, suponga que la tabla de reenvío de un enrutador contiene entradas para los siguientes dos prefijos de red IPv4:

128.10.0.0/16
128.10.2.0/24

Considere lo que ocurre si llega un datagrama destinado a la dirección 128.10.2.3. Aquí lo sorprendente es que el procedimiento de asociación antes descrito tiene éxito para ambas entradas. Es decir, una operación *AND* booleana de una máscara de 16 bits producirá 128.10.0.0 y la misma operación con una máscara de 24 bits producirá 128.10.2.0. Entonces, ¿qué entrada debería usarse?

Para resolver la ambigüedad que resulta de traslapar máscaras de dirección, el reenvío de Internet usa la *coincidencia del prefijo más extenso*. Es decir, en vez de examinar las entradas en orden arbitrario, el software de reenvío se dispone a examinar las entradas con el prefijo más extenso primero. En el ejemplo anterior, el reenvío de Internet elegirá la entrada que corresponda a 128.10.2.0/24. En conclusión:

Para resolver la ambigüedad que puede surgir cuando más de una entrada coincide con un destino, el reenvío de Internet examina las entradas con el prefijo más extenso primero.

22.11 Dirección de destino y dirección del siguiente salto

¿Cuál es la relación entre la dirección de destino en un encabezado de datagrama y la dirección del siguiente salto al que se reenvía el datagrama? El campo *DIRECCIÓN IP DESTINO* en un datagrama contiene la dirección del destino final y no cambia mientras el datagrama pasa a través de Internet. Cuando un enrutador recibe un datagrama, el enrutador usa el destino final D para calcular la dirección del siguiente enrutador al que debe enviarse el datagrama, N . Aunque el enrutador reenvía un datagrama al siguiente salto N , el encabezado en el datagrama retiene la dirección de destino D . En otras palabras:

La dirección de destino de un encabezado de datagrama siempre se refiere al destino final; en cada punto se calcula un siguiente salto, pero la dirección del siguiente salto no aparece en el encabezado del datagrama.

22.12 Entrega del mejor esfuerzo

Además de definir el formato de los datagramas de Internet, el protocolo de Internet define la semántica de la comunicación y usa el término *mejor esfuerzo* para describir el servicio que ofrece. Tanto IPv4 como IPv6 usan el paradigma del mejor esfuerzo. En esencia, el estándar especifica que aunque el IP realiza el mejor esfuerzo por entregar cada datagrama, no garantiza que se encargue de todos los problemas. En específico, el estándar IP reconoce que pueden ocurrir los siguientes problemas:

- Duplicación de datagramas
- Entrega retrasada o desordenada
- Corrupción de datos
- Pérdida de datagramas

Tal vez parezca extraño para el protocolo IP especificar que pueden ocurrir errores. Sin embargo, hay una razón importante: el protocolo IP está diseñado para funcionar sobre cualquier tipo de red. Sabemos de los capítulos anteriores que el equipo de red puede experimentar interferencia por el ruido, lo que puede provocar corrupción o pérdida. En un sistema en el que las rutas pueden cambiar, los paquetes que siguen una ruta pueden tardar más que los que siguen otra, lo cual puede provocar una entrega desordenada. En conclusión:

Puesto que el protocolo IP está diseñado para operar sobre todos los tipos de hardware de red, incluyendo el hardware que experimenta problemas, los datagramas de IP pueden perderse, duplicarse, retrasarse, entregarse desordenados o con datos corruptos.

Por fortuna, veremos que la suite de protocolos TCP/IP incluye protocolos adicionales que manejan muchos de los problemas. También aprenderemos que algunas aplicaciones prefieren usar un servicio del mejor esfuerzo en vez de un servicio que detecte y corrija problemas.

22.13 Encapsulamiento de IP

¿Cómo puede transmitirse un datagrama a través de una red física que no comprende el formato de datagramas? La respuesta recae en una técnica conocida como *encapsulamiento*. Cuando un datagrama IP se encapsula en una trama, todo el datagrama se coloca en el área de carga útil de la misma. El hardware de red trata a una trama que contiene un datagrama de la misma forma como lo haría con cualquier otra trama. De hecho, el hardware no examina ni cambia el contenido de la carga útil. La figura 22.7 ilustra el concepto.



Figura 22.7 Ilustración de un datagrama de IP encapsulado en una trama.

¿Cómo sabe un receptor si la carga útil de una trama entrante contiene un datagrama de IP u otros datos? El emisor y el receptor deben estar de acuerdo en cuanto al valor que se utilice en el campo del tipo de trama. Al colocar un datagrama en una trama, el software en la computadora emisora asigna un valor al campo de tipo de trama; cuando llega la trama, el receptor sabe que el área de carga útil contiene un datagrama de IP. En el capítulo 15 vimos que el estándar Ethernet especifica un tipo de trama de *0x0800* para una trama que transporta un datagrama IPv4, y *0x08DD* para una trama que transporta un datagrama IPv6.

Una trama que transporta un datagrama de IP también debe tener una dirección MAC de destino. Por lo tanto, además de colocar un datagrama en el área de carga útil de una trama, el encapsulamiento requiere que el emisor suministre la dirección MAC de la siguiente computadora a la que debe enviarse el datagrama. Para calcular la dirección apropiada, el protocolo IP de la computadora emisora debe vincular la dirección IP del siguiente salto con una dirección MAC equivalente, que es el destino en el encabezado de la trama.[†] Podemos resumir esto así:

Para transmitirse a través de una red física, un datagrama se encapsula en una trama. La dirección de destino de la trama es la dirección MAC del siguiente salto al que se está enviando el datagrama. La dirección se obtiene traduciendo la dirección IP del siguiente salto a una dirección MAC equivalente.

[†] El capítulo 23 describe cómo se obtiene la dirección MAC.

22.14 Transmisión a través de una interred

El encapsulamiento se aplica una red a la vez. Una vez que el emisor selecciona un siguiente salto, encapsula el datagrama en una trama y transmite el resultado a través de la red física. Cuando la trama alcanza el siguiente salto, el software receptor quita el datagrama de IP y desecha la trama. Si hay que reenviar el datagrama a través de otra red, se crea una nueva trama. La figura 22.8 ilustra cómo se encapsula y desencapsula un datagrama a medida que viaja de un host de origen hacia uno de destino a través de tres redes y dos enrutadores. Cada red puede usar una tecnología de hardware diferente que las otras, lo que significa que los formatos de trama y los tamaños de los encabezados de trama pueden diferir.

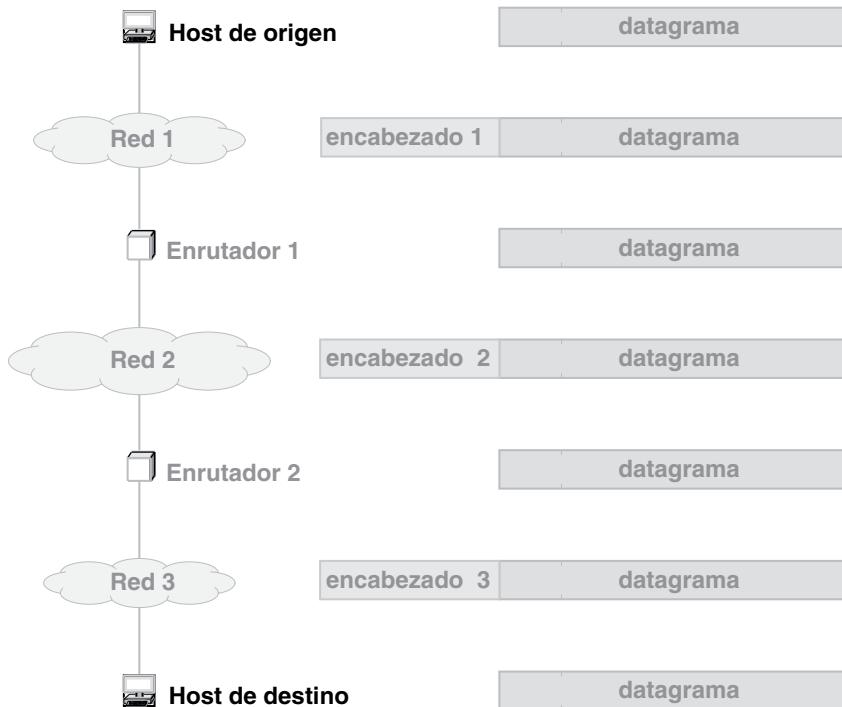


Figura 22.8 Un datagrama IP a medida que viaja a través de Internet.

Como se indica en la figura, los hosts y los enrutadores almacenan en la memoria un datagrama sin un encabezado adicional. Cuando el datagrama pasa a través de una red física, se encapsula en una trama adecuada para la red. El tamaño del encabezado de trama que aparece antes del datagrama depende de la tecnología de red. Por ejemplo, si la Red 1 representa una Ethernet, el encabezado en la trama 1 es un encabezado de Ethernet. De manera similar, si la Red 2 representa una red Wi-Fi, el encabezado en la trama 2 corresponde a un encabezado Wi-Fi.

Es importante tener en cuenta que los encabezados de trama no se acumulan durante un viaje por Internet. Cuando llega un datagrama, éste se elimina de la trama entrante antes de ser encapsulado en una trama saliente. De esta forma, cuando el datagrama llega a su destino final, el único encabezado de trama en el datagrama es el encabezado de la última red a través de la cual llegó. Una vez que se elimina el encabezado, el resultado es el datagrama original. En conclusión:

Cuando un datagrama llega en una trama de red, el receptor extrae el datagrama del área de carga útil de la trama y desecha el encabezado de trama.

22.15 MTU y fragmentación de datagramas

Cada tecnología de hardware especifica la cantidad máxima de datos que una trama puede transportar. El límite se conoce como *unidad máxima de transmisión (MTU)*. No hay excepción para el límite de la MTU: el hardware de red no está diseñado para aceptar o transferir tramas que transporten más datos de lo que la MTU permite. Por lo tanto, un datagrama debe ser más pequeño o igual a la MTU de la red, o no podrá encapsularse para la transmisión.

En una interred que contenga redes heterogéneas, las restricciones de la MTU crean un problema. En particular, puesto que un enrutador puede conectar redes con distintos valores de MTU, un datagrama recibido a través de una red puede ser demasiado grande como para ser enviado por otra red. Por ejemplo, la figura 22.9 ilustra un enrutador que interconecta dos redes con valores MTU de 1500 y 1000.

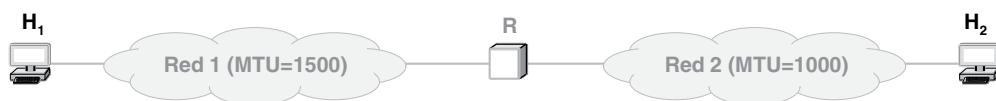


Figura 22.9 Ilustración de un enrutador que conecta dos redes con distintas MTU.

En la figura, el host H_1 se conecta a una red con una MTU de 1500 y puede enviar un datagrama de hasta 1500 octetos. El host H_2 se conecta a una red que tiene una MTU de 1000, lo que significa que no puede enviar o recibir un datagrama mayor de 1000 octetos. Si el host H_1 envía un datagrama de 1500 octetos al host H_2 , el enrutador R no podrá encapsular el datagrama para transmitirlo a través de la Red 2.

Para resolver el problema de las MTU heterogéneas, tanto IPv4 como IPv6 usan una técnica conocida como *fragmentación*. El datagrama se divide en piezas más pequeñas, conocidas como *fragmentos*, y cada fragmento se transmite en una trama independiente. Sin embargo, IPv6 cambió la forma y el tiempo en que se lleva a cabo la fragmentación:

- IPv4 hace que un enrutador realice la fragmentación según sea necesario.
- IPv6 requiere que el host emisor realice la fragmentación.

Vamos a considerar primero la fragmentación de IPv4. Aquí la fragmentación sólo se realiza de ser necesario; es decir, no ocurre ninguna acción hasta que un enrutador encuentra un datagrama más grande que la MTU de la red sobre la que se debe enviar. En ese punto, el enrutador divide el datagrama en fragmentos y envía cada fragmento de manera independiente.

Lo sorprendente es que un fragmento IPv4 tiene el mismo formato que otros datagramas de IPv4: un bit en el campo *BANDERAS* del encabezado indica si un datagrama es un fragmento o un datagrama completo[†]. A otros campos del encabezado se les asigna información que el destino final usa para *reensamblar* los fragmentos y reproducir el datagrama original. En especial, el campo *COMPENSACIÓN FRAGMENTO* del encabezado de un fragmento especifica a qué parte del datagrama original pertenece ese fragmento.

Para fragmentar un datagrama y transmitirlo a través de una red, un enrutador usa la MTU de la red y el tamaño del encabezado para calcular la cantidad máxima de datos que pueden enviarse en cada fragmento, además del número de fragmentos que se necesitarán. Después, el enrutador crea los fragmentos. Usa los campos del encabezado original para crear un encabezado de fragmento. Es decir, el enrutador copia en el encabezado del fragmento los campos *IP ORIGEN* e *IP DESTINO* del datagrama. Por último, el enrutador copia en el fragmento los datos necesarios del datagrama original y transmite el resultado. La figura 22.10 muestra la división.



Figura 22.10 Un datagrama IPv4 dividido en tres fragmentos; el fragmento final es más pequeño que los otros.

Para resumir:

Cada red tiene una MTU que especifica la cantidad máxima de datos que puede transportar una trama. Cuando un enrutador recibe un datagrama IPv4 que es mayor que la MTU de la red a través de la cual se va a enviar, éste divide el datagrama en piezas más pequeñas conocidas como fragmentos. Cada fragmento usa el formato de datagrama IPv4, pero sólo transporta una parte de la carga útil original.

[†] Encontrará el formato de encabezado de datagrama IPv4 en la figura 22.2 de la página 372.

22.16 Fragmentación de un datagrama IPv6

Aunque la fragmentación del IPv6 se asemeja a la fragmentación del IPv4, los detalles difieren. Al igual que IPv4, se copia un prefijo del datagrama original en cada fragmento y se modifica la longitud de la carga útil para que sea igual a la longitud del fragmento. Sin embargo y a diferencia de IPv4, IPv6 no incluye campos para la información de fragmentación en el encabezado base. En su lugar, IPv6 coloca la información de fragmentación en un encabezado de extensión de fragmento independiente; la presencia del encabezado de extensión identifica el datagrama como un fragmento. Lo interesante es que el encabezado de extensión de fragmento de IPv6 contiene la misma información de fragmento que se encuentra en los campos de un encabezado de IPv4.

La fragmentación de IPv6 también difiere de IPv4 debido a que IPv6 usa varios encabezados de extensión. Los enruteadores intermedios usan algunos de los encabezados de extensión y otros no. Por lo tanto, IPv6 divide los encabezados en dos grupos, conocidos como *fragmentables* y *no fragmentables*. Los encabezados fragmentables se dividen en fragmentos como la carga útil, y los encabezados no fragmentables se copian en cada fragmento. La figura 21.11 ilustra la fragmentación de IPv6.

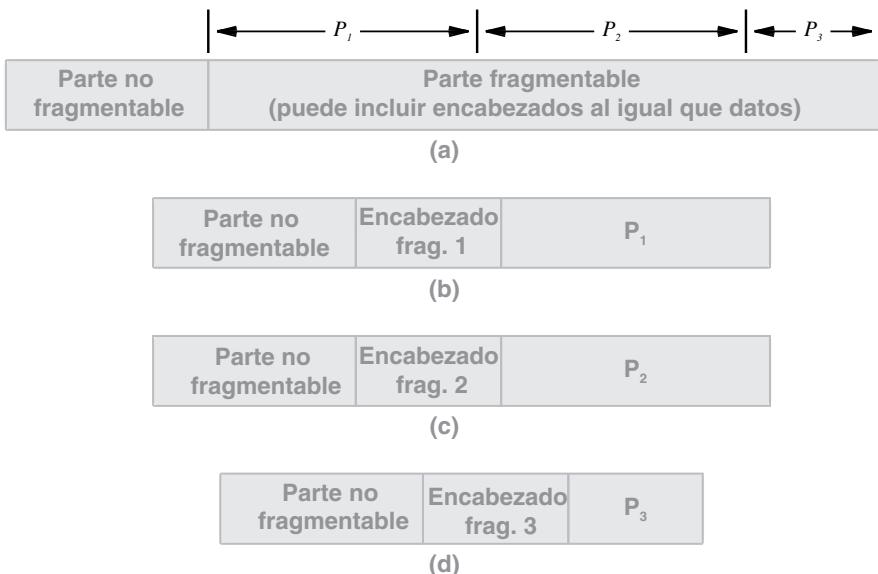


Figura 22.11 Ilustración de la fragmentación de IPv6 con un datagrama (a) dividido en los fragmentos (b) hasta (d).

En la figura, la *parte no fragmentable* indica el encabezado base más los encabezados que utilizan los enruteadores intermedios. Para asegurar que todos los fragmentos se enruten de manera idéntica, la parte no fragmentable se replica en cada fragmento.

Al igual que con el IPv4, el tamaño del fragmento se selecciona de acuerdo con la unidad máxima de transmisión (MTU) de la red a través de la cual deben enviarse los fragmentos. De esta forma, el fragmento final puede ser más pequeño que los otros debido a que contiene el residuo después de haber extraído del datagrama original las piezas del tamaño de la MTU.

La fragmentación en el IPv6 difiere drásticamente de la fragmentación en el IPv4. En este último, cuando un enrutador recibe un datagrama demasiado grande para la red a través de la cual debe enviarse, realiza la fragmentación. En IPv6, la fuente original es responsable de la fragmentación. Es decir, se espera que los hosts seleccionen un tamaño de datagrama que no requiera fragmentación. Un enrutador que recibe un datagrama mayor que la MTU de la red enviará un mensaje de error y desechará el datagrama.

¿Cómo puede un host elegir un tamaño de datagrama que no produzca fragmentación? El host debe aprenderse la MTU de cada red junto con la ruta al destino, y debe elegir un tamaño de datagrama que se adapte al valor más pequeño. La MTU mínima a lo largo de una ruta desde el origen hasta un destino se conoce como la *MTU de ruta*, y el proceso de aprender la MTU de ruta se conoce como *descubrimiento de MTU de ruta*. En general, el descubrimiento de la MTU de ruta es un procedimiento iterativo. Un host selecciona un tamaño de datagrama inicial que parece apropiado (por ejemplo, la MTU de Ethernet de 1500 es una elección común) y comienza a enviar al destino. Si un enrutador necesita enviar un datagrama IPv6 a través de una red que tenga una MTU más pequeña que el datagrama, el enrutador envía un mensaje de error a la fuente original y desecha el datagrama. El origen fragmenta el datagrama en un tamaño más pequeño e intenta de nuevo. Finalmente el origen descubre la *MTU de ruta* (es decir, la MTU mínima a lo largo de una ruta del emisor al destino) y hace a todos los datagramas sucesivos más pequeños. Para resumir:

En IPv6 la fragmentación la realiza el host emisor y no los enrutadores. Si se requiere fragmentación, el host emisor recibe un mensaje de error de ICMP y reduce el tamaño de los fragmentos hasta que éstos puedan enviarse al destino.

22.17 Reensamblaje de un datagrama IP a partir de fragmentos

Al proceso de recrear una copia del datagrama original a partir de fragmentos se le conoce como *reensamblaje*. Un receptor sabe si un datagrama entrante es un fragmento (ya sea por el campo BANDERAS de IPv4 o por la presencia de un encabezado de extensión de fragmento IPv6). Todos los fragmentos de un datagrama dado tienen la misma dirección de destino que el datagrama original del que se derivan. El fragmento que transporta la pieza original de datos tiene un bit adicional establecido. Por lo tanto, un host que realiza el reensamblaje puede saber si todos los fragmentos llegaron con éxito.

Lo interesante es que el protocolo IP especifica que los enrutadores intermedios no deben reensamblar los datagramas. En su lugar, el destino final es responsable de reensamblar los fragmentos. Por ejemplo, considere la configuración de la figura 22.12.

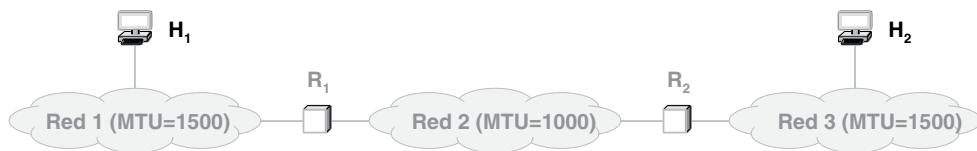


Figura 22.12 Ilustración de tres redes conectadas por dos enrutadores.

En la figura, si el host H_1 envía un datagrama IPv4 de 1500 octetos al host H_2 , el enrutador R_1 dividirá el datagrama en dos fragmentos, que reenviará a R_2 . Si el host envía un datagrama IPv6, el host fragmenta el datagrama y reenvía los fragmentos a R_2 . El enrutador R_2 no vuelve a ensamblar los fragmentos, sino que usa la dirección de destino de un fragmento para reenviarlo como siempre. El host de destino final H_2 recolecta los fragmentos y los vuelve a ensamblar para producir el datagrama original.

Requerir que el destino final vuelva a ensamblar los fragmentos tiene dos ventajas.

- Primero, reduce la cantidad de información de estado en los enrutadores. Cuando un enrutador reenvía un datagrama, no necesita saber si éste es un fragmento o no.
- Segundo, permite que las rutas cambien en forma dinámica. Si un enrutador intermedio fuera a reensamblar los fragmentos, todos ellos necesitarían llegar al enrutador.

Al posponer el reensamblaje hasta el destino final, IP puede pasar algunos fragmentos de un datagrama a lo largo de rutas distintas a las de otros fragmentos. Es decir, Internet puede cambiar las rutas en cualquier momento (por ejemplo, para rodear una falla de hardware).

22.18 Recolección de los fragmentos de un datagrama

Cabe recordar que IP no garantiza la entrega. Por consiguiente, los fragmentos individuales (que se reenvían igual que otros datagramas) pueden perderse o llegar desordenados. Lo que es más importante, si una fuente dada envía varios datagramas al mismo destino, los fragmentos de varios de ellos pueden llegar en orden arbitrario.

¿Cómo reensambla el software de IP los fragmentos que llegan desordenados? Un emisor coloca un número de identificación único en el campo *IDENTIFICACIÓN* de cada datagrama saliente (ya sea en el encabezado IPv4 o en el encabezado de extensión de fragmento IPv6). Cuando se fragmenta un datagrama, el número de identificación se copia en cada fragmento. Un receptor usa el número de identificación y la dirección IP de origen de un fragmento entrante para determinar el datagrama al que pertenece el fragmento. Además, el campo *COMPENSACIÓN FRAGMENTO* (en el encabezado de datagrama IPv4 o el encabezado de extensión de fragmento IPv6) indica a un receptor en qué lugar del datagrama original va la carga útil del fragmento.

22.19 La consecuencia de la pérdida de fragmentos

Dijimos que IP no garantizaba la entrega de los fragmentos. Si una de las redes de la ruta descarta paquetes, podría perderse un datagrama o un fragmento encapsulado. Un datagrama no se puede reensamblar sino hasta que lleguen todos los fragmentos. Por lo tanto, cuando llegan uno o más fragmentos de un datagrama pero se retrasan o pierden otros, surge un problema. Aunque el datagrama no puede volver a ensamblarse, el receptor debe guardar los fragmentos que llegan en caso de que los fragmentos faltantes sólo estén retrasados.

Puesto que ocupan espacio en memoria, un receptor no puede retener los fragmentos por mucho tiempo. Para evitar agotar la memoria, el protocolo IP especifica un tiempo máximo para retener los fragmentos. Cuando llega el primer fragmento de un datagrama específico, el receptor inicia un *temporizador de reensamblaje*. Si llegan todos los fragmentos de un datagrama antes de que el temporizador expire, el receptor cancela el temporizador y vuelve a ensamblar el datagrama. Pero si el temporizador expira antes de que lleguen todos los fragmentos, el receptor desecha los fragmentos que llegaron.

El resultado del temporizador de reensamblaje del protocolo IP es todo o nada: o llegan todos los fragmentos y el protocolo IP vuelve a ensamblar el datagrama, o IP descarta el datagrama incompleto. No hay un mecanismo para que un receptor indique al emisor qué fragmentos llegaron. La consecuencia es:

En una red con pérdidas, como una LAN inalámbrica que experimenta interferencia, la probabilidad de perder un datagrama es mayor si éste se encuentra fragmentado. Por lo tanto y como regla general, hay que evitar la fragmentación siempre que sea posible.

22.20 Fragmentación de un fragmento IPv4

En IPv6, los fragmentos deben ser capaces de recorrer toda la ruta desde el origen hasta el destino. Pero en IPv4 puede surgir un problema secundario. Después de fragmentar un datagrama de IPv4, un enrutador reenvía cada fragmento a su destino, pero el fragmento podría en un momento dado llegar a una red que tenga una MTU más pequeña. El esquema de fragmentación de IPv4 se planeó con cuidado para que fuera posible fragmentar un fragmento. Un enrutador a lo largo de la ruta divide el fragmento en fragmentos más pequeños. Si las redes se ordenan en una secuencia de MTU cada vez menores, cada enrutador a lo largo de la ruta debe fragmentar aún más cada fragmento. Desde luego que los diseñadores trabajan con cuidado para asegurar que dichas situaciones no ocurran en Internet.

IPv4 no distingue entre fragmentos originales y subfragmentos. En particular, un receptor no puede saber si un fragmento entrante es el resultado de que un enrutador haya fragmentado un datagrama o que varios enrutadores fragmentaran los fragmentos. La ventaja de formar todos los fragmentos iguales es que un receptor puede realizar el reensamblaje del datagrama original sin tener que reensamblar primero los subfragmentos. Al hacer esto se ahorra tiempo de CPU y se reduce la cantidad de información necesaria en el encabezado de cada fragmento.

22.21 Resumen

El protocolo de Internet define un datagrama IP como la unidad básica de transferencia a través de una interred TCP/IP. Puesto que el datagrama contiene un encabezado seguido de un área de carga útil, cada datagrama se asemeja a una trama de hardware, y al igual que ésta, el encabezado de un datagrama contiene la información que se usa para transferir el datagrama a un destino específico. Pero a diferencia de una trama de hardware, un encabezado de datagrama contiene direcciones IP en vez de direcciones MAC.

El software de IP de los enrutadores usa una tabla de rutas para determinar el siguiente salto hacia el que debe enviarse un datagrama. Cada entrada en una tabla de reenvío corresponde a una red de destino, lo que significa que el tamaño de una tabla de reenvío es proporcional al número de redes en Internet. Al seleccionar una ruta, IP compara el prefijo de red de una dirección de destino con cada entrada en la tabla. Para evitar la ambigüedad, IP especifica que si una tabla de reenvío contiene dos entradas que coinciden con un destino específico, el reenvío debe asociar el prefijo más extenso.

Aunque IP selecciona un siguiente salto hacia el que debe enviarse un datagrama, la dirección del siguiente salto no aparece en el encabezado del datagrama. En su lugar, el encabezado siempre especifica la dirección del destino final.

Un datagrama de IP se encapsula en una trama para la transmisión. Cada tecnología de red especifica una MTU (unidad máxima de transmisión), que es el tamaño máximo de carga útil permitido. Cuando un datagrama excede la MTU de la red, el protocolo IP fragmenta el datagrama. IPv4 permite a los enrutadores a lo largo de una ruta realizar la fragmentación; IPv6 requiere que el host emisor realice la fragmentación. De ser necesario, un fragmento de IPv4 puede fragmentarse aún más. El destino final reensambla los fragmentos, usando un temporizador para desechar un datagrama si uno o más fragmentos se pierden.

EJERCICIOS

- 22.1** ¿Cuáles son los dos paradigmas básicos de comunicación que los diseñadores consideran al diseñar una interred?
- 22.2** ¿Cómo se adapta el diseño de Internet a las redes heterogéneas, cada una de las cuales con su propio formato de paquetes?
- 22.3** Escriba un programa de computadora para extraer las direcciones de origen y de destino de un datagrama IPv4 e imprímalas en notación decimal con puntos.
- 22.4** Escriba un programa de computadora para extraer las direcciones de origen y de destino de un datagrama IPv6 e imprímalas en notación hexadecimal de dos puntos.
- 22.5** Escriba un programa para extraer todos los campos de un encabezado de datagrama IPv4 o IPv6. Imprima los valores en hexadecimal, decimal con puntos o notación hexadecimal de dos puntos, según sea apropiado.
- 22.6** ¿Cuál es la longitud máxima de un datagrama de IPv4?
- 22.7** Escriba un programa de computadora que reciba como entrada una tabla de reenvío de IP similar a la que se muestra en la figura 22.6(b) y una secuencia de direcciones de destino. Para cada dirección de destino, busque en la tabla secuencialmente para encontrar el siguiente salto correcto e imprima los resultados.

- 22.8** Si la carga útil de un datagrama de IPv4 contiene un valor de datos de 8 bits y ninguna opción de encabezado, ¿qué valores se encontrarán en los campos de encabezado *LON E.* y *LONGITUD TOTAL*?
- 22.9** Si dos prefijos de una tabla de reenvío coinciden con una dirección de destino dada, ¿cuál usará el algoritmo de reenvío?
- 22.10** ¿Una dirección de destino en un datagrama IP hace referencia en algún momento a un enrutador intermedio? Explique.
- 22.11** Suponga que dos enrutadores están mal configurados para formar un ciclo de enrutamiento para cierto destino, *D*. Explique por qué un datagrama destinado para *D* no se quedará indefinidamente en el bucle.
- 22.12** ¿Qué problemas pueden ocurrir a medida que un datagrama de IP pasa a través de Internet?
- 22.13** ¿En qué parte de la trama viaja un datagrama de IP?
- 22.14** Si capturamos un datagrama de IP que pasa a través de una red de Internet, ¿cuántos encabezados de trama aparecerán antes del datagrama?
- 22.15** ¿Cuál es la MTU de una red?
- 22.16** Si un datagrama de IPv4 con una carga útil de 1480 bytes debe enviarse a través de una red con una MTU de 500 bytes, ¿cuántos fragmentos se enviarán? Explique.
- 22.17** Si un datagrama de IPv6 con una carga útil de 1480 bytes y ningún encabezado de extensión debe enviarse a través de una red con una MTU de 500 bytes, ¿cuántos fragmentos se enviarán? Explique.
- 22.18** En Internet, ¿dónde se vuelven a ensamblar los fragmentos?
- 22.19** Al reensamblar fragmentos, ¿cómo sabe el software de IP si los fragmentos entrantes pertenecen al mismo datagrama?
- 22.20** Si se pierde un fragmento, ¿solicita un receptor una nueva copia? Explique.
- 22.21** Busque en Web y lea las especificaciones RFC 1149 y 1217. ¿Son estándares de red serios? (Sugerencia: considere las fechas).

Contenido del capítulo

- 23.1 Introducción, 391
- 23.2 Resolución de direcciones, 391
- 23.3 Un ejemplo de direcciones IPv4, 393
- 23.4 El protocolo de resolución de direcciones (ARP) IPv4, 393
- 23.5 Formato de mensajes del ARP, 394
- 23.6 Encapsulamiento del ARP, 395
- 23.7 Uso de caché y procesamiento de mensajes del ARP, 396
- 23.8 El límite conceptual de direcciones, 398
- 23.9 Protocolo de mensajes de control de Internet (ICMP), 399
- 23.10 Formato de mensajes y encapsulamiento del ICMP, 400
- 23.11 Vinculación de direcciones IPv6 con descubrimiento del vecindario, 401
- 23.12 Software de protocolo, parámetros y configuración, 401
- 23.13 Protocolo de configuración dinámica de host (DHCP), 402
- 23.14 Operación del protocolo DHCP y optimizaciones, 403
- 23.15 Formato de mensajes del DHCP, 404
- 23.16 Acceso indirecto a un servidor DHCP por medio de un retransmisor, 405
- 23.17 Configuración automática de IPv6, 405
- 23.18 Traducción de direcciones de red (NAT), 406
- 23.19 Operación de NAT y direcciones IPv4 privadas, 407
- 23.20 NAT de la capa de transporte (NAPT), 409
- 23.21 NAT y los servidores, 410
- 23.22 Software y sistemas NAT para usar en casa, 410
- 23.23 Resumen, 411

23

Protocolos y tecnologías de soporte

23.1 Introducción

Los capítulos de esta parte del libro hablan sobre Internet y las tecnologías de protocolos relacionadas. Los capítulos anteriores cubren los conceptos básicos como el direccionamiento, el formato de datagramas y el reenvío de IP, así como el encapsulamiento, la fragmentación y el reensamblaje.

En este capítulo continuamos la explicación de la interconexión de redes mediante la introducción de cuatro tecnologías de soporte clave: la vinculación de direcciones, el reporte de errores, el arranque o *bootstrapping*, y la traducción de direcciones. Cada tecnología se encarga de un problema pequeño. Al combinarse con otros protocolos, cada una realiza una contribución considerable en cuanto a la funcionalidad de Internet en general. Los siguientes capítulos amplían la explicación sobre la interconexión de redes con un enfoque en los protocolos de la capa de transporte y los protocolos de enrutamiento de Internet.

23.2 Resolución de direcciones

En el capítulo 22 vimos que a medida que un datagrama viaja por Internet, el emisor inicial y cada enrutador a lo largo de la ruta usan la dirección IP de destino del datagrama para seleccionar una dirección del siguiente salto, luego encapsulan el datagrama en una trama de hardware y transmiten la trama a través de una red. Un paso crucial del proceso de reenvío requiere de una traducción, ya que el reenvío usa direcciones IP y una trama que se transmite a través de una red física debe contener la dirección MAC del siguiente salto. Por consiguiente, el software del protocolo IP debe traducir la dirección IP del siguiente salto en una dirección MAC equivalente. El principio es el siguiente:

Las direcciones IP son abstracciones que proporciona el software del protocolo. Puesto que el hardware de la red física no entiende las direcciones IP, una dirección IP del siguiente salto debe traducirse en una dirección MAC equivalente antes de poder enviar una trama.

Al proceso de traducir una dirección IP de una computadora en una dirección de hardware equivalente, se le conoce como *resolución de direcciones*. Decimos que una dirección IP se *resuelve* en la dirección MAC correcta. La resolución de direcciones siempre se limita a una sola red a la vez. Una computadora puede resolver la dirección de otra computadora sólo si ambas están conectadas a la misma red física. Una computadora nunca tendrá que resolver la dirección de una computadora que está en una red remota. Por ejemplo, considere la interred simple de la figura 23.1.

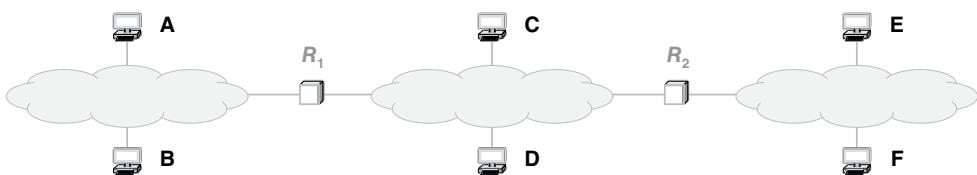


Figura 23.1 Una interred de ejemplo de tres redes y las computadoras conectadas a cada una.

En la figura, si el enrutador R_1 reenvía un datagrama al enrutador R_2 , el enrutador R_1 resolverá la dirección IP de R_2 en una dirección MAC. De manera similar, los hosts A y B se conectan a la misma red física. Si una aplicación en el host A envía datos a una aplicación en el host B, el software de protocolo en A debe resolver la dirección IP de B en la dirección Mac de B; luego debe usar la dirección MAC para enviar directamente la trama.

Pero si una aplicación en el host A envía un mensaje a una aplicación en el host F, que está conectado a una red remota, el software de protocolo en A no intentará resolver la dirección de F. En su defecto, el software de IP que está en A determina que el paquete debe viajar a través del enrutador R_1 y entonces resuelve la dirección de R_1 . Una vez que calcula R_2 como el siguiente salto, el software de IP en R_1 resuelve la dirección de R_2 . De manera similar, R_2 resuelve la dirección de F.

Para resumir:

A la asociación entre una dirección de protocolo y una dirección de hardware se le conoce como resolución de direcciones. Cuando un host o un enrutador necesitan enviar un paquete a otra computadora de la misma red física, usan la resolución de direcciones. Una computadora nunca intenta resolver la dirección de una computadora que esté conectada a una red remota.

23.3 Un ejemplo de direcciones IPv4

Vamos a ver un ejemplo para recordar cómo se usan las direcciones IP y MAC en Internet. El ejemplo además nos ayudará a entender la resolución de direcciones. La figura 23.2 muestra una interred con tres redes y dos hosts. Se muestran las direcciones MAC e IP asignadas a cada sistema. Para evitar que la figura se haga muy grande, abreviamos las direcciones MAC a seis dígitos hexadecimales y usamos direcciones IPv4; sin embargo, IPv6 sigue la misma metodología.

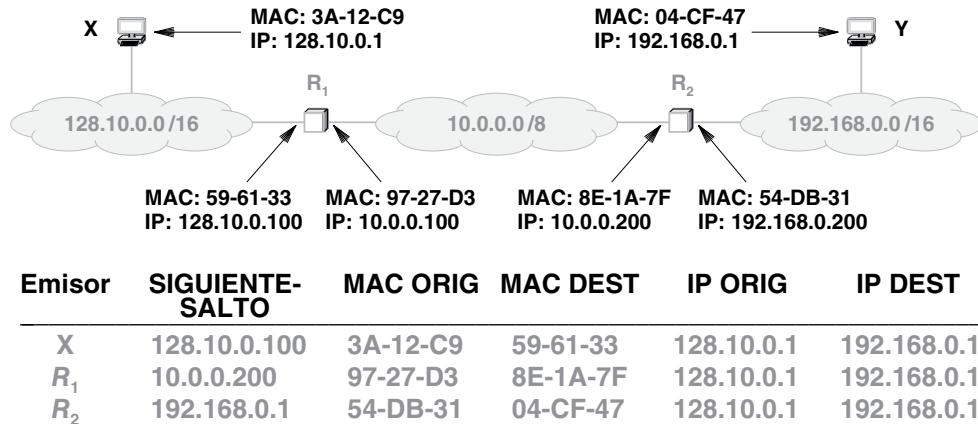


Figura 23.2 Se muestra una interred con direcciones MAC e IP, así como las direcciones de las tramas a medida que un datagrama viaja del host X al host Y .

En la figura, la tabla enumera tres tramas: una que viaja de X a R_1 , una segunda de R_1 a R_2 y una tercera que viaja de R_2 a Y . Las direcciones del *SIGUIENTE-SALTO* enlistadas en la tabla no se transportan en un paquete, sino que el emisor las usa durante el reenvío.

23.4 El protocolo de resolución de direcciones (ARP) IPv4

¿Cómo traduce el software una dirección de protocolo de alto nivel en una dirección MAC? IPv4 e IPv6 siguen diferentes metodologías para la resolución de direcciones. Consideraremos ambas, comenzando con IPv4, que usa el *protocolo de resolución de direcciones (ARP)*.

La idea del ARP es simple. Suponga que las computadoras X y Y están conectadas a la misma Ethernet y que X debe resolver la dirección IPv4 de Y . La computadora X difunde una solicitud que dice: “Busco la dirección MAC de una computadora que tenga la dirección Y de IPv4”. La difusión sólo viaja a través de una red. Todas las computadoras reciben la solicitud, pero únicamente la computadora Y responde. Es decir, cuando recibe una copia de la solicitud, la computadora Y envía una respuesta dirigida a X que dice: “Soy la computadora con la dirección IP Y , y mi dirección MAC es M ”. La figura 23.3 ilustra el intercambio de mensajes.

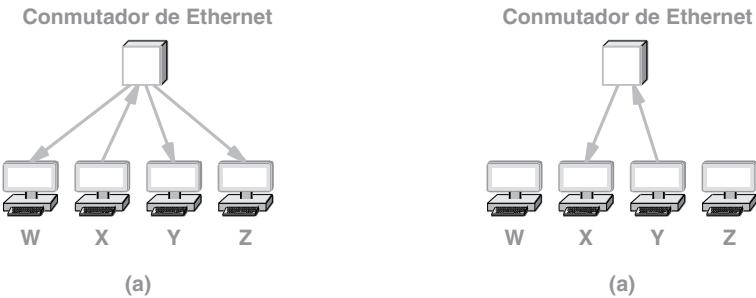


Figura 23.3 Ilustración de un intercambio de mensajes del ARP cuando (a) la computadora *X* difunde una solicitud y (b) la computadora *Y* envía una respuesta.

23.5 Formato de mensajes del ARP

Cuando se inventó el ARP, se estaban creando muchas tecnologías de LAN. Por lo tanto, en vez de limitarlo a las direcciones IPv4 y a Ethernet, los diseñadores crearon un protocolo generalizado. Un mensaje ARP tiene al principio campos de tamaño fijo para especificar el tamaño de las direcciones de hardware y de protocolo que se van a usar. Cuando se usa el ARP con IPv4 y Ethernet, la longitud de las direcciones de hardware se establece en 6 octetos (debido a que una dirección de Ethernet tiene 48 bits de longitud), mientras que la longitud de la dirección del protocolo se establece en 4 (ya que una dirección IPv4 tiene 32 bits de longitud). Lo irónico es que:

Aunque el formato de mensaje del ARP es bastante general como para permitir direcciones de protocolo y de hardware arbitrarias, el ARP casi siempre se usa para vincular una dirección IP a una dirección Ethernet de 48 bits.

La figura 23.4 ilustra el formato de un mensaje del ARP cuando se usa con una dirección IP versión 4 (de 4 octetos) y una dirección de hardware Ethernet (de 6 octetos). Aun cuando algunos campos abarcan dos líneas, cada línea de la figura corresponde a 32 bits de un mensaje ARP (el estándar para los protocolos de Internet). El mensaje contiene los siguientes campos:

TIPO DE DIRECCIÓN DE HARDWARE. Un campo de 16 bits que especifica el tipo de dirección de hardware a utilizar; el valor es 1 para Ethernet.

TIPO DE DIRECCIÓN DE PROTOCOLO. Un campo de 16 bits que especifica el tipo de dirección de protocolo que se va a usar; el valor es 0x0800 para IPv4.

LONG DIRECH. Un entero de 8 bits que especifica el tamaño de las direcciones de hardware en octetos.

LONG DIRECP. Un entero de 8 bits que especifica el tamaño de las direcciones de protocolo en octetos.

0	8	16	24	31			
TIPO DE DIRECCIÓN DE HARDWARE		TIPO DE DIRECCIÓN DE PROTOCOLO					
LONG DIRECH	LONG DIRECP	OPERACIÓN					
DIRECH EMISOR (primeros 4 octetos)							
DIRECH EMISOR (últimos 2 octetos)		DIRECP EMISOR (primeros 2 octetos)					
DIRECP EMISOR (últimos 2 octetos)		DIRECH DESTINO (primeros 2 octetos)					
DIRECH DESTINO (últimos 4 octetos)							
DIRECP DESTINO (los 4 octetos)							

Figura 23.4 El formato para un mensaje del ARP cuando se vincula una dirección IPv4 con una dirección Ethernet.

OPERACIÓN. Un campo de 16 bits que especifica si el mensaje es una solicitud (el campo contiene 1) o una respuesta (el campo contiene 2).

DIRECH EMISOR. Un campo que se extiende por un número de octetos determinado por *LONG DIRECH* y que contiene la dirección de hardware del emisor.

DIRECP EMISOR. Un campo que se extiende por un número de octetos determinado por *LONG DIRECP* y que contiene la dirección de protocolo del emisor.

DIRECH DESTINO. Un campo que se extiende por un número de octetos determinado por *LONG DIRECH* y que contiene la dirección de hardware del destino.

DIRECP DESTINO. Un campo que se extiende por un número de octetos determinado por *LONG DIRECP* y que contiene la dirección de protocolo del destino.

Como se muestra en la figura, un mensaje del ARP contiene campos para dos vinculaciones de direcciones. Una vinculación corresponde al emisor y la otra al receptor destinado, que ARP denomina como el *destino*. Cuando se envía una solicitud, el emisor no conoce la dirección del hardware de destino (la información que se está solicitando). Por lo tanto, el campo *DIRECH DESTINO* en una solicitud ARP puede llenarse con ceros debido a que el contenido no se usa. En una respuesta, la vinculación del destino hace referencia a la computadora inicial que envió la solicitud. Por consiguiente, el par de direcciones de destino de una respuesta no sirve para nada. Sin embargo, la inclusión de los campos de destino sobrevivió de una de las primeras versiones del protocolo.

23.6 Encapsulamiento del ARP

Cuando viaja a través de una red física, un mensaje ARP se encapsula en una trama de hardware. Al igual que con el IP, el mensaje ARP se coloca en el área de carga, por lo que la red subyacente no analiza el mensaje ARP ni interpreta los campos. La figura 23.5 ilustra el encapsulamiento del ARP en una trama de Ethernet.



Figura 23.5 Ilustración del encapsulamiento del ARP en una trama de Ethernet.

El *campo de tipo* en el encabezado de la trama especifica que la trama contiene un mensaje ARP. Ethernet usa el campo de tipo *0x806* para denotar un mensaje ARP. Antes de transmitir la trama, un emisor debe asignar el valor al campo de tipo y un receptor debe examinar este campo en cada trama entrante. Se usa el mismo valor de tipo para solicitudes y respuestas del ARP. Por lo tanto, el tipo de trama no distingue entre los tipos de mensajes ARP. Para determinar si un mensaje entrante es una solicitud o una respuesta, un receptor debe examinar el campo *OPERACIÓN* del mensaje.

23.7 Uso de caché y procesamiento de mensajes del ARP

Aunque el ARP se usa para vincular direcciones, no es eficiente enviar una solicitud ARP para cada datagrama; ya que en cada caso son tres las tramas que deben recorrer la red (una solicitud ARP, una respuesta ARP y el datagrama en sí). Lo que es más importante, como la mayor parte de la comunicación entre computadoras involucra una secuencia de paquetes, es probable que un emisor repita el intercambio muchas veces.

Para reducir el tráfico de red, el software ARP extrae y guarda la información de una respuesta para poder usarla en los próximos paquetes. El software no conserva la información indefinidamente. En su lugar, el ARP mantiene en memoria una pequeña tabla de vinculaciones. El ARP administra la tabla como una *caché*; es decir, cuando llega una respuesta, ésta reemplaza a una entrada, y cuando la tabla se queda sin espacio o cuando no se actualiza una entrada durante un periodo largo (por ejemplo, 20 minutos), la entrada más antigua se elimina. Cuando necesita vincular una dirección, el ARP comienza buscando en la caché. Si la vinculación está presente, el ARP usa la vinculación sin transmitir una solicitud. Si la vinculación no está presente en la caché, el ARP difunde una solicitud, espera una respuesta, actualiza la caché y luego procede a usar la vinculación.

Cabe mencionar que a diferencia de la mayoría de los esquemas de uso de caché, la caché del ARP no se actualiza cuando ocurre una búsqueda (es decir, cuando se hace referencia a una entrada). En vez de ello la caché se actualiza sólo cuando llega un mensaje del ARP a través de la red (ya sea una solicitud o una respuesta). El algoritmo 23.1 expone el procedimiento para manejar un mensaje ARP entrante.

Algoritmo 23.1

Dado:

 Un mensaje ARP entrante (ya sea solicitud o respuesta)

Se realiza:

 Procesar el mensaje y actualizar la caché del ARP

Método:

 Extraer la dirección IP del emisor I y la dirección MAC M

 If (la dirección ya está en la caché del ARP) {

 Reemplazar la dirección MAC en la caché con M

 }

 if (el mensaje es una solicitud y el destino es “yo”) {

 Agregar una entrada en la caché del ARP para el emisor

 siempre y cuando no exista ya;

 Generar y enviar una respuesta;

 }

Algoritmo 23.1 Los pasos que realiza el ARP al procesar un mensaje entrante.

Como especifica el algoritmo, el ARP realiza dos pasos básicos para procesar un mensaje. Primero, el receptor extrae la vinculación de direcciones del emisor y actualiza la caché si ésta ya contiene una entrada para el emisor. La actualización de la caché se hace cargo del caso en el que la dirección del hardware del emisor cambia. En el segundo paso, el receptor examina el campo *OPERACIÓN* del mensaje para determinar si es una solicitud o una respuesta. Si el mensaje es una respuesta, el receptor debe haber emitido previamente una solicitud y espera la vinculación (es decir, la caché contiene una entrada para el emisor, la cual se llenó durante el primer paso). Si el mensaje es una solicitud, el receptor compara el campo *DIRECP DESTINO* con la dirección de protocolo local. Si las dos son idénticas, la computadora es el destino de la solicitud y debe enviar una respuesta ARP. Para formar la respuesta, el ARP comienza con el mensaje entrante, invierte las vinculaciones del emisor y el destino, inserta su dirección de hardware en el campo *DIRECH EMISOR* y cambia el campo *OPERACIÓN* a 2 para indicar una respuesta.

ARP contiene una optimización adicional: cuando encuentra una solicitud a la que debe responder, una computadora extrae la vinculación de la dirección del emisor de la solicitud y la agrega a su caché para usarla después. Para entender la optimización, es necesario conocer dos hechos:

- La mayoría de la comunicación entre computadoras implica tráfico de dos vías; si un mensaje viaja de A a B , hay muchas probabilidades de que una respuesta viaje de B a A .
- Puesto que cada vinculación de direcciones requiere memoria, una computadora no puede almacenar un número arbitrario de vinculaciones de direcciones.

El primer hecho explica por qué al extraer la vinculación de direcciones del emisor, el desempeño del ARP se optimiza. La computadora A sólo envía una solicitud ARP para el destino B cuando A tiene que entregar un paquete a B . Así, cuando B encuentra el destino de una solicitud de A , es probable que cuando llegue el paquete se envíe otro paquete de B a A . Al hacer que B extraiga la vinculación de A de la solicitud ARP entrante, se elimina la necesidad de una solicitud ARP posterior de B a A .

El segundo hecho explica por qué sólo una de las computadoras destino de una solicitud ARP agrega una entrada nueva a la caché del ARP en vez de que lo hagan todas las que reciben una solicitud. Si todas las computadoras insertaran la información, sus cachés se llenarían rápidamente aun cuando la mayoría de ellas nunca se comuniquen con muchas de las computadoras en la red. Por consiguiente, ARP registra únicamente las vinculaciones de direcciones que son factibles de ser requeridas.

23.8 El límite conceptual de direcciones

En el capítulo 1 vimos que TCP/IP usa un modelo de referencia de cinco capas. El protocolo de resolución de direcciones presenta un interesante problema: ¿debe el ARP clasificarse como de capa 2 o de capa 3? Por una parte, el ARP lidia con direcciones MAC; que son parte de la capa 2. Por otra parte, el ARP lidia con direcciones IP, que son parte de la capa 3. Por fortuna, el modelo de capas de TCP/IP ofrece una respuesta: la capa 2 es una capa de interfaz de red entre el protocolo IP y el hardware subyacente. El ARP encaja muy bien en la definición, ya que proporciona la función de la interfaz de direcciones.

El ARP crea un límite conceptual importante entre las direcciones MAC y las direcciones IP. El protocolo ARP oculta los detalles del direccionamiento de hardware y permite que las capas más altas de software usen direcciones IP. Por lo tanto, hay un límite conceptual importante impuesto entre la capa de la interfaz de red y todas las capas superiores: las aplicaciones y las capas más altas del software de protocolo están creadas para usar direcciones de protocolo. La figura 23.6 ilustra el límite de direccionamiento.

La idea importante es:

ARP forma un límite conceptual: los protocolos por encima del ARP usan direcciones IP y los protocolos por debajo del ARP usan direcciones MAC.

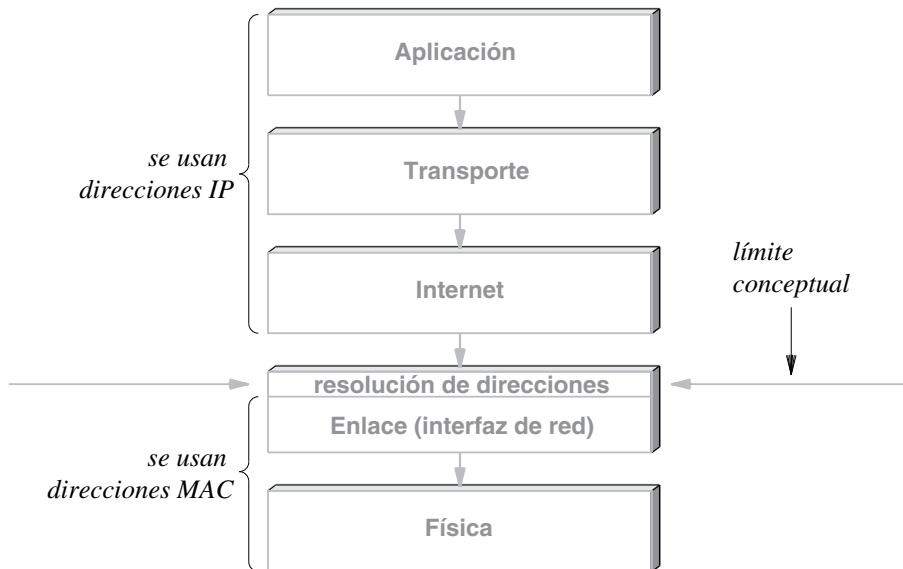


Figura 23.6 Ilustración del límite entre el uso de direcciones IP y de direcciones MAC.

23.9 Protocolo de mensajes de control de Internet (ICMP)

Dijimos que el protocolo IP define un servicio de comunicaciones del mejor esfuerzo en el que los datagramas pueden perderse, duplicarse, retrasarse o entregarse desordenados. Tal vez parezca que un servicio del mejor esfuerzo no necesita detección de errores. Sin embargo, en el caso del protocolo IP este servicio no es descuidado, ya que el IP intenta evitar los errores y reportar los problemas cuando ocurren. De hecho, ya hemos visto un ejemplo de detección de errores en el IP: el campo *TIME TO LIVE (TTL)* o *TIEMPO DE VIDA* de IPv4, conocido como campo *HOP LIMIT* o *LÍMITE DE SALTO* en IPv6, que se usa para evitar que un datagrama viaje indefinidamente por una ruta circular.[†]

El IP incluye un protocolo complementario que se usa para reportar los errores de regreso a la fuente original (es decir, a la computadora que envió el datagrama). Para IPv4, es el *protocolo de mensajes de control de Internet (ICMP o ICMPv4)*. Para IPv6 se creó una versión modificada, conocida como *ICMPv6*. Lo interesante es que cada versión del IP y el protocolo ICMP asociado son codependientes: el IP depende del ICMP para reportar errores y el ICMP usa el IP para transportar los mensajes de error.

Como muchos mensajes son iguales, usaremos el ICMPv4 como un ejemplo. Aunque se han definido más de veinte mensajes de ICMPv4, sólo se usan unos cuantos. La figura 23.7 enumera los mensajes ICMPv4 clave y el propósito de cada uno. Como se indica en la figura, ICMP contiene dos tipos de mensajes: los que se usan para reportar errores y los que se usan para obtener información. Por ejemplo, cada uno de los mensajes *Tiempo excedido* y *Destino inalcanzable* reporta un error cuando un datagrama no puede entregarse con éxito.

[†] Seguiremos la convención de los profesionales de redes y usaremos la abreviación *TTL* para IPv4 o IPv6.

Núm.	Tipo	Propósito
0	Eco de respuesta	Lo usan ping y traceroute
3	Destino inalcanzable	No se pudo entregar el datagrama
5	Redirección	El host debe cambiar una ruta
8	Eco de solicitud	Lo usan ping y traceroute
11	Tiempo excedido	El TTL expiró o se agotó el tiempo de los fragmentos
12	Problema de parámetro	El encabezado IP es incorrecto
30	Traceroute	Lo usa el programa traceroute

Figura 23.7 Ejemplos de mensajes de ICMPv4 con el número de mensaje y el propósito.

Las aplicaciones *ping* y *traceroute* usan los mensajes *Eco de solicitud* y *Eco de respuesta* para probar la conectividad y asignar una ruta a través de Internet. Cuando un host o un enrutador recibe un *eco de solicitud*, el software ICMP envía un *eco de respuesta* que transporta los mismos datos que la solicitud. De esta forma, una aplicación *ping* envía una solicitud a un host remoto, espera una respuesta y declara ya sea que el host es alcanzable o, después de un tiempo de expiración apropiado, declara que el host es inalcanzable. Una aplicación *traceroute* envía una serie de mensajes *eco de solicitud* con el TTL establecido en 1, 2, 3 y así en lo sucesivo. El TTL en el primer mensaje expira después de un salto, el TTL en el segundo mensaje expira después de dos saltos, el TTL en el tercer mensaje expira después de tres saltos y así sucesivamente. Por lo tanto, la aplicación *traceroute* recibe un mensaje de error ICMP de *Tiempo excedido* por parte de cada enrutador que se encuentra a lo largo de la ruta, así como un *eco de respuesta* por parte del destino final.

23.10 Formato de mensajes y encapsulamiento del ICMP

El ICMP usa el IP para transportar cada mensaje de error: cuando un enrutador tiene un mensaje ICMP para enviar, crea un datagrama IP y encapsula el mensaje ICMP en el datagrama. Es decir, el mensaje ICMP se coloca en el área de carga útil del datagrama IP. Luego, el datagrama se reenvía como siempre: encapsulado completamente en una trama para ser transmitido. Los mensajes ICMPv4 siempre se encapsulan en IPv4 y los mensajes ICMPv6 siempre se encapsulan en IPv6. La figura 23.8 ilustra los dos niveles de encapsulamiento.

Los datagramas que transportan los mensajes ICMP no tienen una prioridad especial. Se reenvían como cualquier otro datagrama, con una pequeña excepción: si un datagrama que transporta un mensaje de error ICMP provoca un error, no se envía un mensaje de error. La razón debería ser clara ya que los diseñadores querían evitar que Internet se congestionara al transportar mensajes sobre mensajes de error.

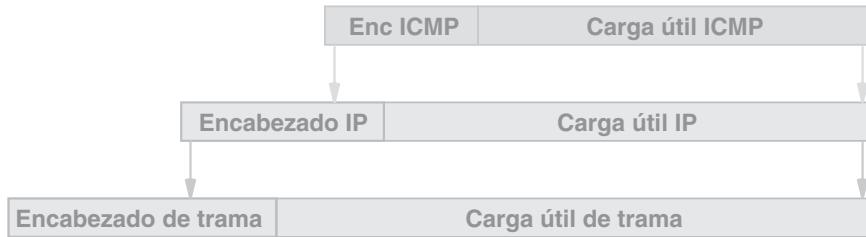


Figura 23.8 Se usan dos niveles de encapsulamiento con un mensaje ICMP.

Podemos resumir:

El protocolo de mensajes de control de Internet incluye mensajes sobre errores y mensajes informativos. El ICMP encapsula los mensajes en IP para la transmisión y el IP usa el ICMP para reportar problemas.

23.11 Vinculación de direcciones IPv6 con descubrimiento del vecindario

Dijimos que el IPv6 no usa el ARP para vinculación de direcciones, sino que usa un protocolo conocido como *descubrimiento del vecindario de ICMPv6 (IPv6-ND)*, el cual se basa en los mensajes ICMPv6. El descubrimiento del vecindario es una parte esencial de IPv6 que proporciona varias funciones además de la vinculación de direcciones. Sin embargo, la función de vinculación de direcciones es imprescindible para la transmisión de datagramas y significa que ND tiene una estrecha integración con el IPv6.

Recuerde que el IPv6 no tiene difusión, así que ¿cómo puede un nodo IPv6 usar ICMPv6 para buscar vecinos? IPv6 define una dirección de multidifusión en la que todos los nodos de una red deben escuchar. Por lo tanto, IPv6-ND puede transmitir por multidifusión un mensaje que pida a los vecinos que respondan. Las respuestas contienen direcciones MAC, que IPv6-ND registra en una tabla similar a una caché del ARP. IPv6-ND contacta periódicamente a cada vecino para verificar que siga activo. De esta forma, la lista permanece actualizada y se usa al enviar un datagrama.

23.12 Software de protocolo, parámetros y configuración

Nuestra explicación de los protocolos de Internet describe su operación una vez que se enciende un host o un enrutador, y que se inician tanto el sistema operativo como el software de protocolo. Aquí surge la pregunta: ¿cómo comienza a funcionar el software de protocolo en un host o en un enrutador? Para un enrutador, la respuesta es simple: un administrador debe especificar los valores iniciales para elementos como la dirección IP de cada conexión de red, el software de protocolo a ejecutar y los valores iniciales para una tabla de reenvío. La configuración se guarda en el disco y un enrutador carga los valores durante el arranque.

La configuración del host es más compleja; por lo general usa un proceso de dos pasos conocido como arranque o *bootstrapping*. El primer paso ocurre cuando la computadora inicia. El sistema operativo llena un conjunto básico de parámetros de operación que permiten al software de protocolo comunicarse a través de una red local. En el segundo paso, el software de protocolo llena la información adicional tal como la dirección IP de la computadora, la máscara de dirección y la dirección de un servidor DNS local. En esencia, el software de protocolo se crea con un conjunto de *parámetros* que controlan la operación y la inicialización llena los valores para esos parámetros. La ventaja de la parametrización surge cuando consideramos un dispositivo móvil, como una tableta o una computadora tipo laptop. El dispositivo puede ejecutar el mismo software de protocolo en todo momento y no necesita cambiar en caso de que cambie la conexión de red de una computadora. Por ejemplo, cuando el dispositivo esté dentro del rango de una zona activa Wi-Fi, el hardware detecta la señal y el software de protocolo se configura para usar la red. Decimos que el software de protocolo puede *configurarse* para una situación específica. En resumen:

El software de protocolo está parametrizado para que pueda ejecutarse en una variedad de entornos de red. Para configurar el software se asignan los valores para un conjunto de parámetros que suministran información sobre la computadora y las redes a las que se conecta.

23.13 Protocolo de configuración dinámica de host (DHCP)

Se crearon varios mecanismos para que una computadora pueda obtener los parámetros de configuración de la red. Uno de los primeros mecanismos, conocido como el *protocolo de resolución inversa de direcciones (RARP)*, se utiliza ahora en los centros de datos en la nube. La versión original del ICMP incluía mensajes de *solicitud de máscara de dirección* y de *descubrimiento de enrutadores* para obtener la máscara de dirección que se usa en una red dada, además de la dirección de un enrutador. El *protocolo de arranque (BOOTP)* permitía a una computadora obtener una dirección IPv4, una máscara de dirección y la dirección de un enrutador predeterminado.

Al igual que la mayoría de los demás protocolos de configuración, BOOTP hacía que un host difundiera una solicitud. Pero a diferencia de otros protocolos de configuración, BOOTP usaba IPv4 para comunicarse con un servidor, enviando una solicitud a la *dirección de destino* con todos los bits en uno, mientras que usaba todos los bits en cero como *dirección de origen*. Un servidor BOOTP usaba la dirección MAC en una trama entrante para enviar una respuesta mediante unidifusión. De tal modo, un host que no conociera su dirección IPv4 podría usar IPv4 para comunicarse con un servidor BOOTP.

La versión inicial de BOOTP usaba una asignación de dirección fija en la que un servidor tenía una base de datos de la dirección IPv4 asignada a cada host de la red. Un servidor BOOTP requería de una administración manual, ya que antes de que una computadora pudiera usar BOOTP para obtener una dirección, el administrador de red tenía que configurar un servidor BOOTP para conocer la dirección IP de la computadora.

La configuración manual del servidor implica que BOOTP no puede manejar situaciones como un punto de acceso Wi-Fi en una cafetería que proporciona acceso a cualquier cliente. Por lo tanto, la IETF extendió BOOTP y le cambió el nombre a *protocolo de configuración dinámica de hosts (DHCP)*. El DHCP permite que una computadora cualquiera se una a una nueva red y obtenga automáticamente una

dirección IP sin tener que configurar un servidor. El concepto algunas veces se denomina *redes plug-and-play* o *redes de conectar y usar*. Podemos resumir este punto así:

DHCP permite que una computadora se mueva a una nueva red y obtenga información de configuración, sin que un administrador tenga que realizar cambios manuales en la base de datos del servidor.

Al igual que BOOTP, DHCP hace que una computadora difunda una solicitud a la que un servidor DHCP envía una respuesta.[†] Un administrador puede configurar un servidor DHCP para suministrar dos tipos de direcciones: direcciones asignadas en forma permanente (igual que BOOTP) o una reserva de direcciones dinámicas que se asignan según la demanda. La mayoría de los servidores están configurados para usar una reserva de direcciones dinámicas que se asignan a hosts arbitrarios. Para evitar que un host obtenga una dirección y la mantenga para siempre, cada asignación de dirección se limita a un tiempo fijo y la asignación se conoce como *arrendamiento*.[‡]

Cuando es necesario, el uso de arrendamientos permite a un servidor DHCP reclamar las direcciones. Cuando expira el arrendamiento ocurre una de dos acciones. El host que usaba la dirección puede optar por abandonar la dirección o puede renegociar con DHCP para extender el periodo de arrendamiento. La negociación ocurre en forma concurrente con otra actividad, de modo que los usuarios no están conscientes de que su computadora renovó un arrendamiento. De hecho, la mayoría de los servidores DHCP están configurados para aprobar cada extensión de arrendamiento, lo que significa que una computadora puede seguir operando durante un periodo extenso sin ninguna interrupción para ejecutar programas de aplicaciones o comunicaciones de red continuas. Sin embargo, un servidor puede configurarse para negar una extensión de arrendamiento por cuestiones administrativas o técnicas. Por ejemplo, considere una red en el salón de clases de una universidad. En tales casos es posible configurar un servidor de modo que todos los arrendamientos expiren al final de la clase (para permitir que el conjunto de direcciones se reasigne a la siguiente clase). DHCP otorga el control absoluto del arrendamiento a un servidor; si éste niega una solicitud de extensión, el host debe dejar de usar esa dirección.

23.14 Operación del protocolo DHCP y optimizaciones

Aunque el protocolo es simple, DHCP incluye varios detalles importantes que optimizan el rendimiento. Los tres más importantes son:

- Recuperación de pérdida o duplicación
- Caché de una dirección de servidor
- Evitar la inundación sincronizada

El primer punto significa que DHCP está diseñado para asegurar que los paquetes faltantes o duplicados no provoquen una configuración incorrecta. Si no se recibe una respuesta, un host vuelve a transmitir su solicitud, y si llega una respuesta duplicada, el host ignora la copia adicional. El segundo punto significa que una vez que un host usa un mensaje de *descubrimiento de DHCP* para encontrar un servidor DHCP, el host coloca en caché la dirección del servidor. Por lo tanto, la renovación del arrendamiento es eficiente.

[†] DHCP usa el término *oferta* para denotar el mensaje que envía un servidor.

[‡] Al momento de establecer una reserva de direcciones, un administrador especifica el tiempo de arrendamiento para cada dirección.

El tercer punto significa que DHCP toma las medidas para evitar solicitudes sincronizadas. Por ejemplo, podrían ocurrir solicitudes sincronizadas si todas las computadoras en una red se reinician al mismo tiempo después de una falla de energía. Para evitar que todos los hosts en una red inunden el servidor DHCP con solicitudes al mismo tiempo, DHCP requiere que cada host se retrase por un tiempo aleatorio antes de transmitir (o retransmitir) una solicitud.

23.15 Formato de mensajes del DHCP

Aunque *DHCPv6* se creó para que las direcciones IPv6 pudieran administrarse en forma central, IPv6 se diseñó para usar la configuración automática[†] en vez de *DHCPv6*. Por lo tanto, nos enfocaremos en el DHCP para IPv4. Puesto que se diseñó como una extensión de *BOOTP*, la versión IPv4 de DHCP adoptó una versión ligeramente modificada del formato del mensaje *BOOTP*. La figura 23.9 ilustra el formato de los mensajes DHCP.



Figura 23.9 El formato del mensaje de DHCP (todas las direcciones son IPv4).

Con la excepción de *OPCIONES*, cada campo en un mensaje DHCP tiene un tamaño fijo. Los primeros siete campos contienen la información que se utiliza para procesar el mensaje. El campo *OP* especifica si el mensaje es una *solicitud* o una *respuesta*. Para distinguir entre varios mensajes que un cliente usa para descubrir servidores o solicitar una dirección, o bien entre los que un servidor usa para reconocer o denegar una solicitud, DHCP incluye un campo *OPCIONES* para un *tipo de mensaje* espe-

[†] En una sección posterior describiremos la configuración automática de IPv6.

cífico. Es decir, el campo *OP* indica si el mensaje viaja del cliente al servidor o viceversa, y uno de los valores del campo *OPCIONES* proporciona el tipo exacto del mensaje.

Los campos *TIPOH* y *LONH* especifican el tipo de hardware de red y la longitud de una dirección de hardware. Un cliente usa el campo *BANDERAS* para especificar si puede recibir respuestas difundidas o dirigidas. El campo *SALTOS* especifica cuántos servidores reenviaron la respuesta y el campo *IDENTIFICADOR TRANSACCIÓN* proporciona un valor que un cliente puede usar para determinar si una respuesta entrante coincide con su solicitud. El campo *SEGUNDOS TRANSCURRIDOS* especifica cuántos segundos transcurrieron desde que el host comenzó a iniciarse. Por último, si conoce su dirección IP (por ejemplo, si la dirección se obtuvo a través de otro mecanismo en vez de usar DHCP), un host llena el campo *DIRECCIÓN IP CLIENTE* en una solicitud.

Los campos posteriores del mensaje se usan en la respuesta para transmitir información de regreso al host que envió una solicitud. Al conceder un arrendamiento, el servidor DHCP devuelve una dirección IPv4 en el campo etiquetado *SU DIRECCIÓN IP*. Lo interesante es que DHCP usa el campo *OPCIONES* para devolver una máscara de dirección y una dirección de enrutador predeterminada.

23.16 Acceso indirecto a un servidor DHCP por medio de un retransmisor

Aunque el DHCP usa la difusión en la red local para encontrar un servidor, éste no requiere que cada red individual tenga un servidor. En su defecto, un *agente de retransmisión de DHCP* reenvía las solicitudes y respuestas entre un cliente y el servidor. Debe haber por lo menos un agente de retransmisión presente en cada red, y el agente de retransmisión debe configurarse con la dirección del servidor DHCP apropiado. Cuando el servidor responde, el agente de retransmisión reenvía la respuesta al cliente.

Tal vez parezca que usar varios agentes de retransmisión no es más efectivo que usar varios servidores DHCP. Sin embargo, los administradores de red prefieren manejar varios agentes de retransmisión por dos razones. Primero, en una red con un servidor DHCP y varios agentes de retransmisión, la administración de las direcciones está centralizada en un solo dispositivo. Por lo tanto, un administrador de red no necesita interactuar con varios dispositivos para cambiar las políticas de arrendamiento o determinar el estado actual. Segundo, muchos enrutadores comerciales contienen un mecanismo que brinda el servicio de retransmisión DHCP en todas las redes a las que se conecta el enrutador. Además, las herramientas del agente de retransmisión de un enrutador son por lo general fáciles de configurar (la configuración consiste en habilitar el reenvío y especificar la dirección de un servidor DHCP), y es poco probable que esta configuración cambie.

23.17 Configuración automática de IPv6

Cuando se creó IPv6, los diseñadores buscaron formas de automatizar la funcionalidad. La configuración de red fue uno de los principales objetivos. En especial, los diseñadores querían que dos nodos IPv6 aislados pudieran comunicarse a través de una red no administrada que no tuviera servidores. Por consiguiente, en vez de usar DHCP, los diseñadores contemplaron que un nodo IPv6 generara su propia dirección IP. Esta metodología se conoce como *configuración automática de IPv6*.

¿Cómo puede un nodo generar una dirección IP única? Hay dos pasos: generar un prefijo y generar un sufijo. La configuración automática especifica que si la red ya tiene un prefijo global único, un nodo debe usar el prefijo. Entonces, el primer paso de la configuración automática consiste en transmitir por multidifusión[†] una solicitud a todos los nodos para descubrir el prefijo de red que se va a usar. Si no hay un prefijo disponible, el nodo usa un valor que se reserva para la comunicación local.

El segundo paso de la configuración automática consiste en generar un sufijo único. En la mayoría de los casos, el tamaño grande de la dirección IPv6 facilita la generación de sufijos. IPv6 usa 64 bits de sufijo para identificar a un host. Una dirección MAC común consiste en 48 bits y se garantiza que es única. Así, un nodo usa su dirección MAC como un sufijo único para IPv6. El estándar *EUI-64* de IEEE define con exactitud cómo es que los 48 bits de una dirección MAC se colocan en un campo de 64 bits (lo sorprendente es que la dirección MAC se divide en dos partes y se insertan 16 bits entre las partes).

23.18 Traducción de direcciones de red (NAT)

A medida que Internet creció y las direcciones comenzaron a escasear, se introdujeron mecanismos de subredes de IP y de direccionamiento sin clases para ayudar a conservar las direcciones.[‡] Se inventó un tercer mecanismo que permite que varias computadoras en un sitio compartan una sola dirección IP válida a nivel global. Conocida como *traducción de direcciones de red (NAT)*, la tecnología proporciona una comunicación *transparente* en el sentido en que un host en el sitio parece tener una conexión normal a Internet, mientras que un host en Internet parece recibir la comunicación de una sola computadora en el sitio, en vez de recibirla de una de varias computadoras. Es decir, los hosts en el sitio y los hosts en Internet ejecutan software TCP/IP y aplicaciones convencionales, y se comunican como siempre a través de Internet.

NAT se ejecuta como un servicio en línea, lo que significa que debe colocarse en la conexión entre Internet y un sitio. Aunque NAT sea conceptualmente independiente de otras herramientas y servicios, la mayoría de las implementaciones integran a NAT en otro dispositivo, tal como un enrutador inalámbrico Wi-Fi. La figura 23.10 ilustra una disposición común.

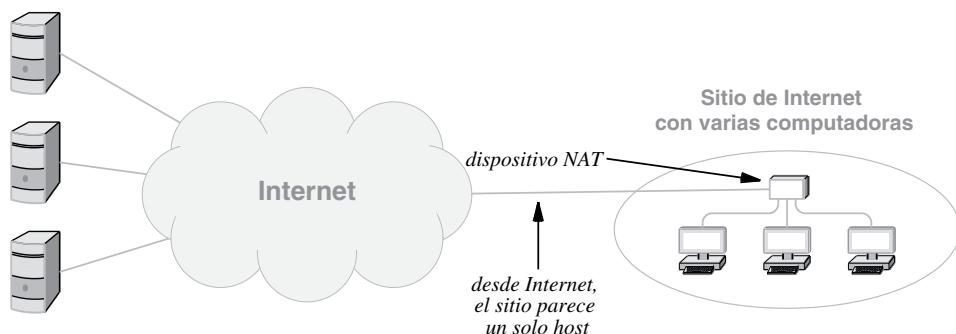


Figura 23.10 La arquitectura conceptual que se utiliza con NAT.

[†] Recuerde que IPv6 usa multidifusión en vez de difusión.

[‡] En el capítulo 21 encontrará una descripción del direccionamiento de subredes y del direccionamiento sin clases.

23.19 Operación de NAT y direcciones IPv4 privadas

El objetivo de NAT es crear una ilusión. Cuando se ve desde Internet, el sitio parece consistir en una sola computadora host a la que se asignó una dirección IP válida; todos los datagramas que se envían desde el sitio parecen originarse de un host, y todos los datagramas que se envían al sitio parecen enviarse a un host. Desde el punto de vista de un host en el sitio, el host puede usar DHCP para adquirir una dirección IPv4 y usar la dirección para comunicarse a través de Internet.

Desde luego que un dispositivo NAT no puede asignar una dirección IPv4 específica a varias computadoras. Si dos o más computadoras usan la misma dirección, surgen conflictos y la comunicación fracaña. Por lo tanto, para asegurar la exactitud, a cada computadora en una red dada se le debe asignar una dirección IP única. NAT resuelve el problema mediante el uso de dos tipos de direcciones. El dispositivo NAT en sí usa DHCP para obtener una dirección IPv4 válida a nivel global del ISP del usuario. Cuando un host en el sitio se conecta y ejecuta DHCP, el dispositivo NAT responde la solicitud y asigna una *dirección privada* única, también conocida como una *dirección no enrutable*. La figura 23.11 enumera los bloques de direcciones que el IETF designó como privados. Un bloque adicional en 169.254.0.0/16 también es privado, pero está destinado para usarse en asignaciones locales de enlace en vez de NAT.

Bloque	Descripción
10.0.0.0/8	Bloque de direcciones privadas de la clase A
172.16.0.0/12	16 bloques contiguos de la clase B
192.168.0.0/16	256 bloques contiguos de la clase C

Figura 23.11 Bloques de direcciones IPv4 privadas (no enrutables) que utiliza NAT.

Como un ejemplo, suponga que un dispositivo NAT específico está usando el bloque de direcciones 192.168.0.0/16 para asignar direcciones privadas a los hosts dentro del sitio. Para asegurar que cada dirección dentro del sitio sea única (es decir, para evitar conflictos), a los hosts se les podrían asignar las direcciones 192.168.0.1, 192.168.0.2 y así sucesivamente.

Por desgracia, las direcciones privadas no son válidas en la red Internet global y los enruteadores en Internet están configurados para rechazar datagramas que especifiquen direcciones no enrutables. Por consiguiente, el direccionamiento privado sólo se usa dentro de un sitio, y antes de admitir en Internet un datagrama proveniente del sitio, NAT debe traducir la dirección IP privada en una dirección IP válida a nivel global. De manera similar, antes de transferir un datagrama a un host en el sitio, NAT debe traducir la dirección IP válida a nivel global de un paquete entrante a una dirección privada.

Para entender la traducción de NAT, hay que pensar en las direcciones IP. Cuando un datagrama sale del sitio, NAT debe reemplazar la dirección de origen IP privada con una dirección válida a nivel global. Cuando llega un datagrama de Internet, NAT debe reemplazar la dirección de destino con la dirección privada correcta de uso interno. Por ejemplo, suponga que a un dispositivo NAT se le

asigna una dirección IP válida a nivel global de 128.210.24.6 y considere las traducciones que ocurren si un host con la dirección privada 192.168.0.1 envía un datagrama a un host en Internet con la dirección 198.133.219.25 y recibe una respuesta. La figura 23.12 ilustra las traducciones NAT que ocurren en cada dirección.

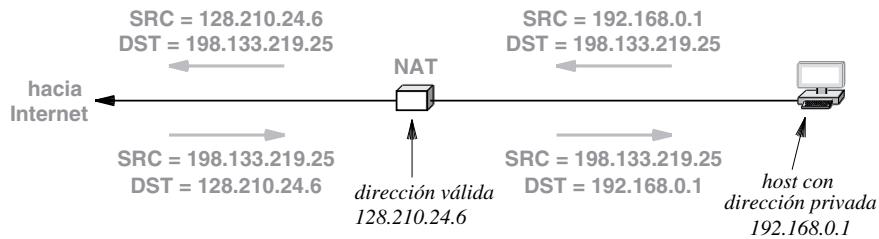


Figura 23.12 Ilustración de la traducción NAT básica que cambia la dirección de origen de un datagrama saliente y la dirección de destino de un datagrama entrante.

Para resumir:

La forma más básica de NAT reemplaza la dirección IP de origen en los datagramas que pasan del sitio hacia Internet, y reemplaza la dirección IP de destino en los datagramas que pasan de Internet hacia el sitio.

La mayoría de las implementaciones de NAT usan una *tabla de traducciones* para almacenar la información necesaria para reescribir las direcciones. Cuando un datagrama sale del sitio, NAT registra el destino junto con la dirección privada del emisor. Cuando llega un datagrama, NAT consulta la tabla para determinar qué computadora en el sitio debe recibir la respuesta. Por ejemplo, la figura 23.13 muestra una tabla de traducción que corresponde a la asignación de direcciones en la figura 23.12.

Dirección	Campo	Valor anterior	Valor nuevo
salida	IP origen	192.168.0.1	128.210.24.6
	IP destino	198.133.219.25	-- sin cambio --
entrada	IP origen	198.133.219.25	-- sin cambio --
	IP destino	128.210.24.6	192.168.0.1

Figura 23.13 Tabla de traducción NAT de ejemplo para la asignación en la figura 23.12.

23.20 NAT de la capa de transporte (NAPT)

La versión básica de NAT antes descrita sólo maneja situaciones en las que cada host en un sitio se comunica con un destino único en Internet. Si dos hosts en el sitio intentan comunicarse con el mismo destino X de Internet, la tabla de traducción contendrá varias entradas para X y NAT no podrá enrutar los datagramas entrantes. El servicio NAT básico falla en situaciones cuando dos o más aplicaciones que se ejecutan en un host determinado de un mismo sitio intentan la comunicación simultánea con diferentes destinos de Internet. Las formas más sofisticadas de NAT solucionan el problema.

La variación de NAT más popular permite a un sitio tener cualquier número de aplicaciones ejecutándose en hosts arbitrarios, todos comunicándose al mismo tiempo con cualquier destino a través de Internet. Por consiguiente, dos computadoras de un sitio pueden comunicarse con Google al mismo tiempo. Aunque se conoce técnicamente como *traducción de direcciones y puertos de red (NAPT)*, el mecanismo es tan popular que la mayoría de los profesionales de redes asumen que el término *NAT* significa también *NAPT*.

La clave para entender NAPT es saber que las aplicaciones usan *números de puerto de protocolos* para distinguir un servicio de otro. Los capítulos 24 y 25 explican los detalles y muestran cómo los protocolos de transporte como UDP y TCP usan números de puerto. Por ahora basta con saber que NAPT puede usar un número de puerto para asociar cada datagrama con una conversación TCP o UDP específica. Es decir, en vez de detenerse en la capa IP, NAPT opera en los encabezados de la capa de transporte. Como consecuencia, las entradas en la tabla de traducciones que usa NAPT contienen una 4-tupla de números de puertos de protocolos de origen y destino, así como de direcciones IP de origen y destino.

Por ejemplo, considere la tabla de traducción que podría resultar si un navegador en la computadora 192.168.0.1 y un navegador en la computadora 192.168.0.2 usan cada uno el puerto local 30000, y cada uno forma una conexión TCP a un servidor Web en el puerto 80, a través de un dispositivo NAPT que usa la dirección 128.10.24.6. Para evitar un conflicto, NAPT debe elegir un puerto de origen TCP alternativo para las conexiones. La figura 23.14 muestra una posibilidad.

Dir.	Campos	Valor anterior	Valor nuevo
salida	IP ORIG:TCP ORIG	192.168.0.1:30000	128.10.24.6:40001
salida	IP ORIG:TCP ORIG	192.168.0.2:30000	128.10.24.6:40002
entrada	IP DEST:TCP DEST	128.10.24.6:40001	192.168.0.1:30000
entrada	IP DEST:TCP DEST	128.10.24.6:40002	192.168.0.2:30000

Figura 23.14 Ejemplo de una tabla de traducción de NAPT para dos conexiones TCP al mismo servidor Web.

En la figura, las aplicaciones de dos computadoras locales usan el puerto local 3000. NAPT se encarga de la situación cambiando los números de puerto sin que haya confusión. En el ejemplo, NAPT selecciona el puerto 40001 para una conexión y 40002 para la otra.

23.21 NAT y los servidores

Dijimos que un sistema NAT crea automáticamente una tabla de traducción observando el tráfico saliente e insertando una nueva asignación en la tabla cada vez que una aplicación en el sitio inicia la comunicación. Por desgracia, la construcción automática de la tabla no funciona bien para la comunicación que se genera de Internet hacia el sitio. Por ejemplo, si varias computadoras en un sitio ejecutan un servidor de base de datos, el dispositivo NAT no puede saber qué computadora debe recibir una conexión Web entrante. Se creó una variante de NAT conocida como *dos veces NAT* (Twice NAT) para que un sitio pueda ejecutar servidores. Dos veces NAT hace que el sistema NAT interactúe con el servidor del sistema de nombres de dominio del sitio. Cuando una aplicación en Internet busca el nombre de dominio de una computadora en el sitio, el servidor DNS en el sitio devuelve la dirección IP válida que se asignó al dispositivo NAT, y también crea una nueva entrada en la tabla de traducción NAT. Por lo tanto, la tabla de traducción se inicializa antes de que llegue el primer paquete. Aunque no es elegante, dos veces NAT funciona para la mayoría de los casos. Sin embargo, es susceptible a falla si una aplicación cliente usa la dirección IP directamente sin realizar una búsqueda de nombre de dominio, o si el cliente usa un proxy DNS para resolver nombres de dominio.

23.22 Software y sistemas NAT para usar en casa

NAT es especialmente útil en una residencia o pequeño negocio que tiene una conexión de banda ancha, ya que permite que un conjunto de computadoras compartan la conexión sin requerir que el cliente compre direcciones IP adicionales del ISP. Además del software que permite a una PC actuar como dispositivo NAT para equipos PC adicionales, hay sistemas de hardware NAT disponibles a bajo costo. Dichos sistemas se llaman por lo general *enrutadores inalámbricos*, ya que permiten a las computadoras conectarse a través de Wi-Fi.[†] La figura 23.15 ilustra cómo se conecta un enrutador inalámbrico.

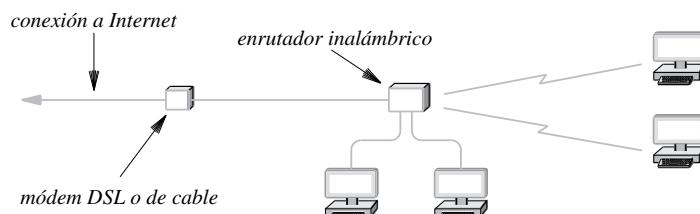


Figura 23.15 Ilustración de las conexiones para un enrutador inalámbrico.

[†] La terminología es ligeramente engañosa, ya que dichos enrutadores también ofrecen por lo general conexiones alámbricas.

23.23 Resumen

IPv4 usa el protocolo de resolución de direcciones ARP para vincular una dirección IPv4 del siguiente salto con una dirección MAC equivalente. ARP define tanto el formato de los mensajes que las computadoras intercambian para resolver una dirección, como el encapsulamiento y las reglas para manejar los mensajes ARP. Como el direccionamiento de hardware difiere entre una red y otra, ARP sólo especifica un patrón general para el formato de los mensajes y permite determinar los detalles mediante el esquema de direccionamiento MAC. ARP especifica que una computadora debe difundir un mensaje requerido, pero que una respuesta debe ser dirigida. Además ARP usa una caché para evitar enviar una solicitud para cada paquete. IPv6 usa un mecanismo de resolución de direcciones alternativo conocido como *descubrimiento del vecindario de IPv6* o IPv6-ND.

Tanto IPv4 como IPv6 definen un mecanismo de reporte de errores complementario conocido como *protocolo de mensajes de control de Internet* (ICMP). Los enrutadores usan ICMP cuando llega un datagrama con valores incorrectos en los campos del encabezado o cuando no puede entregarse un datagrama. Los mensajes ICMP siempre se envían de vuelta a la fuente original de un datagrama, nunca a los enrutadores intermedios. Además de los mensajes que reportan errores, ICMP incluye mensajes informativos como los mensajes *eco de solicitud* y *eco de respuesta* que utilizan la aplicación *ping*. Cada tipo de mensaje ICMP tiene un formato único. Un campo de tipo en el encabezado permite a un receptor dividir un mensaje dado en campos adecuados. Un mensaje ICMP se encapsula en un datagrama IP para la transmisión.

En un principio se utilizaban protocolos independientes para obtener cada uno de los parámetros de configuración necesarios en el arranque. El *protocolo de configuración dinámica de host* (DHCP), que extiende el protocolo de arranque (BOOTP), permite a un host obtener toda la información necesaria con una sola solicitud. Una respuesta DHCP puede proporcionar una dirección IPv4, la dirección de un enrutador predeterminado y la dirección de un servidor de nombres. Cuando asigna una dirección IP de manera automática, DHCP ofrece al host un arrendamiento durante el cual puede usarse la dirección. Una vez que expira el arrendamiento, el host debe renovarlo o dejar de usar la dirección. Cuando se creó IPv6, los diseñadores eligieron un mecanismo de configuración automática que permite a un host generar una dirección IPv6 única; sin embargo, DHCPv6 se creó para permitir a los administradores usar DHCP para asignar direcciones IPv6.

Un mecanismo NAT permite que varias computadoras de un sitio usen Internet a través de una sola dirección IPv4. NAT reescribe los campos de encabezado en cada datagrama que pasa ya sea hacia Internet o hacia el sitio. Para las aplicaciones cliente, las traducciones NAT pueden establecerse de manera automática cuando el dispositivo NAT encuentra el primer paquete saliente de la comunicación. Existen diversas variaciones de NAT. La forma más popular (NAPT) opera sobre encabezados de la capa de transporte y traduce los números de puertos de protocolos así como las direcciones IPv4. NAPT permite que cualquier número de aplicaciones que se ejecutan en computadoras de un mismo sitio, se comuniquen al mismo tiempo con diversos destinos en Internet.

EJERCICIOS

- 23.1** Cuando un enrutador usa una tabla de reenvío para buscar una dirección del siguiente salto, el resultado es una dirección IP. ¿Qué debe ocurrir antes de poder enviar el datagrama?
- 23.2** ¿Qué término se usa para describir la asignación entre una dirección de protocolo y una dirección de hardware?
- 23.3** ¿Puede usarse ARP en una red que no proporciona difusión? ¿Por qué sí o por qué no?
- 23.4** ¿Cuántas respuestas espera recibir una computadora cuando difunde una solicitud ARP? Explique.
- 23.5** ¿Cuántos octetos ocupa un mensaje ARP cuando se usa con direcciones IP y Ethernet?
- 23.6** ¿Cómo sabe una computadora si una trama que llega contiene un datagrama IP o un mensaje ARP?
- 23.7** Suponga que una computadora recibe dos respuestas ARP para una sola solicitud. La primera respuesta afirma que la dirección MAC es M_1 y la segunda respuesta afirma que la dirección MAC es M_2 . ¿Cómo maneja ARP las respuestas?
- 23.8** ARP permite que la resolución de direcciones ocurra en una sola red. ¿Tiene sentido enviar a un servidor remoto una solicitud ARP en un datagrama IP? ¿Por qué sí o por qué no?
- 23.9** ¿Cuándo crea el algoritmo 23.1 una nueva entrada en una caché de ARP?
- 23.10** ¿Qué tipos de direcciones se usan en las capas que están debajo de ARP?
- 23.11** Si un datagrama tiene un valor incorrecto en uno de los campos del encabezado, ¿qué mensaje de error de ICMP se recibirá?
- 23.12** Si existe un ciclo de enrutamiento, ¿qué mensaje de error de ICMP se enviará? Explique el proceso.
- 23.13** Suponga que un usuario especificó una dirección de difusión dirigida como destino para *ping*. ¿Cuáles son los posibles resultados? Explique.
- 23.14** Algunas versiones del programa *traceroute* envían mensajes ICMP y otras envían mensajes UDP. Experimente con la versión de su propia computadora para determinar cuáles mensajes envía.
- 23.15** Dada una trama de Ethernet, ¿qué campos necesita examinar un host para determinar si la trama contiene un mensaje ICMP?
- 23.16** Haga una lista de la información de red clave que puede configurarse cuando una computadora se inicia.
- 23.17** ¿Cuál es la principal diferencia entre BOOTP y DHCP?
- 23.18** Algunas aplicaciones de red retrasan la configuración hasta que se necesita un servicio. Por ejemplo, una computadora puede esperar hasta que un usuario intente imprimir un documento antes de que el software busque las impresoras disponibles. ¿Cuál es la principal ventaja de la configuración diferida? ¿La principal desventaja?
- 23.19** DHCP permite localizar un servidor en una red remota. ¿Cómo puede la computadora enviar mensajes DHCP a un servidor en otra red?
- 23.20** Como alternativa a DHCP, idee un algoritmo distribuido que implemente un esquema de licitación. Suponga que una copia del algoritmo se ejecutará en cada computadora, y haga que éste asigne a cada computadora una dirección de host única.
- 23.21** ¿Cuál es el propósito principal de NAT?

- 23.22** Muchos dispositivos NAT seleccionan el bloque de direcciones 10.0.0.0/8 de la figura 23.11, ya que proporciona la mayor generalidad. Explique por qué.
- 23.23** En la figura 23.12, el ISP asignó una dirección IP al sitio. ¿Cuál es la dirección asignada?
- 23.24** Extienda la figura 23.14 para mostrar las asignaciones que se usarán si una tercera aplicación también intenta llegar al mismo servidor Web.
- 23.25** Cree una tabla de traducción NAPT para un caso en donde tres computadoras en un sitio tengan conexiones TCP hacia tres servidores Web independientes en Internet.
- 23.26** ¿Qué información crucial usa NAPT que no está disponible en la mayoría de los fragmentos IP?
- 23.27** Para optimizar el reensamblaje, algunas versiones del sistema operativo Linux envían primero el último fragmento de un datagrama IP y luego envían los fragmentos restantes en orden. Explique por qué enviar el último fragmento primero no funciona bien con NAPT.
- 23.28** Al usar un enrutador inalámbrico, ¿cuáles son las posibles direcciones IP que pueden asignarse a los hosts?
- 23.29** Modifique la figura 23.12 y la tabla en la figura 23.14 para usar un ejemplo de dirección IPv6.

Contenido del capítulo

- 24.1 Introducción, 415
- 24.2 Protocolos de transporte y comunicación de extremo a extremo, 415
- 24.3 El protocolo de datagramas de usuario, 416
- 24.4 El paradigma sin conexión, 417
- 24.5 Interfaz orientada a mensajes, 417
- 24.6 Semántica de la comunicación UDP, 418
- 24.7 Modos de interacción y entrega por multidifusión, 419
- 24.8 Identificación del punto final con números de puerto de protocolo, 419
- 24.9 Formato de los datagramas UDP, 420
- 24.10 La suma de verificación UDP y el seudoencabezado, 421
- 24.11 Encapsulamiento UDP, 421
- 24.12 Resumen, 422

24

UDP: servicio de transporte de datagramas

24.1 Introducción

Los capítulos anteriores describieron el servicio de entrega de paquetes sin conexión brindado por el protocolo IP y por el protocolo complementario que se usa para reportar errores. Este capítulo considera el UDP, uno de los dos principales protocolos de la capa de transporte que se utilizan en Internet, y el único servicio de transporte sin conexión. El capítulo habla sobre el formato de paquetes del UDP y las formas en que puede usarse este protocolo. Veremos que aunque el UDP es eficiente y flexible, tiene la sorprendente propiedad de usar la semántica de entrega del mejor esfuerzo. Además de hablar sobre UDP, el capítulo cubre el importante concepto de los números de puerto de protocolo.

El siguiente capítulo continúa la explicación mediante un enfoque en el otro protocolo importante de la capa de transporte, el TCP. Los capítulos posteriores hablan sobre el enrutamiento de Internet y la administración de redes, que también usan protocolos de transporte.

24.2 Protocolos de transporte y comunicación de extremo a extremo

Como se muestra en capítulos anteriores, tanto IPv4 como IPv6 ofrecen un servicio de entrega de paquetes a lo largo de Internet (es decir, cuando un datagrama pasa desde un host emisor, a través de una o más redes físicas, hasta un host receptor). A pesar de su habilidad de pasar tráfico a través de Internet, el IP carece de una característica esencial: no puede diferenciar entre varios programas de aplicación que se ejecuten en un host dado. Si un usuario ejecuta una aplicación de correo electrónico y un navegador Web al mismo tiempo, o si ejecuta varias copias de una aplicación específica, todas deben ser capaces de comunicarse en forma independiente.

IP es incapaz de soportar varias aplicaciones, ya que los campos de un encabezado de datagrama sólo identifican computadoras. Es decir, desde el punto de vista del protocolo IP, los campos de origen y de destino de un datagrama identifican a un equipo host. Una dirección IP no contiene bits adicionales para identificar un programa de aplicación en el host. Decimos que IP considera a una computadora como el *punto final* de una comunicación. En contraste, los protocolos de la capa de transporte se conocen como *protocolos de extremo a extremo*, ya que un protocolo de transporte permite que un programa de aplicación individual sea el punto final de la comunicación. En vez de agregar funciones adicionales a IP para identificar las aplicaciones, los diseñadores de los protocolos TCP/IP colocaron los protocolos de extremo a extremo en una capa independiente: la capa 4.

24.3 El protocolo de datagramas de usuario

Como veremos, la suite TCP/IP contiene dos protocolos de transporte principales. El *protocolo de datagramas de usuario (UDP)* y el *protocolo de control de transmisión (TCP)*, los cuales difieren de manera considerable en cuanto al servicio que ofrecen a las aplicaciones. UDP es menos complejo y más fácil de entender. La simpleza y facilidad de entendimiento tienen un costo: UDP no ofrece el tipo de servicio que una aplicación común espera.

Podemos caracterizar el protocolo UDP de la siguiente forma:

- *De extremo a extremo.* UDP es un protocolo de transporte que puede diferenciar entre varios programas de aplicación que se ejecutan en una computadora específica.
- *Sin conexión.* La interfaz que UDP brinda a las aplicaciones sigue un paradigma sin conexión.
- *Orientado a mensajes.* Una aplicación que usa UDP envía y recibe mensajes individuales.
- *Mejor esfuerzo.* UDP ofrece a las aplicaciones la misma semántica de entrega del mejor esfuerzo que IP.
- *Interacción arbitraria.* UDP permite a una aplicación hacer envíos a muchas otras aplicaciones, recibirlas de muchas otras aplicaciones o comunicarse con una sola aplicación.
- *Independiente del sistema operativo.* UDP ofrece un medio para identificar a los programas de aplicación que no dependen de los identificadores que utiliza el sistema operativo local.

La característica más importante de UDP (su semántica del mejor esfuerzo) surge debido a que UDP usa a IP para la transmisión sin que se tomen pasos adicionales para corregir problemas. De hecho, algunas veces UDP se caracteriza como una capa de protocolo *delgada* que ofrece a las aplicaciones la habilidad de enviar y recibir datagramas IP. Podemos resumir esto así:

UDP ofrece un servicio de extremo a extremo que permite a un programa de aplicación enviar y recibir mensajes individuales, cada uno de los cuales viaja en un datagrama independiente. Una aplicación puede optar por limitar la comunicación a otro programa de aplicación o comunicarse con varias aplicaciones.

24.4 El paradigma sin conexión

UDP usa un paradigma de comunicación *sin conexión*, lo cual significa que una aplicación que usa UDP no necesita preestablecer la comunicación antes de enviar los datos, y tampoco necesita informar a la red cuando termina. En su lugar, una aplicación puede generar y enviar datos en cualquier momento. Además, UDP permite a una aplicación insertar un retraso de tiempo con una longitud arbitraria entre la transmisión de dos mensajes. UDP no mantiene el estado y no envía mensajes de control adicionales; la comunicación consiste sólo en los mensajes de datos en sí. En especial, si un par de aplicaciones dejan de enviar datos, no se intercambian otros paquetes. Como resultado, UDP tiene una sobrecarga mucho muy baja. Para resumir esto:

UDP es un protocolo sin conexión, lo cual significa que una aplicación puede enviar datos en cualquier momento y UDP no transmite paquetes que no sean los que transportan los datos del usuario.

24.5 Interfaz orientada a mensajes

UDP ofrece a los programas de aplicación una interfaz *orientada a mensajes*. Cada vez que una aplicación solicita al UDP que envíe un bloque de datos, éste coloca los datos en un solo mensaje para su transmisión. UDP no divide un mensaje en varios paquetes ni combina mensajes para la entrega. Cada mensaje que una aplicación envía se transporta a través de Internet y se entrega al receptor.

La interfaz orientada al mensaje tiene varias consecuencias importantes para los programadores. Del lado positivo, las aplicaciones que usan UDP pueden depender del protocolo para mantener los límites de datos, ya que cada mensaje que UDP entrega a una aplicación receptora será exactamente el mismo que transmitió el emisor. Del lado negativo, cada mensaje UDP debe caber en un solo datagrama IP. Por lo tanto, el tamaño del datagrama IP forma un límite absoluto con base en el tamaño de un mensaje UDP. Lo que es más importante, el tamaño del mensaje UDP puede conducir a un uso ineficiente de la red. Si una aplicación envía mensajes extremadamente pequeños, los datagramas resultantes tendrán una relación grande entre los octetos del encabezado y los octetos de datos. Si una aplicación envía mensajes mucho muy grandes, los datagramas resultantes pueden ser más grandes que la MTU de la red y el protocolo IP los fragmentará.

Permitir que los mensajes UDP sean grandes produce una anomalía particular. Por lo general, un programador de aplicaciones puede obtener una mayor eficiencia mediante el uso de transferencias

grandes. Por ejemplo, se anima a los programadores a que declaren búferes de E/S grandes y especifiquen transferencias que coincidan con el tamaño del búfer. Sin embargo, con UDP el envío de mensajes grandes produce una menor eficiencia, ya que los mensajes grandes provocan fragmentación. Lo que es aún más sorprendente, la fragmentación puede ocurrir en la computadora emisora: una aplicación envía un mensaje grande, UDP coloca todo el mensaje en un datagrama de usuario y lo encapsula en un datagrama de Internet, por lo que IP debe realizar la fragmentación antes de que se pueda enviar el datagrama. En conclusión:

Aunque la intuición de un programador sugiere que al usar mensajes más grandes aumenta la eficiencia, si un mensaje UDP es más grande que la MTU de la red, IP fragmentará el datagrama resultante, lo cual reduce la eficiencia.

Como consecuencia, muchos programadores que usan UDP elegirán un tamaño de mensaje que produzca datagramas que puedan alojarse en una MTU estándar. En especial, debido a que en su mayoría Internet soporta ahora una MTU de 1500 octetos, es común que los programadores seleccionen un tamaño de mensaje de 1400 o 1450 para dejar espacio para los encabezados IP y UDP.

24.6 Semántica de la comunicación UDP

UDP usa IP para todas las entregas. Además, UDP proporciona a las aplicaciones exactamente la misma semántica de entrega del mejor esfuerzo que IP, lo que significa que los mensajes pueden:

- Perderse
- Duplicarse
- Retrasarse
- Entregarse desordenados
- Corromperse

Desde luego que UDP no añade problemas en la entrega intencionalmente, sino que simplemente usa IP para enviar mensajes y, por ende, no detecta ni corrige los problemas de entrega. La semántica de entrega del mejor esfuerzo de UDP tiene consecuencias importantes para las aplicaciones. Una aplicación debe ser inmune a los problemas o el programador debe realizar pasos adicionales para detectar y corregirlos. Como ejemplo de una aplicación que puede tolerar errores en los paquetes, considere una transmisión de audio. Si el emisor coloca una pequeña cantidad de audio en cada mensaje, la pérdida de un solo paquete produce un pequeño hueco en la reproducción, lo cual se escuchará como un tronido o un brinco. Aunque no es lo más deseable, el ruido simplemente resulta molesto pero no tiene mayores consecuencias. En el extremo opuesto, considere una aplicación de compras en línea. Dichas aplicaciones no están escritas para usar UDP debido a que los errores de los paquetes pueden tener graves consecuencias (por ejemplo, la duplicación de un mensaje que transporta un pedido del catálogo puede dar como resultado dos pedidos y en consecuencia podría hacerse un doble cargo en la tarjeta de crédito del comprador).

Podemos resumir lo anterior así:

Puesto que UDP ofrece la misma semántica de entrega del mejor esfuerzo que IP, un mensaje UDP puede perderse, duplicarse, retrasarse, entregarse desordenado o algunos bits podrían corromperse durante el transporte. UDP sólo es adecuado para las aplicaciones de voz o video que pueden tolerar errores en la entrega.

24.7 Modos de interacción y entrega por multidifusión

UDP permite cuatro estilos de interacción:

- 1 a 1
- 1 a varios
- Varios a 1
- Varios a varios

Es decir, una aplicación que usa UDP tiene una opción. Puede elegir una interacción de 1 a 1 donde la aplicación únicamente intercambia mensajes con otra aplicación, una interacción de 1 a varios donde la aplicación envía un mensaje a varios receptores, o una interacción de varios a 1 donde la aplicación recibe mensajes de varios emisores. Por último, un conjunto de aplicaciones puede establecer una interacción de varios a varios donde todas las aplicaciones puedan intercambiar mensajes entre sí.

Aunque es posible lograr una interacción de 1 a varios mediante el envío de una copia individual de un mensaje a cada receptor deseado, UDP permite que el intercambio sea eficiente. En vez de requerir que una aplicación envíe repetidamente un mensaje a varios receptores, UDP permite que una aplicación transmita el mensaje mediante multidifusión IP (o difusión IPv4). Para ello, el emisor envía una dirección IP de multidifusión o de difusión como la dirección IP de destino. Por ejemplo, la entrega a todos los nodos de la red local puede especificarse mediante el uso de la dirección de difusión limitada de IPv4, 255.255.255.255, o la dirección de multidifusión de enlace local de todos los nodos. La entrega mediante difusión o multidifusión es especialmente útil para las redes Ethernet, ya que el hardware que utiliza soporta ambos tipos en forma eficiente.

24.8 Identificación del punto final con números de puerto de protocolo

¿Exactamente cómo identifica UDP un programa de aplicación como el punto final? Tal vez parezca que UDP podría usar el mismo mecanismo que usa el sistema operativo. Por desgracia y como UDP debe abarcar computadoras heterogéneas, no existe un mecanismo común. Por ejemplo, algunos sistemas operativos usan identificadores de procesos, otros usan nombres de trabajos y otros usan identificadores de tareas. Por lo tanto, un identificador que resulta significativo en un sistema tal vez no sea significativo en otro.

Para evitar la ambigüedad, UDP define un conjunto abstracto de identificadores conocido como *números de puerto de protocolo* que son independientes del sistema operativo. Cada computadora que implementa UDP debe proporcionar una correlación entre los números de puerto de protocolo y los identificadores del programa que use el sistema operativo. Por ejemplo, el estándar UDP define el número de puerto de protocolo siete como el puerto para un servicio *echo* y el número de puerto treinta y siete como el puerto para un servicio *timeserver*. Todas las computadoras que ejecutan UDP reconocen los números de puerto de protocolo estándar, sin importar cuál sea el sistema operativo en ejecución. Así, cuando llega un mensaje UDP para el puerto siete, el software de protocolo UDP debe saber qué aplicación en la computadora local implementa el servicio de eco y debe pasar el mensaje entrante al programa.

El modo de comunicación se determina mediante la forma en que una aplicación llena las direcciones y los números de puerto de protocolo para un socket. Para participar en una comunicación de 1 a 1, una aplicación especifica el número de puerto local, la dirección IP remota y el número de puerto de protocolo remoto. UDP sólo pasa los mensajes de aplicación que llegan del emisor especificado. Para participar en la comunicación de varios a 1, la aplicación especifica el número de puerto local pero informa a UDP que el punto final remoto puede ser cualquier sistema. Después, UDP pasa a la aplicación todos los mensajes que llegan para el puerto especificado.[†]

24.9 Formato de los datagramas UDP

Cada mensaje UDP se denomina *datagrama de usuario* y consiste en dos partes: un encabezado corto que especifica los programas de aplicación emisores y receptores, y una carga útil que transporta los datos que se van a enviar. La figura 24.1 ilustra el formato de datagrama de usuario.

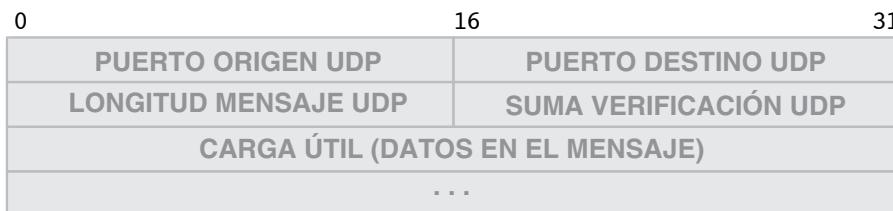


Figura 24.1 El formato de un datagrama de usuario UDP con un encabezado de 8 octetos.

Los primeros dos campos del encabezado UDP contienen números de puerto de protocolo de 16 bits. El campo *PUERTO ORIGEN UDP* contiene el número de puerto de la aplicación emisora y el campo *PUERTO DESTINO UDP* contiene el número de puerto de la aplicación a la que se va a enviar el mensaje. El campo *LONGITUD MENSAJE UDP* especifica el tamaño total del mensaje UDP, medido en bytes de 8 bits.

[†] Sólo una aplicación puede solicitar todos los mensajes para un puerto dado.

24.10 La suma de verificación UDP y el seudoencabezado

Aunque el encabezado UDP contiene un campo de diecisésis bits llamado *SUMA VERIFICACIÓN UDP*, la suma de verificación en sí es opcional. Un emisor puede optar por calcular una suma de verificación o establecer todos los bits del campo de suma de verificación en cero. Cuando llega un mensaje al destino, el software UDP examina el campo de suma de verificación y sólo verifica la suma de verificación cuando el valor es distinto de cero.[†]

Cabe mencionar que el encabezado UDP no contiene identificación del emisor o receptor más que los números de puerto de protocolo. En especial, UDP asume que las direcciones IP de origen y de destino están contenidas en el datagrama IP que transporta. Por lo tanto, las direcciones IP no se transportan en el encabezado UDP.

Al omitir las direcciones IP de origen y de destino, el protocolo UDP se hace más pequeño y eficiente pero introduce la posibilidad de error. En especial, si IP falla y entrega un mensaje UDP a un destino incorrecto, UDP no puede usar campos de encabezado para determinar que ocurrió un error.

Para verificar que los mensajes lleguen al destino correcto sin incurrir en la sobrecarga de los campos de encabezado adicionales, UDP extiende la suma de verificación. Al calcular la suma de verificación, el software UDP incluye un *seudoencabezado* que contiene los campos origen IP, destino IP y tipo (por ejemplo, PROTO o SIGUIENTE-ENCABEZADO) del datagrama IP, así como una longitud de datagrama UDP. Es decir, el emisor calcula una suma de verificación como si el encabezado UDP contuviera campos adicionales. De manera similar, para verificar una suma, el receptor debe obtener la longitud de UDP además de los campos origen, destino y tipo del datagrama IP. Entonces el receptor los adjunta al mensaje UDP antes de comprobar la suma de verificación. La figura 24.2 ilustra los campos en el seudoencabezado.

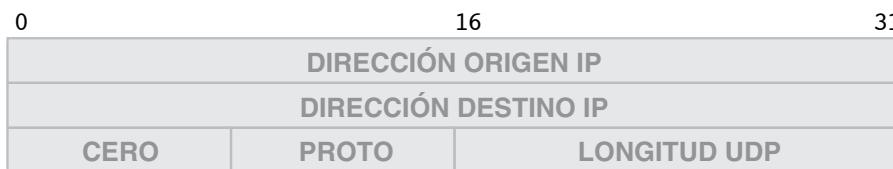


Figura 24.2 Ilustración del seudoencabezado que se utiliza para calcular la suma de verificación de UDP.

24.11 Encapsulamiento UDP

Al igual que ICMP, cada datagrama UDP se encapsula en un datagrama IP para su transmisión a través de Internet. La figura 24.3 ilustra el encapsulamiento.

[†] Al igual que IP, UDP usa una suma de verificación de complementos a uno; si la suma de verificación calculada tiene un valor de cero, un emisor usa la forma de cero que tiene todos los bits en uno.

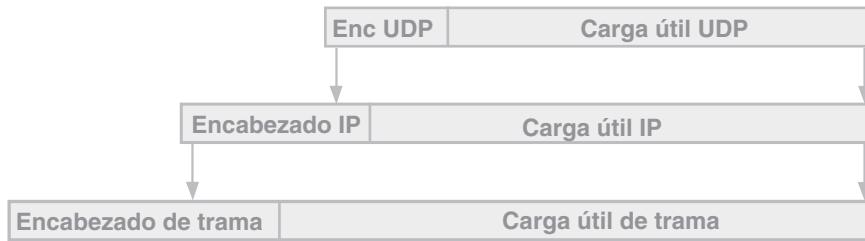


Figura 24.3 El encapsulamiento de un mensaje de UDP en un datagrama IP.

24.12 Resumen

El *protocolo de datagrama de usuario* (UDP) brinda un transporte de mensajes de extremo a extremo sin conexión, desde una aplicación que se ejecuta en una computadora hasta una aplicación que se ejecuta en otra computadora. UDP ofrece la misma semántica de entrega del mejor esfuerzo que IP, lo cual significa que los mensajes pueden perderse, duplicarse o entregarse en desorden. Una de las ventajas de la metodología sin conexión se debe a la habilidad de tener interacciones de 1 a 1, de 1 a varios y de varios a 1 entre las aplicaciones.

Para permanecer independiente de los sistemas operativos, UDP usa números de puerto de protocolo pequeños para diferenciar entre los programas de aplicación. El software de protocolo de una computadora dada debe asignar cada número de puerto de protocolo al mecanismo apropiado (por ejemplo, la ID del proceso) que se utiliza en la computadora.

La suma de verificación UDP es opcional; si un emisor llena el campo de suma de verificación con ceros, el receptor no comprueba la suma de verificación. Para comprobar que el datagrama UDP haya llegado a la ubicación correcta, se calcula una suma de verificación UDP sobre el datagrama más un seudoencabezado.

UDP requiere dos niveles de encapsulamiento. Cada mensaje UDP se encapsula en un datagrama IP para su transmisión a través de Internet. El datagrama se encapsula en una trama para transmitirse a través de una red individual.

EJERCICIOS

- 24.1** ¿Cuál es la diferencia conceptual entre los protocolos IP y los de extremo a extremo?
- 24.2** Enliste las características de UDP.
- 24.3** ¿Necesitan las aplicaciones intercambiar mensajes de control UDP antes de intercambiar datos? Explique.
- 24.4** Calcule el tamaño del mensaje UDP más grande posible al usar IPv4 e IPv6. (Sugerencia: todo el mensaje UDP debe caber en un datagrama IP).

- 24.5** ¿Qué ocurre si un mensaje UDP que contenga una carga útil de 1500 bytes se envía a través de una red Ethernet?
- 24.6** Si una aplicación usa UDP para enviar un mensaje de 8K bytes a través de una red Ethernet, ¿cuántas tramas recorrerán la red?
- 24.7** Cuando llega un mensaje UDP a una computadora, ¿puede el software IP descartar por completo el encabezado de la trama y el encabezado IP antes de pasar el mensaje al software UDP para su procesamiento? Explique.
- 24.8** ¿Cuál es la semántica de UDP?
- 24.9** ¿Qué valores de punto final debe especificar una aplicación que participa en una comunicación de 1 a 1? ¿De 1 a varios? ¿De varios a 1?
- 24.10** ¿Qué es un seudoencabezado y cuándo se utiliza?
- 24.11** Dada una trama de Ethernet, ¿qué campos deben examinarse para determinar si la trama contiene un datagrama IPv4 que transporta un mensaje UDP?
- 24.12** Responda a la pregunta anterior para IPv6.

Contenido del capítulo

- 25.1 Introducción, 425
- 25.2 El protocolo de control de transmisión, 425
- 25.3 El servicio que TCP ofrece a las aplicaciones, 426
- 25.4 Servicio de extremo a extremo y conexiones virtuales, 427
- 25.5 Técnicas que utilizan los protocolos de transporte, 428
- 25.6 Técnicas para evitar la congestión, 432
- 25.7 El arte del diseño de protocolos, 433
- 25.8 Técnicas utilizadas en TCP para manejar la pérdida de paquetes, 434
- 25.9 Retransmisión adaptativa, 435
- 25.10 Comparación de los tiempos de retransmisión, 436
- 25.11 Búferes, control de flujo y ventanas, 437
- 25.12 La negociación en tres pasos de TCP, 438
- 25.13 Control de congestión de TCP, 440
- 25.14 Versiones del control de congestión de TCP, 441
- 25.15 Otras variaciones: SACK y ECN, 441
- 25.16 Formato del segmento de TCP, 442
- 25.17 Resumen, 443

25

TCP: servicio de transporte confiable

25.1 Introducción

Los capítulos anteriores de este libro describieron los servicios de entrega de paquetes sin conexión que proporciona IP, así como el protocolo de datagramas de usuario que se ejecuta sobre éste. En este capítulo hablaremos sobre los protocolos de transporte en general y analizaremos TCP, el principal protocolo de transporte que se utiliza en Internet. En el capítulo veremos cómo el protocolo TCP proporciona una entrega confiable.

TCP logra una tarea que parecía imposible: usa el poco confiable servicio de datagramas que ofrece IP para el envío por Internet, pero brinda a los programas de aplicación un servicio de entrega de datos confiable. TCP debe compensar la pérdida, el retraso, la duplicación y la entrega en desorden, y debe hacerlo sin sobrecargar las redes ni los enrutadores involucrados. Después de repasar el servicio que ofrece TCP a las aplicaciones, el capítulo analizará las técnicas que este protocolo usa para lograr la confiabilidad.

25.2 El protocolo de control de transmisión

Los programadores están entrenados para pensar que la confiabilidad es fundamental en un sistema de computadoras. Por ejemplo, al escribir una aplicación que envía datos a un dispositivo de E/S como una impresora, un programador asume que los datos llegarán correctamente o que el sistema operativo informará a la aplicación que ocurrió un error. Es decir, un programador da por hecho que el sistema utilizado garantiza que los datos se entregarán de manera confiable.

Para que los programadores puedan seguir las técnicas convencionales para crear aplicaciones que se comuniquen a través de Internet, el software de protocolo debe ofrecer la misma semántica que un sistema de computadora convencional: debe garantizar una comunicación oportuna y confiable. Los datos deben entregarse exactamente en el mismo orden en el que se enviaron y no debe haber pérdida ni duplicación.

En la suite TCP/IP, el *protocolo de control de transmisión (TCP)* ofrece un servicio de transporte confiable. TCP es notable debido a que resuelve bien un problema, y aunque han creado otros protocolos, ningún protocolo de transporte de propósito general ha probado funcionar mejor. En consecuencia, la mayoría de las aplicaciones de Internet están diseñadas para usar TCP.

Para resumir:

En Internet, el protocolo de control de transmisión (TCP) es un protocolo de la capa de transporte que brinda una confiabilidad.

25.3 El servicio que TCP ofrece a las aplicaciones

El servicio que ofrece TCP tiene siete características importantes:

- *Orientación a la conexión.* TCP ofrece un servicio orientado a la conexión en el que una aplicación debe primero solicitar una conexión a un destino y luego usar esa conexión para transferir datos.
- *Comunicación de punto a punto.* Cada conexión de TCP tiene únicamente dos extremos.
- *Confiabilidad total.* TCP garantiza que los datos enviados a través de una conexión se entregarán exactamente como se enviaron, completos y en orden.
- *Comunicación dúplex.* Una conexión de TCP permite que los datos fluyan en cualquier dirección y que cualquier programa de aplicación envíe datos en cualquier momento.
- *Interfaz de flujo continuo.* TCP proporciona una interfaz de flujo en la que una aplicación envía una secuencia continua de octetos a través de una conexión. TCP no agrupa los datos en registros o mensajes y no garantiza la entrega de datos en piezas del mismo tamaño que las que transfirió la aplicación emisora.
- *Inicio de conexión confiable.* TCP permite que dos aplicaciones inicien la comunicación de manera confiable.
- *Cierre ordenado de la conexión.* Antes de cerrar una conexión, TCP se asegura de que se hayan entregado todos los datos y que ambos lados hayan acordado cerrar la conexión.

Para resumir:

TCP brinda un servicio de transporte dúplex de flujo continuo, confiable y orientado a la conexión, el cual permite que dos programas de aplicación formen una conexión, envíen datos en cualquier dirección y luego terminen la conexión. Cada conexión de TCP se inicia en forma confiable y termina en forma ordenada.

25.4 Servicio de extremo a extremo y conexiones virtuales

Al igual que UDP, TCP se clasifica como protocolo de *extremo a extremo*, ya que ofrece una comunicación entre una aplicación en una computadora y una aplicación en otra computadora. Está *orientado a la conexión* debido a que las aplicaciones deben solicitar que TCP forme una conexión antes de que puedan transferir datos, y debe cerrar la conexión cuando se complete la transferencia.

Las conexiones que proporciona TCP se llaman *conexiones virtuales* debido a que se logran mediante software. La red Internet no proporciona hardware ni soporte para las conexiones, sino que son los módulos de software TCP de dos máquinas los que intercambian mensajes para lograr la ilusión de una conexión.

Cada mensaje TCP se encapsula en un datagrama IP y se envía a través de Internet. Cuando el datagrama llega al host de destino, IP pasa el contenido a TCP. Cabe mencionar que, aunque TCP usa IP para transportar mensajes, IP no lee ni interpreta esos mensajes. De hecho, IP trata a cada mensaje TCP como si fueran datos a transferir. Por otra parte, TCP trata a IP como un sistema de comunicación de paquetes que comunica a los módulos TCP que están en cada extremo de una conexión. La figura 25.1 ilustra cómo es que TCP ve a la red Internet.

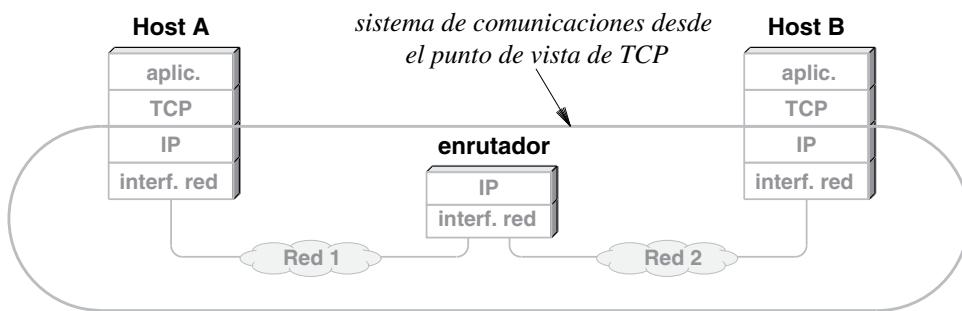


Figura 25.1 Ilustración de la forma en que TCP ve a la red Internet que usa.

Como se muestra en la figura, se necesita un software TCP en cada extremo de la conexión virtual, pero no en los enrutadores intermedios. Desde el punto de vista de TCP, toda la red Internet es un sistema de comunicación que puede aceptar y entregar mensajes sin cambiar o interpretar su contenido.

25.5 Técnicas que utilizan los protocolos de transporte

Un protocolo de transporte de extremo a extremo debe estar diseñado cuidadosamente para lograr una transferencia eficiente y confiable. Los principales problemas son:

- *Comunicación poco confiable.* Los mensajes enviados a través de Internet pueden perderse, duplicarse, corromperse, retrasarse o entregarse desordenados.
- *Reinicio del sistema final.* En cualquier momento durante la comunicación, alguno de los dos sistemas finales podría fallar y reiniciarse. No debe haber confusión entre sesiones, incluso aunque algunos sistemas integrados puedan reiniciar en menos tiempo del que se requiere para que un paquete atraviese Internet.
- *Sistemas finales heterogéneos.* Una aplicación que se ejecuta en un procesador potente puede generar datos con tanta rapidez que rebasaría a una aplicación que se ejecute en un procesador lento.
- *Congestión en Internet.* Si los emisores transmiten datos en forma agresiva, los commutadores y enrutadores intermedios podrían sobresaturarse de paquetes, algo similar a una autopista congestionada.

Ya vimos ejemplos de las técnicas básicas que los sistemas de comunicaciones de datos utilizan para solucionar algunos de los problemas. Por ejemplo, para compensar los bits que se modifican durante la transmisión, un protocolo podría incluir *bits de paridad*, o una *suma de verificación*, o una *comprobación por redundancia cíclica (CRC)*. Los protocolos de transporte más sofisticados hacen más que detectar errores: emplean técnicas que pueden reparar o sortear los problemas. En especial, los protocolos de transporte usan una variedad de herramientas para manejar algunos de los problemas de comunicación más complicados. Las siguientes secciones hablan sobre los mecanismos básicos.

25.5.1 Uso de secuencias para manejar duplicados y entregas desordenadas

Para hacerse cargo de los paquetes duplicados y las entregas desordenadas, los protocolos de transporte usan las *secuencias*. El lado emisor adjunta un número de secuencia a cada paquete. El lado receptor almacena tanto el número de secuencia del último paquete recibido en orden como también una lista de paquetes adicionales que llegaron desordenados. Cuando llega un paquete, el receptor examina el número de secuencia para determinar cómo debe manejar ese paquete. Si el paquete es el siguiente esperado (es decir, llegó en orden), el software de protocolo entrega el paquete a la siguiente capa superior y revisa su lista para ver qué paquetes adicionales pueden entregarse también. Si el paquete llegó desordenado, el software de protocolo agrega el paquete a la lista. Las secuencias también resuelven el problema de la duplicación, ya que un receptor revisa que no haya duplicados examinando el número de secuencia de un paquete recién llegado. Si el paquete ya se entregó o si el número de secuencia coincide con uno de los paquetes que esperan en la lista, el software descarta la nueva copia.

25.5.2 Retransmisión para manejar los paquetes perdidos

Para hacerse cargo de la pérdida de paquetes, los protocolos usan el *reconocimiento positivo con retransmisión*. Cada vez que una trama llega intacta, el software de protocolo receptor envía un pequeño mensaje de *reconocimiento (ACK)* que reporta la recepción exitosa. El emisor debe asegurar que cada paquete se transfiera con éxito. Cada vez que envía un paquete, el software de protocolo del lado emisor inicia un temporizador. Si llega un reconocimiento antes de que expire el temporizador, el software lo cancela; pero si éste expira antes de que llegue un reconocimiento, el software envía otra copia del paquete e inicia el temporizador de nuevo. La acción de enviar una segunda copia se conoce como *retransmitir*, y la copia se conoce comúnmente como una *retransmisión*.

Desde luego que la retransmisión no puede tener éxito si una falla de hardware desconectó la red en forma permanente, o si la computadora receptora tiene una falla. Por lo tanto, es común que los protocolos que retransmiten mensajes limiten el número máximo de retransmisiones. Al llegar a ese límite, el protocolo deja de retransmitir y declara que la comunicación es imposible.

Cabe mencionar que si los paquetes se retrasan, la retransmisión puede introducir paquetes duplicados. Por lo tanto, es común que los protocolos de transporte que incorporan la retransmisión se diseñen para hacerse cargo del problema de los paquetes duplicados.

25.5.3 Técnicas para evitar la repetición

Los retrasos extraordinariamente largos pueden provocar *errores de repetición*, en los que un paquete retrasado afecta la comunicación subsecuente. Por ejemplo, considere la siguiente secuencia de eventos:

- Dos computadoras aceptan comunicarse a la 1 PM.
- Una computadora envía una secuencia de diez paquetes a la otra.
- Un problema de hardware provoca que el paquete 3 se retrase.
- Las rutas cambian para evitar el problema de hardware.
- El software de protocolo en la computadora emisora retransmite el paquete 3 y envía el resto de los paquetes sin error.
- A la 1:05 PM las dos computadoras aceptan comunicarse de nuevo.
- Después de que llega el segundo paquete, llega la copia retrasada del paquete 3 de la conversación anterior.
- Llega el paquete 3 de la segunda conversación.

A menos que un protocolo de transporte se diseñe cuidadosamente para evitar dichos problemas, un paquete de una conversación anterior podría aceptarse en una conversación posterior y el paquete correcto se descartaría como duplicado.

La repetición también pudo ocurrir con los paquetes de control (es decir, paquetes que establecen o terminan la comunicación). Para entender el alcance del problema, considere una situación en la que dos programas de aplicación forman una conexión TCP, se comunican, cierran la conexión y luego forman una nueva conexión. El mensaje que especifica el cierre de la conexión podría dupli-

carse y una copia podría retrasarse lo suficiente como para que se estableciera la segunda conexión. Un protocolo debería estar diseñado de modo que el mensaje duplicado no provocara el cierre de la segunda conexión.

Para evitar la repetición, los protocolos marcan cada sesión con un identificador único (por ejemplo, la hora en la que se estableció la sesión) y requieren que ese identificador único esté presente en cada paquete. El software de protocolo descarta cualquier paquete entrante que contenga un identificador incorrecto. Para evitar la repetición, no debe reutilizarse un identificador sino hasta que haya transcurrido una cantidad de tiempo razonable (por ejemplo, varias horas).

25.5.4 Control de flujo para evitar exceso de datos

Existen varias técnicas para evitar que una computadora veloz envíe tantos datos que se exceda la capacidad de un receptor lento. Usamos el término *control de flujo* para referirnos a las técnicas que se encargan del problema. La forma más simple de control de flujo es un sistema de *parada y arranque* o *stop-and-go*, donde un emisor espera después de transmitir cada paquete. Cuando el receptor está listo para otro paquete, envía un mensaje de control, que por lo general es una forma de reconocimiento.

Aunque los protocolos de parada y arranque evitan el exceso de datos, producen una velocidad de transferencia muy baja. Para entender por qué, considere lo que ocurre en una red que tiene un tamaño de paquete de 1000 octetos, una capacidad de velocidad de transferencia de 2 Mbps y un retraso de 50 milisegundos. El hardware de red puede transportar 2 Mbps de una computadora a otra. Sin embargo, después de transmitir un paquete, el emisor debe esperar 100 milisegundos antes de enviar otro paquete (es decir, 50 milisegundos para que el paquete llegue al receptor y 50 milisegundos para que un reconocimiento viaje de regreso). Por ende, la velocidad máxima a la que pueden enviarse los datos usando el sistema de parada y arranque es un paquete cada 100 milisegundos. Si se expresa como una velocidad de bits, la velocidad máxima que puede lograr el sistema de parada y arranque es de 80,000 bps, lo que constituye tan sólo el 4% de la capacidad del hardware.

Para obtener tasas de velocidad de transferencia altas, los protocolos de transporte usan una técnica de control de flujo conocida como *ventana corrediza*. El emisor y el receptor están programados para usar un *tamaño de ventana* fijo, que es la cantidad máxima de datos que pueden enviarse antes de que llegue un reconocimiento. Por ejemplo, el emisor y el receptor podrían acordar un tamaño de ventana de cuatro paquetes. El emisor comienza con los datos que van a enviarse, extrae los datos para llenar nuestros paquetes (es decir, la primera ventana) y transmite una copia de cada paquete. En la mayoría de los protocolos de transporte, el emisor retiene una copia en caso de que se necesite la retransmisión. El receptor debe haber asignado previamente un espacio suficiente en el búfer para toda la ventana. Si llega un paquete en secuencia, el receptor pasa el paquete a la aplicación receptora y transmite un reconocimiento al emisor. Cuando llega un reconocimiento, el emisor descarta su copia del paquete reconocido y transmite el siguiente paquete. La figura 25.2 ilustra por qué el mecanismo se conoce como *ventana corrediza*.

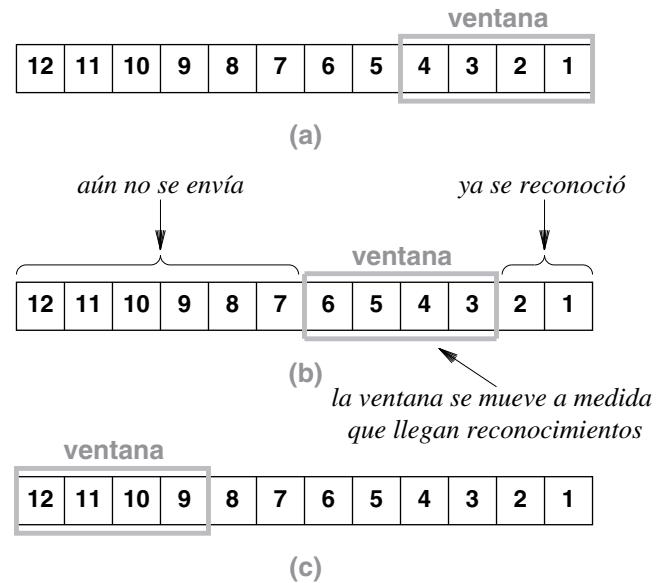


Figura 25.2 Ilustración de una ventana corrediza en posiciones (a) inicial, (b) intermedia y (c) final.

La ventana corrediza puede incrementar considerablemente la velocidad de transferencia. Para entender por qué, compare la secuencia de transmisiones entre un esquema de parada y arranque, y un esquema de ventana corrediza. La figura 25.3 contiene una comparación para una transmisión de 4 paquetes.

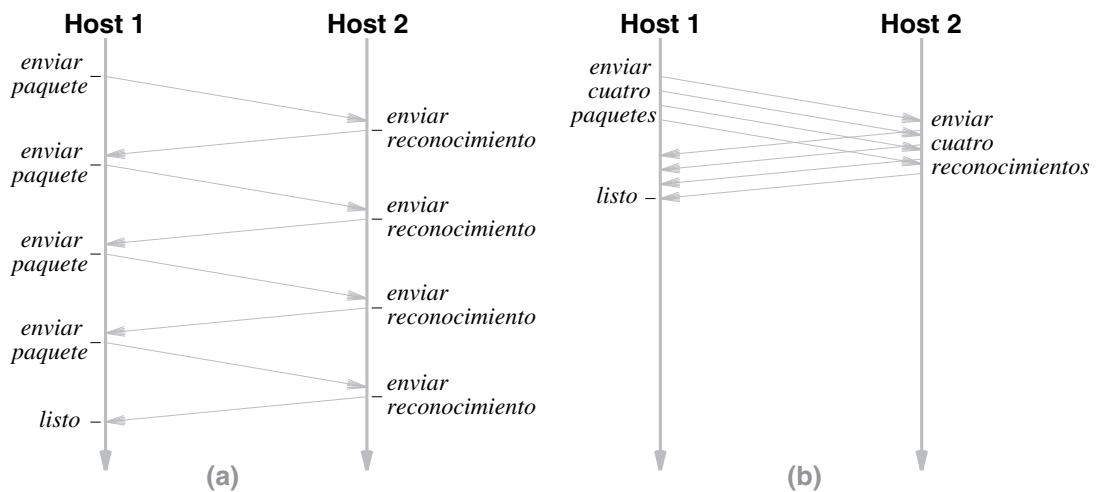


Figura 25.3 Comparación de la transmisión usando (a) parada y arranque, y (b) ventana corrediza.

En la figura 25.3(a), un emisor transmite cuatro paquetes pero espera un reconocimiento antes de enviar cada paquete sucesivo. Si el retraso requerido para enviar un solo paquete en un viaje a través de la red es N , el tiempo total requerido para enviar cuatro paquetes es $8N$. En la figura 25.3(b), un emisor transmite todos los paquetes de la ventana antes de ponerse a esperar. La figura muestra un pequeño retraso entre transmisiones sucesivas de paquetes ya que la transmisión nunca es instantánea; se requiere un tiempo corto (por lo general de unos cuantos microsegundos) para que el hardware complete la transmisión de un paquete y comience a transmitir el siguiente paquete. Por lo tanto, el tiempo total requerido para enviar cuatro paquetes es $2N + \epsilon$, donde ϵ indica el pequeño retraso.

Para comprender el significado de la ventana corrediza, imagine una comunicación extendida que involucra muchos paquetes. En tales casos, el tiempo total requerido para la transmisión es tan grande que podemos ignorar a ϵ . Para dichas redes, un protocolo de ventana corrediza puede incrementar el rendimiento de forma considerable. La mejora potencial es:

$$T_w = T_g \times W \quad (26.1)$$

donde T_w es la velocidad de transferencia que puede obtenerse con un protocolo de ventana corrediza, T_g es la velocidad de transferencia que puede lograrse con un protocolo de parada y arranque, y W es el tamaño de la ventana. La ecuación explica por qué el protocolo de ventana corrediza que se ilustra de la figura 25.3(b) tiene aproximadamente cuatro veces la velocidad de transferencia del protocolo de parada y arranque de la figura 25.3(a). Desde luego que la velocidad de transferencia no puede incrementarse de manera arbitraria con sólo aumentar el tamaño de la ventana. La capacidad de la red impone un límite superior: los bits no pueden enviarse con más rapidez de la que el hardware puede transportarlos. Entonces podemos reescribir la ecuación así:

$$T_w = \min(C, T_g \times W) \quad (26.2)$$

donde C es la capacidad del hardware utilizado.[†]

25.6 Técnicas para evitar la congestión

Para comprender qué tan fácilmente puede ocurrir la congestión, considere cuatro hosts conectados por dos commutadores, como se muestra en la figura 25.4.



Figura 25.4 Cuatro hosts conectados por dos commutadores.

[†] A menudo los profesionales de redes usan el término *ancho de banda* en vez de capacidad; pero hablando en sentido estricto el término no es correcto, ya que los protocolos pueden hacer que la velocidad de datos efectiva de la red sea mucho menor que el ancho de banda del canal.

Suponga que cada conexión en la figura opera a 1 Gbps y considere lo que ocurre si ambas computadoras conectadas al conmutador 1 intentan enviar datos a una computadora conectada al conmutador 2. El conmutador 1 recibe datos a una velocidad agregada de 2 Gbps, pero sólo puede reenviar 1 Gbps al conmutador 2. La situación se conoce como *congestión*. Incluso aunque un conmutador almacene paquetes temporalmente en la memoria, la congestión provoca un retraso mayor. Si la congestión persiste, el conmutador se quedará sin memoria y comenzará a descartar paquetes. Aunque es posible usar la retransmisión para recuperar paquetes perdidos, la retransmisión envía más paquetes a la red. Por consiguiente, si la situación persiste toda una red podría volverse inutilizable. Esta condición se conoce como *colapso por congestión*. En Internet la congestión ocurre generalmente en los enrutadores. Los protocolos de transporte intentan evitar el colapso por congestión mediante el monitoreo de la red y reaccionando con rapidez una vez que comienza la congestión. Hay dos metodologías básicas:

- Hacer que los sistemas intermedios (es decir, los enrutadores) informen a un emisor cuando ocurra la congestión.
- Usar un retraso mayor o la pérdida de paquetes como estimación de la congestión.

El esquema anterior se implementa haciendo que los enrutadores envíen un mensaje especial al origen de los paquetes cuando ocurre la congestión, o haciendo que los enrutadores establezcan un bit en el encabezado de cada paquete que experimente un retraso provocado por la congestión. Cuando se usa la segunda metodología, la computadora que recibe el paquete incluye información dentro del reconocimiento para informar al emisor original.[†]

El uso del retraso y la pérdida para estimar la congestión es razonable en Internet, ya que:

El hardware de red moderno funciona bien; la mayoría de los casos de retraso y pérdida son resultado de la congestión, no de una falla de hardware.

La respuesta apropiada a la congestión consiste en reducir la tasa a la que se transmiten los paquetes. Los protocolos de ventana corrediza pueden lograr el efecto de reducir la tasa, reduciendo temporalmente el tamaño de la ventana.

25.7 El arte del diseño de protocolos

Aunque las técnicas necesarias para resolver problemas específicos son bien conocidas, el diseño de protocolos no es sencillo, debido principalmente a dos razones. Primero, para que la comunicación sea eficiente, hay que elegir cuidadosamente los detalles; los pequeños errores de diseño pueden producir una operación incorrecta, paquetes innecesarios o retrasos. Por ejemplo, si se usan números de secuencia, cada paquete debe contener un número de secuencia en el encabezado. El campo debe ser lo bastante grande como para que los números de secuencia no se reutilicen con frecuencia, pero lo bastante pequeño como para evitar desperdiciar innecesariamente la capacidad. Segundo, los mecanismos del protocolo pueden interactuar en formas inesperadas. Por ejemplo, considere la interacción entre los mecanismos de control de flujo y de control de congestión. Un esquema de ventana corrediza usa considerablemente más capacidad de red para mejorar la velocidad de transferencia. Un mecanismo

[†] Puede ocurrir un retraso extenso entre el tiempo en que ocurre la congestión y cuando se informa al emisor original.

de control de congestión hace lo opuesto al reducir el número de paquetes que se insertan para evitar que la red colapse; el balance entre la ventana corrediza y el control de congestión puede ser engañoso, y es difícil lograr un diseño que haga ambas cosas bien. Es decir, el control de flujo agresivo puede provocar congestión y el control de congestión conservador puede reducir la tasa de transferencia más de lo necesario. Los diseños que intentan cambiar de un comportamiento agresivo a uno conservador cuando ocurre la congestión tienden a oscilar: incrementan con lentitud su uso de la capacidad hasta que la red comienza a experimentar la congestión, reducen su uso hasta que la red se vuelve estable y luego comienzan a incrementarse de nuevo.

El reinicio de un sistema de cómputo representa otro desafío grave para el diseño de protocolos de transporte. Imagine una situación en donde dos programas de aplicación establecen una conexión, comienzan a enviar datos y luego la computadora que recibe los datos se reinicia. Aunque el software de protocolo en la computadora que reinició no tiene conocimiento de una conexión, el software de protocolo en la computadora emisora considera la conexión válida. Si un protocolo no se diseña con cuidado, un paquete duplicado puede hacer que una computadora cree una conexión de manera incorrecta y comience a recibir datos a mitad de un flujo.

25.8 Técnicas utilizadas en TCP para manejar la pérdida de paquetes

¿Cuál de las técnicas antes mencionadas usa TCP para lograr una transferencia confiable? La respuesta es compleja, ya que TCP usa una variedad de esquemas que se combinan de nuevas maneras. Como es de esperarse, TCP usa la *retransmisión* para compensar la pérdida de paquetes. Como TCP proporciona un flujo de datos en ambas direcciones, ambos lados de una comunicación participan en la retransmisión. Cuando TCP recibe datos, envía un *reconocimiento* de vuelta al emisor. Cada vez que envía datos, TCP inicia un temporizador y retransmite los datos si el temporizador expira. Por lo tanto, la retransmisión de TCP básica opera como se muestra en la figura 25.5.

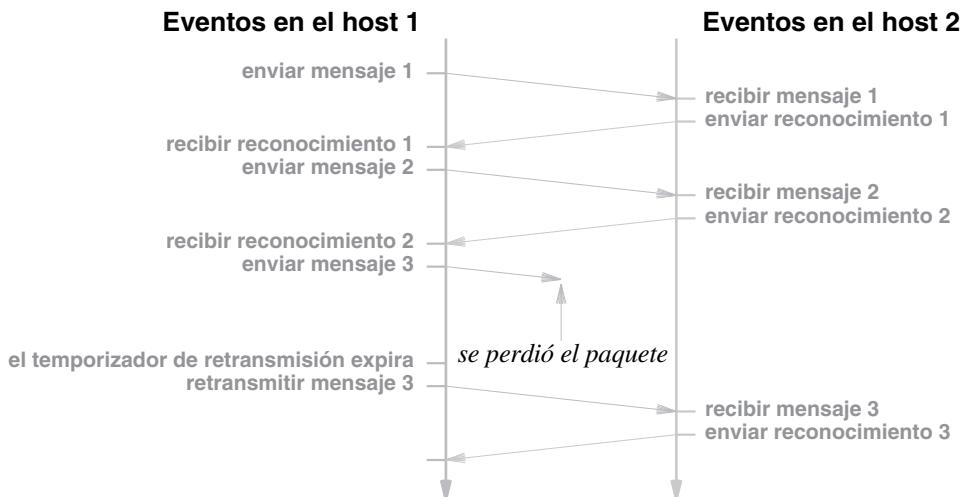


Figura 25.5 Ilustración de la retransmisión de TCP después de una pérdida de paquete.

El esquema de retransmisión de TCP es la clave de su éxito debido a que maneja la comunicación a través de cualquier ruta de Internet. Por ejemplo, una aplicación podría enviar datos a través de un canal de satélite a una computadora en otro país, mientras que otra aplicación envía datos a través de una red de área local a una computadora en el siguiente cuarto. TCP debe estar listo para retransmitir cualquier mensaje que se pierda en cualquier conexión. La pregunta es: ¿cuánto tiempo debe TCP esperar antes de retransmitir? Se espera que los reconocimientos de una computadora en una red de área local lleguen en menos de unos cuantos milisegundos, pero una conexión de satélite requiere cientos de milisegundos. Por otra parte, al esperar demasiado por dicho reconocimiento la red queda inactiva y no se maximiza la velocidad de transferencia. Por consiguiente, en una red de área local el protocolo TCP no debe emplear un retraso de tiempo extenso antes de volver a retransmitir. Por otra parte, retransmitir rápido no funciona bien en una conexión de satélite ya que el tráfico innecesario consume ancho de banda de red y reduce la velocidad de transferencia.

TCP se enfrenta a un desafío más difícil que diferenciar entre los destinos local y remoto: las ráfagas de datagramas pueden provocar una congestión, lo cual ocasiona que los retrasos en la transmisión a lo largo de una ruta cambien rápidamente. De hecho, el tiempo total requerido para enviar un mensaje y recibir un reconocimiento puede aumentar o disminuir considerablemente en unos cuantos milisegundos. Para resumir:

El retraso requerido para que los datos lleguen a un destino y se devuelva un reconocimiento depende del tráfico de Internet, así como de la distancia hasta el destino. Como TCP permite que varios programas de aplicación se comuniquen con varios destinos en forma concurrente y las condiciones del tráfico afectan el retraso, TCP debe manejar una variedad de retrasos que pueden cambiar con rapidez.

25.9 Retransmisión adaptativa

Antes de que se inventara TCP, los protocolos de transporte usaban un valor fijo para el retraso de la retransmisión. El diseñador del protocolo o el administrador de red elegían un valor que fuera lo bastante grande para el retraso esperado. Los diseñadores que trabajaron en TCP se dieron cuenta de que un tiempo de espera fijo no funcionaría bien para Internet. Por lo tanto decidieron hacer que la retransmisión de TCP fuera *adaptativa*. Es decir, TCP monitorea el retraso actual en cada conexión y adapta (cambia) el temporizador de retransmisión para tener en cuenta las condiciones cambiantes.

¿Cómo puede TCP monitorear los retrasos de Internet? De hecho, TCP no puede conocer los detalles de todas las partes de Internet en todo momento. En su lugar, TCP estima un *retraso de ida y vuelta* para cada conexión activa, midiendo el tiempo necesario para recibir una respuesta. Cada vez que envía un mensaje del que espera una respuesta, TCP registra la hora en que se envió el mensaje. Cuando llega una respuesta, TCP resta la hora de envío de la hora actual para producir una nueva estimación del retraso de ida y vuelta para esa conexión. A medida que envía paquetes y recibe reconocimientos, TCP genera una secuencia de estimaciones de ida y vuelta, y usa una función estadística para producir un promedio ponderado. Además de un promedio ponderado, TCP mantiene una estimación

de la varianza y usa una combinación lineal de la media y la varianza estimadas al calcular el tiempo en que se necesita la retransmisión.

La experiencia ha demostrado que la retransmisión adaptativa de TCP funciona bien. Al usar la varianza, TCP puede reaccionar con rapidez cuando el retraso aumenta después de una ráfaga de paquetes. Usar un promedio ponderado ayuda a TCP a reiniciar el temporizador de retransmisión si el retraso regresa a un valor inferior después de una ráfaga temporal. Cuando el retraso permanece constante, TCP ajusta el tiempo de espera de la retransmisión a un valor ligeramente más extenso que el retraso promedio de ida y vuelta. Cuando los retrasos comienzan a variar, TCP ajusta el tiempo de espera de la retransmisión a un valor mayor que el promedio para tomar en cuenta los picos.

25.10 Comparación de los tiempos de retransmisión

Para comprender cómo la retransmisión adaptativa ayuda a TCP a maximizar la velocidad de transferencia en cada conexión, considere un caso de pérdida de paquetes en dos conexiones que tienen distintos retrasos de ida y vuelta. Por ejemplo, la figura 25.6 muestra el tráfico en dichas conexiones.

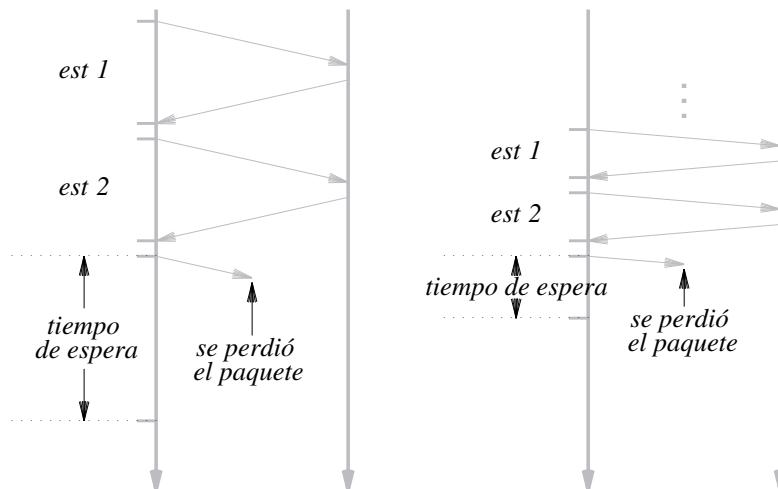


Figura 25.6 Tiempo de espera y retransmisión en dos conexiones TCP que tienen distintos retrasos de ida y vuelta.

Como se muestra en la figura, TCP establece el tiempo de espera de retransmisión de tal modo que sea un poco más largo que el retraso promedio de ida y vuelta. Si el retraso es grande, TCP usa un tiempo de espera de retransmisión grande; si el retraso es pequeño, TCP usa un tiempo de espera pequeño. El objetivo es esperar lo suficiente como para determinar que se perdió un paquete sin tener que esperar más de lo necesario.

25.11 Búferes, control de flujo y ventanas

TCP usa un mecanismo de *ventana* para controlar el flujo de datos. A diferencia del esquema simple de ventanas basado en paquetes que describimos antes, una ventana TCP se mide en bytes. Cuando se establece una conexión, cada extremo de ésta asigna un búfer para contener los datos entrantes y envía el tamaño del búfer al otro extremo. Al llegar los datos, el TCP receptor envía reconocimientos, los cuales especifican el tamaño del búfer restante. TCP usa el término *ventana* para referirse a la cantidad de espacio disponible en búfer en un momento dado. Una notificación que especifica el tamaño de la ventana se conoce como *anuncio de ventana*. Un receptor envía un anuncio de ventana con cada reconocimiento.

Si la aplicación receptora puede leer los datos tan pronto como van llegando, un receptor enviará un anuncio de ventana con valor positivo junto con cada reconocimiento. Pero si el lado emisor opera más rápido que el lado receptor (por ejemplo, si la CPU es más veloz), los datos entrantes llenarán en un momento dado el búfer del receptor y provocarán que éste anuncie una *ventana cero*. Un emisor que recibe un anuncio de ventana cero debe dejar de enviar hasta que el receptor anuncie de nuevo una ventana positiva. La figura 25.7 muestra los anuncios de ventana.

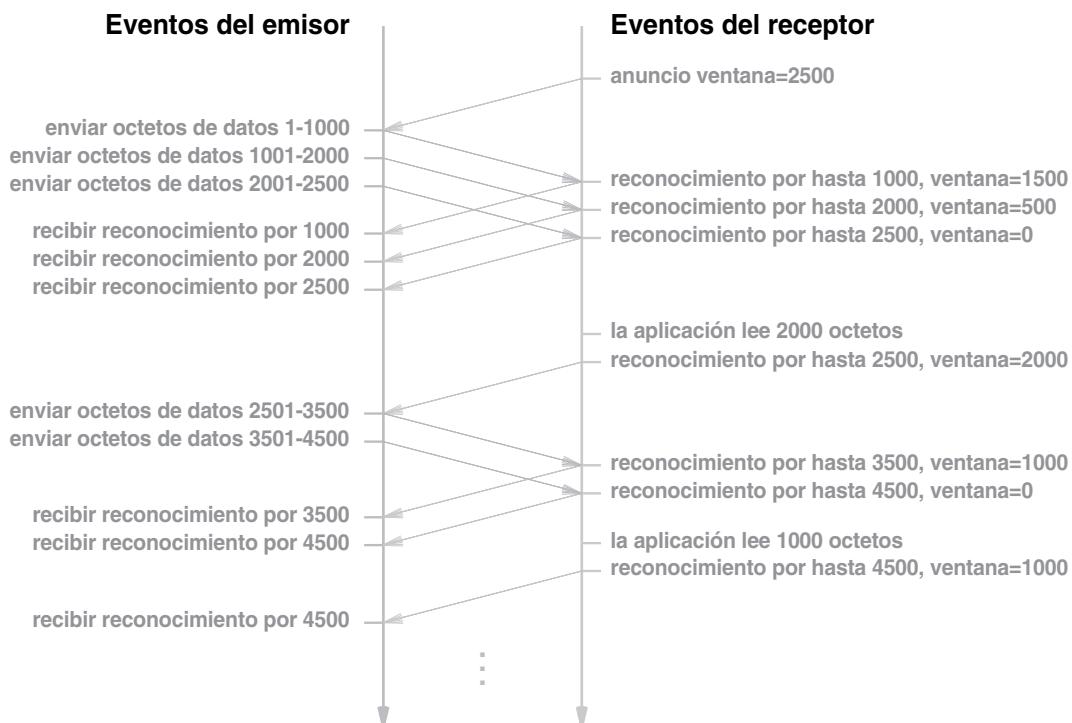


Figura 25.7 Una secuencia de mensajes que ilustra los anuncios de ventana de TCP para un tamaño de segmento máximo de 1000 bytes.

En la figura, el emisor usa un tamaño de segmento máximo de 1000 bytes. La transferencia comienza cuando el receptor anuncia un tamaño de ventana inicial de 2500 bytes. El emisor transmite de inmediato tres segmentos: dos que contienen 1000 bytes de datos y uno que contiene 500 bytes. A medida que llegan los segmentos, el receptor genera un reconocimiento con el tamaño de ventana reducido por la cantidad de datos que llegaron.

Los primeros tres segmentos del ejemplo llenan el búfer del receptor con más rapidez de la que la aplicación puede consumir datos. Por lo tanto, el tamaño de ventana anunciado llega a cero y el emisor no puede transmitir datos adicionales. Una vez que la aplicación receptora consume 2000 bytes de datos, el TCP receptor envía un reconocimiento adicional que anuncia un tamaño de ventana de 2000 bytes. El tamaño de ventana siempre se mide a partir de los datos que se están reconociendo, por lo que el receptor anuncia que puede aceptar 2000 bytes adicionales a los 2500 que ya recibió. El emisor responde transmitiendo dos segmentos adicionales. A medida que llega cada segmento, el receptor envía un reconocimiento con el tamaño de ventana reducido en 1000 bytes (es decir, la cantidad de datos que llegaron).

Una vez más el tamaño de la ventana llega a cero, provocando que el emisor detenga la transmisión. Finalmente la aplicación receptora consume algunos de los datos y el TCP receptor transmite un reconocimiento con un tamaño de ventana positivo. Si el emisor tiene más datos esperando a ser enviados, puede proceder a transmitir otro segmento.

25.12 La negociación en tres pasos de TCP

Para garantizar que se establezcan las conexiones o que terminen de manera confiable, TCP usa una *negociación de 3 pasos* en la que se intercambian tres mensajes. Durante la negociación de 3 pasos que se usa para iniciar una conexión, cada lado envía un mensaje de control que especifica un tamaño inicial del búfer (para el control de flujo) y un número de secuencia. Los expertos han demostrado que el intercambio de 3 pasos de TCP es necesario y suficiente para asegurar un acuerdo sin ambigüedad a pesar de los eventos de pérdida de paquetes, duplicación, retraso y repetición.[†] Además, la negociación asegura que TCP no abra o cierre una conexión sino hasta que ambos extremos estén de acuerdo.

Para entender la negociación de 3 pasos, imagine dos aplicaciones que desean comunicarse en un entorno en el que los paquetes pueden perderse, duplicarse y retrasarse. Ambos lados necesitan estar de acuerdo en iniciar una conversación y ambos lados necesitan saber que el otro lado aceptó. Si el lado A envía una solicitud y el lado B responde (una negociación de 2 pasos), entonces A sabrá que B respondió pero B no sabrá que A recibió la respuesta. Podemos pensar que cada transmisión necesita un reconocimiento. La respuesta de B reconoce la solicitud de A. Sin embargo, A debe también aceptar la respuesta de B (es decir, se intercambian tres mensajes).

TCP usa el término *segmento de sincronización* o *segmento SYN* para describir los mensajes de control que se usan en una negociación de 3 pasos al crear una conexión, y el término *segmento final* o *segmento FIN* para describir los mensajes de control que se usan en una negociación de 3 pasos al cerrar una conexión. La figura 25.8 ilustra la negociación de 3 pasos que se usa para crear una conexión.

[†] Al igual que otros paquetes TCP, los mensajes que se usan para una negociación de 3 pasos pueden retransmitirse.

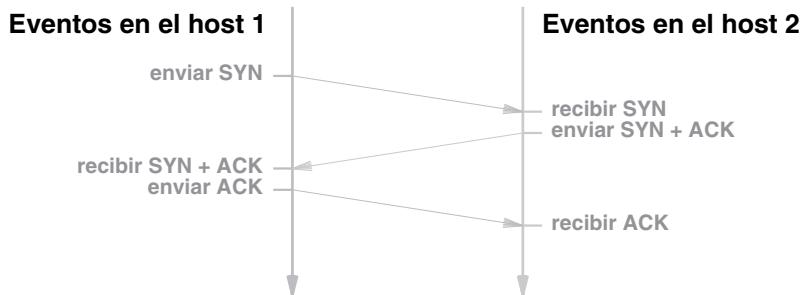


Figura 25.8 La negociación de 3 pasos que se usa para crear una conexión TCP.

En cualquier momento durante el intercambio, uno de los lados podría fallar y reiniciarse, o podría llegar un paquete retrasado de un intercambio anterior. Los desarrolladores de computadoras resolvieron los detalles de todos los posibles problemas y agregaron reglas para asegurar que TCP establezca correctamente las conexiones en todos los casos. Por ejemplo, un aspecto clave de la negociación de 3 vías que se usa para crear una conexión implica la selección de números de secuencia. TCP requiere que cada extremo genere un número de secuencia aleatorio de 32 bits que se convierta en la secuencia inicial para los datos enviados. Si una aplicación intenta establecer una nueva conexión después de que una computadora se reinicie, TCP elige un nuevo número aleatorio. Como la probabilidad de seleccionar un valor aleatorio que coincida con la secuencia utilizada en una conexión anterior es baja, TCP evita los problemas de la repetición. Es decir, si un par de programas de aplicación usan TCP para comunicarse, luego cierran la conexión y establecen una nueva conexión, los números de secuencia de la nueva conexión diferirán de los números de secuencia utilizados en la conexión anterior, con lo cual TCP podrá rechazar los paquetes que lleguen retrasados.

La negociación de 3 pasos que se usa para cerrar una conexión utiliza segmentos *FIN*. Se envía un reconocimiento en cada dirección junto con un segmento FIN para garantizar que todos los datos hayan llegado antes de que se termine la conexión. La figura 25.9 ilustra el intercambio.

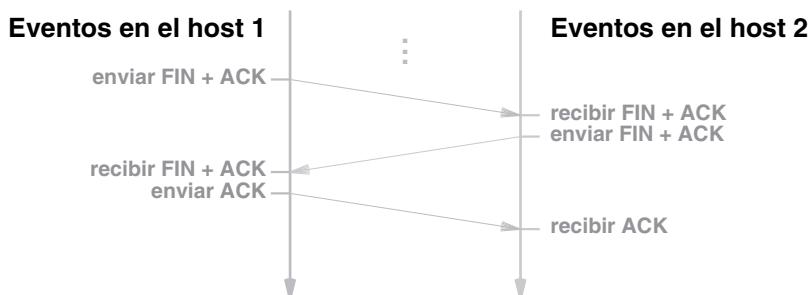


Figura 25.9 La negociación de 3 pasos utilizada para cerrar una conexión.

25.13 Control de congestión de TCP

Uno de los aspectos más interesantes de TCP es un mecanismo para el *control de congestión*. Recuerde que en Internet es más probable que el retraso o la pérdida de paquetes se produzcan debido a la congestión que a una falla de hardware, y que la retransmisión puede profundizar el problema de la congestión al inyectar copias adicionales de un paquete. Para evitar el colapso por congestión, TCP usa cambios en el retraso como una medida contra la congestión y responde reduciendo la velocidad a la que retransmite los datos.

Aunque pensamos en reducir la velocidad de la transmisión, TCP no calcula una velocidad de datos sino que basa la transmisión en el tamaño del búfer. Es decir, el receptor anuncia un tamaño de ventana y el emisor puede transmitir los datos para llenar la ventana del receptor antes de recibir una señal ACK. Para controlar la velocidad de datos, TCP impone una restricción sobre el tamaño de la ventana; al reducir temporalmente este tamaño, el TCP emisor reduce de manera efectiva la velocidad de los datos. El concepto importante es:

Cuando ocurre una congestión, un protocolo de transporte debe reducir la velocidad de la transmisión. Puesto que usa una ventana de tamaño variable, TCP puede lograr una reducción en la velocidad de datos al reducir de manera temporal el tamaño de la ventana. En el caso extremo en el que ocurra una pérdida, TCP reducirá de manera temporal la ventana a la mitad de su valor actual.

TCP usa un mecanismo de control de congestión especial al iniciar una nueva conexión o cuando se pierde un mensaje. En vez de transmitir suficientes datos para llenar el búfer del receptor (es decir, el tamaño de la ventana del receptor), TCP comienza enviando un solo mensaje que contiene datos. Si llega un reconocimiento sin pérdida adicional, TCP duplica la cantidad de datos enviados y envía dos mensajes adicionales. Si llegan ambos reconocimientos, TCP envía cuatro mensajes y así en lo sucesivo. El incremento exponencial continúa hasta que TCP envía la mitad de la ventana anunciada del receptor. Al llegar a la mitad del tamaño de la ventana original, TCP reduce la velocidad de incremento y aumenta el tamaño de la ventana en forma lineal, siempre y cuando no ocurra una congestión. La metodología se conoce como *arranque lento*.

A pesar del nombre, el arranque de TCP no es realmente lento. El incremento exponencial significa que en unos cuantos intercambios de paquetes la transmisión TCP aumenta su velocidad con rapidez. En la red Internet actual, los tiempos de ida y vuelta son cortos (a menudo menores a 100 milisegundos), lo que significa que en menos de un segundo después de iniciar una conexión TCP, la velocidad de transferencia se aproxima al valor máximo que la red y los hosts en comunicación pueden manejar. Sin embargo, el mecanismo de arranque lento reacciona bien en casos en los que Internet se congestiona gravemente al evitar enviar paquetes que empeoren una situación de congestionamiento.

Una vez que hay una conexión TCP en funcionamiento, los mecanismos de control de congestiones responden bien a la congestión inminente. Al retroceder con rapidez, TCP puede aliviar la congestión. En esencia, TCP evita agregar retransmisiones cuando Internet se congestionó. Lo que es más importante: si todas las implementaciones de TCP siguen el estándar, el esquema de control de congestiones implica que todos los emisores retroceden cuando ocurre una congestión.

La idea importante es que TCP maneja mucho más que una sola conexión: el protocolo está diseñado de modo que, si todas las implementaciones de TCP siguen las reglas, actuarán en concierto para evitar un colapso por congestión global.

25.14 Versiones del control de congestión de TCP

Durante muchos años se hicieron pequeños cambios al algoritmo de control de congestiones de TCP, en especial en la década de 1990. Por tradición, cada versión mayor se nombra en honor a una ciudad de Nevada. Una de las primeras versiones mayores, conocida como *Tahoe*, funcionaba como se describe antes. En 1990, una versión conocida como *Reno* introdujo la *recuperación rápida* (también llamada *retransmisión rápida*) para mejorar la velocidad de transferencia cuando la pérdida es ocasional. La siguiente versión de investigación se denominó *Vegas*. Una versión conocida como *NewReno* afinó la heurística e hizo mejoras adicionales. Los distribuidores de sistemas operativos que incluyen protocolos TCP/IP con sus productos tienden a esperar que se validen nuevos algoritmos antes de adoptar un cambio. Pero la mayoría de los sistemas operativos ahora ejecutan NewReno, que se encarga de la transmisión sobre redes típicas y la evasión de congestión de manera efectiva.

25.15 Otras variaciones: SACK y ECN

Como hemos visto, TCP mide el retraso de ida y vuelta, y usa la varianza como una indicación de congestión. Es decir, TCP trata a la red subyacente como una caja negra y usa medidas externas para deducir que ocurrió una congestión. De manera similar, cuando ocurre una pérdida, un TCP emisor deduce que se perdió un paquete.

Los investigadores han planteado la pregunta: ¿podríamos hacer que TCP funcionara mejor si la red ofreciera información más precisa? Para responder la pregunta, los investigadores inventaron dos técnicas: el *reconocimiento selectivo* (SACK) y la *notificación de congestión explícita* (ECN).

El mecanismo SACK cambia el esquema de reconocimiento y permite que un receptor especifique con exactitud las piezas de datos faltantes. Un emisor puede retransmitir únicamente las piezas faltantes y evitar retransmitir los datos que ya llegaron. SACK no ha dado tanto fruto como esperaban los investigadores, ya que la mayoría de las instancias de pérdida no involucran un conjunto aleatorio de paquetes. El esquema TCP original, conocido como *reconocimiento acumulativo*, funciona bien cuando se pierde un bloque grande y contiguo de paquetes.

El esquema ECN se propuso como una forma más precisa de manejar la congestión. Bajo el ECN, los enrutadores que se encuentran entre el origen y el destino monitorean la congestión y marcan cada segmento TCP que pasa a través de una red congestionada. Cuando un paquete llega a su destino, el receptor sabe si la ruta está congestionada. Cuando el receptor devuelve una señal ACK, el receptor indica al emisor si el paquete reconocido experimentó congestión. Una de las desventajas del método de ECN se debe al retraso: un emisor debe esperar a que una señal ACK regrese antes de que el emisor sepa sobre la congestión (y ésta podría desaparecer durante el retraso). ECN no demostró ser tan útil como se esperaba y no se adoptó tan ampliamente en Internet.

25.16 Formato del segmento de TCP

TCP usa un solo formato para todos los mensajes, incluyendo los que transportan datos, los que transportan reconocimientos y los mensajes que forman parte de la negociación de 3 pasos que se usa para crear o terminar una conexión (SYN y FIN). TCP usa el término *segmento* para referirse a un mensaje. La figura 25.10 ilustra el formato del segmento de TCP.

Para entender el formato del segmento, es necesario recordar que una conexión TCP contiene dos flujos de datos, uno fluyendo en cada dirección. Si las aplicaciones en cada extremo están enviando datos al mismo tiempo, TCP puede enviar un solo segmento que transporta los datos salientes, el reconocimiento para los datos entrantes y un anuncio de ventana que especifica la cantidad de espacio de búfer adicional disponible para los datos entrantes. De esta forma, algunos de los campos en el segmento hacen referencia a los flujos de datos que viajan en dirección hacia adelante, mientras que otros campos hacen referencia al flujo de datos que viaja en dirección inversa.

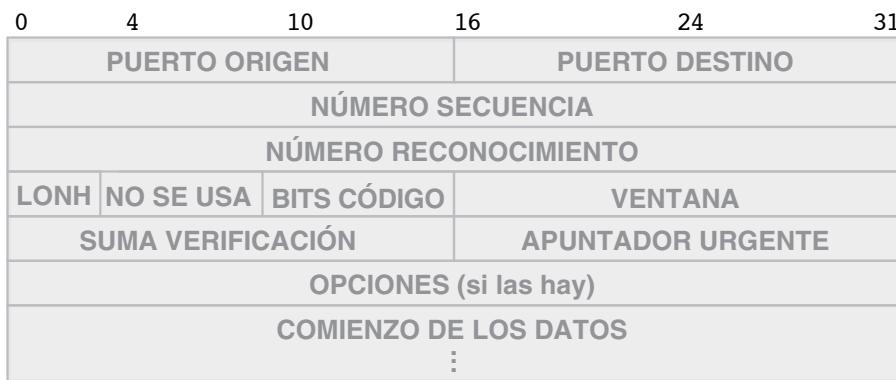


Figura 25.10 El formato del segmento de TCP que se usa para mensajes de datos y de control.

Cuando una computadora envía un segmento, los campos *NÚMERO RECONOCIMIENTO* y *VENTANA* se refieren a los datos entrantes. El *NÚMERO RECONOCIMIENTO* especifica el número de secuencia de los datos que se esperan a continuación, y la *VENTANA* especifica cuánto espacio en búfer adicional está disponible más allá de los datos reconocidos. El reconocimiento siempre se refiere a la primera posición para la que faltan los datos. Si llegan segmentos desordenados, un TCP receptor genera el mismo reconocimiento varias veces hasta que llegan los datos faltantes. El campo *NÚMERO SECUENCIA* se refiere a los datos salientes; proporciona el número de secuencia del primer byte de datos que se transporta en el segmento. Un receptor usa el número de secuencia para reordenar los segmentos que lleguen desordenados y calcula un número de reconocimiento. El campo *PUERTO DESTINO* identifica el programa de aplicación en la computadora receptora que debe recibir los datos, mientras

que el campo *PUERTO ORIGEN* identifica el programa de aplicación que envía los datos. Por último, el campo *SUMA VERIFICACIÓN* contiene una suma de verificación que cubre el encabezado del segmento de TCP y los datos.

Las ideas clave relacionadas con la numeración de secuencias y reconocimientos son:

El campo NUMERO SECUENCIA en un segmento de TCP proporciona el número de secuencia del primer byte de datos transportados en el segmento en dirección hacia adelante; un NÚMERO RECONOCIMIENTO proporciona el primer número de secuencia para el que faltan los datos en dirección inversa.

25.17 Resumen

El protocolo de control de transmisión (TCP) es el principal protocolo de transporte en la suite de protocolos TCP/IP. TCP ofrece a los programas de aplicación un servicio de transporte de flujo de dos vías (dúplex) confiable y totalmente controlado. Después de solicitar a TCP que establezca la conexión, un programa de aplicación puede usar esa conexión para enviar o recibir datos. TCP garantiza la entrega de los datos en orden y sin duplicados. Por último, cuando las dos aplicaciones terminan de usar una conexión, solicitan su terminación.

El protocolo TCP en una computadora se comunica con el protocolo TCP en otra computadora mediante el intercambio de mensajes. Todos los mensajes TCP que se envían de una computadora a otra usan el formato de segmento de TCP, incluyendo los mensajes que transportan datos, los reconocimientos y los anuncios de ventana, así como los mensajes que se usan para establecer y terminar una conexión. Cada segmento TCP viaja en un datagrama IP.

En general, los protocolos de transporte usan una variedad de mecanismos para asegurar el servicio confiable. TCP tiene una combinación especialmente compleja de técnicas que han demostrado ser en extremo exitosas. Además de una suma de verificación en cada segmento, TCP retransmite cualquier mensaje perdido. Para funcionar en Internet, donde los retrasos varían según el tiempo, el tiempo de espera de retransmisión de TCP es adaptativo. TCP mide el retraso actual de ida y vuelta en forma independiente para cada conexión y usa un promedio ponderado de los tiempos de ida y vuelta para seleccionar un tiempo de espera para la retransmisión.

EJERCICIOS

- 25.1** Suponga que los mensajes que se envían entre dos programas pueden perderse, duplicarse, retrasarse o entregarse desordenados. Diseñe un protocolo que permita de manera confiable que los dos programas acepten comunicarse. Entregue a alguien su diseño y vea si pueden encontrar una secuencia de pérdida, duplicación y retraso que haga que el protocolo falle.
- 25.2** Mencione las características de TCP en una lista.

- 25.3** ¿Qué capas de una pila de protocolos se usan en un enrutador? ¿Y en un host?
- 25.4** ¿Cuáles son los principales problemas que debe resolver un protocolo de transporte para lograr una transferencia confiable?
- 25.5** ¿Cuáles son las técnicas que utiliza un protocolo de transporte?
- 25.6** Al usar una ventana corrediza de tamaño N , ¿cuántos paquetes pueden enviarse sin que haya que recibir una señal ACK?
- 25.7** ¿Por qué un protocolo de parada y arranque tiene una velocidad de transferencia bastante baja sobre un canal de satélite GEO que opera a dos megabit por segundo?
- 25.8** Extienda los diagramas en la figura 25.3 para mostrar la interacción que ocurre cuando se envían dieciséis paquetes sucesivos.
- 25.9** ¿Cuál es la principal causa del retraso y pérdida de paquetes en Internet?
- 25.10** ¿Cómo maneja TCP la pérdida de paquetes?
- 25.11** ¿Qué ocurre al rendimiento si un protocolo espera demasiado tiempo para retransmitir? ¿Y si un protocolo no espera lo suficiente para retransmitir?
- 25.12** ¿Cómo calcula TCP un tiempo de espera para la retransmisión?
- 25.13** ¿Qué controla el tamaño de ventana de TCP?
- 25.14** ¿Qué es una señal SYN? ¿Una señal FIN?
- 25.15** Suponga que dos programas usan TCP para establecer una conexión, comunicarse, terminar la conexión y luego abrir una nueva. Suponga además que un mensaje FIN que se envía para cerrar la primera conexión se duplica y se retrasa hasta que se haya establecido la segunda conexión. Si se entrega una copia de la señal FIN anterior, ¿terminará TCP la nueva conexión? ¿Por qué sí o por qué no?
- 25.16** ¿Qué problema en una red hace que TCP reduzca el tamaño de su ventana en forma temporal?
- 25.17** Escriba un programa de computadora para extraer e imprimir los campos en un encabezado de segmento de TCP.
- 25.18** ¿Es necesaria la suma de verificación de TCP, o puede TCP depender de la suma de verificación de IP para asegurar la integridad? Explique.

Contenido del capítulo

- 26.1 Introducción, 447
- 26.2 Comparación entre enrutamiento estático y dinámico, 447
- 26.3 Enrutamiento estático en hosts y una ruta predeterminada, 448
- 26.4 Enrutamiento dinámico y enrutadores, 449
- 26.5 Enrutamiento en la red Internet global, 450
- 26.6 Concepto de sistema autónomo, 451
- 26.7 Los dos tipos de protocolos de enrutamiento de Internet, 451
- 26.8 Rutas y tráfico de datos, 454
- 26.9 El protocolo de puerta de enlace límite (BGP), 454
- 26.10 El protocolo de información de enrutamiento (RIP), 456
- 26.11 Formato de paquetes de RIP, 457
- 26.12 El protocolo de la ruta más corta primero (OSPF), 458
- 26.13 Ejemplo de un gráfico de OSPF, 459
- 26.14 Áreas del OSPF, 459
- 26.15 Sistema intermedio a sistema intermedio (IS-IS), 460
- 26.16 Enrutamiento por multidifusión, 461
- 26.17 Resumen, 465

26

Enrutamiento de Internet y protocolos de enrutamiento

26.1 Introducción

Los capítulos anteriores del libro describieron el concepto fundamental del reenvío de datagramas y explicaron cómo es que IP usa una tabla de reenvío para seleccionar el siguiente salto de cada datagrama. En este capítulo exploraremos un aspecto importante de la tecnología de interconexión de redes: la propagación de la información de enrutamiento que se utiliza para crear y actualizar las tablas de reenvío. El capítulo explica cómo se crean las tablas de reenvío y cómo el software de enrutamiento las actualiza según sea necesario.

El capítulo se enfoca en la propagación de la información de enrutamiento en Internet. Describe varios protocolos de actualización de enrutamiento que se utilizan y explica la distinción entre los protocolos de enrutamiento interiores y los exteriores.

26.2 Comparación entre enrutamiento estático y dinámico

Podemos particionar el enrutamiento IP en dos amplias categorías:

- Enrutamiento estático
- Enrutamiento dinámico

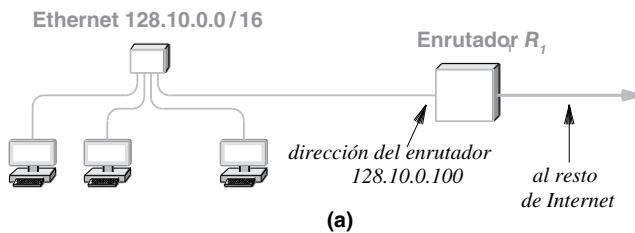
El término *enrutamiento estático* caracteriza a una metodología que crea una tabla de reenvíos cuando el sistema inicia y no cambia las entradas a menos que un administrador las altere manualmente. Por el

contrario, el término *enrutamiento dinámico* caracteriza a una metodología en la que un *software de propagación de rutas* se ejecuta en el sistema y actualiza continuamente la tabla de reenvíos para asegurar que cada datagrama siga una ruta óptima. Es decir, el software se comunica con otros sistemas para aprender rutas óptimas hacia cada destino, verificando continuamente las fallas de la red que provocan cambios en las rutas. Lo irónico es que el enrutamiento dinámico se comporta exactamente como el enrutamiento estático al cargar un conjunto inicial de rutas en una tabla de reenvíos cuando se inicia el sistema.

26.3 Enrutamiento estático en hosts y una ruta predeterminada

El enrutamiento estático es simple, fácil de especificar y no requiere software de enrutamiento adicional. No envía tráfico adicional en las redes, y no se requieren ciclos de CPU para propagar la información de enrutamiento. Sin embargo, el enrutamiento estático es relativamente inflexible, ya que no puede adaptarse a las fallas de la red ni a los cambios en la topología.

¿En dónde se utiliza el enrutamiento estático? La mayoría de los hosts usan enrutamiento estático, en especial en casos en donde el host tiene una conexión de red y un solo enrutador conecta la red al resto de Internet. Por ejemplo, considere la arquitectura que ilustra la figura 26.1: cuatro hosts usan IPv4 se conectan a Ethernet, que se conecta al resto de Internet a través del enrutador R_1 .



Red	Máscara	Siguiente salto
128.10.0.0	255.255.0.0	directo
predeterminada	0.0.0.0	128.10.0.100

(b)

Figura 26.1 (a) Una conexión típica a Internet y (b) la tabla de reenvío estática utilizada por IPv4 en cada host.

Como se indica en la figura, es suficiente con una tabla de reenvío estática con dos entradas para un host común. Una entrada especifica la dirección de la red conectada directamente y la otra entrada especifica que el enrutador R_1 proporciona una *ruta predeterminada* para el resto de los destinos. Cuando una aplicación genera un datagrama para una computadora de la red local (por ejemplo, una impresora local), la primera entrada en la tabla de reenvío indica a IP que debe entregar el datagrama directamente

a su destino. Cuando un datagrama está dirigido a cualquier otro destino en Internet, la segunda entrada en la tabla indica a IP que debe enviar el datagrama al enrutador R_j .

En conclusión:

La mayoría de los hosts de Internet usan enrutamiento estático. La tabla de reenvío del host contiene dos entradas: una para la red a la que se conecta el host y una entrada predeterminada que dirige el resto del tráfico hacia un enrutador específico.

26.4 Enrutamiento dinámico y enrutadores

¿Puede un enrutador de Internet usar enrutamiento estático de la misma forma que un host? Aunque existen casos donde un enrutador usa enrutamiento estático, la mayoría de los enrutadores usan enrutamiento dinámico. Para entender un caso excepcional en donde el enrutamiento estático basta para un enrutador, vea de nuevo la figura 26.1. Podemos imaginar que la figura corresponde a una pequeña organización que es cliente de un ISP. Todo el tráfico que sale del sitio del cliente a través del enrutador R_1 debe viajar al ISP (es decir, a través de una conexión DSL). Como los enrutadores nunca cambian, la tabla de reenvío en el enrutador R_1 puede ser estática. Además, la tabla de reenvío en R_1 puede usar una ruta predeterminada, de la misma forma en que lo hace la tabla de reenvío en un host.

A pesar de unas cuantas excepciones, el enrutamiento estático y las rutas predeterminadas no bastan para la mayoría de los enrutadores; el uso se limita a configuraciones especiales como la anterior. Cuando se interconectan dos ISP, ambos necesitan intercambiar la información de enrutamiento en forma dinámica. Para entender esto, considere tres redes interconectadas por dos enrutadores como se ilustra en la figura 26.2.

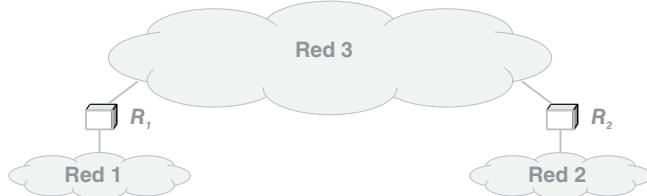


Figura 26.2 Ilustración de una arquitectura que requiere enrutamiento dinámico.

Cada enrutador conoce las redes conectadas directamente. Por lo tanto, el enrutador R_1 conoce las redes 1 y 3, y R_2 las redes 2 y 3. Sin embargo, el enrutador R_1 no conoce la red 2 y R_2 no conoce la red 1, ya que no hay una conexión directa. Para el ejemplo trivial, tal vez parezca que el enrutamiento estático basta. Sin embargo, el método estático no es suficiente para manejar miles de redes. En especial, cada vez que un ISP agrega la red de un nuevo cliente, la información debe pasar por Internet. Lo que es más importante, un proceso manual es demasiado lento como para adaptarse a las fallas de red y la congestión en Internet. En consecuencia, para asegurar que todos los enrutadores obtengan información sobre

cómo llegar a cada posible destino, cada enrutador ejecuta software que usa un protocolo de propagación de rutas para intercambiar información con otros enrutadores. Cuando aprende sobre los cambios en las rutas, el software de enrutamiento actualiza la tabla de reenvío local. Además, como los enrutadores intercambian información en forma periódica, la tabla de reenvío local se actualiza continuamente.

Por ejemplo, en la figura 26.2 los enrutadores R_1 y R_2 intercambiarán información de enrutamiento a través de la red 3. Como resultado, el software de enrutamiento en R_2 instalará una ruta a la red 1 y el software ejecutándose en R_1 instalará una ruta a la red 2. Si el enrutador R_2 falla, el software de propagación de ruta de R_1 detectará que la red 2 ya no es alcanzable y eliminará la ruta de su tabla de reenvío. Más adelante, cuando R_2 vuelva a estar funcionando, el software de enrutamiento de R_1 determinará que la red 2 es alcanzable de nuevo y reinstalará la ruta hacia ésta.

Para resumir:

Cada enrutador ejecuta software de enrutamiento que aprende los destinos que puede alcanzar e informa a otros enrutadores sobre éstos. El software de enrutamiento usa información entrante para actualizar continuamente la tabla de reenvío local.

26.5 Enrutamiento en la red Internet global

Hasta ahora hemos descrito el enrutamiento para la conectividad más sencilla (es decir, situaciones que involucran sólo unos cuantos enrutadores). Esta sección analiza una cuestión más amplia: el enrutamiento en la red Internet global. La sección considera los principios generales, mientras que secciones posteriores explican los protocolos específicos de propagación de rutas.

Anteriormente dijimos que un protocolo de propagación de rutas permite a un enrutador intercambiar la información de enrutamiento con otro. Sin embargo, dicho esquema no puede escalar hacia toda la red Internet, ya que si cada enrutador en Internet tratara de intercambiar información de enrutamiento con todos los demás enrutadores, el tráfico resultante saturaría el núcleo de Internet. Para limitar el tráfico de enrutamiento, Internet usa una jerarquía de enrutamiento. Los enrutadores y las redes en Internet se dividen en grupos. Todos los enrutadores dentro de un grupo intercambian información de enrutamiento entre sí. Entonces, al menos un enrutador en cada grupo sintetiza la información antes de pasarla a otros grupos.

¿Qué tan grande es un grupo? ¿Qué protocolo usan los enrutadores dentro de un grupo? ¿Cómo se representa la información de enrutamiento? ¿Qué protocolo usan los enrutadores entre grupos? Los diseñadores del sistema de enrutamiento de Internet no indicaron un tamaño exacto ni especificaron una representación de datos o protocolo exacto. Más bien, los diseñadores mantuvieron intencionalmente la arquitectura lo bastante flexible como para manejar una amplia variedad de organizaciones. Por ejemplo, para dar cabida a organizaciones de varios tamaños, los diseñadores evitaron especificar un tamaño mínimo o máximo para un grupo. Para poder usar cualquier protocolo de enrutamiento, los diseñadores optaron por permitir que cada organización eligiera un protocolo de enrutamiento por separado.

26.6 Concepto de sistema autónomo

Para captar el concepto de grupos de enrutadores, usamos el término *sistema autónomo (AS)*. Por intuición, podemos pensar en un sistema autónomo como un conjunto contiguo de redes y enrutadores, todos los cuales están bajo el control de una autoridad administrativa. No hay un significado exacto para *autoridad administrativa*; el término es lo bastante flexible como para dar cabida a muchas posibilidades. Por ejemplo, un sistema autónomo puede corresponder a un ISP, a toda una corporación o a una universidad importante. Como alternativa, una organización extensa con varias sedes puede optar por definir un sistema autónomo independiente para cada sede. En especial, cada ISP es comúnmente un sistema autónomo, aunque es posible que un ISP de gran tamaño se divida a sí mismo en varios sistemas autónomos.

La elección del tamaño del sistema autónomo puede hacerse por cuestiones económicas, técnicas o administrativas. Por ejemplo, considere una corporación multinacional. Tal vez sea menos costoso para la corporación dividirse en varios sistemas autónomos, cada uno de los cuales tiene una conexión con un ISP en un país dado, en vez de actuar como un solo sistema autónomo con una conexión al resto de Internet. Otra razón de tener un tamaño específico puede ser por el protocolo de enrutamiento a usar, ya que cuando se usa en muchos enrutadores, un protocolo puede generar un tráfico de enrutamiento excesivo (es decir, el tráfico de enrutamiento puede aumentar a razón del cuadrado del número de enrutadores).

Para resumir:

Internet se divide en un conjunto de sistemas autónomos. Los enrutadores dentro de un sistema autónomo intercambian información de rutina, que después se sintetiza antes de pasarla a otro grupo.

26.7 Los dos tipos de protocolos de enrutamiento de Internet

Ahora que entendemos el concepto de sistema autónomo, podemos definir con más precisión el enrutamiento de Internet. Todos los protocolos de enrutamiento de Internet caen en una de dos categorías:

- Protocolos de puerta de enlace interior (IGP)
- Protocolos de puerta de enlace exterior (EGP)

Después de definir las dos categorías, examinaremos un conjunto de protocolos de enrutamiento de ejemplo que ilustran cada categoría.

26.7.1 Protocolos de puerta de enlace exterior (EGP)

Los enrutadores dentro de un sistema autónomo usan un *protocolo de puerta de enlace interior (IGP)* para intercambiar información de rutina. Hay varios IGP disponibles; cada sistema autónomo es libre de elegir su propio IGP. Por lo general, un IGP es fácil de instalar y operar, pero puede limitar el tamaño o la complejidad de enrutamiento de un sistema autónomo.

26.7.2 Protocolos de puerta de enlace exterior (EGP)

Un enrutador en un sistema autónomo usa un *protocolo de puerta de enlace exterior (EGP)* para intercambiar información de enrutamiento con un enrutador en otro sistema autónomo. Por lo general, los EGP son más complejos de instalar y operar que los IGP, pero los EGP ofrecen mayor flexibilidad y menor sobrecarga (es decir, menos tráfico). Para ahorrar tráfico, un EGP resume la información de enrutamiento de un sistema autónomo antes de pasársela a otro sistema autónomo. Lo más importante es que un EGP implementa restricciones de políticas que permiten a un sistema determinar con exactitud qué información se libera fuera de la organización.

26.7.3 Ilustración de la forma en que se usan los IGP y los EGP

La figura 26.3 ilustra la jerarquía de enrutamiento de dos niveles que se usa en Internet, mostrando los enrutadores en dos sistemas autónomos.

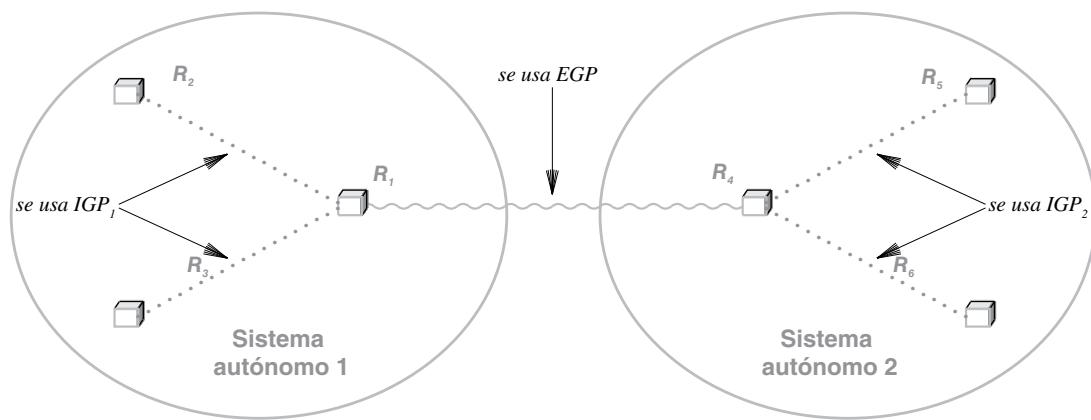


Figura 26.3 Ilustración del enrutamiento de Internet, donde se usa un IGP en cada sistema autónomo y un EGP entre sistemas autónomos.

En la figura, el Sistema autónomo 1 (AS_1) eligió usar el IGP_1 en forma interna y el Sistema autónomo 2 (AS_2) eligió el IGP_2 . Todos los enrutadores en el AS_1 se comunican mediante el IGP_1 y todos los enrutadores en el AS_2 se comunican mediante el IGP_2 . Los enrutadores R_1 y R_4 usan un EGP para comunicarse entre los dos sistemas autónomos. Es decir, R_1 debe resumir la información de su sistema autónomo y enviar el resumen a R_4 . Además, R_1 acepta un resumen de R_4 y usa el IGP_1 para propagar la información a los enrutadores en el AS_1 . R_4 realiza el mismo servicio para el AS_2 .

26.7.4 Rutas óptimas, métrica de enrutamiento y protocolos IGP

Tal vez parezca que en vez de simplemente descubrir una ruta hacia cada destino, el software de enrutamiento debería encontrar todas las rutas posibles y luego elegir una que sea óptima. Aunque por lo general Internet tiene varias rutas entre cualquier origen y destino, no hay un acuerdo universal en cuanto a la ruta óptima. Para entender por qué, considere los requerimientos de diversas aplicaciones. Desde el punto de vista de una aplicación de escritorio remoto, una ruta con el menor retraso es óptima. Para un navegador que descarga un archivo de gráficos grande, una ruta con máxima velocidad de transferencia es óptima. Para una aplicación de webcast de audio que recibe audio en tiempo real, una ruta con la menor inestabilidad es óptima.

Usamos el término *métrica de enrutamiento* para referirnos a una medida que usa el software de enrutamiento al elegir una ruta. Aunque es posible usar la velocidad de transmisión, el retraso o la inestabilidad como métrica de enrutamiento, la mayoría del software de enrutamiento de Internet no lo hace. En su lugar, el enrutamiento común de Internet usa una combinación de dos métricas: el *costo administrativo* y el *conteo de saltos*. En el enrutamiento de Internet, un salto corresponde a una red intermedia (o enrutador intermedio). Por lo tanto, el conteo de saltos para un destino proporciona el número de redes intermedias en la ruta hacia el destino. Los costos administrativos se asignan en forma manual, a menudo para controlar qué rutas puede seguir el tráfico. Por ejemplo, suponga que en una corporación dos rutas conectan el departamento de contabilidad con el de nóminas: una ruta de 2 saltos que incluye una red designada para usarse en el tráfico de clientes y una ruta de 3 saltos que incluye redes para el tráfico corporativo interno. En este ejemplo, la ruta más corta viola la política corporativa de tráfico interno al recorrer una red designada para servir a los clientes. En tales casos, un administrador de redes puede ignorar el costo real de la ruta de 2 saltos asignándole un costo administrativo de 4 saltos (es decir, reemplaza el costo real por un valor administrativo para lograr el efecto deseado). De tal forma, el software de enrutamiento elegirá la ruta con el menor costo (en el ejemplo, la ruta con una métrica de 3 saltos). Por lo tanto, el tráfico interno seguirá la política corporativa. En conclusión:

Aunque la mayoría de los protocolos de enrutamiento de Internet están diseñados para usar una métrica de conteo de saltos, es posible que un administrador de redes anule la métrica para implementar una política en particular.

Los IGP y EGP difieren de una forma importante con respecto a la métrica de enrutamiento: los IGP usan métrica de enrutamiento pero los EGP no. Es decir, cada sistema autónomo selecciona una métrica de enrutamiento y hace que el software de enrutamiento interno envíe la métrica con cada ruta, de modo que el software receptor pueda usar la métrica para elegir rutas óptimas. Sin embargo, fuera de un sistema autónomo, un EGP no intenta elegir una ruta óptima. En vez de ello, el EGP simplemente busca una ruta. La razón es simple: como cada sistema autónomo es libre de elegir una métrica de enrutamiento, un EGP no puede realizar comparaciones significativas. Por ejemplo, suponga que un sistema autónomo informa el número de saltos a lo largo de una ruta hacia el destino *D*, y otro sistema autónomo informa la velocidad de transferencia a lo largo de una ruta diferente hacia *D*. El EGP que recibe los dos informes no puede elegir cuál de las dos rutas tiene menor costo, ya que no hay

forma de convertir saltos a velocidad de transferencia. Por consiguiente, un EGP sólo puede informar la existencia de una ruta y no su costo. Podemos resumir:

Dentro de un sistema autónomo, el software del IGP usa una métrica de enrutamiento para elegir una ruta óptima hacia cada destino. El software de EGP busca una ruta hacia cada destino, pero no puede buscar una ruta óptima debido a que no puede comparar las métricas de enrutamiento de varios sistemas autónomos.

26.8 Rutas y tráfico de datos

Un aforismo en el trabajo con redes sugiere que la respuesta a un anuncio de enrutamiento consiste en el envío de datos, y se leería “Si me envía una ruta, le enviaré datos”. El concepto es simple: el tráfico de datos para un destino dado fluye exactamente en la dirección opuesta del tráfico de enrutamiento. Por ejemplo, suponga que un sistema autónomo perteneciente al ISP_1 contiene la red N . Antes de que pueda llegar tráfico destinado para N , el ISP_1 debe anunciar una ruta a N . Es decir, cuando fluye el anuncio de enrutamiento hacia fuera, los datos comienzan a fluir hacia dentro. La figura 26.4 ilustra el flujo de datos en respuesta a los anuncios de enrutamiento.

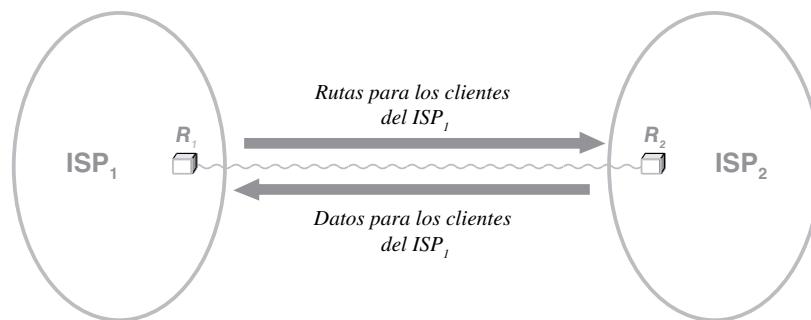


Figura 26.4 El flujo de datos después de que un enrutador en un ISP anuncia rutas.

26.9 El protocolo de puerta de enlace límite (BGP)

Hay un protocolo específico que ha emergido como el protocolo de puerta de enlace exterior más utilizado en Internet. Conocido como el *protocolo de puerta de enlace límite (BGP)*, el protocolo ha sobrevivido tres revisiones importantes. La versión 4 es el estándar actual y se abrevia de manera oficial como *BGP-4*. En la práctica, el número de versión permaneció sin cambios por tanto tiempo que los profesionales de redes usan simplemente *BGP* para referirse a la versión 4.

El BGP tiene las siguientes características:

- *Enrutamiento entre sistemas autónomos.* Como está destinado para usarse como protocolo de puerta de enlace exterior, el BGP ofrece información de enrutamiento al nivel del sistema autónomo. Es decir, todas las rutas se proporcionan como una ruta de sistemas autónomos. Por ejemplo, la ruta hacia un destino dado puede estar conformada por los sistemas autónomos 17, 2, 56 y 12. No se usa una métrica de enrutamiento ni hay forma de que el BGP proporcione los detalles sobre los enrutadores dentro de cada sistema autónomo de la ruta.
- *Provisión para políticas.* El BGP permite al emisor y al receptor cumplir con las políticas. En especial, un administrador puede configurar el BGP para restringir las rutas anunciadas a los sistemas externos.
- *Instalaciones para el enrutamiento de tránsito.* Si acepta pasar el tráfico hacia otros sistemas autónomos, el BGP los clasifica como sistemas de *tránsito*; pero si no los acepta, entonces los clasifica como sistemas *aislados o stubs*. De manera similar, el tráfico que va de paso hacia otro AS se clasifica como tráfico de tránsito. La clasificación permite que el BGP se distinga entre proveedores ISP y otros sistemas autónomos. Lo que es más importante, BGP permite que una corporación se clasifique a sí misma como aislada aun siendo *multiproveedor* (por ejemplo, una corporación con varias conexiones externas puede rehusarse a aceptar tráfico de tránsito).
- *Transporte confiable.* El BGP usa TCP para toda la comunicación. Es decir, un programa BGP de un enrutador que se encuentra en un sistema autónomo forma una conexión TCP hacia un programa similar en otro sistema autónomo, y luego envía datos a través de la conexión. TCP asegura que los datos lleguen completos y en el orden correcto.

BGP proporciona el pegamento que mantiene unido el enrutamiento de Internet. En el centro de Internet, los ISP de nivel 1 usan el BGP para intercambiar información de enrutamiento y aprender sobre los clientes del otro. Para resumir:

El protocolo de puerta de enlace límite (BGP) es el protocolo de puerta de enlace exterior que los ISP de nivel 1 usan para intercambiar información de enrutamiento entre sistemas autónomos en el centro de Internet; la versión actual es BGP-4.

26.10 El protocolo de información de enrutamiento (RIP)

El *protocolo de información de enrutamiento (RIP)* fue de los primeros protocolos de puerta de enlace interior que se usaron en Internet. El RIP tiene las siguientes características:

- *Enrutamiento dentro de un sistema autónomo.* El RIP está diseñado como un protocolo de puerta de enlace interior que se utiliza para pasar información entre enrutadores dentro de un sistema autónomo.
- *Métrica de conteo de saltos.* El RIP mide la distancia en *saltos* de red, en donde cada red entre el origen y el destino cuenta como un solo salto. El RIP cuenta una red conectada en forma directa como un salto.
- *Transporte poco confiable.* El RIP usa el UDP para transferir mensajes entre enrutadores.
- *Entrega por difusión o multidifusión.* El RIP está diseñado para usarse a través de tecnologías de red de área local que soporten difusión o multidifusión (por ejemplo, Ethernet). La versión 1 del RIP usa difusión IPv4; la versión 2 permite la entrega mediante multidifusión.
- *Soporte para CIDR de IPv4 y subredes.* La versión 2 del RIP incluye una máscara de dirección con cada dirección de destino.
- *Soporte para propagación de ruta predeterminada.* Además de especificar destinos explícitos, el RIP permite que un enrutador anuncie una *ruta predeterminada*.
- *Algoritmo de vector de distancias.* El RIP usa la metodología del *vector de distancias* para el enrutamiento, como se define en el algoritmo 18.3.[†]
- *Versión pasiva para hosts.* Aunque sólo un enrutador puede propagar la información de enrutamiento, el RIP permite a un host escuchar en forma pasiva y actualizar su tabla de reenvío. El RIP pasivo es útil en redes donde un host selecciona de entre varios enrutadores.
- *Extensión para IPv6.* Aunque se definió en un principio para IPv4 hay una versión conocida como *RIP de nueva generación (RIPng)* disponible para IPv6.

Para entender cómo es que el RIP propaga las rutas, recuerde cómo funciona el enrutamiento por vector de distancias. Cada mensaje saliente contiene un anuncio con una lista de todas las redes que puede alcanzar el emisor junto con una distancia para cada una. Cuando recibe un anuncio, el software de RIP usa la lista de destinos para actualizar la tabla de reenvío local. Cada entrada en un anuncio de RIP consiste en un par:

(red de destino, distancia)

donde *distancia* es el número de *saltos* hacia el destino. Cuando llega un mensaje, si el receptor no tiene una ruta hacia un destino anunciado o si una distancia anunciada es más corta que la distancia de la ruta actual, el receptor reemplazará su ruta con una ruta hacia el emisor.

[†] Encontrará el algoritmo 18.3 en la página 319.

La principal ventaja del RIP es la simpleza. El RIP requiere poca configuración; un administrador únicamente inicia la ejecución del RIP en cada enrutador de la organización y permite que los enrutadores se difundan mensajes entre sí. Después de un tiempo breve, todos los enrutadores de la organización tendrán rutas hacia todos los destinos.

El RIP también se encarga de la propagación de una ruta predeterminada. La organización sólo tiene que configurar uno de sus enrutadores para tener una ruta predeterminada (por lo general, una organización selecciona un enrutador que se conecte a un ISP). El RIP propaga la ruta predeterminada hacia todos los demás enrutadores en la organización, lo que significa que cualquier datagrama que se envíe a un destino fuera de la organización se reenviará al ISP.

26.11 Formato de paquetes de RIP

El formato de los mensajes de RIP explica cómo funciona un protocolo de enrutamiento por vector de distancias. La figura 26.5 ilustra un mensaje de actualización de RIP que se utiliza con IPv4.

0	8	16	24	31			
COMANDO (1-5)	VERSIÓN (2)	DEBE SER CERO					
FAMILIA DE RED 1		ETIQUETA DE RUTA PARA RED 1					
DIRECCIÓN IP DE RED 1							
MÁSCARA DE DIRECCIÓN PARA RED 1							
SIGUIENTE SALTO PARA RED 1							
DISTANCIA A RED 1							
FAMILIA DE RED 2		ETIQUETA DE RUTA PARA RED 2					
DIRECCIÓN IP DE RED 2							
MÁSCARA DE DIRECCIÓN PARA RED 2							
SIGUIENTE SALTO PARA RED 2							
DISTANCIA A RED 2							
...							

Figura 26.5 El formato de un mensaje de actualización de RIP versión 2 que se usa con IPv4.

Como se indica en la figura, cada entrada contiene la dirección IPv4 de un destino y una distancia hacia ese destino. Además, para poder usar el RIP con CIDR o con el direccionamiento de subredes, una entrada contiene una máscara de dirección de 32 bits. Cada entrada tiene también una dirección del siguiente salto, además de dos campos de 16 bits que identifican a la entrada como una dirección IP y

proporcionan una etiqueta que se utiliza para agrupar las entradas. En total, cada entrada contiene veinte octetos. Podemos resumir:

El RIP es un protocolo de puerta de enlace interior que usa un algoritmo de vector de distancias para propagar la información de enrutamiento.

26.12 El protocolo de la ruta más corta primero (OSPF)

El formato de mensajes de RIP revela una desventaja de los protocolos de vector de distancias: el tamaño de un mensaje es proporcional al número de redes que pueden alcanzarse. El envío de mensajes de RIP presenta un retraso y el procesamiento de mensajes de RIP consume muchos ciclos de CPU. El retraso indica que los cambios en la ruta se propagan con lentitud, un enrutador a la vez. De esta forma, aunque el RIP funciona bien en unos cuantos enrutadores, no se escala bien.

Para satisfacer la demanda de un protocolo de enrutamiento que pueda escalar a organizaciones grandes, el IETF ideó un IGP conocido como el *protocolo de la ruta más corta primero (OSPF)*. El nombre se deriva del uso del algoritmo de SPF de Dijkstra, que calcula las rutas más cortas. El OSPF tiene las siguientes características:

- *Enrutamiento dentro de un sistema autónomo.* El OSPF es un protocolo de puerta de enlace interior que se usa dentro de un sistema autónomo.
- *Soporte para CIDR.* Para adaptarse al direccionamiento CIDR de IPv4, el OSPF incluye una máscara de dirección de 32 bits con cada dirección IPv4.
- *Intercambio autenticado de mensajes.* Un par de enrutadores que usen el OSPF pueden autenticar cada mensaje.
- *Rutas importadas.* El OSPF permite a un enrutador introducir las rutas aprendidas de otros medios (por ejemplo, de BGP).
- *Algoritmo de estado de enlace.* El OSPF usa el *enrutamiento de estado de enlace* como se describe en el capítulo 18.
- *Soporte para la métrica.* El OSPF permite a un administrador asignar un costo a cada ruta.
- *Extensión para IPv6.* La versión 3 del OSPF (*OSPFv3*) puede propagar rutas para destinos IPv6.
- *Soporte para redes multiacceso.* El enrutamiento tradicional de estado de enlace es ineficiente a través de una red multiacceso tal como Ethernet, ya que todos los enrutadores conectados a la red difunden el estado del enlace. El OSPF optimiza el enrutamiento al designar un solo enrutador para difundir en la red.

Para resumir:

El OSPF es un protocolo de puerta de enlace interior que usa un algoritmo de estado de enlace para propagar la información de enrutamiento. Los enruteadores usan el algoritmo de SPF de Dijkstra para calcular las rutas más cortas.

26.13 Ejemplo de un gráfico de OSPF

En el capítulo 18 vimos que el enrutamiento de estado de enlace usa una abstracción teórica en forma de gráfico. Aunque el OSPF permite una relación compleja entre las redes y un gráfico, un ejemplo simple ayudará a explicar el concepto básico.[†] Considere la red y el gráfico asociado que se ilustra en la figura 26.6.

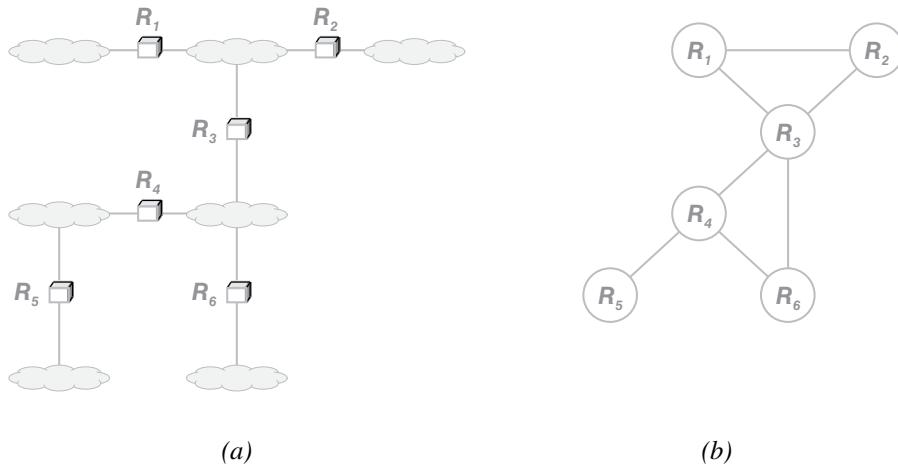


Figura 26.6 (a) Una topología de ejemplo y (b) el gráfico de OSPF correspondiente.

La figura muestra un gráfico de OSPF común en el que cada nodo corresponde a un enrutador. Un borde en el gráfico corresponde a una conexión entre un par de enrutadores (es decir, una red). Para seguir un algoritmo de estado de enlace, cada par de enrutadores conectados por una red se sondan entre sí periódicamente y luego difunden un mensaje de estado de enlace hacia otros enrutadores. Todos los enrutadores reciben el mensaje difundido; cada uno usa el mensaje para actualizar su copia local del gráfico y recalcula las rutas más cortas cuando cambia el estado.

26.14 Áreas del OSPF

La característica específica que hace a OSPF más complejo que los otros protocolos de enrutamiento, también lo hace más poderoso: el enrutamiento jerárquico. Para lograr una jerarquía, el OSPF permite

[†] En la práctica, los gráficos de OSPF son más complejos que los que se muestran.

particionar un sistema autónomo para fines de enrutamiento. Es decir, un administrador puede dividir los enrutadores y las redes de un sistema autónomo en subconjuntos, a los que OSPF llama áreas. Cada enrutador se configura para conocer el límite del área (es decir, exactamente qué otros enrutadores hay en su área). Cuando se ejecuta el OSPF, los enrutadores dentro de un área dada intercambian mensajes de estado de enlace en forma periódica.

Además de intercambiar información dentro de un área, el OSPF permite la comunicación entre áreas. Un enrutador en cada área se configura para comunicarse con un enrutador en una o más áreas adicionales. Los dos enrutadores resumen la información de enrutamiento que han aprendido de otros enrutadores dentro de su respectiva área y luego intercambian el resumen. De esta forma, en vez de difundir a todos los enrutadores del sistema autónomo, el OSPF limita las difusiones del estado de enlace a los enrutadores dentro de un área. Como resultado de la jerarquía, el OSPF puede escalarse para manejar sistemas autónomos mucho más grandes que otros protocolos de enrutamiento.

En conclusión:

Como permite a un administrador particionar los enrutadores y las redes de un sistema autónomo en varias áreas, el OSPF puede escalarse para manejar un número más extenso de enrutadores que otros IGP.

26.15 Sistema intermedio a sistema intermedio (IS-IS)

Diseñado originalmente por Digital Equipment Corporation para formar parte de DECNET V, el IS-IS (*sistema intermedio a sistema intermedio*)[†] es un IGP. El IS-IS se creó aproximadamente al mismo tiempo que el OSPF, y los dos protocolos son similares en muchas formas. Ambos usan la metodología de estado de enlace y emplean el algoritmo de Dijkstra para calcular las rutas más cortas. Además, ambos protocolos requieren dos enrutadores adyacentes para probar periódicamente el enlace entre ellos y difundir un mensaje de estado.

Las principales diferencias entre el OSPF y el IS-IS original pueden resumirse como:

- IS-IS fue en un principio propietario (perteneciente a Digital) y OSPF se creó como un estándar *abierto*, disponible a todos los distribuidores.
- El OSPF se diseñó para ejecutarse sobre IP, mientras que IS-IS se diseñó para ejecutarse sobre CLNS (parte de la desventurada pila de protocolos OSI).
- El OSPF se diseñó para propagar las rutas IPv4 (direcciones y máscaras de dirección de IPv4); IS-IS se diseñó para propagar rutas para los protocolos OSI.
- Con el tiempo, el OSPF ganó muchas características. Como resultado, IS-IS tiene ahora menos sobrecarga.

Cuando se inventaron originalmente los protocolos, la abertura y dedicación hacia IP que tenía OSPF lo hizo más popular que IS-IS. De hecho, IS-IS fue casi olvidado. Con el pasar de los años, la popularidad del OSPF animó al IETF a agregarle características adicionales. Irónicamente, a principios

[†] El nombramiento sigue la terminología de Digital en la que un enrutador se llamaba *sistema intermedio* y un host se llamaba *sistema final*.

de la década de 2000, diez años después de que se diseñaron los protocolos, cambiaron varias cosas para dar a IS-IS una segunda oportunidad. Digital Equipment Corporation se había disuelto y el IS-IS ya no se consideraba propiedad exclusiva. Se definió una versión más nueva de IS-IS para integrarla con IP e Internet. Puesto que el OSPF se creó para IPv4, había que desarrollar una versión totalmente nueva para manejar direcciones IPv6 más grandes. Los ISP más grandes aumentaron a un tamaño en el que la sobrecarga adicional que tiene el OSPF hace que IS-IS resulte más atractivo. Como resultado, IS-IS empezó a regresar.

26.16 Enrutamiento por multidifusión

26.16.1 Semántica de multidifusión IP

Hasta ahora hemos hablado sobre el enrutamiento por unidifusión. Es decir, hemos considerado protocolos de enrutamiento que propagan la información sobre los destinos, cada uno de los cuales tiene una dirección estática y una ubicación que no cambia. Uno de los objetivos de diseño para la propagación de rutas de unidifusión es la *estabilidad*. Los cambios continuos en las rutas no son convenientes ya que producen mayor inestabilidad y los datagramas llegan desordenados. Por consiguiente, una vez que un protocolo de enrutamiento de unidifusión encuentra una ruta más corta, por lo general la retiene hasta que una falla la haga inutilizable.

La propagación de la información de *enrutamiento por multidifusión* difiere de manera considerable de la propagación de rutas por unidifusión. La diferencia surge debido a que la multidifusión por Internet permite la membresía de grupos dinámicos y los emisores anónimos. La membresía de grupos dinámicos significa que una aplicación puede optar por participar en un grupo en cualquier momento y seguir participando durante un tiempo indeterminado. Es decir, la abstracción de multidifusión de IP permite que una aplicación que se ejecuta en cualquier computadora:

- Se une a un grupo de multidifusión en cualquier momento y comience a recibir una copia de todos los paquetes que se envían al grupo. Para unirse a un grupo, un host informa a un enrutador cercano. Si varias aplicaciones en el mismo host deciden unirse a un grupo, el host recibe una copia de cada datagrama que se envía al grupo y hace una copia local para cada aplicación.
- Deje un grupo de multidifusión en cualquier momento. Un host envía periódicamente mensajes de membresía de grupo al enrutador local. Una vez que la última aplicación en el host deja el grupo, el host informa al enrutador local que ya no está participando en el mismo.

Un grupo de multidifusión IP es anónimo de dos formas. Primero, ni un emisor ni un receptor conocen (ni pueden averiguar) la identidad o el número de miembros del grupo. Segundo, los enrutadores y los hosts no saben qué aplicaciones enviarán un datagrama a un grupo, ya que cualquier aplicación puede enviar un datagrama a un grupo de multidifusión en cualquier momento. Es decir, la membresía en un grupo de multidifusión sólo define un conjunto de receptores; un emisor no necesita unirse a un grupo de multidifusión antes de enviar un mensaje al grupo.

Para resumir:

La membresía en un grupo de multidifusión IP es dinámica: una computadora puede unirse a un grupo o dejarlo en cualquier momento. La membresía de grupo define un conjunto de receptores, ya que cualquier aplicación puede enviar un datagrama al grupo, incluso aunque la aplicación no sea un miembro del mismo.

26.16.2 IGMP

¿Cómo se une o deja un host a un grupo de multidifusión? Existe un protocolo estándar que permite a un host informar a un enrutador cercano cada vez que necesita unirse a un grupo de multidifusión específico o dejarlo. Conocido como *protocolo de multidifusión de grupos de Internet (IGMP)*, este protocolo sólo se usa entre un host y un enrutador de la red. Además, el protocolo define el host (y no la aplicación) para que sea miembro de un grupo y no especifica nada sobre las aplicaciones. Si varias aplicaciones en un host dado se unen a un grupo de multidifusión, el host debe hacer copias de cada datagrama que reciba para las aplicaciones locales. Cuando la última aplicación en un host deja un grupo, el host usa IGMP para informar al enrutador local que ya no es miembro del grupo.

26.16.3 Técnicas de reenvío y descubrimiento

Cuando un enrutador se entera que un host en una de sus redes se unió a un grupo de multidifusión, éste debe establecer una ruta hacia el grupo y propagar hacia el host los datagramas que recibe para el grupo. De esta forma, los enrutadores (y no los hosts) tienen la responsabilidad de la propagación de la información de enrutamiento por multidifusión.

La membresía del grupo dinámico y el soporte de emisores anónimos dificulta extremadamente el enrutamiento por multidifusión de propósito general. Además, el tamaño y la topología de los grupos varía de manera considerable entre las aplicaciones. Por ejemplo, es común que en teleconferencias se creen grupos pequeños (por ejemplo, entre dos y cinco miembros) que pueden estar dispersos geográficamente o en la misma organización. Una aplicación de webcast puede crear potencialmente un grupo con millones de miembros esparcidos en todo el mundo.

Para adaptarse a una membresía dinámica, los protocolos de enrutamiento por multidifusión deben ser capaces de cambiar de ruta en forma rápida y continua. Por ejemplo, si un usuario en Francia se une a un grupo de multidifusión con miembros en Estados Unidos y Japón, el software de enrutamiento multidifusión debe primero encontrar a otros miembros del grupo y luego crear una estructura de reenvío óptima. Más importante aún, como cualquier usuario puede enviar un datagrama al grupo, la información sobre las rutas debe extenderse más allá de los miembros del grupo. En la práctica, los protocolos de multidifusión han seguido tres distintas metodologías para el reenvío de datagramas:

- Inundar y podar
- Configuración y uso de túneles
- Descubrimiento basado en dirección central

Inundar y podar. El método de inundar y podar o *flood-and-prune* es ideal en una situación en la que el grupo es pequeño y todos los miembros están conectados a redes de área local contiguas (por ejemplo, un grupo dentro de una corporación). En un principio, los enrutadores reenvían cada datagrama a todas las redes. Es decir, cuando llega un datagrama de multidifusión, un enrutador lo transmite por multidifusión de hardware a todas las redes LAN conectadas directamente. Para evitar los bucles de enrutamiento, los protocolos de inundar y podar usan una técnica conocida como *difusión por ruta inversa (RPB)* que interrumpe los ciclos. Durante la etapa de inundación, los enrutadores intercambian información sobre la membresía de grupo. Si un enrutador descubre que ningún host en cierta red es miembro del grupo, deja de reenviar la multidifusión a la red (es decir, “poda” la red del conjunto).

Configuración y uso de túneles. Este método es ideal en una situación en la que el grupo está disperso geográficamente (es decir, tiene unos cuantos miembros en cada sitio y los sitios están separados por distancias largas). Se configura un enrutador en cada sitio para que sepa sobre los otros sitios. Cuando llega un datagrama por multidifusión, el enrutador en un sitio lo transmite mediante multidifusión por hardware a todas las redes LAN conectadas directamente. Después, el enrutador consulta su tabla de configuración para determinar qué sitios remotos deberían recibir una copia y usa túneles de IP en IP para transferir por multidifusión una copia del datagrama a cada uno de los sitios remotos.

Descubrimiento basado en dirección central. Aunque los métodos “inundar y podar” y “configuración y uso de túneles” pueden manejar bien los casos extremos, se necesita una técnica que permita que la multidifusión escale sin problemas de un grupo pequeño en un área limitada a un grupo grande con miembros en distintas ubicaciones. Para brindar un crecimiento uniforme, algunos protocolos de enrutamiento por multidifusión designan una dirección de unidifusión *central* para cada grupo de multidifusión. Cada vez que un enrutador R_1 recibe un datagrama de multidifusión que debe transmitirse a un grupo, R_1 lo encapsula en un datagrama de unidifusión y lo reenvía a la dirección de unidifusión central del grupo. Como el datagrama de unidifusión viaja a través de Internet, cada enrutador examina el contenido. Cuando el datagrama llega a un enrutador R_2 que participa en el grupo, R_2 elimina y procesa el mensaje de multidifusión. R_2 usa el enrutamiento por multidifusión para reenviar el datagrama a los miembros del grupo. Las solicitudes de unirse al grupo siguen el mismo patrón: si recibe una solicitud para unirse a un grupo, R_2 agrega una nueva ruta a su tabla de reenvíos de multidifusión y comienza a reenviar una copia de cada datagrama de multidifusión a R_1 . Por lo tanto, el conjunto de enrutadores que reciba un grupo de multidifusión específico aumenta desde el centro hacia fuera. En términos de teoría de gráficos, los enrutadores forman un *árbol*.

26.16.4 Protocolos de multidifusión

Aunque se han propuesto muchos protocolos de enrutamiento por multidifusión, no existe en la actualidad un enrutamiento por multidifusión a nivel de Internet. Unos cuantos de los protocolos propuestos son:

Protocolo de enrutamiento de multidifusión por vector de distancias (DVMRP). Como protocolo utilizado por el programa *mrouted* de UNIX y la *red troncal de multidifusión (MBONE)* de Internet, el DVMRP realiza multidifusión local y usa el encapsulamiento de IP en IP para enviar datagramas de multidifusión desde un sitio de Internet a otro.

Árboles basados en ubicación central (CBT). Un protocolo en el que los enrutadores crean un árbol de entrega desde un punto central para cada grupo. CBT depende del enrutamiento por multidifusión para llegar a un punto central.

Multidifusión independiente del protocolo-modo disperso (PIM-SM). Un protocolo que utiliza la misma metodología que CBT para formar un árbol de enrutamiento por multidifusión. Los diseñadores eligieron el término *independiente del protocolo* para enfatizar que, aunque los datagramas de unidifusión se usan para contactar con destinos remotos al establecer el reenvío por multidifusión, PIM-SM no depende de ningún protocolo de enrutamiento de multidifusión específico.

Multidifusión independiente del protocolo-modo denso (PIM-DM). Un protocolo diseñado para usarse dentro de una organización. Los enrutadores que usan la difusión PIM-DM (es decir, inundación) transmiten paquetes por multidifusión a todas las ubicaciones dentro de la organización. Cada enrutador que no sea miembro de un grupo específico envía de vuelta un mensaje para *podar* el árbol de enrutamiento de multidifusión (es decir, una solicitud para detener el flujo de paquetes). El esquema funciona bien para las sesiones de multidifusión de corta duración (por ejemplo, unos cuantos minutos), ya que no requiere configuración antes de que comience la transmisión.

Extensiones de multidifusión para el protocolo de la ruta más corta primero (MOSPF). En vez de ser un protocolo de enrutamiento de multidifusión de propósito general, MOSPF está diseñado para pasar rutas de multidifusión entre enrutadores dentro de una organización. Así, en vez de una metodología de multidifusión de propósito general, MOSPF se basa en OSPF y usa la metodología de enrutamiento de estado de enlace.

La figura 26.7 resume los protocolos de enrutamiento de multidifusión antes descritos.

Protocolo	Tipo
DVMRP	Configuración y uso de túneles
CBT	Descubrimiento basado en dirección central
PIM-SM	Descubrimiento basado en dirección central
PIM-DM	Inundar y podar
MOSPF	Estado de enlace (dentro de una organización)

Figura 26.7 Los protocolos de enrutamiento por multidifusión y la metodología que usa cada uno.

No obstante veinte años de investigación y muchos experimentos, la multidifusión de Internet de propósito general no ha sido exitosa. Incluso las aplicaciones de teleconferencia no proveen suficiente incentivo. Podemos resumir los resultados de la siguiente manera:

Las características dinámicas de la multidifusión de Internet dificultan el problema de la propagación de rutas de multidifusión. Aunque se propusieron muchos protocolos, Internet no tiene actualmente una herramienta de enrutamiento por multidifusión a nivel global.

26.17 Resumen

La mayoría de los hosts usan el enrutamiento estático en el cual la tabla de reenvíos se inicializa durante el arranque del sistema; los enrutadores usan el enrutamiento dinámico en el que el software de propagación de ruta actualiza continuamente la tabla de reenvíos. En términos de enrutamiento, Internet se divide en un conjunto de sistemas autónomos. Los protocolos utilizados para comunicar rutas entre los sistemas autónomos se conocen como *protocolos de puerta de enlace exterior* (EGP), mientras que los protocolos que se usan para comunicar la información de enrutamiento dentro de un sistema autónomo se conocen como *protocolos de puerta de enlace interior* (IGP).

El *protocolo de puerta de enlace límite* (BGP) es el EGP principal en Internet. Los ISP de nivel 1 usan BGP para informar unos a otros sobre sus clientes. Los IGP son: RIP, OSPF e IS-IS.

Como la multidifusión de Internet permite la membresía de grupos dinámicos y cualquier fuente puede enviar a un grupo de multidifusión sin ser un miembro, el problema de la propagación de rutas por multidifusión es difícil. Aunque se han propuesto varios protocolos de enrutamiento por multidifusión, no existe una tecnología de multidifusión a nivel de Internet.

EJERCICIOS

- 26.1** Haga una lista de las dos categorías amplias de enrutamiento de Internet y explique cada una.
- 26.2** ¿Cuáles son las dos entradas que se necesitan en la tabla de reenvíos de un host típico?
- 26.3** Suponga que todos los enrutadores en Internet contienen una ruta predeterminada. Demuestre que debe existir un bucle de enrutamiento.
- 26.4** ¿Qué es un sistema autónomo?
- 26.5** Mencione y explique los dos tipos de protocolos de enrutamiento de Internet.
- 26.6** Suponga que un enrutador en una organización usa un protocolo de enrutamiento para declarar que cierto destino está a diez saltos de distancia cuando el destino está a solo tres saltos. ¿Es la declaración necesariamente un error? Explique.
- 26.7** ¿Cuál es la consecuencia esperada cuando un enrutador anuncia las rutas hacia un destino específico?
- 26.8** Mencione y explique las características de BGP.
- 26.9** ¿Dónde se usa BGP?
- 26.10** ¿Qué tipo de algoritmo de enrutamiento emplea el RIP, y dónde se usa?
- 26.11** Mencione las características del RIP.
- 26.12** Cuando un enrutador recibe un mensaje de RIP, ¿cómo divide cada dirección IP en un prefijo y un sufijo?
- 26.13** Escriba un programa de computadora que lea un mensaje de actualización del RIP e imprima el contenido de cada campo.
- 26.14** El RIP limita los valores de distancia a un máximo de 16 saltos. Idee una intrarrred corporativa de ejemplo que tenga más de 16 enrutadores y más de 16 redes, pero que de todas formas pueda usar el RIP.

- 26.15** Mencione las características del OSPF.
- 26.16** ¿Cuál es el significado de “abierto” en OSPF?
- 26.17** ¿Por qué el OSPF tiene varias áreas?
- 26.18** ¿Qué protocolo tiene menor sobrecarga, OSPF o IS-IS? ¿Cuál tiene más características?
- 26.19** ¿Cuál es el propósito principal del IGMP y dónde se usa?
- 26.20** ¿Cuáles son las tres principales metodologías que se usan para reenviar datagramas de multidifusión?
- 26.21** Suponga que usted y dos amigos en colegios distantes desean participar en una teleconferencia de 3 vías mediante el uso de multidifusión IP. ¿Qué protocolos de enrutamiento de multidifusión elegiría? ¿Por qué?
- 26.22** Aunque cada grupo de multidifusión IP necesita una dirección de multidifusión IP única, usar un servidor central para asignar direcciones únicas crea un cuello de botella. Idee un esquema que permita a un conjunto de computadoras elegir una dirección de multidifusión al azar y resolver un conflicto, en caso de que surja uno.
- 26.23** El tráfico generado por el método de inundar y podar limita el tamaño de la región de la red a través de la cual puede usarse. Estime el tráfico total en una red si cada uno de los G grupos de multidifusión generan tráfico a una velocidad de P paquetes por segundo, cada paquete contiene B bits, N redes constituyen la intranet y cada red contiene al menos un componente de escucha para cada grupo.
- 26.24** ¿Se implementa la multidifusión ampliamente en Internet? Explique.
- 26.25** ¿Qué protocolos de multidifusión permiten que un mensaje de multidifusión se envíe antes de que el protocolo haya establecido rutas?
- 26.26** A pesar de la evidencia de que la multidifusión de Internet era inefectiva, los diseñadores del IPv6 optaron por especificar que IPv6 debería depender en gran medida de la multidifusión. Lea sobre el efecto de la política en organizaciones de estándares y escriba un informe breve.

**Los capítulos 27 a 33 se encuentran
en español en el sitio Web del libro.**

PARTE V

Otros conceptos de tecnología de redes

**Rendimiento de la red, QoS,
seguridad, administración
y tecnologías emergentes**

Capítulos

- 27 Rendimiento de la red (QoS y DiffServ)**
- 28 Multimedia y telefonía IP (VoIP)**
- 29 Seguridad de redes**
- 30 Administración de redes (SNMP)**
- 31 Redes definidas mediante software (SDN)**
- 32 La Internet de las cosas**
- 33 Tendencias en tecnologías y usos de las redes**

Contenido del capítulo

- 27.1 Introducción, 469
- 27.2 Medidas de rendimiento, 469
- 27.3 Latencia o retraso, 470
- 27.4 Capacidad, velocidad de transferencia y caudal útil, 472
- 27.5 Comprensión de la velocidad de transferencia
y el retraso, 473
- 27.6 Fluctuación, 474
- 27.7 La relación entre el retraso y la velocidad de transferencia,
475
- 27.8 Medición del retraso, la velocidad de transferencia
y la fluctuación, 476
- 27.9 Medición pasiva, paquetes pequeños y NetFlow, 478
- 27.10 Calidad del servicio (QoS), 479
- 27.11 QoS de grano fino y de grano grueso, 480
- 27.12 Implementación de QoS, 482
- 27.13 Tecnologías de QoS en Internet, 484
- 27.14 Resumen 485

27

Rendimiento de la red (QoS y DiffServ)

27.1 Introducción

Los primeros capítulos de esta parte consideran las propiedades fundamentales de los sistemas de comunicaciones de datos y describen las relaciones entre señales, frecuencias, ancho de banda, codificación de canales y transmisión de datos. Dichos capítulos explican las medidas de los sistemas de transmisión de datos, hablan sobre el tamaño de las redes de datos y explican que cada tecnología de redes se clasifica como PAN, LAN, MAN o WAN.

Este capítulo considera el tema del rendimiento de una red; habla sobre las medidas cuantitativas de las redes y explica cómo es que los protocolos y las tecnologías de reenvío de paquetes pueden implementar mecanismos que proporcionen prioridad para cierto tráfico.

27.2 Medidas de rendimiento

Usamos de manera informal el término *velocidad* para describir el desempeño de la red y nos referimos a las redes de *baja velocidad* o de *alta velocidad*. Sin embargo, dichas definiciones son inadecuadas debido a que las tecnologías de redes cambian con tanta rapidez que una red clasificada como de “alta velocidad” puede volverse de mediana o baja velocidad en poco menos de tres o cuatro años. Por lo tanto, en vez de descripciones cualitativas, los ingenieros y científicos usan medidas formales y cuantitativas para especificar en forma precisa el rendimiento de una red. Después de revisar las medidas básicas, explicaremos cómo se usan éstas para implementar servicios en niveles. Aunque a menudo los principiantes prefieren descripciones informales, las medidas cuantitativas son importantes ya que hacen posible comparar las características exactas de dos redes y desarrollar mecanismos que proporcionen una prioridad más alta para cierto tráfico. La figura 27.1 enlista las principales medidas de rendimiento de una red, y las siguientes secciones explican cada una de estas medidas.

Medida	Descripción
Latencia (retraso)	El tiempo requerido para transferir datos a través de una red
Velocidad de transferencia (capacidad)	La cantidad de datos que pueden transferirse por unidad de tiempo
Fluctuación (inestabilidad)	Los cambios en el retraso que ocurren y la duración de los cambios

Figura 27.1 Medidas clave del rendimiento de las redes de datos.

27.3 Latencia o retraso

La primera propiedad de las redes que pueden medirse en forma cuantitativa es la *latencia* o el *retraso*. La latencia especifica cuánto tiempo tardan los datos en viajar a través de una red de una computadora a otra, y se mide en fracciones de segundos. Los retrasos a través de Internet dependen de la infraestructura involucrada, así como de la ubicación del par específico de computadoras que se comunican. Aunque los usuarios se preocupan por el retraso total de una red, los ingenieros necesitan medidas más precisas. Por consiguiente, los ingenieros informan comúnmente sobre el retraso máximo y el promedio, y dividen un retraso en varias partes constituyentes. La figura 27.2 enumera los diversos tipos de retraso.

Tipo	Explicación
Retraso de propagación	El tiempo requerido para que una señal viaje a través de un medio de transmisión
Retraso de acceso	El tiempo necesario para obtener acceso a un medio de transmisión (por ejemplo, un cable)
Retraso de conmutación	El tiempo requerido para reenviar un paquete
Retraso de puesta en cola	El tiempo que pasa un paquete en la memoria de un conmutador o enrutador esperando a que lo seleccionen para la transmisión
Retraso de servidor	El tiempo requerido para que un servidor responda a una solicitud y envíe una respuesta

Figura 27.2 Varios tipos de retraso y una explicación de cada uno.

Retraso de propagación. En la red surge cierto retraso debido a que una señal requiere de una pequeña cantidad de tiempo para viajar a través de un medio de transmisión. En general, los retrasos de propagación son proporcionales a la distancia cubierta. Incluso con largos recorridos de cable, una LAN común que se utiliza dentro de un solo edificio tiene un retraso de propagación menor a un milisegundo. Aunque dichos retrasos parecen irrelevantes para un ser humano, una computadora moderna puede ejecutar más de cien mil instrucciones en un milisegundo. Por ello, un retraso de un milisegundo es considerable cuando un conjunto de computadoras necesitan coordinarse (por ejemplo, en la industria financiera, donde el tiempo exacto en que llega un pedido de inventario determina si se acepta o no). Una red que usa un satélite GEO tiene un retraso mucho mayor; incluso a la velocidad de la luz, un bit tarda cientos de milisegundos en viajar al satélite y regresar a la Tierra.

Retraso de acceso. Muchas redes usan medios compartidos. El conjunto de computadoras que comparten un medio deben competir por el acceso. Por ejemplo, una red Wi-Fi inalámbrica usa una metodología CSMA/CA para acceder a los medios. Dichos retrasos se conocen como *retrasos de acceso*. Los retrasos de acceso dependen de la cantidad de estaciones que compiten por el acceso y la cantidad de tráfico que envía cada estación. Los retrasos de acceso permanecen pequeños y fijos a menos que el medio se sobrecargue.

Retraso de conmutación. Un dispositivo electrónico en una red (por ejemplo, un conmutador o enrutador de nivel 2) debe calcular el siguiente salto para cada paquete antes de transmitirlo a través de una interfaz de salida. A menudo, el cálculo implica la búsqueda en una tabla, lo cual significa acceso a la memoria. En algunos dispositivos se necesita tiempo adicional para enviar el paquete a través de un mecanismo de comunicación interno, como un bus. El tiempo requerido para calcular un siguiente salto y comenzar la transmisión se conoce como *retraso de conmutación*. Las CPU rápidas y el hardware de propósito especial han hecho que los retrasos de conmutación sean los menos considerables en una red de computadoras.

Retraso de puesta en cola. El paradigma de almacenar y reenviar que se usa en la conmutación de paquetes significa que un dispositivo como un enrutador recolecta los bits de un paquete, los coloca en memoria, selecciona un siguiente salto y luego espera hasta poder enviar el paquete antes de comenzar la transmisión. Dichos retrasos se conocen como *retrasos de puesta en cola*. En el caso más simple, se coloca un paquete en una cola de salida PEPS (primero en entrar, primero en salir) y el paquete sólo necesita esperar hasta que se envíen los paquetes que llegaron primero. Los sistemas más complejos implementan un algoritmo de selección que da prioridad a algunos paquetes. Los retrasos de puesta en cola son variables, ya que el tamaño de una cola depende en su totalidad de la cantidad de tráfico que haya llegado recientemente. Estos retrasos representan la mayor parte de los retrasos en Internet, y cuando éstos se vuelven grandes, decimos que la red está congestionada.

Retraso de servidor. Aunque no son parte de una red como tal, los servidores son esenciales para la mayoría de la comunicación. El tiempo requerido para que un servidor examine una solicitud, y calcule y envíe una respuesta constituye una parte importante del retraso en general. Los servidores ponen en cola las solicitudes entrantes, lo que significa que el retraso de un servidor es variable y depende de la carga actual. En muchos casos, la percepción que tiene un usuario del retraso en Internet se debe al retraso del servidor en vez de los retrasos de la red.

27.4 Capacidad, velocidad de transferencia y caudal útil

Una segunda propiedad fundamental de las redes que puede medirse en forma cuantitativa es la *capacidad* de una red, lo que a menudo se expresa como la máxima *velocidad de transferencia* que puede sostener la red. La velocidad de transferencia es una medida de la velocidad a la que pueden enviarse los datos a través de la red, y se especifica en *bites por segundo (bps)*. La mayoría de las redes de comunicaciones de datos ofrecen una tasa de velocidad de transferencia de más de 1 Mbps, y las redes de más alta velocidad operan con más rapidez que 1 Gbps. Sin embargo y como hemos visto, surgen casos especiales en donde una red tiene una velocidad de transferencia menor a 1 Kbps.

Puesto que la velocidad de transferencia puede medirse de varias formas, hay que tener cuidado de especificar con exactitud lo que se ha medido. Existen varias posibilidades:

- Capacidad de un solo canal o de varios canales
- Capacidad teórica o tasa de velocidad de transferencia efectiva
- Velocidad de datos lograda por una aplicación (caudal útil)

A menudo los distribuidores anuncian la capacidad teórica de su equipo y la velocidad de transferencia que se logra bajo condiciones óptimas. La capacidad de hardware da un límite superior en cuanto al rendimiento, ya que es imposible que un usuario envíe datos más rápido que la velocidad a la que el hardware puede transferir los bits.

A los usuarios no les importa la capacidad del hardware utilizado, sólo les interesa la velocidad a la que pueden transferirse los datos. Por lo general, los usuarios evalúan la *velocidad de datos efectiva* que logra una aplicación midiendo la cantidad de datos transferidos por unidad de tiempo. El término *caudal útil* describe la velocidad que logran las aplicaciones. La tasa del caudal útil es menor que la capacidad del hardware debido a que los protocolos imponen una sobrecarga. Es decir, parte de la capacidad de la red no está disponible para los datos de los usuarios, ya que los protocolos:

- Envían encabezados, colas e información de control de los paquetes
- Imponen un límite en cuanto al tamaño de la ventana (búfer de recepción)
- Generan retrasos mientras resuelven nombres y direcciones
- Usan una negociación para iniciar y terminar la comunicación
- Reducen la tasa de transmisión cuando se detecta una congestión
- Vuelven a transmitir los paquetes perdidos

La desventaja de usar el caudal útil como medida surge debido a que la cantidad de sobrecarga depende de la pila de protocolos que se utilice. Además de los protocolos de la capa de transporte, la capa de Internet y la capa 2, el caudal útil depende del protocolo de aplicación. Por ejemplo, considere usar el *protocolo de transferencia de archivos (FTP)* para medir el caudal útil a través de una red Ethernet. FTP usa TCP, que a su vez usa IP. Además, FTP no comprime los datos antes de la transmisión, sino que coloca los datos del usuario en segmentos de TCP, el protocolo TCP encapsula cada segmento en un datagrama de IP y este último protocolo encapsula cada datagrama en una trama de Ethernet. Por lo tanto, cada trama tiene un encabezado de Ethernet y un campo CRC, así como un encabezado de datagrama IP

y un encabezado TCP. Si un usuario elige una aplicación alternativa de transferencia de archivos o si se utiliza una pila de protocolos alternativa, el caudal útil puede cambiar. En conclusión:

Aunque proporciona una medida de la velocidad efectiva a la que pueden transferirse los datos a través de una red, el caudal útil depende de la aplicación.

27.5 Comprensión de la velocidad de transferencia y el retraso

En la práctica, la terminología que usan los profesionales de redes para describir la velocidad de transferencia de una red o su capacidad, puede ser confusa. Por ejemplo, los capítulos sobre las comunicaciones de datos definen el ancho de banda de un canal y explican la relación entre el ancho de banda del hardware y la velocidad máxima de datos. Por desgracia los profesionales de redes usan los términos *ancho de banda* y *velocidad* como sinónimos para velocidad de transferencia. De tal forma, podríamos escuchar a alguien decir que cierta red tiene una “velocidad de 1 Gbps”. Como alternativa, algunos anuncios usan la frase “ancho de banda de 1 Gbps”. En un intento por diferenciar entre los dos usos de *ancho de banda*, los ingenieros reservan *ancho de banda* para que signifique *ancho de banda analógico* y usan el término *ancho de banda digital* como sinónimo de *velocidad de transferencia*. Aunque dichas afirmaciones son comunes, pueden ser confusas debido a que la velocidad de transferencia, el retraso y el ancho de banda son propiedades independientes.

De hecho, la velocidad de transferencia es una medida de capacidad y no de velocidad. Para entender la relación, imagine que una red es una carretera entre dos lugares y los paquetes que viajan a través de la red son los auto(móvil)e(s) que viajan por la carretera. La velocidad de transferencia determina cuántos autos pueden entrar a la carretera por segundo y el retraso de propagación determina cuánto tarda un auto en viajar por la carretera de un lugar a otro. Por ejemplo, una carretera que puede aceptar un auto cada cinco segundos tiene una velocidad de transferencia de 0.2 autos por segundo. Si un auto requiere 30 segundos para recorrer toda la carretera, ésta tiene un retraso de propagación de 30 segundos. Ahora considere lo que ocurre si se abre un segundo carril en el camino (es decir, la capacidad se duplica). Existe la posibilidad que entren dos autos al mismo tiempo, por lo que la velocidad de transferencia se duplica a 0.4 autos por segundo. Desde luego que el retraso de 30 segundos permanecerá sin cambios ya que cada auto aún debe recorrer toda la distancia. Por consiguiente, al considerar las medidas de las redes, recuerde que:

El retraso de propagación especifica el tiempo que permanece un solo bit en tránsito dentro de una red. La velocidad de transferencia, que especifica cuántos bits pueden entrar a la red por unidad de tiempo, mide la capacidad de la red.

Los profesionales de redes tienen un aforismo interesante:

Siempre podemos lograr más velocidad de transferencia, pero nunca será posible disminuir el retraso.

La analogía de una carretera ayuda a explicar el aforismo: al agregar más carriles a una carretera aumenta el número de autos que pueden entrar a ésta por unidad de tiempo, pero no disminuye el tiempo total requerido por un auto para recorrer la carretera. Las redes siguen el mismo patrón: al agregar más rutas de transmisión en paralelo aumentará la velocidad de transferencia de la red, pero no disminuirá el retraso de propagación (que depende de la distancia abarcada).

27.6 Fluctuación

Hay una tercera medida de las redes que se está volviendo importante a medida que éstas se utilizan para la transmisión de voz y video en tiempo real. La medida, que se conoce como la *fluctuación* o *inestabilidad* de una red, evalúa la variación en el retraso. Dos redes pueden tener el mismo retraso promedio, pero distintos valores de fluctuación. En especial, si todos los paquetes que recorren cierta red tienen el mismo retraso D , entonces la red no tiene inestabilidad. Pero si los paquetes alternan entre un retraso de $D + \epsilon$ y de $D - \epsilon$, entonces la red tiene el mismo retraso promedio pero una fluctuación distinta de cero.

Para entender por qué es tan importante el retraso, considere el envío de voz a través de una red. Del lado emisor, la señal analógica se muestrea y se convierte en un valor digital de ocho bits que se emite cada 125μ segundos. Las muestras se recolectan en paquetes, los cuales a su vez se transfieren a través de la red. Del lado receptor, los valores digitales se extraen y se convierten de nuevo en una salida analógica. Si la red tiene cero fluctuación (es decir, cada paquete tarda exactamente el mismo tiempo en recorrer la red), la salida de audio coincidirá exactamente con la entrada original; de lo contrario, la salida será defectuosa. Hay dos metodologías generales para manejar la fluctuación:

- Diseñar una red isócrona sin fluctuación
- Usar un protocolo que compense la fluctuación

Las redes telefónicas tradicionales usan la primera metodología: el sistema telefónico implementa una *red isócrona* que garantiza que el retraso a lo largo de todas las rutas sea el mismo. De esta manera, si los datos digitalizados de una llamada telefónica se transmiten a través de dos rutas, el hardware se configura de forma que ambas rutas tengan exactamente el mismo retraso.

En la transmisión de voz o video a través de Internet se usa la segunda metodología. Aunque la red subyacente puede tener una fluctuación considerable, las aplicaciones de voz y video dependen de *protocolos en tiempo real* para compensar la fluctuación.[†] Puesto que el uso de protocolos en tiempo real es mucho menos costoso que crear una red isócrona, las compañías telefónicas están relajando los requerimientos estrictos para la isocronía. Desde luego que un protocolo no puede compensar toda la fluctuación, ya que si la variación en el retraso se vuelve excesiva, la salida se verá afectada. Por lo tanto, aun cuando se use la segunda metodología, los proveedores de servicios intentarán minimizar la fluctuación en sus redes.

[†] En el capítulo 28 hablaremos sobre la transmisión de datos en tiempo real a través de Internet.

27.7 La relación entre el retraso y la velocidad de transferencia

En teoría, el retraso y la velocidad de transferencia de una red son independientes. Pero en la práctica pueden estar relacionados. Para entender por qué, piense en la analogía de la carretera que describimos antes. Si entran autos a la carretera a intervalos de tiempo uniformes, los autos que viajan por la carretera a una velocidad uniforme estarán espaciados en intervalos uniformes. Si un auto reduce su velocidad por alguna razón (por ejemplo, en una intersección), los autos que estén detrás de él también reducirán su velocidad y provocarán una congestión de tráfico temporal. Los autos que entran a la carretera cuando ocurra la congestión experimentarán retrasos más extensos que los autos que viajen por una carretera sin congestión. Ocurre una situación similar en las redes. Si un enrutador tiene una cola de paquetes esperando cuando llegue un nuevo paquete, el nuevo paquete se colocará al final de la cola y tendrá que esperar mientras el enrutador reenvía los paquetes anteriores. Si ocurre una congestión, los paquetes experimentarán retrasos más extensos que los datos que entran a una red con poca actividad.

27.7.1 Uso como estimación del retraso

Los científicos de computadoras estudiaron la relación entre el retraso y la congestión, y descubrieron que en muchos casos el retraso esperado puede estimarse a partir del porcentaje actual de la capacidad de la red que se esté utilizando. Si D_0 indica el retraso cuando una red está inactiva y U es un valor entre 0 y 1 que indica el *uso* actual, el retraso efectivo D se obtiene mediante una fórmula simple:

$$D = \frac{D_0}{(1 - U)} \quad (27.1)$$

Cuando una red está completamente inactiva, U es cero y el retraso efectivo es D_0 . Cuando una red opera a la mitad de su capacidad, el retraso efectivo se duplica. A medida que el tráfico se aproxima a la capacidad de la red (es decir, a medida que U se acerca a 1), el retraso se approxima a infinito. Aunque la fórmula sólo ofrece una estimación del retraso efectivo, podemos concluir que:

La velocidad de transferencia y el retraso no son totalmente independientes. A medida que aumenta el tráfico en una red de computadoras, la congestión provoca que aumente el retraso. Una red que opera cerca del 100% de su capacidad de velocidad de transferencia experimenta un retraso severo.

En la práctica, los administradores de redes entienden que un uso extremadamente alto puede provocar un retraso desastroso. Por ello, la mayoría de ellos trabajan para mantener el nivel de uso bajo y miden el tráfico en cada red de manera constante. Cuando el uso promedio o pico comienza a aumentar y sobrepasa un umbral preestablecido, el administrador aumenta la capacidad de la red. Por ejemplo, si el nivel de uso se vuelve alto en una red Ethernet de 1Gbps, el administrador podría optar por reemplazarla con una red Ethernet de 10 Gbps. Como alternativa, podría optar por dividir una red en dos, colocando la mitad de los hosts en una parte y la otra mitad en la otra parte (dicha división es fácil con un conmutador de VLAN).

¿Qué tan alto debe ser el umbral de uso? No hay una respuesta sencilla. Muchos administradores seleccionan un valor conservador. Por ejemplo, un ISP importante que opera una red troncal de gran tamaño mantiene el nivel de uso en todos sus circuitos digitales en menos del 50%. Otros establecen los umbrales al 80% para ahorrar dinero. En cualquier caso, los administradores están por lo general de acuerdo en que una red no debe operar por encima del 90% de su capacidad.

27.7.2 Producto del retraso por la velocidad de transferencia

Una vez que se conocen el retraso y la velocidad de transferencia de una red, es posible calcular otra cantidad interesante: el *producto del retraso por la velocidad de transferencia*.[†] Para entender el significado de este producto, piense en la analogía de la carretera: cuando entran autos a una carretera a una velocidad fija de T autos por segundo y a un auto le toma D segundos recorrer la carretera, entonces entrarán $T \times D$ autos adicionales a la carretera para cuando el primer auto haya realizado un viaje completo. Por lo tanto, podrá haber un total de $T \times D$ autos en la carretera. En términos de redes, el número de bits que viajan por una red en un momento dado se obtiene de la siguiente forma:

$$\text{Bits presentes en una red} = D \times T \quad (27.2)$$

donde D es el retraso medido en segundos y T es la velocidad de transferencia medida en bits por segundo. Para resumir:

El producto del retraso por la velocidad de transferencia mide el volumen de datos que pueden estar presentes en una red. Una red con una velocidad de transferencia T y un retraso D puede tener un total de $T \times D$ bits en tránsito en cualquier momento dado.

El producto del retraso por la velocidad de transferencia es importante para cualquier red con un retraso muy largo o con una velocidad de transferencia muy grande, ya que afecta la transmisión. Una aplicación emisora puede transmitir un volumen extenso de datos antes de que el destino reciba el primer bit.

27.8 Medición del retraso, la velocidad de transferencia y la fluctuación

Las técnicas que se utilizan para medir la velocidad de transferencia y la fluctuación son relativamente simples. Para evaluar la velocidad de transferencia, un emisor transfiere un volumen grande de datos. Un receptor registra el tiempo desde que comenzaron a llegar los datos hasta que hayan llegado todos y calcula la velocidad de transferencia como la cantidad de datos enviados por unidad de tiempo. La técnica para medir la fluctuación se conoce como *tren de paquetes*: un emisor emite una serie de paquetes con un retraso pequeño fijo entre paquetes. Por lo general, los paquetes en el tren se envían uno detrás de otro. Un receptor registra el tiempo en el que llega cada paquete y usa la secuencia de tiempos para calcular las diferencias en el retraso.

A diferencia de las mediciones de la velocidad de transferencia o de la fluctuación, una medición precisa del retraso en una ruta del host A al host B requiere que los dos hosts tengan sus relojes sincro-

[†] Cuando se usa como medida del hardware, el producto del retraso por la velocidad de transferencia se conoce comúnmente como *producto del retraso por el ancho de banda*.

nizados. Además, para medir el retraso a través de una distancia corta (como una LAN), los relojes deben ser en extremo precisos. En vez de usar relojes sincronizados, muchas herramientas de medición de red seleccionan una metodología más sencilla: miden el tiempo de ida y vuelta, y lo dividen entre dos. Por ejemplo, se puede usar la función *ping*.

Medir el rendimiento de la red puede ser inesperadamente difícil por cuatro razones:

- Las rutas pueden ser asimétricas
- Las condiciones cambian con rapidez
- La medición puede afectar el rendimiento
- El tráfico es en ráfagas

El primer punto explica por qué no se pueden usar tiempos de ida y vuelta para aproximar la medición del retraso. El enrutamiento asimétrico indica que el retraso a lo largo de una ruta de *B* a *A* puede diferir de manera considerable en comparación con el retraso a lo largo de una ruta de *A* a *B*. Así, la mitad del tiempo de ida y vuelta tal vez no proporcione una medida precisa.

El segundo punto explica por qué puede ser difícil obtener una medida precisa del rendimiento de una red. Por ejemplo, considere una red compartida; si sólo un host envía datos, el host disfrutará de poco retraso, de una velocidad de transferencia elevada y de un bajo nivel de fluctuación. A medida que otros hosts comienzan a usar la red, se incrementa el retraso y la fluctuación, y se reduce la velocidad de transferencia. Además, como las condiciones cambian con rapidez, los retrasos pueden variar ampliamente en menos de un segundo. Por lo tanto, incluso si se toman mediciones cada diez segundos, una medición podrá omitir un cambio importante en el rendimiento.

El tercer punto sugiere que enviar tráfico de prueba para medir una red puede afectar el rendimiento de ésta. Por ejemplo, en el banco de pruebas de investigación de PlanetLab, fueron tantos los investigadores que usaron *ping* para medir el rendimiento, que el tráfico de *ping* dominó por completo al resto del tráfico. La situación se volvió tan severa que los administradores establecieron una política para desalentar el uso de *ping*.

El cuarto punto es fundamental: las redes de datos exhiben el comportamiento *en ráfagas*, lo cual significa que el tráfico no es uniforme. Si consideramos el tráfico que envía un host dado, el patrón de explosividad es obvio, ya que la mayoría de los hosts permanecen silenciosos hasta que un usuario ejecuta una aplicación que se comunica a través de Internet. Cuando un usuario introduce un URL en un navegador Web, el navegador obtiene todas las partes de la página y luego deja de comunicarse hasta que el usuario solicite otra página. De manera similar, si un usuario descarga el correo electrónico, la computadora host se comunica con un sistema de correo electrónico, descarga una copia del buzón del usuario y luego espera al usuario.

Lo interesante es que el total del tráfico de datos se da también en ráfagas. Lo ideal sería que la explosividad fuera un fenómeno local y que al combinar el tráfico de millones de usuarios de Internet, el resultado fuera un patrón de uso uniforme. Después de todo, no todos los usuarios leen el correo electrónico exactamente al mismo tiempo; en su lugar, mientras que un usuario está descargando su correo, otro usuario podría estar leyendo los mensajes que descargó previamente. De hecho, las mediciones de la red telefónica de voz muestran que el tráfico telefónico de millones de usuarios produce un total uniforme. Pero cuando se combina el tráfico de un millón de usuarios de Internet, el resultado no es un total uniforme, sino que presenta ráfagas de puntos máximos y mínimos. De hecho, los estadísticos dicen que el tráfico de datos es *autosimilar*, lo que significa que el tráfico es análogo a un *fractal*, donde el mismo perfil estadístico es evidente en cualquier nivel de granulado. De tal modo que si una empresa examina

una LAN, el tráfico de los hosts locales aparecerá en ráfagas. Si un ISP intermedio mide el tráfico de mil usuarios o un ISP grande mide el tráfico de diez millones de usuarios, el tráfico tendrá cantidades absolutas grandes pero exhibirá el mismo patrón estadístico general que el tráfico en una LAN.

Podemos resumir lo siguiente:

A diferencia del tráfico telefónico de voz, el tráfico de datos se da en ráfagas. Se dice que el tráfico de datos es autosimilar debido a que los totales del tráfico de datos exhiben el mismo patrón de ráfagas.

27.9 Medición pasiva, paquetes pequeños y NetFlow

Los administradores de redes que miden las redes distinguen entre dos formas de medición:

- Medición activa
- Medición pasiva

Ya vimos la desventaja de las técnicas de medición *activa*, donde al inyectar tráfico de medición en una red, éste puede cambiar el rendimiento de la misma. La alternativa es la medición *pasiva* que monitorea una red y cuenta los paquetes, pero no inyecta tráfico adicional. Por ejemplo, un ISP puede contar los bytes que se transfieren a través de un enlace en una cantidad de tiempo dada para producir una estimación del uso del enlace. Es decir, el ISP organiza una estación de monitoreo pasiva que observa una red a través de un intervalo de tiempo y acumula el total de bytes en todos los paquetes.

Lo interesante es que un ISP puede optar por medir el número de paquetes enviados, así como el número de bytes de datos. Para entender esto, tenga en cuenta que debido a que el uso de un enlace se mide como un porcentaje de la capacidad y ésta a su vez se mide en bits por segundo, un ISP necesita medir los bits de datos totales que se envían por unidad de tiempo. Sin embargo, la capacidad de los commutadores y enrutadores se mide en paquetes por segundo. La medición surge debido a que un enrutador o commutador realiza el cálculo de reenvío del siguiente salto una vez por paquete, sin importar el tamaño del paquete. Por lo tanto, el esfuerzo computacional realizado para reenviar paquetes es proporcional al número de paquetes procesados en vez del número de bits en un paquete. Cuando llega un flujo de datos a 1 Gbps, un commutador o enrutador realiza menos trabajo si el flujo se divide en unos cuantos paquetes grandes que si el flujo se divide en muchos paquetes pequeños. Los distribuidores de equipo de redes entienden que el rendimiento depende de los paquetes. Si el dispositivo de un distribuidor específico no puede manejar muchos paquetes por segundo, el departamento de marketing del distribuidor puede concentrar la atención en los datos en vez de la velocidad de los paquetes (es decir, reportar el rendimiento de sus productos cuando manejan paquetes grandes). En conclusión:

Para evaluar el uso de los enlaces, un ISP mide los datos totales transferidos a través de un enlace por unidad de tiempo; para evaluar el impacto en un enrutador o conmutador, un ISP mide el número de paquetes que se transfieren por unidad de tiempo.

Cisco creó *NetFlow*, una de las técnicas de medición pasiva más populares, que ahora es un estándar del IETF. Un enrutador que implementa NetFlow muestrea paquetes de manera estadística de acuerdo con los parámetros establecidos por el administrador de red (por ejemplo, muestrea uno de cada mil paquetes). La información se extrae del encabezado de cada paquete muestreado, luego se sintetiza y el resumen se envía a un sistema de administración de redes en donde se procesa (a menudo, los datos se guardan en disco para su posterior análisis). Por lo general, NetFlow extrae las direcciones IP de origen y de destino, el tipo de datagrama y los números de puerto de protocolo. Para asegurar que sea pasivo, un enrutador que ejecuta NetFlow debe enviar los resúmenes de NetFlow a través de un puerto de administración especial en vez de enrutarlos a través de una de las redes que manejan los datos de usuario.

27.10 Calidad del servicio (QoS)

La contraparte de la medición de la red es el *aprovisionamiento de red*, lo que es lo mismo: diseñar una red para proporcionar un nivel específico de servicio. El resto del capítulo considera los mecanismos que pueden usarse para implementar garantías de servicio. En general, el tema se conoce como *calidad del servicio (QoS)*.

Para comprender el concepto de QoS, considere el contrato entre un proveedor de servicios y un cliente. En su forma más simple, un contrato define un servicio especificando la velocidad de datos que garantiza el proveedor. Por ejemplo, un proveedor que ofrece una conexión DSL a Internet podría garantizar una velocidad de datos de 2.2 Mbps. Los contratos más complejos definen *servicios en capas*, donde el nivel de servicio recibido depende de la cantidad pagada. Por ejemplo, un proveedor podría elegir un enfoque de *prioridad* que garantice que los paquetes de un cliente que se suscriba al nivel de servicio platino tendrán prioridad sobre los paquetes de los clientes que se suscriban a un nivel de servicio plata.

Los clientes corporativos de gran tamaño a menudo exigen *garantías de servicio* más estrictas. Por lo general, la industria financiera crea contratos de servicio con límites en el retraso entre ubicaciones específicas. Por ejemplo, una casa de bolsa podría necesitar un contrato de servicio que especifique que los paquetes deben transferirse de la oficina principal de la empresa a la Bolsa de valores de Nueva York en menos de 10 milisegundos; una empresa que respalde todo su centro de datos cada noche podría necesitar un contrato de servicio que garantice una velocidad de transferencia que no sea menor a 1 Gbps en las conexiones TCP que se utilizan para el respaldo.

Un contrato entre ISP y cliente que especifica los detalles del servicio ofrecido, se conoce como *acuerdo de nivel de servicio (SLA)*. Un SLA contiene lenguaje legal y puede ser difícil de interpretar. Por ejemplo, un SLA puede comenzar con la descripción de un circuito rentado que proporciona una

velocidad de datos efectiva de 155 Mbps (tasa de OC-3). Sin embargo, en otra parte del SLA podría aparecer el término *tasa de información comprometida (CIR)* con un valor de cero. Legalmente, dicho contrato significa que el proveedor sólo garantizó 0 bits por segundo en vez de 155 Mbps.

27.11 QoS de grano fino y de grano grueso

¿Cómo puede un proveedor especificar garantías de QoS y qué tecnologías usa para implementarla? La figura 27.3 enumera las dos metodologías generales propuestas para la especificación del servicio. Como se indica en la figura, las metodologías difieren en cuanto al nivel de su granulado y en si es el proveedor o el cliente quien selecciona los parámetros.

Metodología	Descripción
Grano fino	Un proveedor permite a un cliente indicar los requerimientos de QoS específicos para una instancia de comunicación dada; un cliente hace una solicitud cada vez que se crea un flujo (por ejemplo, para cada conexión TCP)
Grano grueso	Un proveedor especifica unas cuantas clases amplias de servicio, cada una de las cuales es adecuada para un tipo de tráfico; un cliente debe adaptar todo el tráfico a las clases

Figura 27.3 Dos metodologías que se propusieron para la especificación de los servicios de QoS.

27.11.1 QoS de grano fino y flujos

Gran parte de los primeros trabajos sobre QoS surgió de las compañías telefónicas. Los diseñadores asumieron una red de datos orientada a la conexión que se modeló a partir del sistema telefónico: cuando un cliente necesitaba comunicarse con un sitio remoto (por ejemplo, un servidor Web), los clientes creaban una conexión. Además, los diseñadores asumieron que un cliente emitiría los requerimientos de QoS para cada conexión y un proveedor calcularía un cargo de acuerdo con la distancia abarcada y el tipo de QoS utilizado.

Las compañías telefónicas incorporaron muchas características de QoS en el diseño del *modo de transmisión asíncrona (ATM)*. Aunque el ATM no sobrevivió y los proveedores casi nunca cobran por cada conexión, parte de la terminología que el ATM creó para QoS de grano fino se sigue utilizando con ligeras modificaciones. En vez de especificar QoS en una conexión, ahora usamos el término *flujo*. Un flujo generalmente se refiere a la comunicación en la capa de transporte, como podría ser una conexión TCP, un conjunto de paquetes UDP que viajan entre un par de aplicaciones o una llamada telefónica VoIP. La figura 27.4 muestra una lista con las cuatro categorías principales de servicio que estuvieron presentes en el ATM y explica cómo se relacionan con los flujos.

Acrónimo	Expansión	Significado
CBR	Tasa constante de bits	Los datos entran al flujo a una velocidad fija, como los datos de una llamada de voz digitalizada que entran exactamente a 64 Kbps
VBR	Tasa variable de bits	Los datos entran al flujo a una velocidad variable dentro de límites estadísticos especificados
ABR	Tasa disponible de bits	El flujo acepta usar la tasa de datos disponible en un momento dado
UBR	Tasa no especificada de bits	No se especifica una tasa de bits para el flujo; la aplicación se conforma con el servicio del mejor esfuerzo

Figura 27.4 Cuatro categorías principales de servicio QoS.

Como se indica en la figura, el servicio CBR es apropiado para un flujo que transfiere datos a una velocidad fija, siendo la voz digitalizada el ejemplo canónico. El servicio VBR es apropiado para un flujo que utiliza una codificación de tasa variable. Por ejemplo, algunos códigos de video envían codificaciones diferenciales, en donde la cantidad de datos enviados para una trama es proporcional a la diferencia entre la trama anterior y la actual. En tales casos, un cliente puede especificar la velocidad de datos promedio esperada, así como la velocidad máxima de datos y la longitud de tiempo en que ocurrirá la velocidad máxima. VBR pide a los usuarios que especifiquen lo siguiente:

- Tasa sostenida de bits (SBR)
- Tasa pico de bits (PBR)
- Tamaño sostenido de ráfaga (SBS)
- Tamaño pico de ráfaga (PBS)

El servicio ABR implica compartir: un cliente está dispuesto a pagar por cualquier cantidad de servicio disponible. Si otros clientes envían datos, la cantidad disponible será menor (y probablemente el proveedor cobrará menos). Por último, un servicio UBR significa que el cliente no desea pagar tarifas más altas y está satisfecho con el servicio del mejor esfuerzo.

Cuando se consideró QoS por primera vez en Internet, las compañías telefónicas argumentaron que se necesitarían servicios de grano fino para que la calidad de las llamadas telefónicas de voz sobre una red de paquetes fuera aceptable. En consecuencia, además del trabajo en el ATM, la comunidad de investigación comenzó a explorar la tecnología QoS de grano fino en Internet. La investigación se conoció como *servicios integrados (IntServ)*.

Después de muchos años de investigación en servicios integrados y de la creación de varios protocolos, la comunidad de investigación y el IETF concluyeron que, en general, una metodología de grano fino era tanto impráctica como innecesaria. Por una parte, un usuario promedio no tendría el suficiente entendimiento de QoS como para elegir los parámetros. Después de todo, ¿qué tasa de velocidad de transferencia especificaría uno para conectarse a un sitio Web cotidiano? Por otra parte, los enrutadores básicos no tienen el suficiente poder de procesamiento como para implementar QoS por flujos. Entonces, la mayoría del trabajo sobre QoS se concentra en definir unas cuantas clases amplias de servicio en vez de tratar de proporciona QoS de extremo a extremo a cada flujo individual. Podemos resumir lo siguiente:

A pesar de muchos años de investigación y de trabajo en los estándares, la metodología de grano fino para QoS se relegó a unos cuantos casos especiales.

27.11.2 QoS de grano grueso y clases de servicios

La alternativa a QoS de grano fino es la metodología de grano grueso, en la que el tráfico se divide en *clases* y se asignan parámetros de QoS a la clase en vez de asignarlos a los flujos individuales. Para entender la justificación de la metodología de grano grueso, es necesario considerar la implementación de QoS en un enrutador de núcleo. Las conexiones a los enrutadores pueden operar a 10 Gbps cada una, lo que significa que los paquetes llegan a una tasa extremadamente alta. Se necesita hardware especial para realizar el reenvío, ya que los procesadores convencionales son demasiado lentos. Además, y puesto que lleva tráfico entre los principales ISP, un enrutador de núcleo puede manejar millones de flujos simultáneos. Por lo tanto, QoS requiere muchos recursos adicionales. Un enrutador debe mantener el estado para millones de flujos y debe realizar un cálculo complejo para cada paquete. Además, un enrutador debe asignar recursos cuando comienza un flujo y desasignarlos cuando termina.

27.12 Implementación de QoS

La figura 27.5 ilustra los cuatro pasos que un conmutador o enrutador usan para implementar QoS.

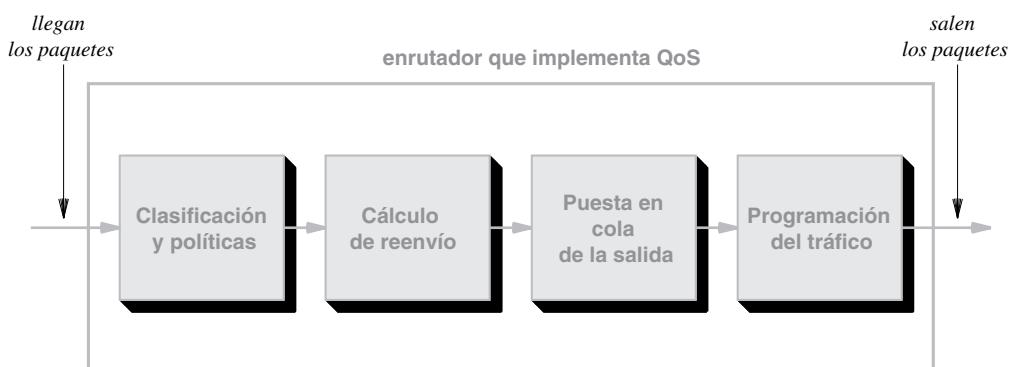


Figura 27.5 Los cuatro pasos clave que se utilizan para implementar QoS.

Clasificación y políticas. Cuando llega un paquete, un enrutador lo *clasifica* asignándole un identificador de flujo. Para un sistema de grano fino, el identificador especifica una conexión individual; para un sistema de grano grueso, el identificador especifica una clase de tráfico. Una vez que se asigna un identificador, el enrutador realiza el *establecimiento de políticas o policing*, lo que significa que el enrutador verifica que el paquete no viole los parámetros para el flujo. En especial, si un cliente envía datos con más rapidez que la velocidad máxima que paga, un agente de limitación de tráfico comienza a descartar paquetes. Una técnica que se utiliza para establecer políticas es el *descarte aleatorio anticipado (RED)* en el que los paquetes de un flujo dado se descartan según la probabilidad. Se establece una cola para el flujo y se usa el tamaño actual de esa cola para determinar la probabilidad de descarte. Antes de que la cola esté medio llena, la probabilidad se establece en cero. Cuando la cola está totalmente llena, la probabilidad se establece en uno. Entre estos dos tamaños de la cola, la probabilidad es linealmente proporcional al número de paquetes en la cola. El uso de RED evita un problema cíclico provocado por el *descarte de la parte final*, en el que se descartan todos los paquetes entrantes una vez que se llena una cola, luego varias sesiones TCP retroceden y comienzan con lentitud, el tráfico aumenta hasta que la cola se llena de nuevo, y el ciclo se repite.

Cálculo de reenvío. Al calcular el siguiente salto, un enrutador o conmutador puede usar el identificador de flujo. En algunos casos, el identificador de flujo determina la ruta a seguir (por ejemplo, se envía todo el tráfico de voz por el puerto 54 a un conmutador de voz). En otros casos, se ignora al identificador de flujo y se utiliza la dirección de destino en cada paquete para seleccionar un siguiente salto. Los detalles exactos del reenvío dependen del propósito de un conmutador o enrutador específicos y de las políticas de QoS del administrador.

Puesta en cola de la salida. La mayoría de las implementaciones de QoS crean un conjunto de colas para cada puerto de salida. Una vez que el cálculo de reenvíos selecciona un puerto de salida para el paquete, el mecanismo de puesta en cola de la salida usa el identificador de flujo para colocar el paquete en una de las colas asociadas con el puerto. Generalmente, el sistema de grano grueso usa una cola por clase. Por lo tanto, si un administrador establece ocho clases de QoS, cada puerto de salida tendrá ocho colas. Por lo general un sistema de grano fino tiene una cola por conexión y las colas se clasifican en una jerarquía. Por ejemplo, un chip de procesador de red proporciona 256,000 colas dispuestas en una jerarquía multinivel.

Programación de tráfico. Un *programador de tráfico* implementa las políticas de QoS seleccionando un paquete a enviar cada vez que haya un puerto inactivo. Por ejemplo, un administrador podría especificar que tres clientes reciban cada uno el 25% de la capacidad y que todos los demás clientes compartan la capacidad restante. Para implementar dicha política, un programador de tráfico podía usar cuatro colas y una metodología *por turnos* para seleccionar los paquetes. Por consiguiente, si todos los clientes envían datos, cada uno de los tres clientes designados recibirá un cuarto de la capacidad, según lo especificado.

Pueden usarse algoritmos más sofisticados de selección de paquetes para implementar formas complejas de compartición proporcional. La complejidad surge debido a que un programador de tráfico debe mantener políticas de largo plazo, aun cuando los paquetes lleguen en ráfagas. Por lo tanto, un programador de tráfico debe adaptarse a situaciones donde cierta cola excede temporalmente su velocidad de datos asignada, siempre y cuando el promedio de largo plazo cumpla con los límites establecidos. De manera similar, un programador de tráfico debe adaptarse a una situación en la que una o más colas están temporalmente vacías, al dividir la capacidad no utilizada entre otras colas.

Se han propuesto y analizado muchos algoritmos de programación de tráfico. No es posible crear un algoritmo práctico que logre la perfección; cada uno es un compromiso entre lo equitativo y la sobrecarga computacional. La figura 27.6 muestra una lista con algunos de los algoritmos de gestión de tráfico que se han propuesto y estudiado.

Algoritmo	Descripción
Cubeta con goteo (Leaky Bucket)	Permite que una cola envíe paquetes a una velocidad fija al incrementar periódicamente un contador de paquetes y usarlo para controlar la transmisión
Cubeta con señales token (Token Bucket)	Permite que una cola envíe datos a una velocidad fija al incrementar periódicamente un contador de bytes y utilizarlo para controlar la transmisión
Turno ponderado (Weighted Round Robin)	Selecciona paquetes de un conjunto de colas de acuerdo con un conjunto de ponderaciones que dividen la capacidad en porcentajes fijos, asumiendo un tamaño de paquete uniforme
Turno por déficit (Deficit Round Robin)	Una variante de la metodología por turnos que toma en cuenta los bytes enviados en vez de los paquetes transferidos, y permite un déficit temporal provocado por un paquete grande

Figura 27.6 Ejemplos de algoritmos de programación de tráfico.

27.13 Tecnologías de QoS en Internet

El IETF diseñó una serie de tecnologías y protocolos relacionados con QoS. Los tres esfuerzos importantes son:

- RSVP y COPS
- DiffServ
- MPLS

RSVP y COPS. Al explorar IntServ, el IETF desarrolló dos protocolos para proporcionar QoS: el *protocolo de reservación de recursos (RSVP)* y el *protocolo de servicios comunes de políticas abiertas (COPS)*. RSVP es una versión de grano fino de QoS. Por consiguiente, se necesita RSVP para cada sesión TCP o UDP. Para usar RSVP, una aplicación envía una solicitud que especifica el tipo de QoS deseado. Cada enrutador a lo largo de la ruta del origen al destino reserva los recursos solicitados y pasa la solicitud al siguiente enrutador. En un momento dado, el host de destino debe aceptar la solicitud. Cuando cada salto a lo largo de la ruta acepta honrar la solicitud, se genera y devuelve un identificador de flujo. Después puede enviarse el tráfico a lo largo de la ruta reservada. COPS es un protocolo complementario para RSVP que se utiliza para especificar e implementar políticas. Un enrutador que implementa las políticas usa COPS para comunicarse con un servidor de políticas y obtener información sobre los parámetros de flujo. Puesto que está diseñado para proporcionar QoS de grano fino por flujos, RSVP raras veces se usa.

DiffServ. Una vez que abandonó IntServ y el QoS de grano fino, el IETF creó los *servicios diferenciados (DiffServ)* para definir un mecanismo de QoS de grano grueso. El esfuerzo de DiffServ produjo una definición de la forma en que pueden especificarse las clases y cómo puede usarse el campo *TIPO DE SERVICIO* en un encabezado IPv4 o IPv6 para especificar la clase de un datagrama. Aunque varios ISP han experimentado con DiffServ, la tecnología no disfruta de una aceptación muy amplia.

MPLS. El capítulo 19 describe la *comutación de etiquetas multiprotocolo (MPLS)* como un mecanismo de comunicación orientado a la conexión, desarrollado sobre IP. Para usar MPLS, un administrador configura rutas de reenvío a través de un conjunto de enrutadores con capacidad para MPLS. En el extremo de una ruta, cada datagrama se encapsula en un encabezado MPLS y se inyecta en la ruta MPLS; en el otro extremo se extrae cada datagrama, se elimina el encabezado MPLS y se reenvía a su destino. En muchos casos, a una ruta MPLS se le asigna una política de programación de tráfico, lo que significa que cuando se inserta un datagrama en una ruta específica, se establecen parámetros de QoS para el datagrama. Por lo tanto, un ISP podría establecer una ruta MPLS para datos de voz que esté separada de la ruta MPLS que se utiliza para otros datos.

27.14 Resumen

Las dos principales medidas de rendimiento de una red son: el retraso (el tiempo requerido para enviar un bit de una computadora a otra) y la velocidad de transferencia (el número de bits por segundo que pueden transmitirse a través de una red). Aunque por lo general la velocidad de transferencia se conoce como velocidad, es una medida de la capacidad de la red. El producto del retraso por la velocidad de transferencia mide la cantidad de datos que pueden transitar en un instante dado. El retraso y la velocidad de transferencia no son independientes: a medida que la velocidad de transferencia se aproxima al 100% de la capacidad, los retrasos aumentan con rapidez.

La fluctuación es una medición de la variación en el retraso y se está volviendo importante en las redes de datos. Puede lograrse un nivel bajo de fluctuación con una red isócrona o con un protocolo que se encargue de la transmisión de audio y video en tiempo real; Internet usa la metodología del protocolo.

Puede ser difícil medir el rendimiento de una red. En las rutas asimétricas, se necesitan relojes sincronizados para medir el retraso, ya que con el tráfico en ráfagas, el rendimiento puede cambiar con rapidez. Puesto que el tráfico adicional que genera la medición puede alterar las condiciones de la red, muchos administradores prefieren tecnologías de medición pasivas como NetFlow.

Se han estudiado técnicas de QoS de grano fino y de grano grueso, aunque los esfuerzos de grano fino por lo general fueron abandonados. El ATM definió algunas categorías de servicio y aún se utilizan acrónimos como: tasa de bits constante (CBR), tasa de bits variable (VBR), tasa de bits disponible (ABR) y tasa de bits no especificada (UBR).

Para implementar QoS, un comutador o enrutador clasifica y asigna políticas a los datos entrantes, reenvía y coloca cada paquete en una cola de salida, y usa un programador de tráfico para seleccionar un paquete a enviar cuando se libere un puerto de salida. Se han propuesto y analizado varios algoritmos de programación de tráfico; cada uno tiene sus ventajas y desventajas en cuanto a lograr un punto óptimo de eficacia y sobrecarga computacional.

El IETF definió RSVP y COPS como parte del esfuerzo de IntServ. Cuando el QoS de grano fino dejó de ser lo primordial, el IETF definió DiffServ. El IETF también definió MPLS como una tecnología de ingeniería de tráfico. Los parámetros de QoS pueden asociarse con cada túnel de MPLS, lo que significa que, una vez que se clasifica un datagrama, su asociación de MPLS define sus parámetros de QoS.

EJERCICIOS

- 27.1** Mencione una lista de las tres medidas principales de rendimiento de una red e incluya sus descripciones.
- 27.2** Mencione cinco tipos de retrasos junto con una explicación de cada uno.
- 27.3** ¿Esperaría que los retrasos de acceso fueran más largos en una LAN o en una WAN? ¿Y los retrasos de puesta en cola? ¿Por qué?
- 27.4** ¿Cómo puede medirse la velocidad de transferencia?
- 27.5** ¿Qué nombre se usa para la forma de velocidad de transferencia más importante para un usuario?
- 27.6** Mencione ejemplos de procesamiento que hagan que el caudal útil sea menor a la capacidad del canal.
- 27.7** Proporcione una explicación del retraso y la velocidad de transferencia en términos de bits transferidos.
- 27.8** Entre el retraso y la velocidad de transferencia, ¿cuál de los dos proporciona el límite más fundamental sobre el rendimiento? ¿Por qué?
- 27.9** Use la función *ping* para medir la latencia de la red con respecto a sitios locales y distantes. ¿Cuáles son los retrasos mínimo y máximo que midió en Internet?
- 27.10** Si usamos *ping* con la dirección IP 127.0.0.1, la latencia es muy baja. Explique por qué.
- 27.11** Descargue una copia del programa *tcpdump* y úselo para medir la velocidad de transferencia en una red Ethernet local. ¿Cuál es el caudal útil? Calcule una estimación del uso que se logró del enlace.
- 27.12** Compare la velocidad de transferencia de una red de 100 Mbps y de una red de 1 Gbps.
- 27.13** ¿Qué es la fluctuación y cuáles son las dos metodologías utilizadas para solucionar este problema?
- 27.14** Algunas veces los profesionales se refieren a una “rodilla” en la curva de retraso. Para entender lo que esto significa, trace el retraso efectivo para los valores de uso entre 0 y 0.95. ¿Puede encontrar un valor de uso para el que la curva parezca incrementarse de manera considerable?
- 27.15** ¿Cuántos datos puede haber “en vuelo” entre una estación terrestre emisora, un satélite y una estación receptora? Para averiguarlo, calcule el producto del retraso por la velocidad de transferencia para una red satelital GEO que opera a 3 Mbps. Suponga que el satélite gira en órbita a 20,000 millas sobre la Tierra y que las transmisiones de radio se propagan a la velocidad de la luz.
- 27.16** ¿Por qué es difícil la medición del rendimiento de una red?
- 27.17** ¿Cuál es la diferencia entre el tráfico de datos y el tráfico de voz?
- 27.18** Explique por qué los ISP cuentan el número de paquetes recibidos por unidad de tiempo en vez de sólo el número de bytes recibidos por unidad de tiempo.
- 27.19** ¿Cuáles son los dos tipos de QoS?
- 27.20** Calcule una estimación del poder de cómputo necesario para implementar la QoS de grano fino en el núcleo de Internet. Suponga que un enlace de 10 Gbps entrega paquetes de 1000 bytes y N operaciones aritméticas por paquete, y calcule el número de operaciones que necesita realizar un procesador por segundo.

- 27.21** Mencione una lista de las cuatro categorías principales de QoS que se derivaron del ATM y proporcione el significado de cada una.
- 27.22** Considere un navegador Web. ¿Qué tipo de QoS sería apropiado para un flujo típico en donde el navegador descarga una página Web? ¿Por qué?
- 27.23** Si dos usuarios crean una sesión de chat a través de Internet, ¿qué categoría de QoS usarán?
- 27.24** ¿Cuáles son los cuatro parámetros que se utilizan para caracterizar un flujo VBR?
- 27.25** Explique los cuatro pasos que se utilizan para implementar QoS.
- 27.26** Si su ISP usa la cubeta con goteo para programar las transmisiones de paquetes, ¿su velocidad de transferencia será más alta con paquetes grandes o pequeños? Explique.
- 27.27** ¿Qué es DiffServ?
- 27.28** ¿Cuál es la diferencia entre el reenvío de MPLS y el reenvío de IP convencional?
- 27.29** ¿Por qué podría una corporación optar por asignar todo el tráfico VoIP a una sola clase de DiffServ?

Contenido del capítulo

- 28.1 Introducción, 489
- 28.2 Transmisión de datos en tiempo real y entrega del mejor esfuerzo, 489
- 28.3 Reproducción con retraso y búferes de fluctuación, 490
- 28.4 Protocolo de transporte en tiempo real (RTP), 491
- 28.5 Encapsulamiento de RTP, 492
- 28.6 Telefonía IP, 493
- 28.7 Señalización y estándares de señalización de VoIP, 494
- 28.8 Componentes de un sistema telefónico IP, 495
- 28.9 Resumen de protocolos y distribución en capas, 498
- 28.10 Características de H.323, 499
- 28.11 Distribución en capas de H.323, 499
- 28.12 Características y métodos de SIP, 500
- 28.13 Ejemplo de una sesión de SIP, 501
- 28.14 Asignación y enrutamiento de números telefónicos, 502
- 28.15 Resumen, 503

28

Multimedia y telefonía IP (VoIP)

28.1 Introducción

Los capítulos de esta parte del libro consideran una variedad de tecnologías y usos de las redes. El capítulo anterior habla sobre el rendimiento de las redes y la QoS. El capítulo presenta las dos formas básicas en que las redes pueden diseñarse para brindar un servicio que se utilice para aplicaciones en tiempo real (como las aplicaciones de voz): una infraestructura isócrona o el uso de protocolos que compensen la fluctuación.

Este capítulo continúa la explicación mediante un análisis de la transferencia de multimedia a través de Internet. El capítulo analiza la forma en que puede enviarse multimedia a través de un mecanismo de comunicación del mejor esfuerzo, describe un protocolo de propósito general para el tráfico en tiempo real y considera en detalle la transmisión de las llamadas telefónicas de voz.

28.2 Transmisión de datos en tiempo real y entrega del mejor esfuerzo

Usamos el término *multimedia* para referirnos a los datos que contienen audio o video, y puede incluir texto. La frase *multimedia en tiempo real* se refiere a los datos multimedia que deben reproducirse exactamente a la misma velocidad a la que se capturó (por ejemplo, un programa de noticias por televisión que incluye audio y video de un evento real).

Surge la pregunta: ¿cómo puede usarse Internet para la transmisión de multimedia en tiempo real? Para entender la dificultad que esto representa, recuerde que Internet ofrece el servicio de entrega del mejor esfuerzo. De tal forma, los paquetes pueden perderse, retrasarse o entregarse desordenados. Si

el audio o el video se digitalizan, se envían a través de Internet sin un tratamiento especial y luego se despliegan exactamente como vayan llegando, la salida resultante será inaceptable. Los primeros sistemas multimedia resolvieron el problema mediante la creación de un sistema de comunicación diseñado de manera específica para manejar audio y video. La red telefónica usa una red isocrónica para proveer una reproducción de audio de alta calidad, y los sistemas de televisión por cable están diseñados para entregar varios canales de video de difusión sin interrupciones ni pérdidas.

En vez de que las redes tengan que hacerse cargo de la transmisión en tiempo real, Internet usa un soporte de protocolo adicional. Curiosamente el problema más importante a resolver es la fluctuación y no la pérdida de paquetes. Para ver el porqué, considere un webcast en vivo. Si un protocolo usa el tiempo de espera y la retransmisión para reenviar el paquete, el paquete retransmitido llegará demasiado tarde como para ser útil; el receptor habrá reproducido el video y el audio de los paquetes sucesivos y no tiene sentido insertar un fragmento del webcast que se omitió antes.

El punto importante es que:

A diferencia de los protocolos de transporte convencionales, un protocolo que transfiere datos en tiempo real sólo se encarga del problema de la fluctuación y no retransmite los paquetes perdidos.

28.3 Reproducción con retraso y búferes de fluctuación

Para solucionar la fluctuación y lograr una reproducción uniforme de datos en tiempo real, se emplean dos técnicas principales:

- *Etiquetas de tiempo o timestamps.* Un emisor proporciona una etiqueta de tiempo para cada pieza de datos. Un receptor usa las etiquetas de tiempo para manejar los paquetes desordenados y desplegar los datos en la secuencia de tiempo correcta.
- *Búfer de fluctuación.* Para lidiar con la fluctuación (es decir, pequeñas variaciones en el retraso), un receptor coloca los datos en el búfer y retrasa la reproducción.

La implementación de un búfer de fluctuación es simple y directa. Un receptor mantiene una lista de elementos de datos y usa etiquetas de tiempo para ordenarla. Antes de comenzar la reproducción, el receptor se retrasa d unidades de tiempo, lo que significa que los datos que se están reproduciendo están d unidades de tiempo detrás de los datos que están llegando. Es decir, si cierto paquete se retrasa menos que d , el contenido del paquete se colocará en el búfer antes de que se necesite para la reproducción. En otras palabras, los elementos se insertan en el búfer de fluctuación con cierta variación en la velocidad, pero el proceso de reproducción extrae datos de un búfer a una tasa fija. La figura 28.1 ilustra la organización de un sistema de reproducción en tiempo real.

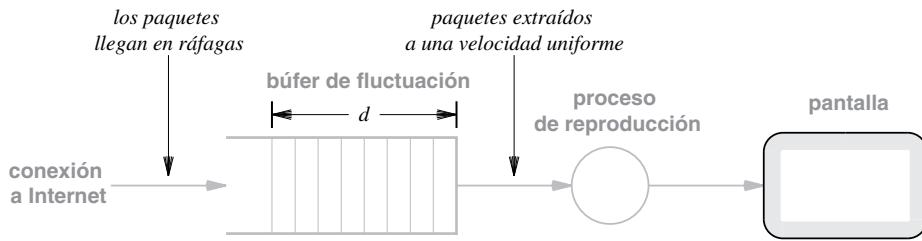


Figura 28.1 Ilustración de un búfer de fluctuación con el retraso d .

28.4 Protocolo de transporte en tiempo real (RTP)

En la suite de protocolos de Internet, el *protocolo de transporte en tiempo real (RTP)* proporciona el mecanismo que se utiliza para transmitir datos en tiempo real a través de Internet. El término *transporte* es un nombre poco apropiado debido a que RTP está sobre la capa de transporte. Entonces, a pesar del nombre, deberíamos considerar a RTP como un protocolo de transferencia.

RTP no asegura la entrega oportuna de los datos ni incluye un búfer de fluctuación o un mecanismo de reproducción, sino que brinda tres elementos en cada paquete que permiten que un receptor implemente un búfer de fluctuación:

- Un *número de secuencia* que permite a un receptor colocar los paquetes entrantes en el orden correcto y detectar los paquetes faltantes.
- Una *etiqueta de tiempo* que permite a un receptor reproducir los datos del paquete en el momento correcto del flujo de multimedia.
- Una serie de *identificadores de origen* que permiten a un receptor conocer el o los orígenes de los datos.

La figura 28.2 ilustra cómo aparecen los campos de número de secuencia, etiqueta de tiempo e identificador de origen en el encabezado de un paquete RTP.

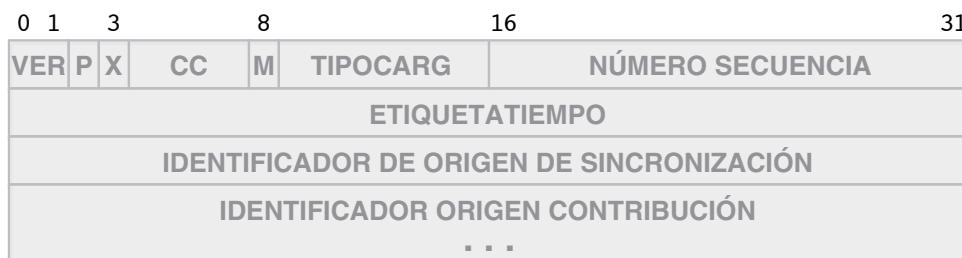


Figura 28.2 El encabezado básico que aparece al inicio de cada paquete RTP.

El campo *VER* proporciona el número de versión de RTP, que en la actualidad es la 2. El campo *P* especifica si la carga útil se rellena con ceros (debido a que ciertas codificaciones requieren un tamaño de bloque fijo). El campo *X* especifica si hay una extensión de encabezado presente y *CC* proporciona un conteo de los orígenes que se combinaron para producir el flujo como se describe a continuación. *M* es un bit marcador que puede usarse para marcar ciertas tramas. En especial, ciertas codificaciones de video envían una trama completa seguida de una serie de cambios incrementales. El bit *M* sólo se establece si un paquete RTP transporta una trama completa. El campo *TIPOCARG* especifica el tipo de carga útil; un receptor usa el valor de *TIPOCARG* para interpretar el resto del paquete.

Cada paquete incluye un campo *NÚMERO SECUENCIA*. La secuencia se incrementa en uno por cada paquete. Al igual que con TCP, un emisor selecciona una secuencia aleatoria de inicio para ayudar a evitar problemas de reproducción. Un campo *ETIQUETATIEMPO* proporciona a un receptor la información sobre la sincronización de la reproducción. Es importante mantener la etiqueta de tiempo independiente de la secuencia en casos en los que el tiempo no está relacionado de manera lineal con la secuencia de los paquetes (por ejemplo, una codificación de video de tamaño variable que envía menos paquetes cuando la imagen no cambia rápidamente).

La *ETIQUETATIEMPO* de RTP no codifica una fecha y hora, sino que RTP selecciona una etiqueta de tiempo inicial aleatoria y luego hace que cada etiqueta de tiempo sucesiva sea relativa al valor inicial. Además, RTP no especifica si el tiempo se mide en segundos, milisegundos o en otras unidades: el tipo de carga útil determina el nivel de granulado de la etiqueta de tiempo. Sin importar qué medida de tiempo se utilice, un emisor debe incrementar continuamente el tiempo aun cuando no se envíen paquetes (por ejemplo, si un códec suprime la transmisión durante períodos silenciosos de un flujo de audio).

Los dos campos *IDENTIFICADOR DE ORIGEN DE SINCRONIZACIÓN* e *IDENTIFICADOR DE ORIGEN DE CONTRIBUCIÓN* identifican los orígenes de los datos. La razón por la que debe identificarse un origen se debe al mecanismo de entrega por multidifusión: un host puede recibir datos de varios orígenes y puede recibir múltiples copias de un paquete dado. La razón por la que se identifican múltiples fuentes surge de una técnica conocida como *mezclado* en la que un sistema intermedio combina en tiempo real datos de varios flujos para producir un nuevo flujo. Por ejemplo, un mezclador puede combinar flujos de video y audio independientes de una película para luego transmitir el flujo combinado por multidifusión.

28.5 Encapsulamiento de RTP

RTP usa UDP para transportar mensajes. Por consiguiente, cada mensaje de RTP se encapsula en un datagrama de UDP para transmitirse a través de Internet. La figura 28.3 ilustra los tres niveles de encapsulamiento que se usan cuando se transfiere un mensaje de RTP a través de una sola red.

Puesto que RTP usa el encapsulamiento UDP, los mensajes resultantes pueden enviarse mediante difusión o multidifusión. La multidifusión es especialmente útil para la entrega de programación de entretenimiento que atrae a una gran audiencia. Por ejemplo, si un proveedor de cable ofrece un programa de televisión o un evento deportivo, varios clientes pueden mirarlo al mismo tiempo. En tales casos, en vez de enviar una copia de un mensaje a cada suscriptor, RTP permite a un proveedor llegar a sus clien-

tes mediante la multidifusión de una copia de un mensaje RTP a través de cada subred lógica. Si cierta multidifusión llega a un promedio de N clientes, la cantidad de tráfico se reducirá por un factor de N .

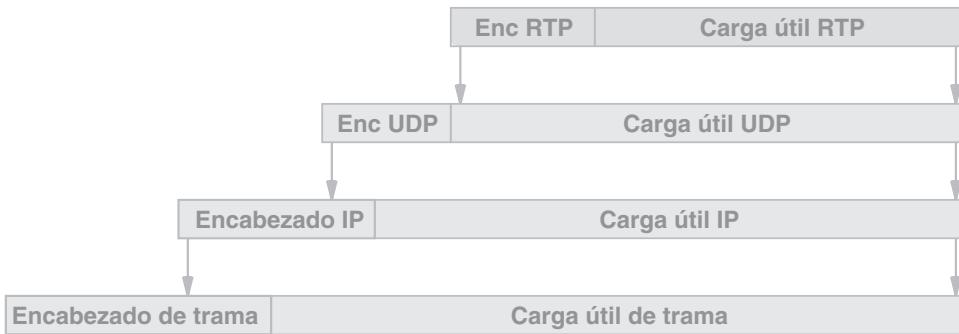


Figura 28.3 Los tres niveles de encapsulamiento que se utilizan con RTP.

28.6 Telefonía IP

El término *telefonía IP* o *voz sobre IP* (*VoIP*) se usa para describir una de las aplicaciones multimedia más utilizadas. Las compañías telefónicas en todo el mundo están reemplazando los conmutadores telefónicos tradicionales con enruteadores IP. Las razones son económicas, ya que los enruteadores cuestan mucho menos que los conmutadores telefónicos tradicionales. Las empresas también comenzaron a usar teléfonos IP por las mismas razones, puesto que al enviar tanto datos como voz en los datagramas IP se reduce el costo debido a que se comparte la infraestructura de red. Basta con un solo conjunto de equipo, cableado y conexiones de red para toda la comunicación, incluyendo las llamadas telefónicas.

La idea básica detrás de la telefonía IP es simple y directa: muestrear audio en forma continua, convertir cada muestra en formato digital, enviar el flujo digitalizado resultante a través de una red IP en forma de paquetes y convertir el flujo de nuevo a audio analógico para su reproducción. No obstante, muchos detalles complican la tarea. Un emisor no puede esperar a llenar un paquete de gran tamaño, pues si lo hace, la transmisión se retrasa por varios segundos. El sistema debe manejar la configuración de las llamadas; es decir, cuando un usuario que desea hacer una llamada marca, el sistema debe traducir el número telefónico en una dirección IP y localizar a la parte especificada. Cuando se inicia una llamada, la parte que la recibe debe aceptar y responder a la llamada. De manera similar, cuando termina una llamada, las dos partes deben acordar cómo deben terminar la comunicación.

Las complicaciones más considerables surgen debido a que la telefonía IP lucha por ser retrocompatible con la *red telefónica de conmutación pública (PSTN)*. Es decir, en vez de restringir las llamadas a teléfonos IP, los mecanismos permiten a quien hace o recibe la llamada usar un teléfono en cualquier parte en la PSTN, incluyendo una ubicación internacional o una conexión celular. Por lo tanto, un sistema telefónico IP debe estar preparado para manejar llamadas que se originen en la PSTN y terminen en un teléfono IP, o viceversa. Los usuarios esperan que un sistema de telefonía IP proporcione

servicios telefónicos existentes, como *reenvío de llamadas*, *llamadas en espera*, *correo de voz*, *llamadas de conferencias* e *identificador de llamadas*. Además, los negocios que operan actualmente una *central telefónica privada (PBX)* pueden requerir un sistema de telefonía IP para ofrecer los servicios equivalentes a una PBX.

28.7 Señalización y estándares de señalización de VoIP

Hay dos grupos que crearon estándares para la telefonía IP: la *Unión Internacional de telecomunicaciones (ITU)* que controla los estándares telefónicos y el *Grupo de tareas sobre ingeniería de Internet (IETF)*, que controla los estándares de TCP/IP. Después de considerar los componentes conceptuales de un sistema de telefonía IP, revisaremos los protocolos que eligió cada grupo.

Por fortuna, ambos grupos están de acuerdo en los fundamentos para la codificación y la transmisión de audio:

- El audio se codifica usando la *modulación de códigos de pulsos (PCM)*
- Se usa RTP para transferir el audio digitalizado

La principal complejidad de la telefonía IP y la razón por la que se han propuesto varios estándares recae en la configuración y la administración de las llamadas. En la terminología de telefonía, al proceso de establecer y terminar una llamada se le conoce como *señalización* e incluye los procesos de asignar un número telefónico a una ubicación, buscar una ruta hacia la parte que recibe la llamada y manejar detalles tales como el reenvío de llamadas. El mecanismo que se utiliza en el sistema telefónico tradicional para manejar la administración de llamadas se conoce como *sistema de señalización 7 (SS7)*.

Una de las preguntas fundamentales sobre la telefonía IP se centra en la metodología a elegir para la señalización: ¿debe el sistema de señalización ser centralizado como el sistema telefónico actual, o debe ser distribuido como la asignación actual de nombres de dominio para direcciones IP? Quienes proponen un método distribuido argumentan que debería ser posible para dos teléfonos IP, ubicados cada uno en cualquier punto en Internet, encontrarse y comunicarse justo como las aplicaciones actuales de Internet (es decir, un teléfono IP actúa como servidor para recibir llamadas entrantes y como cliente para realizar llamadas salientes). En una metodología distribuida no se necesita más infraestructura que DNS y los servicios de reenvío de IP que están disponibles actualmente para las comunicaciones de datos. La metodología distribuida es en especial pertinente para un sistema de telefonía IP local (por ejemplo, un sistema que permite llamadas entre dos teléfonos IP dentro de una misma compañía). Quienes proponen una metodología centralizada argumentan que un modelo telefónico convencional funciona mejor debido a que al dar el control de la configuración de las llamadas a las compañías telefónicas éstas pueden ofrecer garantías en el servicio.

Para que sean compatibles con los teléfonos existentes, los nuevos protocolos deben ser capaces de interactuar con SS7, tanto para colocar llamadas salientes como para aceptar llamadas entrantes. Conforme el debate sobre la metodología básica avanzó, se propusieron cuatro conjuntos de protocolos de señalización para usarse con la telefonía IP. El IETF propuso el *protocolo de iniciación de sesión (SIP)* y el *protocolo de control de puerta de enlace de medios (MGCP)*, mientras que la ITU propuso un conjunto grande y exhaustivo de protocolos bajo la categoría general de *H.323* y los dos grupos propusieron en conjunto a *Megaco (H.248)*. En conclusión:

Los procesos de configuración y terminación de llamadas se conocen como señalización. Se propusieron varios protocolos de señalización para usarse con la telefonía IP.

28.8 Componentes de un sistema telefónico IP

La figura 28.4 enlista los cuatro componentes principales de un sistema de telefonía IP y la figura 28.5 ilustra cómo se utilizan para interconectar redes.

Componente	Descripción
Teléfono IP	Funciona como un teléfono convencional pero usa IP para enviar voz digitalizada
Controlador de puerta de enlace de medios	Proporciona control y coordinación entre teléfonos IP para servicios tales como la configuración, la terminación y el reenvío de llamadas
Puerta de enlace de medios	Proporciona una conexión entre dos redes que usan distintas codificaciones y hace la traducción a medida que pasa una llamada entre ellas
Puerta de enlace de señalización	Conecta dos redes que usan distintos mecanismos de señalización y traduce las solicitudes y respuestas de la administración de llamadas

Figura 28.4 Los cuatro bloques de construcción fundamentales de un sistema de telefonía IP.

Un *teléfono IP* se conecta a una red, usa IP para toda la comunicación y ofrece una interfaz telefónica tradicional que permite a un usuario hacer o recibir llamadas telefónicas. Un teléfono IP puede ser una unidad de hardware independiente (es decir, un teléfono convencional) o puede consistir en una computadora con un micrófono, bocina y software de telefonía IP. La conexión entre un teléfono IP y el resto del mundo puede consistir en una red alámbrica o inalámbrica (por ejemplo, Ethernet[†] o 802.11b).

Un *controlador de puerta de enlace de medios*, que también se conoce como *gatekeeper* o *softswitch*, proporciona el control y la coordinación generales entre un par de teléfonos IP, lo que permite a quien hace la llamada localizar a alguien que la reciba o acceder a los servicios como el reenvío de llamadas.

Una *puerta de enlace de medios* proporciona la traducción de audio a medida que pasa una llamada a través del límite entre una red IP y la PSTN, o el límite entre dos redes IP que usen distintas codificaciones. Por ejemplo, una puerta de enlace de medios en el límite entre la PSTN e Internet desplaza audio digitalizado entre la codificación TDM que se utiliza en un circuito de voz convencional y la codificación de paquetes que se usa en Internet.

[†] Es posible usar la *alimentación a través de Ethernet (PoE)* para suministrar energía a un teléfono IP a través del cable de Ethernet que se usa para los datos.

Una *puerta de enlace de señalización* también abarca el límite entre un par de redes dispares y proporciona la traducción de operaciones de señalización, permitiendo que cualquiera de los lados inicie una llamada (por ejemplo, para permitir que un teléfono IP en Internet haga una llamada a un teléfono en la PSTN). Un controlador de puerta de enlace de medios coordina la operación de los medios y las puertas de enlace de señalización. La figura 28.5 ilustra la forma en que se usan los componentes para interconectar a Internet y la PSTN.

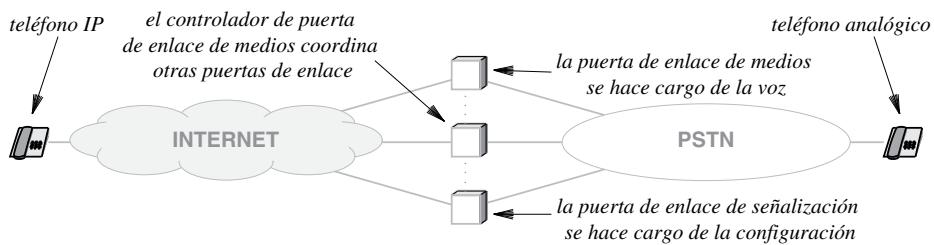


Figura 28.5 Conexiones entre los componentes de la telefonía IP.

Los conceptos y la terminología antes descritos presentan una versión directa y algo simplificada de la telefonía IP que se deriva del trabajo en el IETF y la ITU, así como de *Megaco* y del *protocolo de control de puerta de enlace de medios (MGCP)*. Las implementaciones prácticas del servicio de telefonía IP son más complejas. Las siguientes secciones proporcionan ejemplos.

28.8.1 Terminología de SIP y conceptos

El *protocolo de iniciación de sesión (SIP)* minimiza la necesidad de protocolos adicionales mediante el uso de protocolos existentes siempre que sea posible. Por ejemplo, SIP usa el sistema de nombres de dominio para asignar un número telefónico a una dirección IP. Como resultado, SIP define tres nuevos elementos que constituyen un sistema de señalización:

- Agente de usuario
- Servidor de ubicación
- Servidores de soporte (proxy, redirecciónamiento, registrador)

Agente de usuario. Los documentos del SIP se refieren a un dispositivo que realiza o termina llamadas telefónicas como un *agente de usuario*. Es posible implementar un agente de usuario de SIP en un teléfono IP, una computadora tipo laptop o una puerta de enlace de PSTN que permita a un teléfono IP hacer llamadas a la PSTN. Un agente de usuario contiene dos partes: un *cliente de agente de usuario* que hace llamadas salientes y un *servidor de agente de usuario* que se encarga de las llamadas entrantes.

Servidor de ubicación. Un servidor de ubicación de SIP administra una base de datos de la información sobre cada usuario, como podría ser un conjunto de direcciones IP para ese usuario, los servicios a los que está suscrito y sus preferencias de configuración. Se hace contacto con el servidor de ubicación durante la configuración de la llamada para obtener información sobre la ubicación o ubicaciones que aceptarán cierta llamada.

Servidor proxy. SIP incluye el concepto de un *proxy* que puede reenviar solicitudes de agentes de usuario a otra ubicación. Los servidores proxy manejan el enrutamiento óptimo e implementan políticas (por ejemplo, asegurarse de que quien hace la llamada tenga la autorización de hacerla).

Servidor de redireccionamiento. SIP usa un servidor de redireccionamiento para manejar tareas tales como el reenvío de llamadas y las conexiones a números 800. El servidor de redireccionamiento recibe una solicitud de un agente de usuario y devuelve una ubicación alternativa para que el agente de usuario se contacte.

Servidor registrador. SIP usa un servidor registrador para recibir solicitudes de registro y actualizar la base de datos que consultan los servidores de ubicación. Un registrador es responsable de autenticar las solicitudes de registro y asegurar que la base de datos se mantenga consistente.

28.8.2 Terminología de H.323 y conceptos

El estándar H.323 de la ITU, que define terminología alternativa y conceptos adicionales, se enfoca en la interacción con la PSTN. Aunque es muy extensa y cubre muchos detalles, H.323 puede resumirse de la siguiente manera:

- *Terminal.* Una terminal H.323 proporciona la función de telefonía IP, que también puede incluir herramientas para la transmisión de video y de datos.
- *Controlador de acceso o gatekeeper.* Un controlador de acceso H.323 proporciona funciones de ubicación y de señalización; además coordina la operación de la puerta de enlace que proporciona una conexión a la PSTN.
- *Puerta de enlace o gateway.* H.323 usa una sola puerta de enlace para interconectar el sistema de telefonía IP con la PSTN; la puerta de enlace maneja tanto la señalización como la traducción de medios.
- *Unidad de control multipunto (MCU).* Una MCU proporciona servicios como la conferencia multipunto.

28.8.3 Terminología de ISC y conceptos

Debido a que la ITU y el IETF generaron diversas variantes de la terminología y los conceptos, los distribuidores formaron el *Consorcio internacional Softswitch (ISC)* para crear un modelo funcional uniforme y completo que incorporara todos los modelos de la telefonía IP en un solo marco de trabajo. Para ello, el ISC definió la funcionalidad que puede ser necesaria, incluyendo la señalización entre varios tipos de sistemas, la traducción de las codificaciones, el soporte de servicios como el reenvío de llamadas, así como las funciones de administración como la contabilidad y la facturación. Entonces el ISC definió una lista de funciones suficientes para todas las situaciones:

- *Función de controlador de puerta de enlace de medios (MGC-F).* La MGC-F mantiene la información de estado en los puntos terminales; proporciona la lógica y el control de las llamadas.
- *Función de agente de llamadas (CA-F).* La CA-F es un subconjunto de la MGC-F que mantiene el estado de las llamadas. Algunos ejemplos de CA-F son SIP, H.323 y Q.931.

- *Función de interfuncionamiento (IW-F)*. La IW-F es un subconjunto de la MGC-F que maneja la señalización entre redes heterogéneas tales como SS7 y SIP.
- *Función de enrutamiento y función de contabilidad (R-F/A-F)*. La R-F maneja el enrutamiento de llamadas para la MGC-F y la A-F recopila la información utilizada para contabilidad y facturación.
- *Función de puerta de enlace de señalización (SG-F)*. La SG-F maneja la señalización entre una red IP y la PSTN.
- *Función de señalización de puerta de enlace de acceso (AGS-F)*. La AGS-F maneja la señalización entre una red IP y una red de acceso de conmutación de circuitos como la PSTN.
- *Función de servidor de aplicaciones (AS-F)*. La AS-F maneja un conjunto de servicios de aplicación como el correo de voz.
- *Función de control de servicio (SC-F)*. La SC-F se invoca cuando una AS-F debe controlar o cambiar la lógica de un servicio (por ejemplo, crear una nueva correlación).
- *Función de puerta de enlace de medios (MG-F)*. La MG-F se hace cargo de la traducción de audio digitalizado entre dos formas, y también puede incluir la detección de eventos como cuando un teléfono está descolgado, así como el reconocimiento de señales de *multifrecuencia de tono dual (DTMF)*, que es el estándar de señalización de audio que se conoce como codificación de *tonos*.
- *Función de servidor de medios (MS-F)*. La MS-F manipula un flujo de paquetes de medios a beneficio de una aplicación AS-F.

28.9 Resumen de protocolos y distribución en capas

Puesto que varios grupos han propuesto protocolos para la telefonía IP, en la mayoría de las capas de la pila de protocolos existen algunos que compiten. La figura 28.6 enumera algunos de los protocolos propuestos, junto con su posición en el modelo de referencia de 5 capas de Internet.

Capa	Proceso de llamada	Multimedia de usuario	Datos de usuario	Soporte	Enrutamiento	Transporte de señal
5	H.323 Megaco MGCP SIP	RTP	T.120	RTCP RTSP NTP SDP	ENUM TRIP	SIGTRAN [†]
4	TCP UDP	UDP	TCP	TCP UDP		SCTP
3	IP, RSVP e IGMP					

Figura 28.6 Un resumen de los protocolos de telefonía IP.

[†] SIGTRAN permite la transferencia de señales de PSTN (por ejemplo, SS7, DTMF) a través de una red IP; SCTP multiplexa varios flujos de entrada sobre un solo flujo de la capa de transporte.

28.10 Características de H.323

En vez de un solo protocolo, el estándar *H.323* creado por la ITU consiste en un grupo de protocolos que funcionan en conjunto para manejar todos los aspectos de la comunicación telefónica. Los puntos destacados de H.323 son:

- Maneja todos los aspectos de una llamada telefónica digital.
- Incluye la señalización para configurar la llamada y administrarla.
- Permite la transmisión de video y datos durante una llamada en progreso.
- Envía mensajes binarios que se definen mediante *ASN.1* y se codifican usando las *reglas básicas de codificación (BER)*.
- Incorpora protocolos para la seguridad.
- Usa una unidad de hardware especial conocida como *unidad de control multipunto* para dar soporte a llamadas de conferencias.
- Define a los servidores para manejar tareas tales como la *resolución de direcciones* (es decir, asignar el número telefónico de la parte que recibe la llamada a una dirección IP), la *autenticación*, la *autorización* (es decir, determinar si se permite a un usuario el acceso a un servicio dado), la *contabilidad* y otras características, como el reenvío de llamadas.

28.11 Distribución en capas de H.323

Los protocolos H.323 usan TCP y UDP para el transporte: el audio puede viajar a través de UDP, mientras que una transferencia de datos lo hace a través de TCP. La figura 28.7 ilustra la distribución básica en capas en el estándar H.323.

Capa	Señalización	Registro	Audio	Video	Datos	Seguridad		
5	H.225.0-Q.931 H.250-Anexo G H.245 H.250	H.225.9-RAS	G.711 H.263 G.722 G.723 G.728	H.261 H.323	T.120	H.235		
			RTP, RTCP					
4	TCP, UDP	UDP			TCP	TCP, UDP		
3	IP, RVSP e IGMP							

Figura 28.7 La distribución por capas de los principales protocolos en el estándar H.323.

28.12 Características y métodos de SIP

Los puntos destacados del *protocolo de iniciación de sesión (SIP)* del IETF son:

- Opera en la capa de aplicación
- Abarca todos los aspectos de la señalización, incluyendo la ubicación de la parte que recibe una llamada, la notificación y la configuración (es decir, hacer sonar un teléfono), la disponibilidad (es decir, si la parte acepta o no la llamada) y la terminación
- Proporciona servicios tales como el reenvío de llamadas
- Depende de la multidifusión para las llamadas de conferencia
- Permite que los dos lados negocien las capacidades y seleccionen los medios y los parámetros a usar[†]

Un URI de SIP contiene el nombre de un usuario y el nombre de dominio en el que es posible encontrar al usuario. Por ejemplo, a un usuario llamado *Smith* que trabaja en *Unaempresa, Inc.* se le podría asignar el siguiente URI de SIP:

sip:smith@unaempresa.com

SIP define seis tipos de mensajes básicos y siete extensiones. Los tipos de mensajes básicos se conocen como *métodos*. La figura 28.8 enumera los métodos básicos de SIP.

Método	Propósito
INVITE	Creación de sesiones: se invita a un punto terminal a participar en la sesión
ACK	Respuesta de reconocimiento para INVITE
BYE	Terminación de sesión: termina la llamada
CANCEL	Cancelación de solicitud pendiente (no tiene efecto si se completó la solicitud)
REGISTER	Registro de la ubicación del usuario (es decir, un URL donde se pueda contactar al usuario)
OPCIONES	Consulta para determinar las posibilidades de la parte que recibe la llamada

Figura 28.8 Los seis métodos básicos utilizados por SIP.

[†] SIP usa el *protocolo de descripción de sesiones (SDP)* para describir posibilidades y parámetros.

28.13 Ejemplo de una sesión de SIP

Un ejemplo de los mensajes enviados durante una sesión SIP aclarará parte de los detalles y ayudará a explicar la idea general detrás de la mayor parte de la telefonía IP. La figura 28.9 enlista una secuencia de mensajes que se envían cuando un agente de usuario A hace contacto con un servidor DNS y luego se comunica con un servidor proxy, quien a su vez invoca a un servidor de ubicación.[†] Una vez que se establece la llamada, los dos teléfonos IP se comunican directamente. Por último, SIP se usa para terminar la llamada.

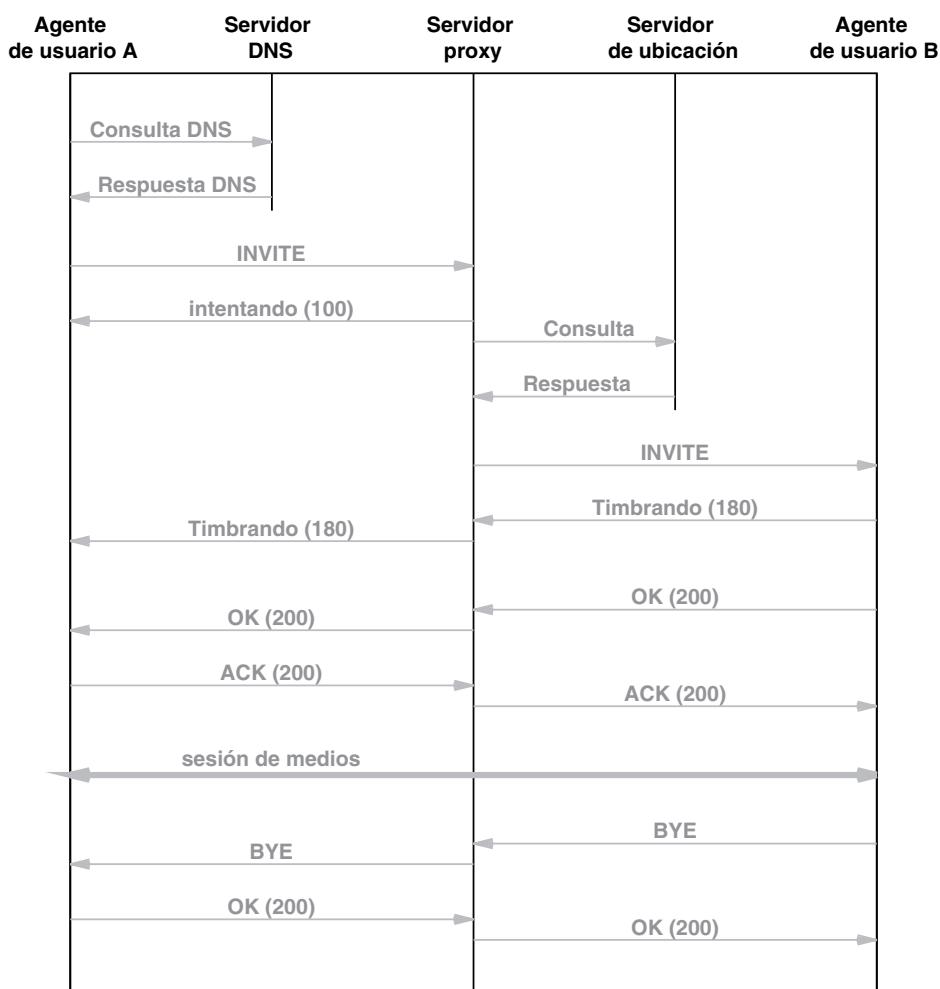


Figura 28.9 Un ejemplo de los mensajes intercambiados por SIP para administrar una llamada telefónica.

[†] En la práctica, SIP soporta la *bifurcación de llamadas*, lo que permite a un servidor de ubicación devolver varias ubicaciones para un usuario (por ejemplo, hogar y oficina), para que un agente de usuario intente el contacto simultáneo.

Por lo general, un agente de usuario se configura con la dirección IP de uno o más servidores DNS (se usa para correlacionar el nombre de dominio de un URI de SIP y una dirección IP) y de uno o más servidores proxy. De manera similar, cada servidor proxy se configura con la dirección de uno o más servidores de ubicación. Por lo tanto, si un servidor dado no está disponible, SIP puede encontrar rápidamente una alternativa.

28.14 Asignación y enrutamiento de números telefónicos

¿Cómo se llama y localiza a los usuarios de teléfonos IP? La PSTN sigue el estándar E.164 de la ITU para números telefónicos y SIP usa direcciones IP. El problema de localizar a los usuarios es complicado, ya que puede haber varios tipos de redes involucradas. Por ejemplo, considere una red integrada que consiste en dos redes PSTN interconectadas por una red IP. Los diseñadores definen dos subproblemas: localizar a un usuario en la red integrada y encontrar una ruta eficiente hacia el usuario. El IETF propuso dos protocolos que corresponden a las asignaciones necesarias para los dos subproblemas.

- ENUM: convierte un número telefónico en un URI
- TRIP: encuentra a un usuario en una red integrada

ENUM. El protocolo *ENUM* (abreviación de *E.164 NUMbers*) del IETF resuelve el problema de convertir un número telefónico E.164 en un *identificador uniforme de recursos (URI)*. En esencia, ENUM usa el sistema de nombres de dominio para almacenar la asignación. Un número telefónico se convierte en un nombre de dominio especial en el dominio:

e164.arpa

La conversión consiste en tratar al número telefónico como una cadena, invertir la cadena y escribir dígitos individuales como segmentos de un nombre de dominio. Por ejemplo, el número telefónico 1-800-555-1234 produce el nombre de dominio:

4.3.2.1.5.5.0.0.8.1.e164.arpa

Una asignación ENUM puede ser de 1 a 1 como un esquema de numeración telefónica convencional, o puede ser de 1 a varios, lo que significa que al teléfono de escritorio y al teléfono móvil del usuario se les puede asignar el mismo número telefónico. Cuando un número corresponde a varios hosts, un servidor DNS devuelve una lista de los hosts junto con el protocolo utilizado para llegar a cada uno de ellos. Un agente de usuario procede a contactar a los hosts de la lista hasta que uno responda.

TRIP. El protocolo *enrutamiento telefónico sobre IP (TRIP)* del IETF resuelve el problema de encontrar a un usuario en una red integrada. Un servidor de ubicación u otro elemento de red pueden usar TRIP para anunciar las rutas. De esta forma, dos servidores de ubicación usan TRIP para informarse entre sí sobre las rutas externas que cada uno conoce. Puesto que es independiente del protocolo de señalización, TRIP puede usarse con SIP u otros mecanismos de señalización.

TRIP divide el mundo en un conjunto de *dominios administrativos de telefonía IP (ITAD)*. En esencia, un anuncio TRIP identifica un punto de egreso; es decir, un servidor de ubicación informa a otro sobre una ruta hacia una puerta de enlace de señalización que se interconecta con otro ITAD. Puesto que la telefonía IP es nueva y la información de enrutamiento puede cambiar en el futuro, TRIP está diseñado para ser extensible.

28.15 Resumen

El *protocolo de transporte en tiempo real* (RTP) adapta la transferencia de multimedia en tiempo real a través de Internet. Un mensaje RTP incluye un número de secuencia y una etiqueta de tiempo independiente, así como una identificación de los orígenes de datos. Un receptor usa la etiqueta de tiempo para colocar los datos en un búfer de fluctuación antes de la reproducción. RTP se encapsula en UDP para la transmisión, lo cual permite la multidifusión y la difusión. No se usa retransmisión debido a que los paquetes recibidos más allá de la ventana de reproducción no pueden reproducirse.

Los términos telefonía IP y VoIP se refieren a las llamadas telefónicas de voz digitalizadas que se transmiten a través de Internet. Uno de los mayores desafíos al crear un sistema de telefonía IP se debe a la retrocompatibilidad, ya que hay que inventar puertas de enlace que conecten un sistema de telefonía IP con la PSTN tradicional. Las puertas de enlace deben proporcionar tanto traducción de medios (es decir, la traducción entre codificaciones de voz digital) como señalización (es decir, la traducción de la configuración de las llamadas).

La ITU y el IETF crearon cada uno estándares para la telefonía IP. El estándar H.323 de la ITU incluye muchos protocolos que proporcionan configuración, administración, autorización y contabilidad de llamadas, así como otros servicios de usuario como el reenvío de llamadas y la transmisión de voz, video y datos a través de una llamada telefónica. El estándar SIP del IETF para la señalización es capaz de localizar a un usuario, configurar una llamada y negociar las capacidades para cada lado de una llamada. SIP usa un conjunto de servidores que manejan varios aspectos de la señalización, como el servidor de nombres de dominio, los servidores proxy y los servidores de ubicación. El Consorcio internacional Softswitch (ISC) definió un marco de trabajo adicional con la intención de abarcar todos los modelos de telefonía IP.

Hay dos protocolos adicionales del IETF que proporcionan funciones de soporte. ENUM usa el sistema de nombres de dominio para signar un número telefónico E.164 a un Identificador uniforme de recursos (por lo general un URI de SIP). TRIP proporciona enrutamiento entre los dominios administrativos de telefonía IP; un servidor de ubicación SIP puede usar TRIP para informar a otros servidores de ubicación sobre las puertas de enlace que forman puntos de egreso de la red.

EJERCICIOS

- 28.1** Defina los datos multimedia. ¿Cuáles son las dos técnicas que se utilizan para solucionar la fluctuación?
- 28.2** Explique cómo es que un búfer de fluctuación permite la reproducción de un flujo de audio, aun cuando Internet presente inestabilidad.

- 28.3** Si se intercepta un mensaje RTP a medida que viaja por Internet, ¿puede interpretarse el campo de la etiqueta de tiempo? De ser así, ¿cómo? Si no es así, ¿por qué no?
- 28.4** Puesto que viaja en UDP, un mensaje RTP podría duplicarse. ¿Necesita un receptor mantener una copia de todos los mensajes recibidos previamente para determinar si un mensaje entrante está duplicado? ¿Por qué sí o por qué no?
- 28.5** RTP contiene un protocolo complementario conocido como *protocolo de control en tiempo real (RTCP)*, el cual permite a un receptor informar a un emisor sobre la calidad de los mensajes recibidos. ¿Cómo puede la codificación de video adaptativa usar el estatus de los mensajes recibidos?
- 28.6** Si la voz se convierte en formato digital usando PCM, ¿cuántos bits de datos se producirán en medio segundo?
- 28.7** Extienda el ejercicio anterior. Calcule el tamaño (en octetos) de un datagrama IP que transporta un cuarto de segundo de audio codificado en PCM, colocado en un paquete RTP y encapsulado en UDP. (Sugerencia: RFC 1889 define el tamaño de un encabezado RTP).
- 28.8** ¿Qué aspectos de la telefonía IP maneja H.323?
- 28.9** Cuando se usa H.323 para enviar datos junto con audio o video, ¿qué protocolo de transporte se utiliza?
- 28.10** ¿Cuáles son los seis métodos básicos que se utilizan con SIP?
- 28.11** Lea el RFC sobre SIP y modifique la figura 28.9 para mostrar los mensajes intercambiados cuando ocurre el reenvío de llamadas. (Sugerencia: vea los mensajes de *redirección* de SIP).
- 28.12** ¿Cuáles son los propósitos de ENUM y TRIP?
- 28.13** Considere la operación de un teléfono IP y un teléfono analógico. ¿Cuál sería mejor durante tiempos de guerra? ¿Por qué?
- 28.14** Busque el dominio e164.arpa. ¿Qué organización es responsable del dominio?

Contenido del capítulo

- 29.1 Introducción, 507
- 29.2 Abusos y ataques criminales, 507
- 29.3 Políticas de seguridad, 511
- 29.4 Responsabilidad y control, 512
- 29.5 Tecnologías de seguridad, 513
- 29.6 Hashing: un mecanismo de integridad y autenticación, 513
- 29.7 Control de acceso y contraseñas, 514
- 29.8 Cifrado: una técnica de seguridad fundamental, 514
- 29.9 Cifrado de clave privada, 515
- 29.10 Cifrado de clave pública, 515
- 29.11 Autenticación con firmas digitales, 516
- 29.12 Autoridades de claves y certificados digitales, 517
- 29.13 Firewall, 519
- 29.14 Implementación de un firewall con un filtro de paquetes, 520
- 29.15 Sistemas de detección de intrusos, 522
- 29.16 Exploración de contenido e inspección detallada de paquetes, 522
- 29.17 Redes privadas virtuales (VPN), 523
- 29.18 El uso de la tecnología VPN para el trabajo a distancia, 525
- 29.19 Comparación entre cifrado de paquetes y uso de túneles, 526
- 29.20 Tecnologías de seguridad, 528
- 29.21 Resumen, 529

29

Seguridad de redes

29.1 Introducción

Los capítulos anteriores describen los sistemas de hardware y software que constituyen Internet. Además explican cómo es que las aplicaciones cliente y servidor usan las herramientas de Internet para comunicarse. Este capítulo considera el importante aspecto de la seguridad de las redes. El capítulo describe los tipos de delitos que se han perpetrado a través de Internet, describe los aspectos clave de la seguridad y explica las tecnologías que se utilizan para incrementar la seguridad de las redes.

29.2 Abusos y ataques criminales

Cada vez que aparece una nueva tecnología, los criminales se preguntan cómo pueden aprovecharla para cometer delitos. Internet no es la excepción; como la mayoría de los usuarios saben, ahora los criminales usan Internet a diario. Aunque los delitos de Internet como los fraudes y el robo de identidad pueden afectar a las personas, los delitos más considerables imponen una amenaza para los negocios. Además del robo descarado de bienes o servicios, a las empresas les preocupan en especial las amenazas a su viabilidad a largo plazo. Por lo tanto, el daño a la reputación, la pérdida de confianza de los clientes, el robo de propiedad intelectual y la prevención de acceso de los clientes son todas cuestiones importantes para una empresa.

Surgen varias preguntas relacionadas con la seguridad:

- ¿Cuáles son los principales problemas y amenazas de seguridad de Internet?
- ¿Qué aspectos técnicos de los protocolos explotan los criminales?
- ¿Cuáles son los aspectos clave de la seguridad?
- ¿Qué tecnologías hay disponibles para ayudar a incrementar la seguridad?

En la figura 29.1 se resumen algunos de los principales problemas de seguridad que existen en Internet.

Problema	Descripción
Phishing o suplantación de identidad	Disfrazarse como un sitio reconocido, tal como un banco, para obtener la información personal de un usuario; por lo general es un número de cuenta y el código de acceso
Tergiversación	Hacer afirmaciones falsas o exageradas sobre productos o servicios, u ofrecer productos falsos o de calidad inferior
Fraudes	Varias formas de artimañas con la intención de engañar a los usuarios ingenuos para que inviertan dinero o sean cómplices de un delito
Negación de servicio	Bloquear de manera intencional un sitio de Internet específico para evitar u obstaculizar las actividades de negocios y el comercio
Pérdida de control	Un intruso gana el control de la computadora de un usuario y la utiliza para perpetrar un delito
Pérdida de datos	Pérdida de propiedad intelectual o demás información de negocios valiosa

Figura 29.1 Principales problemas de seguridad que predominan en Internet.

Al considerar la seguridad, es importante diferenciar entre un delito convencional que se comete utilizando Internet de manera incidental y un delito específico de Internet. Por ejemplo, considere un delito en el que un criminal usa un teléfono VoIP para comunicarse con un cómplice o un incidente en el que un criminal usa Internet para comprar herramientas que se utilizan para cometer un delito. Aunque las autoridades competentes deben encargarse de ellos, dichos delitos tienen poco que ver con las tecnologías de redes y podríamos encontrar con facilidad mecanismos de comunicación alternativos que puedan usarse en vez de Internet. Dos de los delitos más extendidos que se experimentan en Internet son delitos convencionales que usan Internet por casualidad: la tergiversación de los productos que se ofrecen para venta (por ejemplo, en un sitio de subastas en línea) es una forma de publicidad falsa, y el no entregar los productos comprados a través de una subasta es algo similar al fraude convencional de pedidos por correo.

Nuestra explicación se enfocará en dos aspectos del crimen por Internet. Primero examinaremos las formas en que los criminales abusan de las tecnologías de redes. Segundo, consideraremos las técnicas y tecnologías que se crearon para hacer que el delito sea más difícil o costoso. La figura 29.2 enumera las técnicas específicas que usan los atacantes.

Técnica	Descripción
Intercepción electrónica o <i>wiretapping</i>	Crear una copia de los paquetes a medida que recorren una red para obtener información
Reproducción duplicada	Enviar los paquetes capturados de una sesión anterior (por ejemplo, un paquete de contraseña de un inicio de sesión)
Desbordamiento de búfer	Enviar más datos de los que espera un receptor para almacenar los valores en variables más allá del búfer
Falsificación o <i>spoofing</i> de direcciones	Falsificar la dirección IP de origen en un paquete para engañar a un receptor de modo que procese el paquete
Falsificación de nombres	Usar la escritura incorrecta de un nombre reconocido o envenenar un servidor de nombres con una vinculación incorrecta
DoS y DDoS	Inundar un sitio con paquetes para evitar que realice con éxito sus operaciones de negocios normales
Inundación SYN	Enviar un flujo de segmentos aleatorios SYN de TCP para agotar el conjunto de conexiones TCP de un receptor
Descifrado de contraseñas	Sistemas automatizados que adivinan una contraseña o una clave de descifrado o que obtienen acceso sin autorización
Escaneo de puertos	Tratar de conectarse a todos los puertos de protocolos posibles en un host para encontrar una vulnerabilidad
Intercepción de paquetes	Quitar un paquete de Internet para permitir la sustitución y los ataques de intermediario (<i>man-in-the-middle</i>)

Figura 29.2 Técnicas utilizadas en los ataques de seguridad.

La *intercepción electrónica* y la técnica relacionada de *reproducción duplicada* son obvias. Hay un caso especial de reproducción duplicada que no se relaciona en lo absoluto con las redes: un atacante instala software o un dispositivo para registrar las pulsaciones de teclas. Cuando un usuario introduce su contraseña o NIP, el registrador o *logger* registra cada tecla presionada y el atacante puede introducir la misma secuencia de pulsaciones de teclas para obtener acceso posteriormente.

El *desbordamiento del búfer* es una de las debilidades más explotadas de un sistema de computadora. Es un síntoma de mala ingeniería cuando un programador se olvida de revisar el tamaño del búfer al realizar una operación de entrada. Un ataque común envía un paquete jumbo (más grande de lo que el estándar permite) o envía una secuencia de paquetes, uno tras otro, que desbordan un búfer de entrada. Sólo el distribuidor que creó el software puede corregir el problema.

Los ataques de *falsificación* o *spoofing* se utilizan para hacerse pasar como un host de confianza. La forma más simple de falsificación de dirección usa ARP: un atacante difunde una solicitud ARP que vincula una dirección IP arbitraria, A, con la dirección MAC del atacante. Cuando un host en la red envía un paquete a A, el paquete se reenviará en realidad al atacante. Otras formas de spoofing implican el uso de un protocolo de enrutamiento para enviar rutas incorrectas, enviando un mensaje DNS que almacene una vinculación incorrecta en un servidor DNS y escribiendo de manera un poco incorrecta un nombre de dominio reconocido para dar a un usuario la impresión de que llegaron a un sitio Web de confianza. Por ejemplo, un ataque de spoofing utilizó *banksfamerica.com* para enviar correo electrónico a clientes del banco sin que se dieran cuenta que se escribió *banks* en plural, en vez del nombre legítimo con *bank*.

Un ataque de *negación de servicio* (*DoS*) inunda un host (por lo general, un servidor web) con un flujo de paquetes. Aunque el servidor sigue funcionando, el ataque consume todos los recursos, lo que significa que la mayoría de los usuarios que intenten usar el sitio experimentarán retrasos extensos o se rechazarán sus conexiones. Puesto que un administrador puede detectar y deshabilitar un flujo de paquetes de un solo origen, un ataque de *negación de servicio distribuida* (*DDoS*) dispone que un conjunto grande de hosts esparcidos por Internet envíen cada uno un flujo de paquetes, como se ilustra en la figura 29.3. Por lo general, un atacante primero expropia hosts en Internet, carga software en ellos y luego los utiliza para atacar un servidor. Por consiguiente, ninguno de los paquetes enviados por un DDoS vienen directamente de la computadora del atacante.

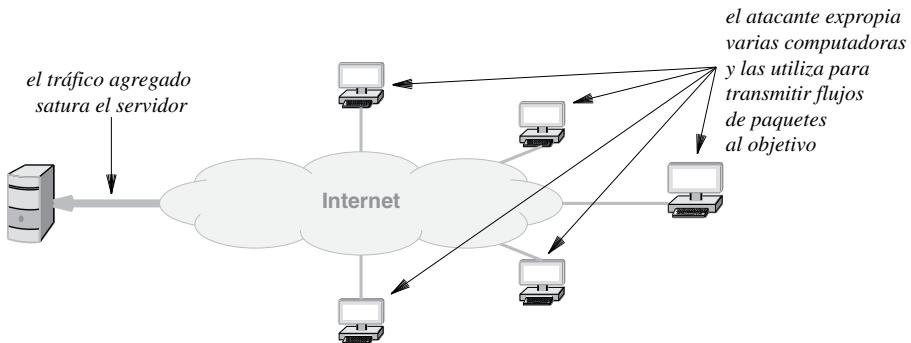


Figura 29.3 Ilustración de un ataque de negación de servicio distribuida.

La *inundación SYN* es una técnica específica que se utiliza para negar el servicio a TCP: cada paquete entrante contiene un mensaje *SYN* de TCP que solicita una nueva conexión de TCP. Un receptor asigna un bloque de control de TCP para la conexión, envía un mensaje *SYN + ACK* y espera una respuesta. En un momento dado se asignan todos los bloques de control y no es posible abrir más conexiones.

La *intercepción de paquetes* hace posible lanzar ataques llamados *man-in-the-middle* donde un intermediario puede modificar paquetes a medida que pasen del origen al destino. Aunque está entre los ataques más difíciles de orquestar, la intercepción de paquetes tiene el mayor potencial de daño, como se ilustra en la figura 29.4.

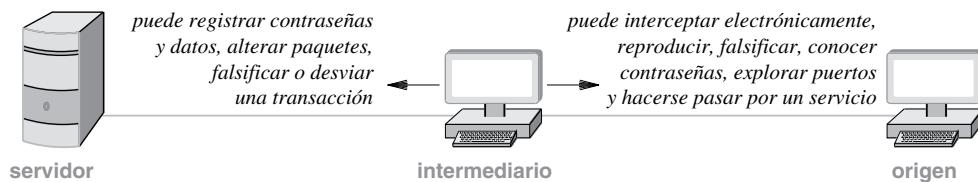


Figura 29.4 Una configuración tipo intermediario y los ataques que permite.

29.3 Políticas de seguridad

¿Qué es una red segura? Aunque el concepto de red segura atrae a la mayoría de los usuarios, las redes no pueden clasificarse simplemente como seguras o no seguras, ya que el término no es absoluto y cada organización define el nivel de acceso que se permite o rechaza. Por ejemplo, tal vez una organización que guarda secretos comerciales valiosos tenga que evitar que personas externas accedan a sus computadoras. Una organización con un sitio Web que tenga información disponible puede definir una red segura como una que permita el acceso arbitrario a los datos, pero que incluya mecanismos para evitar que los externos modifiquen esos datos. Otras organizaciones se enfocan en mantener la información confidencial y definen una red segura como una en la que nadie más que el emisor o el receptor destinado pueden interceptar y leer un mensaje. Por último, una organización de gran tamaño puede necesitar una definición compleja de la seguridad que permita el acceso a los datos o servicios seleccionados y evite el acceso o la modificación de datos y servicios sensibles o confidenciales.

Puesto que no existe una definición absoluta de *red segura*, el primer paso que debe tomar una organización para lograr un sistema seguro es definir la *política de seguridad* de la organización. La política no especifica cómo lograr la protección. Más bien indica con claridad y sin ambigüedades los elementos que deben protegerse.

Las políticas de seguridad son complejas debido a que involucran el comportamiento humano así como la instalación de computadoras y redes (por ejemplo, un visitante que saca una memoria tipo Flash ROM de una organización, una red inalámbrica que puede detectarse fuera de un edificio o los empleados que trabajan desde casa). La evaluación de los costos y beneficios de varias políticas de seguridad también añade complejidad. En especial, no puede definirse una política de seguridad a menos que una organización comprenda el valor de su información. En muchos casos el valor de la información es difícil de evaluar. Considere por ejemplo una simple base de datos de nómina que contiene un registro para cada empleado, las horas que trabajó ese empleado y la tarifa de su sueldo. Si pudieran acceder a esa información, algunos empleados podrían molestarse y exigir sueldos más altos o amenazar con renunciar. Si los competidores obtuvieran la información, podrían usarla para alejar a los empleados de la organización. Lo que es más importante, un competidor podría usar la información en formas inesperadas (por ejemplo, para evaluar el esfuerzo invertido en un proyecto específico). Para resumir:

Idear una política de seguridad de red puede ser complejo, ya que una política racional requiere que una organización relacione la seguridad de redes y computadoras con el comportamiento humano y evalúe el valor de la información.

También es complicado definir una política de seguridad, ya que cada organización debe decidir qué aspectos de la protección son más importantes y a menudo hay que hacer un compromiso entre la seguridad y la facilidad de uso. Por ejemplo, una organización puede considerar lo siguiente:

- *Integridad de los datos.* La integridad se refiere a la protección contra una alteración: ¿los datos que llegan a un receptor son idénticos a los que se enviaron?
- *Disponibilidad de los datos.* La disponibilidad se refiere a la protección contra la interrupción del servicio: ¿permanecen los datos accesibles para usos legítimos?
- *Confidencialidad de los datos.* La confidencialidad se refiere a la protección contra el acceso no autorizado a los datos (por ejemplo, mediante falsificación o intercepción electrónica): ¿los datos están protegidos contra el acceso no autorizado?
- *Privacidad.* La privacidad se refiere a la habilidad de un emisor de permanecer anónimo: ¿se revela la identidad del emisor?

29.4 Responsabilidad y control

Además de los puntos anteriores, una organización debe especificar con exactitud cómo se asigna y controla la responsabilidad de la información. La cuestión de la responsabilidad de la información tiene dos aspectos:

- *Rendición de cuentas.* Se refiere a la forma en que se mantiene un registro de auditoría: ¿qué grupo es responsable de cada elemento de datos? ¿Cómo guarda el grupo los registros de acceso y modificación?
- *Autorización.* Se refiere a la responsabilidad de cada elemento de información y cómo se delega dicha responsabilidad a otros: ¿quién es responsable del lugar en donde reside la información y cómo aprueba una persona responsable el acceso y la modificación?

La cuestión crítica detrás de la rendición de cuentas y la autorización, es el *control*. Una organización debe controlar el acceso a la información de manera similar a como controla el acceso a los recursos físicos tales como oficinas, equipo y suministros. Un aspecto clave del control se relaciona con la *autenticación*, que se refiere a la validación de la identidad. Por ejemplo, suponga que una organización determina una política de autorización que otorga a un empleado más privilegio que a un visitante. La política de autorización carece de significado a menos que la organización tenga un mecanismo de autenticación que distinga entre un visitante y un empleado. La autenticación se extiende más allá del ser humano para incluir computadoras, dispositivos y programas de aplicación. En conclusión:

Las políticas de autorización carecen de significado sin un mecanismo de autenticación que pueda verificar sin ambigüedad la identidad de un solicitante.

29.5 Tecnologías de seguridad

Existen muchos productos de seguridad que realizan una variedad de funciones tanto para computadoras individuales como para grupos de computadoras dentro de una organización. La figura 29.5 sintetiza las técnicas que utilizan dichos productos. Las siguientes secciones explican cada una de las tecnologías.

Técnica	Propósito
Hashing	Integridad de datos
Cifrado	Privacidad
Firmas digitales	Autenticación de mensajes
Certificados digitales	Autenticación del emisor
Firewalls	Integridad del sitio
Sistemas de detección de intrusos	Integridad del sitio
Escaneo de contenido e inspección detallada de paquetes	Integridad del sitio
Redes privadas virtuales (VPN)	Confidencialidad de los datos

Figura 29.5 Las principales técnicas que se utilizan para implementar políticas de seguridad.

29.6 Hashing: un mecanismo de integridad y autenticación

Los capítulos anteriores describen técnicas como *bits de paridad*, *sumas de verificación* y *comprobaciones de redundancia cíclica (CRC)* que protegen los datos contra daños accidentales. Dichas técnicas no proveen integridad de los datos por dos razones. Primero, una falla puede cambiar una suma de verificación así como el valor de los datos, lo que significa que la suma de verificación alterada puede ser válida para los datos alterados. Segundo, si se modifican datos como resultado de un ataque planeado, el atacante puede crear una suma de verificación válida para los datos alterados. Por lo tanto se crearon mecanismos adicionales para garantizar la integridad de los mensajes contra el cambio intencional.

Un método utilizado por los estándares *MD5* y *SHA-1* ofrece un *código de autenticación de mensajes (MAC)* que un atacante no puede descifrar ni falsificar. Los esquemas de codificación comunes usan mecanismos de *hashing criptográfico*. Un esquema de hashing se basa en una *clave secreta* que sólo el emisor y el receptor conocen. El emisor toma un mensaje como entrada, usa la clave para calcular un

hash, H , y lo transmite junto con el mensaje. H es una cadena corta de bits y su longitud es independiente del tamaño del mensaje. El receptor usa la clave para calcular un hash del mensaje y lo compara con H . Si los dos concuerdan, el mensaje llegó intacto. Un atacante (que no tiene la clave secreta) no podrá modificar el mensaje sin introducir un error. De esta forma, H ofrece autenticación de mensajes ya que un receptor sabe que un mensaje que llega con un hash válido es auténtico.

29.7 Control de acceso y contraseñas

Un mecanismo de *control de acceso* controla qué usuarios o programas de aplicación pueden acceder a los datos. Por ejemplo, algunos sistemas operativos implementan una *lista de control de acceso (ACL)* para cada objeto, la cual especifica a quién se le permite acceder a ese objeto. En otros sistemas, a cada usuario se le asigna una *contraseña* para cada recurso protegido. Cuando un usuario necesita acceder a un recurso protegido, a éste se le pide que introduzca la contraseña.

Al extender las listas de control de acceso y las contraseñas a través de una red, hay que llevar a cabo pasos para evitar la divulgación no intencional. Por ejemplo, si un usuario en una ubicación envía una contraseña sin cifrar a través de una red hacia una computadora en otra ubicación, alguien que intercepte electrónicamente la red puede obtener una copia de la contraseña. La intercepción electrónica es bastante fácil cuando los paquetes viajan a través de una LAN inalámbrica debido a que no se requiere una conexión física; cualquiera dentro del rango de la transmisión puede capturar una copia de cada paquete. Además, hay que realizar ciertos pasos para asegurar que las contraseñas no sean fáciles de adivinar, debido a que una red permite que un atacante automatice los intentos de descifrar una contraseña. Por lo tanto, los administradores implementan reglas para elegir contraseñas, como la longitud mínima y la prohibición del uso de palabras comunes (como las de un diccionario).

29.8 Cifrado: una técnica de seguridad fundamental

La *criptografía* es una herramienta fundamental en la seguridad, ya que el *cifrado* o *encriptación* puede garantizar tanto la confidencialidad (o *privacidad*) como la integridad de los datos, así como garantizar la autenticidad de los mensajes, y evitar ataques de reproducción duplicada. En esencia, un emisor aplica el cifrado para codificar los bits del mensaje de tal forma que sólo el receptor destinado pueda descifrarlos. Alguien que intercepte una copia de un mensaje cifrado no podrá extraer la información. Además, un mensaje cifrado puede incluir información tal como la longitud del mensaje, por lo que un atacante no puede truncar el mensaje sin ser descubierto.

La terminología que se utiliza con el cifrado define cuatro elementos:

- Texto simple: un mensaje original antes de cifrarlo
- Criptograma o texto cifrado: un mensaje después de ser cifrado
- Clave de cifrado: una cadena corta de bits utilizada para cifrar un mensaje
- Clave de descifrado: una cadena corta de bits que se utiliza para descifrar un mensaje

Como veremos, en algunas tecnologías la clave de cifrado y la de descifrado son idénticas; en otras son diferentes.

En sentido matemático, pensamos en el cifrado como una función, *cifrar*, que recibe dos argumentos: una clave K_1 y un mensaje de texto simple a cifrar, M . La función produce una versión cifrada del mensaje, el criptograma C :

$$C = \text{cifrar}(K_1, M)$$

Una función *descifrar* invierte la asignación para producir el mensaje original:[†]

$$M = \text{descifrar}(K_2, C)$$

En sentido matemático, *descifrar* es el inverso de *cifrar*:

$$M = \text{descifrar}(K_2, \text{cifrar}(K_1, M))$$

Existen muchas tecnologías de cifrado y pueden dividirse en dos categorías amplias que se definen por la manera en que usan las claves:

- Clave privada
- Clave pública

29.9 Cifrado de clave privada

En un sistema de *clave privada*, como el estándar *DES*, cada par de entidades de comunicación comparten una sola clave que sirve como *clave de cifrado* y como *clave de descifrado*. Surge el nombre debido a que la clave debe mantenerse secreta; si un tercero obtiene una copia de la clave, el tercero podrá descifrar los mensajes que pasan entre el par original. Los sistemas de clave privada son *simétricos* en cuanto a que cada lado puede enviar o recibir mensajes. Para enviar un mensaje, la clave se utiliza para producir un criptograma o texto cifrado, que a su vez se envía a través de una red. Cuando llega un mensaje, el lado receptor usa la clave secreta para decodificar el criptograma y extraer el mensaje original (texto simple). De esta forma, en un sistema de clave privada, un emisor y un receptor pueden usar la misma clave K , lo que significa que:

$$M = \text{descifrar}(K, \text{cifrar}(K, M))$$

29.10 Cifrado de clave pública

La principal alternativa al cifrado de clave privada se conoce como *cifrado de clave pública*, y es usado por el estándar *RSA*. Un sistema de clave pública asigna a cada entidad un par de claves. Para fines de esta explicación, asumiremos que cada entidad es un solo usuario. Una de las claves de usuario (que se conoce como *clave privada*) se mantiene en secreto, mientras que la otra (que se conoce como *clave pública*) se publica junto con el nombre del usuario, por lo que todos conocen el valor de la clave. La función de cifrado tiene la propiedad matemática de que un mensaje de texto simple cifrado con la clave pública no puede descifrarse a menos que sea con la clave privada, y un mensaje de texto simple cifrado con la clave privada no puede descifrarse a menos que sea con la clave pública.

[†] El descifrado puede o no usar la misma clave que el cifrado.

La relación entre el cifrado y el descifrado con las dos claves puede expresarse en forma matemática. Suponga que M denota un mensaje de texto simple, que $u1_publica$ denota la clave pública del usuario 1 y que $u1_privada$ denota la clave privada del usuario 1. Las funciones de cifrado pueden expresarse así:

$$M = \text{descifrar}(u1_publica, \text{cifrar}(u1_privada, M))$$

y

$$M = \text{descifrar}(u1_privada, \text{cifrar}(u1_publica, M))$$

La figura 29.6 ilustra por qué un sistema de clave pública se clasifica como *asimétrico* al mostrar las claves que se utilizan para cifrar los mensajes que se envían en cada dirección.

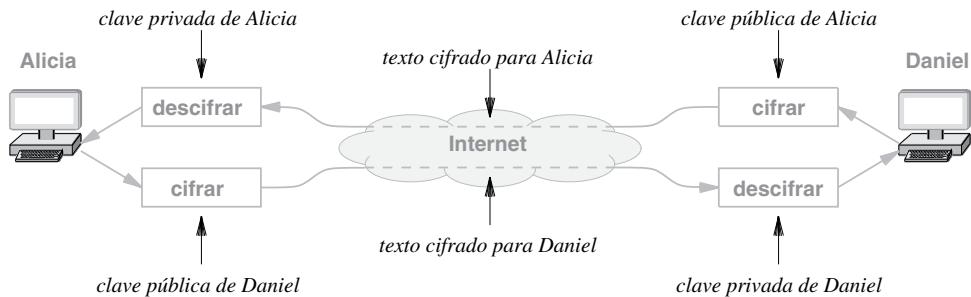


Figura 29.6 Ilustración de la asimetría en un sistema de cifrado de clave pública.

Es seguro revelar una clave pública ya que las funciones utilizadas para el cifrado y el descifrado tienen una *propiedad de una sola vía*. Es decir, si le dice a alguien la clave pública, esta persona no podrá falsificar un mensaje que se cifre con la clave privada.

El cifrado de la clave pública puede usarse para garantizar la confidencialidad. Un emisor que desea que la comunicación siga siendo confidencial usa la clave pública del receptor para cifrar el mensaje. A menos que conozca la clave privada, si un tercero obtiene una copia del texto cifrado mientras pasa a través de la red, no podrá leer el contenido ya que no puede descifrar el mensaje. De esta forma, el esquema asegura que los datos se mantengan confidenciales ya que sólo el receptor puede descifrar el mensaje.

29.11 Autenticación con firmas digitales

También es posible usar un mecanismo de cifrado para autenticar al emisor de un mensaje. La técnica se conoce como *firma digital*. Para firmar un mensaje, el emisor lo cifra usando una clave que sólo él conoce.[†] El receptor usa la función inversa para descifrar el mensaje. El receptor sabe quién envió el mensaje,

[†] Si no se requiere confidencialidad, el mensaje no tiene que cifrarse. En su lugar puede usarse una forma más eficiente de firma digital, en la que se cifra un hash del mensaje.

ya que sólo el emisor tiene la clave necesaria para realizar el cifrado. Para asegurar que los mensajes cifrados no se copien y se vuelvan a enviar más adelante, el mensaje original puede contener la hora y la fecha en que se creó.

Considere la forma en que puede usarse un sistema de clave pública para proveer una firma digital. Para firmar un mensaje, un emisor lo cifra usando su clave privada. Para verificar la firma, el receptor busca la clave pública del emisor y la usa para descifrar el mensaje. Puesto que únicamente el emisor conoce la clave privada, sólo él puede cifrar un mensaje que pueda decodificarse con la clave pública.

Lo interesante es que es posible cifrar un mensaje dos veces para garantizar la autenticación y la confidencialidad. Primero, el mensaje se firma usando la clave privada del emisor para cifrarlo. Segundo, el mensaje cifrado se vuelve a cifrar usando la clave pública del receptor. En sentido matemático, los dos pasos de cifrado pueden expresarse como:

$$X = \text{cifrar}(u2_publica, \text{cifrar}(u1_privada, M))$$

donde M indica el mensaje de texto simple que se va a enviar, X indica la cadena de texto cifrado que resulta del doble cifrado, $u1_privada$ indica la clave privada del emisor y $u2_publica$ indica la clave pública del receptor.

En el extremo receptor, el proceso de descifrado es el inverso del proceso de cifrado. Primero, el receptor usa su clave privada para descifrar el mensaje. El descifrado elimina un nivel de cifrado, pero deja el mensaje firmado en forma digital. Segundo, el receptor usa la clave pública del emisor para descifrar el mensaje de nuevo. El proceso puede expresarse como:

$$M = \text{descifrar}(u1_publica, \text{descifrar}(u2_privada, X))$$

donde X indica el texto cifrado que se transfirió a través de la red, M denota el mensaje de texto simple original, $u2_privada$ indica la clave privada del receptor y $u1_publica$ indica la clave pública del emisor.

Si resulta un mensaje significativo de los dos pasos, debe ser verdad que el mensaje sea confidencial y auténtico. El mensaje debe haber llegado a su receptor destinado, ya que sólo éste tiene la clave privada correcta necesaria para eliminar el cifrado exterior. El mensaje debe haber sido auténtico ya que sólo el emisor tiene la clave privada necesaria para cifrarlo, de modo que la clave pública lo descifrará correctamente.

29.12 Autoridades de claves y certificados digitales

Una de las preguntas fundamentales relacionadas con la tecnología de claves públicas se debe a la forma en que se obtiene una clave pública. Aunque es posible usar un medio convencional (como una libreta telefónica), hacerlo es complicado y propenso a errores debido a que el ser humano tendría que introducir manualmente las claves en sus computadoras. Surge la pregunta: ¿puede idearse un sistema automatizado para distribuir claves públicas? Desde luego que el sistema de distribución debe ser seguro, ya que si la clave pública que se otorga a un usuario es incorrecta, se quebranta la seguridad y no se puede confiar más en el cifrado. El problema se conoce como *problema de distribución de claves* y la

formación de un sistema de distribución de claves viable ha sido un obstáculo para la adopción extendida de los sistemas de claves públicas.

Se han propuesto varios mecanismos de distribución de claves, incluyendo uno que usa el *sistema de nombres de dominio*. En cada caso, un principio simple subyace en el esquema: al conocer una clave pública de una autoridad de claves es posible obtener otras claves públicas en forma segura. De esta forma, un administrador sólo necesita configurar una clave pública. La figura 29.7 ilustra el intercambio de mensajes cuando un usuario decide interactuar con un nuevo sitio web, *W*.

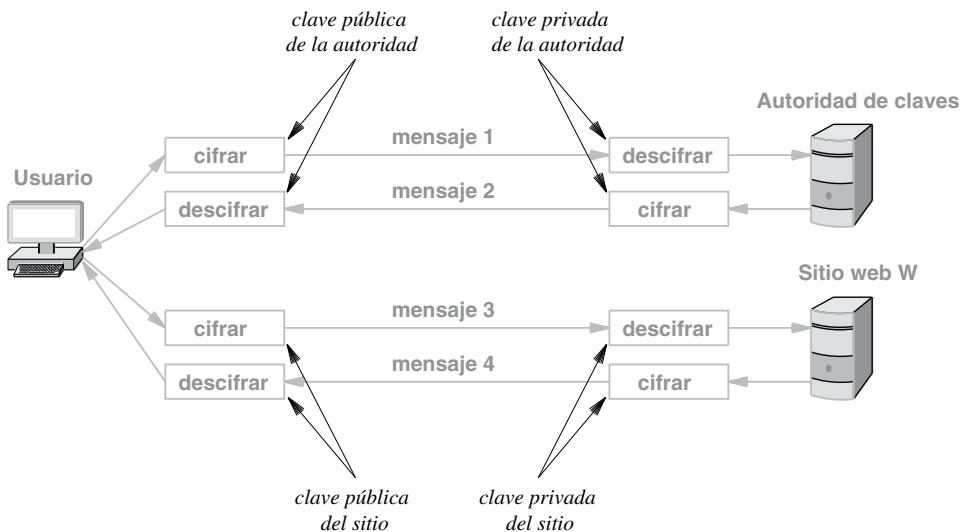


Figura 29.7 Ilustración del uso de una autoridad de claves para obtener una clave pública.

En la figura, un sitio necesita realizar una transacción segura con un sitio web, denominado *W*. Cada uno de los cuatro mensajes es confidencial. Sólo la autoridad de claves puede leer el mensaje 1, ya que se cifró usando la clave pública bien conocida de esta autoridad. El mensaje 2 debe haber sido generado por la autoridad de claves, ya que sólo ésta tiene la clave privada que coincide con la clave pública. Una vez que el usuario obtiene la clave pública para el sitio *W*, puede enviar una solicitud confidencial y sabe que sólo el sitio web especificado puede generar una respuesta (ya que sólo el sitio tiene la clave privada).

Aunque son posibles muchas variaciones, el principio importante es el siguiente:

Es posible crear un sistema de distribución de claves seguro que sólo requiera la configuración manual de una clave pública.

29.13 Firewall

Aunque la tecnología de cifrado ayuda a resolver muchos problemas de seguridad, se necesita una segunda tecnología. Conocida como *firewall* o *cortafuegos de Internet*,[†] esta tecnología ayuda a proteger las computadoras y redes de una organización del tráfico no deseado de Internet. Al igual que un muro cortafuegos, el firewall de Internet está diseñado para evitar que los problemas de Internet se esparzan a las computadoras de una organización.

Una vez que se coloca un firewall entre una organización y el resto de Internet, todos los paquetes que entran o salen de la organización pasan a través del mismo. La figura 29.8 ilustra la arquitectura.

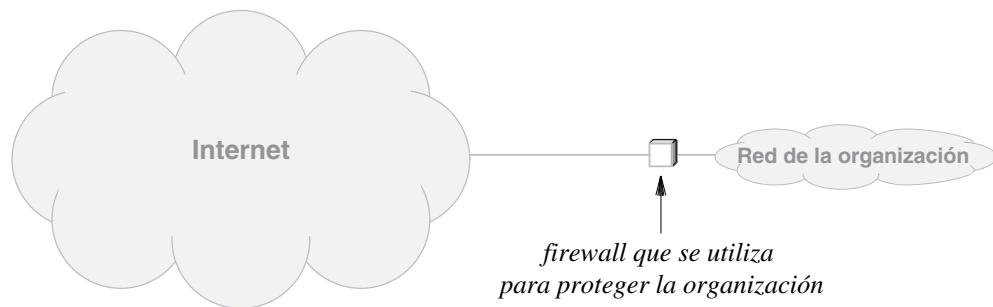


Figura 29.8 Ilustración de un firewall en la ruta entre Internet y la intranet de una organización.

Si una organización tiene varias conexiones a Internet, hay que colocar un firewall en cada una y todos deben estar configurados para implementar la política de seguridad de la organización. Además, el mismo firewall debe estar protegido contra la manipulación no autorizada. Para resumir:

- Todo el tráfico que entra a la organización pasa a través del firewall
- Todo el tráfico que sale de la organización pasa a través del firewall
- El firewall implementa la política de seguridad y descarta los paquetes que no se adhieren a la política
- El firewall en sí es inmune a los ataques de seguridad

Los firewalls son la herramienta de seguridad más importante para manejar la conexión entre dos organizaciones que no confían entre sí. Al colocar un firewall en cada conexión de red externa, una organización puede definir un *perímetro seguro* que evita que personas o equipos externos descubran las computadoras de la organización, inunden sus redes con tráfico no deseado o ataquen una computadora al enviar una secuencia de datagramas IP que provoque un comportamiento errado en la computadora (por ejemplo, que deje de funcionar). Además, un firewall puede evitar la exportación

[†] El término se deriva del aislante a prueba de fuego que se coloca entre dos partes de una estructura, para evitar que el fuego avance entre ellas.

de datos no deseados (por ejemplo, un usuario en la organización importa sin querer un virus que envía una copia del disco del usuario a alguien fuera de la organización).

Desde el punto de vista de un administrador, un firewall tiene una ventaja importante sobre otros esquemas de seguridad: centraliza el control y por lo tanto mejora la seguridad de manera considerable. Para brindar seguridad sin un firewall, una organización tendría que asegurar cada una de sus computadoras. Además, cada computadora debería implementar las mismas políticas. El costo de contratar personal para administrar tantas computadoras es alto. Además, una organización no puede depender de usuarios individuales para que configuren sus computadoras de manera correcta. Con un firewall, un administrador puede restringir todo el tráfico de Internet a un conjunto pequeño de computadoras y usar el personal para configurar y supervisar ese conjunto. En el caso extremo, todo el acceso exterior puede restringirse a una sola computadora. De esta forma, un firewall permite que una organización ahorre dinero y obtenga una mejor seguridad.

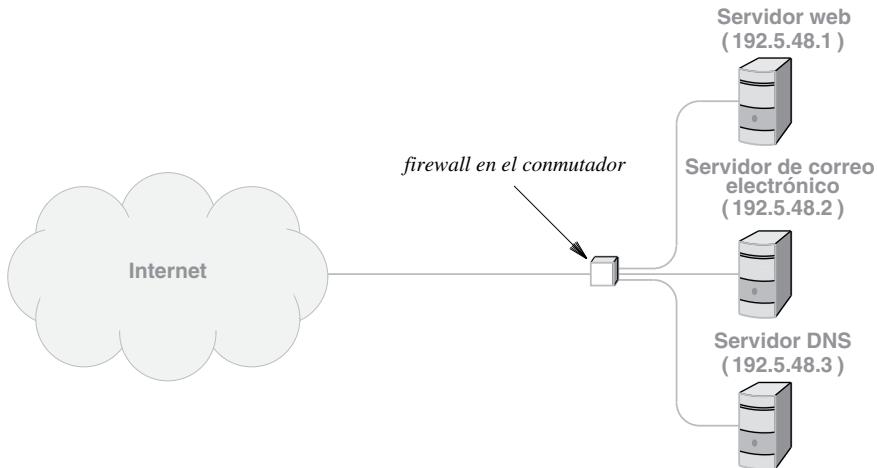
29.14 Implementación de un firewall con un filtro de paquetes

Aunque un firewall puede consistir en un dispositivo independiente, la mayoría de los firewalls están integrados a un commutador o enrutador. En cualquier caso, el mecanismo que se utiliza para crear un firewall se conoce como *filtro de paquetes*. Un filtro consiste en un mecanismo configurable que examina los campos de cada encabezado de paquete y decide si deja pasar el paquete a través del enrutador o lo descarta. Un administrador configura el filtro de paquetes especificando qué paquetes pueden pasar en cada dirección (es más seguro especificar el conjunto de paquetes permitidos en vez de especificar el conjunto de paquetes que se van a rechazar).

Para TCP/IP, una especificación de filtro de paquetes incluye por lo general un tipo de trama (0x0800 para IPv4 y 0x08DD para IPv6), una *dirección de origen* o *dirección de destino* IP (o ambas), un tipo de datagrama y un número de puerto de protocolo. Por ejemplo, para que equipos externos puedan contactarse al servidor web de la organización, un filtro de paquetes podría permitir una trama entrante que contenga un datagrama IP que transporte TCP de una dirección de origen IP y puerto de origen al puerto de destino 80 y una dirección IP de destino igual a la dirección IP del servidor web.

Puesto que permite que un administrador especifique las combinaciones de direcciones y servicios de origen y destino, el filtro de paquetes en un firewall permite controlar el acceso a servicios específicos en computadoras específicas. Por ejemplo, un administrador podría elegir permitir el tráfico entrante para acceder a un servidor web en una computadora, un servidor de correo electrónico en otra y un servidor DNS en una tercera. Desde luego que un administrador debe instalar también reglas de firewalls que permitan que los paquetes de respuesta fluyan hacia fuera del sitio. La figura 29.9 ilustra una configuración de firewall para dicho sitio.

La habilidad de permitir paquetes de manera selectiva para un servicio específico significa que un administrador puede controlar en detalle los servicios visibles externamente. De esta forma, incluso si un usuario inicia sin querer (o de manera intencional) un servidor de correo electrónico en su computadora, los equipos externos no podrán contactarse con el servidor.



Dir	Tipo de trama	IP orig.	IP dest	Tipo IP	Puerto orig.	Puerto dest
ent	0x0800	*	192.5.48.1	TCP	*	80
ent	0x0800	*	192.5.48.2	TCP	*	25
ent	0x0800	*	192.5.48.3	TCP	*	53
ent	0x0800	*	192.5.48.3	UDP	*	53
sal	0x0800	192.5.48.1	*	TCP	80	*
sal	0x0800	192.5.48.2	*	TCP	25	*
sal	0x0800	192.5.48.3	*	TCP	53	*
sal	0x0800	192.5.48.3	*	UDP	53	*

Figura 29.9 Configuración de firewalls de ejemplo para un sitio con tres servidores que ejecutan IPv4. Se usa un asterisco para denotar una entrada comodín que coincide con cualquier valor.

Podemos resumir lo siguiente:

Un firewall usa el filtrado de paquetes para prevenir la comunicación no deseada. La especificación de cada filtro proporciona una combinación de campos de encabezado, incluyendo las direcciones IP de origen y de destino, los números de puerto, así como el tipo de protocolo de transporte.

29.15 Sistemas de detección de intrusos

Un *sistema de detección de intrusos (IDS)* monitorea todos los paquetes que llegan a un sitio y notifica al administrador de éste si se detecta una violación de la seguridad. Un IDS proporciona una capa adicional en materia de seguridad, ya que aun cuando un firewall previene un ataque, un IDS puede notificar al administrador del sitio cuando está ocurriendo un problema.

La mayoría de los IDS pueden configurarse para vigilar tipos específicos de ataques. Por ejemplo, es posible configurar un IDS para que detecte un ataque de *escaneo de puertos* cuando un atacante envíe datagramas UDP a puertos de protocolo UDP consecutivos o intente abrir una conexión TCP en puertos de protocolo TCP consecutivos. De manera similar, es posible configurar un IDS para detectar un ataque potencial de inundación SYN si se vigila en busca de mensajes SYN repetidos de un origen dado. En algunos casos, un IDS y un firewall se interconectan para brindar un filtro automatizado: en vez de sólo notificar al administrador de un sitio sobre un problema, el IDS crea una regla en el firewall que bloquea los paquetes que provocan el problema. Por ejemplo, si un IDS detecta un ataque de inundación SYN proveniente de un origen dado, el IDS puede instalar una regla de firewall que bloquee los paquetes de ese origen. La razón de usar una metodología automatizada es la velocidad: un ser humano tarda varios segundos en responder después de recibir la notificación de un problema y, en una red de gigabits, pueden llegar más de 50,000 paquetes por segundo. Por lo tanto, es necesaria una respuesta rápida para evitar que un problema se vuelva abrumador.

La principal diferencia entre un IDS y un firewall está en que un IDS incluye la *información de estado*. A diferencia de un firewall que aplica reglas a un solo paquete a la vez, un IDS puede conservar un historial de paquetes. De esta forma, aunque un firewall puede determinar si admite o no un paquete SYN dado, un IDS puede observar que están llegando muchos paquetes SYN de un mismo origen. Desde luego, puesto que requiere de más cálculo y acceso a memoria que un firewall, un IDS no puede manejar tantos paquetes por segundo.

29.16 Exploración de contenido e inspección detallada de paquetes

Aunque puede encargarse de muchos problemas de seguridad, un firewall tiene una limitación severa ya que sólo examina los campos en un encabezado de paquetes. Es decir, un firewall no puede probar la carga útil de un paquete. Para entender por qué el contenido de los paquetes puede ser importante, considere los virus de computadora. Una de las formas más comunes en que se introduce un virus a una organización es a través de un adjunto de correo electrónico: el atacante envía un mensaje de correo electrónico con un programa de computadora como un adjunto. Si un usuario inocente abre el adjunto, el programa puede instalar cualquier software en la computadora del usuario, incluyendo *malware*[†] tal como un virus.

¿Cómo puede un sitio evitar problemas como la instalación de un virus? La respuesta está en el *análisis del contenido*. Hay dos tipos de análisis de contenido:

- Escaneo de archivos
- Inspección detallada de paquetes (DPI)

[†] Software malicioso.

Escaneo de archivos. La metodología más simple y directa para analizar el contenido opera sobre archivos completos. El escaneo de archivos es una técnica bien conocida que el software de seguridad instalado en una PC común utiliza. En esencia, un escáner de archivos recibe un archivo como entrada y busca patrones de bytes que indiquen un problema. Por ejemplo, muchos escáneres de virus buscan cadenas de bytes conocidas como *huella digital*. Es decir, una compañía que vende un escáner de virus recolecta copias de virus, coloca cada uno de ellos en un archivo, busca secuencias de bytes que no sean comunes y crea una lista de todas las secuencias. Cuando un usuario ejecuta software escáner de virus, el software busca archivos en el disco del usuario para ver si alguno contiene secuencias de bytes que coincidan con los elementos en la lista. El escaneo de archivos funciona bien para atrapar problemas comunes. Desde luego que se puede producir un *falso positivo* si resulta que un archivo ordinario contiene una cadena en la lista y se puede producir un *falso negativo* si existe un nuevo virus que no contenga ninguna de las cadenas en la lista.

Inspección detallada de paquetes (DPI). La segunda forma de análisis de contenido opera en paquetes en vez de archivos. Es decir, en vez de sólo examinar los encabezados en paquetes que pasan hacia el sitio, un mecanismo de DPI examina también los datos en la carga útil del paquete. Cabe mencionar que DPI no excluye el examen del encabezado; en muchos casos no es posible interpretar el contenido de la carga útil sin examinar los campos en el encabezado del paquete.

Como ejemplo de DPI, considere un ataque en el que un ligero error ortográfico del nombre de un dominio se usa para engañar a un usuario para que confíe en un sitio. Una organización que desea evitar dichos ataques puede poner en la *lista negra* un conjunto de elementos URL que suponen un riesgo de seguridad. El método de proxy requiere que todo usuario en el sitio configure su navegador para usar un *proxy web* (es decir, un sistema web intermedio que revisa un URL antes de obtener la página solicitada). Como alternativa es posible configurar un filtro DPI para inspeccionar cada paquete saliente y vigilar en espera de una solicitud HTTP en cualquiera de los sitios de la lista negra.

La principal desventaja de DPI surge de la sobrecarga computacional. Como la carga útil de un paquete en una trama Ethernet puede ser cerca de veinte veces más grande que un encabezado de paquete, DPI puede requerir veinte veces más procesamiento que la inspección del encabezado. Además la carga no está dividida en campos fijos, lo que significa que los mecanismos de DPI deben analizar el contenido durante una inspección. Como resultado:

Puesto que examinan cargas útiles de paquetes que son mucho más grandes que los encabezados de los paquetes y no están organizadas en campos fijos, los mecanismos de inspección detallada de paquetes (DPI) se limitan a las redes de baja velocidad.

29.17 Redes privadas virtuales (VPN)

Una de las tecnologías de seguridad más importantes y populares utiliza el cifrado para ofrecer un acceso seguro a la intranet de una organización desde cualquier sitio remoto, usando protocolos a través de la red Internet estándar (insegura). Conocida como *red privada virtual (VPN)*, la tecnología se diseñó en un principio para ofrecer una interconexión de bajo costo entre varios sitios geográficos de una organización. Para entender esto, considere las alternativas de interconexión:

- *Conexiones de redes privadas.* Una organización renta circuitos de datos para conectar sus sitios. Cada conexión rentada se extiende de un enrutador que se encuentra en uno de los sitios de la organización, a un enrutador en otro sitio; los datos pasan directamente de un enrutador a otro.
- *Conexiones públicas de Internet.* Cada sitio contrata con un ISP local el servicio de Internet. Los datos que se envían de un sitio corporativo a otro pasan a través de Internet.

La figura 29.10 ilustra dos posibilidades para una organización con tres sitios.

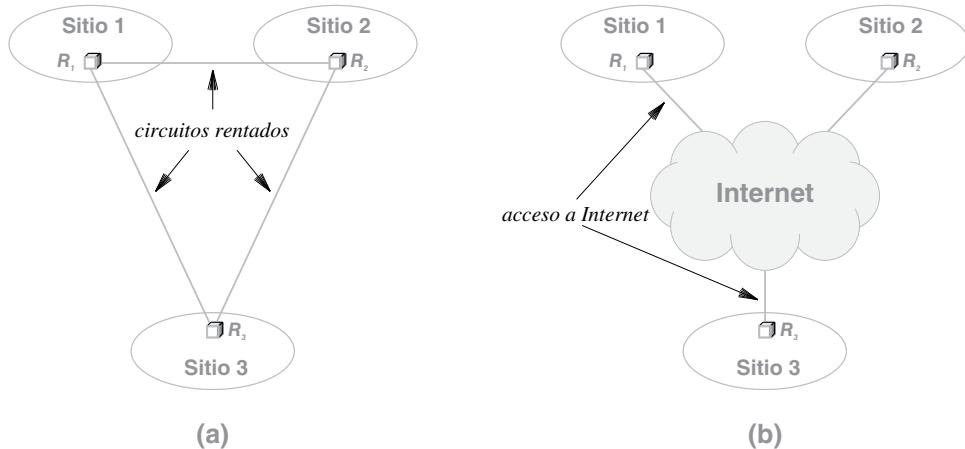


Figura 29.10 Sitios conectados por (a) circuitos rentados e (b) Internet.

La principal ventaja de usar circuitos rentados privados para interconectar sitios surge debido a que la red resultante asegura que los datos permanezcan completamente confidenciales.[†] Las compañías telefónicas se aseguran de que ninguna otra organización tenga acceso a un circuito rentado, lo que significa que ninguna otra organización puede leer los datos que pasan de un sitio a otro. La principal ventaja de usar conexiones de Internet es el bajo costo: en vez de pagar por circuitos dedicados para conectar los sitios, la organización sólo necesita pagar por el servicio de Internet en cada sitio. Por desgracia, Internet no puede garantizar la confidencialidad. Al viajar del origen al destino, un datagrama pasa a través de redes intermedias que pueden estar compartidas. Como consecuencia, personas externas pueden llegar a obtener copias del datagrama y examinar su contenido.

Una VPN combina lo mejor de ambas metodologías al usar Internet para transferir datos entre sitios y llevar a cabo pasos adicionales para asegurar que equipos externos no puedan acceder a los datos. Es decir, en vez de un circuito rentado costoso, una VPN usa el cifrado: todos los paquetes que se reenvían entre los sitios de una organización se cifran antes de enviarse.

[†] Hablando en sentido estricto, el término *privado* es poco apropiado. Sin embargo, los profesionales de redes usan a menudo *privado* cuando quieren decir *confidencial*.

Para crear una VPN aún más resistente a los ataques, una organización puede asignar enrutadores dedicados a la función de VPN y usar un firewall para evitar que los enrutadores de VPN acepten paquetes no autorizados. Por ejemplo, suponga que cada uno de los enrutadores de la figura 29.10(b) se dedican a la función de VPN (es decir, asuma que el sitio tiene enrutadores adicionales que manejan el tráfico normal desde y hacia Internet). Un firewall que protege el enrutador VPN en el sitio 1 puede restringir todos los paquetes entrantes para que tengan una dirección IP de origen del enrutador VPN en el sitio 2 o del enrutador VPN en el sitio 3. De manera similar, un firewall en cada uno de los otros dos sitios restringe los paquetes entrantes en ese sitio. Las restricciones ayudan a que el sistema resultante sea más inmune a la falsificación de direcciones y los ataques de DoS.

29.18 El uso de la tecnología VPN para el trabajo a distancia

Aunque se diseñó en un principio para interconectar sitios, la tecnología VPN se ha vuelto en extremo popular entre los empleados que *trabajan a distancia* (es decir, que trabajan desde una ubicación remota). Hay dos formas de VPN:

- Dispositivo independiente
- Software de VPN

Dispositivo independiente. La organización entrega a un empleado un dispositivo físico que se conoce algunas veces como *enrutador VPN*. El dispositivo se conecta a Internet, establece automáticamente, una comunicación segura a un servidor VPN en el sitio de la organización y proporciona conexiones de red de área local a las que el usuario puede conectar computadoras y teléfonos IP. Lógicamente, el dispositivo VPN extiende la red de la organización al sitio del usuario, para permitir que las computadoras conectadas al dispositivo VPN operen como si estuvieran conectadas a la red corporativa. Por lo tanto, cuando la computadora del usuario inicia y obtiene una dirección IP, el servidor DHCP de la organización es el que emite esta dirección. De manera similar, la tabla de reenvío en la computadora del usuario se configura como si la computadora estuviera en el sitio de la organización, y cada vez que la computadora envía un paquete, el VPN lo cifra y envía la versión cifrada a la organización a través de Internet. Cada vez que llega un paquete de la organización, el dispositivo VPN lo descifra y transmite el resultado a la computadora del usuario.

Software de VPN. Aunque un dispositivo independiente funciona bien para un empleado que trabaja en casa o en una oficina remota, dichos dispositivos son complicados para los empleados que viajan. Para manejar estos casos, una organización usa *software de VPN* que se ejecuta en la computadora personal del usuario. Un usuario se conecta a Internet y luego inicia la aplicación de VPN. Al iniciar, la aplicación de VPN se interpone entre la conexión a Internet; es decir, el software de VPN se dispone a capturar todos los paquetes salientes y entrantes. Cifra cada paquete saliente y envía el paquete cifrado al servidor VPN corporativo y descifra cada paquete entrante.

29.19 Comparación entre cifrado de paquetes y uso de túneles

La anterior explicación de las VPN genera una pregunta interesante: ¿cómo deben cifrarse los datos para la transmisión a través de Internet? Existen tres opciones principales:

- Cifrado de la carga útil
- Túneles de IP en IP
- Túneles de IP en TCP

Cifrado de la carga útil. Para mantener confidencial el contenido de un datagrama, el método de *cifrado de carga útil* cifra el área de carga útil de un datagrama, pero no toca el encabezado. Como los campos del encabezado no están cifrados, personas externas podrán conocer las direcciones de origen y destino que se utilicen, así como los números de puertos de protocolo. Por ejemplo, suponga que el director financiero (CFO) está en un sitio y que el presidente de la compañía está en otro. Suponga además que el CFO envía un mensaje breve de correo electrónico al presidente cuando las noticias financieras son buenas y una explicación extensa siempre que las noticias financieras no son buenas. Alguien externo podría observar que, poco después de que fluye un mensaje corto entre dos computadoras específicas, aumenta el precio de las acciones.

Túneles de IP en IP. Algunas VPN usan la tecnología de *túneles de IP en IP* que mantiene oculto todo el datagrama (incluyendo el encabezado) a medida que pasa de un sitio a otro a través de Internet. Al encontrar un datagrama saliente, el software de VPN emisor cifra todo el datagrama y coloca el resultado dentro de otro datagrama para la transmisión. Por ejemplo, considere las conexiones en la figura 29.10(b) que se muestran en la página 524. Suponga que la computadora *X* en el sitio 1 crea un datagrama para la computadora *Y* en el sitio 2. El datagrama se reenvía a través del sitio 1 a un enrutador R_1 (es decir, el enrutador que conecta el sitio 1 con Internet). La herramienta VPN en R_1 cifra el datagrama original y lo encapsula en un nuevo datagrama para transmitirlo al enrutador R_2 , el enrutador en el sitio 2. Cuando llega el datagrama encapsulado, el software de VPN en R_2 descifra la carga útil para extraer el datagrama original y luego lo reenvía al destino *Y*. La figura 29.11 ilustra la encapsulación.

En la figura 29.11, (a) muestra el datagrama original, (b) muestra el criptograma que resulta del cifrado y (c) muestra el datagrama exterior que se envía de R_1 a R_2 . Cabe mencionar que las direcciones internas están ocultas debido a que todos los datagramas que viajan a través de Internet entre los sitios 1 y 2 enlistan a los enrutadores R_1 y R_2 como las direcciones de origen y de destino.

Para resumir:

Cuando una VPN usa un encapsulamiento de IP en IP, todos los campos en el datagrama original están cifrados, incluyendo el encabezado original.

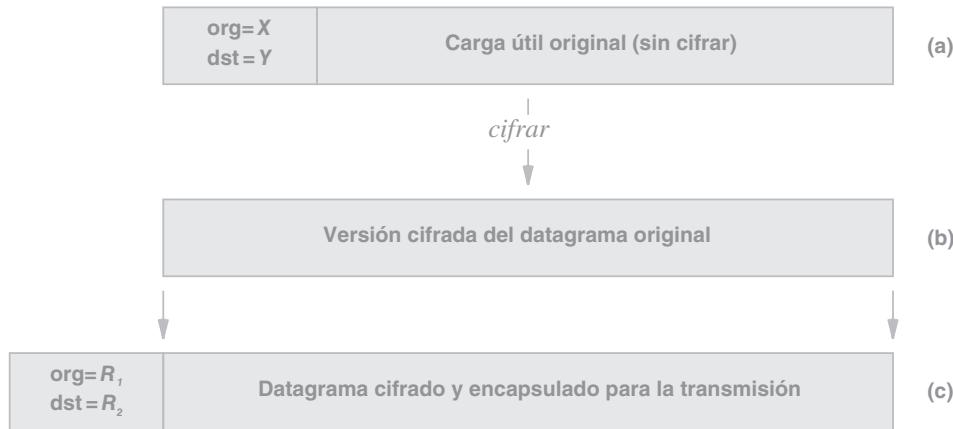


Figura 29.11 Ilustración de un encapsulamiento de IP en IP utilizada con una VPN.

Túneles de IP en TCP. La tercera posible alternativa que se utiliza para mantener los datos confidenciales implica el uso de un túnel TCP. Es decir, dos partes establecen una conexión TCP y luego usan la conexión para enviar datagramas cifrados. Cuando debe enviarse un datagrama éste se cifra por completo, se agrega un pequeño encabezado para marcar el límite entre los datagramas y el resultado se envía a través de la conexión TCP. Por lo general, el encabezado consiste en un entero de dos bytes que especifica la longitud del datagrama. Del otro lado de la conexión TCP, el software VPN receptor lee el encabezado y luego lee el número especificado de bytes adicionales para obtener el datagrama. Una vez que se recibe todo el texto cifrado para un datagrama, el receptor lo descifra y procesa el datagrama original.

La principal ventaja de usar IP en TCP en vez de IP en IP se debe a la entrega confiable: TCP asegura que todos los datagramas enviados entre dos sitios lleguen en forma confiable y en orden. La principal desventaja de usar IP en TCP es el bloqueo de encabezado de línea, ya que como todos los datagramas deben entregarse en orden, si se pierde o retrasa un segmento de TCP, TCP no puede entregar los datos de los segmentos sucesivos, aun cuando hayan llegado correctamente. Si pensamos en una VPN como un medio para transferir una cola de paquetes, toda la cola permanecerá bloqueada hasta que se haya entregado el primer datagrama.

Surge un problema final relacionado con los túneles de VPN, el cual está relacionado con el rendimiento. Hay tres aspectos a considerar:

- La latencia
- La velocidad de transferencia
- La sobrecarga y la fragmentación

Latencia. Para entender la cuestión de latencia, considere una organización en la costa oeste de Estados Unidos y suponga que un empleado viaja a la costa este, aproximadamente a 3000 millas

de distancia. Recuerde que el software de VPN simplemente transfiere los datagramas de regreso a la organización local, ya que una vez que un datagrama llega a la organización, éste debe ser enrutado hacia su destino. Por ejemplo, si el empleado navega en una página web, cada solicitud debe viajar desde la ubicación actual del empleado hasta el servidor VPN de la organización y de ahí hacia el servidor web. La respuesta debe viajar de nuevo al servidor VPN de la organización y por último hacia la ubicación remota del empleado. La latencia requerida para acceder a un recurso cerca del empleado es en especial elevada, ya que los datagramas deben viajar desde el empleado a través de la VPN hacia la organización en la costa oeste y de vuelta al recurso en la costa este. Como resultado, el viaje redondo requiere que un datagrama realice cuatro recorridos del continente.

Velocidad de transferencia. Otro problema con una VPN convencional surge de la velocidad de transferencia disponible en Internet. El problema puede ser importante al usar aplicaciones diseñadas para una LAN de alta velocidad. Por ejemplo, en algunas organizaciones las páginas web que usan los empleados para las actividades comerciales internas de la compañía contienen muchos gráficos. Una LAN en el sitio ofrece la suficiente velocidad de transferencia como para poder descargar rápido las páginas web. Para un usuario remoto conectado a través de una VPN, la baja velocidad de transferencia puede hacer que la espera de una página web sea frustrante.

Sobrecarga y fragmentación. Un tercer aspecto del rendimiento se debe a que el uso de túneles agrega sobrecarga a un datagrama. Para entender el problema, suponga que un sitio usa Ethernet y una aplicación ha creado un datagrama de 1500 bytes de longitud (es decir, el datagrama es igual de grande que la MTU de la red). Cuando un enrutador VPN encapsula el datagrama cifrado en otro datagrama IP, se agregan al menos veinte bytes adicionales para el encabezado del datagrama exterior. El datagrama resultante excede la MTU de la red y se fragmentará antes de la transmisión. Puesto que ambos fragmentos deben llegar antes de que pueda procesarse el datagrama, la probabilidad de retraso o de pérdida es más alta.

29.20 Tecnologías de seguridad

Se han inventado una variedad de tecnologías de seguridad para usarse en Internet. Las más destacadas son:

- *PGP (privacidad bastante buena).* Un sistema criptográfico que pueden usar las aplicaciones para cifrar datos antes de la transmisión. PGP se desarrolló en el Instituto Tecnológico de Massachusetts (MIT) y es bastante popular entre los científicos de computadoras.
- *SSH (shell seguro).* Un protocolo de la capa de aplicación para inicios de sesión remotos que garantiza la confidencialidad al cifrar los datos antes de transmitirlos a través de Internet.
- *SSL (capa de sockets segura).* Una tecnología que diseñó originalmente Netscape Communications y utiliza el cifrado para brindar autenticación y confidencialidad. El software SSL se coloca entre una aplicación y la API de sockets para cifrar los datos antes de transmitirlos por Internet. SSL se usa en una conexión web para permitir que los usuarios

realicen transacciones financieras en forma segura (por ejemplo, enviar un número de tarjeta de crédito a un servidor web).

- *TLS (seguridad de la capa de transporte)*. Diseñada por el IETF a finales de la década de 1990 como sucesora de SSL, la tecnología TLS se basa en la versión 3 de SSL. Tanto SSL como TLS están disponibles para usarse con HTTPS.
- *HTTPS (seguridad de HTTP)*. En realidad no es una tecnología independiente; HTTPS combina HTTP con SSL o TLS y un mecanismo de certificación para ofrecer a los usuarios una comunicación autenticada y confidencial a través de Web. HTTPS usa el puerto TCP 443 en vez del puerto 80.
- *IPsec (seguridad IP)*. Un estándar de seguridad utilizado con datagramas IP. IPsec usa técnicas criptográficas y permite al emisor elegir entre autenticación (es decir, validar al emisor y al receptor del datagrama) o confidencialidad (es decir, cifrar la carga útil del datagrama).
- *RADIUS (servicios de autenticación remota de llamadas de usuarios)*. Un protocolo que se utiliza para brindar autenticación, autorización y contabilidad centralizadas. RADIUS es popular con los ISP que tienen llamadas de usuarios y con los sistemas VPN que proporcionan acceso a usuarios remotos.
- *WEP y WPA (privacidad equivalente por cable y acceso Wi-Fi protegido)*. WEP fue en un principio parte del estándar de LAN inalámbrica Wi-Fi[†] y se utilizó para mantener las transmisiones confidenciales. Los investigadores en la Universidad de California en Berkeley encontraron varias debilidades en WEP, por lo que se desarrolló el WPA (*posteriormente WPA2*) como reemplazo.

29.21 Resumen

Las redes de computadoras e Internet pueden usarse para actividades criminales. Las principales amenazas incluyen el *phishing*, la tergiversación, los fraudes, la negación de servicio, la pérdida de control y la pérdida de información. Las técnicas que se utilizan en los ataques son la intercepción electrónica, la reproducción duplicada, el desbordamiento de búfer, la falsificación de direcciones y de nombres, la DoS con inundación de paquetes y SYN, el descifrado de claves, el escaneo de puertos y la intercepción de paquetes.

Cada organización necesita definir una política de seguridad que especifique los aspectos de la integridad de los datos (protección contra modificación), la disponibilidad de los mismos (protección contra interrupción del servicio) y la confidencialidad o privacidad de la información (protección contra falsificación y descubrimiento). Además, una organización debe considerar la rendición de cuentas (es decir, cómo se mantiene el rastro de una auditoría) y la autorización (es decir, cómo se pasa la responsabilidad de la información de una persona a otra).

[†] WEP se aplica a una variedad de protocolos del IEEE 802.11.

Se creó un conjunto de tecnologías para cubrir varios aspectos de seguridad. El conjunto incluye el cifrado, el *hashing*, las firmas y los certificados digitales, los firewalls, los sistemas de detección de intrusos, la inspección detallada de paquetes, el escaneo de contenido y las redes privadas virtuales. El cifrado está entre las tecnologías más fundamentales que se utilizan en muchos mecanismos de seguridad.

El cifrado de claves privadas usa una sola clave para cifrar y descifrar mensajes. El emisor y el receptor deben mantener la clave en secreto. Los sistemas de cifrado de claves públicas usan un par de claves: una clave se mantiene en secreto y la otra (la clave pública) se anuncia en todas partes. Las firmas digitales usan el cifrado para autenticar mensajes. Una autoridad de claves puede emitir certificados para validar claves públicas.

Un firewall protege un sitio contra el ataque al restringir los paquetes que pueden entrar o salir. Para configurar un firewall, un administrador crea un conjunto de reglas que proporcionan valores específicos para los campos de encabezado de paquetes. Los sistemas de detección de intrusos que mantienen la información de estado pueden identificar ataques como los paquetes SYN repetidos.

Las redes privadas virtuales (VPN) ofrecen los beneficios de confidencialidad y bajo costo. La tecnología VPN permite que un empleado trabaje a distancia. Para mantener la información confidencial, un emisor puede cifrar la carga útil, usar túneles de IP en IP o túneles de IP en TCP. El uso de túneles tiene la ventaja de cifrar tanto los encabezados de los paquetes como la carga útil. Algunas aplicaciones no funcionan bien a través de una VPN debido a que tiene un retraso mayor, una velocidad de transferencia menor y una sobrecarga más alta que una conexión directa.

Existen muchas tecnologías de seguridad. Algunos ejemplos son: PGP, SSH, SSL, TLS, HTTPS, IPsec, RADIUS y WPA.

EJERCICIOS

- 29.1** Mencione en una lista los principales problemas de seguridad en Internet y una descripción breve de cada uno.
- 29.2** Nombre la técnica que se utiliza en los ataques de seguridad.
- 29.3** Suponga que un atacante encuentra la forma de almacenar una vinculación arbitraria en su servidor DNS local. ¿Cómo puede usar el atacante dicha debilidad para obtener la información de su cuenta bancaria?
- 29.4** A menudo los ataques de DoS envían segmentos SYN de TCP. ¿Puede un atacante crear también un ataque DoS si envía segmentos de datos TCP? Explique.
- 29.5** Si una contraseña contiene ocho letras y dígitos en mayúsculas y minúsculas, ¿cuántas posibles contraseñas podría necesitar un atacante para intentar obtener acceso?
- 29.6** ¿Por qué es difícil derivar una política de seguridad?
- 29.7** Suponga que una compañía idea una política de seguridad que especifica que sólo el personal de recursos humanos está autorizado para ver los archivos de nómina. ¿Qué tipo de mecanismo se necesita para implementar dicha política? Explique.
- 29.8** Enliste y describa las ocho técnicas básicas de seguridad.
- 29.9** ¿Qué es una lista de control de acceso (ACL) y cómo se utiliza?
- 29.10** ¿A qué se refiere la *criptografía*?

- 29.11** Lea sobre el *estándar de cifrado de datos (DES)*. ¿Qué tamaño de clave debe usarse para los datos que son en extremo importantes?
- 29.12** Suponga que su amigo tiene una clave pública y una privada para usar con el cifrado de claves públicas. ¿Puede su amigo enviarle un mensaje confidencial (es decir, un mensaje que sólo usted pueda leer)? ¿Por qué sí o por qué no?
- 29.13** Si usted y su amigo tienen cada uno un par de claves públicas y privadas para un sistema de cifrado de claves públicas, ¿cómo pueden usted y su amigo comunicarse a diario sin ser engañados por un ataque de reproducción duplicada?
- 29.14** ¿Cómo pueden dos partes usar el cifrado de claves públicas para firmar un contrato que después se envía a un tercero?
- 29.15** ¿Qué es un certificado digital?
- 29.16** ¿Qué es un firewall y dónde se instala?
- 29.17** Muchos productos de firewall comerciales permiten que un administrador especifique los paquetes a *rechazar* así como los paquetes que debe *aceptar*. ¿Cuál es la desventaja de una configuración que permite el rechazo de paquetes?
- 29.18** Vuelva a escribir la configuración del firewall de la figura 29.9 para que alguien externo pueda usar *ping* con cada uno de los tres servidores.
- 29.19** Vuelva a escribir la configuración del firewall de la figura 29.9 para mover el servidor de correo electrónico a la computadora que ejecuta el servidor web.
- 29.20** Lea sobre los sistemas IDS comerciales y haga una lista de ataques que estos sistemas pueden detectar.
- 29.21** Considere un sistema DPI que busca una cadena de K bytes en cada paquete. Si un paquete contiene 1486 bytes de carga útil, ¿cuál es el mayor número de comparaciones que deben hacerse para examinar el paquete, suponiendo un algoritmo de coincidencia simple?
- 29.22** ¿Por qué no se utiliza la inspección detallada de paquetes en las redes de más alta velocidad?
- 29.23** ¿Cuáles son los dos objetivos de un sistema VPN?
- 29.24** ¿Cuáles son las tres formas en que una VPN puede transferir datos a través de Internet?
- 29.25** Cuando una VPN usa la túneles de IP en IP, ¿qué es lo que evita que un atacante lea el encabezado del datagrama original?
- 29.26** En algunos sistemas VPN, un emisor adjunta una cantidad aleatoria de bits cero adicionales a un datagrama antes de cifrarlo, y el receptor usa el campo de longitud del datagrama para descartar los bits adicionales después de descifrar el datagrama. De esta forma, el único efecto del relleno aleatorio es hacer que la longitud del datagrama cifrado sea independiente de la longitud de la versión sin cifrar. ¿Por qué es importante la longitud?
- 29.27** Enliste ocho tecnologías de seguridad que se utilizan en Internet y describa el propósito de cada una.
- 29.28** Lea sobre las vulnerabilidades en el protocolo WEP. ¿Cómo evita el protocolo WPA esos problemas?

Contenido del capítulo

- 30.1 Introducción, 533
- 30.2 Administración de una intranet, 533
- 30.3 FCAPS: el modelo estándar de la industria, 534
- 30.4 Ejemplos de elementos de red, 536
- 30.5 Herramientas de administración de redes, 536
- 30.6 Aplicaciones de administración de redes, 538
- 30.7 Protocolo simple de administración de redes, 539
- 30.8 Paradigma de obtener y almacenar de SNMP, 539
- 30.9 La MIB de SNMP y los nombres de objetos, 540
- 30.10 Las variables de MIB, 541
- 30.11 Variables de MIB que corresponden a arreglos, 541
- 30.12 Resumen, 542

30

Administración de redes (SNMP)

30.1 Introducción

Los capítulos anteriores describen una variedad de aplicaciones convencionales que se usan en Internet. Este capítulo expande nuestro estudio de las aplicaciones de red al considerar la administración de redes. El capítulo introduce un modelo conceptual que se utiliza en la industria y utiliza este modelo para explicar el alcance de las actividades de administración. Después de explicar por qué la administración de redes es tanto importante como difícil, el capítulo describe las tecnologías de administración de redes. Analiza las herramientas que están disponibles, incluyendo el software de aplicación que los administradores usan para medir o controlar los comutadores, los enruteadores y otros dispositivos que constituyen una intranet. El capítulo explica el paradigma general utilizado por los sistemas de administración y describe la funcionalidad que proporcionan dichos sistemas. Por último, el capítulo considera un ejemplo específico de un protocolo de administración de redes y explica cómo opera el software para el mismo.

30.2 Administración de una intranet

Un *administrador de red* es una persona responsable de planear, instalar, operar, monitorear y controlar los sistemas de hardware y software que constituyen una red de computadoras o intranet. Un administrador planea una red que cumple los requisitos de rendimiento, monitorea operaciones, detecta y corrige problemas que hacen la comunicación inefficiente o imposible y trabaja para evitar condiciones que producirán un problema de nuevo. Puesto que las fallas de hardware o software pueden provocar problemas, un administrador de red debe monitorear ambas.

La administración de redes puede ser difícil por tres razones. Primero, la mayoría de las intranets son heterogéneas, ya que contienen componentes de hardware y software fabricados por varias compañías. Segundo, la tecnología cambia, lo que significa que constantemente aparecen nuevos dispositivos y servicios. Tercero, la mayoría de las intranets son físicamente grandes, con cierta distancia entre los dispositivos. Diagnosticar los problemas en un dispositivo remoto puede ser bastante difícil.

La administración de redes también es difícil debido a que muchos mecanismos de red están diseñados para solucionar los problemas de manera automática. Los protocolos de enrutamiento evitan las fallas y la pérdida intermitente de paquetes puede pasar desapercibida debido a que el TCP hace una retransmisión automática. Por desgracia, la recuperación automática de errores tiene consecuencias. La retransmisión de paquetes usa el ancho de banda de la red que podría usarse para enviar nuevos datos. De manera similar, una falla de hardware no detectada puede volverse crítica si también falla la ruta de respaldo.

Para resumir:

Aunque el hardware de red y el software de protocolo contienen mecanismos para calcular automáticamente una ruta que rodee las fallas o para retransmitir los paquetes perdidos, los administradores de red necesitan detectar y corregir los problemas que surgen.

30.3 FCAPS: el modelo estándar de la industria

La industria de las redes usa el *modelo FCAPS* para caracterizar el alcance de la administración de redes. El acrónimo se deriva de la recomendación M.3400 publicada por la *Unión internacional de telecomunicaciones (ITU)*.[†] FCAPS se expande en una lista de cinco aspectos de la administración. La figura 30.1 sintetiza el modelo.

Abreviación	Significado
F	Detección y corrección de fallas
C	Configuración y operación
A	Contabilidad y facturación
P	Evaluación del rendimiento y optimización
S	Garantía de seguridad y protección

Figura 30.1 El modelo FCAPS de la administración de redes.

Detección y corrección de fallas. La detección de fallas representa una parte importante del aspecto operacional de la administración de redes. Un administrador monitorea el equipo de red para detectar problemas y sigue los pasos apropiados para corregirlos. Las posibles fallas incluyen fallas de software

[†] M.3400 es parte de una serie de estándares que especifican cómo debe configurarse y operar una *red de administración de telecomunicaciones (TMN)*

(por ejemplo, la falla de un sistema operativo en un servidor), fallas de enlace (por ejemplo, si alguien corta por accidente una fibra óptica) y las fallas de equipo (por ejemplo, si falla la fuente de alimentación en un enrutador).

A menudo los usuarios reportan las fallas citando un síntoma de alto nivel como “Perdí el acceso a un disco compartido”. Un administrador debe investigar para determinar si el problema recae en el software, en la seguridad (por ejemplo, una nueva contraseña), en un servidor o en un enlace. Decimos que un administrador realiza un *análisis de la causa raíz*. Con frecuencia un administrador puede determinar la causa al correlacionar muchos informes. Por ejemplo, si varios usuarios en un sitio de repente comienzan a quejarse de que una variedad de servicios no están disponibles, un administrador podría sospechar que el problema recae en una conexión compartida que todos los servicios utilizan.

Configuración y operación. Tal vez parezca que la configuración sea un aspecto trivial de la administración de redes, puesto que ésta sólo necesita realizarse una vez, y luego puede guardarse de modo que un dispositivo instale automáticamente la configuración durante un reinicio. De hecho, la configuración es compleja por tres razones. Primera, una red contiene muchos dispositivos y servicios, y las configuraciones deben ser consistentes entre todos los dispositivos. Segunda, a medida que se agregue nuevo equipo y servicios o que cambien las políticas, un administrador de redes debe considerar todas las configuraciones para asegurar que toda la red implemente los cambios de manera correcta. Tercera, las herramientas actuales permiten que un administrador configure dispositivos y protocolos individuales, pero no hay una forma fácil de configurar un conjunto de dispositivos heterogéneos.

Contabilidad y facturación. En muchas intranets corporativas, la contabilidad y la facturación son triviales. La corporación carga el costo de operar una red a una cuenta central, en forma muy parecida al costo del servicio de energía eléctrica o de teléfono. Sin embargo, en las redes de ISP la contabilidad y la facturación pueden consumir más del tiempo de un administrador que cualquier otro aspecto de la administración. Por ejemplo, si un ISP ofrece el servicio en niveles con límites en cuanto al tráfico que puede enviarse, un sistema debe contabilizar el tráfico de cada cliente por separado. A menudo los acuerdos de servicio especifican que la tarifa que debe pagar un cliente depende de una medida tal como los bytes totales que éste envía por día. Por lo tanto, es importante medir todo el tráfico de un cliente y mantener registros detallados que puedan usarse para generar una factura.

Evaluación y optimización del rendimiento. Un administrador realiza dos tipos de evaluaciones del rendimiento: la *evaluación de diagnóstico* para detectar problemas e ineficiencias y la *evaluación de tendencias*, que permite que un administrador se anticipa a la necesidad de aumentar la capacidad. La evaluación de diagnóstico busca formas de maximizar el uso de una red existente. Por ejemplo, si un administrador encuentra una ruta con poca utilización, podría buscar formas de cambiar el tráfico hacia esa ruta. La evaluación de tendencias busca formas de incrementar el rendimiento de la red para satisfacer las necesidades a futuro. Por ejemplo, la mayoría de los administradores vigilan la utilización del enlace entre su organización e Internet, y hacen planes para incrementar la capacidad de éste cuando la utilización promedio aumenta a más del 50%.

Garantía de seguridad y protección. Puesto que cruza capas de la pila de protocolos y abarca varios dispositivos, la seguridad está entre los aspectos más difíciles de la administración de redes. En especial, la seguridad sigue la analogía del enlace más débil, en el cual la seguridad de un sitio puede comprometerse si la configuración es incorrecta en un dispositivo. Además, como los atacantes idean continuamente nuevas formas de quebrantar la seguridad, una red que sea segura en un momento dado puede estar comprometida más tarde, a menos que un administrador realice cambios.

30.4 Ejemplos de elementos de red

Los sistemas de administración de redes usan el término genérico *elemento de red* para referirse a cualquier dispositivo, sistema o mecanismo de red que puede administrarse. Aunque muchos elementos de red consisten en un dispositivo físico, la definición abarca servicios tales como el DNS. La figura 30.2 enumera algunos ejemplos de elementos de red.

Elementos de red administrables	
Commutador de capa 2	Enrutador IP
Commutador de VLAN	Firewall
Punto de acceso inalámbrico	Circuito digital (CSU/DSU)
Módem DSL de extremo cercano	DSLAM
Servidor DHCP	Servidor DNS
Servidor Web	Balanceador de carga

Figura 30.2 Ejemplos de elementos de red que deben administrarse.

La industria utiliza el término *administración de elemento* para referirse a la configuración y operación de un elemento de red individual. Por desgracia, la mayoría de las herramientas disponibles sólo proporcionan la administración de elementos.[†] Por consiguiente, para crear un servicio de extremo a extremo, un administrador debe configurar cada elemento de red a lo largo de la ruta. Por ejemplo, para crear un túnel MPLS a través de varios enrutadores, un administrador debe configurar cada enrutador por separado. De manera similar, para implementar una política a través de toda una red, un administrador debe configurar cada elemento.

Desde luego que es fácil para un ser humano cometer un error al configurar muchos dispositivos, lo cual hace a la administración de elementos susceptible a la mala configuración. Lo que es más importante, para diagnosticar un error, un administrador debe examinar un sistema a la vez. En conclusión:

Puesto que permite que un administrador configure, monitoree o controle un elemento de red a la vez, un sistema de administración de elementos requiere de mucha labor y es propenso a errores.

30.5 Herramientas de administración de redes

Las herramientas de administración de redes pueden clasificarse en doce categorías que caracterizan su propósito general:

[†] Aunque existen algunas herramientas para cargar una configuración en varios elementos, pocas pueden configurar servicios a través de una red en forma significativa.

- Prueba de capa física
- Accesibilidad y conectividad
- Análisis de paquetes
- Detección de redes
- Interrogación de dispositivos
- Monitoreo de eventos
- Monitoreo del rendimiento
- Análisis de flujo
- Enrutamiento e ingeniería de tráfico
- Configuración
- Implementación de seguridad
- Planeación de redes

La prueba de capa física incluye la prueba de sensor de portadora que se encuentra en muchas tarjetas de interfaz de LAN y en los medidores de fuerza inalámbrica que se usan para medir la fuerza de la señal de RF. *Ping* es el mejor ejemplo de una herramienta de accesibilidad; los administradores de redes la utilizan mucho. Un *analizador de paquetes*, también conocido como *analizador de protocolos*, captura y despliega paquetes o estadísticas sobre paquetes. El analizador *Wireshark* está disponible para descarga.

Una herramienta de detección de redes produce un mapa de la red al sondear sus dispositivos. A menudo un administrador usa dicho mapa para encontrar los elementos en una red y luego usa la herramienta de interrogación de dispositivos para acceder a cada uno. Las herramientas de monitoreo de eventos producen alertas. Por lo general, un administrador configura un dispositivo para enviar una alerta cuando se cruzan ciertos umbrales (por ejemplo, si la utilización de un enlace llega al 80%) y una herramienta de monitoreo despliega una alerta en la estación de trabajo de un administrador. Las herramientas de monitoreo del rendimiento trazan el rendimiento a través del tiempo para ayudar a que un administrador detecte tendencias.

Las herramientas de análisis de flujos, como el analizador NetFlow, ayudan a un administrador a detectar tendencias. En vez de sólo generar informes sobre el tráfico en general, un analizador de flujos puede ayudar a un administrador a detectar los cambios en tipos específicos de tráfico (por ejemplo, un aumento en el tráfico de VoIP).

Las herramientas de enrutamiento, de ingeniería de tráfico y de configuración, están relacionadas. Cada una ayuda a un administrador a controlar los elementos. Las herramientas de enrutamiento controlan la configuración y el monitoreo de los protocolos de actualización de enrutamiento y las tablas de reenvío que resultan de los cambios en el enrutamiento. Las herramientas de ingeniería de tráfico se enfocan en la configuración y el monitoreo de túneles MPLS y los parámetros de relacionados de QoS. Las herramientas de configuración de propósito general permiten a un administrador instalar o cambiar la configuración en los elementos. En especial, ciertas herramientas de configuración automatizan la tarea repetitiva de realizar cambios en un conjunto de elementos (por lo general idénticos). Por ejemplo, si cambia la regla de un firewall y un sitio tiene varios firewalls, una herramienta de configuración automatizada (a menudo un script de Perl) puede instalar el mismo cambio en cada uno.

Existen muchas herramientas que controlan una variedad de elementos de seguridad. Algunas permiten que un administrador especifique una política y la herramienta intenta configurar dispositivos para implementarla o intenta medir los dispositivos para asegurar que la política esté vigente. Un administrador puede usar otras herramientas para evaluar la seguridad, intentando atacar dispositivos o servicios e informando al administrador si el ataque tuvo éxito.

Puesto que implica una proyección, la planeación de redes es compleja. Las herramientas de planeación están entre las más sofisticadas. Por ejemplo, existen herramientas que ejecutan algoritmos de programación lineal para ayudar a un administrador a optimizar la arquitectura de red o planear la administración del tráfico. También existen herramientas que ayudan a un administrador a evaluar la debilidad (por ejemplo, identificar sitios en la red donde dos o más fallas de hardware desconectarán a los usuarios de Internet).

Para resumir:

Existe una amplia variedad de herramientas de seguridad que ayudan a un administrador a configurar, medir, diagnosticar y analizar las redes.

30.6 Aplicaciones de administración de redes

La mayoría de las herramientas antes descritas operan a través de una red. Es decir, un administrador permanece en una sola ubicación y usa la tecnología de redes para comunicarse con un elemento de red dado. Lo sorprendente es que la administración de redes no se define como parte integral de los protocolos de capas inferiores, sino que los protocolos que se utilizan para monitorear y controlar los dispositivos de red operan a nivel de aplicación. Cuando un administrador necesita interactuar con un dispositivo de hardware específico, ejecuta un programa de aplicación que actúa como cliente y un programa de aplicación en el dispositivo de red actúa como servidor. El cliente y el servidor usan los protocolos de transporte convencionales como UDP o TCP para interactuar. Además, en vez de crear una red independiente, la mayoría de los administradores envían el tráfico administrativo a través de la red de producción.

Para evitar confusión entre los programas de aplicación que invocan los usuarios y las aplicaciones que se reservan para los administradores de red, los sistemas de administración de redes evitan los términos *cliente* y *servidor*. En su lugar, la aplicación cliente que se ejecuta en la computadora del administrador se conoce como *administrador* y el servidor que se ejecuta en un dispositivo de red se conoce como *agente*.[†]

El uso de protocolos de transporte convencionales para transportar el tráfico administrativo puede parecer inefectivo debido a que los problemas en el software de protocolo, las rutas mal configuradas o las fallas del hardware utilizado pueden evitar que los paquetes vayan desde o hacia un dispositivo, haciendo imposible la labor de controlar un dispositivo mientras ocurren las fallas. Algunos operadores de red instalan hardware independiente para administrar los dispositivos altamente imprescindibles (por ejemplo, un módem de marcación telefónica se puede conectar directamente a un enrutador de alta velocidad como respaldo para que un administrador lo utilice en caso de que la red principal sea intransitable). En la práctica, dichos sistemas raras veces se necesitan. Usar un protocolo de aplicación para la administración de redes funciona bien por tres razones. Primera, los sistemas de redes modernos son en extremo confiables. Segunda, en casos en los que una falla del hardware evita la comunicación, por lo general un administrador puede comunicarse con dispositivos que sigan funcionando y usar el éxito o fracaso para ayudar a localizar el problema. Tercera, usar protocolos de transporte convencionales significa que los paquetes de un administrador estarán sujetos a las mismas condiciones que el tráfico normal. Por lo tanto, si los retrasos son elevados un administrador lo descubrirá de inmediato.

[†]Aunque seguiremos la convención de usar *administrador* y *agente*, el lector debe tener en cuenta que funcionan exactamente igual que las demás aplicaciones cliente y servidor.

30.7 Protocolo simple de administración de redes

El protocolo estándar utilizado para la administración de redes se conoce como el *protocolo simple de administración de redes (SNMP)*; la versión estándar es la versión 3, que se escribe como *SNMPv3*. El protocolo SNMP define con exactitud la forma en que un administrador se comunica con un agente. Por ejemplo, SNMP define el formato de las solicitudes que envía un administrador a un agente y el formato de las respuestas que devuelve el agente. Además, SNMP define el significado exacto de cada posible solicitud y respuesta. En especial, SNMP especifica que un mensaje SNMP se codifica usando un estándar conocido como *notación de sintaxis abstracta.I* (*ASN.I*).

Aunque los detalles completos de la codificación ASN.1 están más allá del alcance de este libro, un ejemplo simple ayudará a explicar la codificación: considere enviar un entero entre un agente y un administrador. Para alojar valores grandes sin desperdiciar espacio en cada transferencia, ASN.1 usa una combinación de longitud y valor para cada objeto que se va a transferir. Por ejemplo, un entero entre 0 y 255 puede transferirse en un solo octeto. Los enteros en el rango 256 a 65535 requieren dos octetos mientras que los enteros más grandes requieren tres o más octetos. Para codificar un entero, ASN.1 envía un par de valores: una longitud L seguida de L octetos que contienen el entero. Para que los mensajes puedan incluir enteros de cualquier tamaño, ASN.1 permite que la longitud ocupe más de un octeto. Por lo general las longitudes extendidas no son necesarias para los enteros que se utilizan con sistemas de redes comunes. La figura 30.3 ilustra la codificación.

Entero decimal	Equivalente hexadecimal	Byte de longitud	Bytes de valor (en hexadecimal)
27	1B	01	1B
792	318	02	03 18
24,567	5FF7	02	5F F7
190,345	2E789	03	02 E7 89

Figura 30.3 Ejemplos de la codificación ASN.1 para enteros.

30.8 Paradigma de obtener y almacenar de SNMP

El protocolo SNMP no define un conjunto extenso de comandos, sino que usa un *paradigma de obtener y almacenar* en el que hay dos operaciones básicas: *obtener*, que se usa para obtener un valor de un dispositivo y *almacenar*, que se usa para establecer un valor en el dispositivo. Cada objeto que puede obtenerse o almacenarse recibe un nombre único; un comando que especifica una operación *obtener* o *almacenar* debe especificar el nombre del objeto.

Resulta obvio cómo pueden usarse las operaciones *obtener* para monitorear un dispositivo o para obtener su estatus: hay que definir un conjunto de objetos de estatus. Para obtener la información del

estatus, un administrador obtiene el valor asociado con un objeto dado. Por ejemplo, es posible definir un objeto que cuente el número de tramas que descarta un dispositivo debido a que la suma de verificación de las tramas es incorrecta. El dispositivo debe diseñarse para incrementar el contador cada vez que se detecte un error en la suma de verificación. Un administrador puede usar SNMP para obtener el valor asociado con el contador y determinar si están ocurriendo errores de suma de verificación.

Tal vez no parezca obvio usar el paradigma de obtener y almacenar para controlar un dispositivo, ya que las operaciones de control se definen como el efecto secundario de almacenar en un objeto. Por ejemplo, SNMP no incluye comandos independientes para *restablecer* un contador de errores de suma de verificación o para *reiniciar* un dispositivo. En el caso del contador de errores de suma de verificación, almacenar un cero en el objeto es algo intuitivo debido a que restablece un contador en cero. Sin embargo, para operaciones como el reinicio, un agente SNMP debe programarse para interpretar una solicitud de *almacenar* y ejecutar la secuencia correcta de operaciones para lograr el efecto deseado. De esta forma, el software de SNMP podría definir un objeto de reinicio y especificar que al guardar cero en el objeto, el sistema se reiniciará. Claro que los objetos SNMP son virtuales en el sentido en que el dispositivo utilizado no los implementa de manera directa. En su lugar, un agente recibe solicitudes y realiza acciones que corresponden a cada operación *obtener* o *almacenar*. Para resumir:

SNMP usa el paradigma de obtener y almacenar para la interacción entre un administrador y un agente. Un administrador obtiene valores para determinar el estatus de un dispositivo. Las operaciones que controlan el dispositivo se definen como los efectos secundarios de almacenar en objetos.

30.9 La MIB de SNMP y los nombres de objetos

Cada objeto al que SNMP tiene acceso debe definirse y recibir un nombre único. Además, los programas administrador y agente deben estar de acuerdo en los nombres y los significados de las operaciones *obtener* y *almacenar*. En sentido colectivo, el conjunto de todos los objetos a los que SNMP tiene acceso se conoce como *base de información de administración (MIB)*.

De hecho, la definición de una MIB no está vinculada de manera directa con SNMP, sino que este estándar sólo especifica el formato del mensaje y describe cómo se codifican los mensajes, mientras que un estándar independiente especifica las variables de la MIB junto con el significado de las operaciones *obtener* y *almacenar* en cada variable. De hecho, hay documentos de estándares independientes que especifican las variables de MIB para cada tipo de dispositivo.

Los objetos en una MIB se definen con el esquema de nomenclatura ASN.1, que asigna a cada objeto un prefijo largo que garantiza que el nombre será único. Por ejemplo, un entero que cuenta el número de datagramas IPv4 que recibe un dispositivo se llama:

iso.org.dod.internet.mgmt.mib.ip.ipEntRecibe

Además, cuando el nombre del objeto se representa en un mensaje SNMP, a cada parte del nombre se le asigna un entero. Por lo tanto, en un mensaje SNMP, el nombre de *ipEntRecibe* es:

1.3.6.1.2.1.4.3

30.10 Las variables de MIB

Puesto que SNMP no especifica un conjunto de variables de MIB, el diseño es flexible. Pueden definirse y estandarizarse nuevas variables MIB según sea necesario, sin cambiar el protocolo básico. Lo que es más importante, la separación del protocolo de comunicación de la definición de objetos permite que cualquier grupo defina variables de MIB según sea necesario. Por ejemplo, cuando se diseña un nuevo protocolo, el grupo que lo crea puede definir variables de MIB que se utilizarán para monitorear y controlar el software de protocolo. De manera similar, cuando un fabricante crea un nuevo dispositivo de hardware, puede especificar las variables de MIB a utilizar para monitorear y controlar el dispositivo.

Como los diseñadores originales tenían pensado, se crearon muchos conjuntos de variables de MIB. Por ejemplo, hay variables de MIB que corresponden a protocolos como UDP, TCP, IP y ARP, así como variables de MIB para hardware de red como Ethernet. Además, los grupos definieron MIB genéricas para dispositivos de hardware como enruteadores, conmutadores e impresoras.[†]

30.11 Variables de MIB que corresponden a arreglos

Además de las variables simples como los enteros que se usan en los contadores, una MIB puede incluir variables que definan tablas o arreglos. Dichas definiciones son útiles debido a que corresponden a la implementación de la información en un sistema de computadoras. Por ejemplo, considere una tabla de reenvío de IP. En la mayoría de las implementaciones, la tabla de reenvío puede verse como un arreglo donde cada entrada contiene una dirección de destino y un siguiente salto que se utiliza para llegar a esa dirección.

A diferencia de un lenguaje de programación convencional, ASN.1 no incluye una operación de indexación, sino que las referencias indexadas son implícitas, de modo que el emisor debe saber que el objeto al que se está haciendo referencia es una tabla y debe adjuntar la información de indexación en el nombre del objeto. Por ejemplo, la variable de MIB:

prefijo MIB estándar.ip.tablaEnrutamientoIp

corresponde a una tabla de reenvío de IP;[‡] cada una de sus entradas contiene varios campos. En teoría, la tabla se indexa mediante la dirección IP de un destino. Para obtener el valor de un campo específico en una entrada, un administrador especifica un nombre de la forma:

prefijo MIB estándar.ip.tablaEnrutamientoIp.entradaRutaIP.campo.dirIPdest

donde *campo* corresponde a uno de los campos válidos de una entrada y *dirIPdest* es una dirección IPv4 de 4 octetos que se utiliza como índice. Por ejemplo, el campo *sigSaltoRutaIP* corresponde al siguiente salto en una entrada. Cuando se convierte a la representación de enteros, la solicitud para un siguiente salto se vuelve:

1.3.6.1.2.1.4.21.1.7.destino

[†] Además de las variables de MIB genéricas que trabajan con cualquier dispositivo, muchos distribuidores definen variables de MIB específicas para su hardware o software.

[‡] Recuerde que una tabla de reenvío se conocía originalmente como tabla de enruteamiento; el cambio en la terminología ocurrió en la década de 2000.

donde *I.3.6.1.2.1* es el prefijo MIB estándar, *4* es el código para *ip*, *21* es el código de *tablaEnrutamientoIP*, *1* es el código para *entradaRutaIP*, *7* es el código para el campo *sigSaltoRutaIP* y *destino* es el valor numérico para la dirección IPv4 de un destino. Para resumir:

Aunque ASN.1 no cuenta con un mecanismo de indexación, las variables de MIB pueden corresponder a tablas o arreglos. Para emular una tabla o un arreglo con una variable ASN.1, el índice de una entrada se codifica adjuntándolo al nombre de la variable. Cuando el software de agente encuentra un nombre que corresponde a una tabla, éste extrae y utiliza la información del índice para seleccionar la entrada correcta en la tabla.

30.12 Resumen

Un administrador de red es una persona que monitorea y controla los sistemas de hardware y software que constituyen una intranet. El modelo FCAPS define los cinco aspectos básicos de la administración de redes como la detección de fallas, la configuración, la contabilidad, el análisis de rendimiento y la seguridad. Existe una variedad de herramientas para ayudar a un administrador a realizar funciones administrativas. La mayoría de las herramientas sólo proporcionan la administración de elementos. Como resultado, un administrador de red debe manejar manualmente las tareas entre dispositivos, interrogando y controlando un elemento a la vez.

Puesto que el software de administración de redes usa el modelo cliente-servidor, el software requiere dos componentes. El componente que se ejecuta en la computadora de un administrador y actúa como cliente se conoce como *administrador*, mientras que el componente que se ejecuta en un dispositivo de la red y actúa como servidor se conoce como *agente*.

El *protocolo simple de administración de redes (SNMP)* es el protocolo de administración de redes estándar que se usa en Internet. SNMP define el formato y el significado de los mensajes que intercambian un administrador y un agente. En vez de definir muchas operaciones, SNMP usa el paradigma de *obtener* y *almacenar* en el que un administrador envía solicitudes para obtener valores de o almacenar valores en variables. Todas las operaciones se definen como efectos secundarios de las operaciones *almacenar*.

SNMP no define el conjunto de variables que pueden usarse, sino que las variables y sus significados se definen en estándares independientes. Esto permite que los grupos definan un conjunto diferente de variables de MIB para cada dispositivo o protocolo de hardware. Los nombres de las variables de MIB se asignan con base en el estándar ASN.1; todas las variables de MIB tienen nombres de ASN.1 jerárquicos y extensos, los cuales se traducen a una representación numérica más compacta para la transmisión. ASN.1 no incluye tipos de datos agregados como las tablas o arreglos; tampoco incluye un operador de subíndice. En su lugar, para que una variable de MIB emule una tabla o un arreglo, ASN.1 extiende el nombre de la variable adjuntando la información del índice.

EJERCICIOS

- 30.1** Mencione un ejemplo de un mecanismo de protocolo que oculte un error.
- 30.2** Si un usuario se queja de que no puede acceder a un servicio dado, ¿qué aspectos de FCAPS podría involucrar potencialmente la queja?
- 30.3** Si falla un firewall, ¿bajo qué aspecto de FCAPS se encuentra la situación? ¿Por qué?
- 30.4** Busque dos ejemplos de elementos administrables aparte de los que se enlistan en la figura 30.2.
- 30.5** ¿Qué es un analizador de protocolos?
- 30.6** ¿Qué es lo que una herramienta de análisis de flujo ayuda a que un administrador entienda?
- 30.7** ¿Qué términos usa el software de administración de redes en vez de cliente y servidor?
- 30.8** ASN.1 define el formato exacto de un entero. ¿Por qué el estándar ASN.1 no sólo indica que cada entero es un valor de 32 bits?
- 30.9** Se ha argumentado que no debemos usar una red para depurar un problema en esa misma red. ¿Por qué SNMP usa la misma red que está depurando?
- 30.10** Escriba un programa que lea un entero con un tamaño arbitrario en decimal, codifique el entero en el formato que se ilustra en la figura 30.3 e imprima el resultado.
- 30.11** ¿Cuáles son las dos operaciones básicas que utiliza SNMP?
- 30.12** Descargue el software administrador SNMP gratuito y trate de contactar a un dispositivo, como una impresora.
- 30.13** ¿SNMP define un nombre para cada posible variable de MIB? Explique.
- 30.14** ¿Cuál es la principal ventaja de adjuntar información de índice a un nombre en vez de usar un arreglo convencional indexado por enteros?
- 30.15** Averigüe cómo ASN.1 codifica nombres y valores. Escriba un programa de computadora para codificar y decodificar nombres ASN.1 como el que se asigna a *ipEntRecibe*.

Contenido del capítulo

- 31.1 Introducción, 545
- 31.2 El despliegue publicitario y la realidad, 545
- 31.3 Motivación para un nuevo enfoque, 546
- 31.4 Organización conceptual de un elemento de red, 548
- 31.5 Módulos del plano de control y la interfaz de hardware, 549
- 31.6 Un nuevo paradigma: las redes definidas por software, 550
- 31.7 Preguntas sin responder, 551
- 31.8 Controladores compartidos y conexiones de red, 552
- 31.9 Comunicación SDN, 553
- 31.10 OpenFlow: un protocolo de controlador a elemento, 554
- 31.11 Motores de clasificación en los commutadores, 555
- 31.12 TCAM y clasificación de alta velocidad, 556
- 31.13 Clasificación entre varias capas de protocolos, 557
- 31.14 Tamaño de TCAM y la necesidad de patrones múltiples, 557
- 31.15 Elementos que OpenFlow puede especificar, 558
- 31.16 Reenvío de IP tradicional y extendido, 559
- 31.17 Ruta de extremo a extremo con MPLS usando la capa 2, 560
- 31.18 Creación de reglas dinámicas y control de flujos, 561
- 31.19 Un modelo de canalización para tablas de flujo, 562
- 31.20 Efecto potencial de SDN en los fabricantes de red, 563
- 31.21 Resumen, 564

31

Redes definidas por software (SDN)

31.1 Introducción

El capítulo anterior introduce el tema de la administración de redes y presenta el modelo de administración FCAPS. El capítulo describe la idea general de la administración de elementos y explica el paradigma utilizado por el *protocolo simple de administración de redes*, SNMP. Este capítulo concluye la explicación de la administración de redes mediante un enfoque en una nueva tecnología que ha generado un entusiasmo increíble. El capítulo presenta la justificación, sondea la metodología general y explica la tecnología subyacente.

31.2 El despliegue publicitario y la realidad

Al igual que con muchas nuevas tecnologías, se han utilizado estrategias de marketing para generar entusiasmo. Por desgracia, el despliegue publicitario fijó expectativas irracionalmente elevadas. Por ejemplo, el autor escuchó a los anunciantes afirmar que el nuevo enfoque eliminará todo el error humano, garantizará la completa seguridad de la intranet de una empresa, asegurará que las contraseñas de los usuarios sean uniformes entre todos los dispositivos, mejorará la eficiencia del enrutamiento en general y reducirá entre tres y cinco veces el costo de operar una red. Si existiera una tecnología de administración de redes que pudiera lograr todo lo anterior, sin duda sería un milagro. Desde luego que no existe dicha tecnología, en especial una que haga muchas mejoras importantes y al mismo tiempo elimine del 66 al 80% de los costos de operación.

Tomaremos una postura equilibrada. Por una parte consideraremos las motivaciones, analizaremos la nueva tecnología y evaluaremos las ventajas potenciales. Por otra parte consideraremos algunas de las concesiones y desventajas.

31.3 Motivación para un nuevo enfoque

¿Por qué cambiar el paradigma de la administración de redes? Hay varias respuestas. Algunos se enfocan en los errores que introduce el ser humano cuando configura sistemas de redes. Otros se enfocan en las limitaciones de la tecnología actual. Las próximas secciones explican las siguientes motivaciones:

- Generalizar la administración de elementos para la administración de redes
- Pasar de estándares propietarios a abiertos
- Automatizar y unificar la configuración en toda la red
- Cambiar del control por capa al control entre capas
- Adaptar la virtualización utilizada en los centros de datos

31.3.1 Generalizar la administración de elementos

Una de las principales limitaciones de los sistemas actuales de administración de redes recae en su metodología fundamental. Un administrador tiene que lidiar con cada uno de los elementos. Por consiguiente, éste puede configurar un enrutador, medir un circuito rentado o detectar la falla de un interruptor.

Los críticos argumentan que la administración de elementos es una idea de bajo nivel y debe ser reemplazada por un sistema que permita a un administrador emitir comandos que controlen toda una red (es decir, todos los enrutadores, todos los enlaces y todos los commutadores). La idea no es simplemente hacer que todos los dispositivos trabajen de manera idéntica, sino hacer que trabajen en conjunto para obtener una política de administración de alto nivel.

31.3.2 Pasar de estándares propietarios a abiertos

Los dispositivos de red actuales incluyen una interfaz de administración específica del fabricante, quien controla el conjunto de comandos que un administrador puede emitir, así como la sintaxis utilizada para expresar dichos comandos. Aun cuando un fabricante implementa un estándar como SNMP, a menudo opta por incluir extensiones especiales que sólo se aplican a su hardware.

Los críticos argumentan que aunque los sistemas de administración específicos del fabricante pueden ser convenientes, permitir que los fabricantes determinen las características propietarias especiales hace que se continúe con una metodología en la que se administran los elementos en forma individual. Además, los críticos señalan que si todos los fabricantes acordaran implementar un estándar abierto, podría idearse un sistema de administración para coordinar la operación de varios dispositivos de diferentes marcas.

31.3.3 Automatizar y unificar la configuración

Uno de los principales problemas con la administración de redes actual surge debido al error humano. Puesto que los seres humanos deben configurar elementos individuales de red, y debido a que la configuración implica muchos detalles, los errores son comunes. Lo que es más importante, la configuración de un dispositivo que se conecta fuera de una organización puede diferir de la configuración utilizada para un dispositivo interno, que requiere que un administrador personalice la configuración según la ubicación, el rol y el tipo de dispositivo.

Los críticos argumentan que un sistema automatizado permitiría seguir una política general y generar una configuración apropiada para cada elemento de red. El resultado será la consistencia en toda la red. A medida que el tamaño de una intranet aumenta, resulta difícil (por no decir imposible) que los administradores logren dicha consistencia por medio de la configuración manual de los elementos. Lo interesante es que la escala ocurre debido a que las redes tienen muchos dispositivos además de hosts y enrutadores (por ejemplo, puntos de acceso, conmutadores, firewalls, redes VPN y servidores).

31.3.4 Cambiar del control por capa al control entre capas

La administración de redes tradicional divide las responsabilidades de acuerdo con las capas de protocolos. Un administrador asume la responsabilidad de los servicios de la capa 2, como las conexiones de conmutadores, las redes VLAN y las redes con puentes. Otro administrador se enfoca en los servicios de la capa 3, como la asignación de direcciones IP y subredes, las rutas IP y MPLS.

Los críticos argumentan que al dividir la administración de acuerdo con las capas se pierden oportunidades mayores. Por ejemplo, si un sistema administrara las capas 2 y 3 juntas, se podrían crear redes VLAN independientes para ciertos tipos de tráfico IP o asignar un túnel MPLS a una ruta de la capa 2.

31.3.5 Adaptar la virtualización de los centros de datos

Un centro de datos consiste en un conjunto de máquinas físicas conectadas en una red. A diferencia de las computadoras convencionales, cada una de las cuales ejecuta un sistema operativo y a las que se les asigna una dirección de Internet (IPv4 e IPv6), la mayoría de los centros de datos usan la tecnología de *virtualización* como VMWare. Con la virtualización, una máquina física emula varias *máquinas virtuales* (VM). Cada VM ejecuta una copia de un sistema operativo (como MS Windows, Mac OS-X o Linux) y cada una necesita una dirección IP única (o tal vez unas direcciones IPv4 e IPv6 únicas). El problema surge debido a que una VM puede *migrar* (es decir, moverse) de una computadora física a otra.

Cuando una VM se mueve, hay que reconfigurar los dispositivos de red para entregar paquetes en la nueva ubicación. Puesto que el software controla la migración de las VM, los cambios pueden ocurrir con frecuencia y una migración dada puede ocurrir en decenas de milisegundos. Los críticos señalan que las herramientas convencionales de administración de redes no están equipadas para manejar cambios de ruta frecuentes de alta velocidad. Afirman que se necesita nueva tecnología que pueda coordinarse con el software de migración de VM de modo que la red se reconfigure con rapidez cuando ocurra una migración de VM.

31.4 Organización conceptual de un elemento de red

Como antecedente, considere los productos de red comerciales. Los ingenieros que diseñan dichos dispositivos dividen la arquitectura interna en dos partes conceptuales:

- Plano de control
- Plano de datos

Plano de control. El plano de control en un dispositivo de red proporciona funcionalidad administrativa que permite a un administrador de red autorizado configurar, monitorear, apagar, reiniciar o realizar otras tareas en el dispositivo. Incluso en dispositivos de gama alta, las funciones del plano de control se implementan en software y se ejecutan en procesadores integrados independientes que son relativamente lentos en comparación con el hardware que procesa los paquetes. La baja velocidad no es un problema debido a que las funciones de control se realizan con poca frecuencia y muchas funciones del plano de control implican la interacción humana.

Plano de datos. El plano de datos en un sistema de red proporciona la funcionalidad necesaria para procesar paquetes e incluye el hardware de interfaz de red, las instalaciones de ingreso de paquetes, los mecanismos de reenvío de paquetes y las instalaciones de salida de paquetes. Puesto que son responsables de manejar los paquetes a las velocidades de la línea, los módulos del plano de datos se implementan por lo general en el hardware y están altamente optimizados.[†] Por ejemplo, el plano de datos de un conmutador Ethernet de un gigabit que tiene 48 puertos debe ser capaz de manejar un paquete que llega y sale en cada puerto al mismo tiempo, lo que significa que el plano de datos del conmutador necesita una capacidad total de 96 gigabits por segundo.

Desde luego que el plano de control y el plano de datos en un sistema de redes deben estar diseñados para trabajar en estrecha colaboración. El plano de datos está configurado para enviar paquetes de administración de red hacia el módulo del plano de control para su procesamiento. Cuando recibe un comando del administrador de red, el software del plano de control interpreta el comando, recalcula una nueva configuración para el dispositivo y carga la configuración en el hardware del plano de datos. Dependiendo del comando, el software del plano de control puede tardar varios segundos en calcular una nueva configuración. Mientras tanto, el plano de datos sigue operando microsegundo tras microsegundo mediante el reenvío de paquetes. El hardware se organiza de tal forma que, una vez que el plano de control decida realizar un cambio, la configuración pueda cargarse casi al instante. La figura 31.1 ilustra la organización.

Como lo indica la figura, el plano de datos determina la velocidad a la que el dispositivo puede procesar los paquetes. De esta forma, al planear un producto con mayor velocidad, un fabricante debe concentrarse en el plano de datos, ya que la mayor parte del plano de control no cambia debido a que el cálculo de una tabla de reenvío no depende de la velocidad de las redes involucradas. Es decir, calcular una tabla de reenvío para usarla con redes de 1 Gbps es lo mismo que calcular una tabla de reenvío para usarla con redes de 10 Gbps.

[†] Los dispositivos más pequeños, como los enrutadores inalámbricos que se utilizan en las casas, son una excepción: dichos dispositivos tienen un solo procesador que se encarga de las funciones tanto del plano de control como del plano de datos.

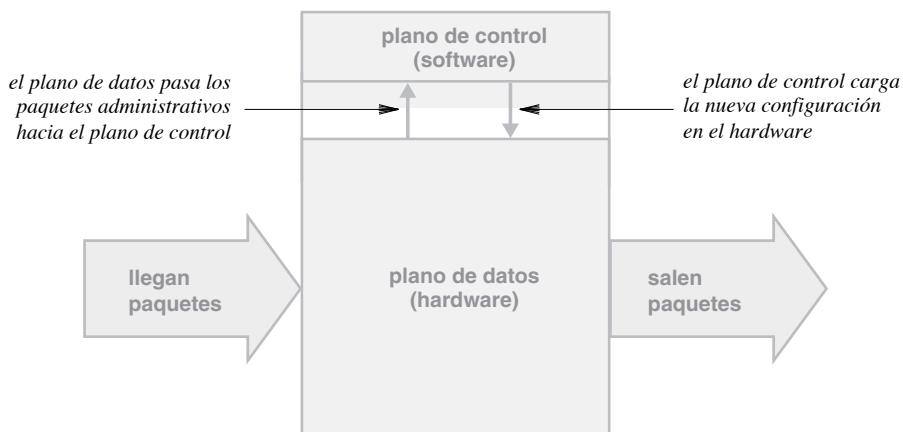


Figura 31.1 La división conceptual de un sistema de red en planos de datos y de control, donde el plano de datos se encarga del procesamiento de todos los paquetes.

31.5 Módulos del plano de control y la interfaz de hardware

En la mayoría de los elementos de red, el plano de control incluye varios módulos de software, cada uno de los cuales ofrece un mecanismo que un administrador puede usar para acceder al sistema. Hay tres interfaces administrativas que se han vuelto bastante populares. En un sistema de red común, un administrador puede elegir una de las tres:

- Acceso a la Interfaz de línea de comandos (CLI) mediante *ssh*
- Acceso a la interfaz web (gráfica) con un navegador (HTTP)
- Acceso a un agente SNMP mediante aplicaciones de administración de SNMP

Aunque todos los módulos del plano de control ofrecen un conjunto de funciones básicas, un fabricante puede optar por ofrecer características en algunas interfaces que no estén disponibles en otras. En especial, para diferenciar sus productos de los competidores, los fabricantes a menudo incluyen comandos de CLI que dan a un administrador capacidades adicionales o facilitan en gran medida la coordinación de varios productos del mismo fabricante.

Para dar soporte a varios módulos, los ingenieros generalmente crean un mecanismo interno común que proporciona una interfaz para el hardware a utilizar. Cada módulo del plano de control invoca la interfaz común para realizar operaciones, como los cambios en la tabla de reenvío. La interfaz es interna (es decir, no puede verse fuera del dispositivo) y sólo puede utilizarse mediante los módulos del plano de control que el fabricante vende. La figura 31.2 ilustra el diseño.

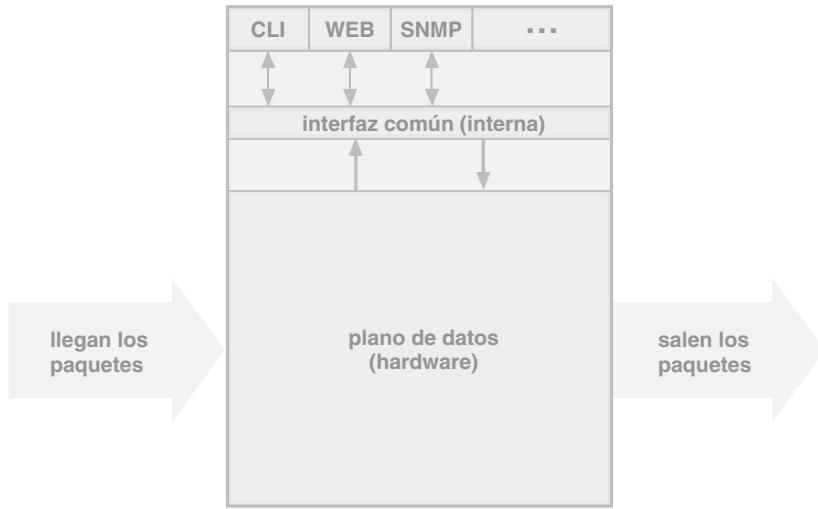


Figura 31.2 La estructura interna de un elemento de red con varios módulos del plano de control y una interfaz interna común.

31.6 Un nuevo paradigma: las redes definidas por software

Conocido como *redes definidas por software* (*SDN*), el nuevo paradigma traslada la mayoría de las funciones del plano de control, del elemento hacia un *controlador* asociado. Aunque son posibles varias implementaciones, la configuración de controlador más común consiste en una PC que ejecuta Linux.

Para que un controlador externo pueda modificar y monitorear el elemento de red, se agrega un módulo de control adicional. El nuevo módulo es bastante simple: en vez de proporcionar una interfaz humana para los administradores, el módulo simplemente acepta comandos de bajo nivel y los pasa al hardware. En esencia, la idea es permitir que el software de administración que se ejecuta en el controlador externo configure las tablas de reenvío en el plano de datos. Para usar el nuevo paradigma, no se usan otros módulos del plano de control. La figura 31.3 ilustra el uso de un controlador externo.

En la figura, el cuadro que se usa para representar el módulo SDN es más delgado que los cuadros utilizados para representar otros módulos del plano de control. La opción indica que la funcionalidad SDN dentro de un elemento de red es mucho menor que la funcionalidad que se encuentra en el software de plano de control tradicional. De hecho, la interfaz SDN es sencilla, ya que el módulo simplemente recibe la información de configuración del plano de datos del controlador externo y usa la interfaz interna para configurar el hardware.

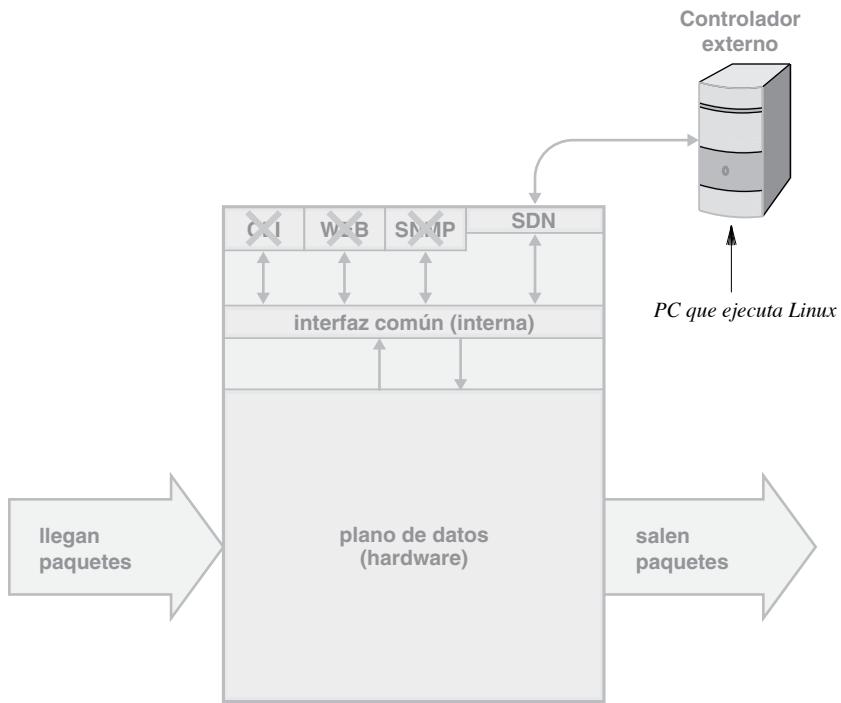


Figura 31.3 El paradigma SDN con un controlador externo que configura un elemento de red.

Podemos resumir esto de la siguiente manera:

Las redes definidas por software extraen las aplicaciones de administración (es decir, las funciones del plano de control) de cada elemento de red y las colocan en un controlador externo. Por lo general, un controlador externo consiste en una PC que ejecuta Linux.

31.7 Preguntas sin responder

Nuestra descripción breve genera muchas preguntas sobre SDN. ¿Cuál es la conexión física entre un controlador externo y un elemento de red? ¿Qué protocolo se usa para comunicarse entre un controlador externo y un elemento de red que se esté controlando?

Además de las preguntas técnicas, surgen preguntas más amplias. ¿Qué software administrativo ejecuta un controlador externo y dónde obtiene uno dicho software? ¿Al agregar un controlador externo a cada elemento de red aumenta el costo total de una red? y de ser así, ¿costará más implementar una intranet basada en SDN que una creada a partir de elementos de red tradicionales? ¿Cómo afectarán

las SDN a la industria y a los fabricantes de redes? El resto del capítulo considera cada una de estas preguntas.

31.8 Controladores compartidos y conexiones de red

Dijimos que cada elemento de red tiene un controlador externo. Sin embargo, en una red real es costoso e innecesario usar un controlador físico independiente para cada elemento de red. Para entender la situación, considere la intranet de una empresa (por ejemplo, una corporación de gran tamaño).

Agregar un controlador independiente a cada elemento de red es en extremo costoso, ya que una intranet corporativa no consiste en unos cuantos commutadores grandes interconectados por unos cuantos enruteadores grandes, sino que la mayoría de las intranets contienen muchos dispositivos de red pequeños, como firewalls, puntos de acceso inalámbricos y commutadores de acceso. Incluso los teléfonos IP se suelen incluir entre los dispositivos de la intranet. Los defensores de las SDN sugieren que los controladores externos pueden consistir en equipos PC económicos, comunes y corrientes. Pero incluso las PC económicas aumentan los costos de manera considerable, ya que el costo de un controlador externo puede ser igual o mayor que el costo de un dispositivo de red pequeño.

Por fortuna es innecesario usar un controlador independiente para cada sistema de red, debido a que un solo controlador físico tiene suficiente potencia como para manejar varios dispositivos. A diferencia del procesamiento del plano de datos, que debe ocurrir microsegundo tras microsegundo, las aplicaciones administrativas sólo se ejecutan en forma ocasional (por ejemplo, cuando cambia algo o cuando un administrador necesita evaluar la red). Incluso los protocolos de enruteamiento sólo envían actualizaciones en forma periódica. Por lo tanto, la mayoría de las aplicaciones de administración de redes no imponen una carga pesada sobre la CPU. La carga de procesamiento es especialmente baja si hay que configurar o controlar varias copias físicas de un dispositivo dado. Por ejemplo, considere cambiar la configuración en un conjunto de puntos de acceso inalámbricos. Una vez que se calcula una configuración, es posible cargar esta misma configuración en todos los puntos de acceso.

¿Cuántos controladores externos necesita una intranet? Cuando los administradores van a elegir cómo implementar los controladores externos, varios factores influyen:

- El conjunto de aplicaciones administrativas que se usarán y la carga esperada que impone cada una sobre un controlador.
- La diversidad de la red medida según la variedad de elementos de red que contiene.
- El tamaño de la red, medido según el número de elementos de red y el alcance geográfico.
- El factor de replicación, que se mide como el número de copias físicas idénticas de un tipo dado de elemento de red.
- El agrupamiento físico de elementos de red (por ejemplo, varios elementos ubicados conjuntamente en un centro de datos, en comparación con los elementos individuales ubicados en varios pisos de un edificio de oficinas).

Para manejar todos los casos posibles, la SDN debe permitir que un controlador externo específico administre varios elementos de redes múltiples. Además, el software administrativo de los controladores externos necesita coordinarse de modo que las configuraciones en todos los elementos de red sean consistentes. Por lo tanto, los diseñadores imaginaron un sistema en el que una capa de controladores se comunican entre sí y cada controlador administra uno o más elementos de red. En terminología de SDN, cada controlador administra un *dominio de SDN*.[†] La figura 31.4 ilustra la arquitectura.

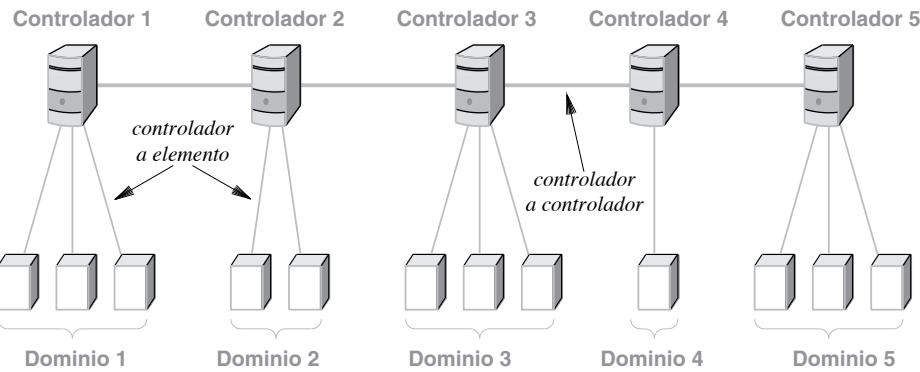


Figura 31.4 La disposición de los controladores de SDN en una intranet grande. Cada controlador administra un dominio de uno o más elementos de red.

Aunque la figura usa el mismo diagrama genérico para todos los elementos de red, la mayoría de las redes tienen varios tipos y tamaños de elementos. Por ejemplo, el elemento en el dominio 4 podría ser un enrutador grande, los elementos en el dominio 3 podrían ser puntos de acceso pequeños y los elementos en el dominio 1 podrían ser conmutadores de Ethernet.

La figura ilustra una idea importante: además de administrar los elementos de red, los controladores se comunican entre sí. Es decir, el software de un controlador no actúa solo, sino que el software en todos los controladores se comunica para asegurar que se comparta un solo conjunto de políticas administrativas y que las configuraciones en todos los elementos sean consistentes con las políticas generales. Más adelante en el capítulo aprenderemos que al permitir que el software establezca una configuración consistente a través de toda una intranet es posible crear rutas de reenvío en formas que los protocolos actuales no soportan.

31.9 Comunicación SDN

Como se muestra en la figura 31.4, un sistema SDN necesita manejar dos nuevos protocolos de comunicación:

- Comunicación de controlador a elemento
- Comunicación de controlador a controlador

[†] Los dominios de SDN no están relacionados con los dominios en el sistema de nombres de dominio.

SDN sigue el paradigma general que fue usado originalmente por SNMP. En SDN, los protocolos de administración operan en la capa de aplicación. Es decir, SDN asume que cada controlador externo y cada elemento de red tiene una pila TCP/IP convencional que el software de administración pueda usar para comunicarse con los elementos de red en el dominio de SDN y con otros controladores de SDN. SDN también sigue la metodología de SNMP en cuanto a la conectividad física: en vez de una *red de control* independiente que se utiliza para transportar tráfico de red, SDN envía tráfico administrativo a través de la misma red de producción que transporta el tráfico de datos.

Al igual que con SNMP, es peligroso usar la red de producción para el tráfico administrativo, ya que un problema en la red de producción puede evitar que un controlador de SDN corrija el problema. Sin embargo, los defensores de SDN dicen que los problemas potenciales tal vez no sean tan severos como con SNMP: un controlador de SDN que pierde una conexión de todas formas puede operar de manera autónoma, mientras que un administrador humano no puede administrar elementos a menos que la red entre el administrador y los elementos esté intacta. En teoría, el software SDN en un controlador aislado puede evaluar la situación y tratar de establecer rutas alternativas. Además, los defensores argumentan que el software que se ejecuta en los controladores puede operar muchas veces más rápido que un administrador humano.

En la práctica, el sueño del software SDN que se recupera de manera automática sigue sin alcanzarse. Aunque los investigadores continúan explorando posibilidades, no se han ideado sistemas prácticos.[†] En conclusión:

Aunque los defensores de SDN afirman que el software que se ejecuta en los controladores podrá manejar los problemas de red de una manera mucho mejor y más rápida que los administradores humanos, los sistemas de administración automatizados siguen siendo materia de investigación.

31.10 OpenFlow: un protocolo de controlador a elemento

¿Qué protocolo o protocolos deben usarse entre un controlador y un elemento de red? Hay un solo protocolo que surge como respuesta. Conocido como *OpenFlow*, el protocolo es una de las pocas tecnologías de SDN que ha recibido amplia aceptación en la industria de las redes.

Definido en un principio por investigadores de la Universidad de Stanford, el protocolo OpenFlow ahora es controlado por la Fundación de redes abiertas (www.opennetworking.org). La versión actual del protocolo es la 1.3. El estándar de OpenFlow especifica:

- El paradigma de comunicación
- La definición y clasificación de elementos
- El formato de los mensajes

Paradigma de comunicación. OpenFlow usa un paradigma de comunicación orientado a la conexión. Por lo general, el software en un controlador abre una conexión TCP hacia el software de SDN

[†] El proyecto GENI (entorno global de innovaciones de red) de la Fundación nacional de ciencia, de Estados Unidos, ha establecido un banco de pruebas para que los investigadores lo utilicen.

en un elemento de red. OpenFlow permite de manera específica la comunicación a través de un canal seguro. Aunque el estándar no restringe el mecanismo de seguridad, se recomienda SSL.[†] Usar TCP sobre SSL significa que un controlador dado puede abrir varias conexiones, una por cada elemento de red que se va a administrar.

Definición y clasificación de elementos. OpenFlow no sigue la metodología de SNMP de definir una MIB de gran tamaño, sino que se concentra en el reenvío de paquetes y usa un modelo de *tabla de flujo*. En una sección posterior se describen las tablas de flujo y la clasificación. Por ahora basta con entender que una tabla de flujo proporciona un conjunto de patrones y una acción para cada patrón. Cuando llega un paquete, el hardware del plano de datos realiza la coincidencia de patrones y aplica la acción del patrón que coincide. Uno de los posibles patrones es un *comodín* que coincide con cualquier paquete. Por lo tanto, es posible construir una tabla de flujo de modo que cada paquete coincida al menos con una entrada.

Formato de mensajes. Una sección posterior contiene los detalles sobre los mensajes de OpenFlow. Por el momento, basta con saber que OpenFlow especifica tanto el formato de los mensajes como el significado de cada campo. En especial, para asegurar la interoperabilidad a través de una variedad de arquitecturas, OpenFlow especifica que todos los valores enteros deben representarse en el orden Big Endian.

31.11 Motores de clasificación en los conmutadores

OpenFlow se diseñó de manera específica teniendo en mente los conmutadores de Ethernet. Por lo tanto, para entender la noción de una tabla de OpenFlow, necesitamos entender el hardware de un conmutador. El plano de datos en un conmutador de gama alta consiste en una pieza de hardware conocida como *motor de clasificación*. Cuando llega un paquete, el hardware lo reenvía al motor de clasificación, que examina el paquete (por lo general sólo el encabezado) y decide cómo reenviarlo. Los administradores de red y los usuarios nunca encuentran el motor de clasificación directamente, sino que la clasificación está oculta en el hardware del plano de datos. Cuando un administrador configura el conmutador, un módulo de software en el plano de control cambia las reglas de clasificación según corresponda. OpenFlow expone la clasificación al software de administración de redes y permite que un controlador externo cambie las reglas de clasificación de manera directa, como se indica en la figura 31.3.[‡]

Podemos pensar en la clasificación como un sistema de coincidencia de patrones implementado con hardware. En su interior, un motor de clasificación contiene un conjunto de patrones y una acción para cada patrón, según lo ilustrado en la figura 31.5. Antes de poder usar el clasificador, hay que precargar los patrones y las acciones. Al llegar un paquete, éste (por lo general, sólo los octetos del encabezado) se carga en una ubicación del clasificador. El clasificador busca los patrones, encuentra uno que coincide y ejecuta la acción correspondiente.

Un giro interesante hace a la clasificación extremadamente poderosa: los patrones pueden contener un valor “no importa” en algunos bits. En esencia, un patrón puede examinar algunos campos de encabezado e ignorar otros. Por ejemplo, para relacionar todos los paquetes de Ethernet que transportan IPv4, un patrón puede especificar que el campo de tipo de Ethernet de 16 bits debe coincidir con 0x0800 y especificar “no importa” para todos los demás campos.

[†] El capítulo 29 describe SSL.

[‡] Encontrará la figura 31.3 en la página 551.

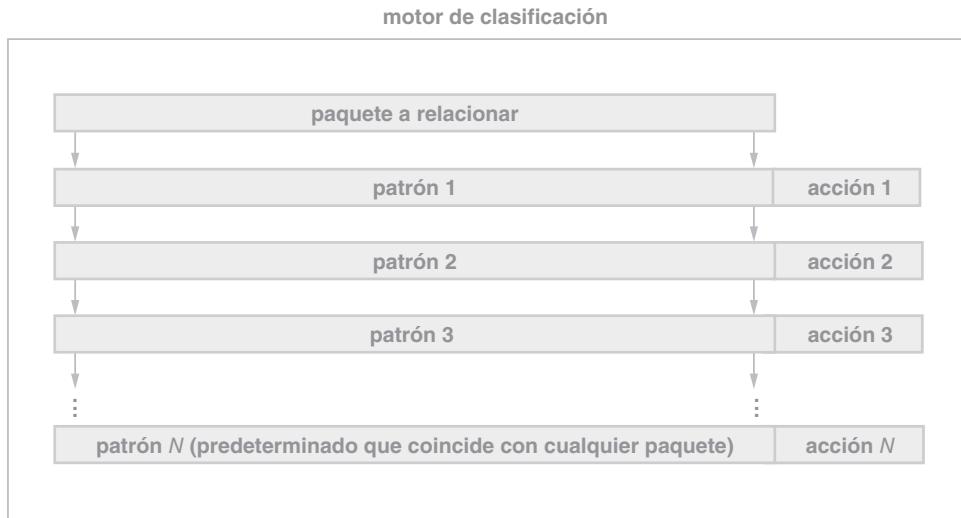


Figura 31.5 Ilustración de un motor de clasificación con un patrón que coincide con cualquier paquete utilizado para proporcionar una acción predeterminada.

31.12 TCAM y clasificación de alta velocidad

Si se implementa la clasificación en software, sólo puede verificarse un patrón a la vez. Por lo tanto, si un administrador especifica N patrones, un clasificador basado en software debe iterar a través de cada uno de los N patrones, para lo cual se requieren N pasos. Lo sorprendente es que un clasificador basado en hardware puede realizar las comparaciones en un paso.

La clave para comprender cómo funciona un clasificador basado en hardware es una tecnología que se conoce como *memoria ternaria de contenido direccional* (*TCAM*). A diferencia de una memoria de computadora convencional, la TCAM no sólo almacena valores, sino que cada celda de memoria contiene lógica que puede realizar una comparación a nivel de bits. Se usa el nombre *ternario* debido a que cada elemento en un patrón tiene uno de tres valores: 1, 0 o “no importa”.

En una TCAM, todas las celdas funcionan en paralelo para realizar una comparación de patrones. Por ejemplo, en la figura 31.5 podemos imaginar que el bloque que contiene un paquete y cada bloque que contiene un patrón forman una pieza de hardware. Podemos también imaginar que hay un conjunto de muchos cables tendidos desde el hardware que contiene un paquete hacia todos los patrones (en la figura, las flechas indican en dónde deberían estar los cables). Cada cable pasa el valor de un bit del paquete a los dispositivos que comparan los patrones.

Una vez que se coloca un paquete en el hardware de la TCAM, todos los dispositivos que comparan paquetes reciben una copia de los bits del paquete y todos actúan al mismo tiempo. Cada dispositivo compara su patrón con los bits del paquete (e ignora los bits “no importa”). Si ocurre una coincidencia, el dispositivo comparador selecciona la acción asociada (un valor entero) y el clasificador devuelve el

entero como resultado de la búsqueda. Si varios patrones coinciden, el clasificador cuenta con hardware que selecciona uno (por lo general, el de la posición más baja de la lista). Podemos resumir lo siguiente:

Puesto que usa hardware que opera en paralelo, la TCAM puede comparar un paquete con un conjunto de patrones en un solo ciclo. Es posible que haya un conjunto grande de patrones debido a que el tiempo de procesamiento no depende del número de patrones.

31.13 Clasificación entre varias capas de protocolos

El uso de la coincidencia de patrones explica una propiedad importante de los mecanismos de clasificación: pueden cruzar varias capas de la pila de protocolos. Por ejemplo, suponga que un administrador desea enviar todo el tráfico web por el puerto 34 del commutador. La acción es “enviar el paquete por el puerto 34”. Pero ¿qué patrón debe usarse? Hay tres elementos clave que determinan si un paquete contiene o no tráfico web:

- El tipo de trama (capa 2) especifica a *IP*
- El campo protocolo IP (capa 3) especifica a *TCP*
- El puerto de protocolo de destino TCP (capa 4) es 80

La idea importante es que aunque abarcan tres capas de la pila de protocolos, los elementos antes mencionados pueden combinarse en un solo patrón. Los defensores de las SDN afirman que la habilidad de un clasificador para abarcar varias capas de protocolos con un solo patrón hace a las SDN más poderosas que los mecanismos convencionales de reenvío de paquetes. Es posible usar la clasificación para manejar el reenvío de la capa 2, las VLAN de la capa 2 y el reenvío IP convencional, así como formas más complejas de reenvío. Por ejemplo, la clasificación puede enviar todo el tráfico VoIP hacia un puerto especificado en el commutador sin importar la dirección IP de destino, y la clasificación puede reenviar todos los paquetes de direcciones IP de origen específicas hacia un túnel de MPLS.

En conclusión:

Un solo patrón de clasificación puede verificar al mismo tiempo elementos de varias capas de la pila de protocolos.

31.14 Tamaño de TCAM y la necesidad de patrones múltiples

Desde luego que la clasificación tiene ciertas desventajas si se le compara con la metodología de demultiplexación que se utiliza en las pilas de protocolos convencionales. El hardware de TCAM que hace que la clasificación sea rápida resulta costoso. Además, TCAM usa una cantidad considerable de energía, además de generar una cantidad importante de calor. Como resultado, los diseñadores que desean

minimizar la energía que usa un commutador y el calor que genera ese commutador, deben minimizar la cantidad de TCAM que usa el motor de clasificación.

Para entender por qué una cantidad limitada de TCAM puede ser un problema, considere las reglas de clasificación necesarias para implementar el ejemplo del tráfico web antes mencionado. Se necesitan al menos dos reglas para manejar IPv4 e IPv6. Una regla IPv4 especificará lo siguiente:

- El campo tipo de trama de Ethernet contiene *0x0800* (IPv4)
- El campo *PROTO* del encabezado IPv4 contiene *6* (TCP)
- El campo puerto de protocolo de destino del encabezado TCP contiene *80* (web)

y una regla IPv6 especificará lo siguiente:

- El campo tipo de trama de Ethernet contiene *0x08DD* (IPv6)
- El campo *SIGUIENTE ENCABEZADO* en el encabezado base IPv6 contiene *6* (TCP)
- El campo puerto de protocolo de destino en el encabezado TCP contiene *80* (web)

Aunque se hacen cargo de los casos más simples, las dos reglas no cubren todas las posibilidades. Por ejemplo, un datagrama IPv6 podría contener un *encabezado de ruta* entre el encabezado base y el encabezado TCP. Puesto que un patrón de clasificación debe especificar ubicaciones de bits exactas dentro de un paquete, se necesita un patrón adicional para relacionar un paquete IPv6 que contiene un encabezado de ruta:

- El campo tipo de trama de Ethernet contiene *0x08DD* (IPv6)
- El campo *SIGUIENTE ENCABEZADO* en el encabezado base IPv6 contiene *43* (encabezado de ruta)
- El campo *SIGUIENTE ENCABEZADO* en el encabezado de ruta contiene *6* (TCP)
- El campo puerto de protocolo de destino en el encabezado TCP contiene *80* (web)

Por desgracia, un datagrama IPv6 puede contener encabezados de extensión adicionales, lo que significa que se necesitan patrones adicionales para cubrir todas las posibles disposiciones de los encabezados. En conclusión:

Puesto que un patrón de clasificación especifica posiciones exactas de bits, se necesitan varios patrones para alojar combinaciones de encabezados opcionales. Si un administrador especifica reglas para muchos flujos de tráfico, el conjunto de patrones puede hacerse muy grande.

31.15 Elementos que Op enFlow puede especificar

OpenFlow define un conjunto de campos de encabezado que pueden usarse en patrones de clasificación. Cuando un controlador externo envía un mensaje de OpenFlow con valores para campos específicos, un commutador crea un patrón de clasificación. La figura 31.6 enumera algunos ejemplos de los campos que pueden usarse. Para entender los elementos de una VLAN, recuerde que la mayoría de los commutadores usan el formato 802.1Q en forma interna (es decir, el commutador inserta una etiqueta de VLAN en el

encabezado de cada paquete Ethernet que llega antes de colocarlo en su memoria interna). Por lo tanto, los patrones de clasificación pueden examinar la etiqueta de VLAN junto con otros campos del encabezado.

Campo	Significado
Campos de la capa 2	
Ingress Port	Puerto del conmutador a través del cual llegó el paquete
Metadata	Campo de 64 bits de metadatos utilizados en la canalización
Ether src	Dirección de origen Ethernet de 48 bits
Ether dst	Dirección de destino Ethernet de 48 bits
Ether Type	Campo de tipo de Ethernet de 16 bits
VLAN id	Etiqueta de VLAN de 12 bits en el paquete
VLAN priority	Número de prioridad VLAN de 3 bits
ARP opcode	Código de operación de ARP de 8 bits
Campos de la capa 3	
MPLS label	Etiqueta MPLS de 20 bits
MPLS class	Clase de tráfico MPLS de 3 bits
IPv4 src	Dirección de origen IPv4 de 32 bits
IPv4 dst	Dirección de destino IPv4 de 32 bits
IPv6 src	Dirección de origen IPv6 de 128 bits
IPv6 dst	Dirección de destino IPv6 de 128 bits
IPv4 Proto	Campo de protocolo IPv4 de 8 bits
IPv6 Next Header	Campo siguiente encabezado IPv6 de 8 bits
TOS	Bits de tipo de servicio IPv4 o IPv6 de 8 bits
Campos de la capa 4	
TCP/UDP/SCTP src	Puerto de origen TCP/UDP/SCTP de 16 bits
TCP/UDP/SCTP dst	Puerto de destino TCP/UDP/SCTP de 16 bits
ICMP type	Campo de tipo ICMP de 8 bits
ICMP code	Campo de código ICMP de 8 bits

Figura 31.6 Ejemplos de campos de encabezado que pueden probarse con OpenFlow.

31.16 Reenvío de IP tradicional y extendido

Las direcciones que se enlistan en la figura 31.6 pueden consistir en una dirección completa o un prefijo. La habilidad de probar un prefijo IP significa que OpenFlow puede configurar todas las formas de reenvío que se encuentran en una pila de protocolos IP tradicional, incluyendo rutas específicas de host, rutas específicas de red y rutas específicas de subred. OpenFlow también puede manejar direcciones de multidifusión IP y difusión de IPv4. Por último, puesto que un patrón de clasificación puede tener todos sus bits establecidos en “no importa”, OpenFlow puede configurar una ruta predeterminada.

Lo sorprendente es que OpenFlow permite nuevas formas de reenvío IP que no son posibles con una pila de protocolos tradicional. Por ejemplo, OpenFlow puede usar la dirección IP de origen en un paquete al elegir una ruta, lo cual permite que se envíen datagramas IP del origen A a lo largo de una ruta diferente que la de los datagramas del origen B, incluso aunque todos los datagramas viajen al mismo destino. Lo que es más importante, pueden crearse reglas de clasificación en las que el siguiente salto dependa del contenido del datagrama, incluyendo los campos de encabezado de la capa 4. Podemos resumir lo siguiente:

Puesto que usa la clasificación, OpenFlow puede definir nuevas formas de reenvío IP en las que la elección de un siguiente salto depende de campos que no sean la dirección IP de destino. OpenFlow puede usar la dirección IP de origen o los campos en el encabezado de la capa 4, como los números de puertos TCP.

31.17 Ruta de extremo a extremo con MPLS usando la capa 2

¿Qué otras posibilidades ofrecen SDN y OpenFlow? El capítulo 19 describe MPLS y la idea de enviar tráfico a lo largo de una ruta MPLS. En las redes actuales, las rutas MPLS son *provisionadas* (es decir, se configuran en forma manual). OpenFlow hace posible que el software de administración cree rutas MPLS en forma dinámica. El uso de la clasificación significa que puede usarse cualquier campo en los encabezados de paquetes para decidir entre transferir un paquete a través de una ruta MPLS o usar el reenvío IP convencional. Además, debido a que el software en los controladores externos puede servir de coordinador, podemos imaginar un conjunto de controladores coordinándose entre sí para construir un túnel MPLS de *extremo a extremo* que atraviese varios comutadores en múltiples dominios SDN. Cada comutador se configura con reglas de reenvío que dirigen los paquetes a lo largo de la ruta.

Además de configurar rutas MPLS, es posible usar OpenFlow para configurar nuevas formas de reenvío. Por ejemplo, suponga que por cuestiones de seguridad un administrador desea controlar *todo* el tráfico que se origine en la computadora X. Las tablas de reenvío de IP tradicionales usan la dirección de destino para determinar cómo se envían los paquetes. Incluso si un administrador configura una regla de clasificación que use la dirección de IP de origen, IPv6 presenta un problema con la reenumeración automatizada, ya que si un atacante desencadena la reenumeración IP, los paquetes ya no seguirían la ruta de clasificación. OpenFlow hace posible especificar una regla de clasificación que no dependa de que la dirección IP permanezca constante. Esta regla usa el campo de dirección MAC en un paquete. Lo que es más importante, puesto que la tecnología SDN permite que los controladores sean los coordinadores, éstos pueden trabajar en conjunto para establecer una ruta de extremo a extremo que sólo examine la dirección MAC de un paquete. Decimos que los controladores pueden establecer una *ruta de capa 2 de extremo a extremo*.

¿Es importante poder usar la información de la capa 2? Puede serlo. Además de establecer una ruta de capa 2 de extremo a extremo, la tecnología SDN permite a un administrador crear redes de capa 2 grandes.[†] Por ejemplo, considere una organización con oficinas en Nueva York, Chicago y San Francisco. Un administrador puede usar SDN para configurar un conjunto de redes VLAN de capa 2 que abarquen las tres ubicaciones. Las computadoras de los representantes de ventas en las tres ubicaciones

[†] En el capítulo 33 veremos que las redes VLAN de capa 2 pueden considerarse una forma especial de redes superpuestas.

pueden compartir una VLAN y las computadoras de los ejecutivos en las tres ubicaciones pueden compartir otra VLAN. Una de las ventajas de usar la capa 2 está en su independencia del protocolo. Si se usa IP, un administrador debe garantizar que el reenvío IPv4 entre sitios sea consistente con el reenvío IPv6. Mantener los dos reenvíos consistentes puede ser difícil debido a que IPv4 ofrece direcciones de difusión de red que IPv6 no ofrece. Una VLAN de capa 2 coloca computadoras en varios sitios en la misma subred IP lógica, con lo cual elimina la necesidad de enrutadores y de una coordinación de la ruta. En conclusión:

Las tecnologías de SDN como OpenFlow hacen posible configurar rutas de capa 2 y redes VLAN de capa 2 que pasen por varios comutadores y varias ubicaciones.

Usar SDN para configurar una red de capa 2 tiene una ventaja en términos de seguridad. Para entender por qué, cabe mencionar que una intranet convencional tiene muchos componentes que controlan la exactitud en general. Se necesitan protocolos de enrutamiento para IPv4 e IPv6. IPv4 depende de ARP e IPv6 depende de la técnica de descubrimiento del vecindario. Cada protocolo puede ser vulnerable a los ataques. En comparación, una red de capa 2 puede configurarse una vez y no necesita protocolos de enrutamiento dinámico. Por lo tanto, hay menos formas de atacar la red.

31.18 Creación de reglas dinámicas y control de flujos

Nuestra descripción implica que deben instalarse todas las reglas de clasificación en el plano de datos antes de que lleguen los paquetes. Sin embargo, OpenFlow permite que el software que se ejecuta en un controlador instale o elimine las reglas de clasificación en forma dinámica. Lo más importante es que las reglas pueden depender de los paquetes que llegan. El mecanismo es simple y directo: instalar un patrón predeterminado con una acción que envíe el paquete al controlador SDN. Cada vez que llega un paquete que no coincide con ninguna de las reglas de clasificación existentes, éste se envía al software de SDN en el controlador. El software puede usar el contenido del paquete para decidir cómo debe procesarlo. El software en el controlador usa OpenFlow para instalar una nueva regla de clasificación y luego reenvía el paquete de vuelta al comutador para su procesamiento.

La creación de reglas dinámicas permite a SDN esperar a que surja tráfico y luego instalar una regla de clasificación para manejarlo. La consecuencia es un sistema de enrutamiento por flujos. Como ejemplo, considere el enrutamiento entre dos edificios. Suponga que hay dos rutas físicas (a menudo las empresas tienen varias rutas para usarlas como respaldo). El reenvío IP tradicional selecciona una ruta primaria para cada dirección IP de destino y sólo usa la ruta alternativa cuando ocurre una falla. Con SDN es posible realizar un balanceo de carga al encontrar una nueva conexión TCP. El primer paquete del nuevo flujo se reenvía hacia el controlador SDN, el controlador verifica la carga en las dos conexiones físicas y configura una regla de clasificación de modo que el nuevo flujo use la conexión con menos carga. En resumen:

La habilidad de agregar reglas de clasificación en forma dinámica significa que un sistema SDN puede realizar el balanceo de carga cada vez que aparezca un nuevo flujo TCP.

31.19 Un modelo de canalización para tablas de flujo

Nuestra descripción de SDN y OpenFlow sólo cubre las ideas básicas; la tecnología contiene muchos detalles y capacidades adicionales. Además del mecanismo de clasificación basado en hardware antes descrito, las versiones anteriores de OpenFlow incluyen funciones que pueden usarse con un mecanismo de clasificación basado en software. Desde luego, usar software para implementar el plano de datos de un dispositivo de red hace que el procesamiento de paquetes en general sea mucho más lento. Sin embargo, el software puede agregar una funcionalidad considerable.

Un ejemplo específico de funcionalidad incrementada surge de un modelo de *canalización*. En vez de un solo mecanismo de clasificación, los diseñadores imaginan una serie de mecanismos de clasificación llamados *tablas de flujo* dispuestos en una serie, como se ilustra en la figura 31.7.



Figura 31.7 Modelo de canalización de OpenFlow para el plano de datos. En teoría, los paquetes pasan a través de una canalización de tablas de flujo.

Cada tabla de flujo en la canalización especifica un conjunto de reglas de clasificación y acciones para cada regla. Las acciones incluyen la modificación de paquetes así como el reenvío de los mismos. Por ejemplo, una tabla de flujo puede especificar el encapsulamiento de un datagrama IP en MPLS o en otro datagrama IP (exterior). O una regla puede especificar la extracción de un paquete interno de un encapsulamiento anterior. Una vez que una tabla de flujo procesa un paquete, éste puede reenviarse o pasarse a la siguiente tabla de flujo de la canalización. Al pasar un paquete a través de varias etapas de tablas de flujo se permite un procesamiento de paquetes más complejo, incluyendo la inspección, el encapsulamiento y el cifrado. Lo que es más importante, OpenFlow dispone que se pasen *metadatos* junto con cada paquete, lo que hace posible que una etapa anterior recopile datos de un paquete, busque información y la pase a las etapas sucesivas. De esta forma, la primera etapa puede elegir un siguiente salto para el paquete, la segunda etapa puede encriptar el paquete y la tercera etapa puede reenviar el paquete encriptado al siguiente salto sin desencriptar una copia para calcular la dirección del siguiente salto.

El modelo de canalización de OpenFlow se implementó en software usando Linux. Es decir, una PC que ejecuta Linux actúa como elemento de red. Un controlador usa OpenFlow para configurar una canalización de tablas de flujo en el elemento. Aunque opera en forma mucho más lenta que un dispositivo de hardware, el sistema basado en software demuestra la funcionalidad de OpenFlow y se usa para mostrar a los fabricantes las ventajas potenciales de una canalización.

31.20 Efecto potencial de SDN en los fabricantes de red

Ahora que estamos familiarizados con las ideas básicas detrás de SDN, podemos considerar el aspecto económico. En un sentido amplio, SDN elimina la funcionalidad del plano de control de los elementos de red y deja sólo la tecnología del plano de datos. Para los fabricantes, el cambio es importante ya que les permite usar la funcionalidad del plano de control para diferenciar su equipo del de los competidores. Es decir, todos los elementos de red deben reenviar paquetes y todos los elementos de red deben adherirse a los estándares para los formatos de los paquetes. Sin embargo, el software del plano de control no se encuentra estandarizado y cada fabricante puede producir su propio conjunto de comandos. Un discurso de ventas típico promete que una red homogénea (una que se construya usando equipo de un solo fabricante) será más eficiente, más confiable y más fácil tanto de administrar como de expandir.

Si una intranet específica adopta SDN, las motivaciones cambian. El software del plano de control se ejecuta en controladores externos y no en los elementos de red. Por lo tanto, no se justifica comprar todo el equipo de un solo fabricante. Lo que es más importante, el hardware del plano de datos se convertirá en un producto básico y los administradores de red basarán su elección de los elementos de red adicionales en el precio, en vez del software del plano de control. Los fabricantes de red acostumbrados a una demanda continua y márgenes de ganancia elevados pueden encontrar que es difícil competir en un mercado de productos básicos.

Desde luego que los administradores de red necesitarán crear o comprar software de SDN que se ejecute en controladores externos. En un principio la mayoría de los administradores harán que su personal cree el software. Sin embargo, la disponibilidad del estándar OpenFlow significa que puede surgir una nueva industria para crear y vender software de administración de redes independiente del fabricante. Algunos defensores de SDN esperan que la comunidad de investigación produzca estándares abiertos. En cualquier caso, la adopción de SDN indicará que se dejan atrás el esquema actual de software de administración propietario y la integración vertical que siguen los fabricantes de equipo.

El punto es que SDN tiene el potencial de cambiar la economía de las redes de una forma importante.

Si se adopta ampliamente, SDN tiene el potencial de reducir los beneficios de los fabricantes de red actuales y crear una nueva industria en el software de administración.

31.21 Resumen

El término *redes definidas por software (SDN)* se usa para caracterizar una metodología para la administración de redes en donde la funcionalidad del plano de control se elimina de los elementos de red y se coloca en controladores independientes. Además de configurar y controlar un conjunto de elementos de red, el software de administración que se ejecuta en un controlador se comunica con el software de otros controladores para ofrecer una coordinación a nivel de red.

Se definió una parte del paradigma SDN: el protocolo OpenFlow utilizado para comunicarse entre un controlador y un elemento de red. OpenFlow define una tabla de flujos como un conjunto de patrones y acciones para cada patrón. Las definiciones de OpenFlow se asocian directamente con el hardware del motor de clasificación que se encuentra en los comutadores típicos. Se propuso un paradigma más avanzado que usa una canalización de tablas de flujo. Hasta ahora, el paradigma de la canalización sólo se ha implementado en software.

Para manejar los paquetes a velocidad alta, el plano de datos de un comutador usa un mecanismo de hardware de propósito especial. Conocido como TCAM, el hardware emplea paralelismo para comparar un paquete con una cantidad arbitraria de patrones en un solo paso. Aunque es costoso y produce cantidades considerables de calor, TCAM realiza comparaciones igual de rápido para un gran número de patrones que para un solo patrón.

Los patrones especifican valores numéricos para algunos campos de encabezados y valores “no importa” para otros. Puesto que los patrones pueden abarcar varias capas de la pila de protocolos, un comutador controlado por SDN permite nuevos paradigmas de reenvío, como rutas de capa 2, que no son posibles con las pilas de protocolos tradicionales.

SDN tiene el potencial de introducir cambios considerables en la industria de las redes. Una vez que se eliminan las instalaciones de administración de redes, el hardware de comutador se convertirá en un producto básico y los administradores de red no estarán tan motivados para comprar todo el equipo de un solo fabricante.

EJERCICIOS

- 31.1** ¿Cómo pasan las SDN de la administración de elementos a la administración a nivel de red?
- 31.2** ¿Cuáles son las dos partes conceptuales de un elemento de red?
- 31.3** Mencione tres interfaces de administración de red que se encuentran en los elementos de red tradicionales.
- 31.4** Cuando se usa SDN, ¿también se necesita SNMP? Explique.
- 31.5** Mencione cinco factores que usa un administrador de red al decidir cuántos controladores SDN implementar y en dónde colocarlos.
- 31.6** ¿Cuáles son las tres propiedades principales que especifica el estándar OpenFlow?
- 31.7** Suponga que un clasificador desea verificar que una trama de Ethernet, que se almacena en formato 802.1Q, contiene un datagrama IPv4 que transporta un segmento TCP destinado para un servidor web. ¿En qué desplazamientos están los elementos de encabezado que debe verificar el clasificador?
- 31.8** ¿Cuál es la ventaja principal de usar TCAM en un motor de clasificación?

- 31.9** Busque Internet para encontrar una lista de fabricantes que vendan commutadores habilitados para OpenFlow.
- 31.10** Mencione un ejemplo de reenvío que sea posible con SDN y no pueda especificarse con el software IP convencional.
- 31.11** Explique cómo es que un controlador SDN configura una regla de clasificación para un nuevo flujo de TCP en forma dinámica.
- 31.12** Investigue para ver si su organización usa SDN en su intranet.
- 31.13** Lea sobre el consorcio de SDN que están creando los fabricantes de equipo de red. ¿Por qué están involucrados los fabricantes y qué esperan lograr?
- 31.14** ¿Cuánto costarán los controladores SDN a una organización? Para obtener una aproximación, haga una estimación del número de controladores que necesitará la intranet de su organización y busque el precio de las PC básicas al comprarlas en suficiente cantidad como para cumplir con su estimación.

Contenido del capítulo

- 32.1 Introducción, 567
- 32.2 Sistemas integrados, 567
- 32.3 Elección de una tecnología de red, 569
- 32.4 Recolección de energía, 570
- 32.5 Comunicación inalámbrica de baja potencia, 570
- 32.6 Topología de malla, 571
- 32.7 La alianza ZigBee, 571
- 32.8 Radios 802.15.4 y redes de malla inalámbricas, 572
- 32.9 Conectividad de Internet y enrutamiento de malla, 573
- 32.10 IPv6 en una red de malla ZigBee, 574
- 32.11 El paradigma del reenvío de ZigBee, 575
- 32.12 Otros protocolos en la pila ZigBee, 576
- 32.13 Resumen, 577

32

La Internet de las cosas

32.1 Introducción

Durante treinta años, Internet se enfocó en proveer comunicaciones que involucren a los seres humanos. Las aplicaciones como el correo electrónico, el chat y la telefonía VoIP requieren de la interacción de dos participantes humanos. Las aplicaciones como la navegación web, la búsqueda y la transferencia de archivos disponen que un ser humano acceda a un servicio.

Este capítulo explora un nuevo uso de Internet: comunicaciones entre máquinas. Los investigadores y los profesionales de redes usan los términos *máquina a máquina (M2M)* e *Internet de las cosas[†] (IoT)* para describir el concepto. Este capítulo explica las causas para que exista un conjunto de máquinas en comunicación, así como una de las tecnologías inalámbricas en desarrollo.

32.2 Sistemas integrados

A diferencia de las primeras aplicaciones que usan computadoras convencionales, las aplicaciones IoT se enfocan en *sistemas integrados*. Es decir, las instalaciones de cálculo y comunicaciones integradas a un dispositivo, como un interruptor de luz, un aparato doméstico, un sistema de calefacción, un aire acondicionado o un sistema de seguridad. De acuerdo con Farnam Jahanian de la Fundación nacional de ciencia, de Estados Unidos: “Actualmente, el número de dispositivos interconectados en red equivale al número de personas de la Tierra. En 3 años los dispositivos de Internet superarán en número a las personas por un factor de tres”.

¿Por qué un aparato doméstico necesitaría comunicarse por Internet? Una razón implica la automatización en el hogar. Si todos los dispositivos eléctricos en un hogar tuvieran conectividad, el propietario

[†]Aunque fracasa en muchas formas al capturar la idea, el término *Internet de las cosas* parece haber ganado aceptación.

podría contactar con los dispositivos para determinar su estado y controlarlos. Sin regresar a su casa, un propietario podría responder una pregunta tal como: ¿dejé encendida la plancha? Y lo que es más importante, el propietario podría apagar la plancha o las luces, e incluso comenzar a precalentar el horno de camino a casa.

32.2.1 Sistemas integrados en la red inteligente

Los ejemplos anteriores todavía involucran al ser humano en el control de los dispositivos. ¿Acaso tiene también sentido la comunicación entre máquinas? Así es. Una instancia en la que será importante la comunicación entre máquinas surge del concepto de una *red inteligente*. En una red inteligente, un electrodoméstico tiene un controlador integrado y conectividad de red. El electrodoméstico puede enviar una consulta para encontrar el costo de la electricidad en diversos momentos y ajustar su programa de acciones según corresponda. Por ejemplo, un lavavajillas podría configurarse para retrasar el ciclo de lavado hasta después de las horas pico, cuando disminuye el costo de la electricidad. De manera similar, un sistema de aire acondicionado podría elevar un poco la temperatura durante horas pico y después enfriar la casa durante horas no pico.

La comunicación entre un electrodoméstico y la compañía eléctrica no necesita un ser humano en cualquiera de los extremos: el sistema integrado en el electrodoméstico actúa en forma autónoma para contactar a un servidor en la compañía de energía y obtener información sobre los horarios y precios. A su vez, los sistemas en la compañía de energía podrían configurarse para enviar las tarifas y los horarios a los sistemas residenciales cada vez que ocurriera un cambio.

32.2.2 Sistemas integrados de seguridad en línea

Los sistemas de automatización del hogar permiten que el propietario monitoree o controle los dispositivos eléctricos. Los sistemas de seguridad inteligentes van un paso más allá de la automatización del hogar al ser proactivos. Es decir, el sistema puede informar al propietario cuando ocurra un evento inesperado. Por ejemplo, si se activa un sensor de movimiento, el sistema puede contactar el teléfono inteligente del propietario, encender las luces y suministrar un flujo continuo de video de una cámara. Después el sistema puede aceptar comandos para ignorar el evento, reiniciar el sensor o tomar alguna acción adicional.

La ventaja considerable de los sistemas de seguridad inteligentes recae en su habilidad de tener una lista de contingencias, en donde las acciones dependen del evento que ocurra, la hora del día y las acciones del propietario. Por ejemplo, podría configurar el sistema para que informe a dos o más teléfonos inteligentes por un evento dado. Además, si un propietario no responde dentro de un tiempo especificado, el sistema podría optar por informar a la policía local.

32.2.3 Sistemas integrados en las ventas minoristas

La Internet de las cosas incluye mucho más que electrodomésticos. Un uso interesante de los sistemas integrados involucra establecimientos de ventas minoristas. Por ejemplo, algunos centros comerciales tienen pantallas electrónicas con sistemas integrados sofisticados. Cuando un comprador se acerca a la pantalla, el sistema usa una cámara para capturar una imagen e invoca a un software que puede analizar esa imagen. El software identifica rostros humanos, analiza cada uno de ellos y estima los rasgos, como la edad y el sexo del individuo. Después la pantalla selecciona anuncios orientados al perfil demográfico del individuo.

Las pantallas electrónicas en los centros comerciales hacen mucho más que sólo mostraranuncios:envían datos en ambas direcciones. En términos de recopilar información, los sistemas rastrean a cada espectador, calculan cuánto tiempo continúa la persona viendo la pantalla e informan las estadísticas a un servidor en la nube. El servidor a su vez recibe la información de las pantallas así como de otras fuentes y ejecuta algoritmos para combinar la información. Una vez que se toman nuevas decisiones, el servidor de la nube descarga la información en las pantallas. Por ejemplo, un servidor de la nube podría tomar muestras del clima en cada sitio y decidir entre anunciar paraguas durante una tormenta o acondicionadores de aire durante una ola de calor.

Se está implementando una tecnología similar en las tiendas de abarrotes. Las cámaras montadas sobre repisas y refrigeradores recopilan las imágenes de los compradores. Estas imágenes se analizan para determinar el tiempo que pasa cada comprador parado en un punto específico y la ubicación aproximada de los artículos que el comprador considera. Después los datos se envían a un servidor en la nube que combina la información de muchas tiendas y recomienda la colocación de los productos para cada tienda individual.

Podemos resumir lo siguiente:

Se está creando un conjunto de nuevas y emocionantes aplicaciones de Internet mediante la integración de un procesador y protocolos de comunicación en un dispositivo más grande, como un electrodoméstico. Muchos de los nuevos sistemas integrados se comunican con las computadoras de manera automática y no requieren que un ser humano inicie o controle la comunicación.

32.3 Elección de una tecnología de red

¿Qué tecnología de red es óptima para los dispositivos IoT? Ninguna tecnología resuelve todos los problemas por sí sola. Algunas aplicaciones requieren grandes volúmenes de datos (por ejemplo, varios flujos continuos de video de alta definición). Otras aplicaciones transfieren sólo pequeñas cantidades de datos (por ejemplo, una solicitud para la tarifa de energía actual y la respuesta de una sola cifra). Basándose en la terminología de planos de control y planos de datos que se utiliza en los elementos de red, los profesionales clasifican las aplicaciones integradas como *orientadas a los datos* u *orientadas al control*. Para soportar una aplicación orientada a los datos, una red debe ofrecer una tasa de velocidad de transferencia elevada.

Otra opción en la tecnología de redes surge de la necesidad de movilidad. Una red alámbrica funciona bien para las instalaciones permanentes, como un sistema de calefacción o un horno de pared. Sin embargo, muchos electrodomésticos son portátiles por lo que una conexión de red alámbrica sería incómoda e inconveniente.

Una restricción final sobre muchos sistemas integrados surge debido a la energía que consume el procesamiento integrado. Para los dispositivos que funcionan con baterías, la cantidad de energía consumida determina la vida total de las baterías; para otros dispositivos, el consumo de energía se relaciona con el costo.

La energía de las baterías es importante en especial para las aplicaciones en las que se colocan sensores en lugares inaccesibles o se distribuyen a través de un área geográfica extensa. Por ejemplo, para medir la tensión en un puente u otro tipo de estructura, los ingenieros civiles colocan sensores operados

por baterías en diversos puntos de la estructura. Los sensores miden los cambios a medida que se coloca una carga sobre la estructura e informan de las mediciones a un servidor central. De manera similar, el propietario de un hogar puede colocar un detector de agua operado por batería cerca de cada tubo de agua debajo de su casa para que le informen si ocurre una fuga. En dichos casos, no sería factible tender líneas de alimentación eléctrica y conexiones de red hacia los sensores. En conclusión:

No hay una sola tecnología de red que sea ideal para la comunicación entre máquinas. Además del volumen de tráfico y la movilidad, el consumo de energía puede ser un factor importante.

32.4 Recolección de energía

La energía de las baterías tiene una desventaja: con el tiempo, los procesos químicos en una batería agotan el suministro de energía y hay que reemplazarla. Se desarrollaron tecnologías para baterías (como las de litio) que pueden dar a una batería una vida excepcionalmente larga, incluso de varios años, bajo una carga ligera. Sin embargo, el costo de reemplazar las baterías aún puede ser bastante alto, como sería el caso de los sensores en un puente grande. Por lo tanto, los investigadores y los ingenieros están explorando alternativas.

Una de las metodologías más interesantes implica la *recolección de energía* en la que un dispositivo integrado es operado mediante la extracción de energía del entorno circundante. Por ejemplo, considere una persona que gira la perilla de una puerta o que acciona un interruptor de luz. La cantidad de energía aplicada en dichas acciones es pequeña. No obstante, si un dispositivo integrado sólo consume microwatts, las acciones mecánicas simples generan suficiente energía como para operar el dispositivo durante un tiempo breve. Si se construye un sistema para recolectar la energía de acciones repetidas, un sistema integrado podría funcionar sin baterías.

32.5 Comunicación inalámbrica de baja potencia

Como dijimos antes, el bajo consumo de energía y la comunicación inalámbrica son importantes para los dispositivos en la Internet de las cosas. Se diseñaron muchas tecnologías inalámbricas (incluyendo Wi-Fi) para maximizar la velocidad de transferencia en vez de minimizar el consumo de energía. Por lo tanto, los investigadores e ingenieros han investigado diseños alternativos con un énfasis en el consumo extremadamente bajo de energía.

En la primera parte del texto vimos que hay concesiones fundamentales involucradas en la reducción del consumo de energía. Un radio que usa menos energía no puede transmitir a través de la misma distancia que un radio que usa más energía; los radios con energía extremadamente baja sólo podrían transmitir hasta unos cuantos metros, lo que no es suficiente para cubrir una residencia común. Lo que es más importante, al bajar la energía baja la relación señal-ruido, lo que significa que la comunicación será más susceptible a la interferencia (es decir, la transmisión sufrirá en una mayor pérdida de paquetes). Cabe señalar que una mayor pérdida de paquetes indica que habrá más retransmisiones, lo cual indica más consumo de energía. Por lo tanto, es necesario diseñar con cuidado los detalles de una tecnología inalámbrica de bajo consumo de energía o, de lo contrario, el sistema en general requerirá más energía de la esperada.

32.6 Topología de malla

Para entender las redes inalámbricas de bajo consumo de energía hay que ignorar las restricciones prácticas, como las obstrucciones y la interferencia electromagnética, para lo cual debemos considerar un modelo simplista en el que la transmisión de un radio de bajo consumo de energía tiene el rango d . Es decir, hay que asumir que los receptores que están a una distancia d o menor pueden recibir transmisiones pero los receptores más alejados que d no pueden. La figura 32.1 ilustra el concepto. En la figura, las transmisiones del nodo A llegan al nodo B pero no llegan al nodo C.

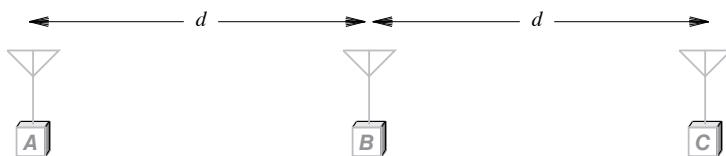


Figura 32.1 Tres nodos de radio de bajo consumo de energía separados por d unidades de distancia, donde d es el rango de las transmisiones de radio.

Surge la siguiente pregunta: dado el alcance limitado, ¿cómo puede distribuirse un conjunto de nodos que usan radios de bajo consumo de energía para cubrir todos los electrodomésticos en una residencia o un conjunto de sensores en un edificio de varios pisos? La respuesta está en el uso de una topología de *malla*. Es decir, cada nodo en la red actúa como un enrutador en miniatura al aceptar el reenvío de paquetes cuando sea necesario.

Las redes de malla se forman de manera automática. Al principio un nodo transmite una solicitud por difusión para encontrar vecinos. Los vecinos dentro del rango del radio reciben la solicitud y envían respuestas, lo cual permite al nodo compilar una lista de vecinos que pueden alcanzarse en forma directa. Una vez que se identifican los vecinos, los nodos participan en un protocolo de enrutamiento de malla que busca rutas a través de la malla y genera tablas de reenvío. Si no hay *huérfanos* o *islas* (es decir, conjuntos de nodos que no estén dentro del alcance de los demás), el algoritmo de enrutamiento de malla formará una red funcional. Por ejemplo, en la figura 32.1, la tabla de reenvío en el nodo B tendría entradas para A y C. Si llegara un paquete de A destinado para C, B usaría la información en su tabla de reenvío para enviar el paquete a C.

32.7 La alianza ZigBee

Un consorcio de distribuidores de equipo de red, conocido como la *alianza ZigBee*, diseñó un conjunto de especificaciones para crear redes de malla de bajo costo y con un uso eficiente de la energía. En especial, las especificaciones de ZigBee se enfocan en los dispositivos asociados con aplicaciones de redes inteligentes, como los aparatos que se usan en el hogar. Encontrará más información en el sitio web:

En vez de crear nuevos estándares de redes, la alianza ZigBee recomienda los estándares de redes existentes siempre que sea posible. Por ejemplo, ZigBee especifica el uso del estándar IPv6 del IETF en la capa 3 y el estándar 802.15.4 del IEEE para redes inalámbricas en la capa 2. El uso de estándares tales como IPv6 significa que las redes ZigBee pueden interoperar con las redes existentes.

32.8 Radios 802.15.4 y redes de malla inalámbricas

El IEEE estandarizó varias tecnologías inalámbricas de bajo consumo de energía, incluyendo *Bluetooth*. Se crearon distintas versiones de 802.15.4 que difieren en la banda de frecuencia que utilizan y en la técnica de modulación que emplean. Dichos aspectos de los estándares de radio son irrelevantes para nuestra explicación; sólo nos interesan las propiedades generales del 802.15.4.

Como es de esperarse, el estándar 802.15.4 transfiere paquetes. Sin embargo, la mayoría de las características son inesperadas. La figura 32.2 sintetiza las propiedades clave.

Propiedad	Valor
Paradigma de red	Conmutación de paquetes
Velocidad de datos máxima	250 Kbps
Tamaño de carga útil (MTU)	127 octetos
Distancia máxima	10 metros

Figura 32.2 Las propiedades clave de un enlace de red inalámbrica IEEE 802.15.4.

Como se muestra en la figura, el diseño del estándar 802.15.4 difiere de las tecnologías inalámbricas convencionales en varias formas. En vez de una red con una velocidad de transferencia mayor a 10 Mbps, el 802.15.4 tiene una velocidad de datos increíblemente baja. En vez de paquetes grandes, un paquete 802.15.4 tiene una carga útil bastante pequeña. En vez de abarcar suficiente distancia para cubrir una residencia completa, un radio 802.15.4 operado por baterías no puede alcanzar ni siquiera 40 pies. En la práctica, la distancia efectiva será incluso menor si la ruta contiene obstrucciones o si hay interferencia electromagnética.

Para entender por qué el estándar eligió dichos parámetros, hay que tener en cuenta dos cosas:

- El criterio de diseño clave es el bajo consumo de energía
- La cantidad de datos a transferir es pequeña

Las restricciones en cuanto al tamaño de los paquetes, la velocidad de los datos y la distancia se derivan del objetivo de bajo consumo de energía y del pequeño volumen de datos a transferir. El bajo consumo de energía es especialmente importante para los dispositivos operados por baterías: aunque tal vez no pueda transmitir muchos datos y tampoco pueda transmitir en forma continua, un nodo de radio

802.15.4 puede funcionar con una batería estándar durante un año o más. En términos de distancia, los diseñadores eligieron una metodología de malla en vez de usar un radio de alta potencia.

32.9 Conectividad de Internet y enrutamiento de malla

Para ser parte de la Internet de las cosas, una red de malla debe conectarse a Internet global. De esta forma, una red inalámbrica ordinaria contiene uno o más nodos que tienen una conexión permanente a Internet. Las especificaciones de ZigBee usan el término *enrutador de frontera* para referirse a un dispositivo con conexión a Internet. Puesto que la mayoría de los nodos en una red de sensores no tienen capacidad de cómputo o de almacenamiento, el sistema está configurado de modo que los nodos usen la nube para el almacenamiento y el cómputo. La dependencia de la nube significa que el enrutamiento de malla debe enfocarse en la comunicación de Internet. Es decir, en vez de calcular una ruta hacia cada uno de los otros nodos de la malla, un protocolo de enrutamiento de malla debe diseñarse para calcular una ruta a través de cada nodo y hasta un enrutador de frontera conectado a Internet.

Las rutas a través de la malla se generan un paso a la vez. Un enrutador de frontera difunde que tiene una conexión hacia Internet. Los nodos dentro del rango del enrutador de frontera reciben la difusión y envían sus tablas de reenvío según corresponda. Despues, cada nodo que tenga un enlace directo al enrutador de frontera difunde un mensaje que sus vecinos reciben. El vecino que no tenga una ruta actualiza su tabla de reenvío y la difunde al segundo conjunto de vecinos. De esta forma, cada nodo en la malla puede establecer una ruta hacia Internet.

¿Qué ocurre si cierto nodo recibe varias difusiones que anuncian rutas a Internet? A diferencia de un protocolo de enrutamiento convencional que busca la ruta más corta, el objetivo de los protocolos de enrutamiento de malla es seleccionar la *mejor* ruta, en donde la definición de *mejor* depende de la calidad del enlace y del número de saltos.

Lo interesante es que un nodo emisor no puede conocer la mejor ruta a través de la cual pueda transmitir. Para entender por qué, considere la figura 32.3 que muestra dos enrutadores de frontera y un nodo inalámbrico.

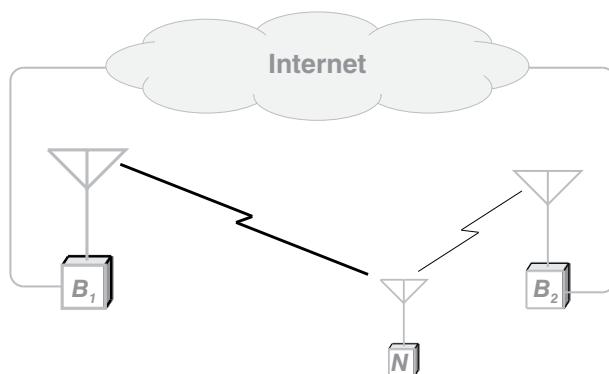


Figura 32.3 Una configuración asimétrica de dos enrutadores de frontera y un nodo inalámbrico N que está más cerca del enrutador B_2 pero que puede recibir transmisiones de ambos enrutadores.

En la figura, el enrutador de frontera B_1 está más alejado del nodo N que el enrutador de frontera B_2 , pero B_1 tiene un transmisor potente. Por lo tanto, si el nodo N mide las transmisiones entrantes, B_1 tendrá la señal más fuerte. Sin embargo, cuando N transmita, B_2 (que está más cerca) podría recibir una señal mucho más fuerte que B_1 . De hecho, tal vez B_1 no pueda recibir las transmisiones del nodo N . En conclusión:

Puesto que las transmisiones pueden ser asimétricas, medir la fuerza de la señal asociada con los paquetes entrantes no nos indica la fuerza de la señal en dirección opuesta.

Para manejar las transmisiones asimétricas, los protocolos ZigBee incluyen un protocolo de *establecimiento de enlace de malla (MLE)*. Un nodo envía mensajes MLE a los vecinos y cada vecino responde con un valor que indica la fuerza de la señal. De esta forma, un emisor sabe qué tan bien puede recibir cada vecino sus transmisiones. Al formar una ruta hacia Internet es importante seleccionar un vecino que pueda recibir bien las transmisiones salientes.

32.10 IPv6 en una red de malla ZigBee

Dijimos que los protocolos ZigBee se basan en IPv6. Sin embargo, IPv6 especifica que no puede usarse en una red a menos que la MTU de la red sea de 1280 octetos como mínimo. ¿Cómo puede usarse IPv6 en una red 802.15.4 que tenga una MTU de 127 octetos? El IETF asumió un compromiso: agregar una capa adicional de protocolos entre IPv6 y los protocolos MAC subyacentes. Conocida como *adaptación 6LoWPAN*, la capa adicional oculta al protocolo IP los detalles de 802.15.4. El nombre *6LoWPAN* se podría leer como *IPv6 sobre redes inalámbricas de área personal con bajo consumo de energía*. En esencia, la capa de adaptación, también conocida como *capa intermedia*, acepta un datagrama, lo divide en piezas y envía las piezas a través del enlace inalámbrico, una a la vez. A diferencia de la fragmentación, que es de extremo a extremo, el protocolo de la capa de adaptación en el siguiente salto recibe las piezas, las agrupa y pasa el resultado a IPv6. Todas las piezas deben llegar en orden o el datagrama se descarta.

Además de la capa de adaptación, se definió una versión especial del protocolo de descubrimiento de red IPv6 para usarlo con redes de malla ZigBee. Llamado *6LoWPAN-ND*, el protocolo se necesita debido a que IPv6 asume que será posible llegar a todos los nodos en una red dada con una sola transmisión, pero las redes de malla no tienen la propiedad de la conectividad universal. En especial, IPv6-ND realiza la detección de direcciones duplicadas preguntando si algún otro nodo tiene cierta dirección. 6LoWPAN-ND usa un paradigma de registro y cada nodo en la malla registra su dirección con el enrutador de frontera. Si un nuevo nodo trata de registrar una dirección duplicada, el enrutador de frontera envía una respuesta 6LoWPAN-ND que rechaza la solicitud. 6LoWPAN-ND también incluye un estado suspendido, el cual permite que un nodo conserve su energía hasta que sea tiempo de renovar un registro de dirección.

32.11 El paradigma de reenvío de ZigBee

La alianza ZigBee optó por usar IPv6 para todo el reenvío de paquetes. Es decir, cada nodo en una malla 802.15.4 actúa como un enrutador IPv6. Por desgracia, IPv6 y los protocolos de enrutamiento convencionales no se diseñaron para un entorno así, lo que significa que se requiere soporte adicional. El IETF adoptó un enfoque interesante para resolver el problema: separar las dos direcciones de reenvío (es decir, de un nodo de la malla hacia Internet y de Internet hacia un nodo de la malla). Es fácil reenviar hacia Internet. Cuando se forma la malla, cada nodo busca la ruta hacia un enrutador de frontera e instala una ruta predeterminada que señala el siguiente salto a través de la ruta.

Para reenviar de Internet hacia un nodo de la malla se requiere el enrutador de frontera. Los nodos de la malla ejecutan un protocolo de enrutamiento conocido como *protocolo de enrutamiento para redes con pérdida y bajo consumo de energía (RPL)*. El diseño del RPL se basa en dos principios:

- La topología de malla es semipermanente, ya que los cambios son poco frecuentes.
- Debido a las limitaciones de memoria, los nodos no pueden conservar una tabla de enrutamiento completa.

Semipermanencia. A diferencia del enrutamiento en un sistema celular, RPL no está diseñado para manejar un movimiento rápido. La idea general es que, aunque un nodo puede moverse, la mayoría de los nodos tienden a permanecer estables durante un periodo relativamente extenso. Los sistemas de iluminación integrados y los electrodomésticos permanentes, como hornos, lavavajillas y calentadores, no se mueven. Los electrodomésticos portátiles como las planchas pueden moverse, pero tienden a permanecer en el mismo punto durante horas o días. Por ende, RPL no invierte energía en enviar actualizaciones con frecuencia o en converger rutas con rapidez.

Limitaciones de memoria. En vez de calcular tablas de reenvío individuales en los nodos, RPL hace que los nodos informen al enrutador de frontera sobre la topología de malla (por ejemplo, sobre las conexiones directas entre los nodos). El enrutador de frontera, que supuestamente tiene más poder de cómputo y almacenamiento que un solo nodo, usa la topología para calcular un *árbol de reenvío* para toda la malla. Cuando llega un datagrama IPv6 de Internet destinado para un nodo de la malla, el enrutador de frontera usa el árbol de reenvío para buscar una ruta a través de la malla hacia el destino. Después, el enrutador de frontera encapsula el datagrama original dentro de otro datagrama IPv6 y coloca un encabezado de ruta de origen en el datagrama exterior que especifica la ruta a través de la malla. A medida que el datagrama encapsulado viaja por la malla, cada nodo a lo largo de la ruta extrae una dirección del encabezado de ruta de origen, usa la dirección como el siguiente salto y reenvía el datagrama. Una dirección en el encabezado de ruta de origen siempre será la dirección de un vecino que se pueda contactar de manera directa, lo que significa que los nodos de la malla sólo necesitan enviar a los vecinos que estén conectados directamente.

La figura 32.4 ilustra el gráfico que calcula un enrutador de frontera para una malla. Cada nodo en el gráfico representa un nodo en la malla y cada enlace en el gráfico proporciona el *padre* de un nodo (es decir, un vecino en la ruta hacia el enrutador de frontera). RPL define el término *gráfico acíclico dirigido y orientado al destino (DODAG)* para referirse al gráfico.

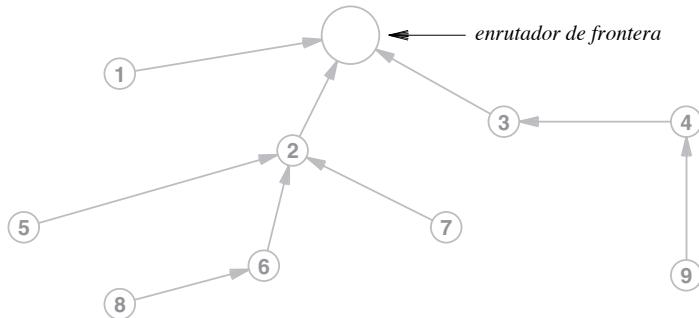


Figura 32.4 Un DODAG de ejemplo para una malla de nueve nodos y un enrutador de frontera.

Aunque la figura muestra nodos espaciados en un área bidimensional para indicar las posibles posiciones, el enrutador de frontera no aprende las ubicaciones físicas. En la práctica, un DODAG simplemente permite al enrutador de frontera calcular rutas. Por ejemplo, al enviar un datagrama al nodo 9 el enrutador de frontera especifica la ruta 3-4-9 y al enviar al nodo 5, el enrutador de frontera especifica la ruta 2-5.

32.12 Otros protocolos en la pila ZigBee

La especificación ZigBee incluye protocolos de seguridad adicionales (por ejemplo, *PANA* y *TLS*), protocolos para acceder a DNS (*mDNS* y *DNS-SD*) y protocolos de capa de aplicación. La figura 32.5 muestra la disposición de los principales protocolos en la pila ZigBee.

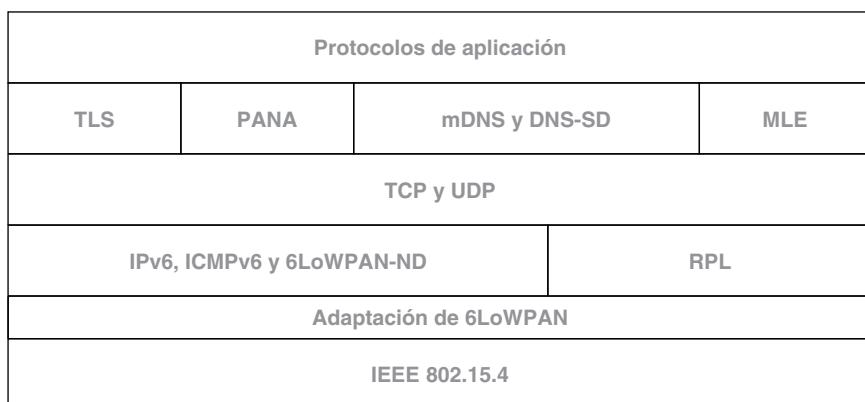


Figura 32.5 La disposición de los principales protocolos en la pila IP de ZigBee.

32.13 Resumen

El término *Internet de las cosas* se usa para describir sistemas integrados conectados que usan la comunicación entre máquinas. Las aplicaciones incluyen la automatización del hogar, las redes inteligentes, la seguridad y los sistemas de ventas minoristas.

Aunque ninguna tecnología de redes individual es la mejor para todas las aplicaciones integradas, muchas necesitan de una comunicación inalámbrica de bajo consumo de energía. Para minimizar el uso de la energía, una red inalámbrica puede usar una metodología de malla; en vez de usar poderosos transmisores de radio que puedan abarcar distancias extensas, los nodos de bajo consumo de energía aceptan reenviar paquetes unos a otros.

La alianza ZigBee ideó especificaciones para una red de malla inalámbrica de bajo consumo de energía que envía datagramas IPv6 a través de enlaces de radio IEEE 802.15.4. Se necesitan protocolos adicionales como 6LoWPAN para poder usar IPv6 a través de una red con una MTU extremadamente pequeña. La pila ZigBee incluye un protocolo de enrutamiento especial para calcular un gráfico de red que permita al enrutador de frontera enviar datagramas por la ruta de origen a través de la malla.

EJERCICIOS

- 32.1** ¿Qué es un sistema integrado?
- 32.2** Mencione tres ejemplos de sistemas integrados conectados a Internet.
- 32.3** ¿Qué tecnología de red es la mejor para la Internet de las cosas? Explique.
- 32.4** Hay un fabricante que vende bombillas de luz con un radio 802.15.4 y un controlador integrado en cada bombilla. Busque en web y haga una lista de otros artículos que pueden usarse en la Internet de las cosas.
- 32.5** ¿Qué es la recolección de energía?
- 32.6** ¿Qué es una red de malla y cómo funciona?
- 32.7** Mencione las características del estándar inalámbrico 802.15.4 del IEEE.
- 32.8** Si un nodo inalámbrico recibe una señal más fuerte del enrutador 1 que del enrutador 2, ¿debería el nodo usar el enrutador 1 como ruta hacia Internet? Explique.
- 32.9** ¿Cuál es la idea básica detrás del protocolo MLE?
- 32.10** ¿Necesita cada nodo en una red de malla ZigBee una tabla de reenvío IP completa? ¿Por qué sí o por qué no?
- 32.11** Cuando un enrutador de frontera usa RPL, ¿qué información recolecta?
- 32.12** Suponga que un nodo de malla de ZigBee envía un datagrama IPv6 a otro nodo de malla. ¿Qué ruta toma el datagrama? (Sugerencia: considere la información que mantiene un nodo individual).

Contenido del capítulo

- 33.1 Introducción, 579
- 33.2 La necesidad de servicios de Internet escalables, 579
- 33.3 Almacenamiento de contenido en caché (Akamai), 580
- 33.4 Balanceadores de carga web, 580
- 33.5 Virtualización de servidores, 581
- 33.6 Comunicación de igual a igual, 581
- 33.7 Centros de datos distribuidos y replicación, 582
- 33.8 Representación universal (XML), 582
- 33.9 Redes sociales, 583
- 33.10 Movilidad y redes inalámbricas, 583
- 33.11 Video digital, 583
- 33.12 Acceso y commutación de mayor velocidad, 584
- 33.13 Computación en la nube, 584
- 33.14 Redes superpuestas, 584
- 33.15 Middleware, 586
- 33.16 Implementación generalizada de IPv6, 586
- 33.17 Resumen, 587

33

Tendencias en tecnologías de redes y usos

33.1 Introducción

Uno de los aspectos más intrigantes de las redes de computadoras surge debido a la introducción continua de nuevas aplicaciones y tecnologías de redes. Las aplicaciones responsables de la mayoría de los paquetes en Internet se inventaron en la década pasada. Muchas aplicaciones que ahora se usan ampliamente no eran viables cuando se inventó Internet debido a que dependen de una comunicación más veloz y confiable.

Este capítulo resume algunas de las tendencias en las tecnologías de redes, sus aplicaciones y servicios. El capítulo considera los desarrollos recientes, así como una investigación a plazo más largo.

33.2 La necesidad de servicios de Internet escalables

En un sentido estricto, el modelo de comunicación cliente-servidor significa que una aplicación (un servidor) inicia primero y espera el contacto de otra aplicación (un cliente). En un sentido más amplio, la industria de las redes usa el término *cliente-servidor* para caracterizar una arquitectura en la que muchos clientes potenciales se conectan a un solo servidor centralizado. Por ejemplo, una corporación que opera un servidor web puede esperar el contacto de diversos usuarios. La desventaja de un servidor centralizado surge del rendimiento resultante, ya que a medida que aumenta el número de clientes, el servidor (o la red de acceso que conduce al servidor) se convierte de inmediato en un cuello de botella, en especial si cada cliente descarga muchos bytes de contenido.

El problema del cuello de botella de un servidor se considera una de las limitaciones más importantes en los servicios de Internet. En consecuencia, tanto la comunidad de investigación de redes como la industria de las redes han investigado formas de proporcionar arquitecturas y tecnologías que permitan que los servicios de Internet escalen e incorporen nuevas tendencias. Se está usando una variedad de metodologías. Las siguientes secciones describen varias de ellas. Para resumir:

Se ha ideado una variedad de tecnologías para que los servicios de Internet puedan escalar; aunque las metodologías difieren de manera considerable, cada una es útil en ciertos casos.

33.3 Almacenamiento de contenido en caché (Akamai)

Una de las primeras tecnologías de escalamiento se enfocaba en almacenar el contenido web en caché. Por ejemplo, a menudo los ISP tienen una caché que conserva una copia de cada página web estática (es decir, una página en la que el contenido no cambia con rapidez). Si un número N de los usuarios de un ISP obtienen la misma página, sólo hay que enviar una solicitud al *servidor de origen*; por lo que es posible satisfacer $N - 1$ solicitudes de la caché.

Las empresas como Akamai extendieron la idea del uso de caché mediante el ofrecimiento de un servicio de caché distribuido. Akamai dispone de un conjunto de servidores ubicados a lo largo de Internet y una organización puede contratar a Akamai para cargar previamente sus cachés con contenido. Para asegurar que las cachés estén actualizadas, el cliente de una organización puede actualizar las cachés de Akamai en forma periódica. Los visitantes del sitio web de la organización obtienen gran parte del contenido de una caché de Akamai cercana en vez del servidor central de la organización. Como resultado se reduce la carga en el servidor central.

33.4 Balanceadores de carga web

Como el uso es elevado y muchos negocios minoristas dependen de web para las ventas directas a los clientes, la optimización del servidor web ha recibido mucha atención. Uno de los mecanismos interesantes que se utilizan para construir un sitio web grande se conoce como *balanceador de carga*. Un balanceador de carga permite que una empresa reemplace un solo servidor por un centro de datos que contenga varias computadoras, cada una de las cuales ejecuta una copia idéntica de un servidor web. El balanceador de carga distribuye las solicitudes entrantes entre los servidores físicos. La figura 33.1 ilustra la arquitectura.

Un balanceador de carga examina cada solicitud HTTP entrante y envía la solicitud a uno de los servidores. El balanceador de carga recuerda las solicitudes recientes y dirige todas las solicitudes de un origen dado hacia el mismo servidor físico. Para asegurarse de devolver la misma respuesta a una solicitud, todos los servidores usan un sistema de base de datos compartido. Por lo tanto, si un cliente coloca un pedido, todas las copias del servidor web podrán acceder al pedido.

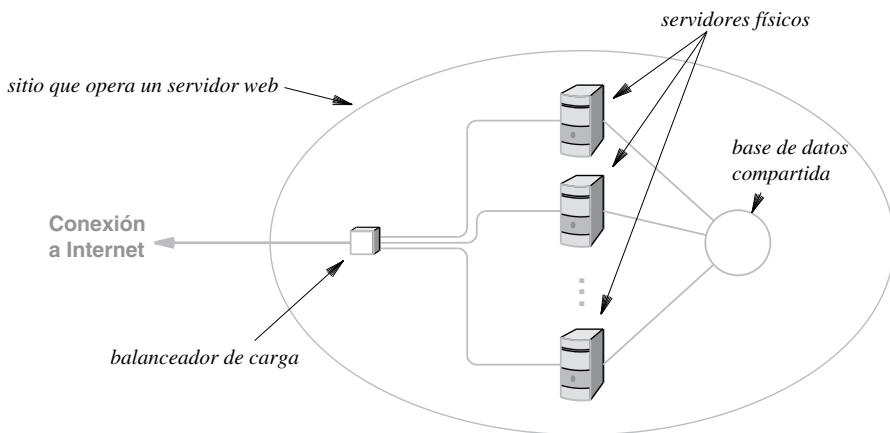


Figura 33.1 Ilustración de un balanceador de carga utilizado para sitios web de gran escala.

33.5 Virtualización de servidores

De la *virtualización de servidores* surge otro enfoque en la escalabilidad. La motivación comienza con una observación: muchos sitios operan con varios servidores (por ejemplo, un servidor de correo electrónico, un servidor web y un servidor de bases de datos). En una arquitectura convencional, cada servidor debe colocarse en una computadora física independiente. Puede ocurrir un problema de rendimiento debido a que los servidores que se ejecutan en la computadora A están todos ocupados, pero los servidores que se ejecutan en la computadora B están inactivos.

Los sistemas de virtualización resuelven el problema al permitir que un sitio tenga un centro de datos con varias computadoras físicas y software que permite a un administrador mover un servidor dado de una computadora a otra en cualquier momento. La tecnología involucrada es compleja y debe hacerse cargo de muchos detalles técnicos, como los cambios en el reenvío de paquetes. Sin embargo, la idea es simple y directa: ejecutar el servidor en un sistema de *máquinas virtuales* (VM) que soporte la migración virtual. Si cierta computadora física se sobresatura, un administrador migra una o más VM a otras computadoras físicas.

33.6 Comunicación de igual a igual

En la década de 1990, varios grupos experimentaron con una técnica general para incrementar la velocidad de descarga de archivos. En vez de obtener un archivo completo de un servidor central o de una caché preestablecida, un cliente obtiene varias piezas de un archivo de varios servidores colocados en servidores a través de Internet. Cada vez que un cliente necesita una pieza de un archivo, el cliente selecciona un servidor cercano que tenga una copia. Para incrementar el número de ubicaciones donde pueda encontrarse una pieza, cada cliente que obtiene una pieza del archivo acepta actuar como servidor y suministrar la pieza a otros clientes. Esta metodología se conoce como *arquitectura de igual a igual* (*p2p*).

Algunos de los sistemas p2p más populares se crearon para poder compartir archivos de música. Por ejemplo, Napster y Kazaa seguían la metodología p2p y cada uno era popular entre los adolescentes. Desde luego que a un usuario ordinario no le importa la tecnología que está detrás; lo único que le interesa es que el sistema les permita obtener copias de archivos de música. Muchos usuarios no están conscientes de que al usar un sistema p2p su computadora acepta propagar los archivos a otros.

33.7 Centros de datos distribuidos y replicación

Aunque las cachés de contenido, los balanceadores de carga, la virtualización y las arquitecturas p2p pueden incrementar la escalabilidad de los servidores, algunos sitios tienen tanto tráfico que se necesita otra solución: la replicación de un sitio completo. Usamos el término *centros de datos distribuidos* para caracterizar la metodología.

Como ejemplo considere el motor de búsqueda de Google. Google recibe miles de millones de contactos a diario. Para manejar la carga, Google creó varios centros de datos espaciados en diversas ubicaciones geográficas. Cuando un usuario introduce el nombre de dominio www.google.com, es dirigido al centro de datos de Google más cercano. Esta metodología puede considerarse una forma de balanceo de carga entre sitios. Desde luego que para ofrecer un servicio consistente, Google debe asegurarse de que un centro de datos devuelva exactamente los mismos resultados de búsqueda que sus otros centros de datos.

33.8 Representación universal (XML)

Otra tendencia en las redes surge debido a la amplia aceptación del *lenguaje de marcación extensible (XML)*. En un principio, XML se diseñó para dar una estructura a los documentos web de modo que varias aplicaciones pudieran entender estos documentos. En vez de fijar etiquetas, XML permite a un programador elegir diversas etiquetas y permite otorgar a cada campo un nombre intuitivo. Por ejemplo, podemos asumir que un documento que contiene las etiquetas <nombre>, <calle>, <ciudad>, <pais> y <codigo_postal> incluye un registro de la dirección de una persona. Una de las ideas clave detrás de XML es su habilidad de codificar documentos que se describen a sí mismos. Es decir, el documento incluye una *hoja de estilos* que especifica una estructura de documento legítima.

XML se convirtió en un estándar para la representación y se utiliza en una variedad de nuevas formas que no se habían previsto en los primeros diseños. Por ejemplo, XML se usa en la interfaz entre un servidor web y una base de datos, y se han creado algunos balanceadores de carga que pueden analizar XML. Además, XML se usa para controlar las descargas en los dispositivos móviles y para representar las especificaciones utilizadas por los sistemas de administración de redes.

33.9 Redes sociales

A principios de la década de 2000, el uso de Internet cambió de un modelo de consumidor a uno de interacción social entre iguales. En un principio, la mayor parte de la información en Internet se suministró a través de *productores*, que eran organizaciones como compañías de medios. Un usuario individual consumía la información, pero no la producía. Para la década de 2000 surgieron sitios como Myspace, Facebook y YouTube que permitían a cualquier usuario crear contenido, lo que significa que un usuario común envía más datos.

El cambio en la interacción es más notable entre usuarios jóvenes. Muchos adolescentes crearon un blog o se suscribieron a uno de los sitios antes mencionados. En Estados Unidos, un porcentaje considerable de parejas recién casadas se conocieron a través de un servicio en línea. Además aumentó el uso del chat en línea y otras formas de comunicación de persona a persona.

33.10 Movilidad y redes inalámbricas

La comunicación móvil está entre las tendencias más importantes y los usuarios esperan estar conectados a Internet de manera continua. La mayoría de los hoteles ofrecen conexiones a Internet para sus huéspedes y ahora las aerolíneas ofrecen el servicio de Internet en muchos aviones. El autor tomó un crucero y le complació descubrir que la conexión de Internet a bordo del barco funcionó tan bien que podía usarse para llamadas telefónicas VoIP.

La demanda de la comunicación móvil disparó el interés en las tecnologías inalámbricas y se crearon muchos estándares inalámbricos. Hay una serie de estándares que continúa ampliando la red Wi-Fi por todas partes. Sin embargo, el cambio más importante ocurrió en la industria de la telefonía celular, con los teléfonos celulares usando IP. En especial, ahora que los proveedores de telefonía celular cambiaron a LTE, todo el sistema está cambiando a IP, lo que significa que hay convergencia entre el servicio celular e Internet.

33.11 Video digital

Muchos proveedores de cable reemplazaron las instalaciones de transmisión analógicas por digitales, y ahora transmiten contenido en forma digital a través de redes de paquetes. De hecho, muchos proveedores usan IP como el protocolo de paquetes, de modo que es más fácil para los proveedores de cable ofrecer el servicio ISP a los clientes.

El uso de IP para video crea oportunidades interesantes. Primero, la televisión e Internet convergen, por lo que es fácil ver programas de televisión en una computadora o usar una televisión digital como pantalla de computadora. Además, IP facilita el despliegue de video *bajo demanda*, donde un usuario puede acceder al contenido cuando lo deseé, controlar la reproducción con funciones de pausa y rebobinado, y capturar el contenido en vivo para verlo después.

33.12 Acceso y conmutación de mayor velocidad

En el extremo de Internet, las tecnologías de acceso como los módems DSL y de cable se han vuelto un estándar. Cuando las tecnologías aparecieron por primera vez, las velocidades de datos de 2 a 6 Mbps (dos veces más veloces que una conexión de marcación telefónica) parecían sorprendentes. Sin embargo, ahora los primeros módems DSL y de cable están siendo reemplazados por otras tecnologías. Los proveedores celulares están comenzando a ofrecer tecnologías que pueden generar hasta 50 Mbps para dispositivos móviles. En algunas partes de Estados Unidos, los ISP ofrecen conexiones ópticas a los clientes residenciales que operan a velocidades de gigabits, tres veces más que los módems DSL y de cable.

Los conmutadores de Ethernet que se usan en los centros de datos empresariales también se están volviendo más rápidos. La Ethernet Gigabit, que alguna vez se utilizó como tecnología de red troncal en los campus universitarios, ahora se considera la velocidad de conexión de escritorio estándar. Las redes troncales usan 10 Gbps y es probable que las velocidades aumenten a 40 Gbps o más. Las velocidades de datos más altas son suficientes para soportar la transmisión en flujo continuo de video de alta definición y otras nuevas aplicaciones.

33.13 Computación en la nube

Las empresas grandes dependen de las redes de computadoras para todos los aspectos de su negocio. Sin embargo, la disponibilidad de un acceso a Internet confiable de alta velocidad hace posible que las empresas cambien su modelo de negocios de una manera considerable, ya que en vez de contratar una gran cantidad de personal interno para dar mantenimiento a los sistemas de hardware y software, las empresas subcontratan sus operaciones de TI con *proveedores de nube* como Amazon. El proveedor de nube mantiene un conjunto de centros de datos que incluyen servicios computacionales y de almacenamiento, incluyendo la actualización del software (por ejemplo, asegurar que se hayan instalado las actualizaciones en todas las computadoras) y el respaldo (garantizar que las copias de los archivos de datos se guarden en forma segura).

En términos de reducción del costo, una ventaja principal de un servicio de nube se debe a su flexibilidad. Si una empresa mantiene sus propias instalaciones de TI, éstas deben tener la capacidad suficiente para las necesidades pico. Por desgracia, el uso varía con el tiempo. Por ejemplo, un despacho contable puede necesitar muchos recursos al final del año fiscal cuando se procesan y presentan las declaraciones de impuestos, pero menos recursos en otros momentos. Con un servicio de nube, un cliente sólo paga por los recursos de cómputo y de almacenamiento cuando los necesita. En vez de computadoras complejas, los empleados sólo necesitan dispositivos básicos (como tabletas) para acceder a los servicios de nube.

33.14 Redes superpuestas

En los últimos años surgió una tecnología general que se conoce como *redes superpuestas*, la cual puede usarse para proveer un acceso restringido, seguridad mejorada y comunicación no estándar. La idea es simple: conectar un conjunto de computadoras a Internet, pero en vez del reenvío IP convencional, definir un conjunto de *túneles* entre las computadoras. Es decir, restringir todo el reenvío de paquetes a

los túneles. La idea es similar a MPLS, sólo que éste requiere que los enrutadores a lo largo de la ruta entiendan el encapsulamiento de paquetes MPLS. Puesto que usa IP, la tecnología de redes superpuestas no requiere que los enrutadores entiendan la funcionalidad de la superposición.

Desde el punto de vista de las aplicaciones que se ejecutan en las computadoras, los túneles definen su conectividad. El mecanismo de superposición da la ilusión de conexiones dedicadas, aun cuando el paquete viaja a través de Internet. Por ejemplo, la figura 33.2 ilustra un conjunto de computadoras conectadas a Internet y una red superpuesta que se impone en las computadoras.

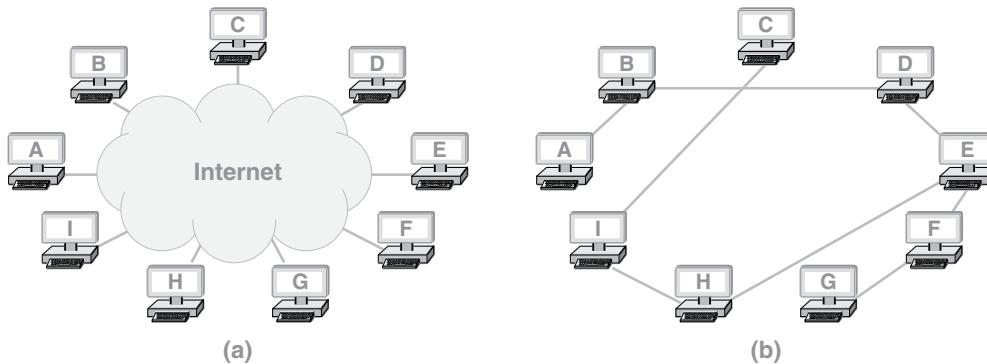


Figura 33.2 Ilustración de (a) un conjunto de computadoras conectadas a Internet, y (b) una posible red superpuesta que se impone en las computadoras.

¿Por qué se interesaría una empresa en usar la tecnología de redes superpuestas? Las superposiciones ofrecen dos beneficios pertinentes para una intranet de producción:

- Confidencialidad
- Control de acceso

Confidencialidad. Cuando transmite un paquete de una computadora a otra, la tecnología de superposición encapsula el paquete en un datagrama exterior. El datagrama interior no es inspeccionado por los saltos intermedios a lo largo de la ruta. Por lo tanto, para mantener todas las transferencias confidenciales, la computadora emisora cifra el datagrama interior antes de transmitirlo y la computadora receptora lo descifra. Durante el viaje entre dos computadoras, el datagrama puede recorrer varias redes o incluso varios ISP, pero ninguno de los saltos intermedios podrá acceder al contenido del datagrama. De hecho, ya hemos visto un ejemplo específico de la tecnología de superposición utilizada para ofrecer confidencialidad: una VPN.

Control de acceso. La tecnología de superposición permite a un administrador dividir una intranet física en varias intranets lógicas. El administrador puede optar por mantener las intranets totalmente aisladas (para evitar que la información se fugue accidentalmente de una a la otra), o puede optar por conectarlas en puntos específicos. La opción de interconexión significa que un administrador puede instalar *salvaguardas* en los puntos de interconexión (como software DPI que examine el contenido de los paquetes). Por ejemplo, en la figura 33.2(b) toda la comunicación entre la computadora C y H debe pasar a través de la computadora I.

Aunque nuestros ejemplos hablan sobre colocar software superpuesto en computadoras individuales, las superposiciones también pueden usarse con enrutadores o commutadores. Es decir, un commutador o un enrutador pueden definir túneles y luego usarlos para transferir datagramas a otros enrutadores en la red superpuesta.

33.15 Middleware

El término *middleware* se refiere al software que se utiliza para permitir que las aplicaciones que se ejecutan en varias computadoras se coordinen y trabajen en conjunto. Por lo general, el middleware se encuentra entre una aplicación y el sistema operativo. El middleware puede usarse también para aplicaciones de administración de redes, así como para aplicaciones que los usuarios invocan. El middleware para la administración de redes se ha vuelto especialmente interesante en años recientes y parece que aparecerán nuevos sistemas de middleware comerciales.

Como ejemplo de lo que puede hacer el middleware, considere el middleware *Shibboleth* desarrollado por Internet2. Shibboleth ofrece servicios de administración y validación de contraseñas a través de un conjunto de organizaciones. En el caso de Internet2, las organizaciones son miembros tales como universidades. Desde el punto de vista de un usuario, Shibboleth permite usar un solo inicio de sesión y contraseña en varias instituciones (tanto para computadoras como para redes). Una organización que implementa Shibboleth automáticamente acepta asignar el inicio de sesión de Shibboleth a un inicio de sesión local. Desde el punto de vista del administrador de red, Shibboleth permite la administración local de computadoras y redes, y sólo requiere una asignación. Por lo tanto, decimos que Shibboleth ofrece “un solo inicio de sesión” para una *federación difusa* de organizaciones.

33.16 Implementación generalizada de IPv6

Ninguna lista de tendencias de redes estaría completa sin una mención de IPv6. El trabajo original comenzó en 1993 y el diseño ha estado en operación por varios años. En los primeros años, los defensores argumentaron que se necesitaba IPv6 debido a que IPv4 no podía manejar audio o video, no era seguro y se quedaría sin direcciones. Cada año desde la creación de IPv6, varios grupos académicos y de la industria han pronosticado la desaparición de IPv4 y el auge de IPv6. Mientras tanto IPv4 se adaptó, ejecuta aplicaciones multimedia y se hizo tan seguro como IPv6. NAT y CIDR extendieron el direccionamiento IPv4 y, en las principales partes de Internet, IPv4 sigue siendo el protocolo fundamental.

En este momento no hay una razón técnica para adoptar IPv6. De hecho, puesto que el procesamiento de paquetes IPv6 incurre en una mayor sobrecarga, cambiar a IPv6 puede limitar las velocidades a las que pueden enviarse los paquetes. Por lo tanto, la justificación de IPv6 se convierte en una concesión económica: es posible eliminar NAT de Internet y tener un direccionamiento de extremo a extremo, pero hacerlo significaría reemplazar todo el equipo y el software de redes.

A pesar del costo, la tendencia hacia IPv6 ha comenzado. Las empresas como Google están a favor de IPv6 debido a que de esta forma cada dispositivo tendrá una dirección global única. En la actualidad NAT significa que muchos usuarios comparten una dirección global (por ejemplo, una familia en la que tanto padres como hijos usan un enrutador inalámbrico). Si cada dispositivo tuviera una dirección única, Google podría rastrear las solicitudes de búsqueda de cada individuo y generar más ingresos por publicidad al dirigir sus anuncios a la audiencia adecuada.

Los operadores celulares (especialmente en Asia) decidieron que se usará IPv6 para los teléfonos celulares. Al cambiar a un sistema basado en IP, los proveedores celulares tendrán que reemplazar todo el equipo. Calculan que si cambian directamente a IPv6 se ahorrarán una transición más adelante. De cualquier forma ha comenzado el cambio a IPv6 y la tendencia continuará.

33.17 Resumen

Internet sigue evolucionando. Se inventan nuevas aplicaciones y tecnologías de manera constante. Las tendencias actuales incluyen tecnologías para mayor velocidad, movilidad y escalabilidad. En términos de las aplicaciones de Internet, la tendencia ha sido hacia las redes sociales. Además, las nuevas tecnologías han permitido que los usuarios promedio produzcan contenido. Las redes superpuestas ofrecen ventajas para las empresas y el middleware puede ofrecer un solo inicio de sesión a través de una federación de organizaciones. El cambio a IPv6 ha comenzado y se espera que continúe.

EJERCICIOS

- 33.1** Explique cómo es que el almacenamiento de contenido en una caché permite que Internet escale.
- 33.2** ¿En dónde se usa un平衡ador de carga?
- 33.3** Un sitio web con N servidores físicos tal vez no pueda procesar N veces tantas solicitudes por segundo, puesto que los recursos compartidos pueden imponer un cuello de botella. Nombre dos recursos que son compartidos.
- 33.4** Además de permitir el escalamiento, la virtualización de servidores también puede permitir que un sitio ahorre energía durante momentos en que la carga es baja (por ejemplo, en fin de semana). Explique cómo.
- 33.5** ¿Con qué uso general se asocia comúnmente la computación de igual a igual?
- 33.6** ¿Tiene sentido un enfoque de centro de datos distribuido para una empresa en la que cada solicitud web requiere acceso a una base de datos central? ¿Por qué sí o por qué no?
- 33.7** Nombre tres ejemplos de aplicaciones de redes sociales.
- 33.8** ¿Cómo convergen los sistemas de telefonía celular e Internet?
- 33.9** ¿Qué ofrece el video digital a los usuarios?
- 33.10** Cuando se usa fibra óptica para transmitir datos a un hogar o negocio, ¿qué tan rápido pueden enviarse datos en comparación con un módem de DSL o de cable?

- 33.11** Mencione ejemplos de nuevas tendencias de redes para negocios.
- 33.12** ¿Qué tecnologías se usan para ofrecer acceso remoto a las aldeas?
- 33.13** Nombre dos tecnologías que se utilizan para aumentar la velocidad de los enrutadores y comunicadores.
- 33.14** ¿Es un solo inicio de sesión a través de varias organizaciones más seguro o menos seguro? Explique.
- 33.15** Compare las redes superpuestas con MPLS e indique cuál de las dos incurre en un costo de capital inicial más bajo.
- 33.16** ¿Por qué están los proveedores de telefonía celular interesados especialmente en IPv6?

Apéndice 1

Una interfaz de programación de aplicaciones simplificada

Introducción

El capítulo 3 describe la API de sockets que utilizan los programadores para crear clientes y servidores. Este apéndice presenta una alternativa: una API simplificada que permite a un programador construir aplicaciones de red sin tener que dominar los detalles de la interfaz de sockets. El apéndice es independiente de los contenidos del libro y no requiere de una comprensión de Internet o de TCP/IP. Por lo tanto, es posible leer y entender el apéndice antes de terminar de estudiar el resto del libro.

Los ejemplos que presentamos en el apéndice demuestran una idea importante:

Un programador puede crear software de aplicación de Internet sin necesidad de entender la tecnología de red subyacente ni los protocolos de comunicación.

Para demostrar esto presentaremos un pequeño conjunto de funciones de biblioteca que manejan la comunicación y le mostraremos cómo pueden usarse estas funciones para escribir aplicaciones de red. El código de ejemplo del capítulo está disponible en el sitio web y recomendamos a los lectores modificar los ejemplos o escribir aplicaciones adicionales. Para que la biblioteca sea lo más fácil de entender, nos enfocaremos sólo en IPv4. Dejaremos la construcción de una biblioteca para IPv6 como ejercicio para el lector.

Un modelo de comunicación de redes

Toda la transferencia en Internet la realizan los programas de aplicaciones. Cuando las aplicaciones usan Internet, lo hacen en pares. Por ejemplo, cuando un usuario navega por una página web, una aplicación de explorador que se ejecuta en la computadora del usuario hace contacto con una aplicación web que se ejecuta en una computadora remota. El explorador envía una solicitud a la que el servidor web responde. Sólo las dos aplicaciones entienden el formato y el significado del mensaje.

El modelo cliente-servidor

Para comunicarse a través de Internet, un par de aplicaciones usan un mecanismo simple: una aplicación empieza primero y espera a que la otra aplicación la contacte. La segunda aplicación debe conocer la ubicación donde la primera está esperando. Este arreglo se conoce como interacción *cliente-servidor*. El programa que espera el contacto es un *servidor* y el programa que inicia el contacto es un *cliente*. Para iniciar el contacto, un cliente debe saber cómo contactar al servidor. En Internet, la ubicación de un servidor se proporciona por un par de identificadores:

(computadora, aplicación)

donde *computadora* identifica a la computadora en la que se ejecuta el servidor y *aplicación* identifica un programa de aplicación específico en esa computadora. Aunque el software de aplicación representa los dos valores como números binarios, los seres humanos nunca tendrán que lidiar directamente con la representación binaria, sino que a los valores también se les asignan nombres alfábéticos que los seres humanos usan; el software traduce de manera automática cada nombre a un valor binario correspondiente.

Paradigma de comunicación

La mayoría de las aplicaciones de Internet siguen el mismo paradigma básico cuando se comunican. Dos aplicaciones establecen la comunicación, intercambian mensajes entre sí y luego terminan la comunicación. Los pasos son:

- La aplicación servidor comienza primero y espera a que un cliente haga contacto.
- El cliente especifica la ubicación del servidor y solicita que se establezca una conexión.
- Una vez que haya una conexión establecida, el cliente y el servidor usan esa conexión para intercambiar mensajes.
- Cuando terminan de enviar datos, el cliente y el servidor envían cada uno un *fin de archivo* y se termina la conexión.

Un ejemplo de interfaz de programas de aplicación

Hasta ahora hemos hablado sobre la interacción entre dos aplicaciones a nivel conceptual. Ahora consideraremos una implementación detallada. Los científicos de computadoras definen una *interfaz de programas de aplicación (API)* como un conjunto de operaciones disponibles para un programador de aplicaciones. La API especifica un conjunto de funciones, los argumentos para cada función y la semántica de la invocación de una función.

Para demostrar la programación de redes, hemos ideado una API simple para la comunicación de red. Después de describir la API consideraremos las aplicaciones que la utilizan. La figura A1.1 enlista las siete funciones que pueden ser invocadas por una aplicación.

Operación	Significado
await_contact	La utiliza un servidor para esperar el contacto de un cliente
make_contact	La utiliza un cliente para contactar a un servidor
appname_to_appnum	Se usa para traducir el nombre de un programa en un valor binario interno equivalente
cname_to_comp	Se usa para traducir un nombre de computadora en un valor binario interno equivalente
send	La utilizan el cliente o el servidor para enviar datos
recv	La utilizan el cliente o el servidor para recibir datos
send_eof	La utilizan tanto el cliente como el servidor una vez que terminan de enviar los datos

Figura A1.1 Una API de ejemplo que consiste en siete funciones suficientes para la mayoría de las aplicaciones de red.[†]

Nota: nuestro código de ejemplo usará también una octava función: *recvln*. Sin embargo, *recvln* no se enumera como una función independiente ya que simplemente consiste en un bucle que llama a *recv* hasta encontrar un fin de línea.

[†] Las funciones *send* y *recv* las suministra directamente el sistema operativo; las demás funciones en la API consisten en rutinas de biblioteca que nosotros escribimos.

Un análisis intuitivo de la API

Un servidor comienza invocando a *await_contact* para esperar el contacto de un cliente. El cliente comienza invocando a *make_contact* para establecer contacto. Una vez que el cliente contacta al servidor, los dos pueden intercambiar mensajes con *send* y *recv*. Las dos aplicaciones deben programarse para que sepan si deben enviar o recibir; si ambos lados tratan de recibir sin enviar, se bloquearán para siempre.

Al terminar de enviar datos, una aplicación invoca a *send_eof* para enviar la condición de fin de archivo. Del otro lado, *recv* devuelve un valor de cero para indicar que se llegó al fin del archivo. Por ejemplo, si el cliente llama a *send_eof*, el servidor encontrará un valor de cero como respuesta de su invocación a *recv*. Una vez que ambos lados hayan invocado a *send_eof*, la comunicación terminará.

Un ejemplo trivial ayudará a explicar la API de ejemplo. Considere una aplicación en la que el cliente hace contacto con un servidor, envía una sola solicitud y recibe una sola respuesta. La figura A1.2 ilustra la secuencia de llamadas a la API que el cliente y el servidor realizan para dicha interacción.

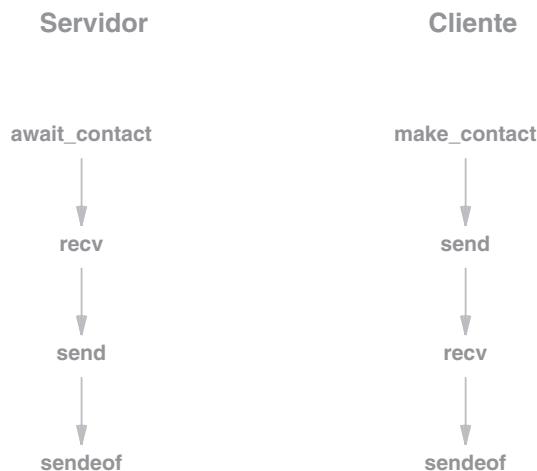


Figura A1.2 Ilustración de las invocaciones a la API que se usan cuando un cliente envía una solicitud y recibe una respuesta del servidor.

Definición de la API

Además de los tipos de datos estándar de C, definimos tres tipos que se usan en todo el código. Al usar tipos, nuestra API es independiente de cualquier sistema operativo y software de red específicos. La figura A1.3 enumera los nombres de los tipos y sus significados.

Nombre del tipo	Significado
appnum	Un valor binario que se usa para identificar una aplicación
computer	Un valor binario que se usa para identificar a una computadora
connection	Un valor que se utiliza para identificar la conexión entre un cliente y un servidor

Figura A1.3 Los tres nombres de tipos que usamos en nuestra API de ejemplo.

Si usamos los tres nombres de tipos de la figura A1.3, podemos definir con precisión la API de ejemplo. Para cada función, las siguientes declaraciones estilo C enlistan el tipo de cada argumento, así como el tipo que devuelve la función.

La función `await_contact`

Un servidor llama a la función `await_contact` para esperar el contacto de un cliente.

```
connection await_contact(appnum a)
```

La llamada recibe un argumento de tipo `appnum` y devuelve un valor de tipo `connection`. El argumento especifica un número que identifica la aplicación servidor; un cliente debe especificar el mismo número al contactar al servidor. El servidor usa el valor de retorno (tipo `connection`) para transferir datos.

La función `make_contact`

Un cliente llama a la función `make_contact` para establecer contacto con un servidor.

```
connection make_contact(computer c, appnum a)
```

La llamada recibe dos argumentos que identifican a una computadora en la que se ejecuta el servidor y el número de aplicación que usa el servidor en esa computadora. El cliente usa el valor de respuesta, que es del tipo `connection`, para transferir datos.

La función appname_to_appnum

Los clientes y servidores usan *appname_to_appnum* para traducir de un nombre de servicio legible por los seres humanos a un valor binario interno. Los nombres de los servicios se encuentran estandarizados en toda Internet (por ejemplo, www indica World Wide Web).

```
appnum appname_to_appnum(char *a)
```

La llamada recibe un argumento del tipo cadena (C usa la declaración *char ** para denotar una cadena) y devuelve un valor binario equivalente de tipo *appnum*.

La función cname_to_comp

Los clientes invocan a *cname_to_comp* para convertir un nombre de computadora legible para los seres humanos en el valor binario interno.

```
computer cname_to_comp(char *c)
```

La llamada recibe un argumento de tipo cadena (*char **) y devuelve un valor binario equivalente de tipo *computer*.

La función send

Tanto clientes como servidores usan *send* para transferir datos a través de la red.

```
int send(connection con, char *bufer, int longitud, int banderas)
```

La llamada recibe cuatro argumentos. El primero especifica una conexión establecida anteriormente con *await_contact* o *make_contact*, la segunda es la dirección de un búfer que contiene datos para enviar, el tercer argumento proporciona la longitud de los datos en bytes (octetos) y el cuarto argumento es cero para la transferencia normal. *Send* devuelve el número de bytes transferidos o un valor negativo si ocurrió un error. Vea también la función auxiliar *send_eof*, que se enumera a continuación, para enviar un *fin de archivo* después de haber enviado todos los datos.

Las funciones *recv* y *recvln*

Tanto clientes como servidores usan *recv* para acceder a los datos que llegan a través de la red. La definición es:

```
int recv(connection con, char *bufer, int longitud, int banderas)
```

La llamada recibe cuatro argumentos. El primero especifica una conexión establecida previamente con *await_contact* o *make_contact*, el segundo es la dirección de un búfer en donde deben colocarse los datos, el tercer argumento proporciona el tamaño del búfer en bytes (octetos) y el cuarto argumento es cero para la transferencia normal. *Recv* devuelve ya sea el número de bytes que se colocaron en el búfer, un cero para indicar que se llegó al *fin de archivo*, o un valor negativo para indicar que ocurrió un error. El código de ejemplo también usa una función de biblioteca *recvln* que llama repetidas veces a *recv* hasta que se haya recibido una línea completa de texto. La definición de *recvln* es:

```
int recvln(connection con, char *bufer, int longitud)
```

La función *send_eof*

Tanto el cliente como el servidor deben usar *send_eof* después de enviar datos para informar al otro lado que ya no habrá más transmisión. Por otro lado, la función *recv* devuelve cero cuando recibe el fin de archivo.

```
int send_eof(connection con)
```

La llamada tiene un argumento que especifica una conexión previamente establecida con *await_contact* o *make_contact*. La función devuelve un valor negativo para indicar que ocurrió un error y un valor de cero en caso contrario.

Resumen de tipos de API

La figura A1.4 sintetiza los argumentos utilizados para cada función en la API de ejemplo. La tabla muestra el tipo de cada argumento así como el tipo de respuesta de la función. La última columna de la figura especifica el tipo de los argumentos más allá de los primeros dos. Aunque cada una de las funciones *send* y *recv* tiene cuatro argumentos, la función de biblioteca *recvln* sólo tiene tres.

Nombre	Devolvió	arg. 1	arg. 2	args. 3 y 4
await_contact	connection	appnum	-	-
make_contact	connection	computer	appnum	-
appname_to_appnum	appnum	char *	-	-
cname_to_comp	computer	char *	-	-
send	int	connection	char *	int
recv	int	connection	char *	int
recvln	int	connection	char *	int
send_eof	int	connection	-	-

Figura A1.4 Un resumen de los tipos de argumentos y de respuesta para la API de ejemplo.

Las siguientes secciones contienen ejemplos de programas de aplicación que ilustran cómo el software cliente y servidor usa nuestra API para comunicarse. Para reducir el tamaño y facilitar la lectura del código, los programas de este capítulo usan argumentos de línea de comandos sin verificar su validez. Un ejercicio sugiere volver a escribir los programas para verificar los argumentos y reportar los errores al usuario.

Código para una aplicación de eco

La primera aplicación que vamos a considerar es simple: un cliente envía datos y el servidor simplemente hace eco de los datos que recibe. Es decir, la aplicación cliente pide repetidas veces al usuario una línea de entrada, envía la línea al servidor y luego despliega lo que el servidor envía de regreso. Aunque no son útiles para un usuario común, las aplicaciones de eco se usan a menudo para probar la conectividad de la red.

Al igual que todas las aplicaciones descritas en este apéndice, la aplicación de eco usa protocolos estándar de Internet. Es decir, los programas cliente y servidor pueden ejecutarse en cualquier computadora conectada a Internet, como se indica en la figura A1.5.

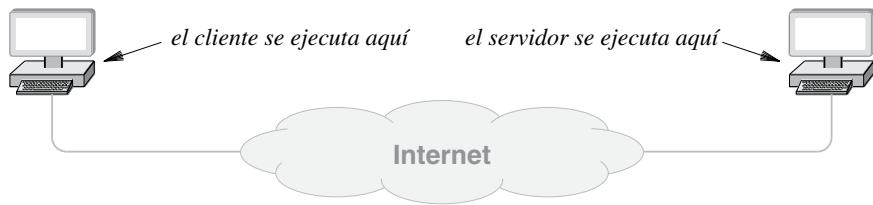


Figura A1.5 El cliente y servidor de eco pueden ejecutarse en cualquier computadora.

Para invocar al servidor, un usuario debe elegir un número de aplicación entre 1 y 32767 que no esté usado por otras aplicaciones, y debe especificar el número como argumento de línea de comandos. Por ejemplo, suponga que alguien que utiliza la computadora *arturo.cs.purdue.eu* selecciona 20000 como el número de aplicación. El servidor se invoca mediante el comando:

```
servidoreco 20000
```

Si alguna otra aplicación está usando el número 20000, el servidor emitirá un mensaje de error apropiado y terminará su ejecución; el usuario deberá elegir otro número.

Una vez que se ha invocado al servidor, se invoca al cliente especificando el nombre de la computadora en la que se ejecuta el servidor y el número de aplicación que usa ese servidor. Por ejemplo, para contactar al servidor antes descrito, un usuario en cualquier computadora en Internet puede introducir el siguiente comando:

```
clienteecho arturo.cs.purdue.edu 20000
```

Código de ejemplo de servidor eco

El archivo *servidoreco.c* contiene el código para el servidor de eco. Lo sorprendente es que incluso con la inserción de comentarios y líneas en blanco adicionales, por cuestión de legibilidad, todo el programa cabe en una sola página del libro. De hecho, después de que el programa hace una verificación para asegurarse de haber sido invocado correctamente, el cuerpo principal del mismo consiste en siete líneas de código:

```

/* servidoreco.c */

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

#define TAMBUFER          256

/*
 * Programa: servidoreco
 * Propósito: esperar una conexión de un cliente y datos de eco
 * Uso: servidoreco <appnum>
 *
 */
int
main(int argc, char *argv[])
{
    conexion      con;
    int           longitud;
    char         buf[TAMBUFER];

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

    /* esperar una conexión de un cliente eco */

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

    /* iterar, haciendo eco de todos los datos recibidos hasta el fin de
     archivo */

    while((longitud = recv(con, buf, TAMBUFER, 0)) > 0)
        (void) send(con, buf, longitud, 0);
    send_eof(con);
    return 0;
}

```

Como hemos visto, el servidor recibe un solo argumento de línea de comandos que especifica el número de aplicación a usar. En C, los argumentos de la línea de comandos se pasan al programa como

un arreglo de cadenas (*argv*) junto con una cuenta entera de argumentos (*argc*). El código extrae el argumento de la línea de comandos de *argv[1]* e invoca a la función estándar de C *atoi* para convertir el valor de una cadena ASCII en binario. Después pasa el resultado como argumento para *await_contact*. Una vez que regresa la llamada a *await_contact*, el servidor llama repetidas veces a *recv* para recibir datos del cliente y a *send* para transmitir los mismos datos de vuelta. La iteración termina cuando *recv* encuentra un fin de archivo y devuelve cero. En ese momento el servidor envía un fin de archivo y termina su ejecución.

Código de ejemplo de cliente eco

El archivo *clienteeco.c* contiene código para una aplicación de cliente eco. Aunque no es tan corto como el servidor eco, el cliente ocupa sólo unas cuantas líneas de código.

```
/* clienteeco.c */

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

#define TAMBUFER          256
#define INDIC_ENTRADA     "Entrada > "
#define INDIC_RECIBIDO    "Recibido> "

int readln(char *, int);

/*
 * Programa:   clienteeco
 * Propósito:  contactar a servidoreco, enviar entrada de usuario e imprimir respuesta del servidor
 * Uso:        clienteeco <nombrecomp> [appnum]
 * Nota:       appnum es opcional. Si no se especifica, se usa el appnum
 *             estándar de eco (7).
 *
 */
int
main(int argc, char *argv[])
{
    computer      comp;
    appnum        app;
    connection    con;
    char          buf[TAMBUFER];
    int           esperados, recibidos, longitud;
```

```
if (argc < 2 || argc > 3) {
    (void) fprintf(stderr, "uso: %s <nombrecorp> [appnum]\n",
                  argv[0]);
    exit(1);
}
/* convierte los argumentos al formato binario comp y appnum */

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

if (argc == 3)
    app = (appnum) atoi(argv[2]);
else
    if ((app = appname_to_appnum("eco")) == -1)
        exit(1);
/* forma una conexión con el servidoreco */

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

(void) printf(INDIC_ENTRADA);
(void) fflush(stdout);

/* itera: lee entrada del usuario, envía al servidor,
   /* recibe respuesta del servidor y despliega para el usuario */

while((longitud = readln(buf, TAMBUFER)) > 0) {

    /* envía la entrada al servidoreco */

    (void) send(con, buf, longitud, 0);
    (void) printf(INDIC_RECIBIDO);
    (void) fflush(stdout);

    /* lee e imprime mismo núm. de bytes del servidor eco */

    esperados = longitud;
    for (recibidos = 0; recibidos < esperados;) {
        longitud = recv(con, buf, (esperados - recibidos) < TAMBUFER ?
                        (esperados - recibidos) : TAMBUFER, 0);
        if (longitud < 0) {
```

```

        send_eof(con);
        return 1;

    }
    (void) write(STDOUT_FILENO, buf, longitud);
    recibidos += longitud;

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

}
/* la iteración termina al encontrar EOF en stdin */

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

```

El programa cliente recibe uno o dos argumentos. El primer argumento especifica el nombre de una computadora en la que se ejecuta el servidor. Si está presente, el segundo argumento especifica el número de aplicación que usa el servidor. Si falta el segundo argumento, el cliente llama a *appname_to_appnum* con el argumento *eco*.

Después de convertir los argumentos al formato binario, el cliente los pasa a *make_contact*, que a su vez contacta con el servidor. Una vez que se establece el contacto, el cliente emite un indicador al usuario y entra en un bucle que lee una línea de entrada, envía la línea al servidor, lee la respuesta de éste e imprime la respuesta para el usuario seguida de un nuevo indicador. Cuando el cliente llega al fin de la entrada (por ejemplo, *readln* devuelve un valor de cero), invoca a *send_eof* para informar al servidor y termina su ejecución.

Varios detalles complican el código para el cliente eco. Primero, el cliente llama a una función *readln* para leer una línea de entrada. Segundo, el cliente evalúa el valor de retorno de cada llamada a la función y termina su ejecución cuando el valor indica que ocurrió un error. Tercero, el cliente llama a *fflush* para asegurar que la salida se despliegue de inmediato en vez de que se acumule en un búfer. Cuarto, y más importante, el cliente no sólo emite una llamada a *recv* cada vez que recibe datos del servidor, sino que entra en un bucle que invoca repetidas veces a *recv* hasta que haya recibido todos los bytes que se enviaron.

El uso de varias llamadas a *recv* conduce a un punto clave sobre nuestra API:

Un receptor no puede asumir que los datos llegarán en piezas del mismo tamaño que las que se enviaron; una llamada a recv puede devolver menos datos de los que se enviaron en una llamada a send.

El texto explica por qué *recv* se comporta así: los datos se dividen en pequeños paquetes para su transmisión. Por lo tanto, una aplicación puede recibir los datos de un paquete a la vez. Lo sorprendente es que también se aplica lo opuesto, ya que aun cuando un emisor llame repetidas veces a *send*, el software de red puede recibir datos de muchos paquetes antes de que la aplicación llame a *recv*. En dichos casos, *recv* devolverá todos los datos a la vez.

Código de ejemplo de un servidor de chat

La segunda aplicación que consideraremos es una forma simplificada de la herramienta de *chat*. Los programas de chat en Internet permiten que un grupo de usuarios se comuniquen mediante la introducción de mensajes de texto que se despliegan en todas las demás pantallas. Nuestro software ofrece una versión simplificada de chat que funciona entre un solo par de usuarios. Cuando un usuario introduce texto, éste se despliega en la pantalla del otro usuario y viceversa. Además, al igual que la aplicación de eco antes descrita, nuestro software de chat puede usarse entre dos computadoras cualesquiera que estén conectadas a Internet. Para comenzar, un usuario selecciona un número de aplicación y ejecuta el servidor. Por ejemplo, suponga que un usuario en la computadora *genoveva.cs.purdue.edu* ejecuta el servidor:

```
servidorchat 25000
```

Un usuario en otra computadora puede invocar al cliente, que hace contacto con el servidor:

```
clientechat genoveva.cs.purdue.edu 25000
```

Para que el código sea lo más corto posible, elegimos un esquema que requiere que los usuarios tomen turnos para introducir texto. Tanto el cliente como el servidor emiten un indicador cuando se espera que el usuario de ese lado introduzca una línea de texto. Primero se pide una entrada al usuario del lado cliente. Cuando se recibe una línea de texto, el cliente envía la línea al servidor y los roles se invierten. Los usuarios se alternan para introducir texto hasta que uno de ellos envíe un fin de archivo.

El código en sí es simple. El servidor comienza esperando el contacto del cliente. Después entra en un bucle en el que obtiene y despliega una línea de texto del cliente, muestra un indicador al usuario local, lee una línea de entrada del teclado y envía la línea al cliente. De esta forma, hasta no recibir un fin de archivo, el servidor itera entre mostrar la salida del cliente y enviar la entrada del teclado al cliente.

El cliente comienza haciendo contacto con el servidor. Una vez que se establece la comunicación, el cliente también entra en un bucle. Durante cada iteración, el cliente pide al usuario local que introduzca una línea de texto, lee una línea del teclado, envíe la línea al servidor y luego reciba y despliegue una línea de texto del servidor. Por ende, el cliente sigue alternando entre enviar una línea de texto que el usuario introduce y desplegar una línea de texto del servidor.

El archivo *servidorchat.c* contiene el código para el servidor de chat.

```
/* servidorchat.c */

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

#define TAMBUFER          256
#define INDIC_ENTRADA     "Entrada > "
#define INDIC_RECIBIDO    "Recibido> "

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

/*
 * Programa: servidorchat
 * Propósito: esperar una conexión de un clientechat y dejar que los
 * usuarios conversen
 * Uso: servidorchat <appnum>
 *
 */
int
main(int argc, char *argv[])
{
    connection  con;
    int         longitud;
    char        buf[TAMBUFER];

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

    (void) printf("Servidor de chat esperando conexión.\n");

    /* espera una conexión de un clientechat */

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

    (void) printf("Se estableció conexión de chat.\n");

    /* itera, leyendo del cliente y del usuario local */
```

```

while((longitud = recvln(con, buf, TAMBUFER)) > 0) { ¢
    (void) printf(INDIC_RECIBIDO);
    (void) fflush(stdout);
    (void) write(STDOUT_FILENO, buf, longitud);

    /* envía una línea al clientechat */

    (void) printf(INDIC_ENTRADA);
    (void) fflush(stdout);
    if ((longitud = readln(buf, TAMBUFER)) < 1)
        break;
    buf[longitud - 1] = '\n';
    (void) send(con, buf, longitud, 0);

}

/* la iteración termina al encontrar EOF en stdin en conexión de
chat */

(void) send_eof(con);
(void) printf("\nConexión de chat cerrada.\n\n");
return 0;

}

```

Las funciones *recvln* y *readln* simplifican el código; cada una consiste en un bucle que itera hasta encontrar una línea completa o fin de archivo. *Recvln* llama a *recv* para recibir de una conexión de red, mientras que *readln* llama a *read* para leer caracteres de un teclado.

La estructura general del servidor de chat es similar al servidor eco que analizamos antes. Al igual que el servidor eco, el servidor de chat espera un solo argumento de línea de comandos que especifica el número de aplicación a utilizar. Una vez que llega el contacto de un cliente, el servidor de chat imprime un mensaje para el usuario local y entra en un bucle. En cada iteración, el servidor recibe una línea de texto de la conexión de red, imprime la línea en la pantalla del usuario, lee una línea de entrada del teclado y envía la línea a través de la red. Cuando detecta un fin de archivo, el servidor envía un fin de archivo y termina de ejecutarse.

Código de ejemplo de un cliente de chat

El archivo *clientechat.c* contiene el código para el cliente de chat. Como es de esperarse, el cliente es un poco más grande que el servidor.

```

/* clientechat.c */

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

```

```
#define TAMBUFER          256
#define INDIC_ENTRADA      "Entrada > "
#define INDIC_RECIBIDO     "Recibido> "

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

/*
 * Programa: clientechat
 * Propósito: contactar a un servidorchat y dejar que los usuarios conversen
 * Uso: clientechat <nombrecomp> <appnum>
 *
 */
int
main(int argc, char *argv[])
{
    computer    comp;
    connection   con;
    char         buf[TAMBUFER];
    int          longitud;

    if (argc != 3) {
        (void) fprintf(stderr, "uso: %s <nombrecomp> <appnum>\n",
                      argv[0]);
        exit(1);
    }

    /* convierte el nombrecap al formato binario comp */

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

    /* hace una conexión con el servidorchat */

    con = make_contact(comp, (appnum) atoi(argv[2]));
    if (con < 0)
        exit(1);

    (void) printf("Se estableció conexión de chat.\n");
    (void) printf(INDIC_ENTRADA);
    (void) fflush(stdout);
```

```
/* itera, leyendo del usuario local y luego del servidorchat */

while((longitud = readln(buf, TAMBUFER)) > 0) {
    buf[longitud - 1] = '\n';
    (void) send(con, buf, longitud, 0);

    /* recibe e imprime una línea del servidorchat */
    if ((longitud = recvln(con, buf, TAMBUFER)) < 1)
        break;
    (void) printf(INDIC_RECIBIDO);
    (void) fflush(stdout);
    (void) write(STDOUT_FILENO, buf, longitud);

    (void) printf(INDIC_ENTRADA);
    (void) fflush(stdout);

}

/* la iteración termina cuando stdin o la conexión indican EOF */

(void) printf("\nLa conexión del chat se cerró.\n");
(void) send_eof(con);
exit(0);

}
```

El cliente comienza contactando con un servidor. Una vez que se establece la comunicación, el cliente entra en un bucle que lee del teclado, envía los datos al servidor, recibe una línea del servidor y despliega esa línea en la pantalla del usuario. La iteración continúa hasta que el cliente recibe una condición de fin de archivo del servidor o un fin de archivo del teclado (un valor de respuesta de cero). En ese momento, el cliente envía un fin de archivo y termina de ejecutarse.

Una aplicación web

La aplicación final de ejemplo que consideraremos consiste en una interacción cliente-servidor para World Wide Web. Para ejecutar el servidor, un usuario selecciona un número de aplicación e invoca al programa servidor. El número de aplicación estándar para un servidor web es 80, pero sólo las aplicaciones privilegiadas pueden especificar el puerto 80. Optamos por usar 27000 en el siguiente ejemplo. Pero si el 27000 no está disponible puede usarse otro número de aplicación, siempre y cuando tanto el cliente como el servidor usen el mismo valor.

Como ejemplo, suponga que un usuario en *micomputadora.edu* opta por ejecutar el servidor y especifica el número de aplicación 27000. El servidor puede invocarse con el comando:

```
servidorweb 27000
```

El programa cliente *clienteweb* puede ejecutarse en cualquier computadora pero necesita conocer la ubicación del servidor. El cliente espera tres argumentos de línea de comandos que especifican la computadora del servidor, un nombre de ruta y un número de aplicación. En nuestro ejemplo, un usuario puede invocar:

```
clienteweb micomputadora.edu/index.html 27000
```

Aunque es mucho muy pequeño, nuestro servidor web cumple con los protocolos estándar. Por lo tanto, es posible usar un navegador web convencional (es decir, comercialmente disponible) para acceder al servidor. Para usar un navegador comercial en vez de nuestro cliente web del ejemplo anterior, hay que escribir el siguiente URL:

```
http://micomputadora.edu:27000/index.html
```

Para que nuestro código fuera lo más breve posible, hicimos unas cuantas suposiciones de simplificación. Por ejemplo, el servidor web sólo provee tres páginas web y lo único que tienen las páginas es texto. Además, cada página está integrada en el código; la página sólo puede cambiarse si se recompila el servidor (los ejercicios que están al final de esta parte sugieren extender el código del servidor para resolver algunas de las limitaciones).

La limitación más importante de nuestra aplicación web recae en el cliente. A diferencia de un navegador web convencional, nuestro código cliente no entiende cómo dar formato a las páginas web y desplegarlas. En su lugar, el cliente sólo imprime el código fuente de la página. A pesar de la limitación, el cliente puede interoperar con un servidor web comercial; puede usarse para imprimir el código fuente de cualquier página disponible en Web.

Código de cliente web de ejemplo

El archivo *clienteweb.c* contiene el código del cliente web.

```
/* clienteweb.c */

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

#define TAMBUFER      256

/*
 * Programa:  clienteweb
 * Propósito: obtiene página de servidor web y la vacía en stdout con
 * encabezados
 * Uso:        clienteweb <nombrecorp> <ruta> [appnum]
 * Nota:       appnum es opcional. Si no se especifica se usa el appnum
 *             estándar de www (80).
 */
int
main(int argc, char *argv[])
{
    computer      comp;
    appnum        app;
    connection    con;
    char          buf[TAMBUFER];
    int           longitud

    if (argc < 3 || argc > 4) {
        (void) fprintf(stderr, "%s%s%s", "uso: ", argv[0],
                      " <nombrecorp> <ruta> [appnum]\n");
        exit(1);
    }

    /* convierte argumentos a computer y appnum binarios */

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

    if (argc == 4)
        app = (appnum) atoi(argv[3]);
    else
        if ((app = appname_to_appnum("www")) == -1)
            exit(1);
```

```
/* contacta al servidor web */

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

/* envía una solicitud HTTP/1.0 al servidor web */

longitud = sprintf(buf, "GET %s HTTP/1.0\r\n\r\n", argv[2]);
(void) send(con, buf, longitud, 0);

/* vacía todos los datos recibidos del servidor a stdout */

while((longitud = recv(con, buf, TAMBUFER, 0)) > 0)
    (void) write(STDOUT_FILENO, buf, longitud);

return 0;

}
```

El código cliente es bastante simple: después de establecer la comunicación con el servidor web, envía una solicitud que debe tener la siguiente forma:

GET /ruta http/1.0 CRLF CRLF

donde la *ruta* indica el nombre de un elemento tal como *index.html*, y *CRLF* representa los dos caracteres *retorno* y *salto de línea*. Después de enviar la solicitud, el cliente recibe e imprime la salida del servidor.

Código de servidor web de ejemplo

El archivo *servidorweb.c* contiene el código de un servidor web (miniatura). El programa contiene tres páginas web y el código necesario para responder a una solicitud:

```
/* servidorweb.c */

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

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

#define TAMBUFER      256
#define NOMBRE_SERVIDOR "Servidor Web Demo CNAI"

#define ERROR_400      "<html><head></head><body><h1>Error 400</h1><p>El servidor no entendió su solicitud.</body></html>\n"
#define ERROR_404      "<html><head></head><body><h1>Error 404</h1><p>Documento no encontrado.</body></html>\n"
#define PAG_INICIO     "<html><head></head><body><h1>Bienvenido al Servidor Demo CNAI</h1><p>Visite: <ul><li><a href=\"http://netbook.cs.pu\ rdue.edu\">La página inicio de Netbook</a><li><a href=\"http://www.comerbooks.com\">Página de inicio de libros de Comer</a></ul></body></html>\n"
#define PAGINA TIEMPO "<html><head></head><body><h1>La fecha actual es\ : %s</h1></body></html>\n"

int recvln(connection, char *, int);
void enviar_encab(connection, int, int);

/*
 * Programa: servidorweb
 * Propósito: servir páginas web integradas a los clientes web
 * Uso: servidorweb <appnum>
 */
int
main(int argc, char *argv[])
{

    connection con;
    int          n;
    char         buf[TAMBUFER], cmd[16], ruta[64], vers[16];
    char         *timestr;
```

```
#if defined(LINUX) || defined(SOLARIS)
    struct timeval      tv;
#elif defined(WIN32)
    time_t              tv;
#endif

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

    while(1) {

        /* esperar contacto de cliente en appnum especificado */

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

        /* lee y analiza la línea de solicitud */

        n = recvln(con, buf, TAMBUFER);
        sscanf(buf, "%s %s %s", cmd, ruta, vers);

        /* omite todos los encabezados — lee hasta que estemos
           \r\n solos */

        while((n = recvln(con, buf, TAMBUFER)) > 0) {
            if (n == 2 && buf[0] == '\r' && buf[1] == '\n')
                break;
        }

        /* revisa si hay un fin de archivo inesperado */

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

        /* revisa una solicitud que no podamos entender */

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

```

/* envía la página web solicitada o un error "no se
   encontró" */

if ( (strcmp(ruta, "/") == 0) ||
     (strcmp(ruta, "/index.html") == 0)) {
    enviar_encab(con, 200, strlen(PAG_INICIO));
    (void) send(con, PAG_INICIO, strlen(PAG_INICIO), 0);
} else if (strcmp(ruta, "/tiempo") == 0) {
#ifndef defined(LINUX) || defined(SOLARIS)
    gettimeofday(&tv, NULL);
    timestr = ctime(&tv.tv_sec);
#endif
#ifndef defined(WIN32)
    time(&tv);
    timestr = ctime(&tv);
#endif
    (void) sprintf(buf, PAGINA_TIEMPO, timestr);
    enviar_encab(con, 200, strlen(buf));
    (void) send(con, buf, strlen(buf), 0);
} else { /* no se encontró */
    enviar_encab(con, 404, strlen(ERROR_404));
    (void) send(con, ERROR_404, strlen(ERROR_404), 0);
}

}
(void) send_eof(conn);
}

}

/*
* enviar_encab - envía un encabezado HTTP 1.0 con estado y longitud de
* contenido dados
*/
void
enviar_encab(connection con, int estado, int longitud)
{
    char *cadest, buf[TAMBUFER];

    /* convierte el código de estado en cadena */

    switch(estado) {
    case 200:
        cadest = "OK";
        break;
    case 400:
        cadest = "Solicitud errónea";
        break;
    case 404:
        cadest = "No se encontró";
        break;
    }
}

```

```

default:
    cadelst = "Desconocido";
    break;

}

/*
* envía una respuesta HTTP/1.0 con encabezados Server,
* Content-Length y Content-Type.
*/
(void) sprintf(buf, "HTTP/1.0 %d %s\r\n", estado, cadelst);
(void) send(con, buf, strlen(buf), 0);

(void) sprintf(buf, "Server: %s\r\n", NOMBRE_SERVIDOR);
(void) send(con, buf, strlen(buf), 0);

(void) sprintf(buf, "Content-Length: %d\r\n", longitud);
(void) send(con, buf, strlen(buf), 0);

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

(void) sprintf(buf, "\r\n");
(void) send(con, buf, strlen(buf), 0);

}

```

Aunque el servidor web puede parecer más complejo que los ejemplos anteriores, la mayor parte de la complejidad resulta de los detalles del servicio web más que de los detalles de red. Además de leer y analizar una solicitud, el servidor debe enviar tanto un encabezado como los datos en la respuesta. El encabezado consiste en varias líneas de texto que se terminan con los caracteres *retorno* y *salto de línea*. Las líneas del encabezado son de la forma que se indica a continuación (donde *tamdatos* indica el tamaño de los datos que siguen, medidos en bytes):

```

HTTP/1.0 status cadena_estado CRLF
Server: Servidor Demo CNAI CRLF
Content-Length: tamdatos CRLF
Content-Type: text/html CRLF
CRLF

```

El procedimiento *enviar_encab* se encarga de la tarea de generar un encabezado. Al invocar a *enviar_encab*, el argumento *estado* contiene un código de estado entero y el argumento *longitud* especifica la longitud del contenido. La instrucción *switch* usa el código para elegir un mensaje de texto apropiado, que se asigna a la variable *cadelst*. *Enviar_encab* usa la función *sprintf* de C para generar

el encabezado completo en un búfer y luego llama a *send* para transmitir las líneas del encabezado a través de la conexión con el cliente.

El código también se complica por el manejo de errores: hay que enviar mensajes de error en un formato que el navegador pueda entender. Si una solicitud se forma de manera incorrecta, nuestro servidor genera un mensaje de error *400*; si el elemento especificado en la solicitud no puede encontrarse (es decir, que la *ruta* sea incorrecta), el servidor genera un mensaje *404*.

Nuestro servidor web difiere de los ejemplos anteriores de una manera considerable: el programa servidor no termina su ejecución después de satisfacer una solicitud, sino que permanece en ejecución, listo para aceptar solicitudes adicionales. Es decir, el programa servidor consiste en un bucle infinito que invoca a *await_contact* para esperar el contacto de un cliente. Cuando llega el contacto, el servidor llama a *recvln* para recibir una solicitud y llama a *send* para enviar una respuesta. Después, el servidor regresa a la parte superior del ciclo para esperar el siguiente contacto. Por lo tanto, una vez que inicia, el servidor se ejecuta de manera indefinida, justo igual que un servidor web comercial.

Manejar varias conexiones con la función select

Aunque nuestra API de ejemplo soporta la interacción de 1 a 1 entre un cliente y un servidor, la API no soporta la interacción de 1 a varios. Para entender por qué, considere varias conexiones. Para crear dichas conexiones, un programa de aplicación individual debe llamar a *make_contact* varias veces, especificando una computadora (*computer*) y un número de aplicación (*appnum*) para cada llamada. Pero una vez que se establecen las conexiones, la aplicación no puede saber cuáles conexiones recibirán primero un mensaje. La aplicación no puede usar *recv* debido a que la llamada se bloqueará hasta que lleguen datos.

Muchos sistemas operativos incluyen una función llamada *select* que resuelve el problema de manejar varias conexiones. En concepto, la llamada a *select* verifica un conjunto de conexiones. La llamada se bloquea hasta que al menos una de las conexiones especificadas haya recibido datos. Después, la llamada devuelve un valor que indica qué conexiones recibieron datos (es decir, las conexiones para las que *recv* no se bloqueará).

Como ejemplo, considere una aplicación que debe recibir solicitudes y enviar respuestas a través de dos conexiones. Dicha aplicación puede tener la siguiente forma general:

```
Llamar a make_contact para formar la conexión 1;  
Llamar a make_contact para formar la conexión 2;  
Repetir de manera indefinida {  
    Llamar a select para determinar qué conexión está lista  
    Si (conexión 1 está lista) {  
        Llamar a recv para leer solicitud de conexión 1;  
        Calcular respuesta a solicitud;  
        Llamar a send para enviar respuesta a través de conexión 1;  
    } si (conexión 2 está lista) {
```

```
        Llamar a recv para leer solicitud de conexión 2;  
        Calcular respuesta a solicitud;  
        Llamar a send para enviar respuesta a través de conexión 2;  
    }  
}
```

Resumen

Es posible para un programador crear aplicaciones de red que operen a través de Internet sin necesidad de entender cómo operan las redes o cómo las tecnologías usadas transportan los datos entre computadoras. El programador debe recibir un conjunto de funciones de alto nivel que formen una Interfaz de programas de aplicación (API). Este apéndice presenta una API de redes que contiene sólo siete primitivas y revisa aplicaciones de ejemplo que muestran que la API es suficiente para construir software que pueda interoperar correctamente con software comercial.

EJERCICIOS

- A1.1** Los ejemplos de código de este apéndice no verifican con cuidado sus argumentos de línea de comandos. Modifique el código para agregar comprobación de errores.
- A1.2** El servicio *echo* es un servicio estándar disponible a través de Internet. Se le asignó el número de aplicación 7. Descargue, compile y use el cliente de eco para determinar si las computadoras en su organización ejecutan un servidor eco estándar.
- A1.3** Modifique el servidor eco de modo que en vez de terminar su ejecución después de manejar un cliente, el servidor espere a otro cliente. (Sugerencia: vea el servidor web).
- A1.4** Descargue, compile y pruebe el software de chat de ejemplo, ejecutándolo en dos computadoras.
- A1.5** Nuestro software de chat requiere que los usuarios tomen turnos para introducir texto. Vuelva a escribir el software para que permita que cualquier usuario escriba un número arbitrario de líneas en cualquier momento. (Sugerencia: use hilos).
- A1.6** Modifique el cliente del chat para que envíe un nombre de usuario con cada mensaje y modifique el servidor para que identifique a un usuario cuando despliegue una línea de salida.
- A1.7** Extienda el ejercicio anterior para que, en vez de enviar el nombre de usuario con cada mensaje, el cliente y el servidor del chat intercambien nombres de usuario cuando hagan contacto por primera vez, recuerden los nombres y desplieguen el nombre apropiado con cada línea de salida.
- A1.8** ¿Por qué el código de ejemplo en este apéndice usa una mezcla de llamadas a *write* y varias formas de *printf*? (Sugerencia: ¿acaso Windows trata de manera idéntica a los sockets, archivos y canalizaciones?)
- A1.9** Idee un software que permita una sesión de chat de *n* vías en la que un usuario pueda unirse y dejar la sesión en cualquier momento.
- A1.10** Use *telnet* para contactar a un servidor web, enviar una solicitud *GET* y recibir una respuesta.

- A1.11** Pruebe el programa cliente web con un servidor web de Internet. Para ello, proporcione el nombre del servidor, una ruta de *index.html* o *index.htm* y el número de aplicación 80.
- A1.12** Agregue otra página al servidor web.
- A1.13** Modifique el servidor web de modo que extraiga el contenido de cada página de un archivo en vez de tenerlas integradas en el código.
- A1.14** Expanda el ejercicio anterior para reconocer los nombres de archivos que terminen en *.gif* y envíelos usando un encabezado *Content-type* con el valor *image/gif* en vez de la cadena *text/html*.
- A1.15** (Avanzado) Cree un cliente y un servidor para un servicio de transferencia de archivos.
- A1.16** (Avanzado) Implemente la Interfaz común de puerta de enlace (CGI) usando la especificación en el RFC 3875:

<http://www.ietf.org/rfc/rfc3875>

- A.17** (Avanzado) Extienda el servidor web de modo que pueda manejar varias conexiones al mismo tiempo. (Sugerencia: use *fork* o *pthread_create*).
- A1.18** (Avanzado) Cree un cliente que contacte a un servidor de correo electrónico SMTP y envíe un mensaje de correo electrónico.

Índice

Constantes y elementos numéricicos

1 a 1 y 1 a varios, 419,614
1000BaseT, 261
100BaseT, 261
10Base2, 257
10Base5, 256
10BaseT, 261
125 µ segundos, 214
127, dirección, 358
128 Kbps, 202
16QAM, 173
1G, 2G, 2.5G, 3G y 4G, conexiones inalámbricas, 282
1xRTT, 283
2430 octetos, 214
2B+D, 202
2-PSK, 173
4G, 282, 283
4-PSK, 173
64 Kbps, 202
6LoWPAN, adaptación, 574
6LoWPAN-ND, 574
801.1d-2004, 297
802.11-2007, 269
802.15, 276
802.1d, 297
802.1q, 297
802.1q-2003, 297
802.1w, 297
802.3 Ethernet, 255

A

accept, función de socket, 42
acceso

archivos, remoto, 59
basado en el contenido, 272
libre de contención, 272
lista de control, 514
múltiple por detección de portadora con detección de colisiones, 247
múltiple por detección de portadora con evasión de colisiones, 249, 272
múltiple por división de código, 193
múltiple por división de frecuencia, 241
múltiple por división de tiempo, 242
punto de, 270
retraso de, 471
tecnología de, 199
Wi-Fi protegido, 528
ACK, 429, 434
ACL, 514
adaptativo(a)
puente, 294
retransmisión, 435
administración de elementos, 536
administrador, 538
ADSL, 203
AF_INET, 38, 41
AF_INET6, 38, 41
Agencia de investigación de proyectos avanzados, 18, 328
Agencia de proyectos de investigación avanzados de defensa, 18
agente, 538
de retransmisión, 405
AGS-F, 498
aislado, 455
a-law, codificación PCM, 109
alcance (de dirección IPv6), 362

- Alianza de industrias electrónicas, 118, 157
 alianza ZigBee, 571
 alimentación a través de Ethernet, 495
 almacenamiento y reenvío, 309
 ALOHAnet, 245
 alta velocidad, 469
 amplitud, 95
 análisis de causa raíz, 535
 ancla, etiqueta, 54
 ancho de banda, 99, 200, 473
 ancho de banda analógico, 99
 anillo
 red, 215
 topología, 228
 vea IBM Token Ring
 anillo en sentido contrario, 215
 ANSI, 118
 antena de plato, 284
 antena parabólica (satélite), 284
 antenas omnidireccionales, 279
 aperiódico, 94
 API, 36, 591
 simplificada, 589
 sockets, 36
 aplicación
 correo electrónico, 64
 eco, 596
 orientada a datos, 569
 orientada a la conexión, 569
 aplicaciones
 interfaz de programación de, 36, 591
 servidor de, 498
 aprovisionamiento, 479, 560
 árbol de expansión distribuido, 296
 árbol de expansión por VLAN, 297
 árboles basados en ubicación central, 463
 área (OSPF), 459
 argc y argv, 599
 ARP
 caché, 396
 protocolo, 393
 ARPA, 18, 328
 ARPANET, 18, 328
 ARQ, 137
 arquitectura de igual a igual, 35, 581
 AS, 451
 AS-F, 498
 asignación de subcanales, 185
 asignación estática de canales, 240
 asíncrona
 TDM, 191
 transmisión, 157
 ASK, 169
 ASN.1, 499, 539
 asociación (inalámbrica), 271
 ataque por inundación SYN, 509, 522
 ATM, 330
 AUI, 256
 autenticación, 499, 512, 513
 autoconfiguración (IPv6), 405
 autorización, 499, 512
 avance de línea, 56
- B**
- backhaul, 274
 baja velocidad, 469
 bajo demanda, 209
 balanceador de carga, 580
 banda de guarda, 184
 banda ultra ancha, 277
 base de información de administración, 540
 Base64, codificación, 68
 BER, 499
 Berkeley, difusión, 359
 BGP 454
 Big-Endian, 156
 bind, función de socket, 40
 bit
 inicio, 157
 más significativo, 156
 menos significativo, 156
 parada, 158
 paridad, 139
 bits por segundo, 472
 Bluetooth, 223, 266, 276
 bobinas de carga, 209
 BOOTP, 402
 bootstrap, 402
 bps, 472
 BPSK, 173
 BSD UNIX, 359
 BSS, 272
 búfer, 309
 búfer de inestabilidad, 490
 buzón de correo, 64
 byte, 139

C

- cable
 - coaxial, 115, 117
 - módem, 206, 326
 - televisión, 206
- caché (Akamai), 580
- CA-F, 497
- calidad del servicio, 479
- CAN, 223
- canal, 183, 204
 - capacidad, 129
 - codificación, 137
 - entrulado, 189
- canal B, 202
- canal D, 202
- canalización de tablas de flujo, 562
- capa, 9
 - de sockets seguros, 528
 - de transporte, 416
 - física, 10
 - interfaz de red, 10
 - intermedia, 574
- capa 2, interruptor de, 297
- capacidad, 472
- carga útil, 370
- CATV, 206
- caudal útil, 472
- CBT, 463
- CCITT, 13, 328
- CDDI, 327
- CDM, 193
- CDMA, 193, 242
- CDMA 2000, 283
- CDMA de banda ancha, 283
- celda personal, 280
- centro de conmutación móvil, 278
- cero compresión, 364
- chat, 602
- cifrado, 513, 514
 - carga útil, 526
 - clave, 514, 515
- CIR, 479
- circuito
 - alimentador, 207
 - conmutación, 220
 - de punto a punto, 210
 - digital, 210
- permanente, 220
- virtual, 220
- clase de tráfico, 374, 482
- clase de una dirección, 349
- clasificación (QoS), 483
- clave, 514, 515
 - descifrado, 514, 515
 - secreta, 513
- CLI, 549
- cliente, 590
- clientechat en *clientechat.c*, 604
- clienteeco en *clienteeco.c*, 599
- clienteweb.c, 608
- close, 40, 45
- closesocket, 40
- CMTS, 208
- CNAME, 76
- CO, 202
- codificación
 - condicional de desfase, 106
 - desplazamiento, 169
 - desplazamiento de amplitud, 169
 - desplazamiento de fase, 169
 - desplazamiento de fase binaria, 173
 - desplazamiento de frecuencia, 169
 - línea, 104
 - Manchester diferencial, 106
 - tonos, 498
- código de autenticación de mensajes, 513
- código de redundancia cíclica, 145
- código RAC, 142
- colapso por congestión, 433
- colisión, 246
- colores en WDM, 187
- columna (SONET), 214
- Comisión federal de comunicaciones, 184
- Comité consultivo internacional telegráfico
 - y telefónico, 13, 328
- compresión, 109
 - con pérdida, 109
 - sin pérdida, 109
- comprobación de paridad simple, 139
- comprobación de redundancia cíclica, 428, 513
- computación en la nube, 1, 23, 584
- comunicación
 - línea de alimentación, 327
 - paradigma, 28, 590
 - protocolo, 8

- punto a punto, 122, 220
 - varios a varios, 222
- concatenación, 213
- concentrador, 228
- conector RJ45, 261
- conexión de control (FTP), 60
- confidencialidad, 512, 514
- configuración, 402
 - configuración (FCAPS), 535
 - confirmación, 429, 434
 - comunicación, 297
 - comunicación de etiquetas, 330
 - comutador de red de área local virtual, 299
 - connect, 40
 - contabilidad (FCAPS), 535
 - contenido
 - almacenar en caché, 580
 - análisis, 522
 - conteo de referencia, 45
 - contraseña, 514
 - control de congestión, 440
 - control de enlace lógico, 225, 255
 - controlador de acceso, 495, 497
 - convergencia de rutas, 321
 - COPS, 484
 - corrección de errores en recepción, 137
 - correo
 - aplicación de interfaz, 64
 - servidor de, 64
 - correo de voz, 494
 - CRC, 145, 428
 - criptografía, 514
 - CRLF, 55, 609
 - CSMA/CA, 249, 272
 - CSMA/CD, 247
 - CSU, 211
 - CTS, 273
 - cuantiza, 107
- D**
- DARPA, 18
- datos
 - centro de, 23, 582
 - conexión (FTP), 60
 - confidencialidad, 512
 - disponibilidad, 512
 - equipo de comunicaciones, 162
- equipo terminal, 162
- integridad, 512
- plano de, 548
- relleno de, 233
- sobre cable, especificaciones de la interfaz del sistema de servicio, 209
- unidad de servicio, 211
- velocidad de, 129
- DAYTIME, protocolo, 50
- dB, 130
- DCE, 162
- DCF, 272
- DDoS, 509, 510
- decibeles, 130
- demodulador, 174
- DES, cifrado, 515
- desbordamiento de búfer, 509
- descarga, 60
- descarte aleatorio anticipado, 483
- descarte de la parte final, 483
- descifrar clave, 509
- descifrar contraseña, 509
- descriptor, 36
- descubrimiento de ruta, 402
- descubrimiento de vecindario, 401
- destino
 - gráfico acíclico dirigido orientado al destino, 575
 - inalcanzable, 399
- detección automática, 261
- detección de fallas (FCAPS), 534
- detección de portadora, 246, 249
- detección y corrección de errores, 137
- DHCP, 402
 - DHCP, agente de retransmisión, 405
 - DHCPv6, 404
- diagrama de constelación, 171
- diccionario, 110
- DiffServ, 485
- DIFS, 273
- difusión limitada, 358
- difusión por ruta inversa, 463
- digital
 - ancho de banda, 473
 - certificado, 513
 - circuito, 210
 - firma, 513, 516
- Dijkstra, algoritmo de, 320

dirección, 229
 bucle de retorno, 127, 358
 con clases, 349
 destino, 254, 348, 373
 difusión dirigida, 357
 esta computadora, 358
 falsificación, 509
 loopback, 358
 máscara, 353, 402
 máscara de subred, 351, 353
 no enrutable, 407
 origen, 254, 348, 373
 prefijo de red, 357
 privada, 407
 punto final, 40
 reservada (IP), 357
 resolución, 392, 499
 sin clases, 351
 subred, 351
 todos ceros, 357, 358
 todos unos, 357, 358
 vinculación, 401
direcccionamiento
 con clases, 349
 jerárquico, 309
 sin clases, 351
dispersión (fibra óptica), 120
distorsión, 136
distribución de software Berkeley, 359
distribución en capas, 398
distribuida
 cálculo de ruta, 316
 centros de datos, 582
 función coordinada, 272
 negociación de servicio, 510
DMT, 204
DNS, 69
DNS, servidor raíz, 72
DNS, solicitud o respuesta, 72
DOCSIS, 209
DODAG, 575
dominio
 difusión, 299
 nivel superior, 69
 SDN, 553
 tiempo, 98
DoS, 509, 510
dos veces NAT, 410

DSL, 202, 326
 alta velocidad de datos, 203
 simétrica, 203
 tasa de bits muy alta, 203
DST, 296
DSU/CSU, 211
DTE, 162
DTMF, 498
dúplex, 160, 161
duplicación de paquetes, 428
DVMRP, 463
DVR, 318
DWDM, 187

E

E.164, 502
E1, E2 y E3, 212
ECN, 441
EDGE y EDGE Evolution, 282, 283
EGP, 451, 452
EGPRS, 282, 283
EIA, 118, 157
en línea, 406
encabezado
 base, 373
 fragmentable, 383
 no fragmentable, 383
encabezados de extensión, 373
encapsulamiento, 379
encriptación simétrica, 515
enlace, 52, 313
enrutador, 337
enrutador de frontera (ZigBee), 573
enrutamiento, 447
 bucle, 321
 dinámico, 315, 447, 449
 estado de enlace, 316, 458
 estático, 315, 447, 448
 métrica, 453
 multidifusión, 461
 protocolo de información, 456
 protocolo para redes con pérdida y bajo consumo de energía, 575
 tabla, 311
entramado, 159
entrega desordenada, 428
ENUM, 498, 502

- EOT, 232
- error de supresión, 136, 137
- error de un solo bit, 137
- escalabilidad, 306
- espacio entre paquetes, 247
- espectro amplio, 268
- estabilidad, 461
- establecimiento de enlace de malla, 574
- estación base, 270
- estándares de la serie T, 211
- estándares de nivel de señal digital, 212
- estándares DS, 212
- estructura, 298
- Ethernet, 246
 - 10Base5, 256
 - AUI, 256
 - cable delgado, 257
 - cable grueso, 256
 - cableado de Thinnet, 257
 - comutador, 297
 - dirección, 229
 - formato de trama, 254
 - par trenzado, 258
 - repetidor, 293
- etiqueta
 - RFID 278
 - tiempo, 286
 - VLAN, 300
- ETIQUETA FLUJO, 374
- evaluación de tendencias, 535
- evaluación del diagnóstico, 535
- evasión de colisiones, 249
- EVDO, 283
- EVDV, 283
- evolución a largo plazo, 283
- Evolution
 - Data Only, 283
 - Data Optimized, 283
- extensiones multipropósito de correo
 - de Internet, 68
- extremo a extremo, 416, 427
- F**
 - falsificación, 509
 - de nombres, 509
 - falso positivo, 523
 - fase, 95
- FCC, 184
- FDDI, 327
- FDMA, 241
- FEC, 137
- femtocelda, 280
- fibra
 - hasta X, 208
 - híbrida coaxial, 207
 - índice escalonado, 120
 - índice gradual, 120
 - multimodal, 120
 - óptica, 119, 120
 - unimodal, 120
- fila (SONET), 214
- fila y columna, código, 142
- filtro, 520
 - filtro divisor, 205
- fin de archivo, 50, 590
- fin de transmisión, 232
- firewall, 513, 519
- flujo
 - ascendente, 200
 - control, 430
 - descendente, 200
 - tabla, 562
- formato de imagen de gráficos, 57
- fractal, 478
- fragmentación, 381
- frecuencia, 95
 - modulación, 167
 - multiplexación por división de, 182, 204
- FSK, 169
- FTP, 60, 472
 - FTP, inicio de sesión anónimo, 61
- FTTB, FTTC, FTTH y FTP, 208
- función coordinada por punto, 272
- función de puerta de enlace de señalización, 498
- función seno, 95
- G**
 - garantías de servicio, 479
 - GEO, 125, 126
 - GET (HTTP), 609
 - gethostbyaddr, 44
 - gethostbyname, 44, 72
 - gethostname, 44
 - getpeername, 44

getsockopt, 44
GIF, imagen, 57
Gig-E, 261
global,
 sistema de posicionamiento, 285
 sistema para comunicaciones móviles, 282
GPRS, 283
GPS, 285
granularidad de QoS, 480
grupo
 celular, 280
 direcciónamiento, 363
 satélite, 128
Grupo unido de expertos en fotografía, 57
GSM, 282, 283

H

H.323, 331, 494, 497, 499
Hamming, distancia, 140
hashing, 513
 criptográfico, 513
HDSL, 203
HFC, 207
hilo de ejecución, 34
hipermedios, 52
hipertexto, 52
hipervínculo, 52
hombre en el medio, 509, 510
host, 342
HSCSD, 283
HSDPA, 283
HTC Evo 4G, 283
HTML, 52
HTTPS, 528
huella, 285

I

IBM Token Ring, 327
ICANN, 69, 351
ICMP, 399, 400, 402
ICMPv4, 399
ICMPv6, 399
iDEN, 282, 283
identificador (ID) de llamadas, 494
identificador único de organización, 229
IDNA, 77

IDS, 522
IEEE, 223
 801.1d-2004, 297
 802.11, 267
 802.16, 274
 802.1d, 297
 802.1q, 297
 802.1q-2003, 297
 802.1w, 297
IETF, 494
IGMP, 462
IGP, 451
ILD, 121
implementación de arreglos, 128
IMT-Advanced, 283
INADDR_ANY, 41
inalámbrico(a), 123, 265
 1G, 2G, 2.5G, 3G y 4G, 282
enrutador, 410
estación, 270
independencia del origen, 313
índice de modulación, 168
industrial, científico y médico, 267
inestabilidad, 160, 453, 474
información de estado, 522
infrarrojo
 (IR), 119, 122
 asociación de datos, 277
inicio de encabezado, 232
inicio de sesión anónimo, 61
inspección detallada de paquetes, 513
Instituto de ingenieros eléctricos y electrónicos, 223
Instituto nacional estadounidense de estándares, 118
integridad, 512
intercepción de paquetes, 509
intercepción electrónica, 509
interconexión
 datos distribuidos por cobre, 327
 datos distribuidos por fibra, 327
 redes, 5, 337
 sistemas abiertos, 13
interfaz de unidad de conexión, 256
interferencia, 136
interfuncionamiento, función de, 498
internacional
 Consorcio Softswitch, 497
 organización de estandarización, 13

- telecomunicaciones móviles avanzadas, 283
 - Unión de telecomunicaciones, 13, 328, 494, 534
 - Internet, 339
 - Corporación de nombres y números asignados, 69, 351
 - dirección, 348
 - dirección de protocolo, 33, 348
 - enrutador, 337
 - enrutamiento, 447
 - firewall, 519
 - global, 339
 - Grupo de tareas sobre ingeniería, 494
 - núcleo, 209
 - paradigma, 28
 - protocolo, 341, 345
 - protocolo de mensajes de control, 399
 - protocolo de multidifusión de grupos, 462
 - proveedor de servicios, 6, 199
 - suma de verificación, 144
 - suscriptor, 199
 - tecnología de acceso, 199
 - interoperabilidad inalámbrica para el acceso a través de microondas, 274
 - interoperar, 8, 50
 - interred (definición), 337
 - intranet, 339
 - IntServ, 481
 - IP
 - datagrama, 370
 - dirección, 33, 348
 - dominio administrativo de telefonía, 503
 - opciones, 372
 - seguridad, 528
 - telefonía, 489, 493
 - teléfono, 495
 - túnel, 526
 - IP en IP, 463, 526
 - IP en TCP, 527
 - ipEntRecibe, 540
 - IPPROTO_TCP, 39
 - IPPROTO_UDP, 39
 - IPsec, 528
 - IPv6
 - autoconfiguración, 405
 - descubrimiento de vecindario, 401
 - IPv6-ND, 401
 - IR *vea* Infrarrojo
 - IrDA, 277
 - IS-95A, 282, 283
 - IS-95B, 283
 - ISC, 497
 - ISDN, 202
 - IS-IS, 460
 - ISM inalámbricas, 267
 - ISO, 13
 - ISP, 6, 199
 - ITAD, 503
 - ITU, 13, 328, 494, 534
 - IW-F, 498
- J**
- JPEG, 57
- L**
- lambda (en óptica), 187
 - LAN, 223
 - láser, 119
 - latencia, 470
 - LED, 121
 - lenguaje de marcación, 52
 - extensible, 78, 582
 - hipertexto, 52
 - lenguaje declarativo (HTML), 52
 - LEO, 125, 128
 - ley de Shannon–Hartley, 120
 - libre para enviar, 273
 - libro de códigos, 140, 142
 - línea
 - codificación, 104
 - de suscriptor, 202, 326
 - de suscriptor digital asimétrica, 203
 - de visión, 275
 - lista negra, 523
 - listen, 42
 - listo para enviar, 273
 - llamada
 - agente, 497
 - bifurcación, 501
 - reenvío o espera, 494
 - llamadas de conferencia, 494
 - LLC / SNAP, 255
 - LLC, 225
 - Little-Endian, 156

localidad de referencia, 74
localizador uniforme de recursos, 54
longitud (802.3), 255
LONGITUD CARGA ÚTIL, 374
loopback, 211
LOS, 275
LSB, 156
LTE, 283
LTE Advanced, 283

M

M.3400, 534
MAC, 225, 513
MAC, capa, 10
malware, 522
MAN, 223
máquina virtual, 547, 581
máscara, 353
MBONE, 463
MCU, 499
MCU (H.323), 497
MD5, 513
MD-F, 498
medición activa, 478
medición pasiva, 478
medida de distancia (peso), 318
medios
 control de acceso, 225
 controlador de puerta de enlace, 495, 497
 puerta de enlace, 495, 498
 servidor, 498
Megaco, 494, 496
mejor esfuerzo, 378
mejorada
 GPRS, 282
 velocidad de datos para GSM Evolution, 282
memoria ternaria de contenido direccional, 556
mensaje opaco, 232
MEO, 125
mezclado, 492
MGC-F, 497
MGCP, 494, 496
MG-F, 498
MIB, 540
microcelda, 280
middleware, 586

MIME, 68
MIMO, 286
mínima distancia de Hamming, 142
MISTP, 297
MLE, 574
modelo
 distribución en capas, 9
 FCAPS, 534
 referencia de 5 capas 10, 341
 referencia de 7 capas, 13
 reloj de arena, 346
módem, 174
 externo, 175
 extremo cercano, 208
 extremo lejano, 208
 fibra, 292
 interno, 175
 marcación telefónica, 175
 óptico, 174
 radiofrecuencia, 174
 RF, 174
modo de transferencia asíncrona, 330
modo de transmisión, 153
modo promiscuo, 293
modulación, 166, 174, 204
 amplitud en cuadratura, 173
 código de pulso, 107, 494
 desplazamiento de fase, 168
 multitono discreto, 204
modulador, 174
MOSPF, 464
mrouted, 463
MSB, 156
MSTP, 297
MTU, 381, 384
MTU de ruta, 384
muestra, 107
mu-law (μ -law), codificación, PCM 109
multifrecuencia de tono dual, 498
multimedia, 489
 en tiempo real, 489
múltiple entrada, múltiple salida, 286
múltiple(s)
 instancias, protocolo de árbol de expansión, 297
 protocolo de árbol de expansión, 297
multiplexación, 182, 286
 densa por división de longitud de onda, 187

espacial, 286
 estadística, 191
 inversa, 204
 por división de código 182, 193
 por división de longitud de onda, 182, 187
 por división de tiempo, 182, 187
 multiplexor de adición/supresión, 215, 326
 multiproveedor, 361, 455

N

NAPT, 409
 NAT, 62
 negación de servicio, 509, 510
 negociación de 3 vías, 438
 NetFlow, 479
 NewReno, TCP 441
 NIC, 229, 256
 NIU, 211
 NLOS, 275
 no periódico, 94
 no terrestre, 125
 nodo, 313
 nombres de dominio internacionalizados, 77
 notación
 CIDR, 354
 decimal con puntos, 350
 hexadecimal de dos puntos, 363
 sintaxis abstracta, 1, 539
 notificación de congestión explícita, 441
 NTP, 498
 núcleo de Internet, 209
 número de secuencia (RTP), 492
 Nyquist, 129

O

o exclusivo, 141, 148
 objetivo (ARP), 395
 OC, 213, 326
 oferta (DHCP), 403
 oficina central, 202
 opciones en IPv4, 372
 OpenFlow, 332, 554
 óptica(o)
 fibra, 119
 módem, 174
 órbita terrestre baja, 128

órbita terrestre geoestacionaria, 126
 orientado a flujos, 28
 orientado a la conexión, 29, 369, 427
 orientado a mensajes, 28, 417
 OSI, 13
 OSPF, 458, 459
 OSPFv3, 458
 OUI, 229

P

p2p, 35, 581
 palabra de código, 140
 palabra de datos, 140
 PAN, 223, 266
 paquete universal, 370
 paquetes
 analizador de, 537
 comutación de, 18, 221
 duplicación de, 428
 filtro de, 520
 intercepción de, 509
 orden de, 428
 pérdida de, 429
 puesta en búfer de, 309
 tren de, 476
 par trenzado, 115
 blindado, 115, 118
 sin blindaje, 115, 118
 parada y arranque, 430
 paradigma cliente-servidor, 30, 590
 paradigma de mensaje, 29
 paradigma de obtener-almacenar, 539
 paridad, 139, 428
 paso de token, 244
 patrón comodín, 555
 PBR, 481
 PBS, 481
 PCF, 272
 PCM, 107, 494
 PDC, 282, 283
 pequeña oficina en casa, 7, 209
 pérdida, 429
 perímetro seguro, 519
 periódica, 94
 periodo y frecuencia, 96
 peso de un enlace, 320
 PGP, 528

- picocelda, 280
pila, 9
PIM-DM, 464
PIM-SM, 464
ping, 400
plan de control, 548
podar, 464
PoE, 495
política de seguridad, 511
políticas (QoS), 483
portadora, 165
portadora óptica, 213, 326
POTS, 203
preámbulo, 157
principio de localidad, 74
privacidad, 512
 bastante buena, 528
 equivalente por cable, 528
privada
 clave, 515
 dirección, 407
 red, 6, 7
problema de distribución de claves, 518
problema de la estación oculta, 249, 272
proceso, 34
producto del retraso por la velocidad
 de transferencia, 476
programación de tráfico, 483
 QoS, 483
propagación electromagnética, 124
propiedad, 6
 una sola vía, 516
protocolo, 8
 acceso aleatorio, 240, 244
 acceso controlado, 240
 analizador, 537
 árbol de expansión, 296
 bootstrap, 402
 canalización, 240
 capa de aplicación, 50
 configuración dinámica de host, 402
 control de transmisión, 416, 426
 datagrama de usuario, 416
 descripción de sesión, 500
 extremo a extremo, 416
 delgado (UDP), 416
 familia de, 9
 independiente, 464
 inicio de sesión, 331, 494, 500
 número de puerto de, 33, 420
 puerta de enlace de límite, 454
 puerta de enlace exterior, 451, 452
 puerta de enlace interior, 451
 reservación de recursos, 484
 resolución de direcciones, 393
 resolución inversa de direcciones, 402
 ruta más corta primero, 458
 simple de administración de red, 539
 simple de transferencia de archivos, 64
 suite, 9
 tiempo real, 474
 transferencia de archivos, 59, 60, 472
 transporte, 416
protocolos web, 51
proveedor de servicios, 6
proxy, 523
 web, 523
PSTN, 493
puente, 293
 adaptativo, 294
 con capacidad de aprendizaje, 294
 de conexión, 209
puerta de enlace
 de acceso, 498
 de señalización, 496
 H.323, 497
puerto, 297
 concentrador, 258
 eco, 420
 escaneo de, 509, 522
 timeserver, 420
puesta en cola para QoS, 483
punto de conexión a la subred, 255
punto final, 416
Puny (algoritmo o código), 77
PVST, 297
- Q**
- Q.931, 497
QAM, 173
QoS, 479
- R**
- radio programable, 286
radiofrecuencia, 123

- radioteléfono móvil celular, 282
- RADIUS, 528
- ráfaga, 477
 - errores, 137
 - tamaño, 137
- RARP, 402
- read, 39
- recolección de energía, 570
- reconocimiento positivo con confirmación, 429
- reconocimiento selectivo, 441
- recv, 39
- recvfrom, 44
- recvmsg, 44
- red
 - ad hoc, 270
 - área amplia, 223
 - área de chip, 223
 - área local, 223
 - área metropolitana, 223
 - área personal, 223, 266
 - comutada, 297
 - controlador de interfaz de, 229, 256
 - de retorno, 274
 - digital de servicios integrados, 202
 - empresarial, 7
 - infraestructura, 270
 - inteligente, 568
 - largo recorrido, 306
 - multiacceso, 225
 - pública, 6
 - superpuesta, 584
 - tarjeta de interfaz de, 229
 - telefónica de comutación pública, 493
 - traducción de direcciones de, 62, 406
 - traducción de direcciones y puertos de, 409
 - troncal de multidifusión, 463
 - unidad de interfaz de, 211
- red(es)
 - administrador de la capa de interfaz de, 533
 - ancho de banda de, 200
 - aprovisionamiento de, 479
 - conectar y usar, 403
 - dirección de, 357
 - elemento de, 536
 - nodo de, 313
 - número de, 348
 - programación de, 589
 - protocolo de, 8
- renumeración de, 361
- RED, 483
- reensamblaje, 384
- reenvío, 376
 - árbol (ZigBee), 575
 - QoS, 483
 - siguiente salto, 311
 - tabla de, 311, 375
- registrador para DNS, 69
- registrador para ICANN, 351
- registro de recursos (DNS), 75
- reglas básicas de codificación, 499
- reiniciar, 540
- relleno
 - bits, 211, 233
 - bytes, 233
 - caracteres, 233
- rendición de cuentas, 512
- rendimiento (FCAPS), 535
- Reno TCP, 441
- renta, 403
- renumeración de redes, 361
- repetidor, 209, 293
- reproducción, 429, 509
- reservación, 243
- resolución de nombres, 72
- restablecer, 540
- restricción de políticas, 452
- resuelto, 392
- retorno, tecla, 56
- retransmisión, 429, 434
 - de trama, 329
- retransmisión/recuperación rápida, 441
- retransmitir, 246
- retraso, 453, 470
 - comutación, 471
 - ida y vuelta, 435
 - propagación, 129, 471
 - puesta en cola, 471
- retroceso, 247
 - exponencial, 247
 - exponencial binario, 247
- RF, 123
- RFC, 68
- RIP, 456
- RIPng, 456
- RIP pasivo, 456
- RPB, 463

- RPL, 575
RS-449, 157, 211
RSA, 515
RSVP, 484
RTS, 273
ruido, 129, 136
ruta, 315
 - específica de host, 377
 - más corta, 320
 - más corta primero, 316
 - predeterminada, 315, 448, 456
 - primaria, 321
- S**
- sa_family, 41
sa_len, 41
SACK, 441
salto, 311, 456
 - conteo, 453
 - límite, 374
satélite, 284
SBR, 481
SBS, 481
SC-F, 498
SDH, 214, 326
SDN, 332
SDP, 498, 500
SDR, 286
SDSL, 203
secuencia de chips, 193
secuencia inactiva, 159
secuencias, 428
segmento, 442
segmento de sincronización, 438
segmento FIN, 438
segmento SYN, 438
seguridad (FCAPS), 535
seguridad de HTTP, 528
seguridad de la capa de transporte, 528
select, 614
semidúplex, 160, 161,
send, 39
señal compuesta, 97
señal simple, 97
señalización, 494
servicio
 - acuerdo a nivel de, 479
 - autenticación remota de llamadas de usuarios, 528
 - datos multimegabit conmutados, 329
 - función de control, 498
 - granularidad fina, 480
 - telefónico analógico tradicional, 203
 - universal, 336
servicios
 - comunes de políticas abiertas, 484
 - diferenciados, 485
 - integrados, 481
servidor, 590
 - concurrente, 34
 - correo electrónico, 64
 - de origen, 580
 - raíz (DNS), 72
 - redirección, 497
 - registrador, 497
 - retraso, 471
 - ubicación, 496
 - virtualización, 581
servidorchat en *servidorchat.c*, 603
servidoreco en *servidoreco.c*, 598
servidorweb.c, 610
setsockopt, 44
seudoencabezado, 421
SG-F, 498
SHA-1, 513
shell seguro, 528
SIFS, 273
sigSaltoRutaIP, 541
SIGTRAN, 498
SIGUIENTE ENCABEZADO, 374
simplex, 160
sin conexión, 369, 417
sin línea de visión, 275
sin memoria, 138
sincronización, 103
síncrono(a)
 - jerarquía digital, 214, 326
 - red óptica, 214, 326
 - señal de transporte, 212
 - TDM, 188
 - transmisión, 158
SIP, 331, 494, 496, 497, 500
 - agente de usuario, 496
 - método, 500
 - proxy, 497

- URI, 500
- sistema
 - detección de intrusos, 513, 522
 - integrado, 567
 - intermedio, 460
 - nombres de dominio, 69
 - señalización, 7 494
 - terminación de módem de cable, 208
- sistema autónomo, 451
- SLA, 479
- SMDS, 329
- SMTP, 64
- SNAP, 255
- SNMP, 539
- SNMPv3, 539
- sobremuestreo, 108
- SOCK_DGRAM, 38
- SOCK_STREAM, 38
- sockaddr, 41
- sockaddr_in, 41
- socket, 36, 38
- softswitch, 495
- software
 - propagación de ruta, 448
 - radio definido por, 286
 - redes definidas por, 332
 - VPN, 525
- SOH, 232
- SOHO, 7, 209
- solicitud
 - comentarios, 68
 - repetición automática, 137
 - respuesta de eco, 400
- sondeo, 242
- SONET, 214, 215, 326
- SPC, 139
- SPF, 316
- SS7, 494
- SSH, 528
- SSL, 528
- STP, 115, 118, 296
- STS, 212
- subcanales, 204
- subcapa (IEEE), 224
- submuestreo, 108
- subred (IPv6), 362
- suite, 9
- suma de verificación, 144, 428, 513
- 16 bits, 144
- complementos a 1, 144
- supergrupo, 186
- suscriptor, 199
- T**
- T1 fraccionada, 212
- T1, T2 y T3, 204, 211, 212
- tabla de traducción, 408
- Tahoe TCP, 441
- tamaño
 - ventana, 430
 - pico de ráfaga, 481
 - sostenido de ráfaga, 481
- tasa de información comprometida, 479
- tasa pico de bits, 481
- tasa sostenida de bits, 481
- TCAM, 556
- TCP, 416, 426
- TCP/IP, 5, 341
- TDM, 187
 - estadística, 191
- TDMA, 242
- tecnologías centrales, 209
- telecomunicaciones
 - Asociación de la industria de las, 118
 - red de administración de, 534
- telefonía, 489
- televisión de antena comunitaria, 206
- temporizador de reensamblaje, 386
- teorema de Nyquist, 108
- teorema de Shannon, 129
- terminal (H.323), 497
- terminal de apertura muy pequeña, 284
- terrestre, 125
- texto cifrado, 514
- texto simple, 514
- Thicknet, 256
- Thinnet, 257
- TIA, 118
- tiempo excedido, 399
- tipo de Ethernet, 254, 396
- tipo de registro DNS 75
- tipos de registro DNS A, AAAA y MX, 75
- TLD, 69
- TLS, 528
- TMN, 534

todos ceros, dirección y sufijo, 357, 358
todos unos, dirección y sufijo, 357, 358
Token Ring (IBM), 327
topología, 227
 bus, 227
 estrella, 228
 malla, 228, 571
 semipermanente, 575
traceroute, 400
tráfico autosimilar, 478
trama, 159, 232
 filtrado, 294
 formato, 254
 tipo, 396
transceptor, 256
tránsito, 455
transmisión
 guiada, 114
 isócrona, 160, 474
 no guiada, 114
 paralelo, 154
 serie, 155
TRIP, 502
troncal, 207, 212
TTL, 399
túnel, 463, 526
turnos, 188

U

UART, 155
UDP, 416
 longitud de mensajes, 420
 puerto de destino, 420
 puerto de servicio, 420
 suma de verificación, 421
UMTS, 283
Unicode, 77
unidad de control multipunto, 497, 499
unidad de servicio de canal, 211
unidad máxima de transmisión, 381
unirse, 202
universal
 receptor y transmisor asíncrono, 155
 receptor y transmisor síncrono-asíncrono, 155
URI, 500, 502
URL, 54
USART, 155

uso, 475
uso de redes asimétricas, 199
usuario
 cliente y servidor de agente de, 496
 datagrama de, 420
UTP, 115, 118
UWB, 277

V

V.32, 176
V.35, 211
varios a 1, 419
VDSL, 203
vector de distancias, 318, 456
velocidad, 473
velocidad de datos efectiva, 472
velocidad de transferencia, 430, 453, 472
ventana, 437
 cero, 437
 deslizante, 430
video bajo demanda, 583
virtual(es)
 circuito, 220
 conexiones, 427
 paquete, 370
 red, 339
 red privada, 513
virtualización, 547
VLAN
 comutador, 299
 etiqueta de, 300
VM, 547, 581
VoIP, 493
voltaje, 157
voz sobre IP, 493
VPN, 513
VSAT, 284

W

WAN, 223
WCDMA, 283
WDM, 187
WEP, 528
Wi-Fi, 326, 528
WiMAX, 274, 283, 326
 Advanced, 283

fija, 274
foro, 274
móvil, 274
Windows Sockets, 40
Wireshark, 537
World Wide Web, 51
WPA, 528
write, 39

X

X.21, 162
xDSL, 202
XML, 78, 582
xor, 141, 148

Z

ZigBee, 266
zona muerta, 271

Esta nueva edición de *Redes de computadoras e Internet* comienza con una explicación sobre las aplicaciones de red y los paradigmas de comunicación que ofrece Internet.

El texto ofrece a los estudiantes una comprensión clara de la estructura que brinda Internet a las aplicaciones antes de estudiar las tecnologías que la implementan. Después de la explicación sobre las aplicaciones, presenta las redes de una manera lógica, de modo que usted comprenda cómo es que cada nueva tecnología se basa en las tecnologías de las capas inferiores.

Entre las novedades de esta edición se encuentran los siguientes:

- Texto totalmente actualizado con las nuevas tecnologías de redes
- Figuras que mejoran la comprensión de las explicaciones presentadas en el texto
- Integración de IPv4 e IPv6 en todos los capítulos
- Cobertura mejorada de MPLS y túneles
- Nuevo capítulo sobre redes definidas por software y OpenFlow
- Nuevo capítulo sobre Internet y ZigBee

Este libro sobresale debido a su amplia cobertura, su organización lógica, la forma como explica los conceptos y la metodología que utiliza para presentar los temas.

Para mayor información visite la página Web del libro,
donde encontrará material totalmente en español:

www.pearsonenespanol.com/comer

Visítenos en:
www.pearsonenespanol.com

