

CUARTA EDICIÓN

Redes de computadoras

ANDREW S. TANENBAUM



PEARSON
Prentice Hall

Redes de computadoras

Cuarta edición

Redes de computadoras

Cuarta edición

Andrew S. Tanenbaum

*Vrije Universiteit
Amsterdam, The Netherlands*

TRADUCCIÓN

Elisa Núñez Ramos
Traductora Profesional

REVISIÓN TÉCNICA

Felipe Antonio Trujillo Fernández
*Maestro en Sistemas, Planeación e Informática
Universidad Iberoamericana*

Adalberto Francisco García Espinosa
*Ingeniero en Sistemas Electrónicos
ITESM–CCM*



Datos de catalogación bibliográfica

TANENBAUM, ANDREW S.
Redes de computadoras

PEARSON EDUCACIÓN, México, 2003
ISBN: 970-26-0162-2
Área: Universitarios

Formato 19 × 23.5 cm Páginas: 912

Authorized translation from the English language edition, entitled *Computer Networks, Fourth Edition*, by Andrew S. Tanenbaum, published by Pearson Education, Inc., publishing as PRENTICE HALL, INC., Copyright © 2003. All rights reserved.

ISBN 0-13-066102-3

Traducción autorizada de la edición en idioma inglés, titulada *Computer Networks, Fourth Edition*, por Andrew S. Tanenbaum, publicada por Pearson Education, Inc., publicada como PRENTICE-HALL INC., Copyright © 2003. Todos los derechos reservados.

Esta edición en español es la única autorizada.

Edición en español

Editor: Guillermo Trujano Mendoza
e-mail: guillermo.trujano@pearsoned.com
Editor de desarrollo: Miguel Gutiérrez Hernández
Supervisor de producción: José D. Hernández Garduño

Edición en inglés

Editorial/production supervision: *Patti Guerrieri*
Cover design director: *Jerry Votta*
Cover designer: *Anthony Gemmellaro*
Cover design: *Andrew S. Tanenbaum*
Art director: *Gail Cocker-Bogusz*
Interior Design: *Andrew S. Tanenbaum*
Interior graphics: *Hadel Studio*
Typesetting: *Andrew S. Tanenbaum*
Manufacturing buyer: *Maura Zaldivar*
Executive editor: *Mary Franz*
Editorial assistant: *Noreen Regina*
Marketing manager: *Dan DePasquale*

CUARTA EDICIÓN, 2003

D.R. © 2003 por Pearson Educación de México, S.A. de C.V.
Atlacomulco 500-5to. piso
Industrial Atoto
53519 Naucalpan de Juárez, Edo. de México
E-mail: editorial.universidades@pearsoned.com

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

Prentice Hall es una marca registrada de Pearson Educación de México, S.A. de C.V.

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 970-26-0162-2

Impreso en México. *Printed in Mexico.*

1 2 3 4 5 6 7 8 9 0 - 06 05 04 03

Para Suzanne, Barbara, Marvin y en recuerdo de Bram y Sweetie π

CONTENIDO

PREFACIO

1	INTRODUCCIÓN	1
1.1	USOS DE LAS REDES DE COMPUTADORAS	3
1.1.1	Aplicaciones de negocios	3
1.1.2	Aplicaciones domésticas	6
1.1.3	Usuarios móviles	9
1.1.4	Temas sociales	12
1.2	HARDWARE DE REDES	14
1.2.1	Redes de área local	16
1.2.2	Redes de área metropolitana	18
1.2.3	Redes de área amplia	19
1.2.4	Redes inalámbricas	21
1.2.5	Redes domésticas	23
1.2.6	Interredes	25
1.3	SOFTWARE DE REDES	26
1.3.1	Jerarquías de protocolos	26
1.3.2	Aspectos de diseño de las capas	30
1.3.3	Servicios orientados a la conexión y no orientados a la conexión	32
1.3.4	Primitivas de servicio	34
1.3.5	Relación de servicios a protocolos	36

1.4 MODELOS DE REFERENCIA	37
1.4.1 El modelo de referencia OSI	37
1.4.2 El modelo de referencia TCP/IP	41
1.4.3 Comparación entre los modelos de referencia OSI y TCP/IP	44
1.4.4 Crítica al modelo OSI y los protocolos	46
1.4.5 Crítica del modelo de referencia TCP/IP	48
1.5 REDES DE EJEMPLO	49
1.5.1 Internet	50
1.5.2 Redes orientadas a la conexión: X.25, Frame Relay y ATM	59
1.5.3 Ethernet	65
1.5.4 LANs inalámbricas: 802.11	68
1.6 ESTANDARIZACIÓN DE REDES	71
1.6.1 Quién es quién en el mundo de las telecomunicaciones	71
1.6.2 Quién es quién en los estándares internacionales	74
1.6.3 Quién es quién en el mundo de los estándares de Internet	75
1.7 UNIDADES MÉTRICAS	77
1.8 PANORAMA DEL RESTO DEL LIBRO	78
1.9 RESUMEN	80

2 LA CAPA FÍSICA 85

2.1 LA BASE TEÓRICA DE LA COMUNICACIÓN DE DATOS	85
2.1.1 El análisis de Fourier	86
2.1.2 Señales de ancho de banda limitado	86
2.1.3 La tasa de datos máxima de un canal	89
2.2 MEDIOS DE TRANSMISIÓN GUIADOS	90
2.2.1 Medios magnéticos	90
2.2.2 Par trenzado	91
2.2.3 Cable coaxial	92
2.2.4 Fibra óptica	93
2.3 TRANSMISIÓN INALÁMBRICA	100
2.3.1 El espectro electromagnético	100
2.3.2 Radiotransmisión	103

2.3.3 Transmisión por microondas	104
2.3.4 Ondas infrarrojas y milimétricas	106
2.3.5 Transmisión por ondas de luz	107
2.4 SATÉLITES DE COMUNICACIONES	109
2.4.1 Satélites geoestacionarios	109
2.4.2 Satélites de Órbita Terrestre Media	113
2.4.3 Satélites de Órbita Terrestre Baja	114
2.4.4 Satélites en comparación con fibra óptica	117
2.5 LA RED TELEFÓNICA PÚBLICA CONMUTADA	118
2.5.1 Estructura del sistema telefónico	119
2.5.2 La política de los teléfonos	122
2.5.3 El circuito local: módems, ADSL e inalámbrico	124
2.5.4 Troncales y multiplexión	137
2.5.5 Comutación	146
2.6 EL SISTEMA TELEFÓNICO MÓVIL	152
2.6.1 Teléfonos móviles de primera generación	153
2.6.2 Teléfonos móviles de segunda generación: voz digital	157
2.6.3 Teléfonos móviles de tercera generación: voz y datos digitales	166
2.7 TELEVISIÓN POR CABLE	169
2.7.1 Televisión por antena comunal	169
2.7.2 Internet a través de cable	170
2.7.3 Asignación de espectro	172
2.7.4 Módems de cable	173
2.7.5 ADSL en comparación con el cable	175
2.8 RESUMEN	177

3 LA CAPA DE ENLACE DE DATOS 183

3.1 CUESTIONES DE DISEÑO DE LA CAPA DE ENLACE DE DATOS	184
3.1.1 Servicios proporcionados a la capa de red	184
3.1.2 Entramado	187
3.1.3 Control de errores	191
3.1.4 Control de flujo	192

3.2 DETECCIÓN Y CORRECCIÓN DE ERRORES	192
3.2.1 Códigos de corrección de errores	193
3.2.2 Códigos de detección de errores	196
3.3 PROTOCOLOS ELEMENTALES DE ENLACE DE DATOS	200
3.3.1 Un protocolo simplex sin restricciones	204
3.3.2 Protocolo simplex de parada y espera	206
3.3.3 Protocolo simplex para un canal con ruido	208
3.4 PROTOCOLOS DE VENTANA CORREDIZA	211
3.4.1 Un protocolo de ventana corrediza de un bit	214
3.4.2 Protocolo que usa retroceso N	216
3.4.3 Protocolo que utiliza repetición selectiva	223
3.5 VERIFICACIÓN DE LOS PROTOCOLOS	229
3.5.1 Modelos de máquinas de estado finito	229
3.5.2 Modelos de red de Petri	232
3.6 EJEMPLOS DE PROTOCOLOS DE ENLACE DE DATOS	234
3.6.1 HDLC—Control de Enlace de Datos de Alto Nivel	234
3.6.2 La capa de enlace de datos en Internet	237
3.7 RESUMEN	242

4 LA SUBCAPA DE CONTROL DE ACCESO AL MEDIO 247

4.1 EL PROBLEMA DE ASIGNACIÓN DEL CANAL	248
4.1.1 Asignación estática de canal en LANs y MANs	248
4.1.2 Asignación dinámica de canales en LANs y MANs	249
4.2 PROTOCOLOS DE ACCESO MÚLTIPLE	251
4.2.1 ALOHA	251
4.2.2 Protocolos de acceso múltiple con detección de portadora	255
4.2.3 Protocolos libres de colisiones	259
4.2.4 Protocolos de contención limitada	261
4.2.5 Protocolos de acceso múltiple por división de longitud de onda	265
4.2.6 Protocolos de LANs inalámbricas	267

4.3 ETHERNET	271
4.3.1 Cableado Ethernet	271
4.3.2 Codificación Manchester	274
4.3.3 El protocolo de subcapa MAC de Ethernet	275
4.3.4 Algoritmo de retroceso exponencial binario	278
4.3.5 Desempeño de Ethernet	279
4.3.6 Ethernet conmutada	281
4.3.7 Fast Ethernet	283
4.3.8 Gigabit Ethernet	286
4.3.9 Estándar IEEE 802.2: control lógico del enlace	290
4.3.10 Retrospectiva de Ethernet	291
4.4 LANS INALÁMBRICAS	292
4.4.1 La pila de protocolos del 802.11	292
4.4.2 La capa física del 802.11	293
4.4.3 El protocolo de la subcapa MAC del 802.11	295
4.4.4 La estructura de trama 802.11	299
4.4.5 Servicios	301
4.5 BANDA ANCHA INALÁMBRICA	302
4.5.1 Comparación entre los estándares 802.11 y 802.16	303
4.5.2 La pila de protocolos del estándar 802.16	305
4.5.3 La capa física del estándar 802.16	306
4.5.4 El protocolo de la subcapa MAC del 802.16	307
4.5.5 La estructura de trama 802.16	309
4.6 BLUETOOTH	310
4.6.1 Arquitectura de Bluetooth	311
4.6.2 Aplicaciones de Bluetooth	312
4.6.3 La pila de protocolos de Bluetooth	313
4.6.4 La capa de radio de Bluetooth	314
4.6.5 La capa de banda base de Bluetooth	315
4.6.6 La capa L2CAP de Bluetooth	316
4.6.7 Estructura de la trama de Bluetooth	316
4.7 CONMUTACIÓN EN LA CAPA DE ENLACE DE DATOS	317
4.7.1 Puentes de 802.x a 802.y	319
4.7.2 Interconectividad local	322
4.7.3 Puentes con árbol de expansión	323
4.7.4 Puentes remotos	325
4.7.5 Repetidores, concentradores, puentes, conmutadores, enrutadores y puertas de enlace	326
4.7.6 LANs virtuales	328
4.8 RESUMEN	336

5 LA CAPA DE RED 343

5.1 ASPECTOS DE DISEÑO DE LA CAPA DE RED	343
5.1.1 Comutación de paquetes de almacenamiento y reenvío	344
5.1.2 Servicios proporcionados a la capa de transporte	344
5.1.3 Implementación del servicio no orientado a la conexión	345
5.1.4 Implementación del servicio orientado a la conexión	347
5.1.5 Comparación entre las subredes de circuitos virtuales y las de datagramas	348
5.2 ALGORITMOS DE ENRUTAMIENTO	350
5.2.1 Principio de optimización	352
5.2.2 Enrutamiento por la ruta más corta	353
5.2.3 Inundación	355
5.2.4 Enrutamiento por vector de distancia	357
5.2.5 Enrutamiento por estado del enlace	360
5.2.6 Enrutamiento jerárquico	366
5.2.7 Enrutamiento por difusión	368
5.2.8 Enrutamiento por multidifusión	370
5.2.9 Enrutamiento para <i>hosts</i> móviles	372
5.2.10 Enrutamiento en redes <i>ad hoc</i>	375
5.2.11 Búsqueda de nodos en redes de igual a igual	380
5.3 ALGORITMOS DE CONTROL DE CONGESTIÓN	384
5.3.1 Principios generales del control de congestión	386
5.3.2 Políticas de prevención de congestión	388
5.3.3 Control de congestión en subredes de circuitos virtuales	389
5.3.4 Control de congestión en subredes de datagramas	391
5.3.5 Desprendimiento de carga	394
5.3.6 Control de fluctuación	395
5.4 CALIDAD DEL SERVICIO	397
5.4.1 Requerimientos	397
5.4.2 Técnicas para alcanzar buena calidad de servicio	398
5.4.3 Servicios integrados	409
5.4.4 Servicios diferenciados	412
5.4.5 Comutación de etiquetas y MPLS	415
5.5 INTERCONECTIVIDAD	418
5.5.1 Cómo difieren las redes	419
5.5.2 Conexión de redes	420
5.5.3 Circuitos virtuales concatenados	422
5.5.4 Interconectividad no orientada a la conexión	423

5.5.5 Entunelamiento	425
5.5.6 Enrutamiento entre redes	426
5.5.7 Fragmentación	427
5.6 LA CAPA DE RED DE INTERNET	431
5.6.1 El protocolo IP	433
5.6.2 Direcciones IP	436
5.6.3 Protocolos de Control en Internet	449
5.6.4 OSPF—Protocolos de Enrutamiento de Puerta de Enlace Interior	454
5.6.5 BGP—Protocolo de Puerta de Enlace de Frontera	459
5.6.6 Multidifusión de Internet	461
5.6.7 IP móvil	462
5.6.8 IPv6	464
5.7 RESUMEN	473

6 LA CAPA DE TRANSPORTE **481**

6.1 EL SERVICIO DE TRANSPORTE	481
6.1.1 Servicios proporcionados a las capas superiores	481
6.1.2 Primitivas del servicio de transporte	483
6.1.3 <i>Sockets</i> de Berkeley	487
6.1.4 Un ejemplo de programación de <i>sockets</i> : un servidor de archivos de Internet	488
6.2 ELEMENTOS DE LOS PROTOCOLOS DE TRANSPORTE	492
6.2.1 Direccionamiento	493
6.2.2 Establecimiento de una conexión	496
6.2.3 Liberación de una conexión	502
6.2.4 Control de flujo y almacenamiento en búfer	506
6.2.5 Multiplexión	510
6.2.6 Recuperación de caídas	511
6.3 UN PROTOCOLO DE TRANSPORTE SENCILLO	513
6.3.1 Las primitivas de servicio de ejemplo	513
6.3.2 La entidad de transporte de ejemplo	515
6.3.3 El ejemplo como máquina de estados finitos	522
6.4 LOS PROTOCOLOS DE TRANSPORTE DE INTERNET: UDP	524
6.4.1 Introducción a UDP	525
6.4.2 Llamada a procedimiento remoto	526
6.4.3 El protocolo de transporte en tiempo real	529

6.5 LOS PROTOCOLOS DE TRANSPORTE DE INTERNET: TCP	532
6.5.1 Introducción a TCP	532
6.5.2 El modelo del servicio TCP	533
6.5.3 El protocolo TCP	535
6.5.4 El encabezado del segmento TCP	536
6.5.5 Establecimiento de una conexión TCP	539
6.5.6 Liberación de una conexión TCP	541
6.5.7 Modelado de administración de conexiones TCP	541
6.5.8 Política de transmisión del TCP	543
6.5.9 Control de congestión en TCP	547
6.5.10 Administración de temporizadores del TCP	550
6.5.11 TCP y UDP inalámbricos	553
6.5.12 TCP para Transacciones	555
6.6 ASPECTOS DEL DESEMPEÑO	557
6.6.1 Problemas de desempeño en las redes de cómputo	557
6.6.2 Medición del desempeño de las redes	560
6.6.3 Diseño de sistemas para un mejor desempeño	562
6.6.4 Procesamiento rápido de las TPDUs	566
6.6.5 Protocolos para redes de gigabits	569
6.7 RESUMEN	573

7 LA CAPA DE APLICACIÓN 579

7.1 DNS—EL SISTEMA DE NOMBRES DE DOMINIO	579
7.1.1 El espacio de nombres del DNS	580
7.1.2 Registros de recursos	582
7.1.3 Servidores de nombres	586
7.2 CORREO ELECTRÓNICO	588
7.2.1 Arquitectura y servicios	590
7.2.2 El agente de usuario	591
7.2.3 Formatos de mensaje	594
7.2.4 Transferencia de mensajes	602
7.2.5 Entrega final	605
7.3 WORLD WIDE WEB	611
7.3.1 Panorama de la arquitectura	612
7.3.2 Documentos Web estáticos	629

7.3.3 Documentos Web dinámicos	643
7.3.4 HTTP—Protocolo de Transferencia de Hipertexto	651
7.3.5 Mejoras de desempeño	656
7.3.6 La Web inalámbrica	662
7.4 MULTIMEDIA	674
7.4.1 Introducción al audio digital	674
7.4.2 Compresión de audio	676
7.4.3 Audio de flujo continuo	679
7.4.4 Radio en Internet	683
7.4.5 Voz sobre IP	685
7.4.6 Introducción al vídeo	692
7.4.7 Compresión de vídeo	696
7.4.8 Vídeo bajo demanda	704
7.4.9 Mbone—Red dorsal de multidifusión	711
7.5 RESUMEN	714

8 SEGURIDAD EN REDES **721**

8.1 CRIPTOGRAFÍA	724
8.1.1 Introducción a la criptografía	725
8.1.2 Cifrados por sustitución	727
8.1.3 Cifrados por transposición	729
8.1.4 Rellenos de una sola vez	730
8.1.5 Dos principios criptográficos fundamentales	735
8.2 ALGORITMOS DE CLAVE SIMÉTRICA	737
8.2.1 DES—El Estándar de Encriptación de Datos	738
8.2.2 AES—El Estándar de Encriptación Avanzada	741
8.2.3 Modos de cifrado	745
8.2.4 Otros cífrados	750
8.2.5 Criptoanálisis	750
8.3 ALGORITMOS DE CLAVE PÚBLICA	752
8.3.1 El algoritmo RSA	753
8.3.2 Otros algoritmos de clave pública	755

8.4 FIRMAS DIGITALES	755
8.4.1 Firmas de clave simétrica	756
8.4.2 Firmas de clave pública	757
8.4.3 Compendios de mensaje	759
8.4.4 El ataque de cumpleaños	763
8.5 ADMINISTRACIÓN DE CLAVES PÚBLICAS	765
8.5.1 Certificados	765
8.5.2 X.509	767
8.5.3 Infraestructuras de clave pública	768
8.6 SEGURIDAD EN LA COMUNICACIÓN	772
8.6.1 Ipsec	772
8.6.2 Firewalls	776
8.6.3 Redes privadas virtuales	779
8.6.4 Seguridad inalámbrica	780
8.7 PROTOCOLOS DE AUTENTICACIÓN	785
8.7.1 Autenticación basada en una clave secreta compartida	786
8.7.2 Establecimiento de una clave compartida: el intercambio de claves de Diffie-Hellman	791
8.7.3 Autenticación que utiliza un centro de distribución de claves	793
8.7.4 Autenticación utilizando Kerberos	796
8.7.5 Autenticación utilizando criptografía de clave pública	798
8.8 SEGURIDAD DE CORREO ELECTRÓNICO	799
8.8.1 PGP—Privacidad Bastante Buena	799
8.8.2 PEM—Correo con Privacidad Mejorada	803
8.8.3 S/MIME	804
8.9 SEGURIDAD EN WEB	805
8.9.1 Amenazas	805
8.9.2 Asignación segura de nombres	806
8.9.3 SSL—La Capa de Sockets Seguros	813
8.9.4 Seguridad de código móvil	816
8.10 ASPECTOS SOCIALES	819
8.10.1 Privacidad	819
8.10.2 Libertad de expresión	822
8.10.3 Derechos de autor	826
8.11 RESUMEN	828

9 LISTA DE LECTURAS Y BIBLIOGRAFÍA 835

- 9.1. SUGERENCIAS DE LECTURAS ADICIONALES 835
 - 9.1.1 Introducción y obras generales 836
 - 9.1.2 La capa física 838
 - 9.1.3 La capa de enlace de datos 840
 - 9.1.4 La subcapa de control de acceso al medio 840
 - 9.1.5 La capa de red 842
 - 9.1.6 La capa de transporte 844
 - 9.1.7 La capa de aplicación 844
 - 9.1.8 Seguridad en redes 846
- 9.2 BIBLIOGRAFÍA EN ORDEN ALFABÉTICO 848

ÍNDICE 869

PREFACIO

La presente es la cuarta edición de este libro. Cada edición ha correspondido a una fase diferente de la manera en que se usaron las redes de computadoras. Cuando apareció la primera edición, en 1980, las redes eran una curiosidad académica. Para la segunda edición, en 1988, las redes ya se usaban en universidades y en grandes empresas. Y en 1996, cuando se editó por tercera vez este libro, las redes de computadoras, en particular Internet, se habían convertido en una realidad cotidiana para millones de personas. El elemento nuevo de la cuarta edición es el rápido crecimiento de las redes inalámbricas en muchas formas.

El panorama de las redes ha cambiado radicalmente desde la tercera edición. A mediados de la década de 1990 existían varios tipos de LANs y WANs, junto con pilas de múltiples protocolos. Para el 2003, la única LAN alámbrica de amplio uso tal vez sea Ethernet y prácticamente todas las WANs estarían en Internet. En consecuencia, se habrá eliminado una gran cantidad de material referente a estas antiguas redes.

Sin embargo, también abundan los nuevos desarrollos. Lo más importante es el gran aumento de redes inalámbricas, como la 802.11, los ciclos locales inalámbricos, las redes celulares 2G y 3G, Bluetooth, WAP (protocolo de aplicaciones inalámbricas), el i-mode y otros. De acuerdo con esto, se ha agregado una gran cantidad de material a las redes inalámbricas. Otro tema importante y novedoso es la seguridad, por lo que se ha agregado todo un capítulo al respecto.

Aun cuando el capítulo 1 tiene la misma función introductoria que en la tercera edición, su contenido se ha revisado y actualizado. Por ejemplo, en dicho capítulo se presentan introducciones a Internet, Ethernet y LANs inalámbricas, además de algunas referencias y datos históricos. También se explican brevemente las redes domésticas.

El capítulo 2 se ha reorganizado. Luego de una breve introducción a los principios de comunicación de datos, hay tres secciones importantes sobre la transmisión (medios guiados, inalámbricos y por satélite), seguidas de otras tres con ejemplos importantes (el sistema público de telefonía conmutada, el sistema de teléfonos celulares y la TV por cable). Entre los nuevos temas tratados en este capítulo están ADSL, banda ancha inalámbrica, MANs inalámbricas y acceso a Internet por cable y DOCSIS.

El capítulo 3 siempre ha presentado los principios fundamentales de los protocolos de punto a punto. Estas ideas son permanentes y no han cambiado durante décadas.

En consecuencia, las series de protocolos de ejemplo detallados que se presentan en este capítulo no han cambiado en lo más mínimo desde la tercera edición.

En contraste, la subcapa MAC ha sido un área de gran actividad en los últimos años, por lo que se presentan muchos cambios en el capítulo 4. La sección sobre Ethernet se ha ampliado para incluir la Ethernet de gigabits. Las secciones nuevas importantes son las que tratan sobre las LANs inalámbricas, banda ancha inalámbrica, Bluetooth y la conmutación de la capa de enlace de datos, incluyendo MPLS.

También se actualizó el capítulo 5, en donde se eliminó todo lo referente a ATM y se incluyó material adicional sobre Internet.

Ahora la calidad del servicio también es un tema importante, incluyendo las exposiciones de los servicios integrados y los servicios diferenciados. Las redes inalámbricas también están presentes aquí, con una explicación del enrutamiento en redes *ad hoc*. Entre otros temas nuevos se encuentran las redes NAT y de igual a igual.

El capítulo 6 trata aún de la capa de transporte, pero aquí también se han hecho algunos cambios, que incluyen un ejemplo de la programación de *sockets*. También se explican un cliente y un servidor de una página en C. Estos programas, disponibles en el sitio Web del libro, se pueden compilar y ejecutar. En conjunto proporcionan un servidor Web de archivos remoto para experimentación. Entre otros temas están la llamada a procedimiento remoto, RTP y el TCP para transacciones.

El capítulo 7 se ha enfocado sobre todo en la capa de aplicación. Después de una breve introducción sobre DNS, el resto del capítulo aborda sólo tres temas: el correo electrónico, Web y multimedia. Pero cada tema se trata con todo detalle. La exposición de cómo funciona Web abarca ahora más de 60 páginas, y cubre una amplia serie de temas, entre ellos las páginas Web estáticas y dinámicas, HTTP, los scripts (secuencias de comandos) de CGI, redes de distribución de información, *cookies* y el uso de caché en Web. También se incluye material sobre cómo se escriben las páginas Web modernas, incluyendo breves introducciones a XML, XSL, XHTML, PHP y más; todo con ejemplos que se pueden probar. Asimismo, hay una exposición sobre Web inalámbrica, enfocándose en i-mode y WAP. El material de multimedia incluye ahora MP3, audio de flujo continuo, radio en Internet y voz sobre IP.

La seguridad ha llegado a ser tan importante que ahora se ha ampliado a un capítulo entero de más de 100 páginas (el capítulo 8). Cubre tanto los principios de la seguridad (algoritmos simétricos y de clave pública, firmas digitales y certificados X.509) como las aplicaciones de estos principios (autenticación, seguridad del correo electrónico y seguridad en Web). El capítulo es amplio (va desde la criptografía cuántica hasta la censura gubernamental) y profundo (por ejemplo, trata en detalle el funcionamiento de SHA-1).

El capítulo 9 contiene una lista totalmente nueva de lecturas sugeridas y una amplia bibliografía de más de 350 citas a la literatura actual. Más de 200 de éstas son a artículos y libros escritos en el 2000 o más recientes.

Los libros de computación están llenos de acrónimos, y éste no es la excepción. Para cuando acabe de leer el presente libro, los siguientes términos le serán familiares: ADSL, AES, AMPS, AODV, ARP, ATM, BGP, CDMA, CDN, CGI, CIDR, DCF, DES, DHCP, DMCA, FDM, FHSS, GPRS, GSM, HDLC, HFC, HTML, HTTP, ICMP, IMAP, ISP, ITU, LAN, LMDS, MAC, MACA, MIME, MPEG, MPLS, MTU, NAP, NAT, NSA, NTSC, OFDM, OSPF, PCF, PCM, PGP, PHP, PKI,

POTS, PPP, PSTN, QAM, QPSK, RED, RFC, RPC, RSA, RSVP, RTP, SSL, TCP, TDM, UDP, URL, UTP, VLAN, VPN, VSAT, WAN, WAP, WDMA, WEP, WWW y XML. Pero no se preocupe. Cada uno se definirá cuidadosamente antes de usarlo.

Para ayudar a los profesores a utilizar este libro como texto en un curso, el autor ha preparado varios apoyos, en inglés, para la enseñanza, como:

- Un manual de solución de problemas.
- Archivos que contienen las figuras del libro en varios formatos.
- Páginas de PowerPoint para un curso que utilice el libro.
- Un simulador (escrito en C) para los protocolos de ejemplo del capítulo 3.
- Una página Web con vínculos a muchos manuales en línea, empresas, listas de preguntas frecuentes, etcétera.

El manual de soluciones sólo podrá adquirirlo directamente con los representantes de Pearson Educación (pero **sólo** está disponible para los profesores; los estudiantes no podrán adquirirlo). El resto del material está en el sitio Web del libro:

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

Localice la portada del libro y haga clic en ella.

Muchas personas me ayudaron durante la preparación de la cuarta edición. Me gustaría agradecer especialmente a: Ross Anderson, Elizabeth Belding Royer, Steve Bellovin, Chatschick Bisdikian, Kees Bot, Scott Bradner, Jennifer Bray, Pat Cain, Ed Felten, Warwick Ford, Kevin Fu, Ron Fulle, Jim Geier, Mario Gerla, Natalie Giroux, Steve Hanna, Jeff Hayes, Amir Herzberg, Philip Homburg, Philipp Hoschka, David Green, Bart Jacobs, Frans Kaashoek, Steve Kent, Roger Kermode, Robert Kinicki, Shay Kutten, Rob Lanphier, Marcus Leech, Tom Maufer, Brent Miller, Shivakant Mishra, Thomas Nadeau, Shlomo Ovadia, Kaveh Pahlavan, Radia Perlman, Guillaume Pierre, Wayne Pleasant, Patrick Powell, Tomas Robertazzi, Medy Sanadidi, Christian Schmutzler, Henning Schulzrinne, Paul Sevinc, Mihail Sichitiu, Bernard Sklar, Ed Skoudis, Bob Strader, George Swallow, George Thiruvathukal, Peter Tomsu, Patrick Verkaik, Dave Vittali, Spyros Voulgaris, Jan-Mark Wams, Ruediger Weis, Bert Wijnen, Joseph Wilkes, Leendert van Doorn y Maarten van Steen.

Un especial agradecimiento a Trudy Levine por demostrar que las abuelas pueden hacer un trabajo fino de revisión de material técnico. Shivakant Mishra ideó muchos de los desafiantes problemas de fin de capítulo. Andy Dornan sugirió lecturas adicionales para el capítulo 9. Jan Looijen proporcionó hardware esencial en un momento crítico. El doctor F. de Nies hizo un excelente trabajo de cortado y pegado justo cuando fue necesario. Mary Franz, mi editora en Prentice Hall, me proporcionó más material de lectura del que había consumido en los 7 años anteriores y fue de gran ayuda en muchos otros aspectos.

Finalmente, llegamos a las personas más importantes: Suzanne, Barbara y Marvin. A Suzanne por su amor, paciencia y los almuerzos. A Barbara y Marvin por ser agradables y joviales (excepto al quejarse de los horribles libros de texto universitarios, manteniéndome, de este modo, alerta). Gracias.

ANDREW S. TANENBAUM

1

INTRODUCCIÓN

Cada uno de los tres últimos siglos fue dominado por una tecnología. El siglo XVIII fue la era de los grandes sistemas mecánicos que acompañaron la Revolución Industrial. El siglo XIX fue la edad de la máquina de vapor. Durante el siglo XX la tecnología clave fue la obtención, el procesamiento y la distribución de la información. Entre otros acontecimientos, vimos la instalación de redes mundiales de telefonía, la invención de la radio y la televisión, el nacimiento y crecimiento sin precedentes de la industria de la computación, así como el lanzamiento de satélites de comunicaciones.

Como resultado del rápido progreso tecnológico, estas áreas están convergiendo de una manera acelerada y las diferencias entre la recolección, transportación, almacenamiento y procesamiento de la información están desapareciendo rápidamente. Organizaciones con cientos de oficinas dispersas en una amplia área geográfica esperan de manera rutinaria poder examinar el estado actual incluso de su sucursal más distante con sólo oprimir un botón. Al aumentar nuestra capacidad de obtener, procesar y distribuir información, la demanda de procesamiento de información cada vez más complejo crece incluso con más celeridad.

Aunque la industria de la computación aún es joven en comparación con otras industrias (como la automotriz y la aeronáutica), ha progresado espectacularmente en poco tiempo. Durante las dos primeras décadas de su existencia, los sistemas de computación estaban altamente centralizados, por lo general, en una sala grande e independiente. Con frecuencia, estas salas tenían paredes de cristal a través de las cuales los visitantes podían atisbar la maravilla electrónica que encerraban. Las compañías o universidades medianas apenas llegaban a tener una o dos computadoras, en tanto que las

instituciones grandes tenían, cuando mucho, una docena. La idea de que en veinte años se pudieran producir en masa millones de computadoras igualmente poderosas pero más pequeñas que un timbre postal era ciencia-ficción.

La fusión de las computadoras y las comunicaciones ha tenido una influencia profunda en la manera en que están organizados los sistemas computacionales. Actualmente, el concepto de “centro de cómputo” como un espacio amplio con una computadora grande a la que los usuarios llevaban su trabajo a procesar es totalmente obsoleto. El modelo antiguo de una sola computadora que realiza todas las tareas computacionales de una empresa ha sido reemplazado por otro en el que un gran número de computadoras separadas pero interconectadas hacen el trabajo. Estos sistemas se denominan **redes de computadoras**. El diseño y la organización de estas redes es el objetivo de este libro.

A lo largo del libro utilizaremos el término “redes de computadoras” para designar un conjunto de computadoras autónomas interconectadas. Se dice que dos computadoras están interconectadas si pueden intercambiar información. No es necesario que la conexión se realice mediante un cable de cobre; también se pueden utilizar las fibras ópticas, las microondas, los rayos infrarrojos y los satélites de comunicaciones. Las redes tienen varios tamaños, formas y figuras, como veremos más adelante. Aunque a algunas personas les parezca extraño, ni Internet ni Web son una red de computadoras. Este concepto quedará claro al finalizar el libro. La respuesta rápida es: Internet no es una red única, sino una red de redes, y Web es un sistema distribuido que se ejecuta sobre Internet.

Existe una gran confusión entre una red de computadoras y un **sistema distribuido**. La diferencia principal radica en que, en un sistema distribuido, un conjunto de computadoras independientes aparece ante sus usuarios como un sistema consistente y único. Por lo general, tiene un modelo o paradigma único que se presenta a los usuarios. Con frecuencia, una capa de software que se ejecuta sobre el sistema operativo, denominada **middleware**, es la responsable de implementar este modelo. Un ejemplo bien conocido de un sistema distribuido es **World Wide Web**, en la cual todo se ve como un documento (una página Web).

En una red de computadoras no existe esta consistencia, modelo ni software. Los usuarios están expuestos a las máquinas reales, y el sistema no hace ningún intento porque las máquinas se vean y actúen de manera similar. Si las máquinas tienen hardware diferente y distintos sistemas operativos, eso es completamente transparente para los usuarios. Si un usuario desea ejecutar un programa de una máquina remota, debe registrarse en ella y ejecutarlo desde ahí.

De hecho, un sistema distribuido es un sistema de software construido sobre una red. El software le da un alto grado de consistencia y transparencia. De este modo, la diferencia entre una red y un sistema distribuido está en el software (sobre todo en el sistema operativo), más que en el hardware.

No obstante, tienen muchas cosas en común. Por ejemplo, tanto los sistemas distribuidos como las redes de computadoras necesitan mover archivos. La diferencia está en quién invoca el movimiento, el sistema o el usuario. Aunque el objetivo principal de este libro son las redes, muchos de los temas se relacionan con los sistemas distribuidos. Para más información acerca de los sistemas distribuidos, vea (Tanenbaum y Van Steen, 2002).

1.1 USOS DE LAS REDES DE COMPUTADORAS

Antes de empezar a examinar con detalle los elementos técnicos, vale la pena dedicar algo de tiempo a precisar por qué la gente se interesa en las redes de computadoras y para qué se pueden utilizar. Después de todo, si nadie se hubiera interesado en ellas, no se habrían construido tantas. Empezaremos con el uso tradicional que les dan las empresas y los individuos, y luego avanzaremos a los últimos desarrollos con respecto a los usuarios móviles y la conexión de redes domésticas.

1.1.1 Aplicaciones de negocios

Muchas compañías tienen una cantidad considerable de computadoras. Por ejemplo, una compañía podría tener computadoras separadas para supervisar la producción, controlar inventarios y hacer la nómina. Al principio estas computadoras tal vez hayan trabajado por separado pero, en algún momento, la administración decidió conectarlas para extraer y correlacionar información acerca de toda la compañía.

Dicho de una manera más general, el asunto aquí es la **compartición de recursos** y el objetivo es hacer que todos los programas, el equipo y, en particular, los datos estén disponibles para todos los que se conecten a la red, independientemente de la ubicación física del recurso y del usuario. Un ejemplo claro y muy difundido es el de un grupo de oficinistas que comparten una impresora. Ninguno de los individuos necesita una impresora privada, y una impresora de alto volumen en red suele ser más barata, rápida y fácil de mantener que varias impresoras individuales.

Sin embargo, compartir información es tal vez más importante que compartir recursos físicos, como impresoras, escáneres y quemadores de CDs. Para las compañías grandes y medianas, así como para muchas pequeñas, la información computarizada es vital. La mayoría de las compañías tiene en línea registros de clientes, inventarios, cuentas por cobrar, estados financieros, información de impuestos, etcétera. Si todas las computadoras de un banco se cayeran, éste no duraría más de cinco minutos. Una moderna planta manufacturera, con una línea de ensamblado controlada por computadora, ni siquiera duraría ese tiempo. Incluso una pequeña agencia de viajes o un despacho jurídico de tres personas, ahora dependen en gran medida de las redes de computadoras para que sus empleados puedan tener acceso de manera instantánea a la información y a los documentos importantes.

En las compañías más pequeñas, es posible que todas las computadoras estén en una sola oficina o en un solo edificio, pero en las más grandes, las computadoras y los empleados pueden estar dispersos en docenas de oficinas y plantas en varios países. No obstante, un vendedor en Nueva York podría requerir algunas veces tener acceso a una base de datos de inventario de productos que se encuentra en Singapur. En otras palabras, el hecho de que un usuario esté a 15,000 km de sus datos no debe ser impedimento para que utilice esos datos como si fueran locales. Esta meta se podría resumir diciendo que es un intento por acabar con la “tiranía de la geografía”.

En términos aún más sencillos, es posible imaginar el sistema de información de una compañía como si consistiera en una o más bases de datos y algunos empleados que necesitan acceder a

ellas de manera remota. En este modelo, los datos están almacenados en computadoras poderosas que se llaman **servidores**. Con frecuencia, éstos se encuentran alojados en una central y un administrador de sistemas les da mantenimiento. En contraste, los empleados tienen en sus escritorios máquinas más sencillas, llamadas **clientes**, con las que pueden acceder a datos remotos —por ejemplo, para incluirlos en las hojas de cálculo que están elaborando. (Algunas veces nos referiremos a los usuarios de las máquinas como “el cliente”, pero debe quedar claro, por el contexto, si el término se refiere a la computadora o a su usuario.) Las máquinas cliente y servidor están conectadas por una red, como se ilustra en la figura 1-1. Observe que hemos representado a la red como un óvalo sencillo, sin detalle alguno. Utilizaremos esta forma cuando nos refiramos a una red en sentido general. Cuando se requieran más detalles, los proporcionaremos.

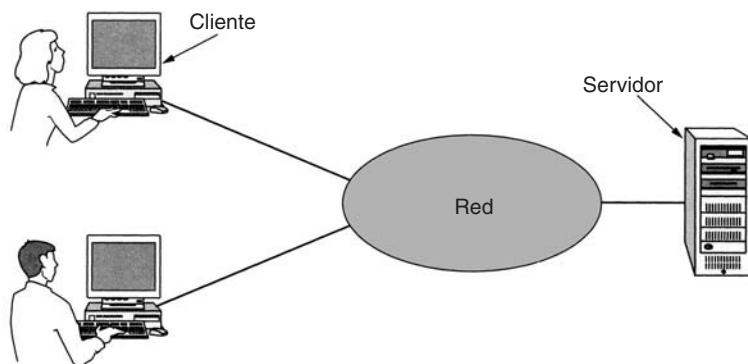


Figura 1-1. Una red con dos clientes y un servidor.

Este conjunto se conoce como **modelo cliente-servidor**. Se utiliza ampliamente y forma la base en gran medida del uso de redes. Es aplicable cuando el cliente y el servidor están en el mismo edificio (por ejemplo, cuando pertenecen a la misma compañía), pero también cuando están bastante retirados. Por ejemplo, cuando una persona en casa accede a una página Web, se emplea el mismo modelo, en el que el servidor remoto de Web es el servidor y la computadora personal del usuario es el cliente. En la mayoría de los casos, un servidor puede manejar una gran cantidad de clientes.

Si vemos el modelo cliente-servidor en detalle, nos daremos cuenta de que hay dos procesos involucrados, uno en la máquina cliente y otro en la máquina servidor. La comunicación toma la siguiente forma: el proceso cliente envía una solicitud a través de la red al proceso servidor y espera una respuesta. Cuando el proceso servidor recibe la solicitud, realiza el trabajo que se le pide o busca los datos solicitados y devuelve una respuesta. Estos mensajes se muestran en la figura 1-2.

Un segundo objetivo de la configuración de una red de computadoras tiene que ver más con la gente que con la información e, incluso, con las computadoras mismas. Una red de computadoras

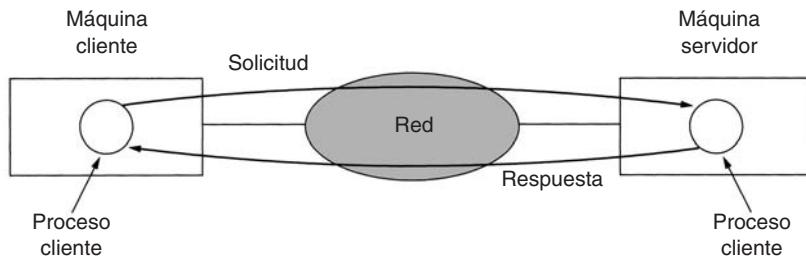


Figura 1-2. El modelo cliente-servidor implica solicitudes y respuestas.

es un poderoso **medio de comunicación** entre los empleados. Casi todas las compañías que tienen dos o más computadoras cuentan con **correo electrónico**, mediante el cual los empleados mantienen generalmente una comunicación diaria. De hecho, una queja común es la gran cantidad de correo electrónico que tenemos que atender, mucho de él sin sentido porque los jefes han descubierto que pueden enviar el mismo mensaje (a menudo sin contenido) a todos sus subordinados con sólo oprimir un botón.

Pero el correo electrónico no es la única forma de comunicación mejorada que las redes de computadoras hacen posible. Con una red es fácil que dos o más personas que trabajan a distancia escriban en conjunto un informe. Si un empleado hace un cambio a un documento en línea, los demás pueden ver el cambio de inmediato, en vez de esperar una carta durante varios días. Esta agilización facilita la cooperación entre grupos de personas que no se encuentran en el mismo lugar, lo cual antes había sido imposible.

Otra forma de comunicación asistida por computadora es la videoconferencia. Con esta tecnología, los empleados en ubicaciones distantes pueden tener una reunión, viéndose y escuchándose unos a otros e incluso escribiendo en una pizarra virtual compartida. La videoconferencia es una herramienta poderosa para eliminar el costo y el tiempo que anteriormente se empleaba en viajar. A veces se dice que la comunicación y el transporte están en competencia, y que el que gane hará obsoleto al otro.

Una tercera meta para cada vez más compañías es hacer negocios de manera electrónica con otras compañías, sobre todo proveedores y clientes. Por ejemplo, los fabricantes de automóviles, de aviones, de computadoras, etcétera, compran subsistemas de diversos proveedores y luego ensamblan las partes. Mediante las redes de computadoras los fabricantes pueden hacer pedidos electrónicamente conforme se requieran. Tener la capacidad de hacer pedidos en tiempo real (es decir, conforme se requieren) reduce la necesidad de tener grandes inventarios y mejora la eficiencia.

Una cuarta meta que se está volviendo más importante es la de hacer negocios con consumidores a través de Internet. Las líneas aéreas, las librerías y los vendedores de música han descubierto que muchos consumidores prefieren realizar sus compras desde casa. Por consiguiente, muchas compañías proporcionan en línea catálogos de sus productos y servicios y levantan pedidos de la misma manera. Se espera que este sector crezca rápidamente en el futuro. Es lo que se conoce como **comercio electrónico**.

1.1.2 Aplicaciones domésticas

En 1977 Ken Olsen era presidente de Digital Equipment Corporation, que en esa época era el segundo proveedor de computadoras en el mundo (después de IBM). Cuando se le preguntó por qué Digital no perseguía el mercado de las computadoras personales en gran volumen, contestó: “No hay razón alguna para que un individuo tenga una computadora en su casa”. La historia demostró lo contrario y Digital ya no existe. ¿Por qué la gente compra computadoras para uso doméstico? En principio, para procesamiento de texto y juegos, pero en los últimos años esto ha cambiado radicalmente. Tal vez la razón más importante ahora sea por el acceso a Internet. Algunos de los usos más comunes de Internet por parte de usuarios domésticos son los siguientes:

1. Acceso a información remota.
2. Comunicación de persona a persona.
3. Entretenimiento interactivo.
4. Comercio electrónico.

El acceso a la información remota se puede realizar por diversas razones. Puede ser que navegue por World Wide Web para obtener información o sólo por diversión. La información disponible incluye artes, negocios, cocina, gobiernos, salud, historia, pasatiempos, recreación, ciencia, deportes, viajes y muchas otras cosas más. La diversión viene en demasiadas formas como para mencionarlas, más algunas otras que es mejor no mencionar.

Muchos periódicos ahora están disponibles en línea y pueden personalizarse. Por ejemplo, en algunos casos le puede indicar a un periódico que desea toda la información acerca de políticos corruptos, incendios, escándalos que involucran a las celebridades y epidemias, pero nada sobre fútbol. Incluso puede hacer que los artículos que usted desea se descarguen en su disco duro o se impriman mientras usted duerme, para que cuando se levante a desayunar los tenga disponibles. Mientras continúe esta tendencia, se provocará el desempleo masivo de los niños de 12 años que entregan los diarios, pero los periódicos lo quieren así porque la distribución siempre ha sido el punto débil en la gran cadena de producción.

El tema más importante después de los periódicos (además de las revistas y periódicos científicos) son las bibliotecas digitales en línea. Muchas organizaciones profesionales, como la ACM (www.acm.org) y la Sociedad de Computación del IEEE (www.computer.org), ya cuentan con muchos periódicos y presentaciones de conferencias en línea. Otros grupos están creciendo de manera rápida. Dependiendo del costo, tamaño y peso de las computadoras portátiles, los libros impresos podrían llegar a ser obsoletos. Los escépticos deben tomar en cuenta el efecto que la imprenta tuvo sobre los manuscritos ilustrados del medioevo.

Todas las aplicaciones anteriores implican las interacciones entre una persona y una base de datos remota llena de información. La segunda gran categoría del uso de redes es la comunicación de persona a persona, básicamente la respuesta del siglo XXI al teléfono del siglo XIX. Millones de personas en todo el mundo utilizan a diario el correo electrónico y su uso está creciendo rápidamente. Ya es muy común que contenga audio y vídeo, así como texto y figuras. Los aromas podrían tardar un poco más.

Muchas personas utilizan los **mensajes instantáneos**. Esta característica, derivada del programa *talk* de UNIX, que se utiliza aproximadamente desde 1970, permite que las personas se escriban mensajes en tiempo real. Una versión, para varias personas, de esta idea es el **salón de conversación** (*chat room*), en el que un grupo de personas puede escribir mensajes para que todos los vean.

Los grupos de noticias mundiales, con debates sobre todo tema imaginable, ya son un lugar común entre un grupo selecto de personas y este fenómeno crecerá para abarcar a la población en general. Estos debates, en los que una persona envía un mensaje y todos los demás suscriptores del grupo de noticias lo pueden leer, van desde los humorísticos hasta los apasionados. A diferencia de los salones de conversación, los grupos de noticias no son en tiempo real y los mensajes se guardan para que cuando alguien vuelva de vacaciones, encuentre todos los mensajes que hayan sido enviados en el ínterin, esperando pacientemente a ser leídos.

Otro tipo de comunicación de persona a persona a menudo se conoce como comunicación de **igual a igual** (*peer to peer*), para distinguirla del modelo cliente-servidor (Parameswaran y cols., 2001). De esta forma, los individuos que forman un grupo espacioso se pueden comunicar con otros del grupo, como se muestra en la figura 1-3. Cada persona puede, en principio, comunicarse con una o más personas; no hay una división fija de clientes y servidores.

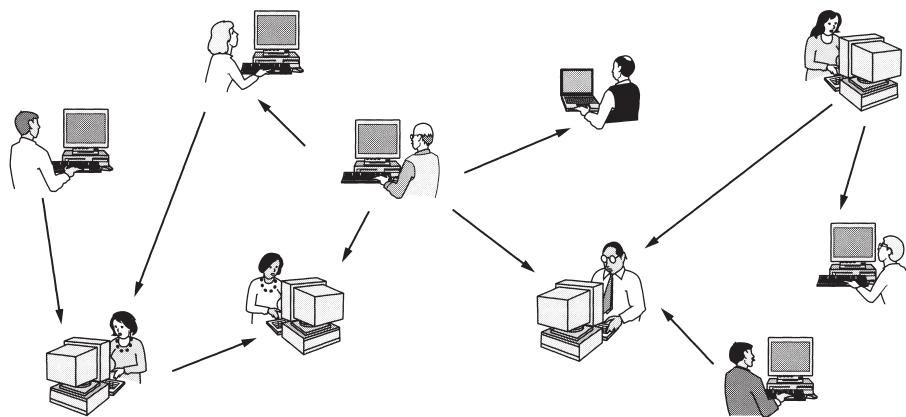


Figura 1-3. En el sistema de igual a igual no hay clientes ni servidores fijos.

La comunicación de igual a igual dominó la mayor parte del 2000 con un servicio llamado Napster, que en su mayor apogeo tenía más de 50 millones de personas canjeando música, lo que fue probablemente la mayor infracción a derechos de autor en toda la historia de la música grabada (Lam y Tan, 2001, y Macedonia, 2000). La idea era muy sencilla. Los miembros registraban en una base de datos central mantenida en el servidor de Napster la música que tenían en sus discos duros. Si un miembro deseaba una canción, verificaba la base de datos para ver quién la tenía e iba directamente ahí para obtenerla. Al no conservar realmente ninguna obra musical en las máquinas, Napster argumentaba que no estaba infringiendo los derechos de autor de nadie. Las cortes no estuvieron de acuerdo y lo clausuraron.

Sin embargo, la siguiente generación de sistemas de igual a igual elimina la base de datos central al hacer que cada usuario mantenga su propia base de datos de manera local, y al proporcionarle una lista de otras personas cercanas que también son miembros del sistema. De esta manera, un nuevo usuario puede ir a cualquiera de ellas para ver qué tiene y obtener una lista de otras más para indagar acerca de más música y más nombres. Este proceso de consulta se puede repetir de manera indefinida hasta construir una enorme base de datos local de lo que hay a disposición. Es una actividad que podría ser tediosa para las personas pero que para las computadoras es muy sencilla.

También existen las aplicaciones legales para la comunicación de igual a igual. Por ejemplo, un club de admiradores que comparte un dominio público de música o cintas de muestra que las nuevas bandas han cedido para efectos de publicidad, familias que comparten fotografías, películas e información genealógica y adolescentes que juegan en línea juegos para varias personas. De hecho, una de las aplicaciones de Internet más populares, el correo electrónico, es esencialmente de igual a igual. Se espera que esta forma de comunicación crezca con rapidez en el futuro.

Los delitos electrónicos no se limitan a la ley de derechos de autor. Otra área activa es la de los juegos electrónicos. Las computadoras han simulado cosas durante décadas. ¿Por qué no simular máquinas tragamonedas, ruedas de la fortuna, repartidores de blackjack y más equipo de juegos electrónicos? El problema es que los juegos electrónicos son legales en muchos lugares (Inglaterra, por ejemplo) y los propietarios de casinos han aprovechado el potencial de los juegos electrónicos por Internet. ¿Qué pasaría si el jugador y el casino estuvieran en países diferentes entre los cuales hay conflicto de leyes? Ésa es una buena pregunta.

Otras aplicaciones orientadas a la comunicación y de rápido crecimiento incluyen el uso de Internet para transportar llamadas telefónicas, el teléfono con vídeo y la radio por Internet. Otra aplicación es el teleaprendizaje, es decir, asistir a clases a las 8:00 A.M. sin el inconveniente de tener que levantarse antes de la cama. A largo plazo, el uso de las redes para mejorar la comunicación de persona a persona puede demostrar que ésta es el área más importante.

Nuestra tercera categoría es el entretenimiento, que es una industria grande y en crecimiento. La aplicación dominante (la que podría impulsar al resto) es el vídeo bajo demanda. De aquí a 10 años, podría seleccionar cualquier película o programa de televisión producido en cualquier país y proyectarlo en su pantalla al instante. Las películas nuevas podrían llegar a ser interactivas, en las que se pediría ocasionalmente al usuario que eligiera el rumbo de la narración, con escenarios alternativos preparados para todos los casos. La televisión en vivo también podría llegar a ser interactiva, permitiendo que la audiencia participe en programas de preguntas, elija entre los competidores, etcétera.

Por otra parte, tal vez el vídeo bajo demanda no sea la aplicación dominante. Podría ser la de los juegos. En la actualidad ya contamos con juegos de simulación de varias personas en tiempo real, como el de las escondidas en un calabozo virtual y simuladores de vuelo en los que los jugadores de un equipo tratan de derribar a los del equipo contrario. Si los juegos se juegan con anteojos y tiempo real tridimensional, con imágenes en movimiento de calidad fotográfica, tenemos un tipo de realidad virtual compartida a nivel mundial.

Nuestra cuarta categoría es el comercio electrónico en el más amplio sentido de la palabra. Comprar desde el hogar ya es una actividad común y permite que los usuarios inspeccionen los

catálogos en línea de miles de compañías. Algunos de estos catálogos proporcionarán pronto la capacidad de obtener un vídeo instantáneo de cualquier producto con sólo hacer clic en el nombre de éste. Si un cliente compra un producto por vía electrónica y no sabe cómo usarlo, podrá consultar el soporte técnico en línea.

Otra área en la que el comercio electrónico ya se está dando es en las instituciones financieras. Mucha gente ya efectúa sus pagos, administra sus cuentas bancarias y maneja sus inversiones de manera electrónica. Seguramente esto crecerá en cuanto las redes sean más seguras.

Un área que prácticamente nadie previó son los mercados de pulgas electrónicos. Las subastas en línea de artículos de segunda mano se han convertido en una industria masiva. A diferencia del comercio electrónico tradicional, que sigue el modelo cliente-servidor, las subastas en línea son más que un sistema de igual a igual, un tipo de sistema de consumidor a consumidor. Algunas de estas formas de comercio electrónico han adoptado una serie de etiquetas con base en que “to” y “2” (en inglés) suenan igual. La figura 1-4 presenta una lista de las abreviaturas más comunes.

Etiqueta	Nombre completo	Ejemplo
B2C	Negocio a consumidor	Pedido de libros en línea
B2B	Negocio a negocio	La fábrica de automóviles hace un pedido de llantas al proveedor
G2C	Gobierno a consumidor	El gobierno distribuye formas fiscales electrónicamente
C2C	Consumidor a consumidor	Subasta en línea de productos de segunda mano
P2P	Igual a igual	Compartición de archivos

Figura 1-4. Algunas formas de comercio electrónico.

Sin duda, el rango de usos de las redes de computadoras crecerá con rapidez y probablemente en formas que nadie puede prever ahora. Después de todo, ¿cuánta gente pudo predecir en 1990 que en diez años las personas podrían escribir mensajes breves en teléfonos celulares durante sus viajes en autobús, lo cual podría ser una forma muy ventajosa para que las compañías telefónicas ganaran dinero? Sin embargo, en la actualidad el servicio de mensajes breves es muy rentable.

Las redes de computadoras podrían llegar a ser sumamente importantes para la gente que no vive en las grandes ciudades, pues les da el mismo acceso a servicios que a las personas que sí viven en ellas. El teleaprendizaje podría afectar radicalmente la educación; las universidades podrían dar servicio a estudiantes nacionales o internacionales. La telemedicina está en inicio (por ejemplo, se utiliza para la supervisión remota de un paciente), pero puede llegar a ser muy importante. Sin embargo, la aplicación clave podría ser algo mundano, como utilizar una *webcam* (cámara conectada a Internet) en su refrigerador, para saber si tiene que comprar leche al regresar del trabajo.

1.1.3 Usuarios móviles

Las computadoras portátiles, como las *notebook* y los asistentes personales digitales (PDAs), son uno de los segmentos de crecimiento más rápido de la industria de la computación. Muchos propietarios de estas computadoras poseen máquinas de escritorio en la oficina y desean estar conectados a su base doméstica cuando están de viaje o fuera de casa. Puesto que no

es posible tener una conexión alámbrica en autos y aviones, hay un gran interés en las redes inalámbricas. En esta sección veremos brevemente algunos usos de ellas.

¿Por qué querría alguien una? Un argumento común es la oficina portátil. Con frecuencia, las personas que están de viaje desean utilizar sus equipos portátiles para enviar y recibir llamadas telefónicas, faxes y correo electrónico, navegar en Web, acceder a archivos remotos e iniciar sesión en máquinas remotas. Y desean hacer esto desde cualquier punto, ya sea por tierra, mar o aire. Por ejemplo, actualmente en las conferencias por computadora, los organizadores suelen configurar una red inalámbrica en el área de la conferencia. Cualquiera que tenga una computadora portátil y un módem inalámbrico puede conectarse a Internet, como si la computadora estuviera conectada a una red alámbrica (cableada). Del mismo modo, algunas universidades han instalado redes inalámbricas en sus campus para que los estudiantes se puedan sentar entre los árboles y consultar los archivos de la biblioteca o leer su correo electrónico.

Las redes inalámbricas son de gran utilidad para las flotas de camiones, taxis, vehículos de entrega y reparadores, para mantenerse en contacto con la casa. Por ejemplo, en muchas ciudades los taxistas trabajan por su cuenta, más que para una empresa de taxis. En algunas de estas ciudades, los taxis tienen una pantalla que el conductor puede ver. Cuando el cliente solicita un servicio, un despachador central escribe los puntos en los que el chofer deberá recoger y dejar al cliente. Esta información se despliega en las pantallas de los conductores y suena un timbre. El conductor que oprima primero un botón en la pantalla recibe la llamada.

Las redes inalámbricas también son importantes para la milicia. Si tiene que estar disponible en breve para pelear una guerra en cualquier parte de la Tierra, probablemente no sea bueno pensar en utilizar la infraestructura de conectividad de redes local. Lo mejor sería tener la propia.

Aunque la conectividad inalámbrica y la computación portátil se relacionan frecuentemente, no son idénticas, como se muestra en la figura 1-5, en la que vemos una diferencia entre **inalámbrica fija** e **inalámbrica móvil**. Incluso en ocasiones las computadoras portátiles son alámbricas. Por ejemplo, si un viajero conecta una portátil a una toma telefónica en su habitación del hotel, tiene movilidad sin una red inalámbrica.

Inalámbrica	Móvil	Aplicaciones
No	No	Computadoras de escritorio en oficinas
No	Sí	Una computadora portátil usada en un cuarto de hotel
Sí	No	Redes en construcciones antiguas sin cableado
Sí	Sí	Oficina portátil; PDA para inventario de almacén

Figura 1-5. Combinaciones de redes inalámbricas y computación móvil.

Por otra parte, algunas computadoras inalámbricas no son móviles. Un ejemplo representativo sería una compañía que posee un edificio antiguo que no tiene cableado de redes y que desea conectar sus computadoras. La instalación de una red inalámbrica podría requerir un poco más que comprar una caja pequeña con algunos aparatos electrónicos, desempacarlos y conectarlos. Sin embargo, esta solución podría ser mucho más barata que contratar trabajadores que coloquen ductos de cable para acondicionar el edificio.

Desde luego, también existen las aplicaciones inalámbricas móviles, que van desde la oficina portátil hasta las personas que pasean por una tienda con un PDA realizando un inventario. En muchos aeropuertos, los empleados de alquiler de coches trabajan en los estacionamientos con computadoras portátiles inalámbricas. Escriben el número de la placa de circulación de los autos alquilados, y su computadora portátil, que tiene una impresora integrada, llama a la computadora principal, obtiene la información del arrendamiento e imprime la factura en el acto.

Conforme se extienda la tecnología inalámbrica, es probable que surjan otras aplicaciones. Echemos un vistazo a algunas de las posibilidades. Los parquímetros inalámbricos tienen ventajas para los usuarios y las autoridades administrativas gubernamentales. Los medidores pueden aceptar tarjetas de crédito o de débito y verificarlas de manera instantánea a través del vínculo inalámbrico. Cuando un medidor expire, se podría verificar la presencia de un auto (emitiendo una señal) y reportar la expiración a la policía. Se ha estimado que con esta medida, los gobiernos de las ciudades de Estados Unidos podrían colectar \$10 mil millones adicionales (Harte y cols., 2000). Además, la entrada en vigor del aparcamiento ayudaría al ambiente, debido a que los conductores que al saber que podrían ser detenidos al estacionarse de manera ilegal, utilizarían el transporte público.

Los expendedores automáticos de alimentos, bebidas, etcétera, se encuentran por todas partes. Sin embargo, los alimentos no entran en las máquinas por arte de magia. Periódicamente, alguien va con un camión y las llena. Si los expendedores automáticos emitieran informes periódicos una vez al día en los que indicaran sus inventarios actuales, el conductor del camión sabría qué máquinas necesitan servicio y qué cantidad de qué productos llevar. Esta información podría conducir a una mayor eficiencia en la planeación de las rutas. Desde luego que esta información también se podría enviar a través de un teléfono de línea común, pero proporcionar a cada expendedor automático una conexión fija telefónica para que realice una llamada al día es costoso debido a los cargos fijos mensuales.

Otra área en la que la tecnología inalámbrica podría ahorrar dinero es en la lectura de medidores de servicios públicos. Si los medidores de electricidad, gas, agua y otros servicios domésticos reportaran su uso a través de una red inalámbrica, no habría necesidad de enviar lectores de medidores. Del mismo modo, los detectores inalámbricos de humo podrían comunicarse con el departamento de bomberos en lugar de hacer tanto ruido (lo cual no sirve de nada si no hay nadie en casa). Conforme baje el costo de los dispositivos de radio y el tiempo aire, más y más medidas e informes se harán a través de redes inalámbricas.

Un área de aplicación totalmente diferente para las redes inalámbricas es la fusión esperada de teléfonos celulares y PDAs en computadoras inalámbricas diminutas. Un primer intento fue el de los diminutos PDAs que podían desplegar páginas Web reducidas al mínimo en sus pequeñas pantallas. Este sistema, llamado **WAP 1.0 (Protocolo de Aplicaciones Inalámbricas)**, falló en gran parte debido a sus pantallas microscópicas, bajo ancho de banda y servicio deficiente. Pero con WAP 2.0 serán mejores los dispositivos y servicios nuevos.

La fuerza que impulsa estos dispositivos es la llamada **comercio móvil** (*m-commerce*) (Senn, 2000). La fuerza que impulsa este fenómeno consiste en diversos fabricantes de PDAs inalámbricos y operadores de redes que luchan por descubrir cómo ganar una parte del pastel del comercio móvil. Una de sus esperanzas es utilizar los PDAs inalámbricos para servicios bancarios y de compras. Una idea es utilizar los PDAs inalámbricos como un tipo de cartera electrónica, que

autorice pagos en tiendas como un reemplazo del efectivo y las tarjetas de crédito. De este modo, el cargo aparecerá en la factura del teléfono celular. Desde el punto de vista de la tienda, este esquema le podría ahorrar la mayor parte de la cuota de la empresa de tarjetas de crédito, que puede ser un porcentaje importante. Desde luego, este plan puede resultar contraproducente, puesto que los clientes que están en una tienda podrían utilizar los PDAs para verificar los precios de la competencia antes de comprar. Peor aún, las compañías telefónicas podrían ofrecer PDAs con lectores de códigos de barras que permitan a un cliente rastrear un producto en una tienda y obtener en forma instantánea un informe detallado de dónde más se puede comprar y a qué precio.

Puesto que el operador de redes sabe dónde está el usuario, algunos servicios se hacen intencionalmente dependientes de la ubicación. Por ejemplo, se podría preguntar por una librería cercana o un restaurante chino. Los mapas móviles y los pronósticos meteorológicos muy locales (“¿Cuándo va a dejar de llover en mi traspasio?”) son otros candidatos. Sin duda, aparecerán otras muchas aplicaciones en cuanto estos dispositivos se difundan más ampliamente.

Un punto muy importante para el comercio móvil es que los usuarios de teléfonos celulares están acostumbrados a pagar por todo (en contraste con los usuarios de Internet, que esperan recibir prácticamente todo sin costo). Si un sitio Web cobrara una cuota por permitir a sus clientes pagar con tarjeta de crédito, provocaría una reclamación muy ruidosa de los usuarios. Si un operador de telefonía celular permitiera que las personas pagaran artículos en una tienda utilizando el teléfono celular y luego cargara una cuota por este servicio, probablemente sus clientes lo aceptarían como algo normal. Sólo el tiempo lo dirá.

Un poco más lejanas están las redes de área personal y las microcomputadoras personales de bolsillo. IBM ha desarrollado un reloj que ejecuta Linux (el cual incluye el sistema de ventanas X11) y tiene conectividad inalámbrica a Internet para enviar y recibir correo electrónico (Narayanaswami y cols., 2002). En el futuro, las personas podrían intercambiar tarjetas de presentación con sólo exponer sus relojes entre sí. Las computadoras de bolsillo inalámbricas pueden dar acceso a las personas a sitios seguros de la misma manera en que lo hacen las tarjetas de banda magnética (posiblemente en combinación con un código de PIN o medición biométrica). Estos relojes también podrían recuperar información relativa a la ubicación actual del usuario (por ejemplo, restaurantes locales). Las posibilidades son infinitas.

Los relojes inteligentes con radio han sido parte de nuestro espacio mental desde que aparecieron en las tiras cómicas de Dick Tracy, en 1946. Pero, ¿polvo inteligente? Los investigadores en Berkeley han empaquetado una computadora inalámbrica en un cubo de 1 mm por lado (Warneke y cols., 2001). Entre las aplicaciones potenciales se incluyen el seguimiento de inventarios, paquetes e incluso pequeños pájaros, roedores e insectos.

1.1.4 Temas sociales

La amplia introducción de las redes ha presentado problemas sociales, éticos y políticos. Mencionemos brevemente algunos de ellos; un estudio completo requeriría todo un libro, por lo menos. Un rasgo popular de muchas redes son los grupos de noticias o boletines electrónicos mediante los cuales las personas pueden intercambiar mensajes con individuos de los mismos intereses. Siempre y cuando los asuntos se restrinjan a temas técnicos o pasatiempos como la jardinería, no surgirán demasiados problemas.

El problema viene cuando los grupos de noticias se enfocan en temas que las personas en realidad tocan con cuidado, como política, religión o sexo. Los puntos de vista enviados a tales grupos podrían ser ofensivos para algunas personas. Peor aún, podrían no ser políticamente correctos. Además, los mensajes no tienen que limitarse a texto. En la actualidad se pueden enviar fotografías en alta resolución e incluso pequeños videoclips a través de redes de computadoras. Algunas personas practican la filosofía de vive y deja vivir, pero otras sienten que enviar cierto material (por ejemplo, ataques a países o religiones en particular, pornografía, etcétera) es sencillamente inaceptable y debe ser censurado. Los diversos países tienen diferentes y conflictivas leyes al respecto. De esta manera, el debate se aviva.

Las personas han demandado a los operadores de redes, afirmando que son responsables, como sucede en el caso de los periódicos y las revistas, del contenido que transmiten. La respuesta inevitable es que una red es como una compañía de teléfonos o la oficina de correos, por lo que no se puede esperar que vigilen lo que dicen los usuarios. Más aún, si los operadores de redes censuraran los mensajes, borrarían cualquier contenido que contuviera incluso la mínima posibilidad de que se les demandara, pero con esto violarían los derechos de sus usuarios a la libre expresión. Probablemente lo más seguro sería decir que este debate seguirá durante algún tiempo.

Otra área divertida es la de los derechos de los empleados en comparación con los de los empleadores. Muchas personas leen y escriben correo electrónico en el trabajo. Muchos empleadores han exigido el derecho a leer y, posiblemente, censurar los mensajes de los empleados, incluso los enviados desde un equipo doméstico después de las horas de trabajo. No todos los empleados están de acuerdo con esto.

Incluso si los empleadores tienen poder sobre los empleados, ¿esta relación también rige a las universidades y los estudiantes? ¿Qué hay acerca de las escuelas secundarias y los estudiantes? En 1994, la Carnegie-Mellon University decidió suspender el flujo de mensajes entrantes de varios grupos de noticias que trataban sexo porque la universidad sintió que el material era inapropiado para menores (es decir, menores de 18 años). Tomó años recuperarse de este suceso.

Otro tema de importancia es el de los derechos del gobierno y los de los ciudadanos. El FBI ha instalado un sistema en muchos proveedores de servicios de Internet para curiosear entre todos los correos electrónicos en busca de fragmentos que le interesen (Blaze y Bellovin, 2000; Sobel, 2001; Zacks, 2001). El sistema se llamaba originalmente **Carnivore** pero la mala publicidad provocó que se cambiara el nombre por uno menos agresivo que sonara como DCS1000. Pero su objetivo sigue siendo el de espiar a millones de personas con la esperanza de encontrar información acerca de actividades ilegales. Por desgracia, la Cuarta Enmienda de la Constitución de Estados Unidos prohíbe que el gobierno realice investigaciones sin una orden de cateo. Decidir si estas palabras, escritas en el siglo XVIII, aún son válidas en el siglo XXI es un asunto que podría mantener ocupadas a las cortes hasta el siglo XXII.

El gobierno no tiene el monopolio de las amenazas contra la privacidad de una persona. El sector privado también hace su parte. Por ejemplo, los archivos pequeños llamados cookies que los navegadores Web almacenan en las computadoras de los usuarios permiten que las empresas rastreen las actividades de éstos en el ciberespacio, y podrían permitir que los números de tarjeta de crédito, del seguro social y otra información confidencial se divulguen por toda la Internet (Berghel, 2001).

Las redes de computadoras ofrecen la posibilidad de enviar mensajes anónimos. En algunas situaciones esta capacidad podría ser deseable. Por ejemplo, los estudiantes, soldados, empleados y ciudadanos pueden denunciar el comportamiento ilegal de algunos profesores, oficiales, superiores y políticos sin temor a represalias. Por otra parte, en Estados Unidos, y en la mayoría de las democracias, la ley otorga específicamente a una persona acusada el derecho de poder confrontar y desafiar a su acusador en la corte. Las acusaciones anónimas no se pueden usar como evidencia.

En resumen, las redes de computadoras, como la imprenta hace 500 años, permiten que el ciudadano común distribuya sus puntos de vista en diversos modos y a audiencias diferentes, lo cual antes no era posible. Este nuevo fondo de libertad ofrece consigo muchos temas sociales, políticos y morales sin resolver.

Junto con lo bueno viene lo malo. Así parece ser la vida. Internet hace posible encontrar con rapidez información, pero una gran cantidad de ella está mal documentada, es falsa o completamente errónea. El consejo médico que obtuvo en Internet podría haber venido de un ganador del Premio Nobel o de un desertor de la preparatoria. Las redes de computadoras también han introducido nuevos tipos de comportamientos antisociales y criminales. La publicidad no deseada (*spam*) se ha convertido en algo común debido a que algunas personas se dedican a reunir millones de direcciones de correo electrónico y las venden en CD-ROMs a comerciantes. Los mensajes por correo electrónico que contienen elementos activos (básicamente programas o macros que se ejecutan en la máquina del receptor) pueden contener virus potencialmente destructores.

El robo de identidad se ha convertido en un problema grave, ya que los ladrones ahora reúnen información sobre una persona para obtener tarjetas de crédito y otros documentos a nombre de ella. Por último, la capacidad de transmitir música y vídeo de manera digital ha abierto la puerta a violaciones masivas de derechos de autor, que son difíciles de detectar y castigar.

Muchos de estos problemas se podrían resolver si la industria de las computadoras tomara la seguridad de las computadoras con seriedad. Si todos los mensajes se codificaran y autenticaran, sería más difícil que se cometieran delitos. Esta tecnología está bien establecida y la estudiaremos en detalle en el capítulo 8. El problema es que los proveedores de hardware y software saben que poner funciones de seguridad cuesta dinero y que sus clientes no las solicitan. Además, una gran cantidad de los problemas proviene de un software con fallas, debido a que los proveedores saturan de funciones sus programas, lo que implica más código e, inevitablemente, más fallas. Un impuesto a las funciones nuevas podría ayudar, pero eso sería como vender un problema por centavos. Reponer el software defectuoso podría ser bueno, pero eso llevaría a la quiebra a toda la industria del software en el primer año.

1.2 HARDWARE DE REDES

Ya es tiempo de centrar nuevamente la atención en los temas técnicos correspondientes al diseño de redes (la parte de trabajo) y dejar a un lado las aplicaciones y los aspectos sociales de la conectividad (la parte divertida). Por lo general, no hay una sola clasificación aceptada en la que se ajusten todas las redes de computadoras, pero hay dos que destacan de manera importante: la tecnología de transmisión y la escala. Examinaremos cada una a la vez.

En un sentido amplio, hay dos tipos de tecnología de transmisión que se utilizan de manera extensa. Son las siguientes:

1. Enlaces de difusión.
2. Enlaces de punto a punto.

Las **redes de difusión** (*broadcast*) tienen un solo canal de comunicación, por lo que todas las máquinas de la red lo comparten. Si una máquina envía un mensaje corto —en ciertos contextos conocido como **paquete**—, todas las demás lo reciben. Un campo de dirección dentro del paquete especifica el destinatario. Cuando una máquina recibe un paquete, verifica el campo de dirección. Si el paquete va destinado a esa máquina, ésta lo procesa; si va destinado a alguna otra, lo ignora.

En una analogía, imagine a alguien que está parado al final de un corredor con varios cuartos a los lados y que grita: “Jorge, ven. Te necesito”. Aunque en realidad el grito (paquete) podría haber sido escuchado (recibido), por muchas personas, sólo Jorge responde (lo procesa). Los demás simplemente lo ignoran. Otra analogía es la de los anuncios en un aeropuerto que piden a todos los pasajeros del vuelo 644 se reporten en la puerta 12 para abordar de inmediato.

Por lo general, los sistemas de difusión también permiten el direccionamiento de un paquete a *todos* los destinos utilizando un código especial en el campo de dirección. Cuando se transmite un paquete con este código, todas las máquinas de la red lo reciben y procesan. Este modo de operación se conoce como **difusión** (*broadcasting*). Algunos sistemas de difusión también soportan la transmisión a un subconjunto de máquinas, algo conocido como **multidifusión** (*multicasting*). Un esquema posible es la reserva de un bit para indicar la multidifusión. Los bits de dirección $n - 1$ restantes pueden contener un número de grupo. Cada máquina puede “suscribirse” a alguno o a todos los grupos. Cuando se envía un paquete a cierto grupo, se distribuye a todas las máquinas que se suscriben a ese grupo.

En contraste, las redes **punto a punto** constan de muchas conexiones entre pares individuales de máquinas. Para ir del origen al destino, un paquete en este tipo de red podría tener que visitar primero una o más máquinas intermedias. A menudo es posible que haya varias rutas o longitudes diferentes, de manera que encontrar las correctas es importante en redes de punto a punto. Por regla general (aunque hay muchas excepciones), las redes más pequeñas localizadas en una misma área geográfica tienden a utilizar la difusión, mientras que las más grandes suelen ser de punto a punto. La transmisión de punto a punto con un emisor y un receptor se conoce como **unidifusión** (*unicasting*).

Un criterio alternativo para la clasificación de las redes es su escala. En la figura 1-6 clasificamos los sistemas de procesadores múltiples por tamaño físico. En la parte superior se muestran las **redes de área personal**, que están destinadas para una sola persona. Por ejemplo, una red inalámbrica que conecta una computadora con su ratón, teclado e impresora, es una red de área personal. Incluso un PDA que controla el audífono o el marcapaso de un usuario encaja en esta categoría. A continuación de las redes de área personal se encuentran redes más grandes. Se pueden dividir en redes de área local, de área metropolitana y de área amplia. Por último, la conexión de dos o más redes se conoce como interred.

Distancia entre procesadores	Procesadores ubicados en el mismo	Ejemplo
1 m	Metro cuadrado	Red de área personal
10 m	Cuarto	
100 m	Edificio	
1 km	Campus	
10 km	Ciudad	Red de área local
100 km	País	
1,000 km	Continente	Red de área metropolitana
10,000 km	Planeta	Red de área amplia
		Internet

Figura 1-6. Clasificación de procesadores interconectados por escala.

Internet es un ejemplo bien conocido de una interred. La distancia es importante como una clasificación en metros porque se utilizan diferentes técnicas en diferentes escalas. En este libro nos ocuparemos de las redes en todas estas escalas. A continuación se proporciona una breve introducción al hardware de redes.

1.2.1 Redes de área local

Las **redes de área local** (generalmente conocidas como **LANs**) son redes de propiedad privada que se encuentran en un solo edificio o en un campus de pocos kilómetros de longitud. Se utilizan ampliamente para conectar computadoras personales y estaciones de trabajo en oficinas de una empresa y de fábricas para compartir recursos (por ejemplo, impresoras) e intercambiar información. Las LANs son diferentes de otros tipos de redes en tres aspectos: 1) tamaño; 2) tecnología de transmisión, y 3) topología.

Las LANs están restringidas por tamaño, es decir, el tiempo de transmisión en el peor de los casos es limitado y conocido de antemano. El hecho de conocer este límite permite utilizar ciertos tipos de diseño, lo cual no sería posible de otra manera. Esto también simplifica la administración de la red.

Las LANs podrían utilizar una tecnología de transmisión que consiste en un cable al cual están unidas todas las máquinas, como alguna vez lo estuvo parte de las líneas de las compañías telefónicas en áreas rurales. Las LANs tradicionales se ejecutan a una velocidad de 10 a 100 Mbps, tienen un retardo bajo (microsegundos o nanosegundos) y cometan muy pocos errores. Las LANs más nuevas funcionan hasta a 10 Gbps. En este libro continuaremos con lo tradicional y mediremos las velocidades de las líneas en megabits por segundo (1 Mbps es igual a 1,000,000 de bits por segundo) y gigabits por segundo (1 Gbps es igual a 1,000,000,000 de bits por segundo).

Para las LANs de difusión son posibles varias topologías. La figura 1-7 muestra dos de ellas. En una red de bus (es decir, un cable lineal), en cualquier instante al menos una máquina es la maestra y puede transmitir. Todas las demás máquinas se abstienen de enviar. Cuando se presenta el conflicto de que dos o más máquinas desean transmitir al mismo tiempo, se requiere un meca-

nismo de arbitraje. Tal mecanismo podría ser centralizado o distribuido. Por ejemplo, el IEEE 802.3, popularmente conocido como **Ethernet**, es una red de difusión basada en bus con control descentralizado, que por lo general funciona de 10 Mbps a 10 Gbps. Las computadoras que están en una Ethernet pueden transmitir siempre que lo deseen; si dos o más paquetes entran en colisión, cada computadora espera un tiempo aleatorio y lo intenta de nuevo más tarde.

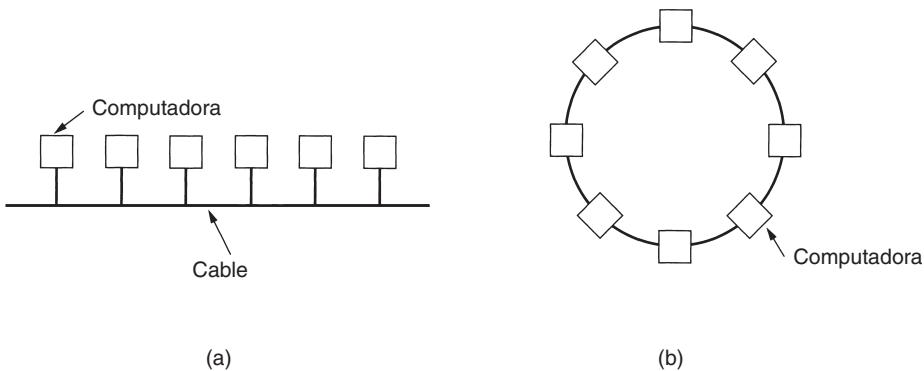


Figura 1-7. Dos redes de difusión. (a) De bus. (b) De anillo.

Un segundo tipo de sistema de difusión es el de anillo. En un anillo, cada bit se propaga por sí mismo, sin esperar al resto del paquete al que pertenece. Por lo común, cada bit navega por todo el anillo en el tiempo que le toma transmitir algunos bits, a veces incluso antes de que se haya transmitido el paquete completo. Al igual que con todos los demás sistemas de difusión, se requieren algunas reglas para controlar los accesos simultáneos al anillo. Se utilizan varios métodos, por ejemplo, el de que las máquinas deben tomar su turno. El IEEE 802.5 (el token ring de IBM) es una LAN basada en anillo que funciona a 4 y 16 Mbps. El FDDI es otro ejemplo de una red de anillo.

Las redes de difusión se pueden dividir aún más en estáticas y dinámicas, dependiendo de cómo se asigne el canal. Una asignación estática típica sería dividir el tiempo en intervalos discretos y utilizar un algoritmo *round-robin*, permitiendo que cada máquina transmita sólo cuando llegue su turno. La asignación estática desperdicia capacidad de canal cuando una máquina no tiene nada que transmitir al llegar su turno, por lo que la mayoría de los sistemas trata de asignar el canal de forma dinámica (es decir, bajo demanda).

Los métodos de asignación dinámica para un canal común pueden ser centralizados o descentralizados. En el método centralizado hay una sola entidad, por ejemplo, una unidad de arbitraje de bus, la cual determina quién sigue. Esto se podría hacer aceptando solicitudes y tomando decisiones de acuerdo con algunos algoritmos internos. En el método descentralizado de asignación de canal no hay una entidad central; cada máquina debe decidir por sí misma cuándo transmitir. Usted podría pensar que esto siempre conduce al caos, pero no es así. Más adelante estudiaremos muchos algoritmos designados para poner orden y evitar el caos potencial.

1.2.2 Redes de área metropolitana

Una **red de área metropolitana (MAN)** abarca una ciudad. El ejemplo más conocido de una MAN es la red de televisión por cable disponible en muchas ciudades. Este sistema creció a partir de los primeros sistemas de antena comunitaria en áreas donde la recepción de la televisión al aire era pobre. En dichos sistemas se colocaba una antena grande en la cima de una colina cercana y la señal se canalizaba a las casas de los suscriptores.

Al principio eran sistemas diseñados de manera local con fines específicos. Después las compañías empezaron a pasar a los negocios, y obtuvieron contratos de los gobiernos de las ciudades para cablear toda una ciudad. El siguiente paso fue la programación de televisión e incluso canales designados únicamente para cable. Con frecuencia, éstos emitían programas de un solo tema, como sólo noticias, deportes, cocina, jardinería, etcétera. Sin embargo, desde su inicio y hasta finales de la década de 1990, estaban diseñados únicamente para la recepción de televisión.

A partir de que Internet atrajo una audiencia masiva, los operadores de la red de TV por cable se dieron cuenta de que con algunos cambios al sistema, podrían proporcionar servicio de Internet de dos vías en las partes sin uso del espectro. En ese punto, el sistema de TV por cable empezaba a transformarse de una forma de distribución de televisión a una red de área metropolitana. Para que se dé una idea, una MAN podría verse como el sistema que se muestra en la figura 1-8, donde se aprecia que las señales de TV e Internet se alimentan hacia un **amplificador head end** paraenseguida transmitirse a las casas de las personas. En el capítulo 2 trataremos con detalle este tema.

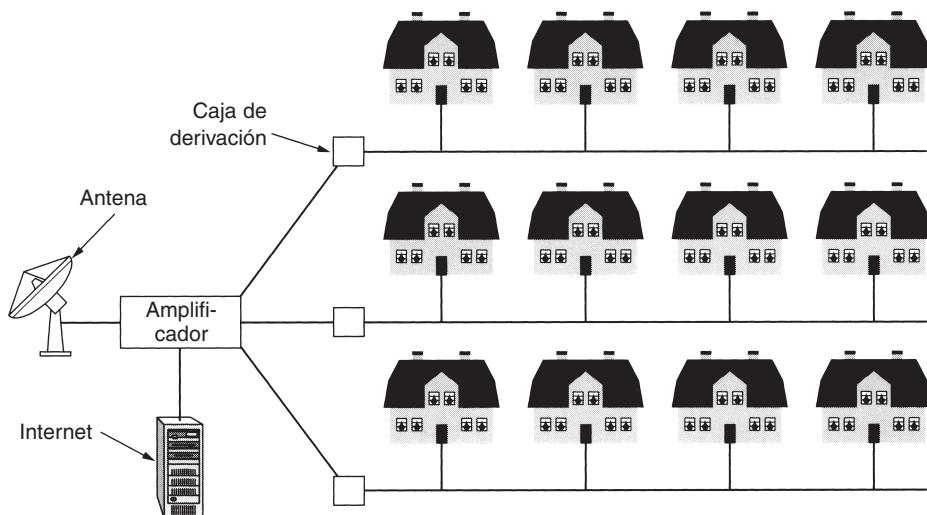


Figura 1-8. Una red de área metropolitana, basada en TV por cable.

La televisión por cable no es solamente una MAN. Desarrollos recientes en el acceso inalámbrico a alta velocidad a Internet dieron como resultado otra MAN, que se estandarizó como IEEE 802.16. En el capítulo 2 veremos esta área.

1.2.3 Redes de área amplia

Una **red de área amplia (WAN)**, abarca una gran área geográfica, con frecuencia un país o un continente. Contiene un conjunto de máquinas diseñado para programas (es decir, aplicaciones) de usuario. Seguiremos el uso tradicional y llamaremos **hosts** a estas máquinas. Los *hosts* están conectados por una **subred de comunicación**, o simplemente **subred**, para abreviar. Los clientes son quienes poseen a los *hosts* (es decir, las computadoras personales de los usuarios), mientras que, por lo general, las compañías telefónicas o los proveedores de servicios de Internet poseen y operan la subred de comunicación. La función de una subred es llevar mensajes de un *host* a otro, como lo hace el sistema telefónico con las palabras del que habla al que escucha. La separación de los aspectos de la comunicación pura de la red (la subred) de los aspectos de la aplicación (los *hosts*), simplifica en gran medida todo el diseño de la red.

En la mayoría de las redes de área amplia la subred consta de dos componentes distintos: líneas de transmisión y elementos de commutación. Las **líneas de transmisión** mueven bits entre máquinas. Pueden estar hechas de cable de cobre, fibra óptica o, incluso, radioenlaces. Los **elementos de commutación** son computadoras especializadas que conectan tres o más líneas de transmisión. Cuando los datos llegan a una línea de entrada, el elemento de commutación debe elegir una línea de salida en la cual reenviarlos. Estas computadoras de commutación reciben varios nombres; commutadores y enrutadores son los más comunes.

En este modelo, que se muestra en la figura 1-9, cada *host* está conectado frecuentemente a una LAN en la que existe un enrutador, aunque en algunos casos un *host* puede estar conectado de manera directa a un enrutador. El conjunto de líneas de comunicación y enrutadores (pero no de *hosts*) forma la subred.

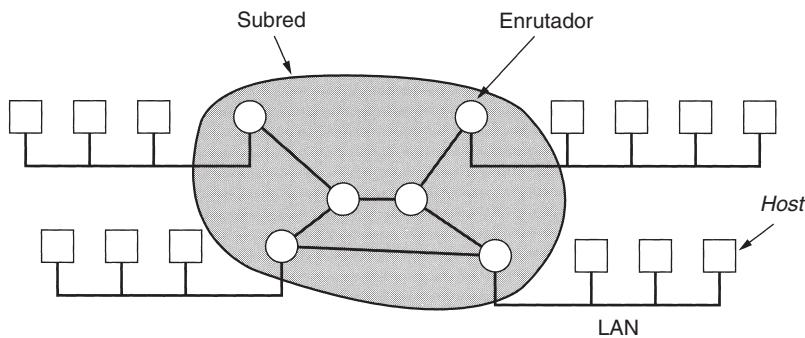


Figura 1-9. Relación entre *hosts* de LANs y la subred.

A continuación se presenta un breve comentario acerca del término “subred”. Originalmente, su **único** significado era el conjunto de enrutadores y líneas de comunicación que movía paquetes del *host* de origen al de destino. Sin embargo, algunos años más tarde también adquirió un segundo

significado junto con el direccionamiento de redes (que expondremos en el capítulo 5). Desgraciadamente, no existe una alternativa de amplio uso con respecto a su significado inicial por lo que, con algunas reservas, utilizaremos este término en ambos sentidos. El contexto dejará en claro su significado.

En la mayoría de las WANs, la red contiene numerosas líneas de transmisión, cada una de las cuales conecta un par de enrutadores. Si dos enrutadores que no comparten una línea de transmisión quieren conectarse, deberán hacerlo de manera indirecta, a través de otros enrutadores. Cuando un paquete es enviado desde un enrutador a otro a través de uno o más enrutadores intermedios, el paquete se recibe en cada enrutador intermedio en su totalidad, se almacena ahí hasta que la línea de salida requerida esté libre y, por último, se reenvía. Una subred organizada a partir de este principio se conoce como subred de **almacenamiento y reenvío** (*store and forward*) o de **comunicación de paquetes**. Casi todas las redes de área amplia (excepto las que utilizan satélites) tienen subredes de almacenamiento y reenvío. Cuando los paquetes son pequeños y tienen el mismo tamaño, se les llama **celdas**.

El principio de una WAN de comutación de paquetes es tan importante que vale la pena dedicarle algunas palabras más. En general, cuando un proceso de cualquier *host* tiene un mensaje que se va a enviar a un proceso de algún otro *host*, el *host* emisor divide primero el mensaje en paquetes, los cuales tienen un número de secuencia. Estos paquetes se envían entonces por la red de uno en uno en una rápida sucesión. Los paquetes se transportan de forma individual a través de la red y se depositan en el *host* receptor, donde se reensamblan en el mensaje original y se entregan al proceso receptor. En la figura 1-10 se ilustra un flujo de paquetes correspondiente a algún mensaje inicial.

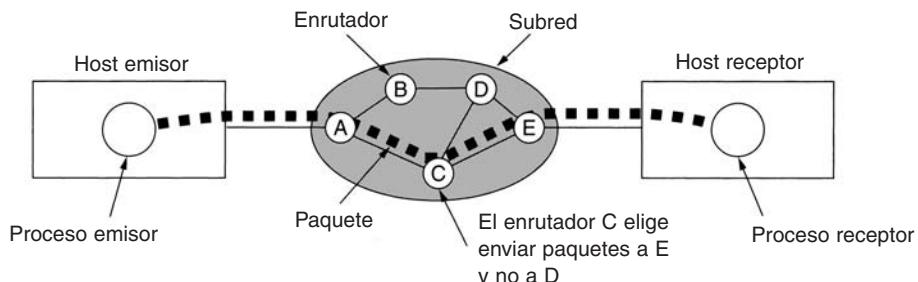


Figura 1-10. Flujo de paquetes desde un emisor a un receptor.

En esta figura todos los paquetes siguen la ruta *ACE* en vez de la *ABDE* o *ACDE*. En algunas redes todos los paquetes de un mensaje determinado *deben* seguir la misma ruta; en otras, cada paquete se enruta por separado. Desde luego, si *ACE* es la mejor ruta, todos los paquetes se podrían enviar a través de ella, incluso si cada paquete se enruta de manera individual.

Las decisiones de enrutamiento se hacen de manera local. Cuando un paquete llega al enrutador *A*, éste debe decidir si el paquete se enviará hacia *B* o hacia *C*. La manera en que el enrutador *A* toma esa decisión se conoce como **algoritmo de enrutamiento**. Existen muchos de ellos. En el capítulo 5 estudiaremos con detalle algunos.

No todas las WANs son de commutación de paquetes. Una segunda posibilidad para una WAN es un sistema satelital. Cada enrutador tiene una antena a través de la cual puede enviar y recibir. Todos los enrutadores pueden escuchar la salida *desde* el satélite y, en algunos casos, también pueden escuchar las transmisiones de los demás enrutadores *hacia* el satélite. Algunas veces los enrutadores están conectados a una subred de punto a punto elemental, y sólo algunos de ellos tienen una antena de satélite. Por naturaleza, las redes satelital son de difusión y son más útiles cuando la propiedad de difusión es importante.

1.2.4 Redes inalámbricas

La comunicación inalámbrica digital no es una idea nueva. A principios de 1901, el físico italiano Guillermo Marconi demostró un telégrafo inalámbrico desde un barco a tierra utilizando el código Morse (después de todo, los puntos y rayas son binarios). Los sistemas inalámbricos digitales de la actualidad tienen un mejor desempeño, pero la idea básica es la misma.

Como primera aproximación, las redes inalámbricas se pueden dividir en tres categorías principales:

1. Interconexión de sistemas.
2. LANs inalámbricas.
3. WANs inalámbricas.

La interconexión de sistemas se refiere a la interconexión de componentes de una computadora que utiliza radio de corto alcance. La mayoría de las computadoras tiene un monitor, teclado, ratón e impresora, conectados por cables a la unidad central. Son tantos los usuarios nuevos que tienen dificultades para conectar todos los cables en los enchufes correctos (aun cuando suelen estar codificados por colores) que la mayoría de los proveedores de computadoras ofrece la opción de enviar a un técnico a la casa del usuario para que realice esta tarea. En consecuencia, algunas compañías se reunieron para diseñar una red inalámbrica de corto alcance llamada **Bluetooth** para conectar sin cables estos componentes. Bluetooth también permite conectar cámaras digitales, auriculares, escáneres y otros dispositivos a una computadora con el único requisito de que se encuentren dentro del alcance de la red. Sin cables, sin instalación de controladores, simplemente se colocan, se encienden y funcionan. Para muchas personas, esta facilidad de operación es algo grandioso.

En la forma más sencilla, las redes de interconexión de sistemas utilizan el paradigma del maestro y el esclavo de la figura 1-11(a). La unidad del sistema es, por lo general, el maestro que trata al ratón, al teclado, etcétera, como a esclavos. El maestro le dice a los esclavos qué direcciones utilizar, cuándo pueden difundir, durante cuánto tiempo pueden transmitir, qué frecuencias pueden utilizar, etcétera. En el capítulo 4 explicaremos con más detalle el Bluetooth.

El siguiente paso en la conectividad inalámbrica son las LANs inalámbricas. Son sistemas en los que cada computadora tiene un módem de radio y una antena mediante los que se puede comunicar con otros sistemas. En ocasiones, en el techo se coloca una antena con la que las máquinas se comunican, como se ilustra en la figura 1-11(b). Sin embargo, si los sistemas están lo suficientemente cerca, se pueden comunicar de manera directa entre sí en una configuración de

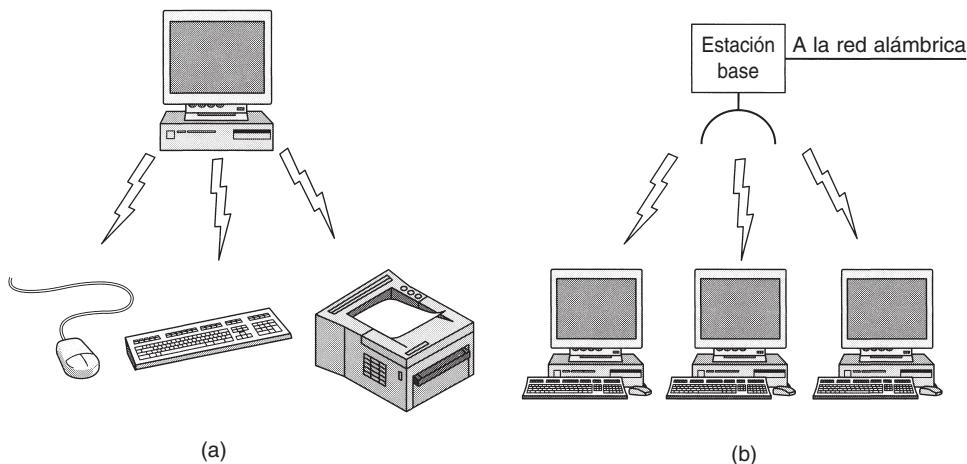


Figura 1-11. (a) Configuración Bluetooth. (b) LAN inalámbrica.

igual a igual. Las LANs inalámbricas se están haciendo cada vez más comunes en casas y oficinas pequeñas, donde instalar Ethernet se considera muy problemático, así como en oficinas ubicadas en edificios antiguos, cafeterías de empresas, salas de conferencias y otros lugares. Existe un estándar para las LANs inalámbricas, llamado **IEEE 802.11**, que la mayoría de los sistemas implementa y que se ha extendido ampliamente. Esto lo explicaremos en el capítulo 4.

El tercer tipo de red inalámbrica se utiliza en sistemas de área amplia. La red de radio utilizada para teléfonos celulares es un ejemplo de un sistema inalámbrico de banda ancha baja. Este sistema ha pasado por tres generaciones. La primera era analógica y sólo para voz. La segunda era digital y sólo para voz. La tercera generación es digital y es tanto para voz como para datos. En cierto sentido, las redes inalámbricas celulares son como las LANs inalámbricas, excepto porque las distancias implicadas son mucho más grandes y las tasas de bits son mucho más bajas. Las LANs inalámbricas pueden funcionar a tasas de hasta 50 Mbps en distancias de decenas de metros. Los sistemas celulares funcionan debajo de 1 Mbps, pero la distancia entre la estación base y la computadora o teléfono se mide en kilómetros más que en metros. En el capítulo 2 hablaremos con mucho detalle sobre estas redes.

Además de estas redes de baja velocidad, también se han desarrollado las redes inalámbricas de área amplia con alto ancho de banda. El enfoque inicial es el acceso inalámbrico a Internet a alta velocidad, desde los hogares y las empresas, dejando a un lado el sistema telefónico. Este servicio se suele llamar servicio de distribución local multipuntos. Lo estudiaremos más adelante. También se ha desarrollado un estándar para éste, llamado IEEE 802.16. Examinaremos dicho estándar en el capítulo 4.

La mayoría de las redes inalámbricas se enlaza a la red alámbrica en algún punto para proporcionar acceso a archivos, bases de datos e Internet. Hay muchas maneras de efectuar estas conexiones, dependiendo de las circunstancias. Por ejemplo, en la figura 1-12(a) mostramos un aeroplano con una serie de personas que utilizan módems y los teléfonos de los respaldos para llamar a la oficina. Cada llamada es independiente de las demás. Sin embargo, una opción mucho

más eficiente es la LAN dentro del avión de la figura 1-12(b), donde cada asiento está equipado con un conector Ethernet al cual los pasajeros pueden acoplar sus computadoras. El avión tiene un solo enrutador, el cual mantiene un enlace de radio con algún enrutador que se encuentre en tierra, y cambia de enrutador conforme avanza el vuelo. Esta configuración es una LAN tradicional, excepto porque su conexión al mundo exterior se da mediante un enlace por radio en lugar de una línea cableada.

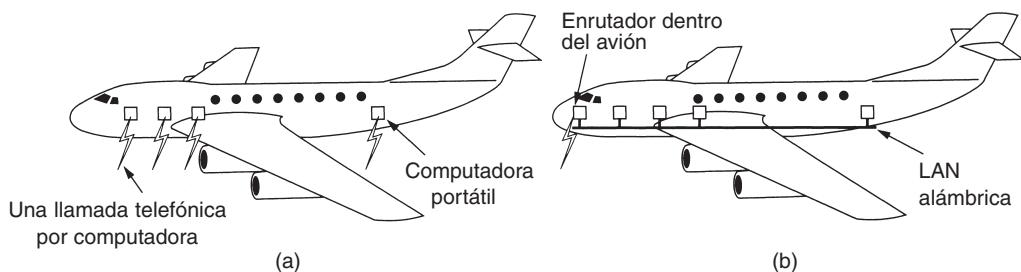


Figura 1-12. (a) Computadoras móviles individuales. (b) LAN dentro del avión.

Muchas personas creen que lo inalámbrico es la onda del futuro (por ejemplo, Bi y cols., 2001; Leeper, 2001; Varshey y Vetter, 2000) pero se ha escuchado una voz disidente. Bob Metcalfe, el inventor de Ethernet, ha escrito: “Las computadoras inalámbricas móviles son como los baños portátiles sin cañería: bacinas portátiles. Serán muy comunes en los vehículos, en sitios en construcción y conciertos de rock. Mi consejo es que coloque cables en su casa y se quede ahí” (Metcalfe, 1995). La historia podría colocar esta cita en la misma categoría que la explicación de T.J. Watson, presidente de IBM en 1945, de por qué esta empresa no entraba en el negocio de las computadoras: “Cuatro o cinco computadoras deberán ser suficientes para todo el mundo hasta el año 2000”.

1.2.5 Redes domésticas

La conectividad doméstica está en el horizonte. La idea fundamental es que en el futuro la mayoría de los hogares estarán preparados para conectividad de redes. Cualquier dispositivo del hogar será capaz de comunicarse con todos los demás dispositivos y todos podrán accederse por Internet. Éste es uno de esos conceptos visionarios que nadie solicitó (como los controles remotos de TV o los teléfonos celulares), pero una vez que han llegado nadie se puede imaginar cómo habían podido vivir sin ellos.

Muchos dispositivos son capaces de estar conectados en red. Algunas de las categorías más evidentes (con ejemplos) son las siguientes:

1. Computadoras (de escritorio, portátiles, PDAs, periféricos compartidos).
2. Entretenimiento (TV, DVD, VCR, videocámara, cámara fotográfica, estereofónicos, MP3).
3. Telecomunicaciones (teléfono, teléfono móvil, intercomunicadores, fax).
4. Aparatos electrodomésticos (horno de microondas, refrigerador, reloj, horno, aire acondicionado, luces).
5. Telemetría (metro utilitario, alarma contra fuego y robo, termostato, cámaras inalámbricas).

La conectividad de computadoras domésticas ya está aquí, aunque limitada. Muchas casas ya cuentan con un dispositivo para conectar varias computadoras para una conexión rápida a Internet. El entretenimiento por red aún no existe, pero cuanto más y más música y películas se puedan descargar de Internet, habrá más demanda para que los equipos de audio y las televisiones se conecten a Internet. Incluso las personas desearán compartir sus propios videos con amigos y familiares, por lo que deberá haber una conexión en ambos sentidos. Los dispositivos de telecomunicaciones ya están conectados al mundo exterior, pero pronto serán digitales y tendrán capacidad de funcionar sobre Internet. Un hogar promedio tal vez tiene una docena de relojes (los de los aparatos electrodomésticos), y todos se tienen que reajustar dos veces al año cuando inicia y termina el tiempo de ahorro de luz de día (horario de verano). Si todos los relojes estuvieran conectados a Internet, ese reajuste se haría en forma automática. Por último, el monitoreo remoto de la casa y su contenido es el probable ganador. Es muy factible que muchos padres deseen invertir en monitorear con sus PDAs a sus bebés dormidos cuando van a cenar fuera de casa, aun cuando contraten a una niñera. Si bien podemos imaginar una red separada para cada área de aplicación, la integración de todas en una sola red es probablemente una mejor idea.

La conectividad doméstica tiene algunas propiedades diferentes a las de otro tipo de redes. Primero, la red y los dispositivos deben ser fáciles de instalar. El autor ha instalado numerosas piezas de hardware y software en varias computadoras durante varios años con resultados diferentes. Al realizar una serie de llamadas telefónicas al personal de soporte técnico del proveedor por lo general recibió respuestas como: 1) Lea el manual; 2) Reinicie la computadora; 3) Elimine todo el hardware y software, excepto los nuestros, y pruebe de nuevo; 4) Descargue de nuestro sitio Web el controlador más reciente y, si todo eso falla, 5) Reformatee el disco duro y reinstale Windows desde el CD-ROM. Decirle al comprador de un refrigerador con capacidad de Internet que descargue e instale una nueva versión del sistema operativo del refrigerador, no conduce a tener clientes contentos. Los usuarios de computadoras están acostumbrados a soportar productos que no funcionan; los clientes que compran automóviles, televisiones y refrigeradores son mucho menos tolerantes. Esperan productos que trabajen al 100% desde que se compran.

Segundo, la red y los dispositivos deben estar plenamente probados en operación. Los equipos de aire acondicionado solían tener una perilla con cuatro parámetros: OFF, LOW, MEDIUM y HIGH (apagado, bajo, medio, alto). Ahora tienen manuales de 30 páginas. Una vez que puedan conectarse en red, no se le haga extraño que tan sólo el capítulo de seguridad tenga 30 páginas. Esto estará más allá de la comprensión de prácticamente todos los usuarios.

Tercero, el precio bajo es esencial para el éxito. Muy pocas personas, si no es que ninguna, pagarán un precio adicional de \$50 por un termostato con capacidad de Internet, debido a que no considerarán que monitorear la temperatura de sus casas desde sus trabajos sea algo importante. Tal vez por \$5 sí lo comprarían.

Cuarto, la principal aplicación podría implicar multimedia, por lo que la red necesita capacidad suficiente. No hay mercado para televisiones conectadas a Internet que proyecten películas inseguras a una resolución de 320×240 píxeles y 10 cuadros por segundo. Fast Ethernet, el caballo de batalla en la mayoría de las oficinas, no es bastante buena para multimedia. En consecuencia, para que las redes domésticas lleguen a ser productos masivos en el mercado, requerirán mejor desempeño que el de las redes de oficina actuales, así como precios más bajos.

Quinto, se podría empezar con uno o dos dispositivos y expandir de manera gradual el alcance de la red. Esto significa que no habrá problemas con el formato. Decir a los consumidores que adquieran periféricos con interfaces IEEE 1394 (FireWire) y años después retractarse y decir que USB 2.0 es la interfaz del mes, es hacer clientes caprichosos. La interfaz de red tendrá que permanecer estable durante muchos años; el cableado (si lo hay) deberá permanecer estable durante décadas.

Sexto, la seguridad y la confianza serán muy importantes. Perder algunos archivos por un virus de correo electrónico es una cosa; que un ladrón desarme su sistema de seguridad desde su PDA y luego saquee su casa es algo muy diferente.

Una pregunta interesante es si las redes domésticas serán alámbricas o inalámbricas. La mayoría de los hogares ya tiene seis redes instaladas: electricidad, teléfono, televisión por cable, agua, gas y alcantarillado. Agregar una séptima durante la construcción de una casa no es difícil, pero acondicionar las casas existentes para agregar dicha red es costoso. Los costos favorecen la conectividad inalámbrica, pero la seguridad favorece la conectividad alámbrica. El problema con la conectividad inalámbrica es que las ondas de radio que utiliza traspasan las paredes con mucha facilidad. No a todos les gusta la idea de que cuando vaya a imprimir, se tope con la conexión de su vecino y pueda leer el correo electrónico de éste. En el capítulo 8 estudiaremos cómo se puede utilizar la encriptación para proporcionar seguridad, pero en el contexto de una red doméstica la seguridad debe estar bien probada, incluso para usuarios inexpertos. Es más fácil decirlo que hacerlo, incluso en el caso de usuarios expertos.

Para abreviar, la conectividad doméstica ofrece muchas oportunidades y retos. La mayoría de ellos se relaciona con la necesidad de que sean fáciles de manejar, confiables y seguros, en particular en manos de usuarios no técnicos, y que al mismo tiempo proporcionen alto desempeño a bajo costo.

1.2.6 Interredes

Existen muchas redes en el mundo, a veces con hardware y software diferentes. Con frecuencia, las personas conectadas a una red desean comunicarse con personas conectadas a otra red diferente. La satisfacción de este deseo requiere que se conecten diferentes redes, con frecuencia incompatibles, a veces mediante máquinas llamadas **puertas de enlace** (*gateways*) para hacer la conexión y proporcionar la traducción necesaria, tanto en términos de hardware como de software. Un conjunto de redes interconectadas se llama **interred**.

Una forma común de interred es el conjunto de LANs conectadas por una WAN. De hecho, si tuviéramos que reemplazar la etiqueta “subred” en la figura 1-9 por “WAN”, no habría nada más que cambiar en la figura. En este caso, la única diferencia técnica real entre una subred y una WAN es si hay *hosts* presentes. Si el sistema que aparece en el área gris contiene solamente enrutadores, es una subred; si contiene enrutadores y *hosts*, es una WAN. Las diferencias reales se relacionan con la propiedad y el uso.

Subredes, redes e interredes con frecuencia se confunden. La subred tiene más sentido en el contexto de una red de área amplia, donde se refiere a un conjunto de enrutadores y líneas de

comunicación poseídas por el operador de redes. Como una analogía, el sistema telefónico consta de oficinas de conmutación telefónica que se conectan entre sí mediante líneas de alta velocidad, y a los hogares y negocios, mediante líneas de baja velocidad. Estas líneas y equipos, poseídas y administradas por la compañía de teléfonos, forman la subred del sistema telefónico. Los teléfonos mismos (los *hosts* en esta analogía) no son parte de la subred. La combinación de una subred y sus *hosts* forma una red. En el caso de una LAN, el cable y los *hosts* forman la red. En realidad, ahí no hay una subred.

Una interred se forma cuando se interconectan redes diferentes. Desde nuestro punto de vista, al conectar una LAN y una WAN o conectar dos LANs se forma una interred, pero existe poco acuerdo en la industria en cuanto a la terminología de esta área. Una regla de oro es que si varias empresas pagaron por la construcción de diversas partes de la red y cada una mantiene su parte, tenemos una interred más que una sola red. Asimismo, si la terminología subyacente es diferente en partes diferentes (por ejemplo, difusión y punto a punto), probablemente tengamos dos redes.

1.3 SOFTWARE DE REDES

Las primeras redes de computadoras se diseñaron teniendo al hardware como punto principal y al software como secundario. Esta estrategia ya no funciona. Actualmente el software de redes está altamente estructurado. En las siguientes secciones examinaremos en detalle la técnica de estructuración de software. El método descrito aquí es la clave de todo el libro y se presentará con mucha frecuencia más adelante.

1.3.1 Jerarquías de protocolos

Para reducir la complejidad de su diseño, la mayoría de las redes está organizada como una pila de **capas** o **niveles**, cada una construida a partir de la que está debajo de ella. El número de capas, así como el nombre, contenido y función de cada una de ellas difieren de red a red. El propósito de cada capa es ofrecer ciertos servicios a las capas superiores, a las cuales no se les muestran los detalles reales de implementación de los servicios ofrecidos.

Este concepto es muy conocido y utilizado en la ciencia computacional, donde se conoce de diversas maneras, como ocultamiento de información, tipos de datos abstractos, encapsulamiento de datos y programación orientada a objetos. La idea básica es que una pieza particular de software (o hardware) proporciona un servicio a sus usuarios pero nunca les muestra los detalles de su estado interno ni sus algoritmos.

La capa *n* de una máquina mantiene una conversación con la capa *n* de otra máquina. Las reglas y convenciones utilizadas en esta conversación se conocen de manera colectiva como protocolo de capa *n*. Básicamente, un **protocolo** es un acuerdo entre las partes en comunicación sobre cómo se debe llevar a cabo la comunicación. Como una analogía, cuando se presenta una mujer con un hombre, ella podría elegir no darle la mano. Él, a su vez, podría decidir saludarla de mano o de beso, dependiendo, por ejemplo, de si es una abogada americana o una princesa europea en

una reunión social formal. Violar el protocolo hará más difícil la comunicación, si no es que imposible.

En la figura 1-13 se ilustra una red de cinco capas. Las entidades que abarcan las capas correspondientes en diferentes máquinas se llaman **iguales** (*peers*). Los iguales podrían ser procesos, dispositivos de hardware o incluso seres humanos. En otras palabras, los iguales son los que se comunican a través del protocolo.

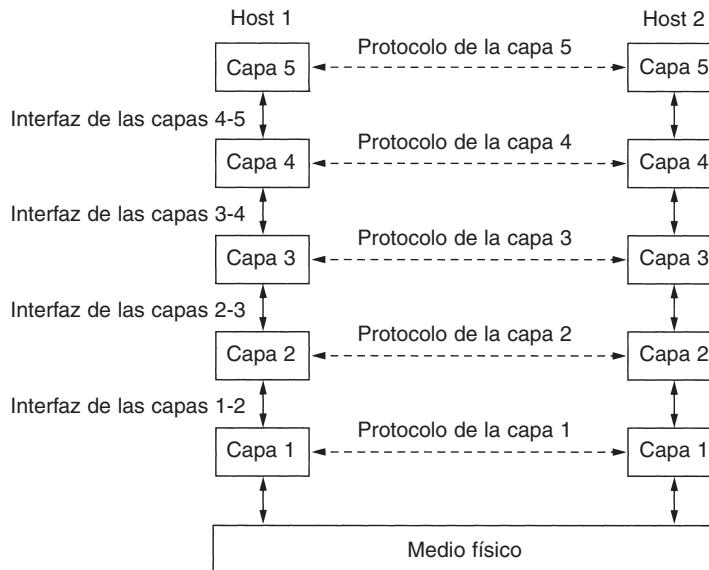


Figura 1-13. Capas, protocolos e interfaces.

En realidad, los datos no se transfieren de manera directa desde la capa n de una máquina a la capa n de la otra máquina, sino que cada capa pasa los datos y la información de control a la capa inmediatamente inferior, hasta que se alcanza la capa más baja. Debajo de la capa 1 se encuentra el **medio físico** a través del cual ocurre la comunicación real. En la figura 1-13, la comunicación virtual se muestra con líneas punteadas, en tanto que la física, con líneas sólidas.

Entre cada par de capas adyacentes está una **interfaz**. Ésta define qué operaciones y servicios primitivos pone la capa más baja a disposición de la capa superior inmediata. Cuando los diseñadores de redes deciden cuántas capas incluir en una red y qué debe hacer cada una, una de las consideraciones más importantes es definir interfaces limpias entre las capas. Hacerlo así, a su vez, requiere que la capa desempeñe un conjunto específico de funciones bien entendidas. Además de minimizar la cantidad de información que se debe pasar entre las capas, las interfaces bien definidas simplifican el reemplazo de la implementación de una capa con una implementación totalmente diferente (por ejemplo, todas las líneas telefónicas se reemplazan con canales por satélite)

porque todo lo que se pide de la nueva implementación es que ofrezca exactamente el mismo conjunto de servicios a su vecino de arriba, como lo hacía la implementación anterior. De hecho, es muy común que diferentes *hosts* utilicen diferentes implementaciones.

Un conjunto de capas y protocolos se conoce como **arquitectura de red**. La especificación de una arquitectura debe contener información suficiente para permitir que un implementador escriba el programa o construya el hardware para cada capa de modo que se cumpla correctamente con el protocolo apropiado. Ni los detalles de la implementación ni las especificaciones de las interfaces son parte de la arquitectura porque están ocultas en el interior de las máquinas y no son visibles desde el exterior. Incluso, tampoco es necesario que las interfaces de todas las máquinas en una red sean las mismas, siempre y cuando cada máquina pueda utilizar correctamente todos los protocolos. La lista de protocolos utilizados por un sistema, un protocolo por capa, se conoce como **pila de protocolos**. Los aspectos de las arquitecturas de red, las pilas de protocolos y los protocolos mismos son el tema principal de este libro.

Una analogía podría ayudar a explicar la idea de comunicación entre múltiples capas. Imagine a dos filósofos (procesos de iguales en la capa 3), uno de los cuales habla urdu e inglés, y el otro chino y francés. Puesto que no tienen un idioma común, cada uno contrata un traductor (proceso de iguales en la capa 2) y cada uno a su vez contacta a una secretaria (procesos de iguales en la capa 1). El filósofo 1 desea comunicar su afición por el *oryctolagus cuniculus* a su igual. Para eso, le pasa un mensaje (en inglés) a través de la interfaz de las capas 2-3 a su traductor, diciendo: “Me gustan los conejos”, como se ilustra en la figura 1-14. Los traductores han acordado un idioma neutral conocido por ambos, el holandés, para que el mensaje se convierta en “Ik vind konijnen leuk”. La elección del idioma es el protocolo de la capa 2 y los procesos de iguales de dicha capa son quienes deben realizarla.

Entonces el traductor le da el mensaje a una secretaria para que lo transmita por, digamos, fax (el protocolo de la capa 1). Cuando el mensaje llega, se traduce al francés y se pasa al filósofo 2 a través de la interfaz de las capas 2-3. Observe que cada protocolo es totalmente independiente de los demás en tanto no cambien las interfaces. Los traductores pueden cambiar de holandés a, digamos, finlandés, a voluntad, siempre y cuando los dos estén de acuerdo y no cambien su interfaz con las capas 1 o 3. Del mismo modo, las secretarias pueden cambiar de fax a correo electrónico o teléfono sin molestar (o incluso avisar) a las demás capas. Cada proceso podría agregar alguna información destinada sólo a su igual. Esta información no se pasa a la capa superior.

Ahora veamos un ejemplo más técnico: cómo proporcionar comunicación a la capa superior de la red de cinco capas de la figura 1-15. Un proceso de aplicación que se ejecuta en la capa 5 produce un mensaje, *M*, y lo pasa a la capa 4 para su transmisión.

La capa 4 pone un **encabezado** al frente del mensaje para identificarlo y pasa el resultado a la capa 3. El encabezado incluye información de control, como números de secuencia, para que la capa 4 de la máquina de destino entregue los mensajes en el orden correcto si las capas inferiores no mantienen la secuencia. En algunas capas los encabezados también pueden contener tamaños, medidas y otros campos de control.

En muchas redes no hay límites para el tamaño de mensajes transmitidos en el protocolo de la capa 4, pero casi siempre hay un límite impuesto por el protocolo de la capa 3. En consecuencia,

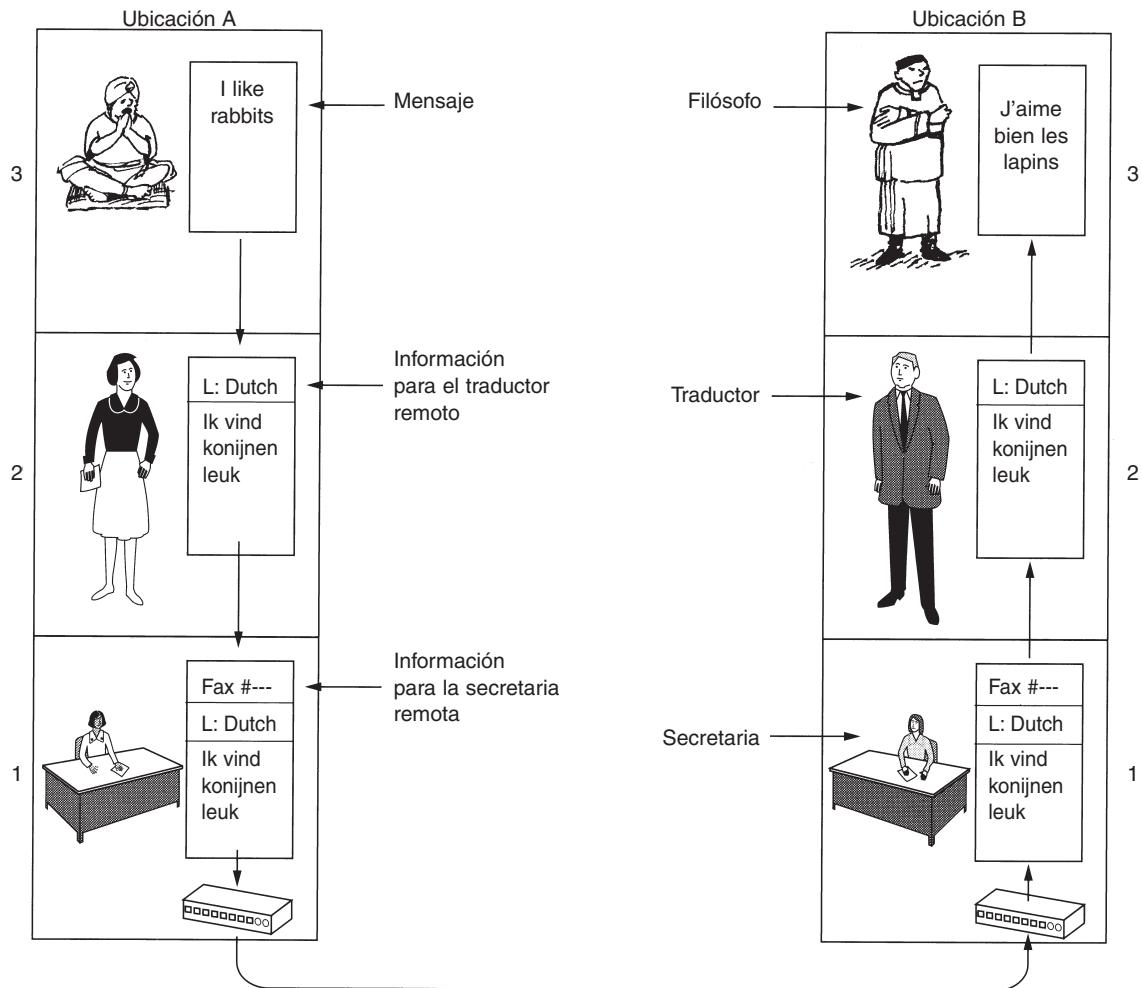


Figura 1-14. Arquitectura filósofo-traductor-secretaria.

la capa 3 debe desintegrar en unidades más pequeñas, paquetes, los mensajes que llegan, y a cada paquete le coloca un encabezado. En este ejemplo, M se divide en dos partes, M_1 y M_2 .

La capa 3 decide cuál de las líneas que salen utilizar y pasa los paquetes a la capa 2. Ésta no sólo agrega un encabezado a cada pieza, sino también un terminador, y pasa la unidad resultante a la capa 1 para su transmisión física. En la máquina receptora el mensaje pasa hacia arriba de capa en capa, perdiendo los encabezados conforme avanza. Ninguno de los encabezados de las capas inferiores a n llega a la capa n .

Lo que debe entender en la figura 1-15 es la relación entre las comunicaciones virtual y real, y la diferencia entre protocolos e interfaces. Por ejemplo, los procesos de iguales en la capa 4 piensan

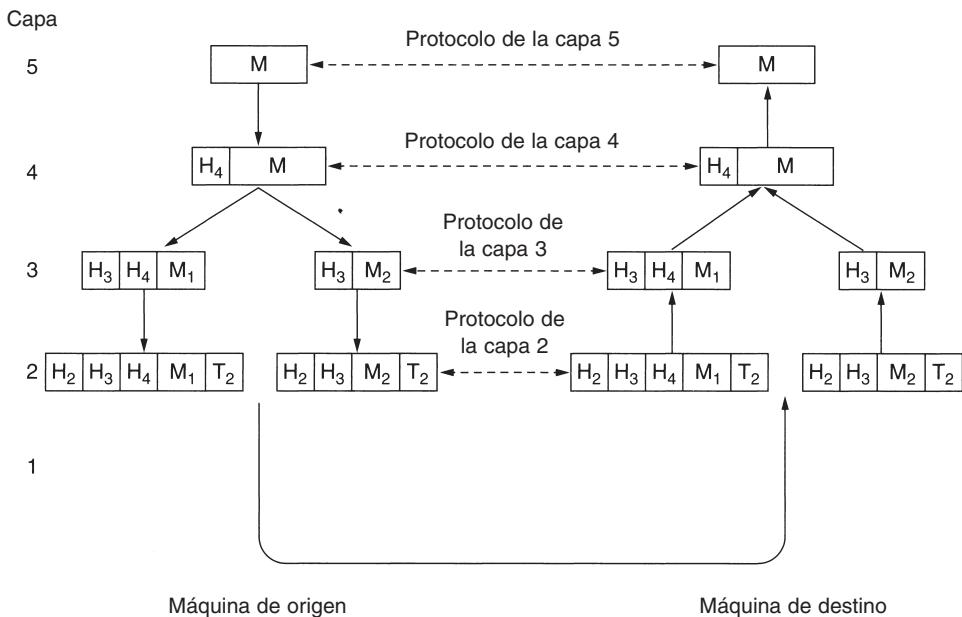


Figura 1-15. Ejemplo de flujo de información que soporta una comunicación virtual en la capa 5.

conceptualmente de su comunicación como si fuera “horizontal”, y utilizan el protocolo de la capa 4. Pareciera que cada uno tuviera un procedimiento llamado algo así como *EnviadoalOtroLado* y *RecibidoDesdeElOtroLado*, aun cuando estos procedimientos en realidad se comunican con las capas inferiores a través de la interfaz de las capas 3-4, no con el otro lado.

La abstracción del proceso de iguales es básica para todo diseño de red. Al utilizarla, la inmanejable tarea de diseñar toda la red se puede fragmentar en varios problemas de diseño más pequeños y manejables, es decir, el diseño de las capas individuales.

Aunque la sección 1.3 se llama “Software de redes”, vale la pena precisar que las capas inferiores de una jerarquía de protocolos se implementan con frecuencia en el hardware o en el firmware. No obstante, están implicados los algoritmos de protocolo complejos, aun cuando estén integrados (en todo o en parte) en el hardware.

1.3.2 Aspectos de diseño de las capas

Algunos de los aspectos clave de diseño que ocurren en las redes de computadoras están presentes en las diversas capas. Más adelante mencionaremos brevemente algunos de los más importantes.

Cada capa necesita un mecanismo para identificar a los emisores y a los receptores. Puesto que una red por lo general tiene muchas computadoras —algunas de las cuales tienen varios procesos—, se necesita un método para que un proceso en una máquina especifique con cuál de ellas

quiere hablar. Como consecuencia de tener múltiples destinos, se necesita alguna forma de **direcciónamiento** a fin de precisar un destino específico.

Otro conjunto de decisiones de diseño concierne a las reglas de la transferencia de datos. En algunos sistemas, los datos viajan sólo en una dirección; en otros, pueden viajar en ambas direcciones. El protocolo también debe determinar a cuántos canales lógicos corresponde la conexión y cuáles son sus prioridades. Muchas redes proporcionan al menos dos canales lógicos por conexión, uno para los datos normales y otro para los urgentes.

El **control de errores** es un aspecto importante porque los circuitos de comunicación física no son perfectos. Muchos códigos de detección y corrección de errores son conocidos, pero los dos extremos de la conexión deben estar de acuerdo en cuál es el que se va a utilizar. Además, el receptor debe tener algún medio de decirle al emisor qué mensajes se han recibido correctamente y cuáles no.

No todos los canales de comunicación conservan el orden en que se les envían los mensajes. Para tratar con una posible pérdida de secuencia, el protocolo debe incluir un mecanismo que permita al receptor volver a unir los pedazos en forma adecuada. Una solución obvia es numerar las piezas, pero esta solución deja abierta la cuestión de qué se debe hacer con las piezas que llegan sin orden.

Un aspecto que ocurre en cada nivel es cómo evitar que un emisor rápido sature de datos a un receptor más lento. Se han propuesto varias soluciones que explicaremos más adelante. Algunas de ellas implican algún tipo de retroalimentación del receptor al emisor, directa o indirectamente, dependiendo de la situación actual del receptor. Otros limitan al emisor a una velocidad de transmisión acordada. Este aspecto se conoce como **control de flujo**.

Otro problema que se debe resolver en algunos niveles es la incapacidad de todos los procesos de aceptar de manera arbitraria mensajes largos. Esta propiedad conduce a mecanismos para desensamblar, transmitir y reensamblar mensajes. Un aspecto relacionado es el problema de qué hacer cuando los procesos insisten en transmitir datos en unidades tan pequeñas que enviarlas por separado es ineficaz. La solución a esto es reunir en un solo mensaje grande varios mensajes pequeños que vayan dirigidos a un destino común y desmembrar dicho mensaje una vez que llegue a su destino.

Cuando es inconveniente o costoso establecer una conexión separada para cada par de procesos de comunicación, la capa subyacente podría decidir utilizar la misma conexión para múltiples conversaciones sin relación entre sí. Siempre y cuando esta **multiplexión** y **desmultiplexión** se realice de manera transparente, cualquier capa la podrá utilizar. La multiplexión se necesita en la capa física, por ejemplo, donde múltiples conversaciones comparten un número limitado de circuitos físicos. Cuando hay múltiples rutas entre el origen y el destino, se debe elegir la mejor o las mejores entre todas ellas. A veces esta decisión se debe dividir en dos o más capas. Por ejemplo, para enviar datos de Londres a Roma, se debe tomar una decisión de alto nivel para pasar por Francia o Alemania, dependiendo de sus respectivas leyes de privacidad. Luego se debe tomar una decisión de bajo nivel para seleccionar uno de los circuitos disponibles dependiendo de la carga de tráfico actual. Este tema se llama **enrutamiento**.

1.3.3 Servicios orientados a la conexión y no orientados a la conexión

Las capas pueden ofrecer dos tipos de servicios a las capas que están sobre ellas: orientados a la conexión y no orientados a la conexión. En esta sección veremos estos dos tipos y examinaremos las diferencias que hay entre ellos.

El **servicio orientado a la conexión** se concibió con base en el sistema telefónico. Para hablar con alguien, usted levanta el teléfono, marca el número, habla y luego cuelga. Del mismo modo, para usar un servicio de red orientado a la conexión, el usuario del servicio primero establece una conexión, la utiliza y luego la abandona. El aspecto esencial de una conexión es que funciona como un tubo: el emisor empuja objetos (bits) en un extremo y el receptor los toma en el otro extremo. En la mayoría de los casos se conserva el orden para que los bits lleguen en el orden en que se enviaron.

En algunos casos, al establecer la conexión, el emisor, el receptor y la subred realizan una **negociación** sobre los parámetros que se van a utilizar, como el tamaño máximo del mensaje, la calidad del servicio solicitado y otros temas. Por lo general, un lado hace una propuesta y el otro la acepta, la rechaza o hace una contrapropuesta.

En contraste, el **servicio no orientado a la conexión** se concibió con base en el sistema postal. Cada mensaje (carta) lleva completa la dirección de destino y cada una se enruta a través del sistema, independientemente de las demás. En general, cuando se envían dos mensajes al mismo destino, el primero que se envíe será el primero en llegar. Sin embargo, es posible que el que se envió primero se dilate tanto que el segundo llegue primero.

Cada servicio se puede clasificar por la **calidad del servicio**. Algunos servicios son confiables en el sentido de que nunca pierden datos. Por lo general, en un servicio confiable el receptor confirma la recepción de cada mensaje para que el emisor esté seguro de que llegó. Este proceso de confirmación de recepción introduce sobrecargas y retardos, que con frecuencia son valiosos pero a veces son indeseables.

Una situación típica en la que un servicio orientado a la conexión es apropiado es en la transferencia de archivos. El propietario del archivo desea estar seguro de que lleguen correctamente todos los bits y en el mismo orden en que se enviaron. Muy pocos clientes que transfieren archivos preferirían un servicio que revuelve o pierde ocasionalmente algunos bits, aunque fuera mucho más rápido.

Un servicio orientado a la conexión confiable tiene dos variantes menores: secuencias de mensaje y flujo de bytes. En la primera variante se conservan los límites del mensaje. Cuando se envían dos mensajes de 1024 bytes, llegan en dos mensajes distintos de 1024 bytes, nunca en un solo mensaje de 2048 bytes. En la segunda, la conexión es simplemente un flujo de bytes, sin límites en el mensaje. Cuando llegan los 2048 bytes al receptor, no hay manera de saber si se enviaron como un mensaje de 2048 bytes o dos mensajes de 1024 bytes o 2048 mensajes de un byte. Si se envían las páginas de un libro en mensajes separados sobre una red a una fotocomponedora, podría ser importante que se conserven los límites de los mensajes. Por otra parte, cuando un usuario inicia sesión en un servidor remoto, todo lo que se necesita es un flujo de bytes desde la computadora del usuario al servidor. Los límites del mensaje no son importantes.

Como lo mencionamos antes, para algunas aplicaciones, los retardos de tránsito ocasionados por las confirmaciones de recepción son inaceptables. Una de estas aplicaciones es el tráfico de voz digitalizada. Es preferible para los usuarios de teléfono escuchar un poco de ruido en la línea de vez en cuando que experimentar un retardo esperando las confirmaciones de recepción. Del mismo modo, tener algunos píxeles erróneos cuando se transmite una videoconferencia no es problema, pero experimentar sacudidas en la imagen cuando se interrumpe el flujo para corregir errores es muy molesto.

No todas las aplicaciones requieren conexiones. Por ejemplo, conforme el correo electrónico se vuelve más común, la basura electrónica también se torna más común. Es probable que el emisor de correo electrónico basura no desee enfrentarse al problema de configurar una conexión y luego desarmarla sólo para enviar un elemento. Tampoco es 100 por ciento confiable enviar lo esencial, sobre todo si eso es más costoso. Todo lo que se necesita es una forma de enviar un mensaje único que tenga una alta, aunque no garantizada, probabilidad de llegar. Al servicio no orientado a la conexión no confiable (es decir, sin confirmación de recepción) se le conoce como **servicio de datagramas**, en analogía con el servicio de telegramas, que tampoco devuelve una confirmación de recepción al emisor.

En otras situaciones se desea la conveniencia de no tener que establecer una conexión para enviar un mensaje corto, pero la confiabilidad es esencial. Para estas aplicaciones se puede proporcionar el **servicio de datagramas confirmados**. Es como enviar una carta certificada y solicitar una confirmación de recepción. Cuando ésta regresa, el emisor está absolutamente seguro de que la carta se ha entregado a la parte destinada y no se ha perdido durante el trayecto.

Otro servicio más es el de **solicitud-respuesta**. En este servicio el emisor transmite un solo datagrama que contiene una solicitud; a continuación el servidor envía la respuesta. Por ejemplo, una solicitud a la biblioteca local preguntando dónde se habla uighur cae dentro de esta categoría. El esquema de solicitud-respuesta se usa comúnmente para implementar la comunicación en el modelo cliente-servidor: el cliente emite una solicitud y el servidor la responde. La figura 1-16 resume los tipos de servicios que se acaban de exponer.

	Servicio	Ejemplo
Orientado a la conexión	Flujo confiable de mensajes	Secuencia de páginas
	Flujo confiable de bytes	Inicio de sesión remoto
No orientado a la conexión	Conexión no confiable	Voz digitalizada
	Datagrama no confiable	Correo electrónico basura
	Datagrama confirmado	Correo certificado
	Solicitud-respuesta	Consulta de base de datos

Figura 1-16. Seis tipos de servicio diferentes.

El concepto del uso de la comunicación no confiable podría ser confuso al principio. Después de todo, en realidad, ¿por qué preferiría alguien la comunicación no confiable a la comunicación

confiable? Antes que nada, la comunicación confiable (en nuestro sentido, es decir, con confirmación de la recepción) podría no estar disponible. Por ejemplo, Ethernet no proporciona comunicación confiable. Ocasionalmente, los paquetes se pueden dañar en el tránsito. Toca al protocolo más alto enfrentar este problema. En segundo lugar, los retardos inherentes al servicio confiable podrían ser inaceptables, en particular para aplicaciones en tiempo real como multimedia. Éstas son las razones de que coexistan la comunicación no confiable y la confiable.

1.3.4 Primitivas de servicio

Un servicio se especifica formalmente como un conjunto de **primitivas** (operaciones) disponibles a un proceso de usuario para que acceda al servicio. Estas primitivas le indican al servicio que desempeñe alguna acción o reporte sobre una acción que ha tomado una entidad igual. Si la pila de protocolos se ubica en el sistema operativo, como suele suceder, por lo general las primitivas son llamadas al sistema. Estas llamadas provocan un salto al modo de *kernel*, que entonces cede el control de la máquina al sistema operativo para enviar los paquetes necesarios.

El conjunto de primitivas disponible depende de la naturaleza del servicio que se va a proporcionar. Las primitivas de servicio orientado a la conexión son diferentes de las del servicio no orientado a la conexión. Como un ejemplo mínimo de las primitivas para servicio que se podrían proporcionar para implementar un flujo de bytes confiable en un ambiente cliente-servidor, considere las primitivas listadas en la figura 1-17.

Primitiva	Significado
LISTEN	Bloquea en espera de una conexión entrante
CONNECT	Establece una conexión con el igual en espera
RECEIVE	Bloquea en espera de un mensaje entrante
SEND	Envía un mensaje al igual
DISCONNECT	Da por terminada una conexión

Figura 1-17. Cinco primitivas de servicio para la implementación de un servicio simple orientado a la conexión.

Estas primitivas se podrían usar como sigue. En primer lugar, el servidor ejecuta LISTEN para indicar que está preparado para aceptar las conexiones entrantes. Una manera común de implementar LISTEN es hacer que bloquee la llamada al sistema. Después de ejecutar la primitiva, el proceso del servidor se bloquea hasta que aparece una solicitud de conexión.

A continuación, el proceso del cliente ejecuta CONNECT para establecer una conexión con el servidor. La llamada CONNECT necesita especificar a quién conecta con quién, así que podría tener un parámetro que diera la dirección del servidor. El sistema operativo, en general, envía un paquete al igual solicitándole que se conecte, como se muestra en (1) en la figura 1-18. El proceso del cliente se suspende hasta que haya una respuesta. Cuando el paquete llega al servidor, es procesado ahí por el sistema operativo. Cuando el sistema ve que el paquete es una solicitud de

conexión, verifica si hay un escuchador. En ese caso hace dos cosas: desbloquea al escuchador y envía de vuelta una confirmación de recepción (2). La llegada de esta confirmación libera entonces al cliente. En este punto tanto el cliente como el servidor están en ejecución y tienen establecida una conexión. Es importante observar que la confirmación de recepción (2) es generada por el código del protocolo mismo, no en respuesta a una primitiva al nivel de usuario. Si llega una solicitud de conexión y no hay un escuchador, el resultado es indefinido. En algunos sistemas el paquete podría ser puesto en cola durante un breve tiempo en espera de un LISTEN.

La analogía obvia entre este protocolo y la vida real es un consumidor (cliente) que llama al gerente de servicios a clientes de una empresa. El gerente de servicios empieza por estar cerca del teléfono en caso de que éste suene. Entonces el cliente hace la llamada. Cuando el gerente levanta el teléfono se establece la conexión.

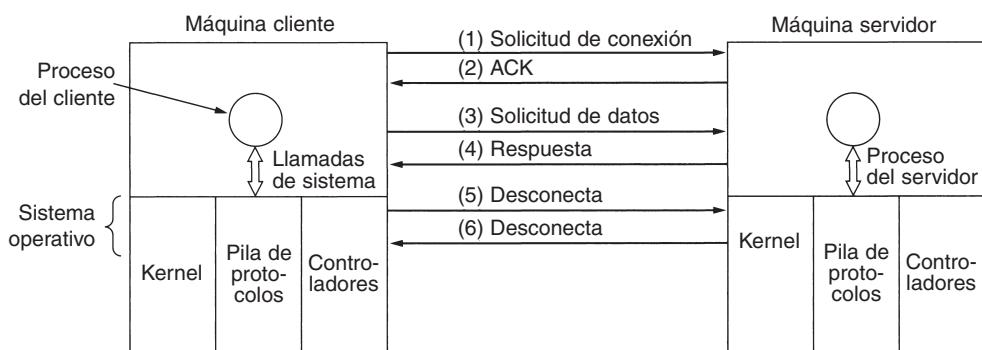


Figura 1-18. Paquetes enviados en una interacción simple cliente-servidor sobre una red orientada a la conexión.

El paso siguiente es que el servidor ejecute RECEIVE para prepararse para aceptar la primera solicitud. Normalmente, el servidor hace esto de inmediato en cuanto está libre de LISTEN, antes de que la confirmación de recepción pueda volver al cliente. La llamada RECEIVE bloquea al servidor.

Entonces el cliente ejecuta SEND para transmitir sus solicitudes (3) seguidas de la ejecución de RECEIVE para obtener la respuesta.

La llegada del paquete de solicitud a la máquina servidor desbloquea el proceso del servidor para que pueda procesar la solicitud. Una vez hecho su trabajo, utiliza SEND para devolver la respuesta al cliente (4). La llegada de este paquete desbloquea al cliente, que ahora puede revisar la respuesta. Si el cliente tiene solicitudes adicionales las puede hacer ahora. Si ha terminado, puede utilizar DISCONNECT para finalizar la conexión. Por lo común, un DISCONNECT inicial es una llamada de bloqueo, que suspende al cliente y envía un paquete al servidor en el cual le indica que ya no es necesaria la conexión (5). Cuando el servidor recibe el paquete también emite un DISCONNECT, enviando la confirmación de recepción al cliente y terminando la conexión. Cuando el paquete del servidor (6) llega a la máquina cliente, el proceso del cliente se libera y finaliza la conexión. En pocas palabras, ésta es la manera en que funciona la comunicación orientada a la conexión.

Desde luego, no todo es tan sencillo. Hay muchas cosas que pueden fallar. La temporización puede estar mal (por ejemplo, CONNECT se hace antes de LISTEN), se pueden perder paquetes, etcétera. Más adelante veremos en detalle estos temas, pero por el momento la figura 1-18 resume cómo podría funcionar la comunicación cliente-servidor en una red orientada a la conexión.

Dado que se requieren seis paquetes para completar este protocolo, cabría preguntarse por qué no se usa en su lugar un protocolo no orientado a la conexión. La respuesta es que en un mundo perfecto podría utilizarse, en cuyo caso bastaría dos paquetes: uno para la solicitud y otro para la respuesta. Sin embargo, en el caso de mensajes grandes en cualquier dirección (por ejemplo, en un archivo de megabytes), errores de transmisión y paquetes perdidos, la situación cambia. Si la respuesta constara de cientos de paquetes, algunos de los cuales se podrían perder durante la transmisión, ¿cómo sabría el cliente si se han perdido algunas piezas? ¿Cómo podría saber que el último paquete que recibió fue realmente el último que se envió? Suponga que el cliente esperaba un segundo archivo. ¿Cómo podría saber que el paquete 1 del segundo archivo de un paquete 1 perdido del primer archivo que de pronto apareció va en camino al cliente? Para abreviar, en el mundo real un simple protocolo de solicitud-respuesta en una red no confiable suele ser inadecuado. En el capítulo 3 estudiaremos en detalle una variedad de protocolos que soluciona éstos y otros problemas. Por el momento, baste decir que a veces es muy conveniente tener un flujo de bytes ordenado y confiable entre procesos.

1.3.5 Relación de servicios a protocolos

Servicios y protocolos son conceptos distintos, aunque con frecuencia se confunden. Sin embargo, esta distinción es tan importante que por esa razón ponemos énfasis de nuevo en ese punto. Un *servicio* es un conjunto de primitivas (operaciones) que una capa proporciona a la capa que está sobre ella. El servicio define qué operaciones puede realizar la capa en beneficio de sus usuarios, pero no dice nada de cómo se implementan tales operaciones. Un servicio está relacionado con la interfaz entre dos capas, donde la capa inferior es la que provee el servicio y la superior, quien lo recibe.

Un *protocolo*, en contraste, es un conjunto de reglas que rigen el formato y el significado de los paquetes, o mensajes, que se intercambiaron las entidades iguales en una capa. Las entidades utilizan protocolos para implementar sus definiciones del servicio. Son libres de cambiar sus protocolos cuando lo deseen, siempre y cuando no cambie el servicio visible a sus usuarios. De esta manera, el servicio y el protocolo no dependen uno del otro.

En otras palabras, los servicios se relacionan con las interacciones entre capas, como se ilustra en la figura 1-19. En contraste, los protocolos se relacionan con los paquetes enviados entre entidades iguales de máquinas diferentes. Es importante no confundir estos dos conceptos.

Vale la pena hacer una analogía con los lenguajes de programación. Un servicio es como un tipo de datos abstractos o un objeto en un lenguaje orientado a objetos. Define operaciones que se deben realizar en un objeto pero no especifica cómo se implementan estas operaciones. Un protocolo se relaciona con la *implementación* del servicio y, como tal, el usuario del servicio no puede verlo.

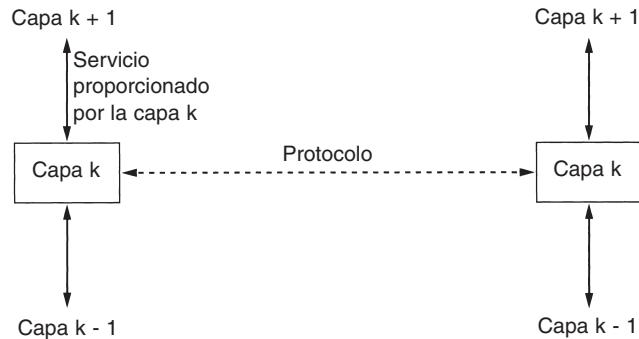


Figura 1-19. La relación entre un servicio y un protocolo.

Muchos protocolos antiguos no distinguían el servicio del protocolo. En efecto, una capa típica podría haber tenido una primitiva de servicio SEND PACKET y el usuario proveía un apuntador a un paquete ensamblado totalmente. Este arreglo significa que el usuario podía ver de inmediato todos los cambios del protocolo. En la actualidad, la mayoría de los diseñadores de redes señalan a este tipo de diseño como un error grave.

1.4 MODELOS DE REFERENCIA

Ahora que hemos visto en teoría las redes con capas, es hora de ver algunos ejemplos. En las dos secciones siguientes veremos dos arquitecturas de redes importantes: los modelos de referencia OSI y TCP/IP. Aunque los *protocolos* asociados con el modelo OSI ya casi no se usan, el *modelo* en sí es muy general y aún es válido, y las características tratadas en cada capa aún son muy importantes. El modelo TCP/IP tiene las propiedades opuestas: el modelo en sí no se utiliza mucho pero los protocolos sí. Por estas razones analizaremos con detalle ambos modelos. Además, a veces podemos aprender más de las fallas que de los aciertos.

1.4.1 El modelo de referencia OSI

El modelo OSI se muestra en la figura 1-20 (sin el medio físico). Este modelo está basado en una propuesta desarrollada por la ISO (Organización Internacional de Estándares) como un primer paso hacia la estandarización internacional de los protocolos utilizados en varias capas (Day y Zimmermann, 1983). Fue revisado en 1995 (Day, 1995). El modelo se llama **OSI (Interconexión de Sistemas Abiertos)** de ISO porque tiene que ver con la conexión de sistemas abiertos, es decir, sistemas que están abiertos a la comunicación con otros sistemas. Para abreviar, lo llamaremos modelo OSI.

El modelo OSI tiene siete capas. Podemos resumir brevemente los principios que se aplicaron para llegar a dichas capas:

1. Una capa se debe crear donde se necesite una abstracción diferente.
2. Cada capa debe realizar una función bien definida.
3. La función de cada capa se debe elegir con la intención de definir protocolos estandarizados internacionalmente.
4. Los límites de las capas se deben elegir a fin de minimizar el flujo de información a través de las interfaces.
5. La cantidad de capas debe ser suficientemente grande para no tener que agrupar funciones distintas en la misma capa y lo bastante pequeña para que la arquitectura no se vuelva inmanejable.

A continuación analizaremos una por una cada capa del modelo, comenzando con la capa inferior. Observe que el modelo OSI no es en sí una arquitectura de red, debido a que no especifica los servicios y protocolos exactos que se utilizarán en cada capa. Sólo indica lo que debe hacer cada capa. Sin embargo, ISO también ha producido estándares para todas las capas, aunque éstos no son parte del modelo de referencia mismo. Cada uno se ha publicado como un estándar internacional separado.

La capa física

En esta capa se lleva a cabo la transmisión de bits puros a través de un canal de comunicación. Los aspectos del diseño implican asegurarse de que cuando un lado envía un bit 1, éste se reciba en el otro lado como tal, no como bit 0. Las preguntas típicas aquí son: ¿cuántos voltios se deben emplear para representar un 1 y cuántos para representar un 0?, ¿cuántos nanosegundos dura un bit?, ¿la transmisión se debe llevar a cabo en ambas direcciones al mismo tiempo?, ¿cómo se establece la conexión inicial y cómo se finaliza cuando ambos lados terminan?, ¿cuántos pines tiene un conector de red y para qué se utiliza cada uno? Los aspectos de diseño tienen que ver mucho con interfaces mecánicas, eléctricas y de temporización, además del medio físico de transmisión, que está bajo la capa física.

La capa de enlace de datos

La tarea principal de esta capa es transformar un medio de transmisión puro en una línea de comunicación que, al llegar a la capa de red, aparezca libre de errores de transmisión. Logra esta tarea haciendo que el emisor fragmente los datos de entrada en **tramas de datos** (típicamente, de algunos cientos o miles de bytes) y transmitiendo las tramas de manera secuencial. Si el servicio es confiable, el receptor confirma la recepción correcta de cada trama devolviendo una **trama de confirmación de recepción**.

Otra cuestión que surge en la capa de enlace de datos (y en la mayoría de las capas superiores) es cómo hacer que un transmisor rápido no sature de datos a un receptor lento. Por lo general se necesita un mecanismo de regulación de tráfico que indique al transmisor cuánto espacio de búfer tiene el receptor en ese momento. Con frecuencia, esta regulación de flujo y el manejo de errores están integrados.

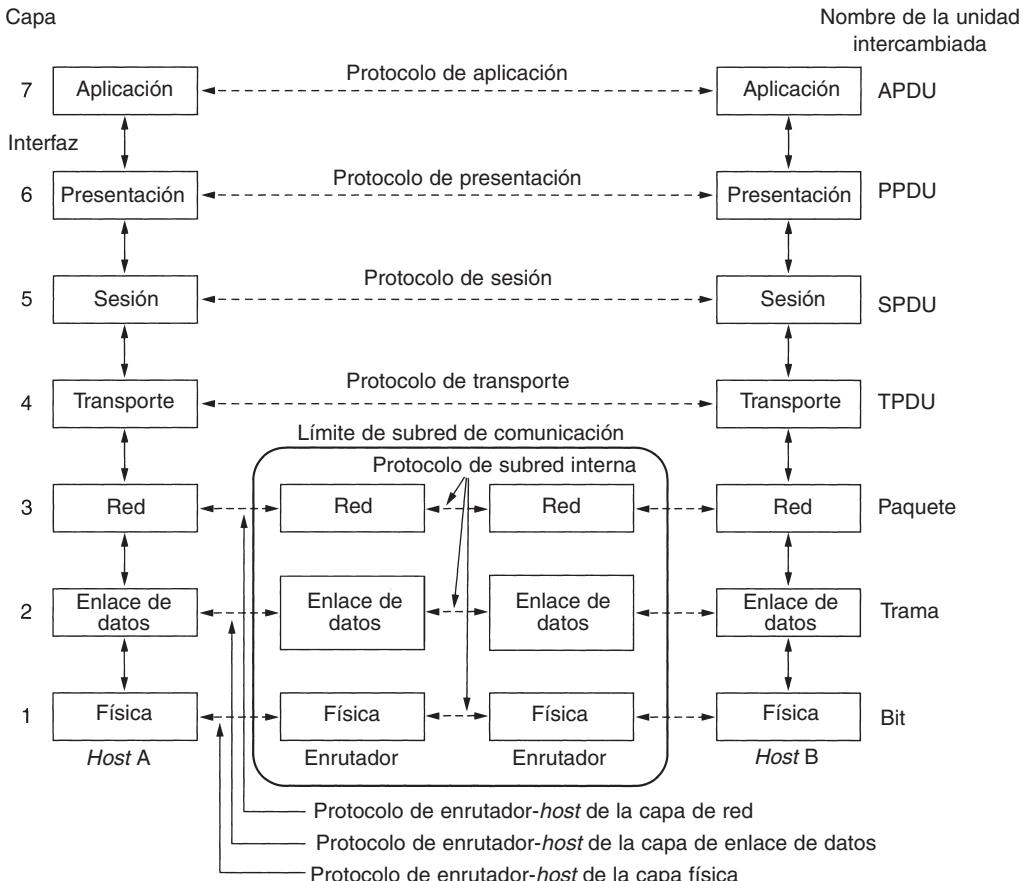


Figura 1-20. El modelo de referencia OSI.

Las redes de difusión tienen un aspecto adicional en la capa de enlace de datos: cómo controlar el acceso al canal compartido. Una subcapa especial de la capa de enlace de datos, la subcapa de control de acceso al medio, se encarga de este problema.*

La capa de red

Esta capa controla las operaciones de la subred. Un aspecto clave del diseño es determinar cómo se enrutan los paquetes desde su origen a su destino. Las rutas pueden estar basadas en tablas estáticas (enrutamiento estático) codificadas en la red y que rara vez cambian.**

*En esta capa se define el direccionamiento físico, que permite a los *hosts* identificar las tramas destinadas a ellos. Este direccionamiento es único, identifica el hardware de red que se está usando y el fabricante, y no se puede cambiar. (N. del R.T.)

**En el enrutamiento estático la ruta que seguirán los paquetes hacia un destino particular es determinada por el administrador de la red. Las rutas también pueden determinarse cuando los enrutadores intercambian información de enrutamiento (enrutamiento dinámico). En este tipo de enrutamiento los enrutadores deciden la ruta que seguirán los paquetes hacia un destino sin la intervención del administrador de red. En el enrutamiento dinámico las rutas pueden cambiar para reflejar la topología o el estado de la red. (N. del R.T.)

Si hay demasiados paquetes en la subred al mismo tiempo, se interpondrán en el camino unos y otros, lo que provocará que se formen cuellos de botella. La responsabilidad de controlar esta congestión también pertenece a la capa de red, aunque esta responsabilidad también puede ser compartida por la capa de transmisión. De manera más general, la calidad del servicio proporcionado (retardo, tiempo de tránsito, inestabilidad, etcétera) también corresponde a la capa de red.

Cuando un paquete tiene que viajar de una red a otra para llegar a su destino, pueden surgir muchos problemas. El direccionamiento utilizado por la segunda red podría ser diferente del de la primera.* La segunda podría no aceptar todo el paquete porque es demasiado largo. Los protocolos podrían ser diferentes, etcétera. La capa de red tiene que resolver todos estos problemas para que las redes heterogéneas se interconecten.

En las redes de difusión, el problema de enrutamiento es simple, por lo que la capa de red a veces es delgada o, en ocasiones, ni siquiera existe.

La capa de transporte

La función básica de esta capa es aceptar los datos provenientes de las capas superiores, dividirlos en unidades más pequeñas si es necesario, pasar éstas a la capa de red y asegurarse de que todas las piezas lleguen correctamente al otro extremo. Además, todo esto se debe hacer con eficiencia y de manera que aíslle a las capas superiores de los cambios inevitables en la tecnología del hardware.

La capa de transporte también determina qué tipo de servicio proporcionar a la capa de sesión y, finalmente, a los usuarios de la red. El tipo de conexión de transporte más popular es un canal punto a punto libre de errores que entrega mensajes o bytes en el orden en que se enviaron. Sin embargo, otros tipos de servicio de transporte posibles son la transportación de mensajes aislados, que no garantiza el orden de entrega, y la difusión de mensajes a múltiples destinos. El tipo de servicio se determina cuando se establece la conexión. (Como observación, es imposible alcanzar un canal libre de errores; lo que se quiere dar a entender con este término es que la tasa de error es tan baja que se puede ignorar en la práctica.)

La capa de transporte es una verdadera conexión de extremo a extremo, en toda la ruta desde el origen hasta el destino. En otras palabras, un programa en la máquina de origen lleva a cabo una conversación con un programa similar en la máquina de destino, usando los encabezados de mensaje y los mensajes de control. En las capas inferiores, los protocolos operan entre cada máquina y sus vecinos inmediatos, y no entre las máquinas de los extremos, la de origen y la de destino, las cuales podrían estar separadas por muchos enrutadores. En la figura 1-20 se muestra la diferencia entre las capas 1 a 3, que están encadenadas, y las capas 4 a 7, que operan de extremo a extremo.

La capa de sesión

Esta capa permite que los usuarios de máquinas diferentes establezcan **sesiones** entre ellos. Las sesiones ofrecen varios servicios, como el **control de diálogo** (dar seguimiento de a quién le toca

*El direccionamiento usado en esta capa es un direccionamiento lógico, diferente al direccionamiento físico empleado en la capa de enlace de datos. Este direccionamiento lógico permite que una interfaz o puerto pueda tener más de una dirección de capa de red. (N. del R.T.)

transmitir), **administración de token** (que impide que las dos partes traten de realizar la misma operación crítica al mismo tiempo) y **sincronización** (la adición de puntos de referencia a transmisiones largas para permitirles continuar desde donde se encontraban después de una caída).

La capa de presentación

A diferencia de las capas inferiores, a las que les corresponde principalmente mover bits, a la **capa de presentación** le corresponde la sintaxis y la semántica de la información transmitida. A fin de que las computadoras con diferentes representaciones de datos se puedan comunicar, las estructuras de datos que se intercambiarán se pueden definir de una manera abstracta, junto con una codificación estándar para su uso “en el cable”. La capa de presentación maneja estas estructuras de datos abstractas y permite definir e intercambiar estructuras de datos de un nivel más alto (por ejemplo, registros bancarios).

La capa de aplicación

Esta capa contiene varios protocolos que los usuarios requieren con frecuencia. Un protocolo de aplicación de amplio uso es **HTTP (Protocolo de Transferencia de Hipertexto)**, que es la base de World Wide Web. Cuando un navegador desea una página Web, utiliza este protocolo para enviar al servidor el nombre de dicha página. A continuación, el servidor devuelve la página. Otros protocolos de aplicación se utilizan para la transferencia de archivos, correo electrónico y noticias en la red.

1.4.2 El modelo de referencia TCP/IP

Tratemos ahora el modelo de referencia usado en la abuela de todas las redes de computadoras de área amplia, ARPANET, y en su sucesora, la Internet mundial. Aunque daremos más adelante una breve historia de ARPANET, es útil mencionar algunos de sus aspectos ahora. ARPANET fue una red de investigación respaldada por el DoD (Departamento de Defensa de Estados Unidos). Con el tiempo, conectó cientos de universidades e instalaciones gubernamentales mediante líneas telefónicas alquiladas. Posteriormente, cuando se agregaron redes satelitales y de radio, los protocolos existentes tuvieron problemas para interactuar con ellas, por lo que se necesitaba una nueva arquitectura de referencia. De este modo, la capacidad para conectar múltiples redes en una manera sólida fue una de las principales metas de diseño desde sus inicios. Más tarde, esta arquitectura se llegó a conocer como el **modelo de referencia TCP/IP**, de acuerdo con sus dos protocolos primarios. Su primera definición fue en (Cerf y Kahn, 1974). Posteriormente se definió en (Leiner y cols., 1985). La filosofía del diseño que respalda al modelo se explica en (Clark, 1988).

Ante el temor del DoD de que algunos de sus valiosos *hosts*, enrutadores y puertas de enlace de interredes explotaran en un instante, otro objetivo fue que la red pudiera sobrevivir a la pérdida de hardware de la subred, sin que las conversaciones existentes se interrumpieran. En otras palabras, el DoD quería que las conexiones se mantuvieran intactas en tanto las máquinas de origen

y destino estuvieran funcionando, aunque algunas de las máquinas o líneas de transmisión intermedias quedaran fuera de operación repentinamente. Además, se necesitaba una arquitectura flexible debido a que se preveían aplicaciones con requerimientos divergentes, desde transferencia de archivos a transmisión de palabras en tiempo real.

La capa de interred

Todos estos requerimientos condujeron a la elección de una red de conmutación de paquetes basada en una capa de interred no orientada a la conexión. Esta capa, llamada **capa de interred**, es la pieza clave que mantiene unida a la arquitectura. Su trabajo es permitir que los *hosts* injecten paquetes dentro de cualquier red y que éstos viajen a su destino de manera independiente (podría ser en una red diferente). Tal vez lleguen en un orden diferente al que fueron enviados, en cuyo caso las capas más altas deberán ordenarlos, si se desea una entrega ordenada. Observe que aquí el concepto “interred” se utiliza en un sentido genérico, aun cuando esta capa se presente en Internet.

Aquí la analogía es con el sistema de correo tradicional. Una persona puede depositar una secuencia de cartas internacionales en un buzón y, con un poco de suerte, la mayoría de ellas se entregarán en la dirección correcta del país de destino. Es probable que durante el trayecto, las cartas viajen a través de una o más puertas de enlace de correo internacional, pero esto es transparente para los usuarios. Además, para los usuarios también es transparente el hecho de que cada país (es decir, cada red) tiene sus propios timbres postales, tamaños preferidos de sobre y reglas de entrega.

La capa de interred define un paquete de formato y protocolo oficial llamado **IP (Protocolo de Internet)**. El trabajo de la capa de interred es entregar paquetes IP al destinatario. Aquí, el enrutamiento de paquetes es claramente el aspecto principal, con el propósito de evitar la congestión. Por estas razones es razonable decir que la capa de interred del modelo TCP/IP es similar en funcionalidad a la capa de red del modelo OSI. La figura 1-21 muestra esta correspondencia.

La capa de transporte

La capa que está arriba de la capa de interred en el modelo TCP/IP se llama **capa de transporte**. Está diseñada para permitir que las entidades iguales en los *hosts* de origen y destino puedan llevar a cabo una conversación, tal como lo hace la capa de transporte OSI. Aquí se han definido dos protocolos de transporte de extremo a extremo. El primero, **TCP (Protocolo de Control de Transmisión)**, es un protocolo confiable, orientado a la conexión, que permite que un flujo de bytes que se origina en una máquina se entregue sin errores en cualquier otra máquina en la interred. Divide el flujo de bytes entrantes en mensajes discretos y pasa cada uno de ellos a la capa de interred. En el destino, el proceso TCP receptor reensambla en el flujo de salida los mensajes recibidos. TCP también maneja el control de flujo para asegurarse de que un emisor rápido no sature a un receptor lento con más mensajes de los que puede manejar.

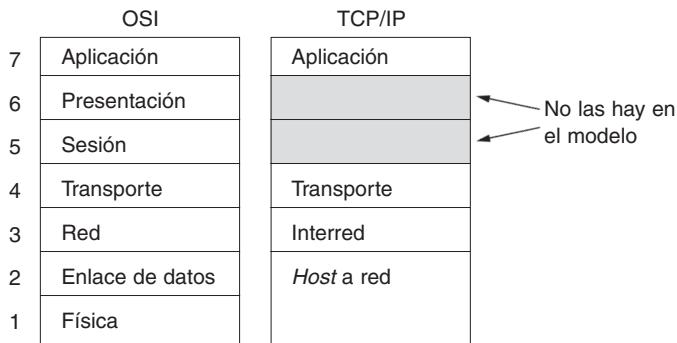


Figura 1-21. El modelo de referencia TCP/IP.

El segundo protocolo de esta capa, **UDP (Protocolo de Datagrama de Usuario)**, es un protocolo no confiable y no orientado a la conexión para aplicaciones que no desean la secuenciación o el control de flujo de TCP y que desean proporcionar el suyo. También tiene un amplio uso en consultas únicas de solicitud-respuesta de tipo cliente-servidor en un solo envío, así como aplicaciones en las que la entrega puntual es más importante que la precisa, como en la transmisión de voz o vídeo. La relación de IP, TCP y UDP se muestra en la figura 1-22. Puesto que el modelo se desarrolló, se ha implementado IP en muchas otras redes.

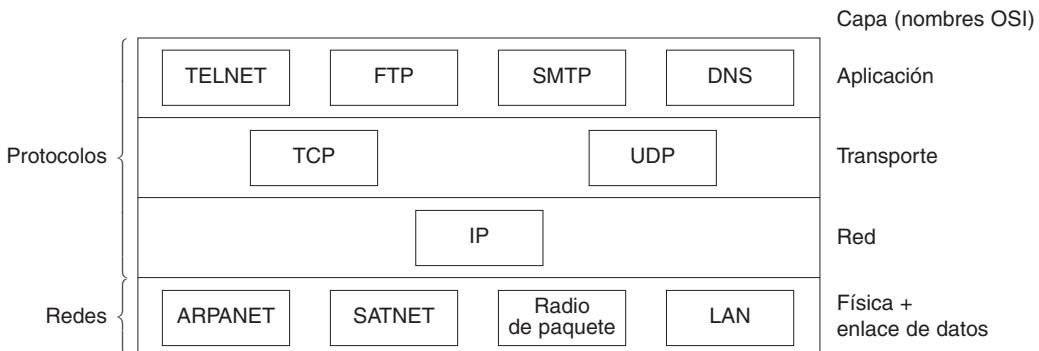


Figura 1-22. Protocolos y redes en el modelo TCP/IP inicialmente.

La capa de aplicación

El modelo TCP/IP no tiene capas de sesión ni de presentación. No se han necesitado, por lo que no se incluyen. La experiencia con el modelo OSI ha probado que este punto de vista es correcto: son de poco uso para la mayoría de las aplicaciones.

Arriba de la capa de transporte está la **capa de aplicación**. Contiene todos los protocolos de nivel más alto. Los primeros incluyeron una terminal virtual (TELNET), transferencia de archivos (FTP) y correo electrónico (SMTP), como se muestra en la figura 1-22. El protocolo de terminal virtual permite que un usuario en una máquina se registre en una máquina remota y trabaje ahí. El protocolo de transferencia de archivos proporciona una manera de mover con eficiencia datos de una máquina a otra. El correo electrónico era originalmente sólo un tipo de transferencia de archivos, pero más tarde se desarrolló un protocolo especializado (SMTP) para él. Con el tiempo, se han agregado muchos otros protocolos: DNS (Sistema de Nombres de Dominio) para la resolución de nombres de *host* en sus direcciones de red; NNTP, para transportar los artículos de noticias de USENET; HTTP, para las páginas de World Wide Web, y muchos otros.

La capa *host* a red

Deabajo de la capa de interred hay un gran vacío. El modelo de referencia TCP/IP en realidad no dice mucho acerca de lo que pasa aquí, excepto que puntualiza que el *host* se tiene que conectar a la red mediante el mismo protocolo para que le puedan enviar paquetes IP. Este protocolo no está definido y varía de un *host* a otro y de una red a otra. Este tema rara vez se trata en libros y artículos sobre TCP/IP.

1.4.3 Comparación entre los modelos de referencia OSI y TCP/IP

Los modelos de referencia OSI y TCP/IP tienen mucho en común. Los dos se basan en el concepto de una pila de protocolos independientes. Asimismo, la funcionalidad de las capas es muy parecida. Por ejemplo, en ambos modelos las capas que están arriba de, incluyendo a, la capa de transporte están ahí para proporcionar un servicio de transporte independiente de extremo a extremo a los procesos que desean comunicarse. Estas capas forman el proveedor de transporte. De nuevo, en ambos modelos, las capas que están arriba de la de transporte son usuarias orientadas a la aplicación del servicio de transporte.

A pesar de estas similitudes fundamentales, los dos modelos también tienen muchas diferencias. En esta sección nos enfocaremos en las diferencias clave entre estos dos modelos de referencia. Es importante tener en cuenta que estamos comparando los *modelos de referencia*, no las *pilas de protocolos* correspondientes. Más adelante explicaremos los protocolos. Si desea un libro dedicado a comparar y contrastar TCP/IP y OSI, vea (Piscitello y Chapin, 1993).

Tres conceptos son básicos para el modelo OSI:

1. Servicios.
2. Interfaces.
3. Protocolos.

Probablemente la contribución más grande del modelo OSI es que hace explícita la distinción entre estos tres conceptos. Cada capa desempeña algunos servicios para la capa que está arriba de ella. La definición de *servicio* indica qué hace la capa, no la forma en que la entidad superior tiene acceso a ella, o cómo funciona dicha capa. Define el aspecto semántico de la capa.

La *interfaz* de una capa indica a los procesos que están sobre ella cómo accederla. Especifica cuáles son los parámetros y qué resultados se esperan. Incluso, no dice nada sobre cómo funciona internamente la capa.

Por último, una capa es quien debe decidir qué *protocolos* de iguales utilizar. Puede usar cualesquier protocolos que desee, en tanto consiga que se haga el trabajo (es decir, proporcione los servicios ofrecidos). También puede cambiarlos cuando desee sin afectar el software de las capas superiores.

Estas ideas encajan muy bien con las ideas modernas sobre la programación orientada a objetos. Un objeto, como una capa, cuenta con un conjunto de métodos (operaciones) que pueden ser invocados por procesos que no estén en dicho objeto. La semántica de estos métodos define el conjunto de servicios que ofrece el objeto. Los parámetros y resultados de los métodos forman la interfaz del objeto. El código interno del objeto es su protocolo y no es visible o no tiene importancia fuera del objeto.

Originalmente, el modelo TCP/IP no distinguía entre servicio, interfaz y protocolo, aunque las personas han tratado de readaptarlo con el propósito de hacerlo más parecido al OSI. Por ejemplo, los únicos servicios ofrecidos realmente por la capa de interred son SEND IP PACKET y RECEIVE IP PACKET.

Como consecuencia, los protocolos del modelo OSI están mejor ocultos que los del modelo TCPI/IP y se pueden reemplazar fácilmente conforme cambia la tecnología. La facilidad para realizar tales cambios es uno de los objetivos principales de tener protocolos en capas.

El modelo de referencia OSI se vislumbró *antes* de que se inventaran los protocolos correspondientes. Esta clasificación significa que el modelo no estaba diseñado para un conjunto particular de protocolos, un hecho que lo hizo general. Una deficiencia de esta clasificación es que los diseñadores no tenían mucha experiencia con el asunto y no tenían una idea concreta de qué funcionalidad poner en qué capa.

Por ejemplo, originalmente la capa de enlace de datos sólo trataba con redes de punto a punto. Cuando llegaron las redes de difusión, se tuvo que extender una nueva subcapa en el modelo. Cuando las personas empezaron a construir redes reales utilizando el modelo OSI y los protocolos existentes, se descubrió que estas redes no coincidían con las especificaciones de los servicios solicitados (maravilla de maravillas), por lo que se tuvieron que integrar subcapas convergentes en el modelo para proporcionar un espacio para documentar las diferencias. Por último, el comité esperaba en un principio que cada país tuviera una red, controlada por el gobierno y que utilizara los protocolos OSI, pero nunca pensaron en la interconectividad de redes. Para no hacer tan larga la historia, las cosas no sucedieron como se esperaba.

Con TCP/IP sucedió lo contrario: los protocolos llegaron primero y el modelo fue en realidad una descripción de los protocolos existentes. No había problemas para ajustar los protocolos al modelo. Encajaban a la perfección. El único problema era que el *modelo* no aceptaba otras pilas de protocolos. Como consecuencia, no era útil para describir otras redes que no fueran TCP/IP.

Volviendo de los asuntos filosóficos a los más específicos, una diferencia patente entre los dos modelos es el número de capas: el modelo OSI tiene siete y el TCP/IP sólo cuatro. Los dos tienen capas de (inter)red, transporte y aplicación, pero las otras capas son diferentes.

Otra diferencia está en el área de la comunicación orientada a la conexión comparada con la no orientada a la conexión. El modelo OSI soporta ambas comunicaciones en la capa de red, pero sólo la de comunicación orientada a la conexión en la capa de transporte, donde es importante (porque el servicio de transporte es transparente para los usuarios). El modelo TCP/IP sólo tiene un modo en la capa de red (no orientado a la conexión) pero soporta ambos modos en la capa de transporte, lo que da a los usuarios la oportunidad de elegir. Esta elección es importante especialmente para protocolos sencillos de solicitud-respuesta.

1.4.4 Crítica al modelo OSI y los protocolos

Ni el modelo OSI y sus protocolos ni el modelo TCP/IP y sus protocolos son perfectos. Se les pueden hacer, y se les han hecho, críticas. En ésta y en la siguiente sección veremos algunas de estas críticas. Empezaremos con el modelo OSI y más adelante examinaremos el modelo TCP/IP.

En la época en la que se publicó la segunda edición de este libro (1989), a muchos expertos en el campo les pareció que el modelo OSI y sus protocolos iban a dominar el mundo y a desplazar a todos los demás. Eso no sucedió. ¿Por qué? Sería útil echar un vistazo a algunas lecciones. Éstas se pueden resumir así:

1. Aparición inoportuna.
2. Mala tecnología.
3. Malas implementaciones.
4. Malas políticas.

Aparición inoportuna

Primero veamos la razón número uno: aparición inoportuna. El tiempo en que se establece un estándar es absolutamente crítico para el éxito. David Clark, del M.I.T., tiene una teoría de estándares que llama *apocalipsis de los dos elefantes*, la cual se ilustra en la figura 1-23.

Esta figura muestra la cantidad de actividad que rodea a un sujeto nuevo. Cuando se descubre primero el sujeto, hay una explosión de actividad de investigación en forma de exposiciones, documentos y reuniones. Después de un tiempo esta actividad disminuye, las empresas descubren el sujeto y surge la ola de miles de millones de dólares de inversión.

Es esencial que los estándares se escriban en el punto intermedio entre los dos “elefantes”. Si los estándares se escriben demasiado pronto, antes de que se termine la investigación, el tema podría no estar entendido por completo; el resultado son malos estándares. Si se escriben demasiado tarde, varias empresas podrían haber hecho ya inversiones importantes en diversas maneras de hacer las cosas que los estándares han ignorado. Si el intervalo entre los dos elefantes es muy corto (porque cada cual tiene prisa por empezar), las personas que están desarrollando los estándares podrían fracasar.

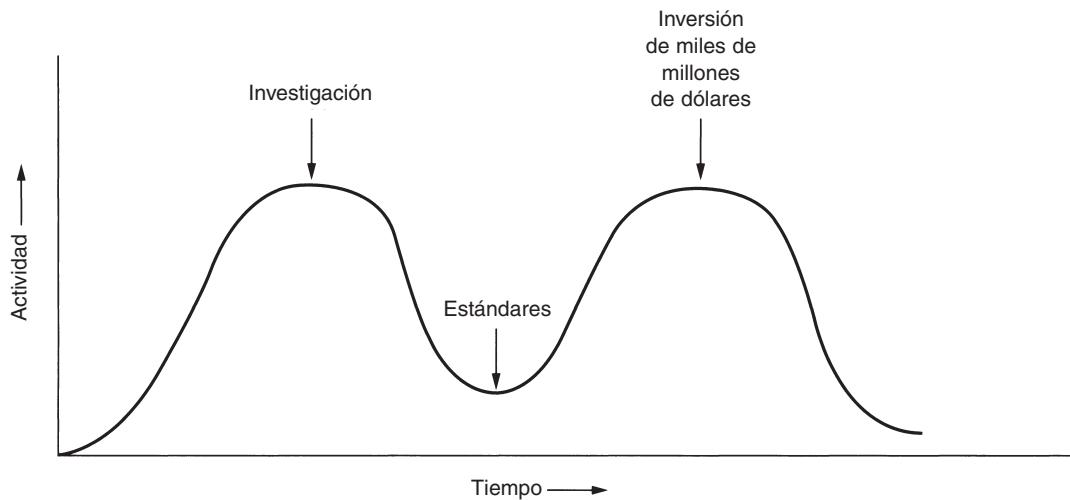


Figura 1-23. El apocalipsis de los dos elefantes.

Al parecer, los protocolos OSI estándar han sido vencidos. Los protocolos TCP/IP competidores ya eran ampliamente utilizados por las universidades investigadoras al momento en que aparecieron los protocolos OSI. Mientras la ola de los miles de millones de inversión aún no golpeaba, el mercado académico era bastante grande para que los proveedores empezaran a hacer ofertas cautivas de los productos TCP/IP. Cuando OSI llegó, no quisieron soportar una segunda pila de protocolos hasta que se vieran forzados, por lo que no hubo ofertas iniciales. Puesto que cada empresa esperaba que la otra diera el primer paso, ninguna lo hizo y OSI nunca prosperó.

Mala tecnología

La segunda razón por la que OSI no tuvo éxito es que tanto el modelo como los protocolos tienen defectos. La elección de las siete capas fue más política que técnica, y dos de las capas (la de sesión y la de presentación) están casi vacías, mientras que las otras dos (la de enlace de datos y la de red) están saturadas.

El modelo OSI, junto con el servicio asociado de definiciones y protocolos, es extraordinariamente complejo. Si se apilan, los estándares impresos ocupan una fracción importante de un metro de papel. Incluso son difíciles de implementar y de operación deficiente. En este contexto, nos viene a la mente un enigma propuesto por Paul Mockapetris y citado en (Rose, 1993):

P: ¿Qué obtiene cuando cruza un gángster con un estándar internacional?

R: Alguien que le hace una oferta que usted no entiende.

Además de ser incomprendible, otro problema con OSI es que algunas funciones, como direccionamiento, control de flujo y control de errores, reaparecen una y otra vez en cada capa. Por

ejemplo, Saltzer y cols. (1984) han apuntado que para ser efectivo el control de errores, se debe hacer en la capa superior, puesto que repetirlo una y otra vez en cada una de las capas inferiores suele ser innecesario e ineficaz.

Malas implementaciones

Ante la enorme complejidad del modelo y los protocolos, no es de sorprender que las implementaciones iniciales fueran grandes, pesadas y lentas. Todos los que lo intentaron fracasaron. No le tomó mucho tiempo a las personas asociar OSI con “baja calidad”. Aunque los productos mejoraron con el paso del tiempo, la imagen persistió.

En contraste, una de las primeras implementaciones de TCP/IP era parte de UNIX de Berkeley y fue bastante buena (sin mencionar que era gratis). Las personas pronto empezaron a utilizarla, lo que la llevó a un uso mayor por la comunidad, y esto a su vez condujo a mejoras que la llevaron a un mayor uso en la comunidad. Aquí la espiral fue ascendente en vez de descendente.

Malas políticas

A causa de la implementación inicial, muchas personas, sobre todo en el nivel académico, pensaban que TCP/IP era parte de UNIX, y en la década de 1980, UNIX no parecía tener paternidad alguna en la universidad.

Por otra parte, se tenía la idea de que OSI sería la criatura de los ministerios de telecomunicación de Europa, de la comunidad europea y más tarde del gobierno de los Estados Unidos. Esta creencia era cierta en parte, pero no ayudaba mucho la idea de un manoj de burócratas gubernamentales intentando poner en marcha un estándar técnicamente inferior al mando de los investigadores y programadores pobres que estaban en las trincheras desarrollando realmente redes de computadoras. Algunas personas compararon este desarrollo con la ocasión en que IBM anunció, en la década de 1960, que PL/I era el lenguaje del futuro, o cuando más tarde el DoD corregía esto anunciando que en realidad era Ada.

1.4.5 Crítica del modelo de referencia TCP/IP

El modelo de referencia TCP/IP y los protocolos también tienen sus problemas. En primer lugar, el modelo no distingue claramente los conceptos de servicio, interfaz y protocolo. Una buena ingeniería de software requiere la diferenciación entre la especificación y la implementación, algo que OSI hace con mucho cuidado y que TCP/IP no hace. En consecuencia, el modelo TCP/IP no es una guía para diseñar redes nuevas mediante tecnologías nuevas.

En segundo lugar, el modelo TCP/IP no es general del todo y no está bien ajustado para describir ninguna pila de protocolos más que de TCP/IP. Por ejemplo, es completamente imposible tratar de utilizar el modelo TCP/IP para describir Bluetooth.

En tercer lugar, la capa *host a red* no es en realidad una capa del todo en el sentido normal del término, como se utiliza en el contexto de los protocolos de capas. Es una interfaz (entre la capa de red y la de enlace de datos). La distinción entre una interfaz y una capa es crucial y nadie debe ser descuidado al respecto.

En cuarto lugar, el modelo TCP/IP no distingue (ni menciona) las capas física y de enlace de datos. Son completamente diferentes. La capa física tiene que ver con las características de transmisión de comunicación por cable de cobre, por fibra óptica e inalámbrica. El trabajo de la capa de enlace de datos es delimitar el inicio y fin de las tramas y captarlas de uno a otro lado con el grado deseado de confiabilidad. Un modelo adecuado debería incluir ambas como capas separadas. El modelo TCP/IP no hace esto.

Por último, aunque los protocolos IP y TCP se idearon e implementaron con sumo cuidado, muchos de los demás protocolos fueron hechos con fines específicos, producidos por lo general por estudiantes de licenciatura que los mejoraban hasta que se aburrían. Posteriormente, las implementaciones de tales protocolos se distribuyeron de manera gratuita, lo que dio como resultado un uso amplio y profundo y, por lo tanto, que fueran difíciles de reemplazar. Algunos de ellos ahora están en apuros. Por ejemplo, el protocolo de terminal virtual, TELNET, se diseñó para una terminal de teletipo mecánica de 10 caracteres por segundo. No sabe nada de interfaces gráficas de usuario ni de ratones. No obstante, 25 años más tarde aún tiene un amplio uso.

En resumen, a pesar de sus problemas, el *modelo* OSI (excepto las capas de sesión y presentación) ha probado ser excepcionalmente útil en la exposición de redes de computadoras. En contraste, los *protocolos* OSI no han sido muy populares. Sigue lo contrario con TCP/IP: el *modelo* es prácticamente inexistente, pero los *protocolos* tienen un amplio uso. En este libro utilizaremos un modelo OSI modificado pero nos concentraremos principalmente en el modelo TCP/IP y los protocolos relacionados, así como en los novísimos 802, SONET y Bluetooth. En efecto, utilizaremos el modelo híbrido de la figura 1-24 como marco de trabajo para este libro.

5	Capa de aplicación
4	Capa de transporte
3	Capa de red
2	Capa de enlace de datos
1	Capa física

Figura 1-24. Modelo de referencia híbrido que se usará en este libro.

1.5 REDES DE EJEMPLO

El tema de las redes de computadoras cubre muchos y diversos tipos de redes, grandes y pequeñas, bien conocidas y no tan bien conocidas. Tiene diferentes objetivos, escalamientos y tecnologías. En las siguientes secciones veremos algunos ejemplos para tener una idea de la variedad que se puede encontrar en el área de la conectividad de redes.

Empezaremos con Internet, que es probablemente la red más conocida y veremos su historia, evolución y tecnología. Luego consideraremos ATM, cuyo uso es frecuente en el núcleo de redes (telefónicas) grandes. Desde el punto de vista técnico difiere muy poco de Internet, y contrasta gratamente. Después presentaremos Ethernet, la red de área local dominante. Y, por último, veremos el IEEE 802.11, el estándar para las LANs inalámbricas.

1.5.1 Internet

Internet no es del todo una red, sino un inmenso conjunto de redes diferentes que usan ciertos protocolos comunes y proporcionan ciertos servicios comunes. Es un sistema poco común porque nadie lo planeó y nadie lo controla. Para entenderlo mejor, empecemos desde el principio y veamos cómo se desarrolló y por qué. Si desea leer una historia maravillosa sobre Internet, recomendamos ampliamente el libro de John Naughton (2000). Es uno de esos raros libros cuya lectura no sólo es divertida, sino que también contiene 20 páginas de *ibidem*s y op. cits. para el historiador serio. Parte del material que se muestra a continuación se basa en dicho libro.

Desde luego, se ha escrito una infinidad de libros técnicos sobre Internet y sus protocolos. Para más información, vea (Maufer, 1999).

ARPANET

Nuestro relato empieza a fines de la década de 1950. Durante el auge de la Guerra Fría, el DoD quería una red de control y comando que pudiera sobrevivir a una guerra nuclear. En esa época todas las comunicaciones militares usaban la red telefónica pública, que se consideraba vulnerable. La razón de esta creencia se puede entresacar de la figura 1-25(a). Los puntos negros representan las oficinas de conmutación telefónica, a cada una de las cuales se conectaban miles de teléfonos. Estas oficinas de conmutación estaban, a su vez, conectadas a oficinas de conmutación de más alto nivel (oficinas interurbanas), para conformar una jerarquía nacional con sólo una mínima redundancia. La vulnerabilidad del sistema estaba en que la destrucción de algunas de las oficinas interurbanas clave podía fragmentar el sistema en muchas islas incomunicadas.

Hacia 1960, el DoD firmó un contrato con RAND Corporation para encontrar una solución. Uno de sus empleados, Paul Baran, presentó el diseño de amplia distribución y tolerancia a fallas que se muestra en la figura 1-25(b). Puesto que las trayectorias entre cualquiera de las oficinas de conmutación eran ahora más grandes de lo que las señales análogas podían viajar sin distorsión, Baran propuso que se utilizara la tecnología digital de conmutación de paquetes a través del sistema. Baran escribió varios informes al DoD describiendo en detalle sus ideas. A los oficiales del Pentágono les agració el concepto y pidieron a AT&T, en ese entonces el monopolio telefónico estadounidense, que construyera un prototipo. AT&T desechó las ideas de Baran. La corporación más grande y rica del mundo no iba a permitir que un jovenzuelo le dijera cómo construir un sistema telefónico. Dijeron que la red de Baran no se podía construir y la idea se desechó.

Pasaron varios años y el DoD aún no tenía un mejor sistema de control y comando. Para entender qué sucedió a continuación, tenemos que volver al 7 de octubre de 1957, cuando la Unión soviética lanzó el Sputnik, su primer satélite artificial, con lo cual se le adelantó a Estados Unidos.

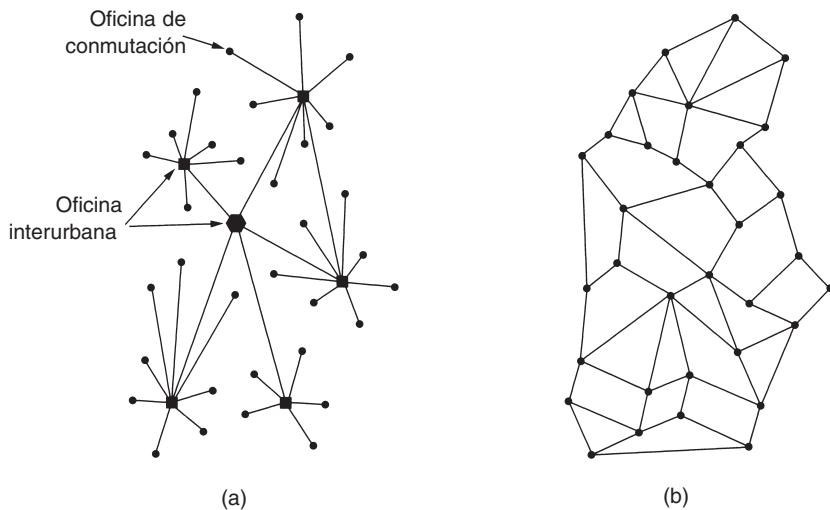


Figura 1-25. (a) Estructura de un sistema telefónico. (b) Sistema de conmutación distribuida propuesto por Baran.

Cuando el presidente Eisenhower trató de encontrar quién estaba dormido en sus laureles, se espantó al encontrarse con que la armada, el ejército y la fuerza aérea se peleaban por el presupuesto de investigación del Pentágono. Su respuesta inmediata fue crear una organización única de investigación para la defensa, **ARPA (Agencia de Proyectos de Investigación Avanzada)**. Ésta no tenía científicos ni laboratorios; de hecho, no tenía más que una oficina y un presupuesto pequeño (por normas del Pentágono). Hacía su trabajo otorgando subvenciones y contratos a universidades y empresas cuyas ideas le parecían prometedoras.

Durante los primeros años, ARPA trataba de imaginarse cuál sería su misión, pero en 1967 la atención de su entonces director, Larry Roberts, se volvió hacia las redes. Se puso en contacto con varios expertos para decidir qué hacer. Uno de ellos, Wesley Clark, sugirió la construcción de una subred de conmutación de paquetes, dando a cada *host* su propio enrutador, como se ilustra en la figura 1-10.

Después del escepticismo inicial, Roberts aceptó la idea y presentó un documento algo vago al respecto en el Simposio sobre Principios de Sistemas Operativos ACM SIGOPS que se llevó a cabo en Gatlinburg, Tennessee, a fines de 1967 (Roberts, 1967). Para mayor sorpresa de Roberts, otro documento en la conferencia describía un sistema similar que no sólo había sido diseñado, sino que ya estaba implementado bajo la dirección de Donald Davies en el National Physical Laboratory en Inglaterra. El sistema del NPL no era un sistema a nivel nacional (sólo conectaba algunas computadoras en el campus del NPL), pero demostró que era posible hacer que la conmutación de paquetes funcionara. Además, citaba el trabajo anterior de Baran, el cual había sido descartado. Roberts salió de Gatlinburg determinado a construir lo que más tarde se conocería como **ARPANET**.

La subred constaría de minicomputadoras llamadas **IMPs (Procesadores de Mensajes de Interfaz)**, conectadas por líneas de transmisión de 56 kbps. Para alta confiabilidad, cada IMP estaría conectado al menos a otros dos IMPs. La subred iba a ser de datagramas, de manera que si se destruían algunos IMPs, los mensajes se podrían volver a enrutar de manera automática a otras rutas alternativas.

Cada nodo de la red iba a constar de un IMP y un *host*, en el mismo cuarto, conectados por un cable corto. Un *host* tendría la capacidad de enviar mensajes de más de 8063 bits a su IMP, el cual los fragmentaría en paquetes de, a lo sumo, 1008 bits y los reenviaría de manera independiente hacia el destino. Cada paquete se recibiría íntegro antes de ser reenviado, por lo que la subred sería la primera red electrónica de conmutación de paquetes de almacenamiento y reenvío.

Entonces ARPA lanzó una convocatoria para construir la subred. Doce empresas licitaron. Después de evaluar las propuestas, ARPA seleccionó a BBN, una empresa de consultoría de Cambridge, Massachusetts, y en diciembre de 1968 le otorgó el contrato para construir la subred y escribir el software de ésta. BBN eligió utilizar como IMPs minicomputadoras Honeywell DDP-316 especialmente modificadas con palabras de 16 bits y 12 KB de memoria central. Los IMPs no tenían discos, ya que las partes móviles se consideraban no confiables. Estaban interconectadas por líneas de 56 kbps alquiladas a las compañías telefónicas. Aunque 56 kbps ahora es la elección de las personas que no pueden permitirse ADSL o cable, entonces era la mejor opción.

El software estaba dividido en dos partes: subred y *host*. El software de la subred constaba del extremo IMP de la conexión *host* a IMP, del protocolo IMP a IMP y de un protocolo de IMP origen a IMP destino diseñado para mejorar la confiabilidad. En la figura 1-26 se muestra el diseño original de ARPANET.

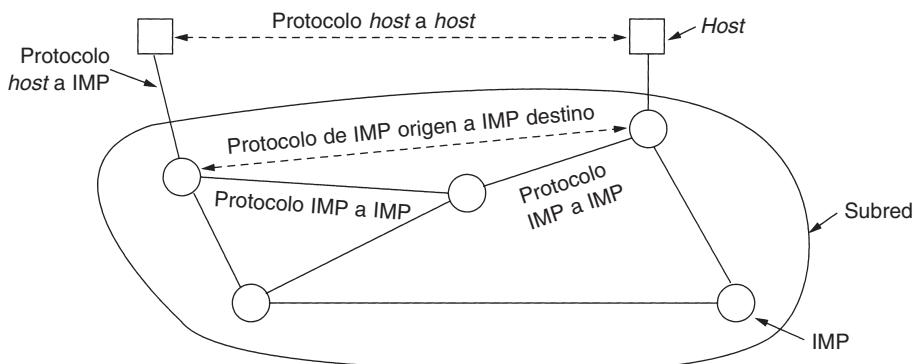


Figura 1-26. Diseño original de ARPANET.

Fuera de la subred también se necesitaba software, es decir, el extremo *host* de la conexión *host* a IMP, el protocolo *host* a *host* y el software de aplicación. Pronto quedó claro que BBN sintió que cuando se aceptaba un mensaje en un cable *host* a IMP y se ponía en un cable *host* a IMP en el destino, el trabajo estaba hecho.

Roberts tenía un problema: los *hosts* también necesitaban software. Para resolverlo convocó a una reunión de investigadores de red —en su mayoría estudiantes de licenciatura de Snowbird, Utah— durante el verano de 1969. Los estudiantes esperaban que algún experto en redes les explicara el gran diseño de la red y su software y que luego les asignara el trabajo de escribir parte de él. Se quedaron asombrados al descubrir que no había ningún experto ni un gran diseño. Tenían que averiguar qué era lo que se tenía que hacer.

No obstante, en diciembre de 1969 de alguna manera surgió una red experimental con cuatro nodos: en UCLA, UCSB, SRI y la Universidad de Utah. Se eligieron estas cuatro porque todas tenían un gran número de contratos de ARPA y todas tenían computadoras *host* diferentes incompatibles en su totalidad (precisamente para hacerlo más divertido). La red creció con rapidez a medida que se entregaban e instalaban más IMPs; pronto abarcó Estados Unidos. La figura 1-27 muestra qué tan rápido creció ARPANET en los primeros tres años.

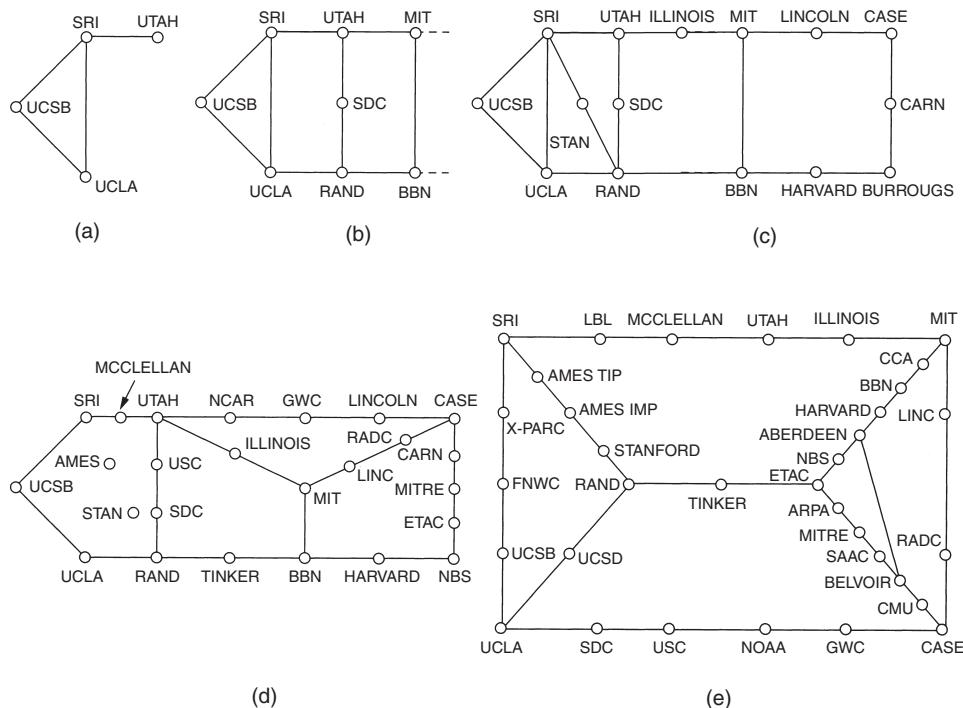


Figura 1-27. Crecimiento de ARPANET: (a) Diciembre de 1969. (b) Julio de 1970. (c) Marzo de 1971. (d) Abril de 1972. (e) Septiembre de 1972.

Además de ayudar al crecimiento de la novel ARPANET, ARPA también proporcionó fondos para la investigación sobre el uso de redes satelitales y redes de radio de paquetes móviles. En una demostración, ahora famosa, un camión que viajaba por California utilizó la red de radio de

paquetes para enviar mensajes a SRI, que luego los reenvió por ARPANET a la Costa Este, donde se expidieron al University College en Londres a través de una red satelital. Esto permitió que el investigador que iba en el camión usara una computadora que se encontraba en Londres mientras manejaba por California.

Este experimento también demostró que los protocolos existentes de ARPANET no eran adecuados para ejecutarse a través de varias redes. Esta observación condujo a más investigación sobre los protocolos, culminando con la invención del modelo y los protocolos de TCP/IP (Cerf y Kahn, 1974). TCP/IP está diseñado de manera específica para manejar comunicación por interredes, aspecto cuya importancia se acrecentó conforme cada vez más y más redes se adhirieron a ARPANET.

Para alentar la adopción de estos nuevos protocolos, ARPA concedió varios contratos a BBN y a la Universidad de California en Berkeley para integrarlos en UNIX de Berkeley. Los investigadores en Berkeley desarrollaron una interfaz de programa adecuada para la red (sockets) y escribieron muchos programas de aplicación, utilería y administración para hacer más fácil la conectividad.

El momento era perfecto. Muchas universidades habían adquirido recientemente una segunda o tercera computadora VAX y una LAN para conectarlas, pero no tenían software de redes. Cuando llegó 4.2BSD junto con TCP/IP, sockets y muchas utilerías de red, el paquete completo se adoptó de inmediato. Además, con TCP/IP, fue fácil para las LANs conectarse a ARPANET y muchas lo hicieron.

Durante la década de 1980, se conectaron redes adicionales, en particular LANs, a ARPANET. Conforme crecía el escalamiento, encontrar *hosts* llegó a ser muy costoso, por lo que se creó el **DNS (Sistema de Nombres de Dominio)** para organizar máquinas dentro de dominios y resolver nombres de *host* en direcciones IP. Desde entonces, el DNS ha llegado a ser un sistema de base de datos distribuido generalizado para almacenar una variedad de información relacionada con la elección de un nombre. En el capítulo 7 estudiaremos en detalle este tema.

NSFNET

A finales de la década de 1970, la NFS (Fundación Nacional para las Ciencias, de Estados Unidos) vio el enorme impacto que ARPANET estaba teniendo en la investigación universitaria, permitiendo que científicos de todo el país compartieran datos y colaboraran en proyectos de investigación. Sin embargo, para estar en ARPANET, una universidad debía tener un contrato de investigación con el DoD, lo cual muchas no tenían. La respuesta de la NSF fue diseñar un sucesor de ARPANET que pudiera estar abierto a todos los grupos de investigación de las universidades. Para tener algo concreto con que empezar, la NSF decidió construir una red dorsal (o troncal) para conectar sus seis centros de supercomputadoras en San Diego, Boulder, Champaign, Pittsburgh, Ithaca y Princeton. A cada supercomputadora se le dio un hermano menor, que consistía en una microcomputadora LSI-11 llamada *fuzzball*. Estas computadoras estaban conectadas a líneas alquiladas de 56 kbps y formaban una subred, utilizando la misma tecnología de hardware que ARPANET. Sin embargo, la tecnología de software era diferente: las *fuzzball* utilizan TCP/IP desde el inicio, creando así la primera WAN TCP/IP.

La NSF también fundó algunas redes regionales (alrededor de 20) que se conectaban a la red dorsal para que los usuarios en miles de universidades, laboratorios de investigación, bibliotecas y museos, tuvieran acceso a cualquiera de las supercomputadoras y se comunicaran entre sí. Toda la red, incluyendo la red dorsal y las redes regionales, se llamó **NSFNET**. Ésta se conectó a ARPANET a través de un enlace entre un IMP y una *fuzzball* en el cuarto de máquinas de Carnegie-Mellon. En la figura 1-28 se muestra la primera red dorsal NSFNET.

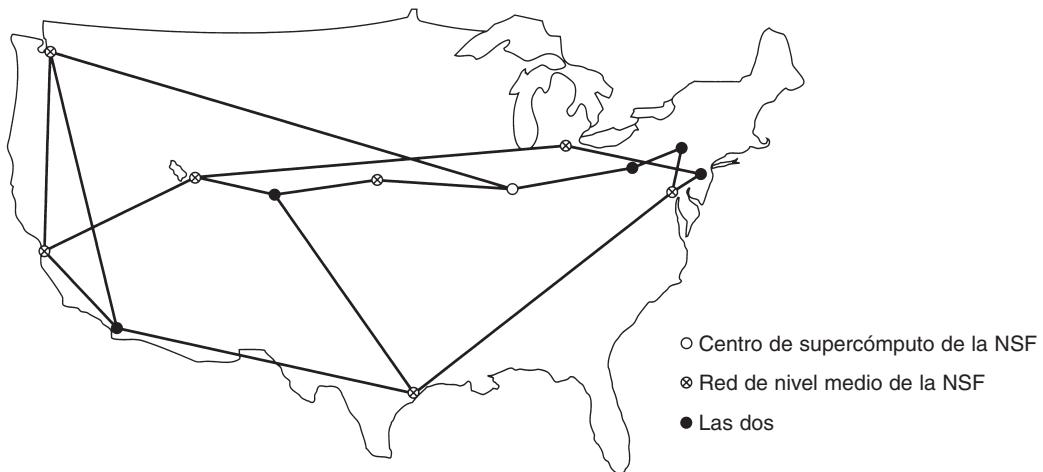


Figura 1-28. La red dorsal NSFNET en 1988.

NSFNET fue un éxito instantáneo y pronto se saturó. Inmediatamente, la NSF empezó a planear su sucesor y otorgó un contrato al consorcio MERIT de Michigan para que lo creara. Se alquilaron a MCI (puesto que se fusionó con WorldCom) canales de fibra óptica a 448 kbps para establecer la versión 2 de la red dorsal. Se utilizaron PC-RTs de IBM como enrutadores. Esta segunda red dorsal también se sobrecargó pronto, y en 1990 se escaló a 1.5 Mbps.

Al continuar el crecimiento, la NSF se percató de que el gobierno no podría financiar por siempre el uso de redes. Además, las empresas comerciales se querían unir, pero los estatutos de la NSF les prohibían utilizar las redes por las que la NSF había pagado. En consecuencia, la NSF alentó a MERIT, MCI e IBM a que formaran una corporación no lucrativa, **ANS (Redes y Servicios Avanzados)**, como el primer paso hacia la comercialización. En 1990, ANS adquirió NSFNET y escaló los enlaces de 1.5 Mbps a 45 Mbps para formar **ANSNET**. Esta red operó durante cinco años y luego fue vendida a America Online. Pero para entonces varias empresas estaban ofreciendo servicios IP comerciales y fue evidente que el gobierno se debía retirar del negocio de las redes.

Para facilitar la transición y hacer que todas las redes regionales se pudieran comunicar con las demás redes regionales, la NSF concedió contratos a cuatro diferentes operadores de redes para establecer un **NAP (Punto de Acceso a la Red)**. Estos operadores eran PacBell (San Francisco),

Ameritech (Chicago), MFS (Washington, D.C.) y Sprint (Nueva York, donde para efectos de NAP, Pennsauken, Nueva Jersey se toma en cuenta como si fuera la ciudad de Nueva York). Todo operador de red que quisiera proporcionar el servicio de red dorsal a las redes regionales de la NSF se tenía que conectar a todos los NAPs.

Este arreglo significaba que un paquete que se originara en cualquier red regional tenía la opción de contar con operadores de red dorsal desde su NAP al NAP de destino.

En consecuencia, los operadores de red dorsal se vieron forzados a competir por el negocio de las redes regionales con base en el servicio y el precio, que, desde luego, era la idea. Como resultado, el concepto de una única red dorsal predeterminada fue reemplazado por una infraestructura competitiva orientada a la comercialización. A muchas personas les gusta criticar al gobierno federal por no ser innovador, pero en el área de redes, el DoD y la NSF fueron los creadores de la infraestructura que cimentó la base para Internet y luego dejaron que la industria la operara.

Durante la década de 1990, muchos otros países y regiones también construyeron redes nacionales de investigación, con frecuencia siguiendo el patrón de ARPANET y NSFNET. Éstas incluían EuropaNET y EBONE en Europa, que empezaron con líneas de 2 Mbps y luego las escalaron a 34 Mbps. Finalmente, en Europa la infraestructura de redes quedó en manos de la industria.

Uso de Internet

El número de redes, máquinas y usuarios conectados a ARPANET creció rápidamente luego de que TCP/IP se convirtió en el protocolo oficial el 10. de enero de 1983. Cuando NSFNET y ARPANET estaban interconectadas, el crecimiento se hizo exponencial. Muchas redes regionales se unieron y se hicieron conexiones a redes en Canadá, Europa y el Pacífico.

En algún momento a mediados de la década de 1980, las personas empezaron a ver el conjunto de redes como una interred y más tarde como Internet, aunque no hubo una inauguración oficial con algún político rompiendo una botella de champaña sobre una *fuzzball*.

El aglutinante que mantiene unida la Internet es el modelo de referencia TCP/IP y la pila de protocolos de TCP/IP. TCP/IP hace posible el servicio universal y se puede comparar con la adopción de la medida estándar para el ancho de vía del ferrocarril en el siglo XIX o la adopción de los protocolos de señalización comunes para las compañías telefónicas.

¿Qué significa en realidad estar en Internet? Nuestra definición es que una máquina está en Internet si ejecuta la pila de protocolos de TCP/IP, tiene una dirección IP y puede enviar paquetes IP a todas las demás máquinas en Internet. La sola capacidad para enviar y recibir correo electrónico no basta, puesto que el correo electrónico es la puerta de entrada a muchas redes fuera de Internet. Sin embargo, el aspecto se nubla de alguna manera por el hecho de que millones de computadoras personales pueden llamar a un proveedor de servicios de Internet mediante un módem, recibir direcciones IP temporales y enviar paquetes IP a otros *hosts* de Internet. Tiene sentido decir que tales máquinas están en Internet en tanto estén conectadas al enrutador del proveedor de servicios.

Tradicionalmente (es decir, de 1970 a 1990) Internet y sus predecesores tenían cuatro aplicaciones principales:

1. **Correo electrónico.** La capacidad para redactar, enviar y recibir correo electrónico ha sido posible desde los inicios de ARPANET y su gran popularidad. Muchas personas obtienen docenas de mensajes al día y consideran esto como su primer medio de interactuar con el mundo exterior, más allá del teléfono y el correo caracol que se han quedado atrás. Hoy en día los programas de correo electrónico están disponibles en prácticamente todo tipo de computadora.
2. **Noticias.** Los grupos de noticias son foros especializados en los que los usuarios con un interés común pueden intercambiar mensajes. Existen miles de grupos de noticias, dedicados a temas técnicos y no técnicos, entre ellos computadoras, ciencia, recreación y política. Cada grupo de noticias tiene su propia etiqueta, estilo, hábitos y penas en que se incurre al violarlas.
3. **Inicio remoto de sesión.** Mediante los programas telnet, rlogin o ssh, los usuarios de cualquier parte en Internet pueden iniciar sesión en cualquier otra máquina en la que tengan una cuenta.
4. **Transferencia de archivos.** Con el programa FTP, los usuarios pueden copiar archivos de una máquina en Internet a otra. Por este medio se encuentra disponible una vasta cantidad de artículos, bases de datos y otra información.

Hasta principios de la década de 1990, Internet era muy visitada por investigadores académicos, del gobierno e industriales. Una nueva aplicación, **WWW (World Wide Web)** cambió todo eso y trajo millones de usuarios nuevos no académicos a la red. Esta aplicación —inventada por Tim Berners-Lee, físico del CERN— no cambió ninguna de las características subyacentes pero las hizo más fáciles de usar. Junto con el navegador Mosaic, escrito por Marc Andreessen en el Centro Nacional para Aplicaciones de Supercómputo en Urbana, Illinois, WWW hizo posible que un sitio estableciera páginas de información que contienen texto, imágenes, sonido e incluso vídeo, y vínculos integrados a otras páginas. Al hacer clic en un vínculo, el usuario es transportado de inmediato a la página a la que apunta dicho vínculo. Por ejemplo, muchas compañías tienen una página de inicio con entradas que apuntan a otras páginas que contienen información de productos, listas de precios, ventas, soporte técnico, comunicación con empleados, información para los accionistas y más.

En muy poco tiempo han aparecido páginas de otro tipo, incluyendo mapas, tablas del mercado accionario, catálogos de fichas bibliográficas, programas de radio grabados e incluso una página que apunta al texto completo de muchos libros cuyos derechos de autor han expirado (Mark Twain, Charles Dickens, etcétera). Muchas personas también tienen páginas personales (páginas de inicio).

Gran parte de su crecimiento durante la década de 1990 estuvo alimentado por empresas llamadas **ISPs (proveedores de servicios de Internet)**. Hay compañías que ofrecen a los usuarios individuales domésticos la capacidad de llamar a una de sus máquinas y conectarse a Internet, obteniendo así acceso al correo electrónico, WWW y otros servicios de Internet. Estas compañías suscribieron contratos con decenas de millones de usuarios nuevos por un año durante el final de

la década de 1990, cambiando por completo el carácter de la red de ser un campo de recreo para académicos y militares a uno de utilidad pública, muy semejante al sistema telefónico. Ahora el número de usuarios de Internet se desconoce, pero lo cierto es que son cientos de millones en todo el mundo y tal vez pronto lleguen a rebasar los mil millones.

Arquitectura de Internet

En esta sección trataremos de dar un breve panorama de lo que es Internet hoy. Debido a las muchas fusiones entre compañías telefónicas (telcos) e ISPs, las aguas se han enturbiado y a veces es difícil decir quién hace qué. En consecuencia, la siguiente descripción será, por necesidad, algo más sencilla que la realidad. En la figura 1-29 se muestra el panorama general. Ahora examinemos esta figura parte por parte.

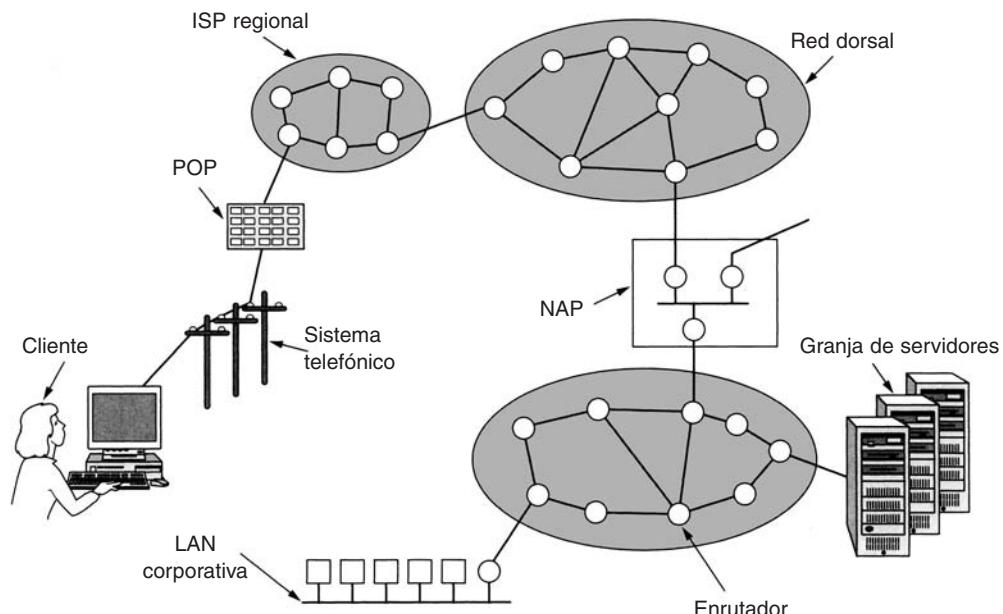


Figura 1-29. Panorama de Internet.

Un buen lugar para empezar es con un cliente en casa. Supongamos que nuestro cliente llama a su ISP desde una conexión de línea telefónica conmutada, como se muestra en la figura 1-29. El módem es una tarjeta dentro de su PC que convierte las señales digitales que la computadora produce en señales analógicas que pueden pasar sin obstáculos a través del sistema telefónico. Estas señales se transfieren al **POP (Punto de Presencia)** del ISP, donde se retiran del sistema telefónico y se inyectan en la red regional del ISP. A partir de este punto, el sistema es totalmente digital y de conmutación de paquetes. Si el ISP es la telco local, es probable que el POP esté ubicado en la

oficina de conmutación telefónica, donde termina el cableado de teléfono de los clientes. Si el ISP no es la telco local, el POP podría ser alguna de las oficinas de conmutación en el camino.

La red regional de ISPs consta de enrutadores interconectados en las diversas ciudades en las que el ISP opera o da servicio. Si el paquete está destinado a un *host* servido directamente por el ISP, el paquete se entrega al *host*. En caso contrario, se entrega al operador de la red dorsal del ISP.

En la cima de la cadena alimenticia están los operadores de las principales redes dorsales, empresas como AT&T y Sprint. Éstas operan grandes redes dorsales internacionales, con miles de enrutadores conectados por fibra óptica de banda ancha. Grandes corporaciones y servicios de *hosting* que ejecutan granjas de servidores (máquinas que pueden servir miles de páginas Web por segundo) con frecuencia se conectan de manera directa a la red dorsal. Los operadores de redes dorsales alientan esta conexión directa rentando espacio en lo que se llama **hoteles de portadores**, que son básicamente gabinetes de equipos en el mismo cuarto que el enrutador para conexiones cortas y rápidas entre las granjas de servidores y la red dorsal.

Si un paquete dado a la red dorsal se destina a un ISP o a una compañía servida por la red dorsal, se envía al enrutador más cercano y se pierde cualquier responsabilidad por este paquete. Sin embargo, en el mundo hay muchas redes dorsales, de varios tamaños, de manera que un paquete podría tener que ir a una red dorsal competidora. Para que los paquetes viajen entre redes dorsales, todas las redes principales se conectan a los NAPs explicados antes. Básicamente, un NAP es un cuarto lleno de enrutadores, al menos uno por red dorsal. Una LAN en el cuarto conecta todos los enrutadores, de modo que los paquetes se pueden reenviar desde una red dorsal hacia cualquier otra. Además de estar conectadas en los NAPs, las redes dorsales más grandes tienen numerosas conexiones directas entre sus enrutadores, una técnica conocida como **igualdad privada** (*private peering*). Una de las muchas paradojas de Internet es que los ISPs que compiten en público entre sí por clientes, con frecuencia cooperan estableciendo igualdades privadas entre ellos (Metz, 2001).

Aquí termina nuestro rápido viaje por Internet. En los siguientes capítulos tendremos mucho que decir sobre los componentes individuales y su diseño, algoritmos y protocolos. También vale la pena mencionar de paso que algunas empresas tienen interconectadas todas sus redes internas existentes, utilizando con frecuencia la misma tecnología que Internet. Por lo general, estas **intranets** son accesibles sólo dentro de la empresa pero, por otra parte, funcionan del mismo modo que Internet.

1.5.2 Redes orientadas a la conexión:

X.25, Frame Relay y ATM

Desde el inicio de la conectividad surgió una guerra entre aquellos que apoyan a las subredes no orientadas a la conexión (es decir, de datagramas) y quienes apoyan a las subredes orientadas a la conexión. Los principales defensores de las subredes no orientadas a la conexión vienen de la comunidad ARPANET/Internet. Recuerde que el deseo original del DoD al fundar y construir ARPANET era tener una red que pudiera funcionar incluso después de que varios impactos de armas nucleares destruyeran numerosos enrutadores y líneas de transmisión. Por lo tanto, la tolerancia a

errores era importante en su lista de prioridades, no tanto lo que pudieran cobrar a los clientes. Este enfoque condujo a un diseño no orientado a la conexión en el que cada paquete se enruta independientemente de cualquier otro paquete. Por lo tanto, si algunos enrutadores se caen durante una sesión, no hay daño puesto que el sistema puede reconfigurarse a sí mismo de manera dinámica para que los paquetes subsiguientes puedan encontrar alguna ruta a su destino, aun cuando sea diferente de la que utilizaron los paquetes anteriores.

El campo orientado a la conexión viene del mundo de las compañías telefónicas. En el sistema telefónico, quien llama debe marcar el número de la parte a la que desea llamar y esperar la conexión antes de poder hablar o enviar los datos. Esta configuración de conexión establece una ruta a través del sistema telefónico que se mantiene hasta que se termina la llamada. Todas las palabras o paquetes siguen la misma ruta. Si una línea o conmutador se cae en el trayecto, la llamada se cancela. Esta propiedad es precisamente lo que al DoD no le gustaba.

Entonces, ¿por qué le gustaba a las compañías telefónicas? Por dos razones:

1. Calidad en el servicio.
2. Facturación.

Al establecer de antemano una conexión, la red puede reservar recursos como espacio de búfer y capacidad de procesamiento (CPU) en los enrutadores. Si se intenta establecer una llamada y los recursos disponibles son insuficientes, la llamada se rechaza y el invocador recibe una señal de ocupado. De esta manera, una vez que se establece una conexión, ésta da un buen servicio. Con una red no orientada a la conexión, si llegan demasiados paquetes al mismo enrutador al mismo tiempo, el enrutador se saturará y tal vez pierda algunos paquetes. Tal vez el emisor advierta esto y los envíe de nuevo, pero la calidad del servicio será accidentada e inadecuada para audio o vídeo a menos que la red tenga poca carga. No es necesario decir que proveer una calidad de audio adecuada es algo en lo que las compañías telefónicas ponen mucho cuidado, de ahí su preferencia por las conexiones.

La segunda razón por la que las compañías telefónicas prefieren el servicio orientado a la conexión es que están acostumbradas a cobrar por el tiempo de conexión. Cuando hace una llamada de larga distancia (sea nacional o internacional) se le cobra por minuto. Cuando llegaron las redes, se vieron atraídas precisamente hacia un modelo en el que el cobro por minuto fuera fácil de hacer. Si se tiene que establecer una conexión antes de enviar los datos, en ese momento es cuando el reloj de la facturación empieza a correr. Si no hay conexión, no hay cobro.

Irónicamente, mantener registros de facturación es muy costoso. Si una compañía telefónica adoptara una tarifa mensual plana sin límite de llamadas y sin facturación o mantenimiento de un registro, probablemente ahorraría una gran cantidad de dinero, a pesar del incremento en llamadas que generaría esta política. Sin embargo, hay políticas, regulaciones y otros factores que pesan en contra de hacer esto. Curiosamente, el servicio de tarifa plana existe en otros sectores. Por ejemplo, la TV por cable se factura en una tasa mensual plana, independientemente de cuántos programas vea. Podría haberse diseñado con pago por evento como concepto básico, pero no fue así, en parte por lo costoso de la facturación (y dada la calidad de la mayoría de los programas televisivos, la vergüenza no se puede descontar del todo). Asimismo, muchos parques de diversiones

cobran una cuota de admisión por día con acceso ilimitado a los juegos, en contraste con las ferias ambulantes que cobran por juego.

Dicho esto, no nos debería sorprender que todas las redes diseñadas por la industria de la telefonía hayan sido subredes orientadas a la conexión. Lo que sí es de sorprender es que Internet también se está inclinado en esa dirección, a fin de proporcionar un mejor servicio de audio y vídeo, un tema al que volveremos en el capítulo 5. Por ahora examinaremos algunas redes orientadas a la conexión.

X.25 y Frame Relay

Nuestro primer ejemplo de red orientada a la conexión es la **X.25**, que fue la primera red de datos pública. Se desplegó en la década de 1970, cuando el servicio telefónico era un monopolio en todas partes y la compañía telefónica de cada país esperaba que hubiera una red de datos por país —la propia. Para utilizar X.25, una computadora establecía primero una conexión con la computadora remota, es decir, hacía una llamada telefónica. Esta conexión daba un número de conexión para utilizarlo en los paquetes de transferencia de datos (ya que se podían abrir muchas conexiones al mismo tiempo). Los paquetes de datos eran muy sencillos, consistían en un encabezado de 3 bytes y hasta 128 bytes de datos. El encabezado constaba de un número de conexión de 12 bits, un número de secuencia de paquete, un número de confirmación de recepción y algunos bits diversos. Las redes X.25 funcionaron por casi diez años con resultados mixtos.

En la década de 1980, las redes X.25 fueron reemplazadas ampliamente por un nuevo tipo de red llamada **Frame Relay**. Ésta es una red orientada a la conexión sin controles de error ni de flujo. Como era orientada a la conexión, los paquetes se entregaban en orden (en caso de que se entregaran todos). Las propiedades de entrega en orden, sin control de errores ni de flujo hicieron el Frame Relay parecido a la LAN de área amplia. Su aplicación más importante es la interconexión de LANs en múltiples oficinas de una empresa. Frame Relay disfrutó de un éxito modesto y aún se sigue utilizando en algunas partes.

Modo de Transferencia Asíncrona

Otro tipo de red orientada a la conexión, tal vez el más importante, es **ATM (Modo de Transferencia Asíncrona)**. La razón de tan extraño nombre se debe a que en el sistema telefónico la mayor parte de la transmisión es síncrona (lo más parecido a un reloj), y en ATM no sucede así.

ATM se diseñó a principios de la década de 1990 y se lanzó en medio de una increíble exageración (Ginsburg, 1996; Goralski, 1995; Ibe, 1997; Kimn y cols., 1994, y Stallings, 2000). ATM iba a resolver todos los problemas de conectividad y telecomunicaciones fusionando voz, datos, televisión por cable, télex, telégrafo, palomas mensajeras, botes conectados por cordón, tambores, señales de humo y todo lo demás, en un solo sistema integrado que pudiera proporcionar todos los servicios para todas las necesidades. Eso no sucedió. En gran parte, los problemas fueron semejantes a los ya descritos en el tema de OSI, es decir, una aparición inoportuna, junto con tecnología, implementación y políticas equivocadas. Habiendo noqueado a las compañías telefónicas en el primer asalto, gran parte de la comunidad de Internet vio a ATM como cuando Internet era el contrincante de las telcos: la segunda parte. Pero no fue así en realidad y esta vez incluso los in-

transigentes fanáticos de los datagramas se dieron cuenta de que la calidad de servicio de Internet dejaba mucho que desear. Para no alargar la historia, ATM tuvo mucho más éxito que OSI y actualmente tiene un uso profundo dentro del sistema telefónico, con frecuencia en el transporte de los paquetes IP. Como en la actualidad las empresas portadoras la utilizan principalmente para su transporte interno, los usuarios no se percatan de su existencia pero, definitivamente, vive y goza de salud.

Circuitos virtuales de ATM

Puesto que las redes ATM están orientadas a la conexión, el envío de datos requiere que primero se envíe un paquete para establecer la conexión. Conforme el mensaje de establecimiento sigue su camino a través de la subred, todos los conmutadores que se encuentran en la ruta crean una entrada en sus tablas internas tomando nota de la existencia de la conexión y reservando cualesquier recursos que necesite la conexión. Con frecuencia a las conexiones se les conoce como **circuitos virtuales**, en analogía con los circuitos físicos utilizados en el sistema telefónico. La mayoría de las redes ATM soportan también **circuitos virtuales permanentes**, que son conexiones permanentes entre dos *hosts* (distantes). Son similares a las líneas alquiladas del mundo telefónico. Cada conexión, temporal o permanente, tiene un solo identificador de conexión. En la figura 1-30 se ilustra un circuito virtual.

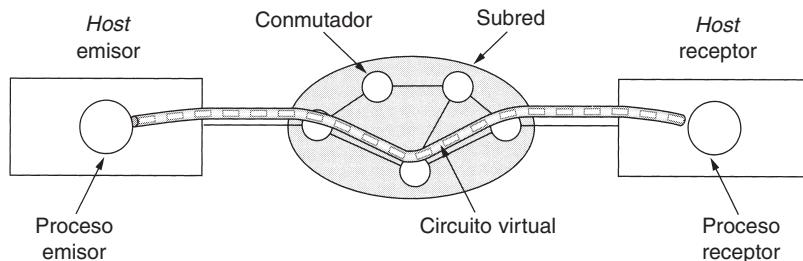


Figura 1-30. Un circuito virtual.

Una vez establecida la conexión, cada lado puede empezar a transmitir datos. La idea básica en que se fundamenta ATM es transmitir toda la información en paquetes pequeños, de tamaño fijo, llamados **celdas**. Las celdas tienen un tamaño de 53 bytes, de los cuales cinco son del encabezado y 48 de carga útil, como se muestra en la figura 1-31. Parte del encabezado es el identificador de la conexión, por lo que los *hosts* emisor y receptor y todos los conmutadores intermedios pueden saber qué celdas pertenecen a qué conexiones. Esta información permite que cada conmutador sepa cómo enviar cada celda entrante. La conmutación de celdas se hace en el hardware, a alta velocidad. De hecho, el principal argumento para tener celdas de tamaño fijo es que así es fácil construir conmutadores de hardware para manejar celdas pequeñas, de longitud fija. Los paquetes de longitud variable de IP se tienen que enrutar mediante software, que es un proceso más lento.

Otro punto a favor de ATM es que el hardware se puede configurar para enviar una celda entrante a múltiples líneas de salida, una propiedad necesaria para el manejo de un programa de televisión que se va a difundir a varios receptores. Por último, las celdas pequeñas no bloquean ninguna línea por mucho tiempo, lo que facilita la garantía en la calidad del servicio.

Todas las celdas siguen la misma ruta al destino. La entrega de celdas no está garantizada, pero el orden sí. Si las celdas 1 y 2 se envían en ese orden, entonces deben arribar en el mismo orden, nunca primero la 2 y luego la 1. No obstante, una de las dos o ambas se pueden perder en el trayecto. A los niveles más altos de protocolos les corresponde la recuperación de celdas perdidas. Observe que aunque esta garantía no es perfecta, es mejor que la de Internet. Ahí los paquetes no sólo se pierden, sino que además se entregan en desorden. ATM, en contraste, garantiza que las celdas nunca se entregarán en desorden.



Figura 1-31. Una celda ATM.

Las redes ATM se organizan como las WANs tradicionales, con líneas y conmutadores (enrutadores). Las velocidades más comunes para las redes ATM son de 155 y 622 Mbps, aunque también se soportan velocidades más altas. Se eligió la velocidad de 155 Mbps porque ésta es la que se requiere para transmitir televisión de alta definición. La elección exacta de 155.52 Mbps se hizo por compatibilidad con el sistema de transmisión SONET de AT&T, punto que estudiaremos en el capítulo 2. La velocidad de 622 Mbps se eligió para que se pudieran enviar cuatro canales de 155 Mbps.

El modelo de referencia ATM

ATM tiene su propio modelo de referencia, el cual es diferente del OSI y también del TCP/IP. En la figura 1.32 se muestra el modelo de referencia ATM. Consta de tres capas: la física, la ATM y la de adaptación ATM, además de lo que el usuario deseé poner arriba de ellas.

La capa física tiene que ver con el medio físico: voltajes, temporización de bits y otros aspectos más. ATM no prescribe un conjunto particular de reglas, tan sólo especifica que las celdas ATM se pueden enviar tal cual por cable o fibra, pero también se pueden empacar dentro de la carga útil de otros sistemas de transporte. En otras palabras, ATM se ha diseñado para ser independiente del medio de transmisión.

La **capa ATM** se encarga de las celdas y su transporte. Define la disposición de una celda e indica qué significan los campos del encabezado. También tiene que ver con el establecimiento y la liberación de los circuitos virtuales. El control de congestión también se ubica aquí.

Puesto que la mayoría de las aplicaciones no necesita trabajar de manera directa con las celdas (aunque algunas podrían hacerlo), se ha definido una capa superior a la capa ATM para que

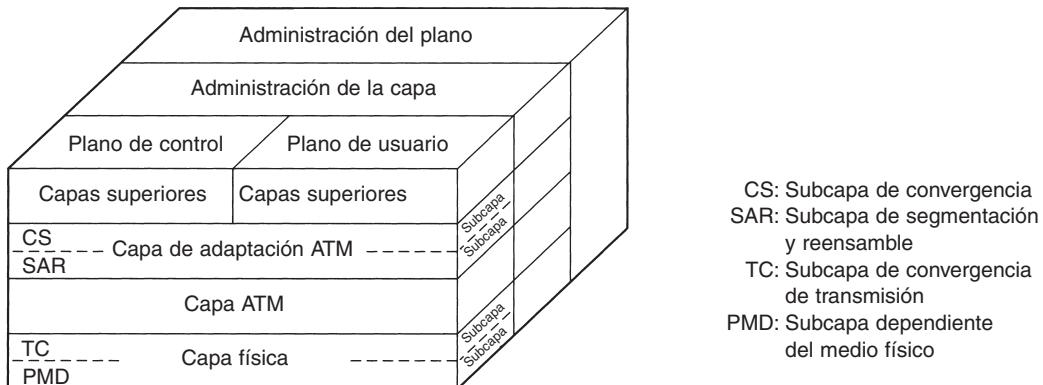


Figura 1-32. El modelo de referencia ATM.

los usuarios envíen paquetes más grandes que una celda. La interfaz de ATM segmenta estos paquetes, transmite de forma individual las celdas y las reensambla en el otro extremo. Esta capa es la **AAL (Capa de Adaptación ATM)**.

A diferencia de los primeros modelos de referencia bidimensionales, el modelo ATM se define como si fuera tridimensional, lo que se puede apreciar en la figura 1-32. El **plano de usuario** trata con el transporte de datos, control de flujo, corrección de errores y otras funciones de usuario. En contraste, el **plano de control** se ocupa de la administración de la conexión. Las funciones de administración del plano y de la capa se relacionan con la administración de recursos y coordinación entre capas.

Cada una de las capas física y AAL se dividen en dos subredes, una en la parte inferior que hace el trabajo y en la subcapa de convergencia en la parte superior que proporciona la interfaz propia de la capa superior inmediata. En la figura 1-33 se muestran las funciones de las capas y subcapas.

La subcapa **PMD (Dependiente del Medio Físico)** interactúa con el cable real. Mueve los bits dentro y fuera y maneja la temporización de bits, es decir, el tiempo que existe entre cada bit al transmitirlos. Esta capa será diferente para diferentes transportadoras y cables.

La otra subcapa de la capa física es la subcapa **TC (Convergencia de Transmisión)**. Cuando se transmiten las celdas, la capa TC las envía como una cadena de bits a la capa PMD. Esto es sencillo. En el otro extremo, la subcapa TC recibe una serie de bits de entrada de la subcapa PMD. Su trabajo es convertir este flujo de bits en un flujo de celdas para la capa ATM. Maneja todos los aspectos relacionados con las indicaciones de dónde empiezan y terminan las celdas en el flujo de bits. En el modelo ATM, esta funcionalidad se da en la capa física. En el modelo OSI y en gran parte de las demás redes, el trabajo de entramado, es decir, convertir una serie de bits en bruto en una secuencia de tramas o celdas, es la tarea de la capa de enlace de datos.

Como mencionamos antes, la capa ATM maneja celdas, incluyendo su generación y transporte. La mayoría de los aspectos interesantes de ATM se encuentra ubicada aquí. Es una combinación de las capas de enlace de datos y de red del modelo OSI; no hay una división en subcapas.

Capa OSI	Capa ATM	Subcapa ATM	Funcionalidad
3/4	AAL	CS	Provisión de la interfaz estándar (convergencia)
		SAR	Segmentación y reensamblado
2/3	ATM		Control de flujo Generación/extracción de encabezado de celda Círculo virtual/administración de ruta Multiplexión/desmultiplexión de celdas
2	Física	TC	Desacoplamiento de proporción de celdas Generación y comprobación de la suma de verificación de encabezados Generación de celdas Empaque/desempaque de celdas a partir del sobre contenedor Generación de tramas
			Temporización de bits Acceso a la red física
1		PMD	

Figura 1-33. Las capas y subcapas de ATM y sus funciones.

La capa AAL se divide en una subcapa **SAR (Segmentación y Reensamble)** y una **CS (Subcapa de Convergencia)**. La subcapa inferior fragmenta paquetes en celdas en el lado de transmisión y los une de nuevo en el destino. La subcapa superior permite que los sistemas ATM ofrezcan diversos tipos de servicios a diferentes aplicaciones (por ejemplo, la transferencia de archivos y el vídeo bajo demanda tienen diferentes requerimientos respecto a manejo de errores, temporización, etcétera).

Puesto que quizá ATM esté en declive, no lo explicaremos más en este libro. No obstante, puesto que existe una base instalada considerable, es probable que aún siga en uso durante algunos años. Para más información sobre ATM, vea (Dobrowsky y Grise, 2001, y Gadecki y Heckart, 1997).

1.5.3 Ethernet

Internet y ATM se diseñaron para conectividad de área amplia. Sin embargo, muchas empresas, universidades y otras organizaciones tienen un gran número de computadoras que requieren interconexión. Esta necesidad dio origen a la red de área local. En esta sección diremos algo sobre la LAN más popular: Ethernet.

La historia empieza en la prístina Hawaii a principios de la década de 1970. En este caso, “prística” se puede interpretar como “que no tiene un sistema telefónico funcional”. En tanto los días son más agradables para los vacacionistas cuando no son interrumpidos por el teléfono, no fue así para el investigador Norman Abramson y sus colegas de la Universidad de Hawaii, quienes estuvieron tratando de conectar usuarios de las islas remotas a la computadora principal de Hono-

lulu. Conectar sus propios cables bajo el Océano Pacífico parecía imposible, de modo que buscaron una solución diferente.

La primera que encontraron fueron los radios de onda corta. Cada terminal estaba equipada con un radio pequeño de dos frecuencias: un canal ascendente (a la computadora central) y otro descendente (desde la computadora central). Cuando el usuario deseaba conectarse con la computadora, sólo transmitía por el canal ascendente un paquete que contenía los datos. Si en ese instante nadie más estaba transmitiendo, probablemente el paquete saldría y su recepción sería confirmada en el canal descendente. Si había contención por el canal ascendente, la terminal detectaría la falta de confirmación de recepción y haría otro intento. Puesto que sólo habría un emisor en el canal descendente (la computadora central), nunca habría colisiones ahí. Este sistema, llamado ALOHANET, trabajaba muy bien en condiciones de bajo tráfico pero se caía cuando el flujo de tráfico ascendente era pesado.

En esa misma época, un estudiante llamado Bob Metcalfe hizo su licenciatura en el M.I.T. y luego se mudó para obtener su doctorado en Harvard. Durante sus estudios, conoció el trabajo de Abramson. Se interesó tanto en él que después de graduarse en Harvard decidió pasar el verano en Hawaii trabajando con Abramson antes de empezar a trabajar en el Centro de Investigación de Palo Alto de Xerox (PARC). Cuando llegó al PARC, vio que los investigadores de ahí habían diseñado y construido lo que más tarde se llamarían computadoras personales. Pero las máquinas estaban aisladas. Aplicando su conocimiento del trabajo de Abramson, junto con su colega David Boggs, diseñó e implementó la primera red de área local (Metcalfe y Boggs, 1976).

Llamaron **Ethernet** al sistema, por lo de *luminiferous ether*, a través del cual se pensó alguna vez que se propagaba la radiación electromagnética. (Cuando, en el siglo XIX, el físico inglés James Clerk Maxwell descubrió que la radiación electromagnética se podía describir mediante una ecuación de onda, los científicos supusieron que el espacio debía estar lleno de algún medio etéreo en el cual se propagaba la radiación. Sólo después del famoso experimento de Michelson-Morley en 1887, los físicos descubrieron que la radiación electromagnética se podía propagar por el vacío.)

Aquí el medio de transmisión no era el vacío, sino un cable coaxial grueso (el éter) de más de 2.5 km de largo (con repetidoras cada 500 metros). El sistema podía contener hasta 256 máquinas por medio de transceptores acoplados al cable. Un cable con múltiples máquinas en paralelo se llama **cable de derivación múltiple** (*multidrop*). El sistema se ejecutaba a 2.94 Mbps. En la figura 1-34 se presenta un esbozo de su arquitectura. Ethernet tenía una mejora importante respecto de ALOHANET; antes de transmitir, una computadora tenía que escuchar el cable para ver si había alguien más transmitiendo. En caso de que ya lo hubiera, la computadora se mantenía en espera de que la transmisión actual terminara. Al hacerlo así se evitaba interferir con las transmisiones existentes, dando una mayor eficiencia. ALOHANET no trabajaba de este modo porque para una terminal en una isla era imposible detectar la transmisión de otra terminal en una isla distante. El problema se resolvía con un cable único.

A pesar de que la computadora escucha antes de transmitir, surge un problema: ¿qué sucede si dos o más computadoras esperan hasta que se complete la transmisión actual y luego empiezan

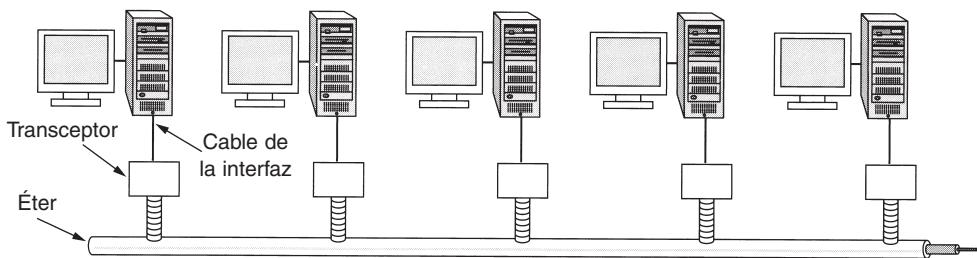


Figura 1-34. Arquitectura de la Ethernet original.

a transmitir al mismo tiempo? La solución es que cada computadora escuche durante su propia transmisión y, si detecta interferencia, mande una señal para poner en alerta a todos los transmisores. Después espera un tiempo aleatorio antes de reintentarlo. Si sucede una colisión, el tiempo aleatorio de espera se duplica y así sucesivamente, para separar las transmisiones que están en competencia y dar a alguna la oportunidad de transmitir primero.

La Ethernet de Xerox fue tan exitosa que DEC, Intel y Xerox diseñaron un estándar en 1978 para una Ethernet de 10 Mbps, llamado **estándar DIX**. Con dos cambios menores, en 1983 el estándar DIX se convirtió en el estándar IEEE 802.3.

Por desgracia para Xerox, ya tenía fama de hacer inventos originales (como el de las computadoras personales) y luego fallar en la comercialización de los mismos, como se menciona en un relato titulado *Fumbling the Future* (Smith y Alexander, 1988). Cuando Xerox mostró poco interés en hacer algo con Ethernet aparte de ayudar a estandarizarlo, Metcalfe formó su propia empresa, 3Com, con el propósito de vender adaptadores Ethernet para PCs. Ha vendido más de 100 millones.

Ethernet continuó su desarrollo y aún está en desarrollo. Han salido nuevas versiones a 100 y 1000 Mbps, e incluso más altas. También se ha mejorado el cableado y se han agregado commutación y otras características. En el capítulo 4 explicaremos Ethernet en detalle.

De paso, vale la pena mencionar que Ethernet (IEEE 802.3) no es el único estándar de LAN. El comité también estandarizó Token Bus (802.4) y Token Ring (802.5). La necesidad de tres estándares más o menos incompatibles tiene poco que ver con la tecnología y mucho con la política. En el momento de la estandarización, General Motors estaba impulsando una LAN en la que la topología era la misma que la usada en Ethernet (un cable linear), pero las computadoras transmitían por turnos pasando un pequeño paquete de computadora a computadora, llamado **token**. Una computadora podía transmitir sólo si poseía el *token*, lo que evitaba colisiones. General Motors anunció que este esquema era esencial para la manufactura de automóviles y que no estaba preparado para cambiar su postura. No obstante este anuncio, el 802.4 prácticamente desapareció.

Del mismo modo, IBM tenía su favorito: su Token Ring patentado. En este esquema el *token* se pasaba a través del anillo y la computadora que poseyera el *token* podía transmitir antes de poner el *token* de nuevo en el anillo. A diferencia del 802.4, este esquema, estandarizado como 802.5,

aún se usa en algunos sitios de IBM, pero prácticamente en ninguna parte más. Sin embargo, se está desarrollando una versión de 1 gigabit (802.5v), pero parece poco probable que alcance a Ethernet. Resumiendo, había una guerra entre Ethernet, Token Bus y Token Ring, pero Ethernet ganó, en gran medida porque fue la primera y los retadores no pudieron superarlo.

1.5.4 LANs inalámbricas: 802.11

Casi al mismo tiempo que aparecieron las computadoras portátiles, muchas personas tuvieron el sueño de andar por la oficina y poder conectar a Internet su computadora. En consecuencia, varios grupos empezaron a trabajar para cumplir con esta meta. El método más práctico es equipar las computadoras de la oficina y las portátiles con transmisores y receptores de radio de onda corta que les permitan comunicarse. Este trabajo condujo rápidamente a que varias empresas empezaran a comercializar las LANs inalámbricas.

El problema es que no había compatibilidad entre ninguna de ellas. Esta proliferación de estándares implicaba que una computadora equipada con un radio de marca *X* no funcionara en un cuarto equipado con una estación de base marca *Y*. Finalmente, la industria decidió que un estándar de LAN inalámbrica sería una buena idea, por lo que al comité del IEEE que estandarizó las LANs alámbricas se le encargó la tarea de diseñar un estándar para LANs inalámbricas. El estándar resultante se llamó 802.11. En la jerga común se le conoce como **WiFi**. Es un estándar importante y merece respeto, así que lo llamaremos por su nombre propio, 802.11.

El estándar propuesto tenía que trabajar en dos modos:

1. En presencia de una estación base.
2. En ausencia de una estación base.

En el primer caso, toda la comunicación se hacía a través de la estación base, que en la terminología del 802.11 se conoce como **punto de acceso**. En el segundo caso, las computadoras podrían enviarse mensajes entre sí directamente. Este modo se llama a veces **red ad hoc**. Un ejemplo típico es el de dos o más personas que se encuentran juntas en un cuarto no equipado con una LAN inalámbrica y cuyas computadoras se comunican entre sí de manera directa. Los dos modos se ilustran en la figura 1-35.

La primera decisión fue la más sencilla: cómo llamarlo. Todos los otros estándares LAN tenían números como 802.1, 802.2, hasta 802.10, por lo que el estándar LAN se llamó o publicó como 802.11. El resto fue más difícil.

En particular, varios de los diversos retos que había que enfrentar eran: encontrar una banda de frecuencia adecuada, de preferencia mundial; enfrentar el hecho de que las señales de radio tienen un rango finito; asegurarse de que se mantuviera la privacidad de los usuarios; tomar en cuenta la vida limitada de las baterías; preocuparse por la seguridad humana (*¿las ondas de radio causan cáncer?*); comprender las implicaciones de la movilidad de las computadoras y, por último, construir un sistema con suficiente ancho de banda para que sea económicamente viable.

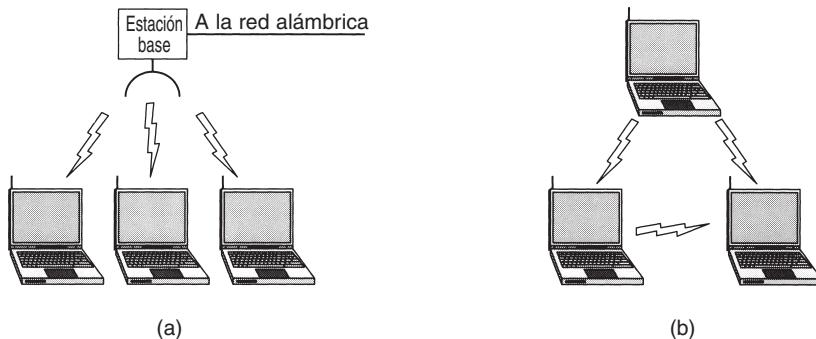


Figura 1-35. (a) Red inalámbrica con una estación base. (b) Red *ad hoc*.

Cuando empezó el proceso de estandarización (a mediados de la década de 1990), Ethernet ya había llegado a dominar las redes de área local, por lo que el comité decidió hacer que el 802.11 fuera compatible con Ethernet sobre la capa de enlace de datos. En particular, se podría enviar un paquete IP sobre la LAN inalámbrica del mismo modo en que una computadora conectada mediante cable enviaba un paquete IP a través de Ethernet. No obstante, existen algunas diferencias inherentes con Ethernet en las capas física y de enlace de datos y tuvieron que manejarse mediante el estándar.

Primero, una computadora en Ethernet siempre escucha el medio antes de transmitir. Sólo si el medio está inactivo la computadora puede empezar a transmitir. Esta idea no funciona igual en las LANs inalámbricas. Para ver por qué, examine la figura 1-36. Suponga que la computadora *A* está transmitiendo a la computadora *B*, pero el alcance del radio del transmisor de *A* es muy corto para encontrar a la computadora *C*. Si *C* desea transmitir a *B* puede escuchar el medio antes de empezar, pero el hecho de que no escuche nada no quiere decir que su transmisión tendrá éxito. El estándar 802.11 tenía que resolver este problema.

El segundo problema que se tenía que resolver es que los objetos sólidos pueden reflejar una señal de radio, por lo que ésta se podría recibir múltiples veces (a través de varias rutas). Esta interferencia da como resultado lo que se llama **desvanecimiento por múltiples trayectorias**.

El tercer problema es que una gran cantidad de software no toma en cuenta la movilidad. Por ejemplo, muchos procesadores de texto tienen una lista de impresoras de entre las cuales los usuarios pueden elegir para imprimir un archivo. Cuando la computadora en la que se ejecuta el procesador de texto se coloca en un nuevo entorno, la lista interna de impresoras ya no es útil.

El cuarto problema es que si una computadora portátil se mueve lejos de la estación base que está usando y dentro del rango de una estación base diferente, se requiere algún tipo de manejo. Aunque este problema ocurre con los teléfonos celulares, eso no sucede con Ethernet y requiere solución. En particular, la red prevista consta de múltiples celdas, cada una con su propia estación base pero con las estaciones base conectadas por Ethernet, como se muestra en la figura 1-37. Desde fuera todo el sistema se vería como una Ethernet sola. La conexión entre el sistema 802.11 y el mundo exterior se conoce como **portal**.

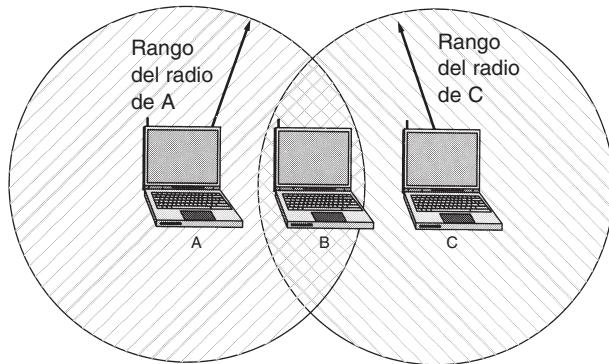


Figura 1-36. El rango de un solo radio no podría cubrir todo el sistema.

Después de algún trabajo, el comité se presentó en 1997 con un estándar que se dirigía a éstos y otros respectos. La LAN inalámbrica descrita se ejecutaba a 1 o 2 Mbps. Casi de inmediato la gente comenzó a quejarse de que era demasiado lenta, de manera que empezaron a trabajar en estándares más rápidos. Una división desarrollada con el comité tuvo como resultado dos nuevos estándares en 1999. El estándar 802.11a utiliza una banda de frecuencia más ancha y se ejecuta a velocidades de hasta 54 Mbps. El estándar 802.11b utiliza la misma banda de frecuencia que el 802.11, pero se vale de una técnica de modulación diferente para alcanzar 11 Mbps. Algunas personas ven esto como un aspecto psicológico importante puesto que 11 Mbps es más rápido que la Ethernet alámbrica original. Es posible que el 802.11 original de 1 Mbps desaparezca con rapidez, pero aún no queda claro cuál de los nuevos estándares será el ganador.

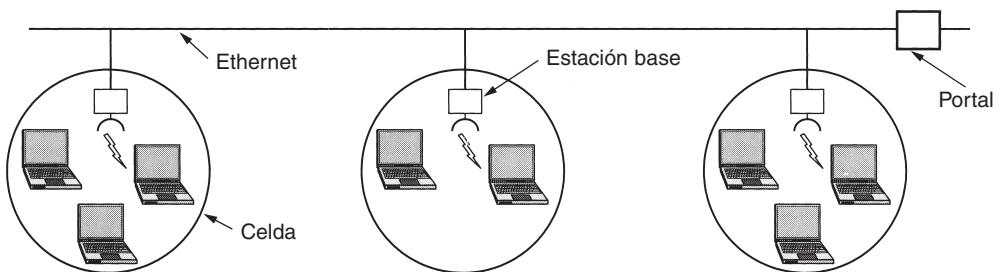


Figura 1-37. Una red 802.11 de múltiples celdas.

Para hacer las cosas todavía más complicadas, el comité 802 ha creado otra variante, el 802.11g, que utiliza la técnica de modulación del 802.11a pero la banda de frecuencia del 802.11b. En el capítulo 4 trataremos en detalle al 802.11.

Sin lugar a dudas, el 802.11 va a causar una revolución en computación y en el acceso a Internet. Aeropuertos, estaciones de trenes, hoteles, centros comerciales y universidades lo están instalando rápidamente. Incluso cafeterías de lujo están instalando el 802.11 para que los *yuppies* que se reúnen puedan navegar en Web mientras toman su café con leche. Es posible que el 802.11 haga por Internet lo que las computadoras portátiles hicieron por la computación: volverla móvil.

1.6 ESTANDARIZACIÓN DE REDES

Existen muchos fabricantes y proveedores de redes, cada uno con sus propias ideas de cómo se deben hacer las cosas. Sin coordinación sería un caos total y los usuarios nunca conseguirían nada. La única manera de resolver esta situación es ponerse de acuerdo en la adopción de algunos estándares para redes.

Los estándares no sólo permiten que computadoras diferentes se comuniquen, sino que también incrementan el mercado de productos que se ajustan al estándar. Un mercado grande conduce a la producción masiva, economías de escala en la producción, implementaciones VLSI y otros beneficios que disminuyen el precio e incrementan aún más la aceptación. En las siguientes secciones daremos un vistazo al importante, pero poco conocido, mundo de la estandarización internacional.

Los estándares se dividen en dos categorías: de facto y de jure. Los estándares *de facto* (“de hecho”) son los que simplemente surgieron, sin ningún plan formal. La PC de IBM y sus sucesoras son estándares de facto para oficinas chicas y equipos domésticos porque docenas de fabricantes decidieron copiar exactamente las máquinas de IBM. Del mismo modo, UNIX es el estándar de facto para sistemas operativos en los departamentos de ciencias de la computación de las universidades.

En contraste, los estándares *de jure* (“por derecho”), son formales, legales, adoptados por alguna institución de estandarización autorizada. Por lo general, las autoridades de estandarización internacional se dividen en dos clases: las establecidas por acuerdos entre los gobiernos de cada país, y las incluidas de manera voluntaria, sin acuerdos entre organizaciones. En el área de los estándares de redes de computadoras hay varias organizaciones de cada tipo, las cuales explicaremos a continuación.

1.6.1 Quién es quién en el mundo de las telecomunicaciones

La situación legal de las compañías telefónicas del mundo varía considerablemente de un país a otro. En un extremo están los Estados Unidos con sus 1500 empresas telefónicas privadas individuales. Antes de que AT&T se dividiera, en 1984, era la empresa más grande del mundo y dominaba el escenario. Proporcionaba servicio telefónico a casi 80% de los usuarios de Estados

Unidos, con sucursales diseminadas en la mitad del país; las compañías oponentes atendían al resto de los clientes (en su mayor parte rurales). Desde que se dividió, AT&T sigue proporcionando servicio de larga distancia, pero ahora en competencia con otras empresas. Las siete Compañías Operadoras Regionales de Bell que surgieron de la división de AT&T y numerosas empresas independientes proporcionan servicios de telefonía local y celular. Debido a las frecuentes fusiones y otros cambios, la industria está en un estado de movimiento constante.

Las empresas que dan servicios de comunicación al público en Estados Unidos se llaman **portadoras comunes** (*carriers*). Las ofertas y precios se describen en un documento llamado **tarifa**, el cual debe ser aprobado por la Comisión Federal de Comunicaciones para el tráfico interestatal e internacional, pero el tráfico estatal interno lo aprueban las comisiones de servicios públicos.

En el otro extremo están los países en los cuales el gobierno respectivo tiene el monopolio de todas las comunicaciones, como correos, telégrafos, teléfonos y a veces la radio y la televisión. La mayor parte del mundo cae dentro de esta categoría. En algunos casos la autoridad de la telecomunicación es una compañía nacionalizada y en otros es simplemente una rama del gobierno, conocida generalmente como **PTT** (administración de **Correos, Telégrafos y Teléfonos**). La tendencia a nivel mundial es hacia una liberación y competencia, y alejarse del monopolio gubernamental. La mayoría de los países europeos tiene privatizadas (parcialmente) sus PTTs, pero en otras partes el proceso avanza con lentitud.

Con tantos proveedores diferentes de servicios, es claro que se necesita una compatibilidad a escala mundial para asegurarse de que las personas (y las computadoras) de un país puedan llamar a sus contrapartes en otro. En realidad, esta necesidad ha existido desde hace mucho tiempo. En 1865, los representantes de muchos gobiernos de Europa se reunieron para formar el predecesor de la actual **ITU (Unión Internacional de Telecomunicaciones)**. Su trabajo era estandarizar las telecomunicaciones internacionales, que en esos días se hacían mediante el telégrafo. Incluso entonces era patente que si la mitad de los países utilizaba el código Morse y la otra utilizaba un código diferente, surgiría un problema. Cuando el teléfono entró al servicio internacional, la ITU empezó a trabajar en la estandarización de la telefonía. En 1947 la ITU se convirtió en una agencia de las Naciones Unidas.

La ITU tiene tres sectores principales:

1. Radiocomunicaciones (ITU-R).
2. Estandarización de telecomunicaciones (ITU-T).
3. Desarrollo (ITU-D).

La ITU-R se ocupa de asignar frecuencias de radio en todo el mundo a los grupos de interés en competencia. Nos enfocaremos en primer lugar en la ITU-T, que se ocupa de los sistemas telefónicos y de comunicación de datos. De 1956 a 1993, la ITU-T se conocía como **CCITT** (del francés *Comité Consultatif International Télégraphique et Téléphonique*, Comité Consultivo Internacional para la Telegrafía y Telefonía). El 1o. de marzo de 1993 el CCITT se reorganizó para hacerlo menos burocrático y cambió de nombre para reflejar su nuevo papel. Tanto la ITU-T como el CCITT emitieron recomendaciones en el área de comunicaciones telefónicas y de datos.

Es frecuente encontrar algunas de las recomendaciones del CCITT, como la X.25 del CCITT, aunque desde 1993 las recomendaciones llevan la etiqueta de la ITU-T.

La ITU-T tiene cuatro clases de miembros:

1. Gobiernos nacionales.
2. De sector.
3. Asociados.
4. Agencias reguladoras.

La ITU-T tiene alrededor de 200 miembros gubernamentales, entre ellos casi todos los miembros de las Naciones Unidas. Puesto que Estados Unidos no tiene una PTT, alguien más tenía que representarlos en la ITU-T. Esta tarea recayó en el Departamento de Estado, probablemente porque la ITU-T tenía que ver con los países extranjeros, que era la especialidad del Departamento de Estado.

Hay aproximadamente 500 miembros de sector, incluyendo compañías telefónicas (por ejemplo, AT&T, Vodafone, WorldCom), fabricantes de equipos de telecomunicación (como Cisco, Nokia, Nortel), fabricantes de computadoras (como Compaq, Sun, Toshiba), fabricantes de chips (como Intel, Motorola, TI), compañía de medios (como AOL Time Warner, CBS, Sony) y otras empresas interesadas (como Boeing, Samsung, Xerox). Varias organizaciones científicas no lucrativas y consorcios industriales también son miembros de sector (por ejemplo, IFIP e IATA). Los miembros asociados son organizaciones más pequeñas que se interesan en un grupo de estudio en particular. Las agencias reguladoras son quienes vigilan el negocio de la telecomunicación, como la Comisión Federal de Comunicaciones de Estados Unidos.

La tarea de la ITU-T es hacer recomendaciones técnicas sobre telefonía, telegrafía y las interfaces de comunicación de datos. Estas recomendaciones suelen convertirse en estándares reconocidos internacionalmente, por ejemplo el V.24 (también conocido en Estados Unidos como EIA RS-232), el cual especifica la ubicación y significado de los diversos pines en el conector utilizado para la mayoría de las terminales asíncronas y módems externos.

Es preciso observar que las recomendaciones de la ITU-T técnicamente son sólo sugerencias que los gobiernos pueden adoptar o ignorar (ya que los gobiernos parecen adolescentes de 13 años a quienes no les gusta recibir órdenes). En la práctica, un país que desee adoptar un estándar telefónico diferente del utilizado por el resto del mundo, es libre de hacerlo, pero el precio es el aislamiento. Esto podría funcionar en Corea del Norte, pero fuera de ahí sería un verdadero problema. El sofisma de llamar “recomendaciones” a los estándares de la ITU-T era y es necesario para mantener en calma el nacionalismo de varios países.

El trabajo verdadero de la ITU-T se realiza en sus 14 grupos de estudio, a veces de hasta 400 personas, que abarcan aspectos que van desde la facturación telefónica hasta servicios de multimedia. Para conseguir la realización de los proyectos, los grupos de estudio se dividen en equipos de trabajo, que a su vez se dividen en equipos de expertos, que a su vez se dividen en grupos específicos. Una vez burócrata, jamás se deja de serlo.

A pesar de todo esto, en realidad la ITU-T hace su trabajo. Desde que surgió, ha producido cerca de 3000 recomendaciones que ocupan cerca de 60,000 páginas de papel. Muchas de ellas se han llevado a la práctica en gran medida. Por ejemplo, una de sus recomendaciones es el popular estándar V.90 para módems de 56 kbps.

En tanto las comunicaciones completen la transición, que empezó en la década de 1980, de ser nacionales totalmente a ser globales totalmente, los estándares llegarán a ser más importantes cada vez, y más y más organizaciones querrán estar implicadas en su establecimiento. Para más información sobre la ITU, vea (Irmer, 1994).

1.6.2 Quién es quién en los estándares internacionales

Los estándares internacionales son producidos y publicados por la **ISO (Organización de Estándares Internacionales)**,[†] una organización voluntaria no surgida de un acuerdo, fundada en 1946. Sus miembros son las organizaciones de estándares nacionales de los 89 países miembro. Entre ellos se encuentran ANSI (Estados Unidos), BSI (Gran Bretaña), AFNOR (Francia), DIN (Alemania) y otros 85.

La ISO emite estándares sobre una gran cantidad de temas, desde los más básicos (literalmente) como tuercas y pernos, hasta el revestimiento de los postes telefónicos (sin mencionar las semillas de cacao [ISO 2451], las redes de pesca [ISO 1530], ropa interior femenina [ISO 4416] y algunos otros objetos que no se pensaría que fueran sujetos de estandarización). Se han emitido más de 13,000 estándares, entre ellos los estándares de OSI. La ISO tiene casi 200 comités técnicos, numerados por el orden de su creación, refiriéndose cada uno a un objeto específico. El TC1 se ocupa de las tuercas y pernos (estandariza la rosca de los tornillos). El TC97 trata con computadoras y procesamiento de información. Cada TC tiene subcomités (SCs) divididos en grupos de trabajo (WGs).

El trabajo real lo hacen sobre todo los WGs, integrados por más de 100,000 voluntarios en todo el mundo. Muchos de estos “voluntarios” son asignados a trabajar en asuntos de la ISO por sus empleadores, cuyos productos se están estandarizando. Otros son oficiales gubernamentales ansiosos de que lo que se hace en su país llegue a ser estándar internacional. Los expertos académicos también están activos en muchos de los WGs.

En cuanto a estándares de telecomunicación, la ISO y la ITU-T suelen cooperar (la ISO es miembro de la ITU-T), para evitar la ironía de dos estándares internacionales oficiales mutuamente incompatibles.

El representante de Estados Unidos en la ISO es el **ANSI (Instituto Estadounidense de Estándares Nacionales)**, que a pesar de su nombre es una organización privada no gubernamental y no lucrativa. Sus miembros son fabricantes, empresas portadoras comunes y otras partes interesadas. La ISO suele adoptar los estándares ANSI como estándares internacionales.

El procedimiento seguido por la ISO para adoptar estándares se ha diseñado para obtener el mayor consenso posible. El proceso inicia cuando alguna de las organizaciones de estándares

[†]Para los puristas, el verdadero nombre de la ISO es Organización Internacional para la Estandarización.

nacionales siente la necesidad de un estándar internacional en un área determinada. Entonces se forma un grupo de trabajo para presentar un **CD (Borrador de Comité)**. El CD se distribuye a todos los miembros, que tienen seis meses para criticarlo. Si la mayoría lo aprueba, se revisa y distribuye un documento revisado, llamado **DIS (Borrador de Estándar Internacional)** para comentarios y votación. Con base en los resultados de esta vuelta, se prepara, aprueba y publica el texto final del **IS (Estándar Internacional)**. En áreas de gran controversia, un CD o un DIS podría llegar a tener varias versiones antes de lograr suficientes votos y todo el proceso puede llegar a tardar años.

El **NIST (Instituto Nacional de Estándares y Tecnología)** es parte del Departamento de Comercio de Estados Unidos. Se llamaba Oficina Nacional de Estándares. Emite estándares que son obligatorios para compras hechas por el gobierno de Estados Unidos, excepto por los del Departamento de Defensa, que tiene sus propios estándares.

Otro representante importante en el mundo de los estándares es el **IEEE (Instituto de Ingenieros Eléctricos y Electrónicos)**, la mayor organización de profesionales del mundo. Además de publicar multitud de periódicos y organizar cientos de conferencias cada año, el IEEE tiene un grupo de estandarización que desarrolla estándares en el área de ingeniería eléctrica y computación. El comité 802 del IEEE ha estandarizado muchos tipos de LANs. Estudiaremos algunos de sus resultados más adelante. El trabajo real lo hace un conjunto de grupos de trabajo, que se listan en la figura 1.38. La tasa de éxito de los diversos grupos de trabajo del 802 ha sido baja; el hecho de tener un número 802.x no garantiza el éxito. Pero el impacto de las historias de éxito (en especial, del 802.3 y el 802.11) ha sido tremendo.

1.6.3 Quién es quién en el mundo de los estándares de Internet

El amplio mundo de Internet tiene sus propios mecanismos de estandarización, muy diferentes de los de la ITU-T y la ISO. La diferencia se puede resumir diciendo que quienes asisten a las reuniones de estandarización de la ITU o la ISO van de traje, pero las personas que asisten a las juntas de estandarización de Internet van de mezclilla (excepto cuando se reúnen en San Diego, donde van de *short* y camiseta).

En las reuniones de la ITU y la ISO abundan los oficiales corporativos y burócratas, para quienes la estandarización es su trabajo. Se refieren a la estandarización como una Cosa Buena y dedican sus vidas a ella. Por otro lado, la gente de Internet prefiere la anarquía por cuestión de principios. Sin embargo, con cientos de millones de personas haciendo sus propias cosas, la comunicación es escasa. Por lo tanto, los estándares, aunque deplorables, son necesarios.

Cuando se configuró ARPANET, el DoD creó un comité informal para supervisarla. En 1983 se dio otro nombre al comité: **IAB (Consejo de Actividades de Internet)** y se le encendió una misión un poco más amplia, que era la de mantener a los investigadores de ARPANET y de Internet apuntando más o menos en la misma dirección; algo muy parecido a juntar una manada de gatos. El significado del acrónimo “IAB” se cambió a **Consejo para la Arquitectura de Internet**.

Número	Tema
802.1	Supervisión y arquitectura de LANs
802.2 ↓	Control lógico de enlace
802.3 *	Ethernet
802.4 ↓	Token bus (se utilizó por un corto tiempo en plantas manufactureras)
802.5	Token ring (entrada de IBM al mundo de las LANs)
802.6 ↓	Cola dual, bus dual (primera red de área metropolitana)
802.7 ↓	Grupo de consultoría técnico de tecnologías de banda ancha
802.8 †	Grupo de consultoría de tecnologías de fibra óptica
802.9 ↓	LANs síncrona (para aplicaciones de tiempo real)
802.10 ↓	LANs virtuales y seguridad
802.11 *	LANs inalámbricas
802.12 ↓	Demanda de prioridad (AnyLAN de Hewlett-Packard)
802.13	Número de mala suerte. Nadie lo quiso
802.14 ↓	Módems de cable (desaparecido: primero surgió un consorcio en la industria)
802.15 *	Redes de área personal (Bluetooth)
802.16 *	Redes inalámbricas de área ancha
802.17	Anillo de paquete elástico

Figura 1-38. Los grupos de trabajo del 802. Los importantes se marcan con *. Los que se marcan con ↓ están en hibernación. El que tiene la † se desintegró.

Cada uno de los aproximadamente 10 miembros del IAB encabezaba una fuerza de trabajo relacionada con algún asunto importante. El IAB se reunía varias veces al año para discutir los resultados y para dar retroalimentación al DoD y a la NSF, que proporcionaban la mayor parte de los fondos en aquel entonces. Cuando se necesitaba un estándar (por ejemplo, un nuevo algoritmo de enrutamiento), los miembros del IAB le daban solución y después anuncianan los cambios para que los estudiantes que estuvieran a cargo de la implementación del software pudieran realizarlos. La comunicación se llevaba a cabo mediante una serie de informes técnicos denominados **RFCs (Solicitudes de Comentarios)**. Estos informes se almacenan en línea y cualquiera que esté interesado en ellos puede descargarlos de www.ietf.org/rfc. Los RFCs se encuentran organizados por el orden cronológico de su creación. Actualmente existen alrededor de 3000. En este libro mencionaremos muchos RFCs.

Para 1989 Internet había crecido tanto que este estilo sumamente informal dejó de ser funcional. Muchos fabricantes ofrecían productos de TCP/IP en ese entonces y no deseaban cambiarlos tan sólo porque 10 investigadores habían concebido una mejor idea. El IAB fue reorganizado de nueva cuenta en el verano de 1989. Los investigadores fueron transferidos a la **IRTF (Fuerza de Trabajo para la Investigación sobre Internet)**, que fue puesta bajo el mando del IAB, junto con la **IETF (Fuerza de Trabajo para la Ingeniería de Internet)**. El IAB se renovó con nuevos

miembros, que representaban a un rango más amplio de organizaciones que tan sólo a la comunidad de investigadores. Inicialmente fue un grupo que se autorrenovaba, cuyos miembros servían durante dos años y ellos mismos designaban a sus sucesores. Más tarde se creó la **Sociedad de Internet**, integrada por gente interesada en Internet. En cierto sentido, esta sociedad se asemeja al ACM o al IEEE. Es dirigida por administradores electos que designan a los miembros del IAB.

El propósito de esta división era que la IRTF se concentrara en proyectos de investigación a largo plazo, en tanto que la IETF se encargaba de proyectos de ingeniería a corto plazo. La IETF se dividió en grupos de trabajo, cada uno a cargo de un problema específico. Inicialmente, los líderes de cada grupo se reunían en un comité para dirigir los proyectos de ingeniería. Entre los temas de los grupos de trabajo están nuevas aplicaciones, información de usuario, integración OSI, enrutamiento y direccionamiento, seguridad, administración de redes y estándares. Con el tiempo se formaron tantos grupos de trabajo (más de 70), que se ordenaron por áreas y el líder de cada área formaba parte del comité directivo.

Además, se adoptó un proceso de estandarización más formal, con base en el ISO. Para convertirse en **Estándar Propuesto**, la idea fundamental debía explicarse completamente en un RFC y despertar suficiente interés en la comunidad. Para avanzar a la etapa de **Estándar Borrador**, una implementación funcional debía haber sido rigurosamente probada por al menos dos sitios independientes durante al menos cuatro meses. Si el IAB se convence de que la idea suena lógica y el software funciona, declara que el RFC es un Estándar de Internet. Algunos de estos estándares se han convertido en estándares del DoD (MIL-STD), con lo cual son obligatorios para los proveedores del DoD. En cierta ocasión, David Clark hizo el siguiente comentario, ahora famoso, acerca del proceso de estandarización de Internet: “consenso apretado y código que funcione”.

1.7 UNIDADES MÉTRICAS

Para evitar confusiones, vale la pena indicar de manera explícita que en este libro, como en las ciencias de la computación en general, se utilizan unidades métricas en lugar de las unidades inglesas tradicionales. En la figura 1.39 se muestran los principales prefijos del sistema métrico. Por lo general, los prefijos se abrevian mediante sus primeras letras, con las unidades mayores que uno en mayúsculas (KB, MB, etcétera). Una excepción (por razones históricas) es kbps, de kilobits por segundo. Por lo tanto, una línea de comunicación de 1 Mbps transmite 10^6 bits por segundo y un reloj de 100 pseg (o 100 ps) marca cada 10^{-10} segundos. Dado que tanto mili como micro empiezan con la letra “m”, se tiene que hacer una distinción. Por lo general, “m” es para mili y “μ” (la letra griega mu) es para micro.

También vale la pena señalar que para medir el tamaño de la memoria, de disco, de archivo y de bases de datos, en la práctica común de la industria de la computación las unidades tienen equivalencias ligeramente distintas. En ésta, kilo equivale a 2^{10} (1024) en vez de 10^3 (1000) porque las memorias siempre son potencias de dos. De esta forma, 1 KB de memoria son 1024 bytes, no 1000 bytes. De manera similar, 1 MB de memoria son 2^{20} (1,048,576) bytes, 1 GB de memoria son 2^{30} (1,073,741,824) bytes, y 1 TB de base de datos son 2^{40} (1,099,511,627,776) bytes. No obstante, una línea de comunicación de 1 kbps transmite 1000 bits por segundo y una LAN de

Exp.	Explícito	Prefijo	Exp.	Explícito	Prefijo
10^{-3}	0.001	mili	10^3	1,000	Kilo
10^{-6}	0.000001	micro	10^6	1,000,000	Mega
10^{-9}	0.000000001	nano	10^9	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.00000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.0000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zeta
10^{-24}	0.00000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

Figura 1-39. Los principales prefijos métricos.

10 Mbps corre a 10,000,000 de bits por segundo debido a que estas velocidades no son potencias de dos. Desgraciadamente, mucha gente mezcla estos dos sistemas, en particular en lo referente a los tamaños de disco. Para evitar la ambigüedad, en este libro utilizaremos los símbolos KB, MB y GB para 2^{10} , 2^{20} y 2^{30} bytes, respectivamente, y los símbolos kbps, Mbps y Gbps para 10^3 , 10^6 y 10^9 bits por segundo, respectivamente.

1.8 PANORAMA DEL RESTO DEL LIBRO

Este libro estudia tanto los principios como la práctica de las redes de computadoras. La mayoría de los capítulos inician con un análisis de los principios relevantes, seguido por diversos ejemplos que ilustran estos principios. Por lo general, los ejemplos se toman de Internet y de las redes inalámbricas puesto que ambos son importantes y muy distintos. Donde es necesario, se dan otros ejemplos.

El libro se estructura de acuerdo con el modelo híbrido que se presenta en la figura 1-24. El análisis de la jerarquía de protocolos empieza en el capítulo 2, a partir de la capa más baja. El segundo capítulo proporciona algunos antecedentes en el campo de la comunicación de datos. Se presentan sistemas alámbricos, inalámbricos y satelitales. Este material se relaciona con la capa física, aunque veremos únicamente los aspectos de arquitectura y no los de hardware. También se analizan numerosos ejemplos de la capa física, como la red telefónica pública conmutada, la telefonía celular y la red de televisión por cable.

En el capítulo 3 se presenta la capa de enlace de datos y sus protocolos a través de ejemplos que crecen en complejidad. También se cubre el análisis de estos protocolos. Más tarde se examinan algunos protocolos importantes que se usan con mucha frecuencia, entre ellos HDLC (que se emplea en redes de baja y mediana velocidad) y PPP (que se utiliza en Internet).

El capítulo 4 tiene que ver con la subcapa de acceso al medio, que forma parte de la capa de enlace de datos. El aspecto principal al que se enfrenta esta subcapa es cómo determinar quién uti-

lizará la red cuando ésta consiste en un solo canal compartido, como ocurre en la mayoría de las LANs y en algunas redes satelitales. Se dan muchos ejemplos de LANs alámbricas, LANs inalámbricas (en especial Ethernet), MANs inalámbricas, Bluetooth y redes satelitales. También se analizan los puentes y los conmutadores de enlace de datos.

El capítulo 5 aborda la capa de red, en particular el enrutamiento, con muchos algoritmos de enrutamiento, tanto estáticos como dinámicos. Aun con el uso de buenos algoritmos de enrutamiento, si existe más tráfico del que puede manejar la red, se genera congestión, por lo que analizaremos el tema de la congestión y cómo evitarla. Es aún mejor garantizar una calidad específica en el servicio que tan sólo evitar la congestión. También analizaremos este punto. La conexión de redes heterogéneas para conformar interredes acarrea numerosos problemas que también examinaremos. Daremos una amplia cobertura a la capa de red en Internet.

El capítulo 6 se encarga de la capa de transporte. Gran parte del capítulo se dedica a los protocolos orientados a la conexión, puesto que muchas aplicaciones los necesitan. Se analiza en detalle un ejemplo de servicio de transporte y su implementación. Se proporciona el código para este sencillo ejemplo con el propósito de mostrar cómo se puede implementar. Tanto UDP como TCP, protocolos de transporte de Internet, se abordan en detalle, al igual que sus aspectos de desempeño. Asimismo, veremos aspectos relacionados con las redes inalámbricas.

El capítulo 7 presenta la capa de aplicación, sus protocolos y aplicaciones. El primer tema es el DNS, que es el directorio telefónico de Internet. A continuación trataremos el correo electrónico, junto con un análisis de sus protocolos. Más adelante pasaremos a Web, con explicaciones minuciosas sobre contenido estático, contenido dinámico, lo que sucede tanto en el cliente como en el servidor, protocolos, rendimiento, la Web inalámbrica, entre otros temas. Por último, examinaremos la multimedia en red, con temas como audio de flujo continuo, radio en Internet y vídeo bajo demanda.

El capítulo 8 se relaciona con la seguridad de red. Este tema tiene aspectos que se relacionan con todas las capas, por lo cual es más sencillo abordarlo después de haber explicado minuciosamente todas las capas. El capítulo inicia con una introducción a la criptografía. Más adelante muestra cómo se puede utilizar ésta para garantizar la seguridad en las comunicaciones, el correo electrónico y Web. El libro finaliza con un análisis de algunas áreas en las cuales la seguridad afecta la privacidad, la libertad de expresión, la censura y otros aspectos sociales con los cuales choca directamente.

El capítulo 9 contiene listas de lecturas sugeridas, con comentarios, organizadas por capítulo. Su propósito es ayudar a los lectores que deseen llevar más allá el estudio sobre las redes. El capítulo también tiene una bibliografía alfabética de todas las referencias que se dan en el libro.

El sitio Web del autor puede consultarlo desde:

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

el cual contiene una página con vínculos a muchos tutoriales, FAQs, compañías, consorcios industriales, organizaciones profesionales, organizaciones de estándares, tecnologías, documentos y muchas cosas más.

1.9 RESUMEN

Las redes de computadoras se pueden utilizar para diversos servicios, tanto para compañías como para individuos. Para las compañías, las redes de computadoras personales que utilizan servidores compartidos con frecuencia dan acceso a información corporativa. Por lo general, estas redes siguen el modelo cliente-servidor, con estaciones de trabajo clientes en los escritorios de los empleados que acceden a servidores instalados en la sala de máquinas. Para los individuos, las redes ofrecen acceso a una diversidad de recursos de información y entretenimiento. Los individuos acceden a Internet mediante una llamada al ISP a través de un módem, aunque una cantidad creciente de usuarios cuenta con una conexión fija en casa. Un área con gran futuro es la de las redes inalámbricas, con nuevas aplicaciones como acceso móvil al correo electrónico y el comercio móvil.

A grandes rasgos, las redes se pueden dividir en LANs, MANs, WANs e interredes, con sus propias características, tecnologías, velocidades y nichos. Las LANs ocupan edificios y operan a altas velocidades. Las MANs abarcan toda una ciudad, por ejemplo, el sistema de televisión por cable, el cual es utilizado por mucha gente para acceder a Internet. Las WANs se extienden por un país o un continente. Las LANs y las MANs pueden ser o no conmutadas (es decir, no tienen enrutadores); las WANs son conmutadas. Las redes inalámbricas se están volviendo sumamente populares, en especial las LANs inalámbricas. Las redes se interconectan para formar interredes.

El software de red consta de protocolos, que son reglas mediante las cuales se comunican los procesos. Los protocolos son de dos tipos: orientados a la conexión y no orientados a la conexión. La mayoría de las redes soporta jerarquías de protocolos, en la cual cada capa proporciona servicios a las capas superiores a ella y las libera de los detalles de los protocolos que se utilizan en las capas inferiores. Las pilas de protocolos se basan generalmente en el modelo OSI o en el modelo TCP/IP. Ambos modelos tienen capas de red, de transporte y de aplicación, pero difieren en las demás capas. Entre los aspectos de diseño están la multiplexión, el control de flujo y el control de errores. Gran parte del libro está dedicada a los protocolos y su diseño.

Las redes ofrecen servicios a sus usuarios. Los servicios pueden ser orientados a la conexión o no orientados a ésta. En algunas redes se proporciona servicio no orientado a la conexión en una capa y servicio orientado a la conexión en la capa superior.

Las redes bien conocidas incluyen Internet, ATM, Ethernet y la LAN IEEE 802.11 inalámbrica. Internet evolucionó de ARPANET, a la cual se agregaron otras redes para conformar una interred. La Internet actual es en realidad un conjunto de miles de redes, más que de una sola red. El aspecto que la distingue es el uso generalizado de la pila de protocolos TCP/IP. ATM tiene un uso muy extendido en los sistemas telefónicos para el tráfico de datos de larga distancia. Ethernet es la LAN más popular y se utiliza en la mayoría de las compañías y universidades. Por último, las LANs inalámbricas a velocidades sorprendentemente altas (hasta 54 Mbps) comienzan a desplegarse en forma masiva.

Para que varias computadoras se comuniquen entre sí es necesaria una gran cantidad de estandarización, tanto en el software como en el hardware. Organizaciones como la ITU-T, el ISO, el IEEE y el IAB manejan diferentes partes del proceso de estandarización.

PROBLEMAS

1. Imagine que ha entrenado a su San Bernardo, Byron, para que transporte una caja con tres cintas de 8 mm en lugar del barrilito de brandy. (Cuando se llene su disco, usted tendrá una emergencia.) Cada una de estas cintas tiene capacidad de 7 gigabytes. El perro puede trasladarse adondequiera que usted vaya, a una velocidad de 18 km/hora. ¿Para cuál rango de distancias tiene Byron una tasa de datos más alta que una línea de transmisión cuya tasa de datos (sin tomar en cuenta la sobrecarga) es de 150 Mbps?
2. Una alternativa a una LAN es simplemente un enorme sistema de compartición de tiempo con terminales para todos los usuarios. Mencione dos ventajas de un sistema cliente-servidor que utilice una LAN.
3. Dos factores de red ejercen influencia en el rendimiento de un sistema cliente-servidor: el ancho de banda de la red (cuántos bits por segundo puede transportar) y la latencia (cuánto tiempo toma al primer bit llegar del cliente al servidor). Mencione un ejemplo de una red que cuente con ancho de banda y latencia altos. A continuación, mencione un ejemplo de una que cuente con ancho de banda y latencia bajos.
4. ¿Además del ancho de banda y la latencia, qué otros parámetros son necesarios para dar un buen ejemplo de la calidad de servicio ofrecida por una red destinada a tráfico de voz digitalizada?
5. Un factor en el retardo de un sistema de commutación de paquetes de almacenamiento y reenvío es el tiempo que le toma almacenar y reenviar un paquete a través de un commutador. Si el tiempo de commutación es de 10 μ seg, ¿esto podría ser un factor determinante en la respuesta de un sistema cliente-servidor en el cual el cliente se encuentre en Nueva York y el servidor en California? Suponga que la velocidad de propagación en cobre y fibra es 2/3 de la velocidad de la luz en el vacío.
6. Un sistema cliente-servidor utiliza una red satelital, con el satélite a una altura de 40,000 km. ¿Cuál es el retardo en respuesta a una solicitud, en el mejor de los casos?
7. En el futuro, cuando cada persona tenga una terminal en casa conectada a una red de computadoras, serán posibles las consultas públicas instantáneas sobre asuntos legislativos pendientes. Con el tiempo, las legislaturas existentes podrían eliminarse, para dejar que la voluntad popular se exprese directamente. Los aspectos positivos de una democracia directa como ésta son bastante obvios; analice algunos de los aspectos negativos.
8. Cinco enrutadores se van a conectar en una subred de punto a punto. Los diseñadores podrían poner una línea de alta velocidad, de mediana velocidad, de baja velocidad o ninguna línea, entre cada par de enrutadores. Si toma 100 ms de tiempo de la computadora generar e inspeccionar cada topología, ¿cuánto tiempo tomará inspeccionarlas todas?
9. Un grupo de $2^n - 1$ enrutadores están interconectados en un árbol binario centralizado, con un enrutador en cada nodo del árbol. El enrutador i se comunica con el enrutador j enviando un mensaje a la raíz del árbol. A continuación, la raíz manda el mensaje al enrutador j . Obtenga una expresión aproximada de la cantidad media de saltos por mensaje para un valor grande de n , suponiendo que todos los pares de enrutadores son igualmente probables.
10. Una desventaja de una subred de difusión es la capacidad que se desperdicia cuando múltiples *hosts* intentan acceder el canal al mismo tiempo. Suponga, por ejemplo, que el tiempo se divide en ranuras discretas, y que cada uno de los *hosts* n intenta utilizar el canal con probabilidad p durante cada parte. ¿Qué fracción de las partes se desperdicia debido a colisiones?

11. Mencione dos razones para utilizar protocolos en capas.
12. Al presidente de Specialty Paint Corp. se le ocurre la idea de trabajar con una compañía cervecera local para producir una lata de cerveza invisible (como medida para reducir los desechos). El presidente indica a su departamento legal que analice la situación, y éste a su vez pide ayuda al departamento de ingeniería. De esta forma, el ingeniero en jefe se reúne con su contraparte de la otra compañía para discutir los aspectos técnicos del proyecto. A continuación, los ingenieros informan los resultados a sus respectivos departamentos legales, los cuales a su vez se comunican vía telefónica para ponerse de acuerdo en los aspectos legales. Por último, los dos presidentes corporativos se ponen de acuerdo en la parte financiera del proyecto. ¿Éste es un ejemplo de protocolo con múltiples capas semejante al modelo OSI?
13. ¿Cuál es la diferencia principal entre comunicación orientada a la conexión y no orientada a ésta?
14. Dos redes proporcionan servicio confiable orientado a la conexión. Una de ellas ofrece un flujo confiable de bytes y la otra un flujo confiable de mensajes. ¿Son idénticas? Si es así, ¿por qué se hace la distinción? Si no son idénticas, mencione un ejemplo de algo en que difieran.
15. ¿Qué significa “negociación” en el contexto de protocolos de red? Dé un ejemplo.
16. En la figura 1-19 se muestra un servicio. ¿Hay algún otro servicio implícito en la figura? Si es así, ¿dónde? Si no lo hay, ¿por qué no?
17. En algunas redes, la capa de enlace de datos maneja los errores de transmisión solicitando que se retransmitan las tramas dañadas. Si la probabilidad de que una trama se dañe es p , ¿cuál es la cantidad media de transmisiones requeridas para enviar una trama? Suponga que las confirmaciones de recepción nunca se pierden.
18. ¿Cuál de las capas OSI maneja cada uno de los siguientes aspectos?:
 - (a) Dividir en tramas el flujo de bits transmitidos.
 - (b) Determinar la ruta que se utilizará a través de la subred.
19. Si la unidad que se transmite al nivel de enlace de datos se denomina trama y la que se transmite al nivel de red se llama paquete, ¿las tramas encapsulan paquetes o los paquetes encapsulan tramas? Explique su respuesta?
20. Un sistema tiene una jerarquía de protocolos de n capas. Las aplicaciones generan mensajes con una longitud de M bytes. En cada una de las capas se agrega un encabezado de h bytes. ¿Qué fracción del ancho de banda de la red se llena con encabezados?
21. Mencione dos similitudes entre los modelos de referencia OSI y TCP/IP. A continuación mencione dos diferencias entre ellos.
22. ¿Cuál es la principal diferencia entre TCP y UDP?
23. La subred de la figura 1-25(b) se diseñó para resistir una guerra nuclear. ¿Cuántas bombas serían necesarias para partir los nodos en dos conjuntos inconexos? Suponga que cualquier bomba destruye un nodo y todos los enlaces que se conectan a él.
24. Internet está duplicando su tamaño aproximadamente cada 18 meses. Aunque no se sabe a ciencia cierta, una estimación indica que en el 2001 había 100 millones de *hosts* en Internet. Utilice estos datos para calcular la cantidad esperada de *hosts* para el año 2010. ¿Cree que esto es real? Explique por qué.

25. Cuando un archivo se transfiere entre dos computadoras, pueden seguirse dos estrategias de confirmación de recepción. En la primera, el archivo se divide en paquetes, y el receptor confirma la recepción de cada uno de manera individual, aunque no confirma la recepción del archivo como un todo. En contraste, en la segunda estrategia la recepción de los paquetes no se confirma de manera individual, sino la del archivo completo. Comente las dos estrategias.
26. ¿Por qué ATM utiliza celdas pequeñas de longitud fija?
27. ¿Qué tan grande era un bit, en metros, en el estándar 802.3 original? Utilice una velocidad de transmisión de 10 Mbps y suponga que la velocidad de propagación en cable coaxial es $\frac{2}{3}$ la velocidad de la luz en el vacío.
28. Una imagen tiene 1024×768 píxeles con 3 bytes/píxel. Suponga que la imagen no se encuentra comprimida. ¿Cuánto tiempo tomará transmitirla sobre un canal de módem de 56 kbps? ¿Sobre un módem de cable de 1 Mbps? ¿Sobre una red Ethernet a 10 Mbps? ¿Sobre una red Ethernet a 100 Mbps?
29. Ethernet y las redes inalámbricas tienen algunas similitudes y diferencias. Una propiedad de Ethernet es que sólo se puede transmitir una trama a la vez sobre una red de este tipo. ¿El 802.11 comparte esta propiedad con Ethernet? Comente su respuesta.
30. Las redes inalámbricas son fáciles de instalar, y ello las hace muy económicas puesto que los costos de instalación eclipsan por mucho los costos del equipo. No obstante, también tienen algunas desventajas. Mencione dos de ellas.
31. Cite dos ventajas y dos desventajas de contar con estándares internacionales para los protocolos de red.
32. Cuando un sistema tiene una parte fija y una parte removible (como ocurre con una unidad de CD-ROM y el CD-ROM), es importante que exista estandarización en el sistema, con el propósito de que las diferentes compañías puedan fabricar tanto la parte removible como la fija y todo funcione en conjunto. Mencione tres ejemplos ajenos a la industria de la computación en donde existan estándares internacionales. Ahora mencione tres áreas donde no existan.
33. Haga una lista de sus actividades cotidianas en las cuales intervengan las redes de computadoras. ¿De qué manera se alteraría su vida si estas redes fueran súbitamente desconectadas?
34. Averigüe cuáles redes se utilizan en su escuela o lugar de trabajo. Describa los tipos de red, las topologías y los métodos de commutación que utilizan.
35. El programa *ping* le permite enviar un paquete de prueba a un lugar determinado y medir cuánto tarda en ir y regresar. Utilice *ping* para ver cuánto tiempo toma llegar del lugar donde se encuentra hasta diversos lugares conocidos. Con los resultados, trace el tiempo de tránsito sobre Internet como una función de la distancia. Lo más adecuado es utilizar universidades, puesto que la ubicación de sus servidores se conoce con mucha precisión. Por ejemplo, *berkeley.edu* se encuentra en Berkeley, California; *mit.edu* se localiza en Cambridge, Massachusetts; *vu.nl* está en Amsterdam, Holanda; *www.usyd.edu.au* se encuentra en Sydney, Australia, y *www.uct.ac.za* se localiza en Cape Town, Sudáfrica.
36. Vaya al sitio Web de la IETF, *www.ietf.org*, y entérese de lo que hacen ahí. Elija un proyecto y escriba un informe de media página acerca del problema y la solución que propone.
37. La estandarización es sumamente importante en el mundo de las redes. La ITU y la ISO son las principales organizaciones oficiales encargadas de la estandarización. Vaya a los sitios Web de estas organiza-

ciones, en www.itu.org y www.iso.org, respectivamente, y analice el trabajo de estandarización que realizan. Escriba un breve informe sobre las cosas que han estandarizado.

38. Internet está conformada por una gran cantidad de redes. Su disposición determina la topología de Internet. En línea se encuentra una cantidad considerable de información acerca de la topología de Internet. Utilice un motor de búsqueda para investigar más sobre la topología de Internet y escriba un breve informe sobre sus resultados.

2

LA CAPA FÍSICA

En este capítulo analizaremos la capa que está en la parte más baja de la jerarquía de la figura 1-24. Dicha capa define las interfaces mecánica, eléctrica y de temporización de la red. Comenzaremos con un análisis teórico de la transmisión de datos, el cual nos llevará a descubrir que la Madre Naturaleza establece límites en lo que se puede enviar a través de un canal.

Después trataremos tres tipos de medios de transmisión: dirigidos (cable de cobre y fibra óptica), inalámbricos (radio terrestre) y por satélite. Este material proporcionará información a fondo de las principales tecnologías de transmisión que se utilizan en las redes actuales.

El resto del capítulo se dedicará a dar tres ejemplos de sistemas de comunicación que se utilizan en la práctica en las redes de computadora de área amplia: el sistema telefónico (fijo), el sistema de telefonía móvil y el sistema de televisión por cable. Los tres utilizan una red dorsal de fibra óptica, pero están organizados de diferente manera y utilizan tecnologías distintas en la última milla (la conexión hacia el cliente).

2.1 LA BASE TEÓRICA DE LA COMUNICACIÓN DE DATOS

Mediante la variación de algunas propiedades físicas, como el voltaje o la corriente, es posible transmitir información a través de cables. Al representar el valor de este voltaje o corriente como una función simple del tiempo, $f(t)$, podemos modelar el comportamiento de la señal y analizarlo matemáticamente. Este análisis es el tema de las siguientes secciones.

2.1.1 El análisis de Fourier

A principios del siglo XIX, el matemático francés Jean-Baptiste Fourier probó que cualquier función periódica de comportamiento razonable, $g(t)$ con un periodo T , se puede construir sumando una cantidad (posiblemente infinita) de senos y cosenos:

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

donde $f = 1/T$ es la frecuencia fundamental, a_n y b_n son las amplitudes de seno y coseno de los n -ésimos (términos) **armónicos** y c es una constante. Tal descomposición se conoce como **serie de Fourier**. A partir de ella, es posible reconstruir la función, es decir, si se conoce el periodo T y se dan las amplitudes, la función original del tiempo puede encontrarse realizando las sumas que se muestran en la ecuación (2-1).

Una señal de datos que tenga una duración finita (la cual todas poseen) se puede manejar con sólo imaginar que el patrón se repite una y otra vez por siempre (es decir, el intervalo de T a $2T$ es el mismo que de 0 a T , etcétera).

Las amplitudes a_n se pueden calcular para cualquier $g(t)$ dada multiplicando ambos lados de la ecuación (2-1) por $\sin(2\pi kft)$ y después integrando de 0 a T . Puesto que

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

sólo un término de la sumatoria perdura: a_n . La sumatoria de b_n desaparece por completo. De manera similar, al multiplicar la ecuación (2-1) por $\cos(2\pi kft)$ e integrando entre 0 y T , podemos derivar b_n . Con sólo integrar ambos lados de la ecuación como está, podemos encontrar c . Los resultados de realizar estas operaciones son los siguientes:

$$a_n = \frac{2}{T} \int_0^T g(t) \sin(2\pi nft) dt \quad b_n = \frac{2}{T} \int_0^T g(t) \cos(2\pi nft) dt \quad c = \frac{2}{T} \int_0^T g(t) dt$$

2.1.2 Señales de ancho de banda limitado

Para ver cómo se relaciona todo esto con la comunicación de datos, consideremos un ejemplo específico: la transmisión del carácter “b” ASCII codificado en un byte de 8 bits. El patrón de bits que se va a transmitir es 01100010. La parte izquierda de la figura 2-1(a) muestra la salida de voltaje que produce la computadora transmisora. El análisis de Fourier de la señal produce los coeficientes:

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

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

$$c = 3/4$$

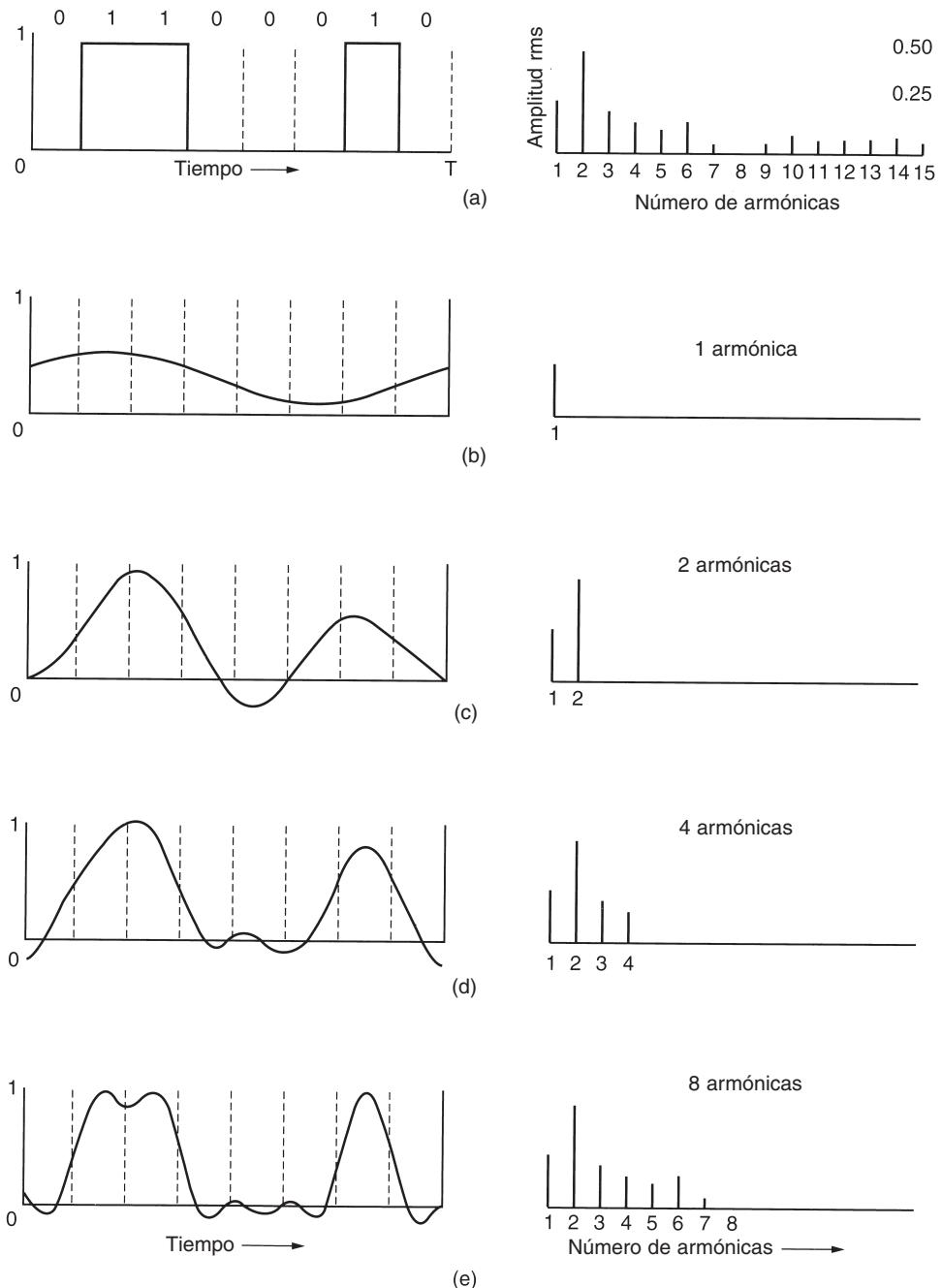


Figura 2-1. (a) Una señal binaria y sus amplitudes de raíz cuadrada media de Fourier. (b)-(e) Aproximaciones sucesivas a la señal original.

En el lado derecho de la figura 2-1(a) se muestran las amplitudes de raíz cuadrada media, $\sqrt{a_n^2 + b_n^2}$, para los primeros términos. Estos valores son importantes porque sus cuadrados son proporcionales a la energía transmitida en la frecuencia correspondiente.

Ninguna instalación transmisora puede transmitir señales sin perder cierta potencia en el proceso. Si todos los componentes de Fourier disminuyeran en la misma proporción, la señal resultante se reduciría en amplitud, pero no se distorsionaría [es decir, tendría la misma forma cuadrada que tiene en la figura 2-1(a)]. Desgraciadamente, todas las instalaciones de transmisión disminuyen los distintos componentes de Fourier en diferente grado, lo que provoca distorsión. Por lo general, las amplitudes se transmiten sin ninguna disminución desde 0 hasta cierta frecuencia f_c [medida en ciclos/seg o Hertz (Hz)], y todas las frecuencias que se encuentren por arriba de esta frecuencia de corte serán atenuadas. El rango de frecuencias que se transmiten sin atenuarse con fuerza se conoce como **ancho de banda**. En la práctica, el corte en realidad no es abrupto, por lo que con frecuencia el ancho de banda ofrecido va desde 0 hasta la frecuencia en la que el valor de la amplitud es atenuado a la mitad de su valor original.

El ancho de banda es una propiedad física del medio de transmisión y por lo general depende de la construcción, grosor y longitud de dicho medio. En algunos casos, se introduce un filtro en el circuito para limitar la cantidad de ancho de banda disponible para cada cliente. Por ejemplo, un cable de teléfono podría tener un ancho de banda de 1 MHz para distancias cortas, pero las compañías telefónicas agregan un filtro que restringe a cada cliente a aproximadamente 3100 Hz. Este ancho de banda es adecuado para el lenguaje inteligible y mejora la eficiencia del sistema al limitar a los usuarios en el uso de los recursos.

Ahora consideremos cómo luciría la señal de la figura 2-1(a) si el ancho de banda fuera tan lento que sólo las frecuencias más bajas se transmitieran [es decir, si la función fuera aproximada por los primeros términos de la ecuación 2-1(a)]. La figura 2-1(b) muestra la señal que resulta de un canal que permite que sólo pase la primera armónica (la fundamental, f'). De manera similar, la figura 2-1(c)-(e) muestra el espectro y las funciones reconstruidas de canales de ancho de banda más grande.

Dada una tasa de bits de b bits/seg, el tiempo requerido para enviar 8 bits (por ejemplo) 1 bit a la vez es $8/b$ seg, por lo que la frecuencia de la primera armónica es $b/8$ Hz. Una línea telefónica normal, llamada con frecuencia **línea con calidad de voz**, tiene una frecuencia de corte introducida de manera artificial arriba de 3000 Hz. Esta restricción significa que el número de armónicas más altas que pasan es de aproximadamente $3000/(b/8)$ o $24,000/b$ (el corte no es abrupto).

Para algunas tasas de datos, los números resultan como se muestra en la figura 2-2. A partir de estos números, queda claro que tratar de transmitir a 9600 bps por una línea telefónica transformará la figura 2-1(a) en algo similar a lo que se muestra en la figura 2-1(c), lo que dificulta la recepción precisa del flujo de bits binarios original. Debería ser obvio que a tasas de datos mucho mayores que 38.4 kbps, no hay la menor esperanza para las señales *binarias*, aun si la transmisión se encuentra completamente libre de ruidos. En otras palabras, limitar el ancho de banda limita la tasa de datos, incluso en canales perfectos. Sin embargo, existen esquemas de codificación refinados que utilizan diferentes niveles de voltaje y pueden alcanzar tasas de datos mayores. Este tema lo trataremos con mayor detalle más adelante en el capítulo.

Bps	T (mseg)	Primera armónica (Hz)	# de armónicas enviadas
300	26.67	37.5	80
600	13.33	75	40
1200	6.67	150	20
2400	3.33	300	10
4800	1.67	600	5
9600	0.83	1200	2
19200	0.42	2400	1
38400	0.21	4800	0

Figura 2-2. Relación entre tasa de datos y armónicas.

2.1.3 La tasa de datos máxima de un canal

En 1924, un ingeniero de AT&T, Henry Nyquist, se dio cuenta de que incluso un canal perfecto tiene una capacidad de transmisión finita. Derivó una ecuación que expresa la tasa de datos máxima para un canal sin ruido de ancho de banda finito. En 1948, Claude Shannon continuó el trabajo de Nyquist y lo extendió al caso de un canal sujeto a ruido aleatorio (es decir, termodinámico) (Shannon, 1948). Sólo resumiremos brevemente sus ahora clásicos resultados.

Nyquist probó que si se pasa una señal cualquiera a través de un filtro pasa-bajas de ancho de banda H , la señal filtrada se puede reconstruir por completo tomando sólo $2H$ muestras (exactas) por segundo. No tiene sentido muestrear la línea a una rapidez mayor que $2H$ veces por segundo porque los componentes de mayor frecuencia que tal muestreo puede recuperar ya se han filtrado. Si la señal consiste en V niveles discretos, el teorema de Nyquist establece:

$$\text{tasa de datos máxima} = 2H \log_2 V \text{ bits/seg}$$

Por ejemplo, un canal sin ruido de 3 kHz no puede transmitir señales binarias (es decir, de dos niveles) a una tasa mayor que 6000 bps.

Hasta aquí sólo hemos considerado canales sin ruido. Si el ruido aleatorio está presente, la situación se deteriora rápidamente. Y el ruido aleatorio (térmico) siempre está presente debido al movimiento de las moléculas del sistema. La cantidad de ruido térmico presente se mide por la relación entre la potencia de la señal y la potencia del ruido, llamada **relación señal a ruido**. Si indicamos la potencia de la señal con una S y la potencia del ruido con N , la relación señal a ruido es S/N . Por lo general, la relación misma no se expresa; en su lugar, se da la cantidad $10 \log_{10} S/N$. Estas unidades se conocen como **decibeles** (dB). Una relación S/N de 10 es 10 dB, una relación de 100 es 20 dB, una de 1000 es 30 dB, y así sucesivamente. Los fabricantes de amplificadores estereofónicos a menudo caracterizan el ancho de banda (rango de frecuencia) en el cual su producto es lineal dando la frecuencia de 3 dB en cada extremo. Éstos son los puntos a los que el factor de amplificación ha sido dividido (puesto que $\log_{10} 3 \approx 0.5$).

El resultado principal de Shannon es que la tasa de datos máxima de un canal ruidoso cuyo ancho de banda es H Hz y cuya relación señal a ruido es S/N , está dada por

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

Por ejemplo, un canal con un ancho de banda de 3000 Hz y con una relación señal a ruido térmico de 30 dB (los parámetros típicos de la parte analógica del sistema telefónico) no puede transmitir más allá de 30,000 bps, sin importar cuántos niveles de señal se utilicen, ni con qué frecuencia se tomen los muestreros. El resultado de Shannon se dedujo aplicando argumentos de la teoría de la información y es válido para cualquier canal sujeto a ruido térmico. Los ejemplos contrarios se deben clasificar en la misma categoría de las máquinas de movimiento perpetuo. Sin embargo, cabe señalar que éste solamente es un límite superior y que los sistemas reales rara vez lo alcanzan.

2.2 MEDIOS DE TRANSMISIÓN GUIADOS

El propósito de la capa física es transportar un flujo de datos puro de una máquina a otra. Es posible utilizar varios medios físicos para la transmisión real. Cada uno tiene su propio nicho en términos de ancho de banda, retardo, costo y facilidad de instalación y mantenimiento. Los medios se clasifican de manera general en medios guiados, como cable de cobre y fibra óptica, y medios no guiados, como radio y láser a través del aire. Analizaremos estos temas en las siguientes secciones.

2.2.1 Medios magnéticos

Una de las formas más comunes para transportar datos de una computadora a otra es almacenarlos en cintas magnéticas o medios extraíbles (por ejemplo, DVDs grabables), transportar físicamente la cinta o los discos a la máquina de destino y leer dichos datos ahí. Si bien este método no es tan avanzado como utilizar un satélite de comunicaciones geosíncrono, con frecuencia es más rentable, especialmente para aplicaciones en las que un ancho de banda alto o el costo por bit transportado es un factor clave.

Un cálculo simple aclarará este punto. Una cinta Ultrium estándar puede almacenar 200 gigabits. Una caja de $60 \times 60 \times 60$ cm puede contener aproximadamente 1000 de estas cintas, con una capacidad total de 200 terabytes, o 1600 terabits (1.6 petabits). Una caja de cintas puede enviarse a cualquier parte de Estados Unidos en 24 horas por Federal Express y otras compañías. El ancho de banda efectivo de esta transmisión es de 1600 terabits/86,400 seg o 19 Gbps. Si el destino está a sólo una hora por carretera, el ancho de banda se incrementa a casi 400 Gbps. Ninguna red de computadoras puede aprovechar esto.

En el caso de un banco que diariamente tiene que respaldar muchos gigabytes de datos en una segunda máquina (para poder continuar en caso de que suceda alguna inundación o un terremoto), es probable que ninguna otra tecnología de transmisión pueda siquiera acercarse en rendimiento a la cinta magnética. Es cierto que la rapidez de las redes se está incrementando, pero también las densidades de las cintas.

Si vemos ahora el costo, obtendremos un panorama similar. El costo de una cinta Ultrium es de aproximadamente \$40 cuando se compra al mayoreo. Una cinta puede reutilizarse al menos 10 veces, por lo que el costo de la cinta podría ser de \$4000 por caja, por uso. Agreguemos otros \$1000 por el envío (probablemente menos), y tenemos un costo de más o menos \$5000 por almacenar 200 TB. Esto equivale a 3 centavos por cada gigabyte. Ninguna red puede superar esto. La moraleja es:

Nunca subestime el ancho de banda de una camioneta repleta de cintas que va a toda velocidad por la carretera

2.2.2 Par trenzado

Aunque las características del ancho de banda de una cinta magnética son excelentes, las de retardo son pobres. El tiempo de transmisión se mide en minutos u horas, no en milisegundos. Para muchas aplicaciones se necesita una conexión en línea. Uno de los medios de transmisión más viejos, y todavía el más común, es el **cable de par trenzado**. Éste consiste en dos alambres de cobre aislados, por lo regular de 1 mm de grueso. Los alambres se trenzan en forma helicoidal, igual que una molécula de DNA. Esto se hace porque dos alambres paralelos constituyen una antena simple. Cuando se trenzan los alambres, las ondas de diferentes vueltas se cancelan, por lo que la radiación del cable es menos efectiva.

La aplicación más común del cable de par trenzado es en el sistema telefónico. Casi todos los teléfonos están conectados a la compañía telefónica mediante un cable de par trenzado. La distancia que se puede recorrer con estos cables es de varios kilómetros sin necesidad de amplificar las señales, pero para distancias mayores se requieren repetidores. Cuando muchos cables de par trenzado recorren de manera paralela distancias considerables, como podría ser el caso de los cables de un edificio de departamentos que van hacia la compañía telefónica, se suelen atar en haces y se cubren con una envoltura protectora. Los cables dentro de estos haces podrían sufrir interferencias si no estuvieran trenzados. En algunos lugares del mundo en donde las líneas telefónicas se instalan en la parte alta de los postes, se observan frecuentemente dichos haces, de varios centímetros de diámetro.

Los cables de par trenzado se pueden utilizar para transmisión tanto analógica como digital. El ancho de banda depende del grosor del cable y de la distancia que recorre; en muchos casos pueden obtenerse transmisiones de varios megabits/seg, en distancias de pocos kilómetros. Debido a su comportamiento adecuado y bajo costo, los cables de par trenzado se utilizan ampliamente y es probable que permanezcan por muchos años.

Hay varios tipos de cableado de par trenzado, dos de los cuales son importantes para las redes de computadoras. Los cables de par trenzado **categoría 3** consisten en 2 alambres aislados que se trenzan de manera delicada. Cuatro de estos pares se agrupan por lo regular en una envoltura de plástico para su protección. Antes de 1988, la mayoría de los edificios de oficinas tenía un cable de categoría 3 que iba desde un **gabinete de cableado** central en cada piso hasta cada oficina. Este esquema permitió que hasta cuatro teléfonos comunes o dos teléfonos de múltiples líneas en cada oficina se conectaran con el equipo de la compañía telefónica en el gabinete de cableado.

A comienzos de 1988 se introdujeron los cables de par trenzado **categoría 5** más avanzados. Son similares a los de la categoría 3, pero con más vueltas por centímetro, lo que produce una menor diafonía y una señal de mejor calidad a distancias más largas. Esto los hace más adecuados para una comunicación más rápida entre computadoras. Las siguientes son las categorías 6 y 7, que tienen capacidad para manejar señales con anchos de banda de 250 y 600 MHz, respectivamente (en comparación con los 16 y 100 MHz de las categorías 3 y 5, respectivamente).

Todos estos tipos de cableado comúnmente se conocen como **UTP (Par Trenzado sin Blindaje)**, en comparación con los cables de par trenzado costosos, blindados y voluminosos que IBM introdujo a principios de la década de 1980, los cuales no ganaron popularidad fuera de las instalaciones de IBM. En la figura 2-3 se muestra un cableado de par trenzado.



Figura 2-3. (a) UTP categoría 3. (b) UTP categoría 5.

2.2.3 Cable coaxial

Otro medio común de transmisión es el **cable coaxial** (conocido frecuentemente tan sólo como “coax”). Este cable tiene mejor blindaje que el de par trenzado, así que puede abarcar tramos más largos a velocidades mayores. Hay dos clases de cable coaxial que son las más utilizadas. Una clase: el cable de 50 ohms, se usa por lo general para transmisión digital. La otra clase, el cable de 75 ohms, se utiliza comúnmente para la transmisión analógica y la televisión por cable, pero se está haciendo cada vez más importante con el advenimiento de Internet a través de cable. Esta distinción se basa en hechos históricos, más que en técnicos (por ejemplo, las antenas antiguas de dipolos tenían una impedancia de 300 ohms y era fácil utilizar los transformadores adaptadores de impedancia 4:1).

Un cable coaxial consiste en un alambre de cobre rígido como núcleo, rodeado por un material aislante. El aislante está forrado con un conductor cilíndrico, que con frecuencia es una malla de tejido fuertemente trenzado. El conductor externo se cubre con una envoltura protectora de plástico. En la figura 2-4 se muestra una vista en corte por capas de un cable coaxial.

La construcción y el blindaje del cable coaxial le confieren una buena combinación de ancho de banda alto y excelente inmunidad al ruido. El ancho de banda posible depende de la calidad y longitud del cable, y de la relación señal a ruido de la señal de datos. Los cables modernos tienen un ancho de banda de cerca de 1 GHz. Los cables coaxiales solían ser ampliamente usados en el sistema telefónico para las líneas de larga distancia, pero en la actualidad han sido reemplazados por la fibra óptica en rutas de distancias considerables. Sin embargo, el cable coaxial aún se utiliza ampliamente en la televisión por cable y en las redes de área metropolitana.

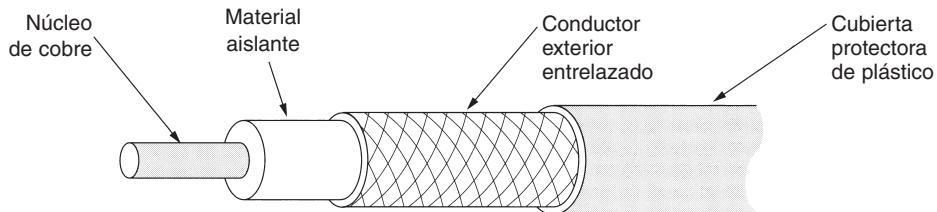


Figura 2-4. Un cable coaxial.

2.2.4 Fibra óptica

Muchas personas de la industria de la computación se enorgullecen de lo rápido que está mejorando la tecnología en esta área. La PC original de IBM (1981) se ejecutaba a una velocidad de reloj de 4.77 MHz. Veinte años más tarde, las PCs pueden correr a 2 GHz, con un factor de ganancia de 20 por década. No está nada mal.

En el mismo periodo, la comunicación de datos de área amplia pasó de 56 kbps (ARPANET) a 1 Gbps (comunicación óptica moderna), con un factor de ganancia de más de 125 por década, y al mismo tiempo la tasa de error pasó de 10^{-5} por bit hasta casi cero.

Además, las CPUs individuales están empezando a aproximarse a límites físicos, como la velocidad de la luz y los problemas de la disipación de calor. En contraste, con la tecnología *actual* de fibras, el ancho de banda alcanzable ciertamente está por encima de los 50,000 Gbps (50 Tbps) y muchas personas se están esforzando arduamente para encontrar mejores tecnologías y materiales. El límite práctico de señalización actual de aproximadamente 10 Gbps se debe a nuestra incapacidad para convertir con mayor rapidez las señales eléctricas a ópticas, aunque en el laboratorio se han alcanzado hasta 100 Gbps en una sola fibra.

En la competencia entre la computación y la comunicación, esta última ganó. La generación de científicos e ingenieros de computación acostumbrados a pensar en términos de los bajos límites de Nyquist y Shannon impuestos por el alambre de cobre aún no ha comprendido todas las implicaciones del ancho de banda prácticamente infinito (aunque no sin un costo). El nuevo sentido común debería ser que todas las computadoras son desesperadamente lentas y que las redes deberían tratar de evitar las tareas de cómputo a cualquier precio, sin importar cuánto ancho de banda se desperdicie. En esta sección analizaremos la fibra óptica para ver cómo funciona esa tecnología de transmisión.

Un sistema de transmisión óptico tiene tres componentes: la fuente de luz, el medio de transmisión y el detector. Convencionalmente, un pulso de luz indica un bit 1 y la ausencia de luz indica un bit 0. El medio de transmisión es una fibra de vidrio ultradelgada. El detector genera un pulso eléctrico cuando la luz incide en él. Al agregar una fuente de luz en un extremo de una fibra óptica y un detector en el otro, se tiene un sistema de transmisión de datos unidireccional que acepta una señal eléctrica, la convierte y transmite mediante pulsos de luz y, luego, reconvierte la salida a una señal eléctrica en el extremo receptor.

Este sistema de transmisión tendría fugas de luz y sería inútil en la práctica excepto por un principio interesante de la física. Cuando un rayo de luz pasa por un medio a otro —por ejemplo, de sílice fundida al aire—, el rayo se refracta (se dobla) en la frontera de la sílice y el aire, como se muestra en la figura 2-5(a). En ella vemos un rayo de luz que incide en la frontera con un ángulo α_1 y que emerge con un ángulo β_1 . El grado de refracción depende de las propiedades de los dos medios (en particular sus índices de refracción). Para ángulos con incidencias mayores de ciertos valores críticos, la luz se refracta nuevamente a la sílice; ninguna parte de él escapa al aire. Por lo tanto, un rayo de luz que incide en un ángulo mayor o igual que el crítico queda atrapado dentro de la fibra, como se muestra en la figura 2-5(b), y se puede propagar por varios kilómetros prácticamente sin pérdida.

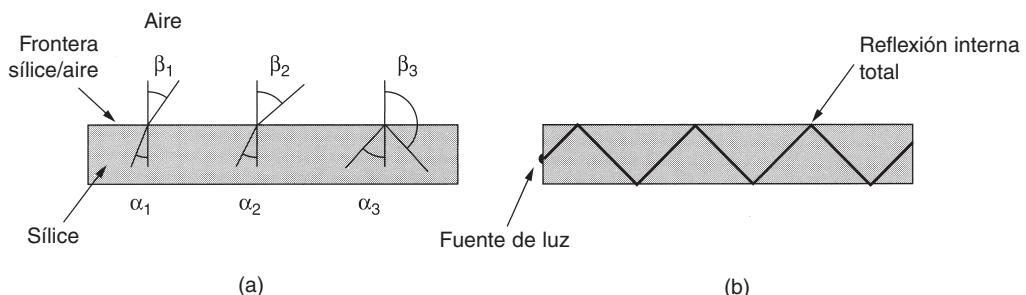


Figura 2-5. (a) Tres ejemplos de un rayo de luz procedente del interior de una fibra de sílice que incide sobre la frontera de la sílice y el aire con diferentes ángulos. (b) Luz atrapada por reflexión interna total.

El diagrama de la segunda figura únicamente muestra un rayo atrapado, pero puesto que cualquier rayo de luz que incida en la frontera con un ángulo mayor que el crítico se reflejará internamente, muchos rayos estarán rebotando con ángulos diferentes. Se dice que cada rayo tiene un **modo** diferente, por lo que una fibra que tiene esta propiedad se denomina **fibra multimodo**.

Por otro lado, si el diámetro de la fibra se reduce a unas cuantas longitudes de onda de luz, la fibra actúa como una guía de ondas y la luz se puede propagar sólo en línea recta, sin rebotar, lo cual da como resultado una **fibra monomodo**. Las fibras monomodo son más caras, pero se pueden utilizar en distancias más grandes. Las fibras monomodo disponibles en la actualidad pueden transmitir datos a 50 Gbps a una distancia de 100 km sin amplificación. En el laboratorio se han logrado tasas de datos todavía mayores a distancias más cortas.

Transmisión de la luz a través de fibra óptica

Las fibras ópticas se hacen de vidrio, que a su vez se fabrica con arena, una materia debajo costo disponible en cantidades ilimitadas. La fabricación de vidrio era conocida por los antiguos egipcios, pero su vidrio no tenía más de 1 mm de espesor, porque de lo contrario la luz no podía atravesarlo. Durante el Renacimiento se forjó un vidrio suficientemente transparente para utilizarlo en ventanas. El vidrio utilizado para fabricar fibras ópticas modernas es tan transparente que si

el océano estuviera lleno de éste en lugar de agua, el fondo del mar sería tan visible desde la superficie como lo es el suelo desde un avión en un día claro.

La atenuación de la luz dentro del vidrio depende de la longitud de onda de la luz (así como de algunas propiedades físicas del vidrio). En la figura 2-6 se muestra la atenuación para la clase de vidrio que se usa en las fibras, en decibeles por kilómetro lineal de fibra. La atenuación en decibeles está dada por la fórmula:

$$\text{Atenuación en decibeles} = 10 \log_{10} \frac{\text{potencia transmitida}}{\text{potencia recibida}}$$

Por ejemplo, un factor de pérdida de dos da como resultado una atenuación de $10 \log_{10} 2 = 3$ dB. La figura muestra la parte cercana al infrarrojo del espectro, que es la que se utiliza en la práctica. La luz visible tiene longitudes de onda ligeramente más cortas, de 0.4 a 0.7 micras (1 micra es 10^{-6} metros). Los puristas de la métrica se referirían a estas longitudes de onda como 400 nm a 700 nm, pero nosotros nos apegaremos al uso tradicional.

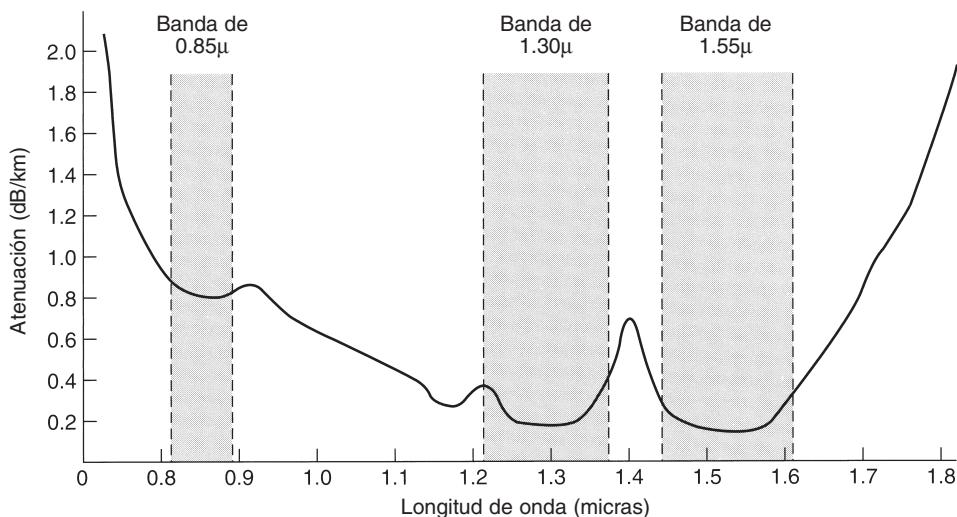


Figura 2-6. Atenuación de la luz dentro de una fibra en la región de infrarrojo.

Para las comunicaciones se utilizan tres bandas de longitud de onda, las cuales se centran en 0.85, 1.30 y 1.55 micras, respectivamente. Las últimas dos tienen buenas propiedades de atenuación (una pérdida de menos de 5% por kilómetro). La banda de 0.85 micras tiene una atenuación más alta, pero a esa longitud de onda, los láseres y los componentes electrónicos se pueden fabricar con el mismo material (arseniuro de galio). Las tres bandas tienen una anchura de entre 25,000 y 30,000 GHz.

La longitud de los pulsos de luz transmitidos por una fibra aumenta conforme se propagan. Este fenómeno se llama **dispersión cromática**. La magnitud de ésta depende de la longitud de

onda. Una forma de evitar que se encimen estos pulsos dispersos es incrementar la distancia entre ellos, pero esto solamente se puede hacer reduciendo la tasa de transmisión. Por fortuna, se ha descubierto que al dar a los pulsos cierta forma especial relacionada con el recíproco del coseno hiperbólico, casi todos los efectos de la dispersión se disipan y puede ser posible enviar pulsos a miles de kilómetros sin una distorsión apreciable de la forma. Estos pulsos se llaman **solitones**. Se está realizando un enorme esfuerzo de investigación para llevar a la práctica el uso de los solitones.

Cables de fibra

Los cables de fibra óptica son similares a los coaxiales, excepto por el trenzado. La figura 2-7(a) muestra una fibra individual vista de lado. Al centro se encuentra el núcleo de vidrio, a través del cual se propaga la luz. En las fibras multimodo el diámetro es de 50 micras, aproximadamente el grosor de un cabello humano. En las fibras monomodo el núcleo es de 8 a 10 micras.

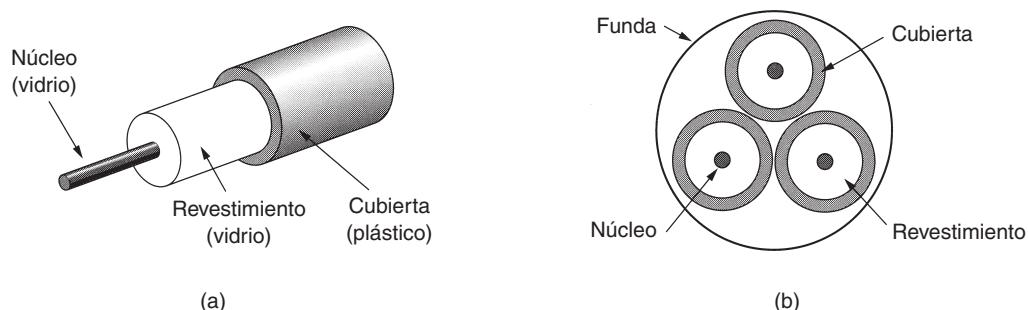


Figura 2-7. (a) Vista de lado de una fibra individual. (b) Vista de extremo de una funda con tres fibras.

El núcleo está rodeado por un revestimiento de vidrio con un índice de refracción menor que el del núcleo, con el fin de mantener toda la luz en este último. A continuación está una cubierta plástica delgada para proteger al revestimiento. Las fibras por lo general se agrupan en haces, protegidas por una funda exterior. La figura 2-7(b) muestra una funda con tres fibras.

Las cubiertas de fibras terrestres por lo general se colocan en el suelo a un metro de la superficie, donde a veces pueden sufrir daños ocasionados por retroexcavadoras o tuzas. Cerca de la costa, las cubiertas de fibras transoceánicas se entierran en zanjas mediante una especie de arado marino. En las aguas profundas, simplemente se colocan al fondo, donde los barcos de arrastre pueden tropezar con ellas o los calamares gigantes pueden atacarlas.

Las fibras se pueden conectar de tres formas diferentes. Primera, pueden terminar en conectores e insertarse en enchufes de fibra. Los conectores pierden entre 10 y 20% de la luz, pero facilitan la reconfiguración de los sistemas.

Segunda, se pueden empalmar de manera mecánica. Los empalmes mecánicos acomodan dos extremos cortados con cuidado, uno junto a otro, en una manga especial y los sujetan en su lugar. La alineación se puede mejorar pasando luz a través de la unión y haciendo pequeños ajustes para maximizar la señal. Personal especializado realiza los empalmes mecánicos en alrededor de cinco minutos, y la pérdida de luz de estos empalmes es de 10%.

Tercera, se pueden fusionar (fundir) dos tramos de fibra para formar una conexión sólida. Un empalme por fusión es casi tan bueno como una sola fibra, pero aun aquí hay un poco de atenuación.

Con los tres tipos de empalme pueden ocurrir reflejos en el punto del empalme, y la energía reflejada puede interferir la señal.

Por lo general se utilizan dos clases de fuente de luz para producir las señales: LEDs (diodos emisores de luz) y láseres semiconductores. Estas fuentes tienen propiedades diferentes, como se muestra en la figura 2-8, y su longitud de onda se puede ajustar mediante la inserción de interferómetros Fabry-Perot o Mach-Zehnder entre la fuente y la fibra. Los interferómetros Fabry-Perot son cavidades simples de resonancia que consisten en dos espejos paralelos. La luz incide de manera perpendicular en los espejos. La longitud de la cavidad separa las longitudes de onda que caben en ella un número entero de veces. Los interferómetros de Mach-Zehnder separan la luz en dos haces. Éstos viajan distancias ligeramente diferentes. Se vuelven a combinar en el extremo y quedan en fase sólo para ciertas longitudes de onda.

Elemento	LED	Láser semiconductor
Tasa de datos	Baja	Alta
Tipo de fibra	Multimodo	Multimodo o monomodo
Distancia	Corta	Larga
Tiempo de vida	Largo	Corto
Sensibilidad a la temperatura	Menor	Considerable
Costo	Bajo	Elevado

Figura 2-8. Comparación de diodos semiconductores y LEDs como fuentes de luz.

El extremo receptor de una fibra óptica consiste en un fotodiodo, el cual emite un pulso eléctrico cuando lo golpea la luz. El tiempo de respuesta típico de un fotodiodo es 1 nseg, lo que limita las tasas de datos a aproximadamente 1 Gbps. El ruido térmico también es un problema, por lo que un pulso de luz debe llevar suficiente potencia para que se pueda detectar. Al hacer que los pulsos tengan suficiente potencia, la tasa de errores puede disminuirse de manera considerable.

Redes de fibra óptica

La fibra óptica se puede utilizar en LANs, así como en transmisiones de largo alcance, aunque conectarse a ellas es más complicado que a una Ethernet. Una forma de superar el problema es reconocer que una red de anillo es en realidad una colección de enlaces punto a punto, como se muestra en la figura 2-9. La interfaz en cada computadora pasa el flujo de pulsos de luz hacia el siguiente enlace y también sirve como unión T para que la computadora pueda enviar y aceptar mensajes.

Se usan dos tipos de interfaz. Una interfaz pasiva consiste en dos derivaciones fusionadas a la fibra principal. Una derivación tiene un LED o un diodo láser en su extremo (para transmitir) y la otra tiene un fotodiodo (para recibir). La derivación misma es pasiva por completo y, por lo

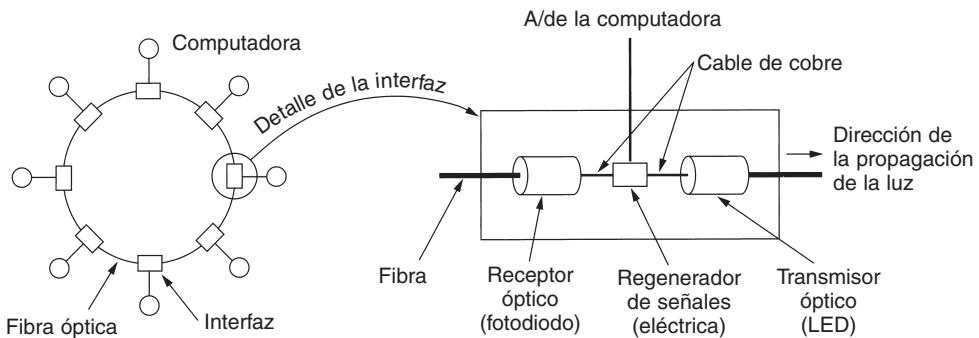


Figura 2-9. Anillo de fibra óptica con repetidores activos.

mismo, es extremadamente confiable pues un LED o un fotodiode descompuesto no romperá el anillo, sólo dejará fuera de línea a una computadora.

El otro tipo de interfaz, mostrado en la figura 2-9, es el **repetidor activo**. La luz entrante se convierte en una señal eléctrica que se regenera a toda su intensidad si se debilitó y se retransmite como luz. La interfaz con la computadora es un alambre ordinario de cobre que entra en el regenerador de señales. En la actualidad también se usan los repetidores puramente ópticos. Estos dispositivos no requieren las conversiones óptica a eléctrica a óptica, lo que significa que pueden operar con anchos de banda muy altos.

Si falla un repetidor activo, el anillo se rompe y la red se cae. Por otro lado, puesto que la señal se regenera en cada interfaz, los enlaces individuales de computadora a computadora pueden tener una longitud de kilómetros, virtualmente sin un límite para el tamaño total del anillo. Las interfaces pasivas pierden luz en cada unión, de modo que la cantidad de computadoras y la longitud total del anillo se restringen en forma considerable.

La topología de anillo no es la única manera de construir una LAN con fibra óptica. También es posible tener difusión por hardware utilizando la construcción de **estrella pasiva** de la figura 2-10. En este diseño, cada interfaz tiene una fibra que corre desde su transmisor hasta un cilindro de sílice, con las fibras entrantes fusionadas a un extremo del cilindro. En forma similar, las fibras fusionadas al otro extremo del cilindro corren hacia cada uno de los receptores. Siempre que una interfaz emite un pulso de luz, se difunde dentro de la estrella pasiva para iluminar a todos los receptores, con lo que se alcanza la difusión. En efecto, la estrella pasiva combina todas las señales entrantes y transmite el resultado combinado por todas las líneas. Puesto que la energía entrante se divide entre todas las líneas que salen, la cantidad de nodos en la red está limitada por la sensibilidad de los fotodioides.

Comparación de la fibra óptica y el alambre de cobre

Es instructivo comparar la fibra con el cobre. La fibra tiene muchas ventajas. Para empezar, puede manejar anchos de banda mucho mayores que el cobre. Tan sólo por esto, su uso sería indispensable en redes de alto rendimiento. Debido a la baja atenuación, sólo se necesitan repetidores cada 50 km aproximadamente en líneas largas, contra casi cada 5 km cuando se usa cobre, lo que

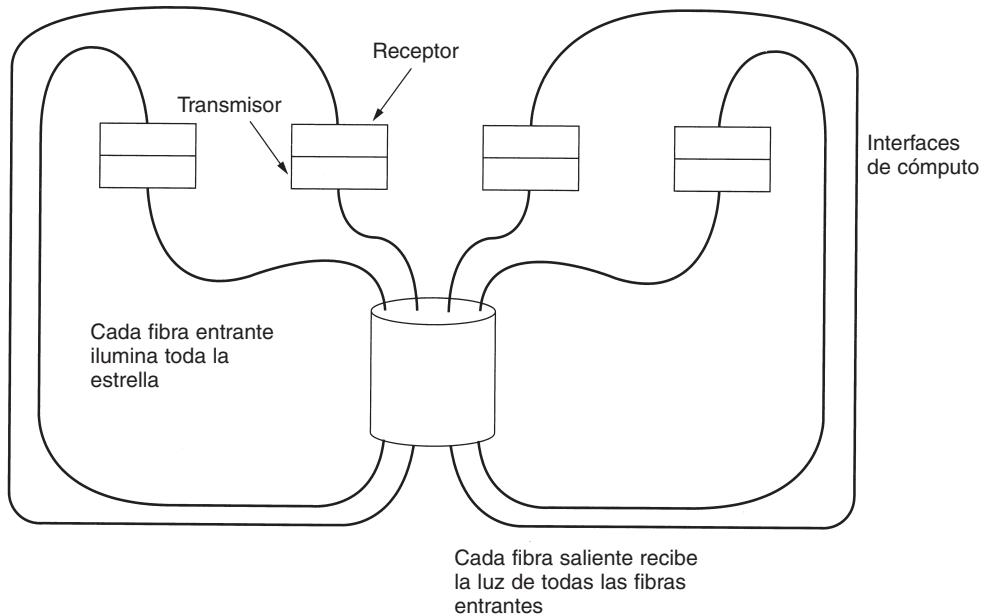


Figura 2-10. Conexión de estrella pasiva en una red de fibra óptica.

implica un ahorro considerable. La fibra también tiene la ventaja de que las sobrecargas de energía, la interferencia electromagnética o los cortes en el suministro de energía no la afectan. Las sustancias corrosivas del ambiente tampoco la afectan, lo que la hace ideal para ambientes fabriles pesados.

A las compañías telefónicas les gusta la fibra por una razón diferente: es delgada y ligera. Muchos conductos de cable existentes están completamente llenos, por lo que no hay espacio para agregar más capacidad. Al eliminar todo el cobre y reemplazarlo por fibra, se vacían los conductos y el cobre tiene un valor de reventa excelente para los refinadores de cobre quienes lo aprecian como materia prima de alta calidad. Además, las fibras son más ligeras que el cobre. Mil cables de par trenzado de 1 km pesan 8000 kg. Dos fibras tienen más capacidad y pesan sólo 100 kg, lo cual reduce de manera significativa la necesidad de sistemas mecánicos de apoyo que tienen que mantenerse. Para las nuevas rutas, la fibra se impone debido a su bajo costo de instalación.

Por último, las fibras no tienen fugas de luz y es difícil intervenirlas y conectarse a ellas. Estas propiedades dan a las fibras una seguridad excelente contra posibles espías.

Su parte negativa consiste en que es una tecnología poco familiar que requiere habilidades de las cuales carece la mayoría de los ingenieros, y en que las fibras pueden dañarse con facilidad si se doblan demasiado. Debido a que la transmisión óptica es unidireccional, la comunicación en ambos sentidos requiere ya sea dos fibras o dos bandas de frecuencia en una fibra. Por último, las interfaces de fibra cuestan más que las eléctricas. No obstante, el futuro de todas las comunicaciones fijas de datos para distancias de más de unos cuantos metros claramente es la fibra. Para un análisis de todos los aspectos de la fibra óptica y sus redes, vea (Hecht, 2001).

2.3 TRANSMISIÓN INALÁMBRICA

En nuestra era han surgido los adictos a la información: gente que necesita estar todo el tiempo en línea. Para estos usuarios móviles, el cable de par trenzado, el cable coaxial y la fibra óptica no son útiles. Ellos necesitan obtener datos para sus computadoras *laptop*, *notebook*, de bolsillo, de mano o de reloj pulsera sin estar limitados a la infraestructura de comunicaciones terrestre. Para estos usuarios, la comunicación inalámbrica es la respuesta. En las siguientes secciones veremos la comunicación inalámbrica en general, y veremos que tiene otras aplicaciones importantes además de proporcionar conectividad a los usuarios que desean navegar por Web desde la playa.

Algunas personas creen que en el futuro sólo habrá dos clases de comunicación: de fibra óptica e inalámbrica. Todos los aparatos fijos (es decir, no móviles): computadoras, teléfonos, faxes, etcétera, se conectarán con fibra óptica; todos los aparatos móviles usarán comunicación inalámbrica.

Sin embargo, la comunicación inalámbrica también tiene ventajas para los dispositivos fijos en ciertas circunstancias. Por ejemplo, si es difícil tender fibras hasta un edificio debido al terreno (montañas, selvas, pantanos, etcétera), podría ser preferible un sistema inalámbrico. Vale la pena mencionar que la comunicación digital inalámbrica moderna comenzó en las islas de Hawái, en donde partes considerablemente grandes del océano Pacífico separaban a los usuarios, y el sistema telefónico era inadecuado.

2.3.1 El espectro electromagnético

Cuando los electrones se mueven crean ondas electromagnéticas que se pueden propagar por el espacio libre (aun en el vacío). El físico británico James Clerk Maxwell predijo estas ondas en 1865 y el físico alemán Heinrich Hertz las observó en 1887. La cantidad de oscilaciones por segundo de una onda electromagnética es su **frecuencia**, f , y se mide en **Hz** (en honor a Heinrich Hertz). La distancia entre dos puntos máximos (o mínimos) consecutivos se llama **longitud de onda** y se designa de forma universal con la letra griega λ (lambda).

Al conectarse una antena del tamaño apropiado a un circuito eléctrico, las ondas electromagnéticas pueden ser difundidas de manera eficiente y ser captadas por un receptor a cierta distancia. Toda la comunicación inalámbrica se basa en este principio.

En el vacío, todas las ondas electromagnéticas viajan a la misma velocidad, no importa cuál sea su frecuencia. Esta velocidad, por lo general llamada **velocidad de la luz**, c , es de aproximadamente 3×10^8 m/seg o de un pie (30 cm) por nanosegundo. En el cobre o en la fibra óptica, la velocidad baja a casi $2/3$ de este valor y se vuelve ligeramente dependiente de la frecuencia. La velocidad de la luz es el límite máximo de velocidad. Ningún objeto o señal puede moverse más rápido que la luz.

La relación fundamental entre f , λ y c (en el vacío) es:

$$\lambda f = c \quad (2-2)$$

Puesto que c es una constante, si conocemos el valor de f , podremos encontrar el de λ , y viceversa. Como regla general, cuando λ se expresa en metros y f en MHz, $\lambda f \approx 300$. Por ejemplo, las

ondas de 100 MHz son de aproximadamente 3 metros de longitud, las de 1000 MHz son de 0.3 metros y las ondas de 0.1 metros de longitud tienen una frecuencia de 3000 MHz.

En la figura 2-11 se muestra el espectro electromagnético. Las porciones de radio, microondas, infrarrojo y luz visible del espectro pueden servir para transmitir información modulando la amplitud, frecuencia o fase de las ondas. La luz ultravioleta, los rayos X y los rayos gamma serían todavía mejores, debido a sus frecuencias más altas, pero son difíciles de producir y modular, no se propagan bien entre edificios y son peligrosos para los seres vivos. Las bandas que se listan en la parte inferior de la figura 2-11 son los nombres oficiales de la ITU y se basan en las longitudes de onda, de modo que la banda LF va de 1 a 10 km (aproximadamente 30 a 300 kHz). Los términos LF, MF y HF se refieren a las frecuencias baja, media y alta, respectivamente. Como podrá observar, cuando se asignaron los nombres, nadie esperaba que se sobrepasarían los 10 MHz, por lo que posteriormente a las bandas más altas se les nombró como bandas VHF (frecuencia muy alta), UHF (frecuencia ultraalta), EHF (frecuencia extremadamente alta) y THF (frecuencia tremadamente alta). No hay más nombres aparte de éstos, pero IHF, AHF y PHF (increíblemente alta frecuencia, asombrosamente alta frecuencia y prodigiosamente alta frecuencia) sonarían bien.

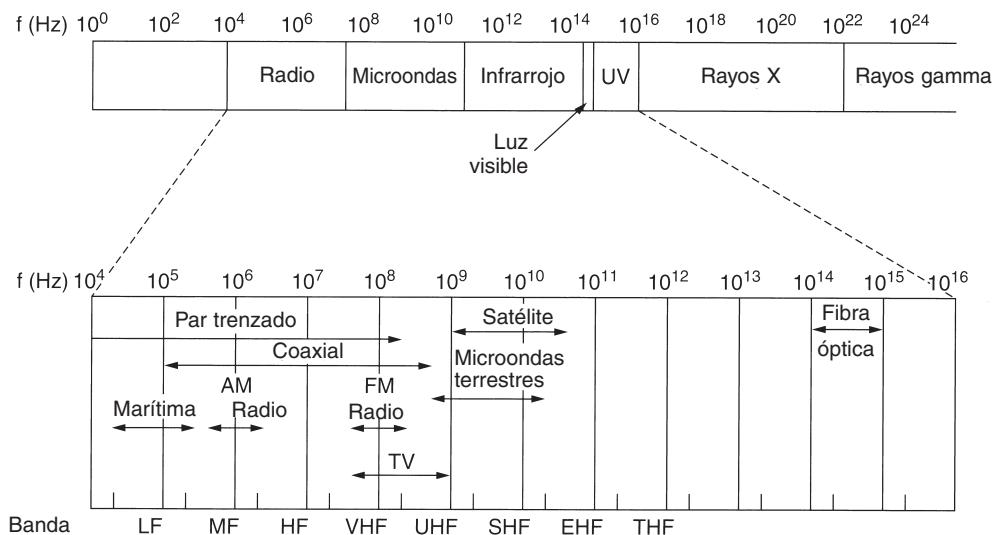


Figura 2-11. El espectro electromagnético y sus usos para comunicaciones.

La cantidad de información que puede transportar una onda electromagnética se relaciona con su ancho de banda. Con la tecnología actual, es posible codificar unos cuantos bits por hertz a frecuencias bajas, pero a frecuencias altas el número puede llegar hasta 8, de modo que un cable coaxial con un ancho de banda de 750 MHz puede transportar varios gigabits/seg. La figura 2-11 debe dejar en claro ahora por qué a la gente de redes le gusta tanto la fibra óptica.

Si resolvemos la ecuación (2-2) para f y la diferenciamos con respecto a λ , obtenemos

$$\frac{df}{d\lambda} = -\frac{c}{\lambda^2}$$

Si ahora usamos diferencias finitas en lugar de diferenciales y sólo consideramos los valores absolutos, obtenemos

$$\Delta f = \frac{c\Delta\lambda}{\lambda^2} \quad (2-3)$$

Por lo tanto, dado el ancho de una banda de longitud de onda, $\Delta\lambda$, podemos calcular la banda de frecuencia correspondiente, Δf , y a partir de ella, la tasa de datos que puede producir la banda. Cuanto más ancha sea ésta, mayor será la tasa de datos. Por ejemplo, considere la banda de 1.30 micras de la figura 2-6. Aquí tenemos $\lambda = 1.3 \times 10^{-6}$ y $\Delta\lambda = 0.17 \times 10^{-6}$, de manera que Δf es de aproximadamente 30 THz. A 8 bits/Hz, obtenemos 240 Tbps.

La mayoría de las transmisiones ocupa una banda de frecuencias estrecha (es decir, $\Delta f/f \ll 1$) a fin de obtener la mejor recepción (muchos watts/Hz). Sin embargo, en algunos casos se utiliza una banda ancha, con dos variaciones. En el **espectro disperso con salto de frecuencia**, el transmisor salta de frecuencia en frecuencia cientos de veces por segundo. Es popular en la comunicación militar debido a que de esta manera es difícil detectar las transmisiones y casi imposible intervenirlas. Ofrece buena resistencia al desvanecimiento por múltiples trayectorias debido a que la señal directa siempre llega primero al receptor. Las señales reflejadas siguen una trayectoria más larga y llegan más tarde. Para ese entonces, tal vez el receptor ya haya cambiado de frecuencia y no acepte señales de la frecuencia anterior, con lo que se elimina la interferencia entre las señales directas y reflejadas. En años recientes, esta técnica también se ha aplicado comercialmente —por ejemplo, tanto 802.11 como Bluetooth la utilizan.

Como nota curiosa, la atractiva austriaca Hedy Lamarr, la primera mujer que apareció desnuda en una película cinematográfica (el filme checoslovaco *Extase* de 1933), colaboró en la invención de esta técnica. Su primer esposo, un fabricante de armamento, le comentó lo fácil que era bloquear las señales de radio, las cuales en ese entonces se utilizaban para controlar los torpedos. Cuando descubrió que su esposo estaba vendiendo armas a Hitler, se horrorizó y se disfrazó de criada para escapar de él rumbo a Hollywood para continuar su carrera como actriz de cine. En su tiempo libre, inventó el salto de frecuencia para ayudar a los aliados en la guerra. Su diseño utilizaba 88 frecuencias, el número de teclas (y frecuencias) de un piano. Por su invento, ella y el compositor de música George Antheil, su amigo, recibieron la patente 2,292,387 de Estados Unidos. Sin embargo, no pudieron convencer a la Marina de Estados Unidos de que su invento era útil y, por lo tanto, nunca recibieron regalías. Años después de que la patente expirara, su invento cobró popularidad.

El otro tipo de espectro disperso, el **espectro disperso de secuencia directa** —el cual dispersa la señal a través una banda de frecuencia ancha—, está ganando popularidad en el mundo comercial. En particular, algunos teléfonos móviles de segunda generación lo utilizan, y dominará en los de tercera generación, gracias a su buena eficiencia espectral, inmunidad al ruido y otras propiedades. Algunas LANs inalámbricas también lo utilizan. Posteriormente volveremos al tema del espectro disperso. Si desea ver una historia fascinante y detallada de las comunicaciones por espectro disperso, vea (Scholtz, 1982).

Por el momento, supondremos que todas las transmisiones utilizan una banda de frecuencia estrecha. Ahora veremos cómo se emplean las distintas partes del espectro electromagnético de la figura 2-11, comenzando por la radio.

2.3.2 Radiotransmisión

Las ondas de radio son fáciles de generar, pueden viajar distancias largas y penetrar edificios sin problemas, y por ello su uso está muy generalizado en la comunicación, tanto en interiores como en exteriores. Las ondas de radio también son omnidireccionales, lo que significa que viajan en todas direcciones a partir de la fuente, por lo que no es necesario que el transmisor y el receptor se encuentren alineados físicamente.

En ocasiones la radio omnidireccional es buena, y en otras no lo es tanto. En la década de 1970, General Motors decidió equipar sus Cadillacs nuevos con frenos antibloqueo controlados por computadora. Cuando el conductor pisaba el pedal del freno, la computadora accionaba los frenos de manera intermitente en lugar de bloquearlos firmemente. Un buen día, un oficial que patrullaba las carreteras de Ohio encendió su nuevo radio móvil para llamar a su cuartel general y, de repente, el Cadillac que iba junto a él empezó a comportarse de manera muy extraña. El oficial le indicó al conductor que se detuviera a un lado del camino y, cuando lo hizo, el conductor alegó que él nada había hecho y que el carro se había vuelto loco.

Con el tiempo empezó a surgir un patrón: los Cadillacs ocasionalmente se comportaban de manera muy extraña, pero sólo en las principales carreteras de Ohio y sólo cuando alguna patrulla de caminos estaba cerca. Durante mucho tiempo General Motors no pudo comprender por qué los Cadillacs funcionaban bien en todos los demás estados e incluso en los caminos secundarios de Ohio. Después de una búsqueda intensa descubrieron que el cableado de los Cadillacs constituía una excelente antena para la frecuencia que usaba el nuevo sistema de radio de las patrullas de caminos de Ohio.

Las propiedades de las ondas de radio dependen de la frecuencia. A bajas frecuencias, esas ondas cruzan bien casi cualquier obstáculo, pero la potencia se reduce de manera drástica a medida que se aleja de la fuente, aproximadamente en proporción a $1/r^2$ en el aire. A frecuencias altas, las ondas de radio tienden a viajar en línea recta y a rebotar en los obstáculos. También son absorbidas por la lluvia. En todas las frecuencias, las ondas de radio están sujetas a interferencia por los motores y otros equipos eléctricos.

Por la capacidad del radio de viajar largas distancias, la interferencia entre usuarios es un problema. Por esta razón, todos los gobiernos reglamentan estrictamente el uso de radiotransmisores, con una excepción, que veremos más adelante.

En las bandas VLF, LF y MF las ondas de radio siguen la curvatura de la Tierra, como se ilustra en la figura 2-12(a). Estas ondas se pueden detectar quizás a 1000 km en las frecuencias más bajas, y a menos en frecuencias más altas. La difusión de radio AM usa la banda MF, y es por ello que las estaciones de radio AM de Boston no se pueden oír con facilidad en Nueva York. Las ondas de radio en estas bandas cruzan con facilidad los edificios, y es por ello que los radios portátiles funcionan en interiores. El problema principal al usar bandas para comunicación de datos es su ancho de banda bajo (vea la ecuación 2-3).

En las bandas HF y VHF, las ondas a nivel del suelo tienden a ser absorbidas por la tierra. Sin embargo, las ondas que alcanzan la ionosfera, una capa de partículas cargadas que rodea a la Tierra a una altura de 100 a 500 km, se refractan y se envían de regreso a nuestro planeta, como se muestra en la figura 2-12(b). En ciertas condiciones atmosféricas, las señales pueden rebotar varias veces. Los operadores de radio aficionados usan estas bandas para conversar a larga distancia. El ejército se comunica también en las bandas HF y VHF.

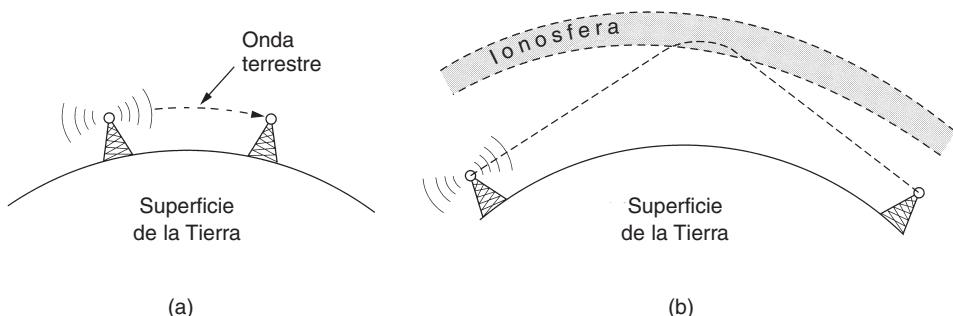


Figura 2-12. (a) En las bandas VLF, LF y MF, las ondas de radio siguen la curvatura de la Tierra.
 (b) En la banda HF las ondas rebotan en la ionosfera.

2.3.3 Transmisión por microondas

Por encima de los 100 MHz las ondas viajan en línea recta y, por lo tanto, se pueden enfocar en un haz estrecho. Concentrar toda la energía en un haz pequeño con una antena parabólica (como el tan familiar plato de televisión por satélite) produce una relación señal a ruido mucho más alta, pero las antenas transmisora y receptora deben estar bien alineadas entre sí. Además, esta direccionalidad permite que varios transmisores alineados en una fila se comuniquen sin interferencia con varios receptores en fila, siempre y cuando se sigan algunas reglas de espaciado. Antes de la fibra óptica, estas microondas formaron durante décadas el corazón del sistema de transmisión telefónica de larga distancia. De hecho, MCI, uno de los primeros competidores de AT&T después de que esta compañía fue desregularizada, construyó todo su sistema utilizando las comunicaciones mediante microondas que iban de torre en torre ubicadas a decenas de kilómetros una de la otra. Incluso el nombre de la compañía reflejó esto (MCI son las siglas de Microwave Communications, Inc.). Tiempo después, MCI adoptó la fibra y se fusionó con WorldCom.

Ya que las microondas viajan en línea recta, si las torres están muy separadas, partes de la Tierra estorbarán (piense en un enlace de San Francisco a Ámsterdam). Como consecuencia, se necesitan repetidores periódicos. Cuanto más altas sean las torres, más separadas pueden estar. La distancia entre los repetidores se eleva en forma muy aproximada con la raíz cuadrada de la altura de las torres. Con torres de 100 m de altura, los repetidores pueden estar separados a 80 km de distancia.

A diferencia de las ondas de radio a frecuencias más bajas, las microondas no atraviesan bien los edificios. Además, aun cuando el haz puede estar bien enfocado en el transmisor, hay cierta divergencia en el espacio. Algunas ondas pueden refractarse en las capas atmosféricas más bajas y tardar un poco más en llegar que las ondas directas. Las ondas diferidas pueden llegar fuera de fase con la onda directa y cancelar así la señal. Este efecto se llama **desvanecimiento por múltiples trayectorias** y con frecuencia es un problema serio que depende del clima y de la frecuencia. Algunos operadores mantienen 10% de sus canales inactivos como repuesto para activarlos cuando el desvanecimiento por múltiples trayectorias cancela en forma temporal alguna banda de frecuencia.

La creciente demanda de espectro obliga a los operadores a usar frecuencias más altas. Las bandas de hasta 10 GHz ahora son de uso rutinario, pero con las de aproximadamente 4 GHz surge un problema: son absorbidas por el agua. Estas ondas sólo tienen unos centímetros de longitud y la lluvia las absorbe. Este efecto sería útil si se quisiera construir un enorme horno de microondas externo para rostizar a los pájaros que pasen por ahí, pero para la comunicación es un problema grave. Al igual que con el desvanecimiento por múltiples trayectorias, la única solución es interrumpir los enlaces afectados por la lluvia y enrutar la comunicación por otra trayectoria.

En resumen, la comunicación por microondas se utiliza tanto para la comunicación telefónica de larga distancia, los teléfonos celulares, la distribución de la televisión, etcétera, que el espectro se ha vuelto muy escaso. Esta tecnología tiene varias ventajas significativas respecto a la fibra. La principal es que no se necesita derecho de paso; basta con comprar un terreno pequeño cada 50 km y construir en él una torre de microondas para saltarse el sistema telefónico y comunicarse en forma directa. Así es como MCI logró establecerse tan rápidamente como una compañía nueva telefónica de larga distancia. (Sprint siguió un camino totalmente diferente: la fundó el ferrocarril Southern Pacific Railroad, que ya poseía una gran cantidad de derechos de paso, limitándose a enterrar la fibra junto a las vías.)

Las microondas también son relativamente baratas. Erigir dos torres sencillas (podrían ser simplemente postes grandes con cables de retén) y poner antenas en cada una puede costar menos que enterrar 50 km de fibra a través de un área urbana congestionada o sobre una montaña, y también puede ser más económico que rentar la fibra de la compañía de teléfonos, en especial si ésta aún no ha recuperado por completo la inversión hecha por el cobre que quitó cuando instaló la fibra.

Las políticas del espectro electromagnético

Para evitar el caos total, hay acuerdos nacionales e internacionales acerca de quién utiliza cuáles frecuencias. Puesto que todos desean una tasa de transferencia de datos más alta, también desean más espectro. Los gobiernos nacionales asignan espectros para la radio AM y FM, la televisión y los teléfonos móviles, así como para las compañías telefónicas, la policía, la marina, la navegación, la milicia, el gobierno y muchos otros usuarios en competencia. A nivel mundial, una agencia de la ITU-R (WARC) trata de coordinar esta asignación de manera que se puedan fabricar los dispositivos que operan en diversos países. Sin embargo, los países no están atados a las recomendaciones de la ITU-R por lo que la FCC (Comisión Federal de Comunicaciones), que hace la asignación para Estados Unidos, ha rechazado ocasionalmente las recomendaciones de la ITU-R (por lo general, porque estas recomendaciones pedían a algún grupo políticamente poderoso que cediera una parte del espectro).

Incluso cuando una parte del espectro se ha asignado para un uso en particular, como para los teléfonos móviles, existe el aspecto adicional de cuál empresa portadora tiene permitido utilizar cuáles frecuencias. En el pasado se utilizaban tres algoritmos. El más antiguo, llamado **concurso de méritos** (*beauty contest*), requiere que cada empresa portadora explique por qué su propuesta es la que sirve mejor para los intereses públicos. Después los funcionarios del gobierno deciden

cuál de todas esas historias los convence más. Debido a que alguno de estos funcionarios otorgaban propiedad valuada en miles de millones de dólares a la compañía de su preferencia, esto conducía a soborno, corrupción, nepotismo, etcétera. Además, incluso un funcionario escrupulosamente honesto que piense que una compañía extranjera podría hacer mejor trabajo que cualquiera de las nacionales, tiene que dar muchas explicaciones.

Esta observación nos lleva al segundo algoritmo, en el que se realiza un sorteo entre las compañías interesadas. El problema con esta idea es que las compañías que no tienen ningún interés en utilizar el espectro, pueden entrar al sorteo. Por ejemplo, si un restaurante de comida rápida o una cadena de zapaterías gana, puede revender el espectro a una empresa portadora, sacando una ganancia inmensa y sin ningún riesgo.

La concesión de ganancias inesperadas a compañías atentas, aunque aleatorias, ha sido severamente criticada por muchos, lo que nos lleva al algoritmo 3: subastar el ancho de banda al mejor postor. Cuando en el año 2000 Inglaterra subastó las frecuencias necesarias para los sistemas móviles de la tercera generación, esperaba obtener aproximadamente \$4 mil millones. En realidad recibió \$40 mil millones debido a que las empresas portadoras cayeron en la desesperación ante la posibilidad de perder el mercado móvil. Este evento despertó la avaricia de los gobiernos vecinos y los motivó a llevar a cabo sus propias subastas. Funcionó, pero también dejó a algunas empresas portadoras con deudas enormes que ahora las tienen al borde de la bancarrota. Incluso en los mejores casos, les tomará muchos años recuperar la inversión en la licencia.

Un enfoque totalmente diferente para asignar frecuencias es no asignarlas por completo. Tan sólo se deja que todos transmitan a voluntad, pero se regula la potencia utilizada de manera que las estaciones tengan un rango tan corto que no interfieran entre ellas. Por consiguiente, la mayoría de los gobiernos han apartado algunas bandas de frecuencias, llamadas bandas **ISM (industriales, médicas y científicas)** de uso no autorizado. Los dispositivos para abrir puertas de garaje, teléfonos inalámbricos, juguetes controlados por radio, ratones inalámbricos y muchos otros dispositivos inalámbricos domésticos utilizan las bandas ISM. Para minimizar la interferencia entre estos dispositivos no coordinados, la FCC exige que todos los dispositivos que utilizan las bandas ISM utilicen técnicas de espectro disperso. En algunos otros países se aplican reglas similares.

La ubicación de las bandas ISM varía un poco de país a país. Por ejemplo, en Estados Unidos los dispositivos cuya potencia esté debajo de 1 watt, pueden utilizar las bandas que se muestran en la figura 2-13 sin requerir una licencia de la FCC. La banda de 900 MHz funciona mejor, pero está atestada y no está disponible en todo el mundo. La banda de 2.4 GHz está disponible en la mayoría de los países, pero está sujeta a interferencia por parte de los hornos de microondas e instalaciones de radar. Bluetooth y algunas de las LANs inalámbricas 802.11 operan en esta banda. La banda de 5.7 GHz es nueva y no se ha desarrollado del todo, por lo que el equipo que la utiliza es costoso, pero debido a que 802.11a la utiliza, se popularizará con rapidez.

2.3.4 Ondas infrarrojas y milimétricas

Las ondas infrarrojas y milimétricas no guiadas se usan mucho para la comunicación de corto alcance. Todos los controles remotos de los televisores, grabadoras de video y estéreos utilizan comunicación infrarroja. Estos controles son relativamente direccionales, económicos y fáciles de

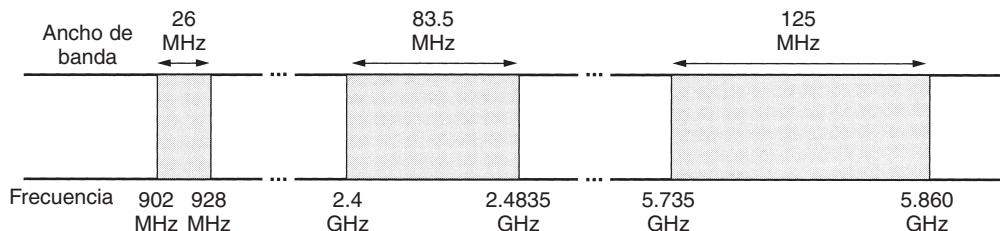


Figura 2-13. Las bandas ISM de Estados Unidos.

construir, pero tienen un inconveniente importante: no atraviesan los objetos sólidos (párese entre su televisor y su control remoto y vea si todavía funciona). En general, conforme pasamos de la radio de onda larga hacia la luz visible, las ondas se comportan cada vez más como la luz y cada vez menos como la radio.

Por otro lado, el hecho de que las ondas infrarrojas no atraviesen bien las paredes sólidas también es una ventaja. Esto significa que un sistema infrarrojo en un cuarto de un edificio no interferirá con un sistema similar en cuartos adyacentes. Por esta razón, la seguridad de estos sistemas contra el espionaje es mejor que la de los sistemas de radio. Además, no es necesario obtener licencia del gobierno para operar un sistema infrarrojo, en contraste con los sistemas de radio, que deben tener licencia afuera de las bandas ISM. La comunicación infrarroja tiene un uso limitado en el escritorio; por ejemplo, para conectar computadoras portátiles e impresoras, aunque no es un protagonista principal en el juego de la comunicación.

2.3.5 Transmisión por ondas de luz

La señalización óptica sin guías se ha utilizado durante siglos. Paul Revere utilizó señalización óptica binaria desde la Iglesia Old North justo antes de su famoso viaje. Una aplicación más moderna es conectar las LANs de dos edificios por medio de láseres montados en sus azoteas. La señalización óptica coherente con láseres es inherentemente unidireccional, de modo que cada edificio necesita su propio láser y su propio fotodetector. Este esquema ofrece un ancho de banda muy alto y un costo muy bajo. También es relativamente fácil de instalar y, a diferencia de las microondas, no requiere una licencia de la FCC.

Sin embargo, la ventaja del láser, un haz muy estrecho, aquí también es una debilidad. Apuntar un rayo láser de 1 mm de anchura a un blanco del tamaño de la punta de un alfiler a 500 m de distancia requiere la puntería de una Annie Oakley moderna. Por lo general, se añaden lentes al sistema para desenfocar ligeramente el rayo.

Una desventaja es que los rayos láser no pueden penetrar la lluvia ni la niebla densa, pero normalmente funcionan bien en días soleados. Sin embargo, en una ocasión el autor asistió a una conferencia en un moderno hotel de Europa en el que los organizadores tuvieron la atención de proporcionar un salón lleno de terminales para que los asistentes leyieran su correo electrónico durante las presentaciones aburridas. Puesto que la PTT local no estaba dispuesta a instalar un gran

número de líneas telefónicas sólo para tres días, los organizadores colocaron un láser en el techo, lo apuntaron al edificio de ciencias de la computación de su universidad, el cual está a unos cuantos kilómetros de allí; lo probaron la noche anterior a la conferencia y funcionó a la perfección. A las 9 a.m. del siguiente día, que era brillante y soleado, el enlace falló por completo y permaneció caído todo el día. Esa noche los organizadores volvieron a probar con mucho cuidado el enlace y de nuevo funcionó a la perfección. El patrón se repitió durante dos días más de forma idéntica.

Después de la conferencia, los organizadores descubrieron el problema. Durante el día, el calor del sol causaba corrientes de convección que se elevaban desde el techo del edificio, como se muestra en la figura 2-14. Este aire turbulento desviaba el rayo y lo hacía danzar alrededor del detector. Una “vista” atmosférica como ésta hace titilar a las estrellas (y es la razón por la cual los astrónomos ponen sus telescopios en las cimas de las montañas, para quedar tan arriba en la atmósfera como sea posible). Este fenómeno es también la causa del aspecto trémulo de las carreteras en un día caluroso y de las imágenes ondulantes cuando se mira sobre un radiador caliente.

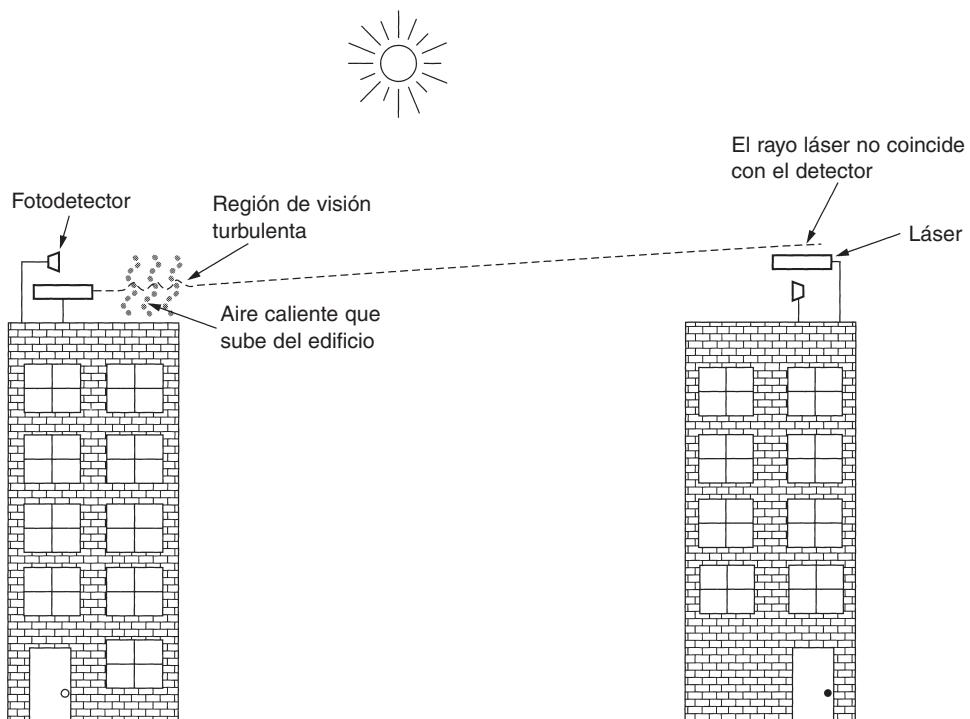


Figura 2-14. Las corrientes de convección pueden interferir los sistemas de comunicación por láser. Aquí se ilustra un sistema bidireccional con dos láseres.

2.4 SATÉLITES DE COMUNICACIONES

En la década de 1950 y principios de la de 1960, hubo intentos por establecer sistemas de comunicación mediante el rebote de señales sobre globos climáticos. Por desgracia, las señales que se recibían eran demasiado débiles para darles un uso práctico. Entonces, la Marina de Estados Unidos descubrió una especie de globo climático en el cielo —la Luna— y desarrolló un sistema de comunicaciones por repetición (o de barco a tierra) que rebotaba señales de él.

Progresos posteriores en el campo de las comunicaciones por el cielo tuvieron que esperar hasta que se lanzó el primer satélite de comunicaciones. La principal diferencia entre un satélite artificial y uno real es que el primero puede amplificar las señales antes de mandarlas de regreso, convirtiendo lo que parecía una idea estrañamente genial en un poderoso sistema de comunicaciones.

Los satélites de comunicaciones tienen algunas propiedades interesantes que los hacen atractivos para muchas aplicaciones. En su forma más simple, un satélite de comunicaciones se puede considerar como un enorme repetidor de microondas en el cielo. Contiene numerosos **transpondedores**, cada uno de los cuales se encarga de una parte del espectro, amplifica la señal entrante y a continuación la retransmite en otra frecuencia para evitar interferencia con la señal entrante. Los haces pueden ser amplios y cubrir una fracción sustancial de la superficie de la Tierra, o estrechos, y abarcar sólo algunos cientos de kilómetros de diámetro. Este modo de operación se conoce como de **tubo doblado**.

De acuerdo con la ley de Kepler, el periodo orbital de un satélite varía según el radio de la órbita a la $3/2$ potencia. Entre más alto esté el satélite, más largo es el periodo. Cerca de la superficie de la Tierra, el periodo es de aproximadamente 90 minutos. En consecuencia, los satélites con órbitas bajas desaparecen de la vista con bastante rapidez, aunque algunos de ellos son necesarios para proporcionar una cobertura continua. A una altitud de cerca de 35,800 km, el periodo es de 24 horas. A una de 384,000 km, el periodo es cercano a un mes, como puede atestiguar cualquiera que haya observado la Luna con regularidad.

El periodo de un satélite es importante, aunque no es el único punto para determinar dónde colocarlo. Otro aspecto es la presencia de los cinturones de Van Allen, capas de partículas altamente cargadas de energía, atrapadas por el campo magnético de la Tierra. Cualquier satélite que vuela dentro de ellas sería destruido rápidamente por las partículas con una alta carga de energía. Del análisis de estos factores resulta que hay tres regiones para colocar con seguridad los satélites. En la figura 2-15 se muestran estas regiones y algunas de sus propiedades. Enseguida describiremos brevemente los satélites que habitan cada una de estas regiones.

2.4.1 Satélites geoestacionarios

En 1945, el escritor de ciencia-ficción Arthur C. Clarke calculó que un satélite a una altitud de 35,800 km en una órbita ecuatorial circular aparecería permanecer inmóvil en el cielo, por lo que no sería necesario rastrearlo (Clarke, 1945). Se dio a la tarea de describir un sistema de comunicaciones completo que utilizaba estos (tripulados) **satélites geoestacionarios**, incluyendo

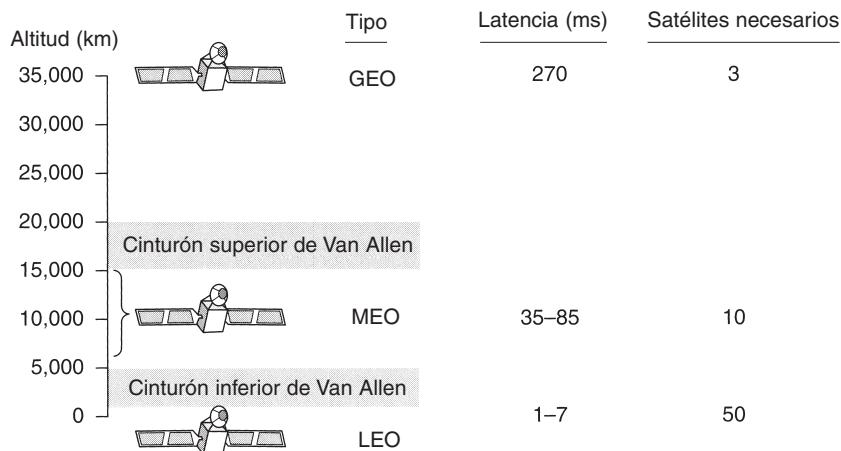


Figura 2-15. Satélites de comunicaciones y algunas de sus propiedades, entre ellas: altitud sobre la Tierra, tiempo de duración de un viaje de ida y vuelta y la cantidad de satélites necesarios para abarcar toda la Tierra.

las órbitas, paneles solares, radiofrecuencias y procedimientos de lanzamiento. Desafortunadamente, llegó a la conclusión de que los satélites no eran prácticos debido a la imposibilidad de poner en órbita amplificadores de tubos catódicos frágiles que consumían una gran cantidad de energía, por lo cual nunca le dio seguimiento a esta idea, aunque escribió algunos relatos de ciencia ficción al respecto.

La invención del transistor cambió las cosas, y el primer satélite de comunicaciones artificial, Telstar, fue lanzado en julio de 1962. Desde entonces, los satélites de comunicaciones se han convertido en un negocio multimillonario y en el único aspecto del espacio exterior altamente rentable. Con frecuencia, a estos satélites que vuelan a grandes alturas se les llama satélites **GEO (Órbita Terrestre Geoestacionaria)**.

Con la tecnología actual, es poco aconsejable utilizar satélites geoestacionarios espaciados a menos de dos grados en el plano ecuatorial de 360 grados para evitar interferencia. Con un espaciamiento de dos grados, sólo puede haber $360/2 = 180$ de estos satélites a la vez en el cielo. Sin embargo, cada transpondedor puede utilizar múltiples frecuencias y polarizaciones para incrementar el ancho de banda disponible.

Para evitar el caos total en el cielo, la ITU asigna la posición orbital. Este proceso tiene fuertes connotaciones políticas, y países que apenas están saliendo de la edad de piedra demandan “sus” posiciones orbitales (con el propósito de alquilarlas al mejor postor). No obstante, algunos países sostienen que la propiedad no se extiende a la Luna y que ningún país tiene derechos legales sobre las posiciones orbitales que se encuentran arriba de su territorio. Por si esto no fuera suficiente, las telecomunicaciones comerciales no son la única aplicación. Las compañías televisoras, los gobiernos y la milicia también quieren su tajada del pastel orbital.

Los satélites modernos pueden ser bastante grandes, pesar hasta 4000 kg y consumir varios kilowatts de electricidad producida por paneles solares. La gravedad del Sol, la Luna y los planetas

tiende a desplazar a los satélites de sus órbitas y orientaciones asignadas, efecto contrarrestado por los motores turbo integrados de los satélites. Esta actividad de ajuste se conoce como **control de la posición orbital**. Sin embargo, cuando se termina el combustible de los motores, por lo general a los 10 años, el satélite navega a la deriva y cae sin remedio, por lo cual es necesario desactivarlo. Con el tiempo, la órbita se deteriora y el satélite reingresa a la atmósfera y se incendia o en ocasiones se estrella contra la Tierra.

Las posiciones orbitales no son la única manzana de la discordia. También lo son las frecuencias, debido a que las transmisiones de los enlaces descendentes interfieren con los usuarios de microondas existentes. En consecuencia, la ITU ha asignado bandas de frecuencia específicas a los usuarios de satélites. Las principales se muestran en la figura 2-16. La banda C fue la primera que se destinó al tráfico comercial por satélite. Tiene dos rangos de frecuencia, el inferior para el tráfico descendente o de bajada (proveniente del satélite) y el superior para el tráfico ascendente o de subida (hacia el satélite). Para permitir que el tráfico fluya en ambos sentidos al mismo tiempo, se requieren dos canales, uno para cada sentido. Estas bandas están sobresaturadas debido a que las empresas portadoras también las utilizan para los enlaces de microondas terrestres. Las bandas L y S fueron incorporadas en el año 2000 mediante un acuerdo internacional. No obstante, son estrechas y saturadas.

Banda	Enlace descendente	Enlace ascendente	Ancho de banda	Problemas
L	1.5 GHz	1.6 GHz	15 MHz	Bajo ancho de banda; saturada
S	1.9 GHz	2.2 GHz	70 MHz	Bajo ancho de banda; saturada
C	4.0 GHz	6.0 GHz	500 MHz	Interferencia terrestre
Ku	11 GHz	14 GHz	500 MHz	Lluvia
Ka	20 GHz	30 GHz	3500 MHz	Lluvia, costo del equipo

Figura 2-16. Principales bandas de satélite.

La siguiente banda más ancha disponible para los operadores de telecomunicaciones es la banda Ku (K abajo). Esta banda aún no está saturada, y a estas frecuencias es posible espaciar los satélites a cerca de un grado. No obstante, hay otro problema: la lluvia. El agua es un gran absorbente de estas microondas cortas. La buena noticia es que por lo general las tormentas se confinan a sitios específicos, por lo que el problema se soluciona con la instalación de varias estaciones terrestres con suficiente separación en vez de una sola, al costo de más antenas, cables y aparatos electrónicos que permitan pasar rápidamente de una estación a otra. También se ha asignado ancho de banda para tráfico comercial por satélite en la banda Ka (K arriba), pero el equipo necesario para utilizar esta banda aún es caro. Además de estas bandas comerciales, también hay muchas bandas gubernamentales y militares.

Un satélite moderno tiene alrededor de 40 transpondedores, cada uno con un ancho de banda de 80 MHz. Por lo general, cada transpondedor opera como un tubo doblado, pero algunos satélites recientes tienen capacidad de procesamiento a bordo, lo cual les permite una operación más refinada. La división de los transpondedores en canales era estática en los primeros satélites: el

ancho de banda se dividía simplemente en bandas de frecuencia fija. En nuestros días, cada haz del transpondedor se divide en ranuras temporales, y varios usuarios su turnan para utilizarlo. Más tarde en este mismo capítulo analizaremos en detalle estas dos técnicas (multiplexión por división de frecuencia y multiplexión por división de tiempo).

Los primeros satélites geoestacionarios tenían un solo haz espacial que iluminaba cerca de un tercio de la superficie de la Tierra, al cual se le conoce como **huella**. Con la considerable reducción en precio, tamaño y requerimientos de energía de los componentes microelectrónicos, se ha vuelto posible una estrategia de difusión mucho más refinada. Cada satélite está equipado con múltiples antenas y transpondedores. Cada haz descendente se puede concentrar en un área geográfica pequeña, de tal forma que es posible llevar a cabo simultáneamente una gran cantidad de transmisiones hacia y desde el satélite. Normalmente, estos haces, conocidos como **haces reducidos**, tienen forma elíptica y pueden ser tan pequeños como algunos cientos de kilómetros. Por lo general, un satélite de comunicaciones para los Estados Unidos de América tiene un haz ancho para los 48 estados contiguos y haces reducidos para Alaska y Hawaii.

Un avance reciente en el mundo de los satélites de comunicaciones es el desarrollo de microestaciones de bajo costo, llamadas **VSATs (Terminales de Apertura Muy Pequeña)** (Abramson, 2000). Estas diminutas terminales tienen antenas de un metro o más pequeñas (en comparación con los 10 metros que mide una antena GEO estándar) y pueden producir alrededor de un watt de energía. Por lo general, el enlace ascendente funciona a 19.2 kbps, pero el enlace descendente funciona con frecuencia a 512 kbps o más. La televisión de difusión directa por satélite utiliza esta tecnología para transmisión unidireccional.

En muchos sistemas VSAT, las microestaciones no tienen suficiente potencia para comunicarse directamente una con la otra (a través del satélite, por supuesto). En vez de ello, como se muestra en la figura 2-17, es necesaria una estación especial en tierra, la **estación central**, que cuenta con una antena grande, para retransmitir el tráfico entre VSATs. En este modo de operación, el emisor o el receptor tienen una antena grande y un amplificador potente. La desventaja es que existe un retardo más prolongado al contar con estaciones de usuario más económicas.

Las VSATs tienen un futuro prometedor en las zonas rurales. Aún no tienen una amplia aceptación, pero más de la mitad de la población del mundo vive a una hora de distancia del teléfono más cercano. El tendido de redes telefónicas a miles de pequeñas poblaciones excede el presupuesto de la mayoría de los gobiernos del tercer mundo, pero lo que sí es factible es la instalación de antenas VSAT de un metro alimentadas por celdas solares. Las VSATs proporcionarán la tecnología que enlazará al mundo.

Los satélites de comunicaciones tienen diversas propiedades radicalmente distintas a las de los enlaces terrestres de punto a punto. Para empezar, aun cuando las señales hacia y desde un satélite viajan a la velocidad de la luz (cerca de 300,000 km/seg), el largo viaje de ida y vuelta provoca un retardo sustancial para los satélites GEO. Dependiendo de la distancia entre el usuario y la estación terrestre, así como de la elevación del satélite en el horizonte, el tiempo de tránsito de un extremo al otro es de entre 250 y 300 msec. Un valor común es de 270 msec (540 msec para un sistema VSAT con una estación central).

Con propósitos de comparación, los enlaces terrestres de microondas tienen un retardo de propagación de casi 3 μ seg/km, en tanto que los enlaces de cable coaxial o la fibra óptica tienen un

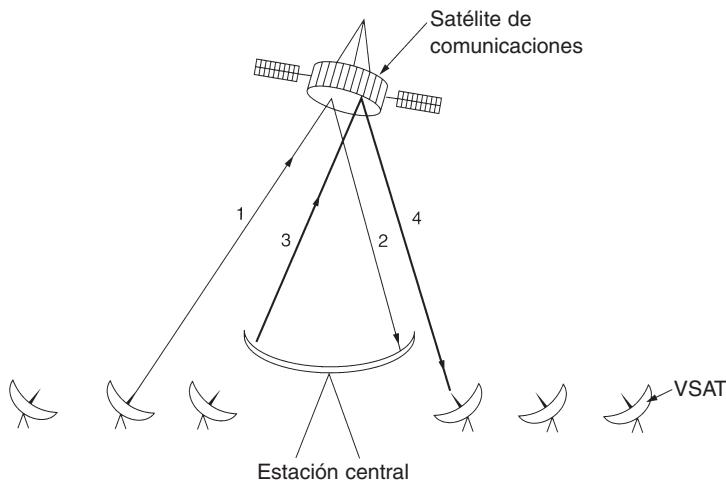


Figura 2-17. VSATs con una estación central.

retardo de aproximadamente 5 μ seg/km. El último es más lento que el primero debido a que las señales electromagnéticas viajan más rápido en el aire que en materiales sólidos.

Otra propiedad importante de los satélites es que son esencialmente medios de difusión. No cuesta más enviar un mensaje a miles de estaciones dentro de la huella de un transpondedor de lo que cuesta enviarlo a una sola estación. Esta propiedad es muy útil para algunas aplicaciones. Por ejemplo, es posible que un satélite difunda páginas Web populares a los cachés de una gran cantidad de computadoras diseminadas en un área amplia. Aun cuando la difusión se puede simular con líneas punto a punto, la difusión por satélite es mucho más económica. Por otro lado, los satélites son un verdadero desastre en el aspecto de seguridad y privacidad: cualquiera puede escuchar todo. La encriptación es esencial cuando se requiere seguridad.

Los satélites también tienen la propiedad de que el costo de transmitir un mensaje es independiente de la distancia que se recorra. Una llamada al otro lado del océano tiene el mismo costo que una al otro lado de la calle. Los satélites también cuentan con excelentes tasas de error y se pueden desplegar de manera casi instantánea, un aspecto importante para las comunicaciones militares.

2.4.2 Satélites de Órbita Terrestre Media

Los satélites **MEO (Órbita Terrestre Media)** se encuentran a altitudes mucho más bajas, entre los dos cinturones de Van Allen. Vistos desde la Tierra, estos satélites se desplazan lentamente y tardan alrededor de seis horas para dar la vuelta a la Tierra. Por consiguiente, es necesario rastrearlos conforme se desplazan. Puesto que son menores que los GEO, tienen una huella más pequeña y se requieren transmisores menos potentes para alcanzarlos. Hoy en día no se utilizan para telecomunicaciones, por lo cual no los examinaremos aquí. Los 24 satélites **GPS (Sistema de Posicionamiento Global)** que orbitan a cerca de 18,000 km son ejemplos de satélites MEO.

2.4.3 Satélites de Órbita Terrestre Baja

En una altitud más baja encontramos a los satélites **LEO (Órbita Terrestre Baja)**. Debido a la rapidez de su movimiento, se requieren grandes cantidades de ellos para conformar un sistema completo. Por otro lado, como los satélites se encuentran tan cercanos a la Tierra, las estaciones terrestres no necesitan mucha potencia, y el retardo del viaje de ida y vuelta es de tan sólo algunos milisegundos. En esta sección examinaremos tres ejemplos, dos sobre las comunicaciones de voz y uno sobre el servicio de Internet.

Iridium

Como ya mencionamos, durante los primeros 30 años de la era de los satélites casi no se utilizaban los satélites de órbita baja porque aparecían y desaparecían con mucha rapidez. En 1990, Motorola abrió un nuevo camino al solicitar permiso a la FCC (Comisión Federal de Comunicaciones de Estados Unidos) para lanzar 77 satélites de órbita baja para el proyecto Iridium (el iridio es el elemento 77). El plan fue modificado más tarde para utilizar sólo 66 satélites, por lo que el proyecto debió haberse renombrado como Dysprosium (elemento 66), pero quizás este nombre sonaba demasiado a enfermedad. El propósito era que tan pronto como un satélite se perdiera de vista, otro lo reemplazaría. Esta propuesta desató un frenesí entre otras compañías. De pronto, todos querían lanzar una cadena de satélites de órbita baja.

Después de siete años de improvisación de socios y financiamiento, los socios lanzaron los satélites Iridium en 1997. El servicio de comunicaciones empezó en noviembre de 1998. Por desgracia, la demanda comercial de teléfonos por satélite grandes y pesados fue insignificante porque la red telefónica móvil había crecido de manera espectacular desde 1990. En consecuencia, el proyecto Iridium no fue rentable y quebró en agosto de 1999 en lo que fue uno de los fracasos corporativos más espectaculares de la historia. Los satélites y otros activos (con valor de 5000 millones de dólares) fueron adquiridos posteriormente por un inversionista en 25 millones de dólares en una especie de venta de garaje extraterrestre. El servicio Iridium se reinició en marzo de 2001.

El negocio de Iridium era (y es) ofrecer servicio de telecomunicaciones en todo el mundo a través de dispositivos de bolsillo que se comunican directamente con los satélites Iridium. Proporciona servicio de voz, datos, búsqueda de personas, fax y navegación en cualquier parte, sea en tierra, mar y aire. Entre sus clientes están las industrias marítima, de la aviación y exploración petrolera, así como personas que viajan a partes del mundo que carecen de infraestructura de telecomunicaciones (por ejemplo, desiertos, montañas, selvas y algunos países del tercer mundo).

Los satélites Iridium están a una altitud de 750 km, en órbitas polares circulares. Están dispuestos en forma de collar de norte a sur, con un satélite a cada 32 grados de latitud. La Tierra completa se cubre con seis collares, como se aprecia en la figura 2-18(a). Quienes no tengan muchos conocimientos sobre química pueden pensar que esta disposición es un gran átomo de dispropósito, con la Tierra como núcleo y los satélites como electrones.

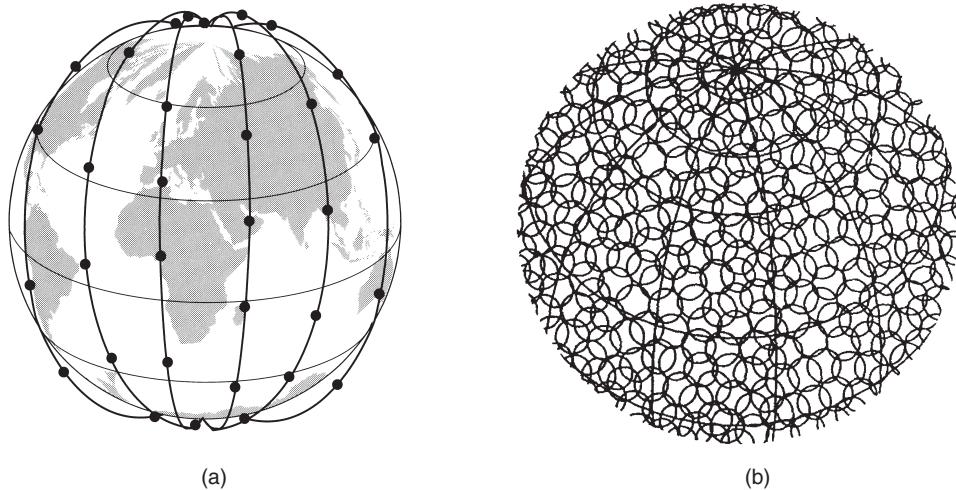


Figura 2-18. (a) Los satélites Iridium forman seis collares alrededor de la Tierra. (b) 1628 celdas en movimiento cubren la Tierra.

Cada satélite tiene un máximo de 48 celdas (haces reducidos), con un total de 1628 celdas sobre la superficie de la Tierra, como se muestra en la figura 2-18(b). Cada satélite tiene una capacidad de 3840 canales, o 253,440 en total. Algunos de estos canales se utilizan para localización de personas y navegación, en tanto que otros, para datos y voz.

Una propiedad interesante de Iridium es que la comunicación entre clientes distantes tiene lugar en el espacio, con un satélite retransmitiendo datos al siguiente, como se muestra en la figura 2-19(a). Aquí vemos que quien llama está en el Polo Norte y hace contacto con un satélite que se encuentra directamente arriba de él. La llamada se retransmite a través de otros satélites y por último es entregada al destinatario en el Polo Sur.

Globalstar

Globalstar es un diseño alterno para Iridium. Se basa en 48 satélites LEO pero utiliza un esquema de conmutación diferente al de Iridium. En tanto que Iridium retransmite las llamadas de satélite en satélite, lo cual requiere un equipo de conmutación refinado en los satélites, Globalstar utiliza un diseño de tubo doblado tradicional. La llamada que se originó en el Polo Norte en la figura 2-19(b) es devuelta a la Tierra y recogida por la enorme estación terrestre. A continuación la llamada se enruta, a través de una red terrestre, a la estación terrestre más cercana al destinatario y se entrega mediante una conexión de tubo doblado como se muestra. La ventaja de este esquema es que mucha de la complejidad queda en tierra, donde es más sencillo manejarla. Asimismo,

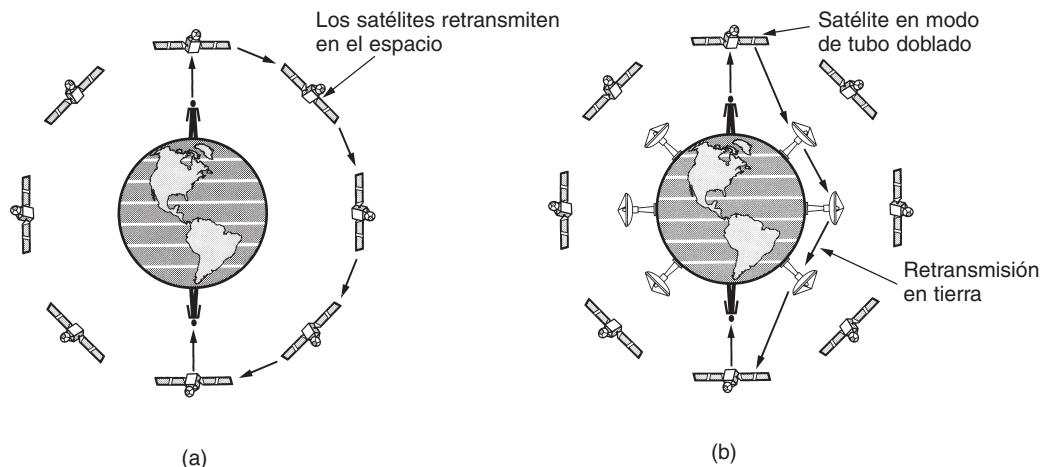


Figura 2-19. (a) Retransmisión en el espacio. (b) Retransmisión en tierra.

el uso de antenas grandes en las estaciones terrestres que pueden producir una señal potente y recibir una señal débil, permite la utilización de teléfonos de baja potencia. Después de todo, el teléfono produce tan sólo unos cuantos miliwatts de potencia, por lo cual la señal que llega a las estaciones terrestres es sumamente débil, aun cuando el satélite la haya amplificado.

Teledesic

Iridium está destinada a usuarios de teléfonos que se encuentran en lugares extremos. Nuestro siguiente ejemplo, **Teledesic**, está destinada a usuarios de Internet de todo el mundo deseosos de ancho de banda. Fue concebida en 1990 por Craig McCaw, pionero de la telefonía móvil, y por Bill Gates, fundador de Microsoft, quienes estaban inconformes con el lento ritmo al cual las compañías telefónicas de todo el mundo proporcionaban ancho de banda alto a los usuarios de computadoras. La meta del sistema Teledesic es ofrecer a los millones de usuarios concurrentes de Internet un enlace ascendente de hasta 100 Mbps y un enlace descendente de hasta 720 Mbps mediante antenas tipo VSAT pequeñas y fijas, que ignoran por completo el sistema telefónico. Para las compañías telefónicas esto es demasiado bello para ser realidad.

El diseño original consistía en un sistema de 288 satélites de huella pequeña, dispuestos en 12 planos justo debajo del cinturón inferior de Van Allen a una altitud de 1350 km. Posteriormente se modificó el diseño a 30 satélites con huellas más grandes. La transmisión se realiza en la banda Ka, relativamente poco saturada y con un ancho de banda alto. El sistema es de comutación de paquetes en el espacio, en el cual cada satélite tiene la capacidad de enrutar paquetes a los satélites vecinos. Cuando un usuario necesita ancho de banda para enviar paquetes, tal ancho de banda se solicita y asigna de manera dinámica en alrededor de 50 msec. Si todo marcha bien, el sistema está programado para empezar a funcionar en 2005.

2.4.4 Satélites en comparación con fibra óptica

Una comparación entre comunicación por satélite y comunicación terrestre es aleccionadora. Apenas hace 20 años se podía afirmar que el futuro de las comunicaciones estaba en los satélites. Después de todo, el sistema telefónico ha cambiado poco en los pasados 100 años y no hay señales de que cambie en los próximos 100 años. Este lento movimiento fue ocasionado en gran parte por las regulaciones que obligaban a las compañías telefónicas a ofrecer un buen servicio de voz a precios razonables (lo cual hicieron), y a cambio obtuvieron utilidades garantizadas sobre sus inversiones. Para quienes tenían que transmitir datos, había módems de 1200 bps. Por mucho, esto es todo lo que había.

Esta situación cambió radicalmente en 1984 con la entrada de la competencia en Estados Unidos y un poco más tarde en Europa. Las compañías telefónicas comenzaron a reemplazar sus viejas redes con fibra óptica e introdujeron servicios de ancho de banda alto como ADSL (Línea Digital de Suscriptor Asimétrica). También suspendieron su añeja práctica de cargar precios artificialmente altos a los usuarios de larga distancia para subsidiar el servicio local.

De buenas a primeras, las conexiones terrestres de fibra óptica dieron la impresión de que serían las ganadoras a largo plazo. No obstante, los satélites de comunicaciones tienen algunos nichos de mercado importantes a los cuales la fibra óptica no se dirige (en ocasiones porque no puede). A continuación veremos algunos de ellos.

En primer lugar, a pesar de que una fibra óptica tiene más ancho de banda potencial que todos los satélites que se han lanzado, este ancho de banda no está disponible para la mayoría de los usuarios. La fibra que se instala actualmente se utiliza en el sistema telefónico para manejar muchas llamadas de larga distancia al mismo tiempo, no para ofrecer un ancho de banda alto a los usuarios individuales. Con los satélites, es factible que un usuario instale una antena en el techo de la casa y evada por completo el sistema telefónico para conseguir un ancho de banda alto. Teledesic se apoya en esta idea.

Un segundo nicho es el de la comunicación móvil. En estos días mucha gente desea comunicarse mientras trotta, maneja, navega o vuela. Los enlaces terrestres de fibra óptica no sirven para este uso, pero los enlaces por satélite sí. Sin embargo, es posible que una combinación de radio celular y fibra óptica funcionara para la mayoría de los casos (aunque quizás no para aquellos que viajen por aire o por mar).

Un tercer nicho es para aquellas situaciones en las cuales se requiere difusión. Un mensaje enviado desde un satélite se puede recibir en miles de estaciones terrestres al mismo tiempo. Por ejemplo, para una organización que transmita un flujo de precios de acciones, bonos o commodities a miles de operadores de bolsa le resultaría más económico un sistema por satélite que simular la difusión en tierra.

Un cuarto nicho es el de las comunicaciones en lugares agrestes o con una infraestructura terrestre pobemente desarrollada. Por ejemplo, Indonesia tiene su propio satélite para el tráfico telefónico interno. El lanzamiento de un satélite resultó más económico que el enlace de miles de cables bajo el mar entre las 13,667 islas que conforman el archipiélago.

Un quinto nicho de mercado para los satélites son las áreas donde es difícil o extremadamente costoso conseguir un derecho para el tendido de fibra óptica.

Sexto, cuando un despliegue rápido es primordial, como en un sistema de comunicaciones militar en época de guerra, los satélites ganan con facilidad.

En resumen, al parecer la tendencia general de las comunicaciones en el futuro será la fibra óptica terrestre en combinación con radio celular, pero los satélites son mejores para algunos usos especializados. Sin embargo, hay un imponderable que se aplica en todos los casos: el aspecto económico. Aunque la fibra óptica ofrece más ancho de banda, es muy probable que las comunicaciones terrestres y por satélite competirán agresivamente en precio. Si los avances tecnológicos reducen de manera drástica el costo de despliegue de un satélite (por ejemplo, algún transbordador espacial futuro que pueda diseminar docenas de satélites en un solo lanzamiento) o los satélites de órbita baja se popularizan, no hay certeza de que la fibra óptica ganará en todos los mercados.

2.5 LA RED TELEFÓNICA PÚBLICA CONMUTADA

Cuando dos computadoras propiedad de la misma empresa u organización, localizadas cerca una de la otra, necesitan comunicarse, es más fácil conectarlas mediante un cable. Las LANs funcionan de esta manera. Sin embargo, cuando las distancias son considerables o hay muchas computadoras o los cables tienen que pasar por una vía pública o alguna zona restringida, los costos de tender cables privados por lo general son prohibitivos. Además, en casi todos los países del mundo también es ilegal el enlace de líneas de transmisión privadas a través (o por debajo) de una propiedad pública. En consecuencia, los diseñadores de redes deben depender de las instalaciones de telecomunicaciones existentes.

Por lo general, estas instalaciones, en especial la **PSTN (Red Telefónica Pública Conmutada)**, fueron diseñadas hace muchos años, con un propósito completamente distinto: transmitir la voz humana en una forma más o menos reconocible. Su aplicabilidad en las comunicaciones de computadora a computadora es muy limitada, pero esta situación está cambiando rápidamente con la introducción de la fibra óptica y la tecnología digital. De cualquier manera, el sistema telefónico está tan estrechamente entrecruzado con las redes de computadoras (de área amplia) que bien vale la pena dedicarle un poco de tiempo a su estudio.

Con el propósito de entender la importancia del problema, realicemos una comparación burda pero ilustrativa de las propiedades de una conexión típica de computadora a computadora a través de un cable local y otra mediante una línea de acceso telefónico. Un cable entre dos computadoras puede transferir datos a 10^9 bps, o tal vez un poco más. En contraste, una línea de acceso telefónico tiene una tasa máxima de datos de 56 kbps, una diferencia de un factor de casi 20,000. Es como la diferencia entre un pato contoneándose campantemente entre la hierba y un cohete a la Luna. Si la línea de acceso telefónico se reemplaza por una conexión ADSL, sigue habiendo una diferencia de un factor de 1000-2000.

Por supuesto, el problema es que los diseñadores de sistemas de cómputo suelen trabajar con sistemas de cómputo y cuando de repente se enfrentan con un sistema cuyo desempeño (según lo que ellos piensan) es tres o cuatro órdenes de magnitud peor, ellos, lo cual no es una sorpresa,

dedican mucho tiempo y esfuerzo para tratar de averiguar cómo utilizarlo de manera eficiente. En las siguientes secciones describiremos el sistema telefónico y mostraremos cómo funciona. Para obtener mayor información sobre los aspectos técnicos del sistema telefónico vea (Bellamy, 2000).

2.5.1 Estructura del sistema telefónico

Tan pronto como Alexander Graham Bell patentó el teléfono en 1876 (tan sólo unas cuantas horas antes que su rival, Elisha Gray), hubo una gran demanda por su nuevo invento. El mercado inicial fue para la venta de teléfonos, los cuales se vendían en pares. Le tocaba al cliente conectarlos con un solo alambre. Los electrones regresaban por tierra. Si el propietario de un teléfono deseaba comunicarse con otros n propietarios de teléfono, tenía que enlazar alambres individuales a todas las n casas. Después de un año, las ciudades se cubrieron de alambres que pasaban sobre las casas y los árboles convirtiéndose en una maraña. De inmediato quedó en claro que el modelo de conexión de cada teléfono con todos los demás, como se muestra en la figura 2-20(a), no iba a funcionar.

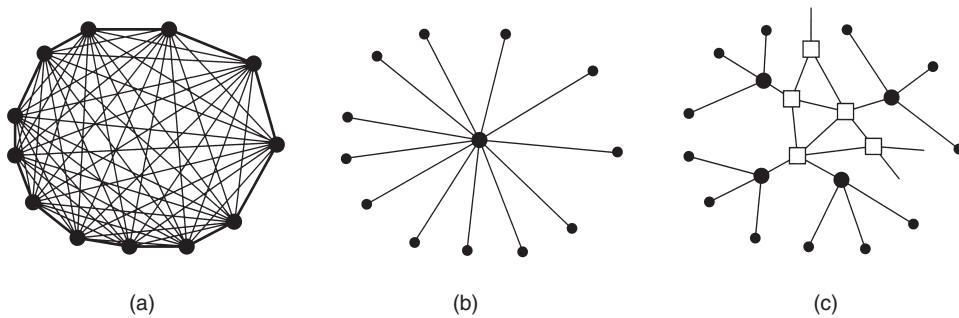


Figura 2-20. (a) Red totalmente interconectada. (b) Comutador centralizado. (c) Jerarquía de dos niveles.

Bell tuvo la suficiente visión para darse cuenta de esto y formó la Bell Telephone Company, la cual abrió su primera oficina de conmutación en 1878 (en New Haven, Connecticut). La compañía colocó un alambre en la casa u oficina de cada cliente. Para realizar una llamada, el cliente debía dar vueltas a una manivela en el teléfono para producir un sonido en la oficina de la compañía de teléfonos que atrajera la atención del operador, que a continuación conectaba manualmente a quien llamaba con el receptor de la llamada por medio de un cable puenteador. El modelo de la oficina de conmutación se muestra en la figura 2-20(b).

Muy pronto surgieron por todas partes oficinas de conmutación del Bell System y la gente quiso hacer llamadas de larga distancia entre ciudades, de modo que el Bell System empezó a conectar las oficinas de conmutación. El problema original pronto reapareció: conectar cada oficina de conmutación con todas las demás por medio de un cable entre ellas pronto dejó de ser práctico, así que se inventaron las oficinas de conmutación de segundo nivel. Poco después, fueron necesarias múltiples oficinas de segundo nivel, como se muestra en el diagrama de la figura 2-20(c). Por último, la jerarquía creció a cinco niveles.

Para 1890, las tres partes principales del sistema telefónico ya estaban en su lugar: las oficinas de conmutación, los cables entre los clientes y las oficinas de conmutación (a estas alturas cables de par trenzado balanceados y aislados, en lugar de cables abiertos con retorno a tierra) y las conexiones de larga distancia entre las oficinas de conmutación. Aunque desde entonces se han realizado mejoras en las tres áreas, el modelo básico del Bell System ha permanecido intacto en lo esencial por más de 100 años. Para una historia técnica corta del sistema telefónico vea (Hawley, 1991).

Previo a la división de AT&T en 1984, el sistema telefónico fue organizado como una jerarquía de múltiples niveles, con alta redundancia. A pesar de su simplicidad, la siguiente descripción da una idea de la situación. Cada teléfono tiene dos alambres de cobre que van directamente a la **oficina central local** de la compañía telefónica. Por lo general, la distancia va de 1 a 10 km, y en las ciudades es más corta que en las áreas rurales. Tan sólo en Estados Unidos existen cerca de 22,000 oficinas centrales. En el ámbito de las comunicaciones, las conexiones de dos alambres entre el teléfono de cada suscriptor y la oficina central se conocen como **circuito local**. Si los circuitos locales de todo el mundo se extendieran de extremo a extremo, llegarían a la Luna y regresarían a la Tierra 1000 veces.

En cierto momento, el 80% del valor del capital de AT&T fue el cobre en los circuitos locales. En efecto, AT&T era entonces la más grande mina de cobre del mundo. Por fortuna, este hecho no era muy conocido en la comunidad inversionista. De haberse sabido, algún pirata corporativo podría haber comprado la AT&T, cancelado todo el servicio telefónico en Estados Unidos, extraído todos los cables y vendido el cableado a algún refinador de cobre para obtener una ganancia rápida.

Si un suscriptor conectado a una oficina central determinada llama a otro suscriptor conectado a la misma oficina central, el mecanismo de conmutación dentro de la oficina establece una conexión eléctrica directa entre los dos circuitos locales. Esta conexión permanece intacta mientras dura la llamada.

Si el teléfono al que se llama está conectado a otra oficina central, se tiene que usar un procedimiento diferente. Cada oficina central tiene varias líneas salientes a uno o más centros de conmutación cercanos, llamados **oficinas interurbanas** (o, si están dentro de la misma área local, **oficinas en tandem**). Estas líneas se llaman **troncales de conexión interurbanas**. Si sucede que tanto la oficina central de quien llama como la de quien es llamado tienen una troncal de conexión a la misma oficina interurbana (algo muy probable si no están muy alejadas), la conexión se puede establecer dentro de la oficina interurbana. En la figura 2-20(c) se muestra una red telefónica que consiste únicamente en teléfonos (los puntos pequeños), oficinas centrales (los puntos grandes) y oficinas interurbanas (los cuadrados).

Si el que llama y el que es llamado no tienen una oficina interurbana en común, la trayectoria se deberá establecer en un nivel más alto de la jerarquía. Hay oficinas primarias, seccionales y regionales que forman una red que conecta a las oficinas interurbanas. Las centrales interurbanas, primarias, seccionales y regionales se comunican entre sí mediante **troncales interurbanas** de gran ancho de banda. La cantidad de tipos diferentes de centros de conmutación y su topología varían de país a país dependiendo de su densidad telefónica (por ejemplo, ¿pueden dos oficinas

seccionales tener una conexión directa o deben pasar por una oficina regional?). La figura 2-21 muestra cómo se podría enrutar una conexión de media distancia.

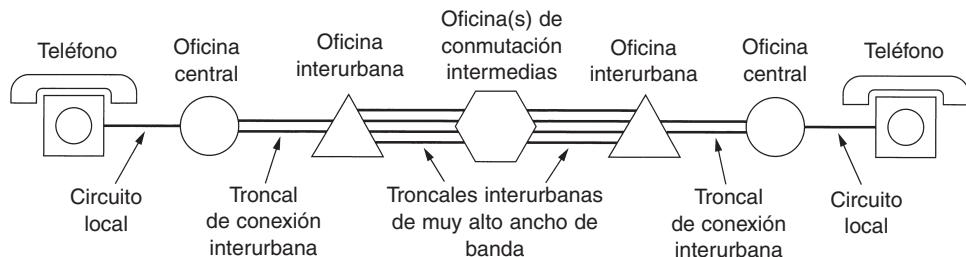


Figura 2-21. Ruta típica de un circuito para una llamada de media distancia.

Para telecomunicaciones se usan diversos medios de transmisión. En nuestros días, los circuitos locales consisten en pares trenzados, aunque en los primeros días de la telefonía eran comunes los cables no aislados espaciados a 25 cm en los postes telefónicos. Entre las oficinas de conmutación se usan ampliamente cables coaxiales, microondas y, en especial, fibra óptica.

En el pasado, la transmisión en todo el sistema telefónico era analógica, con la señal de voz real transmitida como un voltaje eléctrico entre la fuente y el destino. Con el advenimiento de la fibra óptica, la electrónica digital y las computadoras, actualmente todas las troncales y los conmutadores son digitales, y el circuito local queda como el único elemento de tecnología analógica del sistema. Existe preferencia por la transmisión digital porque en ésta no es necesario reproducir exactamente una forma de onda analógica después de que ha pasado por muchos amplificadores en una llamada larga. Es suficiente con distinguir correctamente un 0 de un 1. Esta propiedad da más confiabilidad a la transmisión digital que a la analógica. Su mantenimiento también es más económico y sencillo.

En síntesis, el sistema telefónico consiste en tres componentes principales:

1. Circuitos locales (cables de par trenzado que van hacia las casas y las empresas).
2. Troncales (fibra óptica digital que conecta a las oficinas de conmutación).
3. Oficinas de conmutación (donde las llamadas pasan de una troncal a otra).

Después de una breve digresión sobre la política de los teléfonos, regresaremos a cada uno de estos tres componentes en detalle. Los circuitos locales dan acceso a todo mundo al sistema completo, debido a lo cual son cruciales. Por desgracia, también son la parte más débil del sistema. Para las troncales de largo alcance, la principal consideración es cómo reunir múltiples llamadas y enviarlas juntas por la misma fibra. Este tema se llama multiplexión, y estudiaremos tres formas diferentes de hacerlo. Por último, existen dos formas fundamentalmente distintas de efectuar la conmutación, así que veremos ambas.

2.5.2 La política de los teléfonos

Durante las décadas anteriores a 1984, el Bell System proporcionó tanto el servicio local como el de larga distancia en casi todo Estados Unidos. En la década de 1970, el gobierno estadounidense se convenció de que éste era un monopolio ilegal y entabló un juicio para dividirlo. El gobierno ganó, y el 1o. de enero de 1984 la AT&T se dividió en AT&T Long Lines, 23 **BOCs (Compañías Operativas de Bell)** y algunas otras partes pequeñas. Las 23 BOCs se agruparon en siete BOCs regionales (RBOCs) para hacerlas económicamente viables. La naturaleza entera de la telecomunicación en Estados Unidos se cambió de la noche a la mañana por orden judicial (*no* por una ley del Congreso).

Los detalles exactos del desmantelamiento se describieron en el llamado **MFJ (Juicio Final Modificado)**, un claro contrasentido (si el juicio se pudo modificar, obviamente no era final). Este suceso condujo a un aumento en la competencia, mejor servicio y menores precios en larga distancia para los consumidores y las empresas. No obstante, los precios del servicio local se elevaron cuando los subsidios a las llamadas de larga distancia se eliminaron y el servicio local tuvo que ser autosuficiente. Muchos otros países consideran ahora la introducción de la competencia por caminos similares.

Para dejar en claro quiénes podrían actuar y cómo, Estados Unidos se dividió en más de 160 **LATAs (Áreas de Acceso y Transporte Local)**. En forma muy aproximada, una LATA es casi tan grande como el área cubierta por un código de área. Dentro de una LATA normalmente había una **LEC (Portadora de Intercambio Local)** que tenía un monopolio sobre el servicio tradicional de telefonía dentro de la LATA. Las LECs más importantes eran las BOCs, aunque algunas LATAs contenían una o más de las 1500 compañías telefónicas independientes que operaban como LECs.

Un tipo de compañía diferente maneja todo el tráfico interLATA: una **IXC (Portadora Entre Centrales)**. Originalmente, AT&T Long Lines era la única IXC seria, pero ahora WorldCom y Sprint son competidores bien establecidos en el negocio de las IXCs. Una de las consideraciones durante la división fue asegurar que todas las IXCs serían tratadas con igualdad en términos de calidad de líneas, tarifas y cantidad de dígitos que tendrían que marcar sus clientes para usarlos. La forma como esto se resolvió se ilustra en la figura 2-22. Aquí vemos tres LATAs de ejemplo, cada una con varias oficinas centrales. Las LATAs 2 y 3 tienen también una pequeña jerarquía con oficinas en tandem (oficinas interurbanas intraLATA).

Cualquier IXC que desee manejar llamadas que se originen en una LATA puede construir allí una oficina de comutación llamada **POP (Punto de Presencia)**. La LEC es necesaria para conectar cada IXC a cada oficina central, ya sea en forma directa, como en las LATAs 1 y 3, o indirecta, como en la LATA 2. Además, los términos de la conexión, tanto técnicos como financieros, deben ser idénticos para todas las IXCs. De esta forma, un suscriptor en, digamos, la LATA 1 puede elegir cuál IXC usar para llamar a los suscriptores en la LATA 3.

Como parte del MFJ, se prohibió a las IXCs ofrecer servicio telefónico local y a las LECs ofrecer servicio telefónico interLATA, aunque ambas eran libres de participar en otros negocios, como la operación de restaurantes de pollos fritos. En 1984, éste era un dictamen bastante claro. Desgraciadamente, la tecnología tiene la mala costumbre de hacer obsoletas las leyes. Ni la televisión por cable ni los teléfonos celulares estaban considerados en el acuerdo. Conforme la te-

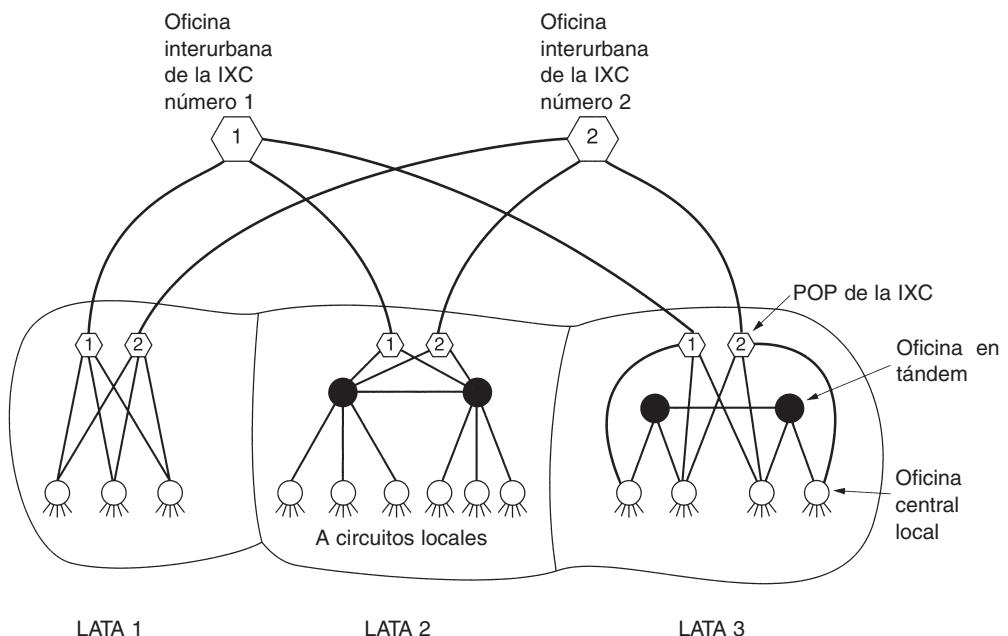


Figura 2-22. Relación entre LATAs, LECs e IXCs. Todos los círculos son oficinas de conmutación LEC. Cada hexágono pertenece a la IXC cuyo número contiene.

levisión por cable pasó de ser unidireccional a bidireccional, y la popularidad de los teléfonos celulares subió como la espuma, tanto las LECs como las IXC comenzaron a comprar o a fusionarse con los operadores de cable y celulares.

Para 1995, el Congreso vio que tratar de mantener la distinción entre las diversas clases de compañías ya no era sostenible y esbozó una propuesta de ley para permitir a las compañías de televisión por cable, a las compañías telefónicas locales, a los operadores de larga distancia y a los operadores de teléfonos celulares participar en los negocios de unos y otros. La intención era que así cualquier compañía podría ofrecer a sus clientes un solo paquete integrado que contuviera televisión por cable, teléfono y servicios de información, y que las diferentes compañías compitieran en servicio y precio. La propuesta se convirtió en ley en febrero de 1996. Como resultado, algunas BOCs se convirtieron en IXC y algunas otras compañías, como los operadores de televisión por cable, empezaron a competir con las LECs por el servicio telefónico local.

Un aspecto interesante de la ley de 1996 fue la obligación para las LECs de implementar portabilidad para los números locales. Esto quiere decir que un cliente puede cambiar de compañía telefónica local sin necesidad de obtener un nuevo número telefónico. Esta cláusula elimina un enorme obstáculo para los usuarios y los anima a cambiar de LEC, con lo cual se incrementa la competencia. En consecuencia, el panorama de las telecomunicaciones en Estados Unidos está atravesando una reestructuración radical. De nueva cuenta, muchos otros países están siguiendo esta línea. Con frecuencia, otros países esperan para ver cómo funciona esta clase de experimentos en Estados Unidos. Si da resultado, adoptan el esquema; si falla, buscan otras alternativas.

2.5.3 El circuito local: módems, ADSL e inalámbrico

Es hora de iniciar el estudio detallado del funcionamiento del sistema telefónico. Las principales partes del sistema se ilustran en la figura 2-23. Aquí vemos los circuitos locales, las troncales y las oficinas interurbanas y oficinas centrales, las cuales tienen equipo que conmuta las llamadas. Una oficina central tiene hasta 10,000 circuitos locales (en Estados Unidos y otros países grandes). De hecho, hasta hace poco tiempo el código de área + caracteres de sustitución indicaban la oficina central, de tal manera que (212) 601-xxxx se refería a una oficina central específica con 10,000 suscriptores, numerados de 0000 a 9999. Con el surgimiento de la competencia por el servicio local, este sistema dejó de ser funcional porque diversas compañías querían apoderarse del código de oficina central. Asimismo, el número de códigos prácticamente se había consumido, por lo que hubo necesidad de introducir esquemas de correspondencia complejos.

Empecemos con el tema que la mayoría de la gente conoce: el circuito local de dos alambres que parte de la oficina central de una compañía telefónica hacia hogares y pequeñas empresas. El circuito local se conoce también como de “última milla” (la conexión hacia el cliente), aunque la longitud puede ser de varias millas. Durante más de 100 años ha utilizado señalización analógica y es probable que continúe haciéndolo durante algún tiempo, debido al costo elevado de la conversión a digital. No obstante, el cambio se está dando incluso en este último bastión de la transmisión analógica. En esta sección estudiaremos el circuito local tradicional y los avances que están teniendo lugar, con especial atención en la comunicación de datos desde computadoras caseras.

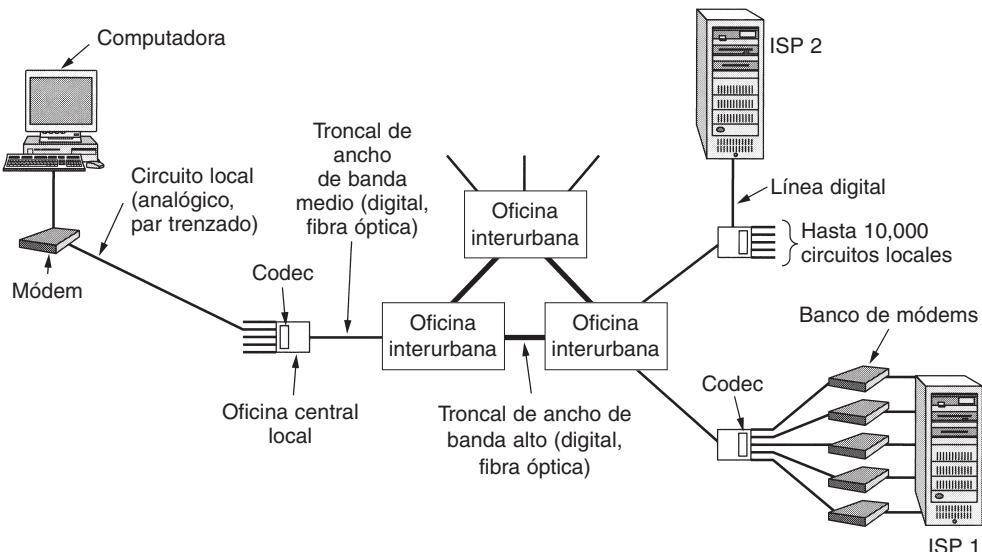


Figura 2-23. Uso de transmisión analógica y digital para una llamada de computadora a computadora. Los módems y los codecs realizan la conversión.

Cuando una computadora desea enviar datos digitales sobre una línea analógica de acceso telefónico, es necesario convertir primero los datos a formato analógico para transmitirlos sobre el

circuito local. Un dispositivo conocido como módem realiza esta conversión, tema que estudiaremos en breve. Los datos se convierten a formato digital en la oficina central de la compañía telefónica para transmitirlos sobre las troncales que abarcan largas distancias.

Si en el otro extremo hay una computadora con un módem, es necesario realizar la conversión inversa —digital a analógico— para recorrer el circuito local en el destino. Esta disposición se muestra en la figura 2-23 para el ISP 1 (proveedor de servicios de Internet), que tiene un banco de módems, cada uno conectado a un circuito local diferente. Este ISP puede manejar tantas conexiones como módems tenga (suponiendo que su servidor o sus servidores tengan suficiente potencia de cómputo). Esta disposición fue la común hasta que aparecieron los módems de 56 kbps, por razones que veremos más adelante.

La señalización analógica consiste en la variación del voltaje con el tiempo para representar un flujo de información. Si los medios de transmisión fueran perfectos, el receptor recibiría exactamente la misma señal enviada por el transmisor. Por desgracia, los medios no son perfectos, por lo cual la señal recibida no es la misma que la transmitida. Si los datos son digitales, esta diferencia puede conducir a errores.

Las líneas de transmisión tienen tres problemas principales: atenuación, distorsión por retardo y ruido. La **atenuación** es la pérdida de energía conforme la señal se propaga hacia su destino. La pérdida se expresa en decibeles por kilómetro. La cantidad de energía perdida depende de la frecuencia. Para ver el efecto de esta dependencia de la frecuencia, imagine una señal no como una simple forma de onda, sino como una serie de componentes de Fourier. Cada componente es atenuado en diferente medida, lo que da por resultado un espectro de Fourier distinto en el receptor.

Por si esto no fuera poco, los diferentes componentes de Fourier se propagan a diferente velocidad por el cable. Esta diferencia de velocidad ocasiona una **distorsión** de la señal que se recibe en el otro extremo.

Otro problema es el **ruido**, que es energía no deseada de fuentes distintas al transmisor. El movimiento al azar de los electrones en un cable causa el ruido térmico y es inevitable. La diafonía se debe al acoplamiento inductivo entre dos cables que están cerca uno de otro. A veces, al hablar por teléfono se escucha otra conversación en el fondo. Ésa es la diafonía. Finalmente, hay ruido de impulso, causado por picos en la línea de suministro de energía o por otros fenómenos. En el caso de datos digitales, el ruido de impulso puede eliminar uno o más bits.

Módems

Debido a los problemas antes mencionados, en especial al hecho de que tanto la atenuación como la velocidad de propagación dependen de la frecuencia, es indeseable tener un rango amplio de frecuencias en la señal. Desgraciadamente, las ondas cuadradas, como las de los datos digitales, tienen un espectro amplio y por ello están sujetas a una fuerte atenuación y a distorsión por retardo. Estos efectos hacen que la señalización de banda base (CC, corriente continua) sea inadecuada, excepto a velocidades bajas y distancias cortas.

La señalización de CA (corriente alterna) se utiliza para superar los problemas asociados a la señalización de CC, en especial en las líneas telefónicas. Se introduce un tono continuo en el rango de 1000 a 2000 Hz, llamado **portadora de onda senoidal**, cuya amplitud, frecuencia o fase se

pueden modular para transmitir la información. En la **modulación de amplitud** se usan dos niveles diferentes de amplitud para representar 0 y 1, respectivamente. En la **modulación de frecuencia**, conocida también como **modulación por desplazamiento de frecuencia**, se usan dos (o más) tonos diferentes. En la forma más simple de la **modulación de fase** la onda portadora se desplaza de modo sistemático 0 o 180 grados a intervalos espaciados de manera uniforme. Un mejor esquema es utilizar desplazamientos de 45, 135, 225 o 315 grados para transmitir 2 bits de información por intervalo. Asimismo, al requerir siempre un desplazamiento de fase al final de cada intervalo se facilita que el receptor reconozca los límites de los intervalos.

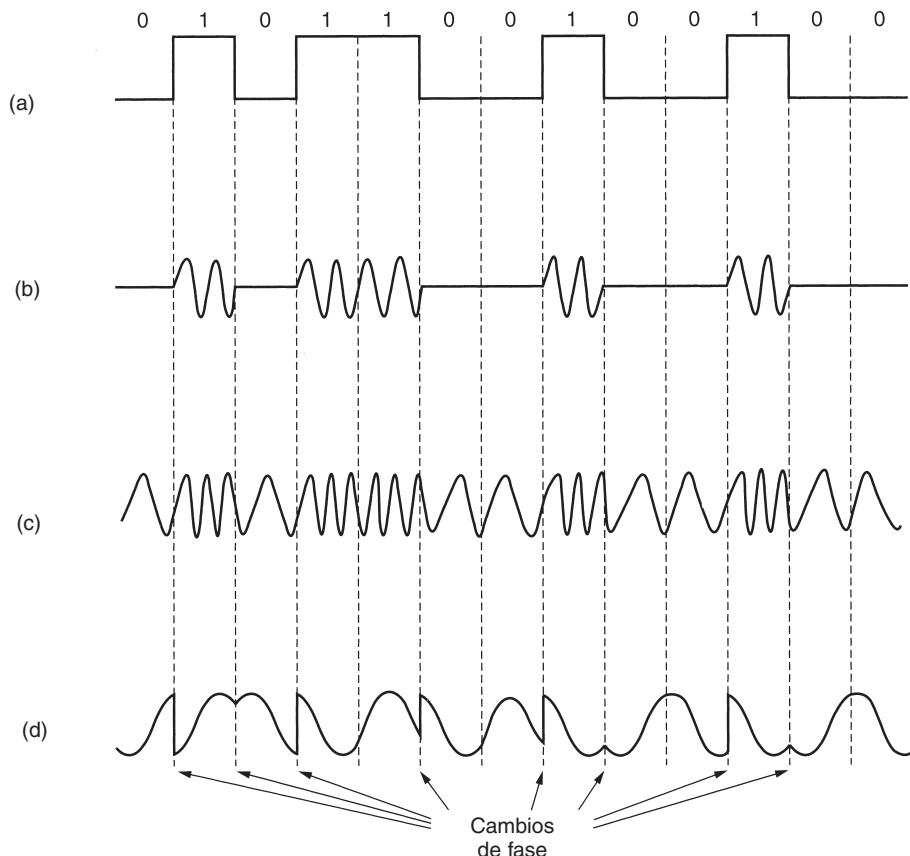


Figura 2-24. (a) Señal binaria. (b) Modulación de amplitud. (c) Modulación de frecuencia. (d) Modulación de fase.

La figura 2-24 ilustra los tres tipos de modulación. En la figura 2-24(a) una de las amplitudes es distinta de cero y la otra es cero. En la figura 2-24(b) se utilizan dos frecuencias. En la figura 2-24(c) está presente o ausente un desplazamiento de fase en cada límite de bit. Un **módem** (por modulador-demodulador) es un dispositivo que acepta un flujo de bits en serie como entrada y que

produce una portadora modulada mediante uno (o más) de estos métodos (o viceversa). El módem se conecta entre la computadora (digital) y el sistema telefónico (analógico).

Para alcanzar velocidades cada vez más altas, no basta sólo incrementar la velocidad de muestreo. El teorema de Nyquist dice que aun con una línea perfecta de 3000 Hz (cosa que decididamente no son las líneas de acceso telefónico), no tiene sentido muestrear más allá de 6000 Hz. En la práctica, la mayoría de los módems muestrea 2400 veces por segundo y el objetivo es conseguir más bits por muestra.

El número de muestras por segundo se mide en **baudios**. Un **símbolo** se envía durante cada baudio. De esta manera, una línea de n baudios transmite n símbolos por segundo. Por ejemplo, una línea de 2400 baudios envía un símbolo más o menos cada 416.667 µseg. Si el símbolo consta de 0 voltios para un 0 lógico y de 1 voltio para un 1 lógico, la tasa de bits es de 2400 bps. Sin embargo, si se utilizan los voltajes 0, 1, 2 y 3, cada símbolo consta de 2 bits, por lo que una línea de 2400 baudios pueden transmitir 2400 símbolos por segundo a una tasa de datos de 4800 bps. De manera similar, con cuatro posibles desplazamientos de fase también hay 2 bits por símbolo, con lo cual la tasa de bits es otra vez el doble que la tasa de baudios. La última técnica se utiliza ampliamente y se denomina **QPSK (Codificación por Desplazamiento de Fase en Cuadratura)**.

Es común la confusión de los conceptos ancho de banda, baudio, símbolo y tasa de bits, por lo que los definiremos a continuación. El ancho de banda de un medio es el rango de frecuencias que atraviesa al medio con atenuación mínima. Es una propiedad física del medio (por lo general, de 0 a alguna frecuencia máxima) y se mide en hertzios (Hz). La tasa de baudios es la cantidad de muestras por segundo que se realizan. Cada muestra envía una porción de información, es decir, un símbolo. Por lo tanto, la tasa de baudios y la tasa de símbolos significan lo mismo. La técnica de modulación (por ejemplo, QPSK) determina la cantidad de bits por símbolo. La tasa de bits es la cantidad de información que se envía por el canal y es igual a la cantidad de símbolos por segundo por la cantidad de bits por símbolo.

Todos los módems avanzados utilizan una combinación de técnicas de modulación con el propósito de transmitir muchos bits por baudio. Con frecuencia se combinan múltiples amplitudes y múltiples desplazamientos de fase para transmitir muchos bits por símbolo. En la figura 2-25(a) vemos puntos con amplitud constante a los 45, 135, 225 y 315 grados (distancia desde el origen). La fase de un punto la indica el ángulo que se forma con el eje de las X al trazar una línea desde el punto hacia el origen. La figura 2-25(a) tiene cuatro combinaciones válidas y se puede utilizar para transmitir 2 bits por símbolo. Es QPSK.

En la figura 2-25(b) se muestra un esquema de modulación distinto, en el cual se utilizan cuatro amplitudes y cuatro fases, que permiten un total de 16 combinaciones diferentes. Este esquema de modulación se puede utilizar para transmitir 4 bits por símbolo. Se conoce como **QAM-16 (Modulación de Amplitud en Cuadratura)**. En algunas ocasiones también se utiliza el término **16-QAM**. Por ejemplo, QAM-16 se puede utilizar para transmitir 9600 bps sobre una línea de 2400 baudios.

En la figura 2-25(c) se presenta otro esquema de modulación que incluye amplitud y fase. En éste se pueden conseguir 64 combinaciones diferentes, por lo cual es posible transmitir 6 bits por símbolo. Se conoce como **QAM-64**. También se utilizan QAMs de orden más alto.

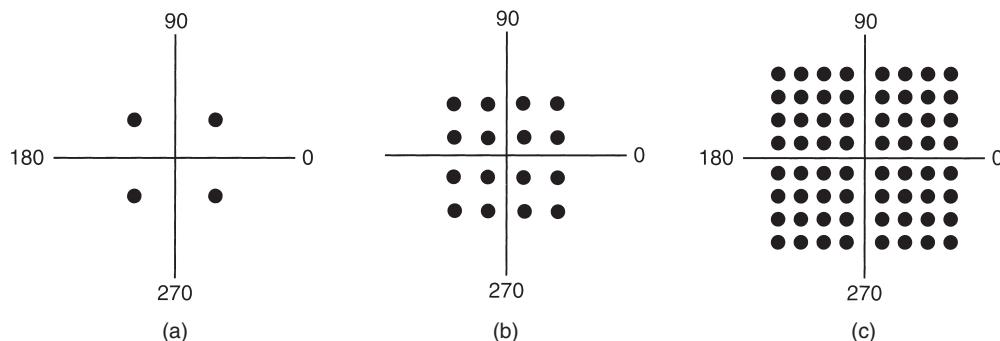


Figura 2-25. (a) QPSK. (b) QAM-16. (c) QAM-64.

A los diagramas como los de la figura 2-25, que muestran las combinaciones permitidas de amplitud y fase, se les llama **diagramas de constelación**. Cada estándar de módem de alta velocidad tiene su propio diagrama de constelación y se puede comunicar solamente con otros módems que utilicen el mismo modelo (aunque la mayoría de los módems puede emular a todos los modelos más lentos).

Cuando hay muchos puntos en un diagrama de constelación, incluso la cantidad mínima de ruido en la amplitud o fase detectada puede dar como resultado un error y, potencialmente, muchos bits malos. Con el propósito de reducir la posibilidad de error, los estándares para los módems de velocidades más altas realizan corrección de errores mediante la incorporación de bits adicionales en cada muestra. Los esquemas se conocen como **TCM (Modulación por Codificación de Malla)**. Así, por ejemplo, el estándar V.32 de módem utiliza 32 puntos de constelación para transmitir 4 bits de datos y 1 bit de paridad por símbolo a 2400 baudios, para alcanzar 9600 bps con corrección de errores. Su diagrama de constelación se muestra en la figura 2-26(a). La decisión de “girar” 45 grados alrededor del origen se tomó por razones de ingeniería; las constelaciones giradas y sin girar tienen la misma capacidad de información.

El siguiente escalón después de 9600 bps es 14,400 bps. Se conoce como **V.32 bis**. Esta velocidad se alcanza al transmitir 6 bits de datos y 1 bit de paridad por muestra a 2400 baudios. Su diagrama de constelación tiene 128 puntos cuando se utiliza QAM-128, y se muestra en la figura 2-26(b). Los fax-módems transmiten a esta velocidad las páginas que han sido digitalizadas como mapas de bits. QAM-256 no se utiliza en ningún módem telefónico estándar, pero sí en redes de cable, como veremos más adelante.

Enseguida del módem telefónico V.32 se encuentra el **V.34**, el cual corre a 28,800 bps, 2400 baudios y 12 bits de datos por símbolo. El último módem de esta serie es el **V.34 bis**, el cual transfiere 14 bits de datos por símbolo a 2400 baudios para alcanzar una velocidad de 33,600 bps.

Para incrementar aún más la tasa de datos efectiva, muchos módems comprimen los datos antes de enviarlos, y alcanzan tasas de datos efectivas mayores a 33,600 bps. Por otra parte, casi todos los módems prueban la línea antes de empezar a transmitir datos del usuario, y si encuentran una falta de calidad, reducen la velocidad a una menor a la máxima que tiene asignada. Por lo tanto, la velocidad *efectiva* del módem que percibe el usuario puede ser menor, igual o mayor a la que oficialmente tiene asignada.

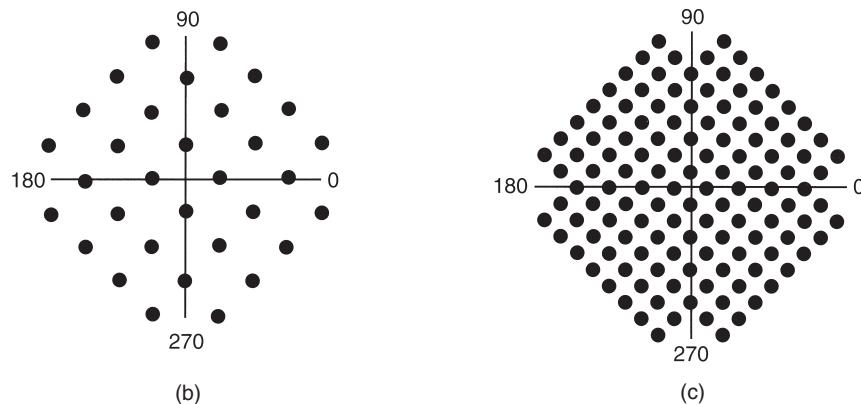


Figura 2-26. (a) V.32 para 9600 bps. (b) V32 bis para 14,400 bps.

Todos los módems modernos transmiten tráfico en ambas direcciones al mismo tiempo (mediante el uso de frecuencias distintas para las diferentes direcciones). La conexión que permite el flujo de tráfico en ambas direcciones de manera simultánea se conoce como **dúplex total**. Una carretera de dos carriles es dúplex total. La conexión que permite el tráfico en ambas direcciones, pero sólo en un sentido a la vez, se denomina **semidúplex**. Una vía de ferrocarril es semidúplex. La conexión que permite el tráfico en una sola dirección se conoce como **símplex**. Una calle de un solo sentido es simplex. Otro ejemplo de una conexión simplex lo constituye una fibra óptica con un láser en un extremo y un detector de luz en el otro.

La razón por la cual los módems estándar llegan hasta 33,600 es que el límite de Shannon para el sistema telefónico es de aproximadamente 35 kbps, así que velocidades mayores a este límite violarían las leyes de la física (departamento de termodinámica). Para saber si los módems de 56 kbps son posibles desde un punto de vista teórico, continúe leyendo.

¿Pero a qué se debe que el límite teórico sea de 35 kbps? La respuesta está en la longitud promedio de los circuitos locales y en la calidad de estas líneas. La longitud promedio de los circuitos locales determina los 35 kbps. En la figura 2-23, una llamada que se origina en la computadora de la izquierda y que termina en el ISP 1 recorre dos circuitos locales como señal analógica, una vez en el punto de origen y otra en el punto de destino. En cada uno de estos circuitos se agrega ruido a la señal. Si pudiéramos prescindir de uno de estos circuitos locales, podría duplicarse la tasa máxima.

El ISP 2 hace precisamente esto. Cuenta con una alimentación digital pura proveniente de la oficina central más cercana. La señal digital que se utiliza en las troncales es alimentada directamente al ISP 2, con lo cual se elimina la necesidad de codecs, módems y transmisión analógica en su extremo. Así, cuando un extremo de la conexión es puramente digital, como ocurre con la mayoría de los ISPs actuales, la tasa máxima de datos puede ser de hasta 70 kbps. El máximo entre dos usuarios caseros con líneas analógicas es de 33.6 kbps.

La razón por la cual se utilizan los módems de 56 kbps se relaciona con el teorema de Nyquist. El canal telefónico tiene un ancho de alrededor de 4000 Hz (incluyendo las bandas de protección o guarda). De esta forma, la cantidad máxima de muestras independientes por segundo

es de 8000. La cantidad de bits por muestra en Estados Unidos es de 8, uno de los cuales se utiliza con propósitos de control, con lo cual es posible transmitir 56,000 bits por segundo de datos de usuario. En Europa los 8 bits están disponibles para los usuarios, lo cual permitiría utilizar módems de 64,000 bits por segundo, pero se eligió la cifra de 56,000 para apegarse a un estándar internacional.

Este estándar para módems se denomina **V.90**. Hace posible un canal ascendente o de subida (del usuario al ISP) de 33.6 kbps y un canal descendente o de bajada (del ISP al usuario) de 56 kbps, debido a que por lo regular hay más transporte de datos del ISP al usuario que al revés (por ejemplo, la solicitud de una página Web requiere sólo algunos bytes, pero el envío de la misma puede constituir varios megabytes). En teoría, podría ser factible un canal ascendente de más de 33.6 kbps de ancho, pero como muchos circuitos locales son demasiado ruidosos incluso para 33.6 kbps, se decidió asignar más ancho de banda al canal descendente para incrementar las posibilidades de que funcione en realidad a 56 kbps.

El paso siguiente al V.90 es el **V.92**. Estos módems tienen capacidad de 48 kbps en el canal ascendente si la línea puede manejarlo. También determinan la velocidad apropiada que se utilizará en alrededor de la mitad de los 30 segundos en que lo hacen los módems más antiguos. Por último, permiten que una llamada telefónica entrante interrumpa una sesión en Internet, siempre y cuando la línea tenga el servicio de llamada en espera.

Líneas digitales de suscriptor

Cuando la industria telefónica alcanzó por fin los 56 kbps, se congratuló a sí misma por haber realizado un buen logro. Mientras tanto, la industria de TV por cable ofrecía velocidades de hasta 10 Mbps en cables compartidos, y las compañías de satélite planeaban ofrecer más allá de 50 Mbps. Conforme el acceso a Internet se tornaba una parte importante de su negocio, las compañías telefónicas (LECs) se dieron cuenta de que necesitaban un producto más competitivo. En respuesta comenzaron a ofrecer nuevos servicios digitales sobre el circuito local. Los servicios con mayor ancho de banda que el servicio telefónico común se denominan en ocasiones como de **banda ancha**, aunque en realidad el término es más un concepto de marketing que un concepto técnico específico.

En un principio había muchas ofertas que se traslapaban, todas bajo el nombre general de **xDSL (Línea Digital de Suscriptor)**, por diversos *x*. Más adelante analizaremos estos servicios, pero primero nos enfocaremos en el que tal vez se convierta en el más popular: **ADSL (DSL Asimétrica)**. Debido a que ADSL aún está en desarrollo y no todos los estándares están totalmente establecidos, algunos de los detalles que mencionaremos podrían cambiar con el paso del tiempo, aunque el panorama general debe permanecer igual. Para obtener mayor información sobre ADSL, vea (Summers, 1999, y Vetter y cols., 2000).

La razón por la cual los módems son tan lentos es que los teléfonos fueron creados para transportar la voz humana y todo el sistema se ha optimizado cuidadosamente con este propósito. Los datos siempre han sido un aspecto secundario. En el lugar donde cada circuito local termina en la oficina central, el cable pasa a través de un filtro que atenúa todas las frecuencias abajo de 300 Hz y arriba de 3400 Hz. El corte no es abrupto —300 Hz y 3400 Hz son los puntos a 3 dB—, de tal

manera que el ancho de banda se indica como 4000 Hz aun cuando la distancia entre los puntos a 3 dB es de 3100 Hz. Por lo tanto, los datos también se restringen a esta banda estrecha.

El truco para que xDSL funcione es que cuando un cliente se suscribe al servicio, la línea de entrada se conecta a un tipo distinto de conmutador, que no cuenta con el filtro, gracias a lo cual toda la capacidad del circuito local queda disponible. En esta situación, el ancho de banda artificial de 3100 Hz generado por el filtro ya no es el factor limitante, sino el medio físico del circuito local.

Por desgracia, la capacidad del circuito local depende de varios factores, entre ellos su longitud, espesor y calidad general. En la figura 2-27 se muestra una gráfica del ancho de banda potencial como una función de la distancia. En esta figura se da por sentado que todos los demás factores son óptimos (cables nuevos, haces moderados de cables, etcétera).

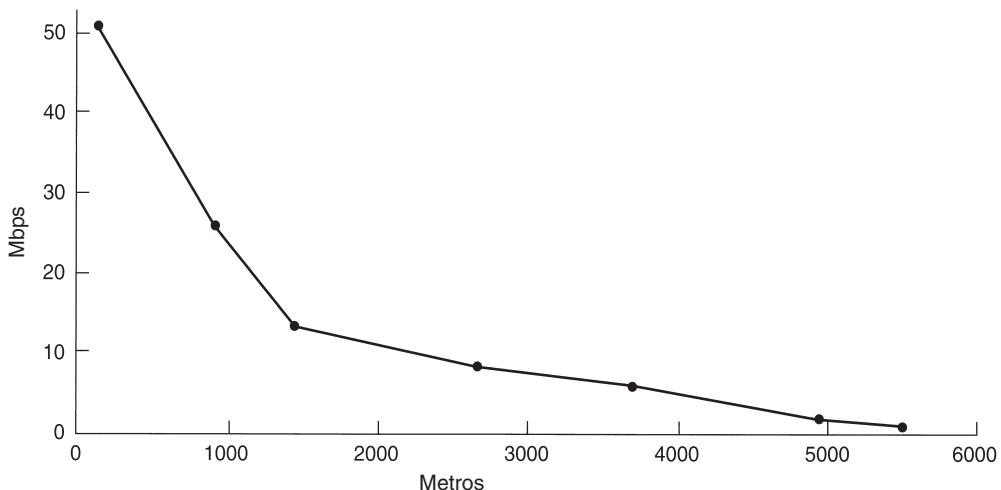


Figura 2-27. Ancho de banda contra distancia sobre la categoría 3 UTP para DSL.

La implicación de esta figura crea un problema para las compañías telefónicas. Cuando eligen la velocidad que ofrecerán, al mismo tiempo eligen un radio a partir de sus oficinas centrales más allá del cual no pueden proporcionar el servicio. Esto quiere decir que cuando un cliente distante intenta adquirir el servicio, podría obtener la siguiente respuesta: "Muchas gracias por su interés, pero no podemos darle el servicio porque usted vive 100 metros más lejos de la oficina central más cercana. ¿Podría mudarse?" Entre más baja sea la velocidad elegida, más amplio será el radio y podrán abarcarse más clientes. Pero entre más baja sea la velocidad, el servicio será menos atractivo y será menos la gente dispuesta a pagar por él. Aquí es donde se encuentran los negocios y la tecnología. (Una posible solución es construir minioficinas centrales en los vecindarios, pero es una alternativa costosa.)

Todos los servicios xDSL se diseñaron para que cumplieran algunos objetivos. Primero, los servicios deben funcionar sobre los circuitos locales existentes de par trenzado, categoría 3. Segundo, no deben afectar las máquinas de fax ni los teléfonos existentes de los clientes. Tercero, deben superar por mucho los 56 kbps. Cuarto, siempre deben funcionar, con sólo una tarifa mensual, no por minuto.

AT&T hizo la oferta inicial de ADSL, el cual funcionaba dividiendo el espectro disponible en el circuito local, de alrededor de 1.1 MHz, en tres bandas de frecuencia: **POTS (Servicio Telefónico Convencional)**, canal ascendente (del usuario a la oficina central) y canal descendente (de la oficina central al usuario). La técnica en la cual se cuenta con múltiples bandas de frecuencia se conoce como multiplexión por división de frecuencia; en una sección posterior la analizaremos con detalle. Las ofertas subsecuentes de otros proveedores han tomado un enfoque distinto, y al parecer el siguiente es el probable ganador, así que lo describiremos a continuación.

El enfoque alternativo, llamado **DMT (MultiTono Discreto)**, se ilustra en la figura 2-28. En efecto, lo que hace es dividir el espectro disponible de 1.1 MHz en el circuito local en 256 canales independientes de 4 kHz cada uno. El canal 0 se utiliza para el POTS. Los canales 1-5 no se emplean, con el propósito de evitar que las señales de voz y de datos interfieran entre sí. De los 250 canales restantes, uno se utiliza para control del flujo ascendente y uno para control del flujo descendente. El resto está disponible para datos del usuario.

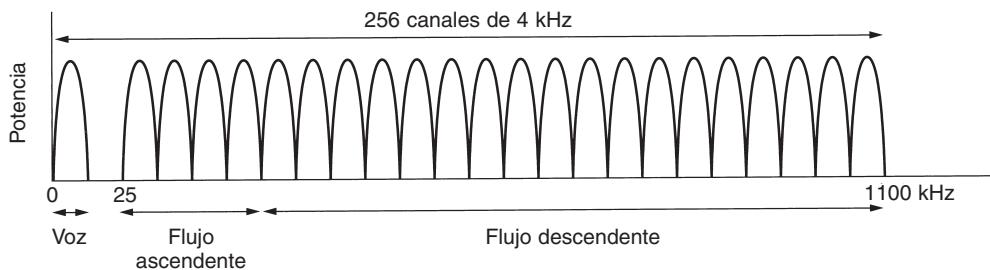


Figura 2-28. Operación de ADSL con modulación multitono discreta.

En principio, cada uno de los canales restantes se puede utilizar para un flujo de datos de dúplex total, pero las armónicas, la diafonía y otros efectos mantienen a los sistemas en la práctica muy por debajo del límite teórico. Queda a cargo del proveedor determinar cuántos canales se utilizarán para el flujo ascendente y cuántos para el flujo descendente. Es técnicamente posible una combinación de 50-50 de flujo ascendente y flujo descendente, pero la mayoría de los proveedores asigna entre 80 y 90% del ancho de banda al canal descendente debido a que el grueso de los usuarios descargan más datos que los que envían. Esta situación dio lugar a la “A” (asimétrica) de ADSL. Una división común es asignar 32 canales para el flujo ascendente y el resto al flujo descendente. También es posible establecer algunos de los canales de flujo ascendente más altos como bidireccionales para el ancho de banda incrementado, aunque esta optimización requiere agregar un circuito especial para cancelar el eco.

El estándar ADSL (ANSI T1.413 y el ITU G.992.1) permite velocidades de hasta 8 Mbps para el flujo descendente y de 1 Mbps para el flujo ascendente. No obstante, pocos proveedores ofrecen esta velocidad. Por lo general, los proveedores ofrecen 512 kbps para el flujo descendente y 64 kbps para el flujo ascendente (en el servicio estándar) y 1 Mbps para el flujo descendente y 256 kbps para el flujo ascendente (en el servicio premium).

Dentro de cada canal se utiliza un esquema de modulación similar a V.34, aunque la tasa de muestreo es de 4000 baudios en vez de 2400. La calidad de la línea en cada canal se monitorea

de manera constante y la tasa de datos se ajusta cada vez que es necesario, por lo cual canales distintos podrían tener tasas de datos diferentes. Los datos actuales se envían con modulación QAM, con un máximo de 15 bits por baudio, utilizando un diagrama de constelación análogo al de la figura 2-25(b). Por ejemplo, con 224 canales descendentes y 15 bits/baudio a 4000 baudios, el ancho de banda del flujo descendente es de 13.44 Mbps. En la práctica, la relación señal a ruido nunca es suficientemente buena para alcanzar esta tasa, pero en trayectorias cortas sobre circuitos de alta calidad es posible lograr 8 Mbps, razón por la cual el estándar llega hasta este punto.

En la figura 2-29 se muestra una disposición ADSL común. En este esquema, un técnico de la compañía telefónica debe instalar un **NID** (**Dispositivo de Interfaz de Red**) en la residencia del cliente. Esta pequeña caja de plástico delimita el fin de la propiedad de la compañía telefónica y el inicio de la propiedad del cliente. Cerca del NID (o en ocasiones en combinación con él) hay un **divisor**, un filtro analógico que separa la banda de 0-4000 Hz utilizada por la voz (POTS) de los datos. La señal POTS se enruta hacia el teléfono o máquina de fax existente, y la señal de datos se enruta a un módem. El módem ADSL es en realidad un procesador de señales digitales configurado para funcionar como 250 módems QAM operando en paralelo a diferentes frecuencias. Debido a que la mayoría de los módems ADSL actuales son externos, la computadora debe estar conectada a él a una velocidad alta. Por lo general, esto se consigue al colocar una tarjeta Ethernet en la computadora y poner a funcionar una Ethernet bastante corta de dos nodos tan sólo con la computadora y el módem ADSL. En ocasiones se utiliza el puerto USB en lugar de Ethernet. Sin duda, las tarjetas internas para módem ADSL estarán disponibles a futuro.

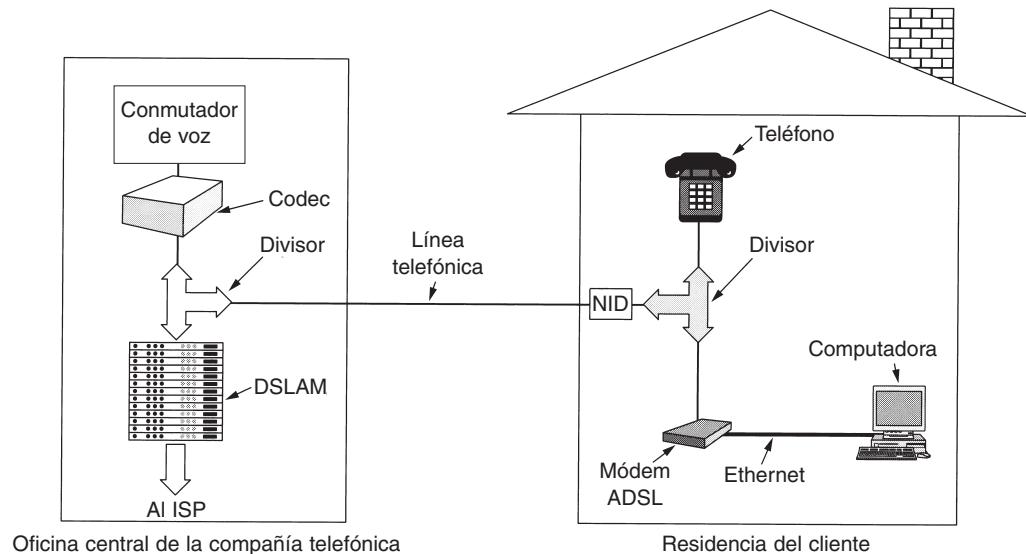


Figura 2-29. Configuración típica de un equipo ADSL.

En el otro extremo del cable, en la oficina central, se instala un divisor correspondiente. Aquí, se filtra la porción de voz de la señal y se envía al conmutador de voz normal. Las señales por arriba

de 26 kHz se enrutan hacia un nuevo tipo de dispositivo conocido como **DSLAM (Multiplexor de Acceso de Línea Digital de Suscriptor)**, el cual contiene el mismo tipo de procesador digital de señales que el módem ADSL. Una vez que la señal digital se extrae de un flujo de bits, se elaboran paquetes y se envían al ISP.

Esta completa separación entre el sistema de voz y ADSL facilita relativamente a la compañía telefónica el despliegue de ADSL. Todo lo que tiene que hacer es comprar un DSLAM y un divisor y conectar a los suscriptores ADSL al divisor. Otros servicios de ancho de banda alto (por ejemplo, ISDN) requieren cambios mucho más significativos al equipo de conmutación existente.

Una desventaja del diseño de la figura 2-29 es la presencia del NID y el divisor en la residencia del cliente. La instalación de estos componentes en la residencia del cliente sólo puede realizarla un técnico de la compañía telefónica, lo cual resulta bastante costoso. En consecuencia, también se ha estandarizado un diseño alternativo sin divisor. Informalmente se le conoce como G-lite, pero el número de estándar ITU es G.992.2. Es el mismo que el de la figura 2-29, aunque sin el divisor. La línea telefónica existente se utiliza tal como está. La única diferencia es que se tiene que colocar un microfiltro en cada conector telefónico, entre el teléfono o el módem ADSL y el cable. El microfiltro para el teléfono es un filtro pasa bajas que elimina frecuencias por arriba de 3400 Hz; el microfiltro para el módem ADSL es un filtro pasa altas que elimina las frecuencias por abajo de 26 kHz. El inconveniente es que este sistema no es tan confiable como el de divisor, por lo que G-lite sólo se puede utilizar hasta 1.5 Mbps (en comparación con los 8 Mbps para ADSL con un divisor). No obstante, G-lite aún requiere un divisor en la oficina central pero este tipo de instalación es relativamente económica y sencilla.

ADSL es tan sólo un estándar de la capa física. Lo que se ejecuta encima de él depende de la empresa portadora. Con frecuencia, ATM es la elección debido a su capacidad para manejar calidad de servicio y al hecho de que muchas compañías telefónicas ejecutan ATM en la red central.

Circuitos locales inalámbricos

Desde 1996 en Estados Unidos y un poco más tarde en otros países, existe libertad para las compañías que desean entrar a la competencia con la compañía telefónica local (antes monopolista), llamada **ILEC (LEC Obligada)**. Las candidatas más probables son las compañías telefónicas de larga distancia (IXCs). Cualquier IXC que desee entrar al negocio telefónico local en alguna ciudad debe hacer lo siguiente: primero, debe comprar o alquilar un edificio para establecer su primera oficina central en dicha ciudad. Segundo, debe equipar la oficina con conmutadores telefónicos y otros dispositivos, todos los cuales están disponibles para venta directa al público. Tercero, debe tender una conexión de fibra óptica entre la oficina central y su central interurbana más cercana para que los clientes locales tengan acceso a su red nacional. Cuarto, debe conseguir clientes, por lo general, promoviendo un mejor servicio o precios más bajos que los de la ILEC.

Aquí empieza la parte difícil. Suponga que la compañía consigue algunos clientes. ¿De qué manera la nueva compañía telefónica local, conocida como **CLEC (LEC Competitiva)**, conectará los teléfonos y computadoras de los clientes a su flamante nueva oficina central? La adquisición de los derechos de paso necesarios y el tendido de los cables o fibras son extremadamente costosos.

Muchas CLECs han encontrado una alternativa de bajo costo en lugar del tradicional circuito local con cable de par trenzado: el **WLL (Circuito Local Inalámbrico)**.

De cierta manera, un teléfono fijo que utiliza un circuito local inalámbrico se parece un poco a un teléfono móvil, pero existen tres diferencias técnicas importantes. Primera, el cliente del circuito local inalámbrico con frecuencia desea conectividad de alta velocidad a Internet, al menos similar a la de ADSL. Segunda, al nuevo cliente probablemente no le importe que un técnico de la CLEC tenga que instalar una gran antena direccional en su techo, la cual apunta a la oficina central de la CLEC. Tercera, el usuario no se mueve, con lo cual se evitan todos los problemas asociados a la movilidad y la transferencia de celdas (*cell handoff*) que estudiaremos más tarde en este capítulo. Por lo tanto, estamos ante el surgimiento de una nueva industria: la **inalámbrica fija** (teléfono local y servicio de Internet ofrecidos por CLECs sobre circuitos locales inalámbricos).

Aunque los WLLs empezaron a funcionar de manera formal en 1998, debemos remontarnos a 1969 para conocer su origen. En ese año la FCC asignó dos canales de televisión (a 6 MHz cada uno) para la televisión educativa a 2.1 GHz. En años posteriores se agregaron 31 canales más a 2.5 GHz para totalizar 198 MHz.

La televisión educativa nunca se popularizó y en 1998 la FCC decidió quitarle las frecuencias y asignarlas a la radio bidireccional. De inmediato fueron utilizadas para los circuitos locales inalámbricos. A estas frecuencias, las microondas tienen una longitud de 10-12 cm. Poseen un rango de casi 50 km y pueden penetrar la vegetación y la lluvia moderadamente bien. Los 198 MHz de nuevo espectro fueron puestos inmediatamente en uso para los circuitos locales inalámbricos en un servicio denominado **MMDS (Servicio de Distribución Multipunto y Multicanal)**. El MMDS se puede considerar como una MAN (red de área metropolitana), al igual que su similar LMDS (que se analiza más abajo).

La gran ventaja de este servicio es que la tecnología está bien desarrollada y que el equipo se consigue con facilidad. La desventaja consiste en que el ancho de banda total disponible es modesto y deben compartirlo muchos usuarios de una enorme área geográfica.

El bajo ancho de banda del MMDS despertó el interés en las ondas milimétricas como alternativa. No se asignaron frecuencias en el rango de 28-31 GHz en Estados Unidos y de 40 GHz en Europa debido a la dificultad de construir circuitos integrados de silicio que operen a esas velocidades. El problema fue resuelto con la invención de circuitos integrados de arseniuro de galio, lo que abrió las bandas milimétricas para la radiocomunicación. La FCC respondió a la demanda al asignar 1.3 GHz a un nuevo servicio de circuito local inalámbrico llamado **LMDS (Servicio Local de Distribución Multipunto)**. Esta porción de ancho de banda es la mayor que la FCC ha asignado de una sola vez para cualquier uso. En Europa se asignó una porción similar, aunque a 40 GHz.

En la figura 2-30 se muestra cómo funciona LMDS. Se puede apreciar una torre con varias antenas, cada una de las cuales apunta a una dirección distinta. Debido a que las ondas milimétricas son altamente direccionales, cada antena define un sector, independiente de los demás. A esta frecuencia, el rango es de 2-5 km, lo cual quiere decir que se necesitan muchas antenas para abarcar una ciudad.

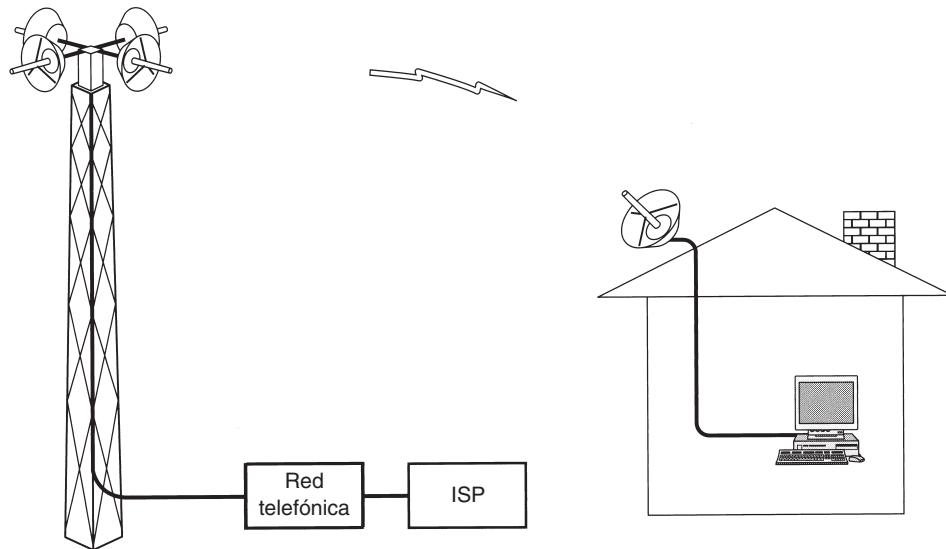


Figura 2-30. Arquitectura de un sistema LMDS.

Al igual que ADSL, LMDS asigna el ancho de banda de manera asimétrica, dando prioridad al canal descendente. Con la tecnología actual, cada sector puede contar con 36 Gbps de flujo descendente y 1 Mbsp de flujo ascendente, compartidos por todos los usuarios del sector. Si cada usuario activo descarga 3 páginas de 5 KB por minuto, el usuario ocupa un promedio de 2000 bps de espectro, lo cual permite un máximo de 18,000 usuarios activos por sector. No obstante, para mantener un retardo razonable debe haber un máximo de 9000 usuarios activos. Con cuatro sectores, como se muestra en la figura 2-30, puede soportarse una población de 36,000 usuarios activos. Suponiendo que uno de tres clientes esté en línea durante las horas de máxima actividad, una torre con cuatro antenas puede dar servicio a 100,000 usuarios dentro de un radio de 5 km de la torre. Muchas CLECs potenciales han realizado estos cálculos, y algunas de ellas han llegado a la conclusión de que con la cantidad necesaria para realizar una modesta inversión en torres de ondas milimétricas, se pueden meter al negocio de la telefonía local e Internet y ofrecer tasas de datos comparables a las de la televisión por cable, incluso a un menor precio.

Sin embargo, LMDS tiene algunos problemas. Por una parte, las ondas milimétricas se propagan en línea recta, por lo cual debe haber una línea visual despejada entre las antenas colocadas en el techo y la torre. Por otra parte, las hojas absorben bien estas ondas, por lo tanto, la torre debe tener suficiente altura para evitar los árboles en la línea visual. Lo que podría parecer una línea visual despejada en diciembre, tal vez no esté despejada en julio cuando los árboles están repletos de hojas. La lluvia también absorbe estas ondas. Hasta cierto punto, los errores producidos por la lluvia se pueden compensar con códigos de corrección de errores o incrementando la potencia cuando llueve. Con todo, es más probable que el servicio LMDS se estrene primero en climas secos, como en Arizona en vez de en Seattle.

Es poco probable que los circuitos locales inalámbricos se popularicen si no surgen estándares que animen a los fabricantes a producir equipo y que aseguren a los usuarios la oportunidad de cambiar de CLEC sin necesidad de comprar equipo nuevo. Con el propósito de proporcionar esta estandarización, el IEEE estableció el comité 802.16 para que se encargara de preparar el estándar para LMDS. El estándar 802.16 se publicó en abril de 2002. El IEEE denomina **MAN inalámbrica** al 802.16.

El estándar 802.16 del IEEE se diseñó para telefonía digital, acceso a Internet, conexión de dos LANs remotas, difusión por televisión y radio, entre otros usos. En el capítulo 4 lo veremos con más detalle.

2.5.4 Troncales y multiplexión

La economía de escala desempeña un papel importante en el sistema telefónico. Cuesta prácticamente lo mismo instalar y mantener una troncal de ancho de banda alto que una de ancho de banda bajo entre dos oficinas de commutación (es decir, el gasto principal es la excavación de zanjas y no el cable de cobre o la fibra óptica). En consecuencia, las compañías telefónicas han desarrollado esquemas complejos para multiplexar muchas conversaciones en una sola troncal física. Estos esquemas de multiplexión se pueden dividir en dos categorías principales: **FDM (Multiplexión por División de Frecuencia)** y **TDM (Multiplexión por División de Tiempo)**. En FDM el espectro de frecuencia se divide en bandas de frecuencia, y cada usuario posee exclusivamente alguna banda. En TDM los usuarios esperan su turno (en *round-robin*), y cada uno obtiene en forma periódica toda la banda durante un breve lapso de tiempo.

La radiodifusión AM ilustra ambas clases de multiplexión. El espectro asignado es de alrededor de 1 MHz, aproximadamente de 500 a 1500 kHz. A los diferentes canales lógicos (estaciones) se les asigna una frecuencia distinta, y cada uno funciona en una porción del espectro con una separación entre canales lo bastante grande para evitar la interferencia. Este sistema es un ejemplo de multiplexión por división de frecuencia. Además (en algunos países), las estaciones individuales tienen dos subcanales lógicos: música y publicidad. Éstos se alternan en la misma frecuencia, primero una ráfaga de música y después una ráfaga de publicidad, luego más música, y así sucesivamente. Esta situación es multiplexión por división de tiempo.

A continuación examinaremos la multiplexión por división de frecuencia y después veremos cómo se puede aplicar FDM a la fibra óptica (multiplexión por división de longitud de onda). Después nos enfocaremos en TDM y terminaremos con un sistema TDM avanzado que se usa para fibra óptica (SONET).

Multiplexión por división de frecuencia

La figura 2-31 muestra cómo utilizar FDM para multiplexar tres canales telefónicos de calidad de voz. Los filtros limitan el ancho de banda utilizable a cerca de 3000 Hz por canal de calidad de voz. Cuando se multiplexan muchos canales juntos, se asignan 4000 Hz a cada canal para mantenerlos bien separados. Primero se eleva la frecuencia de los canales de voz, cada uno en una

cantidad diferente, después de lo cual se pueden combinar, porque en ese momento no hay dos canales que ocupen la misma porción del espectro. Observe que aunque existen separaciones entre los canales (bandas de protección), hay cierta superposición entre canales adyacentes porque los filtros no tienen bordes bien definidos. Esta superposición significa que un pico fuerte en el borde de un canal se detectará en el adyacente como ruido no térmico.

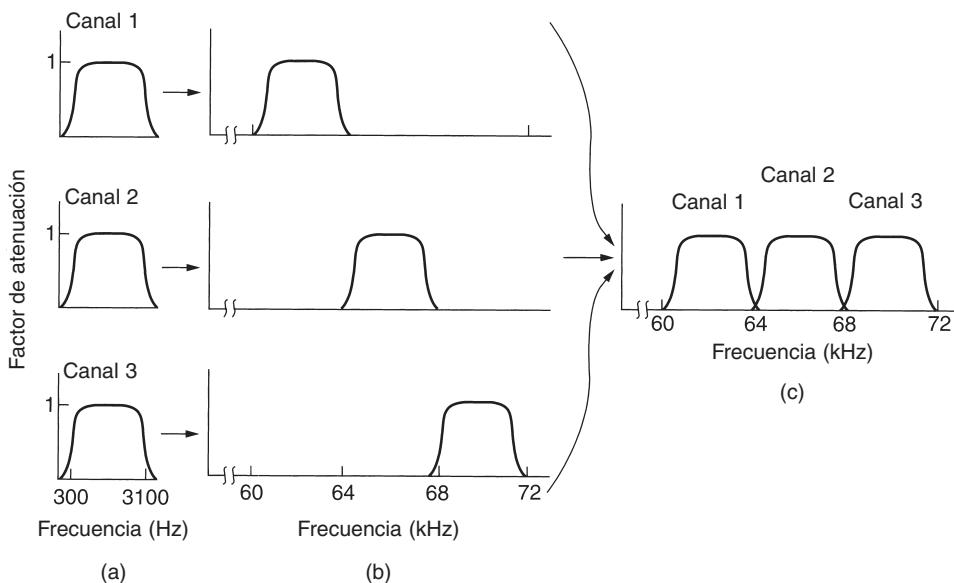


Figura 2-31. Multiplexión por división de frecuencia. (a) Los anchos de banda originales. (b) Incremento de frecuencia de los anchos de banda. (c) El canal multiplexado.

Los esquemas de FDM que se emplean en el mundo están normalizados hasta cierto punto. Un estándar muy difundido es el de 12 canales de voz a 4000 Hz multiplexados dentro de la banda de 60 a 108 kHz. Esta unidad se llama **grupo**. La banda de 12 a 60 kHz a veces se usa para otro grupo. Muchas empresas portadoras ofrecen un servicio de líneas alquiladas de 48 a 56 kbps que se basan en este grupo. Se pueden multiplexar cinco grupos (60 canales de voz) para formar un **supergrupo**. La siguiente unidad es el **grupo maestro**, que se compone de cinco supergrupos (en el estándar del CCITT) o de 10 supergrupos (en el sistema Bell). También existen otros estándares que llegan hasta 230,000 canales de voz.

Multiplexión por división de longitud de onda

Para los canales de fibra óptica se utiliza una variante de la multiplexión por división de frecuencia llamada **WDM (Multiplexión por División de Longitud de Onda)**. En la figura 2-32 se muestran los principios básicos de la WDM en fibra. Aquí, cuatro fibras se juntan en un combinador óptico, cada una con su energía presente a diferentes longitudes de onda. Los cuatro haces se combinan en una sola fibra compartida para transmisión a un destino distante. En el extremo

distante, el haz se divide en tantas fibras como hayan entrado. Cada fibra saliente contiene un núcleo corto especialmente construido que filtra todas las longitudes de onda, excepto una. Las señales resultantes pueden enrutararse a su destino o recombinarse en diferentes formas para transporte adicional multiplexado.

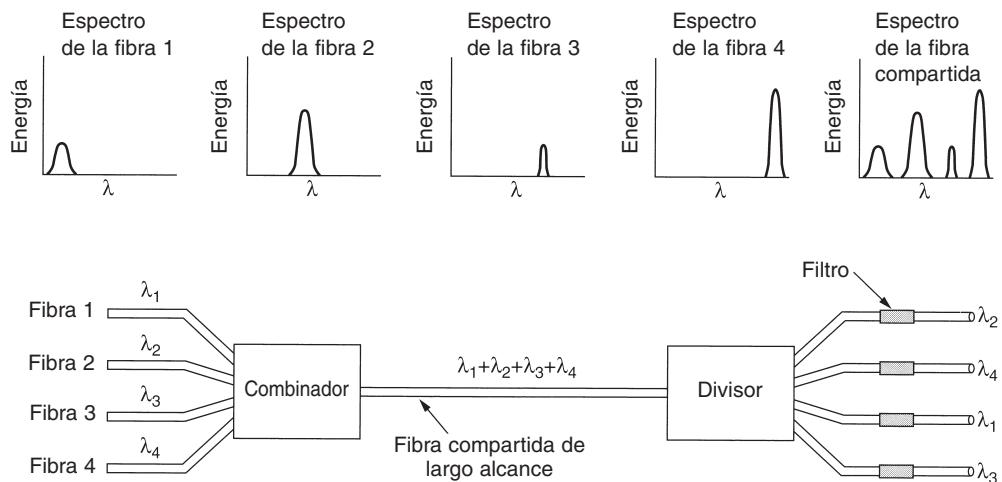


Figura 2-32. Multiplexión por división de longitud de onda.

En realidad, aquí nada es nuevo. Se trata simplemente de multiplexión por división de frecuencia a frecuencias muy altas. Siempre y cuando cada canal tenga su propio rango de frecuencia (es decir, longitud de onda), y todos los intervalos estén separados, se pueden multiplexar juntos en la fibra de largo alcance. La única diferencia con respecto a la FDM eléctrica es que un sistema óptico que usa una rejilla de difracción es totalmente pasivo y, por ello, muy confiable.

La tecnología WDM ha progresado de tal manera que ha dejado en vergüenza a la tecnología de computadoras. La WDM fue inventada en 1990. Los primeros sistemas comerciales tenían ocho canales, cada uno de los cuales era de 2.5 Gbps. En 1998, los sistemas con 40 canales de 2.5 Gbps ya estaban en el mercado. En 2001 había productos con 96 canales de 10 Gbps, con un total de 960 Gbps. Éste es suficiente ancho de banda como para transmitir 30 películas completas por segundo (en MPEG-2). Los sistemas con 200 canales ya están trabajando en el laboratorio. Cuando el número de canales es muy grande y las longitudes de onda están espaciadas entre sí de manera estrecha, por ejemplo a 0.1 nm, el sistema se conoce como **DWDM (WDM Densa)**.

Cabe señalar que la razón por la que WDM es popular es que la energía de una sola fibra por lo general es de unos cuantos gigahertz debido a que en la actualidad es imposible convertir con mayor rapidez entre los medios óptico y eléctrico. Al ejecutar muchos canales en paralelo sobre diferentes longitudes de onda, el ancho de banda agregado se incrementa de manera lineal de acuerdo con el número de canales. Puesto que el ancho de banda de una sola banda de fibra es de alrededor de 25,000 GHz (vea la figura 2-6), teóricamente hay espacio para 2500 canales de 10 Gbps incluso a 1 bit/Hz (también son posibles tasas más altas).

Otro desarrollo novedoso es mediante amplificadores ópticos. Anteriormente, era necesario dividir todos los canales cada 100 km y convertir cada uno en una señal eléctrica para una amplificación por separado antes de volver a convertirlos a ópticos y combinarlos. En la actualidad todos los amplificadores pueden regenerar toda la señal una vez cada 1000 km sin necesidad de múltiples conversiones óptico-eléctricas.

En el ejemplo de la figura 2-32 tenemos un sistema de longitud de onda fija. Los bits de la fibra entrante 1 van a la fibra saliente 3, los de la fibra entrante 2 van a la fibra saliente 1, etcétera. Sin embargo, es posible construir sistemas WDM conmutados. En un dispositivo de ese tipo los filtros de salida se pueden ajustar mediante interferómetros de Fabry-Perot o de Mach-Zehnder. Para mayor información acerca de WDM y su aplicación en la conmutación de paquetes en Internet, vea (Elmirghani y Mouftah, 2000; Hunter y Andonovic, 2000, y Listani y cols., 2001).

Multiplexión por división de tiempo

La tecnología WDM es excelente, pero aún hay mucho cable de cobre en el sistema telefónico, por lo tanto, regresemos a ese tema por un momento. Aunque FDM aún se utiliza sobre cables de cobre o canales de microondas, requiere circuitos analógicos y no es fácil hacerla con una computadora. En contraste, TDM puede manejarse por completo mediante dispositivos digitales y a ello se debe su popularidad en los últimos años. Desgraciadamente, sólo se puede utilizar para datos digitales. Puesto que los circuitos locales producen señales analógicas, se necesita una conversión de analógico a digital en la oficina central, en donde todos los circuitos locales individuales se juntan para combinarse en troncales.

A continuación analizaremos la forma en que las múltiples señales analógicas de voz se digitalizan y combinan en una sola troncal digital saliente. Los datos de cómputo que se envían a través de un módem también son analógicos, por lo que la siguiente descripción también se aplica a ellos. Las señales analógicas se digitalizan en la oficina central con un dispositivo llamado **codec** (codificador-decodificador), con lo que se produce una serie de números de 8 bits. El codec toma 8000 muestras por segundo (125 μ seg/muestra) porque el teorema de Nyquist dice que esto es suficiente para capturar toda la información del ancho de banda de 4 kHz del canal telefónico. A una velocidad de muestreo menor, la información se perdería; a una mayor, no se ganaría información extra. Esta técnica se llama **PCM (Modulación por Codificación de Impulsos)**. La PCM es el corazón del sistema telefónico moderno. En consecuencia, virtualmente todos los intervalos de tiempo dentro del sistema telefónico son múltiplos de 125 μ seg.

Cuando la transmisión digital empezó a surgir como una tecnología factible, el CCITT era incapaz de lograr un acuerdo respecto al estándar internacional para la PCM. En consecuencia, ahora se usan diversos esquemas incompatibles en diferentes países alrededor del mundo.

Un método muy utilizado en Estados Unidos y Japón es el de la portadora **T1**, descrito en la figura 2-33. (Técnicamente hablando, el formato se llama DS1 y la portadora se llama T1, pero aquí no haremos esa sutil distinción.) La portadora T1 consiste en 24 canales de voz que se multiplexan juntos. Por lo común, las señales analógicas se muestran por asignación cíclica (*en round robin*), alimentando el flujo analógico resultante al codec en lugar de tener 24 codecs y después mezclar la salida digital. Cada uno de los 24 canales inserta, a la vez, 8 bits en el flujo de salida.

Siete bits son de datos y uno es de control, con lo que se obtienen $7 \times 8000 = 56,000$ bps de datos, y $1 \times 8000 = 8000$ bps de información de señalización por canal.

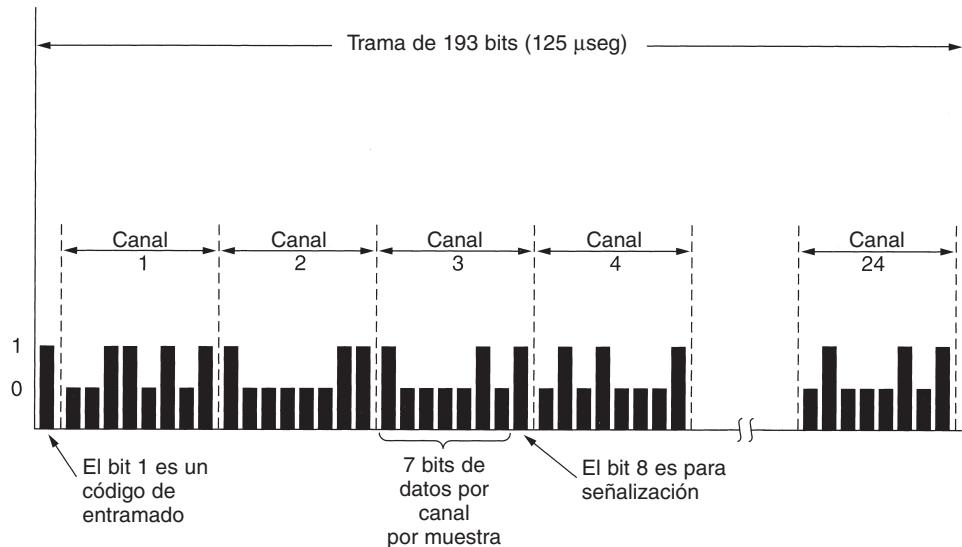


Figura 2-33. La portadora T1 (1.544 Mbps).

Una trama consiste en $24 \times 8 = 192$ bits más un bit extra para entramado, lo que da 193 bits cada $125 \mu\text{seg}$. Esto produce una tasa de transmisión de datos bruta de 1.544 Mbps. El bit número 193 se usa para sincronización de la trama y sigue el patrón 0101010101... Por lo general, el receptor verifica de manera continua este bit para asegurarse de que no ha perdido la sincronización. Si llegara a perder sincronía, el receptor puede esperar hasta detectar otra vez el patrón y volverse a sincronizar. Los clientes analógicos no pueden generar el patrón de bits porque corresponde a una onda senoidal a 4000 Hz, que sería filtrada. Desde luego, los clientes digitales pueden generar este patrón, pero hay poca probabilidad de que esté presente cuando la trama pierda sincronía. Cuando se utiliza un sistema T1 exclusivamente para datos, sólo 23 de los canales llevan datos. El vigésimo cuarto lleva un patrón especial de sincronización que permite la recuperación rápida en caso de que la trama pierda sincronía.

Cuando el CCITT por fin llegó a un acuerdo, sintió que 8000 bps de información de señalización era demasiado, de modo que su estándar de 1.544 Mbps se basa en un elemento de datos de 8 bits en lugar de 7; es decir, la señal analógica se cuantiza en 256 niveles discretos en lugar de 128. Hay dos variantes (incompatibles). En la **señalización por canal común**, el bit extra (que se anexa al final y no al principio de la trama de 193 bits) adopta los valores 10101010... en las tramas nenes y contiene información de señalización para todos los canales de las tramas pares.

En la otra variante, la **señalización por canal asociado**, cada canal tiene su propio subcanal privado de señalización. Se establece un subcanal privado asignando uno de los ocho bits de usuario

de cada sexta trama a funciones de señalización, así que cinco de cada seis muestras tienen 8 bits de ancho y la otra sólo tiene 7. El CCITT también recomendó una portadora PCM a 2.048 Mbps llamada **E1**. Esta portadora empaca 32 muestras de datos de 8 bits en la trama básica de 125 μ seg. Treinta de los canales se usan para información y dos para señalización. Cada grupo de cuatro tramas proporciona 64 bits de señalización, la mitad de los cuales se usa para señalización por canal asociado y el resto se usa para sincronización de tramas o se reserva para que cada país los use como quiera. Fuera de Norteamérica y Japón, se utiliza la portadora E1 de 2.048 Mbps en lugar de la T1.

Una vez que la señal de voz se digitaliza, es tentador tratar de aplicar técnicas estadísticas para reducir la cantidad de bits necesarios por canal. Estas técnicas no sólo son apropiadas para codificar la voz, sino también para digitalizar cualquier señal analógica. Todos los métodos de compactación se basan en el principio de que la señal cambia con relativa lentitud en comparación con la frecuencia de muestreo, de modo que mucha de la información en el nivel digital de 7 u 8 bits es redundante.

Un método llamado **modulación diferencial por codificación de impulsos** consiste en transmitir no la amplitud digitalizada sino la diferencia entre su valor actual y el previo. Puesto que los saltos de ± 16 en una escala de 128 no son probables, podrían bastar 5 bits en lugar de 7. Si la señal llegara a saltar de manera alocada en forma ocasional, la lógica de codificación podría requerir varios períodos de muestreo para "recuperarse". En el caso de la voz, se puede ignorar el error que se introduce.

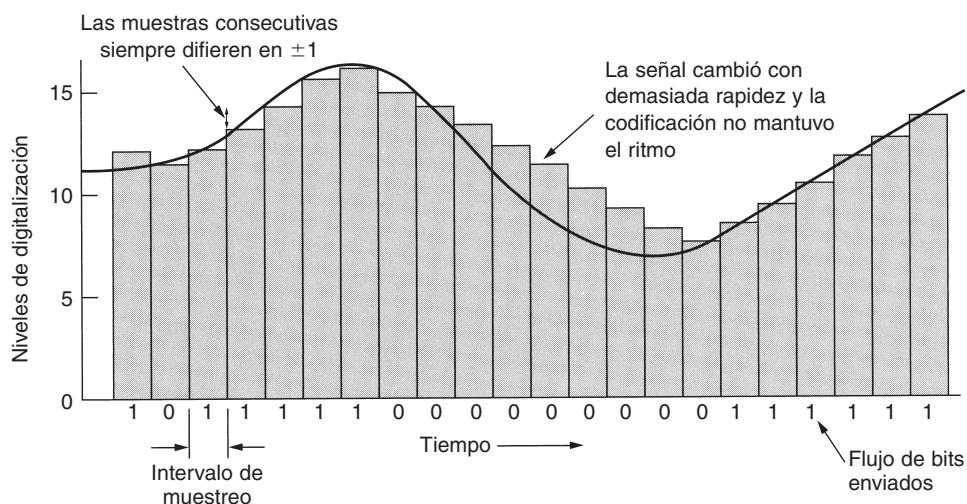


Figura 2-34. Modulación delta.

Una variante de este método de compactación requiere que cada valor muestreado difiera de su predecesor en +1 o -1. Bajo estas condiciones, se transmite un solo bit, que indica si la nueva

muestra está por arriba o por debajo de la anterior. En la figura 2-34 se ilustra esta técnica, llamada **modulación delta**. Al igual que todas las técnicas de compactación que suponen cambios pequeños de nivel entre muestras consecutivas, la codificación delta se puede meter en problemas si la señal cambia con demasiada rapidez, como se aprecia en la figura. Cuando esto sucede, se pierde información.

Una mejora a la PCM diferencial consiste en extrapolar algunos valores previos para predecir el siguiente valor y codificar a continuación la diferencia entre la señal real y la que se predice. Desde luego, el transmisor y el receptor deben utilizar el mismo algoritmo de predicción. A tales esquemas se les conoce como **codificación por predicción** y son útiles porque reducen el tamaño de los números que se codificarán y, por tanto, la cantidad de bits que se enviarán.

La multiplexión por división de tiempo permite que se multiplexen varias portadoras T1 en portadoras de orden más alto. La figura 2-35 muestra cómo se puede hacer esto. A la izquierda vemos que se multiplexan cuatro canales T1 en un canal T2. La multiplexión en T2 y superiores se hace bit por bit, en lugar de byte por byte, como en los 24 canales de voz que forman una trama T1. Cuatro flujos T1 a 1.544 Mbps deberían generar 6.176 Mbps, pero T2 en realidad es de 6.312 Mbps. Los bits adicionales sirven para entramar y para recuperar en caso de que la portadora pierda sincronía. T1 y T3 son utilizadas ampliamente por los clientes, mientras que T2 y T4 sólo se utilizan en el sistema telefónico mismo, por lo que no son muy conocidas.

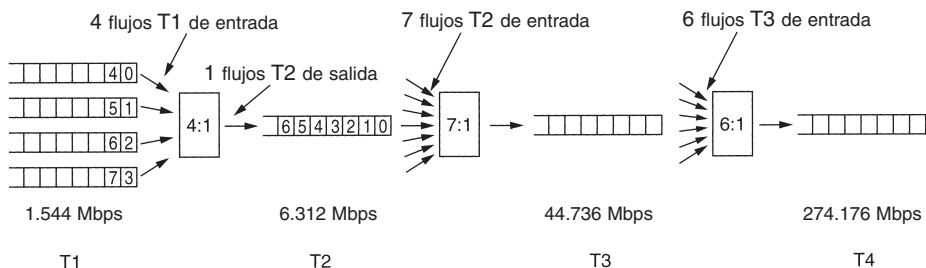


Figura 2-35. Multiplexión de flujos T1 en portadoras más altas.

En el siguiente nivel se combinan siete flujos T2 bit por bit que forman un flujo T3. A continuación se unen seis flujos T3 para formar un flujo T4. En cada paso se agrega una pequeña sobrecarga para entramado y recuperación en caso de que la sincronización entre el emisor y el receptor se pierda.

Así como existe un desacuerdo en lo tocante a la portadora básica entre Estados Unidos y el resto del mundo, también hay desacuerdo respecto a cómo se ha de multiplexar en portadoras de ancho de banda más alto. El esquema de Estados Unidos de dar pasos de 4, 6 y 7 no pareció lógico a todo el mundo, de modo que el estándar del CCITT prescribe la multiplexión de cuatro flujos en uno en cada nivel. Además, los datos de entramado y de recuperación son diferentes entre el estándar de Estados Unidos y el del CCITT. La jerarquía del CCITT para 32, 128, 512, 2048 y 8192 canales funciona a velocidades de 2.048, 8.848, 34.304, 139.264 y 565.148 Mbps.

SONET/SDH

En los primeros días de la fibra óptica, cada compañía telefónica tenía su propio sistema óptico TDM patentado. Después de que AT&T se dividió en 1984, las compañías telefónicas locales se tuvieron que conectar a múltiples empresas portadoras de larga distancia, todas con diferentes sistemas ópticos TDM, de modo que se hizo obvia la necesidad de estandarización. En 1985, Bellcore, la división de investigación de las RBOCs, empezó a trabajar en un estándar llamado **SONET (Red Óptica Síncrona)**. Más tarde, el CCITT se unió al esfuerzo, lo que dio como resultado que en 1989 se produjera un estándar SONET y un conjunto de recomendaciones paralelas del CCITT (G.707, G.708 y G.709). A las recomendaciones del CCITT se les llama **SDH (Jerarquía Digital Síncrona)** pero difieren de SONET sólo en detalles menores. Virtualmente todo el tráfico telefónico de larga distancia en Estados Unidos y una buena parte del mismo en los demás países tiene ahora troncales que funcionan con SONET en la capa física. Si desea información adicional, vea (Bellamy, 2000; Goralski, 2000, y Shepard, 2001).

El diseño de SONET tuvo cuatro objetivos principales. Antes que nada, SONET tenía que hacer posible la interconexión de diferentes operadores telefónicos. El logro de este objetivo requirió que se definiera un estándar de señalización con respecto a la longitud de onda, la temporización, la estructura del entramado, etcétera.

Segundo, se necesitaron medidas para unificar los sistemas digitales estadounidense, europeo y japonés, todos los cuales se basaban en canales PCM de 64 kbps, pero combinados en formas diferentes (e incompatibles).

Tercero, SONET tenía que proporcionar un mecanismo para multiplexar varios canales digitales. En el momento en que se creó SONET, la portadora digital de mayor velocidad que se usaba ampliamente en Estados Unidos era la T3, a 44.736 Mbps. La T4 ya se había definido, pero no se utilizaba mucho, y todavía no se había definido nada por encima de la velocidad de T4. Parte de la misión de SONET era continuar la jerarquía a gigabits/seg y más allá. También se necesitaba una forma estándar de multiplexar canales más lentos en un solo canal SONET.

Cuarto, SONET tenía que proporcionar apoyo para las operaciones, la administración y el mantenimiento (OAM). Los sistemas anteriores no hacían esto muy bien.

Una decisión temprana fue convertir a SONET en un sistema TDM tradicional, con todo el ancho de banda de la fibra dedicado a un canal que contuviera ranuras de tiempo para los distintos subcanales. Como tal, SONET es un sistema síncrono, controlado por un reloj maestro con una precisión de alrededor de 1 parte en 10^9 . En una línea SONET, los bits se envían a intervalos de suma precisión, controlados por el reloj maestro. Cuando posteriormente se propuso que la conmutación de celdas fuera la base de ATM, el hecho de que permitiera la llegada de celdas a intervalos irregulares le confirió la etiqueta de Modo de Transferencia *Asíncrona* para contrastarlo con el funcionamiento síncrono de SONET. Con este último, el emisor y el remitente están atados a un reloj común; con ATM no lo están.

La trama básica de SONET es un bloque de 810 bytes que se emite cada 125 µseg. Puesto que SONET es síncrona, las tramas se emiten haya o no datos útiles que enviar. La velocidad de 8000 tramas/seg coincide perfectamente con la tasa de muestreo de los canales PCM que se utilizan en todos los sistemas de telefonía digital.

Las tramas de 810 bytes de SONET se pueden describir mejor como un rectángulo de bytes de 90 columnas de ancho por nueve filas de alto. De este modo, $8 \times 810 = 6480$ bits se transmiten 8000 veces por segundo, lo que da una tasa de datos bruta de 51.84 Mbps. Éste es el canal básico de SONET y se llama **STS-1 (Señal Síncrona de Transporte 1)**. Todas las troncales de SONET son múltiplos de STS-1.

Las primeras tres columnas de cada trama se reservan para información de administración del sistema, como se ilustra en la figura 2-36. Las primeras tres filas contienen el encabezado de sección (*section overhead*); las siguientes seis contienen el encabezado de línea (*line overhead*). El encabezado de sección se genera y verifica al comienzo y al final de cada sección, mientras que el encabezado de línea se genera y verifica al comienzo y al final de cada línea.

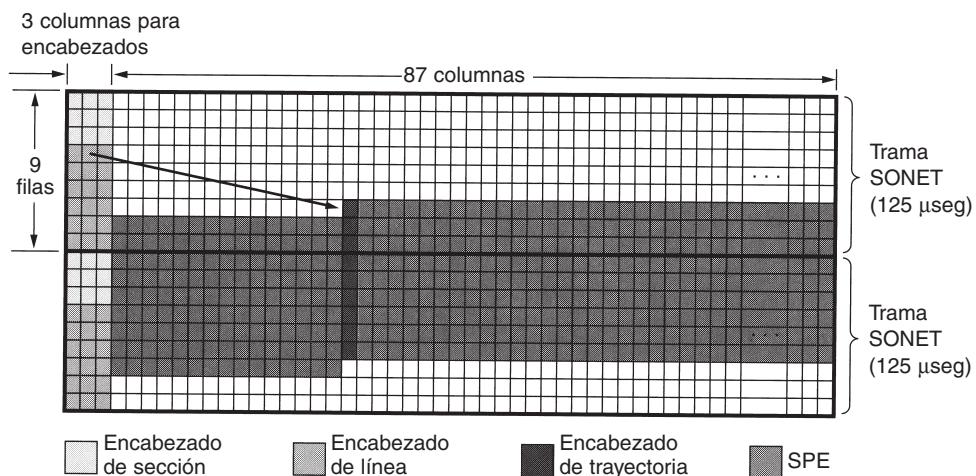


Figura 2-36. Dos tramas SONET consecutivas.

Un transmisor SONET envía tramas consecutivas de 810 bytes, sin huecos entre ellas, incluso cuando no hay datos (en cuyo caso envía datos ficticios). Todo lo que el receptor ve es un flujo continuo de bits, de modo que, ¿cómo sabe dónde comienza cada trama? La respuesta es que los dos primeros bytes de cada trama contienen un patrón fijo que el receptor busca. Si lo encuentra en el mismo lugar en una gran cantidad de tramas consecutivas, asume que está sincronizado con el emisor. En teoría, por lo general un usuario puede insertar este patrón en la carga útil, pero en la práctica no es posible hacer esto debido al multiplexado de múltiples usuarios que se realiza en la misma trama, entre otras razones.

Las 87 columnas restantes contienen $87 \times 9 \times 8 \times 8000 = 50.112$ Mbps de datos de usuario. Sin embargo, los datos de usuario, llamados **SPE (Contenedor o Sobre de Carga Útil Síncrona)**, no siempre empiezan en la fila 1, columna 4. La SPE puede empezar en cualquier parte dentro de la trama. La primera fila del encabezado de línea contiene un apuntador al primer byte. La primera columna de la SPE es del encabezado de trayectoria (es decir, el encabezado para el protocolo de la subcapa de la trayectoria de extremo a extremo).

La facultad de que la SPE empiece en cualquier lugar dentro de la trama SONET o incluso abarque dos tramas, como se muestra en la figura 2-36, confiere una flexibilidad adicional al sistema. Por ejemplo, si una carga útil llega a la fuente mientras se está construyendo una trama SONET ficticia, se puede insertar en la trama actual, en lugar de retenerla hasta el inicio de la siguiente.

En la figura 2-37 se muestra la jerarquía de multiplexión de SONET. Se definieron tasas de STS-1 a STS-192. La portadora óptica que corresponde a cada STS-*n* se llama OC-*n*, pero es la misma bit por bit, excepto por un cierto reordenamiento de bits necesario para la sincronización. Los nombres de SDH son diferentes y empiezan en OC-3 porque los sistemas basados en el CCITT no tienen una tasa de transmisión cercana a los 51.84 Mbps. La portadora OC-9 está presente porque se aproxima mucho a la velocidad de una de las principales troncales de alta velocidad que se usan en Japón. OC-18 y OC-36 se utilizan en Japón. La tasa de datos bruta incluye todos los encabezados. La tasa de datos de SPE excluye los encabezados de línea y de sección. La tasa de datos de usuario excluye todos los encabezados y cuenta solamente las 86 columnas disponibles para la carga útil.

SONET		SDH	Tasa de datos (Mbps)		
Eléctrica	Óptica	Óptica	Bruta	SPE	De usuario
STS-1	OC-1		51.84	50.112	49.536
STS-3	OC-3	STM-1	155.52	150.336	148.608
STS-9	OC-9	STM-3	466.56	451.008	445.824
STS-12	OC-12	STM-4	622.08	601.344	594.432
STS-18	OC-18	STM-6	933.12	902.016	891.648
STS-24	OC-24	STM-8	1244.16	1202.688	1188.864
STS-36	OC-36	STM-12	1866.24	1804.032	1783.296
STS-48	OC-48	STM-16	2488.32	2405.376	2377.728
STS-192	OC-192	STM-64	9953.28	9621.504	9510.912

Figura 2-37. Tasas de multiplexión de SONET y SDH.

Por cierto, cuando una portadora, como la OC-3, no se multiplexa, sino que conduce datos de una fuente única, se agrega la letra *c* (de concatenado) a la designación, de modo que OC-3 indica una portadora de 155.52 Mbps consistente en tres portadoras OC-1 independientes, pero OC-3c indica un flujo de datos de una sola fuente a 155.52 Mbps. Los tres flujos OC-1 dentro de un flujo OC-3c se entrelazan por columnas, primero la columna 1 del flujo 1, a continuación la columna 1 del flujo 2, después la columna 1 del flujo 3 seguida de la columna 2 del flujo 1, y así sucesivamente, lo que produce una trama de 270 columnas de ancho y 9 filas de profundidad.

2.5.5 Conmutación

Desde el punto de vista de un ingeniero de telefonía ordinario, el sistema telefónico se divide en dos partes: planta externa (los circuitos locales y troncales, puesto que están fuera de las oficinas de conmutación) y planta interna (los conmutadores, que están dentro de las oficinas de

comutación). Sólo hemos visto la planta externa. Llegó el momento de examinar la planta interna.

En la actualidad se utilizan dos técnicas de conmutación diferentes: conmutación de circuitos y conmutación de paquetes. A continuación presentaremos una breve introducción a cada una de ellas. Después veremos con detalle la conmutación de circuitos, porque así es como trabaja el sistema telefónico actual. Más adelante, en capítulos posteriores, examinaremos a fondo la conmutación de paquetes.

Comutación de circuitos

Cuando usted o su computadora hacen una llamada telefónica, el equipo de conmutación del sistema telefónico busca una trayectoria física que vaya desde su teléfono al del receptor. Esta técnica se llama **comutación de circuitos** y se muestra de manera esquemática en la figura 2-38(a). Cada uno de los seis rectángulos representa una oficina de conmutación de la empresa portadora (oficina central, oficina interurbana, etcétera). En este ejemplo, cada oficina tiene tres líneas de entrada y tres de salida. Cuando una llamada pasa por una oficina de conmutación, se establece una conexión física (en forma conceptual) entre la línea por la que llegó la llamada y una de las líneas de salida, lo que se representa mediante las líneas punteadas.

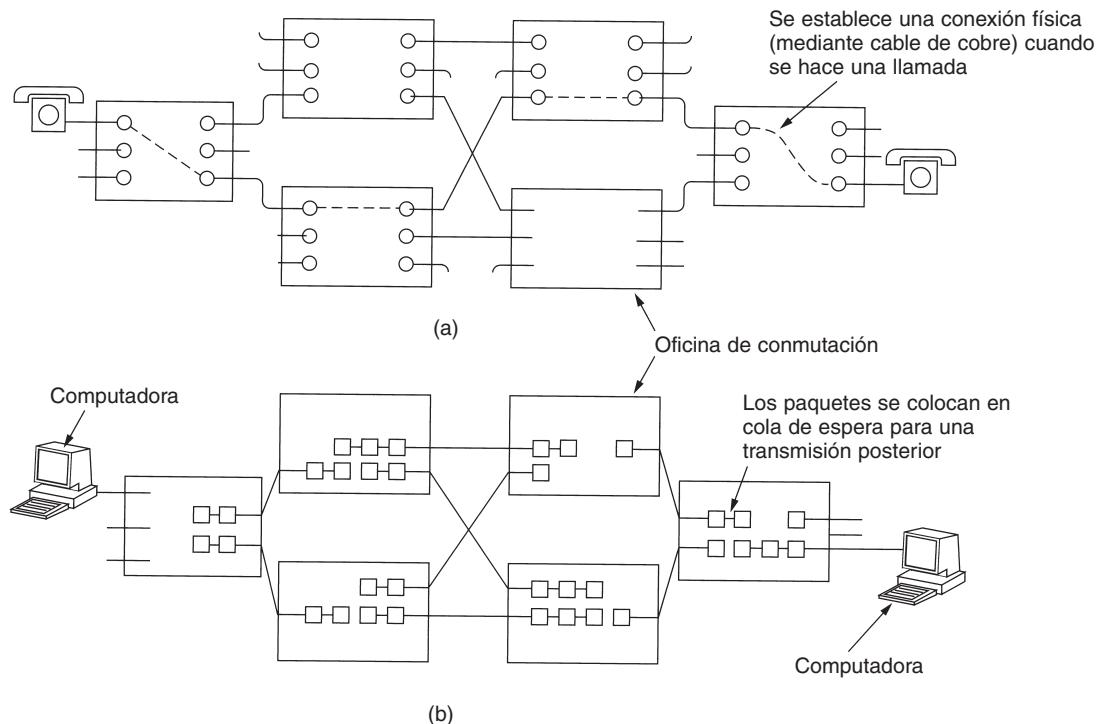


Figura 2-38. (a) Comutación de circuitos. (b) Comutación de paquetes.

En los primeros días del teléfono, se establecía la conexión cuando el operador conectaba un cable puenteador en los enchufes de entrada y de salida. Por cierto, existe una pequeña y sorprendente historia asociada a la invención del equipo de conmutación automática de circuitos: lo inventó el dueño de una funeraria del siglo XIX, un hombre llamado Almon B. Strowger. Poco después de que se inventara el teléfono, cuando alguien moría, alguno de los deudos llamaba a la operadora del pueblo y decía: "Por favor, comuníqueme con una funeraria". Desgraciadamente para el señor Strowger, había dos funerarias en el pueblo, y la esposa del dueño de la otra era la operadora de teléfonos. Strowger pronto se dio cuenta de que si no inventaba el equipo de comunicación telefónica automática iba a quedar en bancarrota, así que eligió la primera opción. Durante casi 100 años, el equipo de conmutación de circuitos empleado en todo el mundo se conoció como el aparato de Strowger. (La historia no registra si la ahora desempleada operadora de conmutador obtuvo trabajo como operadora de información, contestando preguntas como: ¿Me da por favor el número de una funeraria?)

Desde luego, el modelo que se muestra en la figura 2-39(a) está altamente simplificado, porque partes de la trayectoria de "cobre" entre los dos teléfonos pueden ser, de hecho, enlaces de microondas en los cuales se multiplexan miles de llamadas. Sin embargo, la idea básica es válida: una vez que se ha establecido una llamada, existe una trayectoria dedicada entre ambos extremos y continuará existiendo hasta que termine la llamada.

La alternativa a la conmutación de circuitos es la conmutación de paquetes, que se muestra en la figura 2-38(b). Con esta tecnología, los paquetes individuales se envían conforme se necesite, y no se les asigna por adelantado ninguna trayectoria dedicada.

Una propiedad importante de la conmutación de circuitos es la necesidad de establecer una trayectoria de un extremo a otro *antes* de que se pueda enviar cualquier dato. El tiempo que transcurre entre que se termina de marcar y que el timbre comienza a sonar puede ser fácilmente de 10 seg, y más en las llamadas de larga distancia o internacionales. Durante este intervalo de tiempo, el sistema telefónico busca una trayectoria de cobre, como se muestra en la figura 2-39(a). Observe que antes de que pueda comenzar la transmisión de datos, la señal de petición de llamada se debe propagar hasta el destino y se debe confirmar su recepción. En muchas aplicaciones de computadora (por ejemplo, la verificación de crédito en un punto de venta), los tiempos de establecimiento largos son indeseables.

Al existir una trayectoria de cobre entre las partes en comunicación, una vez que se termina de establecer, el único retardo de los datos es el tiempo de propagación de la señal electromagnética, alrededor de 5 mseg por cada 1000 km. Otra ventaja de la trayectoria establecida es que no hay peligro de congestión; es decir, una vez que la llamada entra, no hay posibilidad de obtener una señal de ocupado, aunque podría obtener una antes de establecer la conexión debido a la falta de capacidad de conmutación o de troncal.

Conmutación de mensajes

Una estrategia de conmutación alterna es la **conmutación de mensajes** que se muestra en la figura 2-39(b). Cuando se usa esta forma de conmutación, no se establece por adelantado una trayectoria de cobre físico entre el emisor y el receptor. En cambio, cuando el emisor tiene un blo-

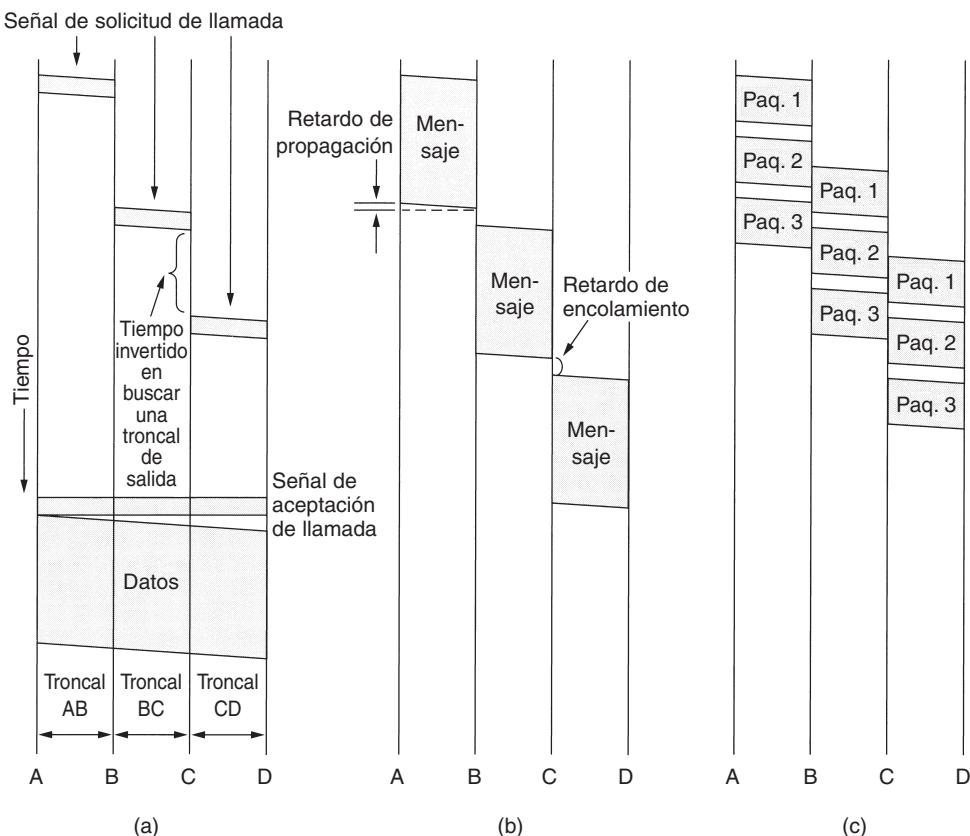


Figura 2-39. Tiempos de los eventos en (a) conmutación de circuitos, (b) conmutación de mensajes, (c) conmutación de paquetes.

que de datos para enviar, éste se almacena en la primera oficina de conmutación (es decir, enruteador) y después se reenvía, un salto a la vez. Cada bloque se recibe en su totalidad, se inspecciona en busca de errores y, después, se retransmite. Una red que utiliza esta técnica se conoce como red de **almacenamiento y reenvío** (*store and forward*), como se mencionó en el capítulo 1.

Los primeros sistemas de telecomunicación electromecánicos usaban conmutación de mensajes para enviar telegramas. El mensaje se perforaba en cinta de papel (fuera de línea) en la oficina emisora y después se leía y transmitía por una línea de comunicación a la siguiente oficina a lo largo del recorrido, donde se perforaba en cinta de papel. Allí, una operadora arrancaba la cinta de papel y la insertaba en una de las muchas lectoras de cinta, una por cada troncal de salida. Tal oficina de conmutación se llamaba **oficina de arrancado de cinta de papel**. La cinta de papel desapareció hace mucho tiempo y la conmutación de mensajes ya no se utiliza, por lo que ya no la analizaremos en este libro.

Comutación de paquetes

Con la conmutación de mensajes, no hay límite para el tamaño de los bloques, lo que significa que los enruteadores (en un sistema moderno) deben tener discos para almacenar en forma temporal los bloques grandes. También significa que un solo bloque puede acaparar una línea de enruteador a enruteador durante minutos, lo que hace inútil la conmutación de mensajes para el tráfico interactivo. Con el fin de resolver estos problemas se inventó la **comutación de paquetes**, como se describió en el capítulo 1. Las redes de conmutación de paquetes establecen un límite superior al tamaño del bloque, lo que permite almacenar los paquetes en la memoria principal del enruteador y no en el disco. Al asegurarse de que ningún usuario pueda monopolizar una línea de transmisión durante mucho tiempo (milisegundos), las redes de conmutación de paquetes pueden manejar tráfico interactivo. En la figura 2-39 (b) y (c) se muestra una ventaja adicional de la conmutación de paquetes sobre la conmutación de mensajes: el primer paquete de un mensaje de varios paquetes se puede reenviar antes de que el segundo haya llegado por completo, lo que reduce el retardo y mejora el rendimiento. Por estas razones, las redes de computadoras por lo general son de conmutación de paquetes, ocasionalmente de conmutación de circuitos y nunca de conmutación de mensajes.

La conmutación de circuitos y la de paquetes difieren en muchos aspectos. Para empezar, la conmutación de circuitos requiere que un circuito se establezca de extremo a extremo antes de que comience la comunicación. La conmutación de paquetes no requiere un establecimiento previo. El primer paquete puede simplemente enviarse tan pronto como esté disponible.

El resultado del establecimiento de conexión mediante la conmutación de circuito es la reserva de ancho de banda que se realiza desde el emisor hasta el receptor. Todos los paquetes siguen esta trayectoria. Entre otras propiedades, el hecho de que todos los paquetes sigan la misma trayectoria significa que no llegarán en desorden a su destino. Con la conmutación de paquetes no hay trayectoria, por lo que diferentes paquetes pueden seguir trayectorias distintas, dependiendo de las condiciones de la red en el momento en el que se enviaron. Pueden llegar en desorden.

La conmutación de paquetes es más tolerante a las fallas que la conmutación de circuitos. De hecho, ésa es la razón por la cual se inventó. Si falla la conmutación, todos los circuitos que la están utilizando se cancelan y no se puede enviar nada más a través de ellos. Con la conmutación de paquetes, los paquetes pueden enrutar evitando a los enruteadores averiados.

Establecer con antelación una trayectoria también abre la posibilidad de reservar ancho de banda con antelación. Si se reserva ese ancho de banda, cuando un paquete llega, puede enviarse de manera inmediata a través de él. Con la conmutación de paquetes no se reserva ningún ancho de banda, por lo que los paquetes podrían tener que esperar su turno para ser reenviados.

Reservar ancho de banda con antelación significa que cuando llegue un paquete no habrá congestión (a menos de que lleguen más paquetes que los esperados). Por otra parte, cuando se intenta establecer un circuito, el intento puede fallar debido a la congestión. Por lo tanto, la congestión puede ocurrir en diversas ocasiones con la conmutación de circuitos (al momento del establecimiento) y con la de paquetes (cuando el paquete se envía).

Si un circuito se ha reservado para un usuario en particular y no hay tráfico que enviar, el ancho de banda de ese circuito se desperdicia. No se puede utilizar para otro tráfico. La conmutación de paquetes no desperdicia ancho de banda y, por lo tanto, es más eficiente desde el punto de vista del sistema. Entender esta compensación es crucial para entender la diferencia entre la con-

mutación de circuitos y la de paquetes. La compensación está entre un servicio garantizado con desperdicio de recursos contra un servicio no garantizado pero sin desperdicio de recursos.

La conmutación de paquetes utiliza transmisión de almacenamiento y reenvío. Un paquete se almacena en la memoria del enrutador y luego se reenvía al siguiente enrutador. Con la conmutación de paquetes los bits simplemente fluyen de manera continua a través del cable. La técnica de almacenamiento y reenvío agrega retardo.

Otra diferencia es que la conmutación de circuitos es totalmente transparente. El emisor y el receptor pueden usar cualquier tasa de transmisión, formato o método de entramado de bits que quieran. La empresa portadora no lo sabe ni le interesa. Con la conmutación de paquetes la empresa portadora determina los parámetros básicos. Una analogía burda sería comparar un camino con una vía de tren. En el primero, el usuario determina el tamaño, la velocidad y la naturaleza del vehículo; en la vía del tren esto lo hace el prestador de servicios. Esta transparencia es la que hace posible que coexistan voz, datos y fax dentro del sistema telefónico.

Una diferencia final entre la conmutación de circuitos y la de paquetes es el algoritmo de cobro. En la conmutación de circuitos, el cobro se ha basado históricamente en la distancia y el tiempo. En el caso de los teléfonos móviles, la distancia, por lo general, no es importante, excepto cuando se trata de llamadas internacionales, y el tiempo tiene poca importancia (por ejemplo, un plan de llamadas con 2000 minutos libres cuesta más que uno con 1000 minutos libres y algunas veces las llamadas de noche o de fin de semana son más baratas de lo normal). En el caso de la conmutación de paquetes, el tiempo de conexión no es un problema, pero con frecuencia el volumen del tráfico sí lo es. Por lo general, los ISPs (proveedores de servicios de Internet) cargan a los usuarios domésticos una tarifa mensual porque es más sencillo y sus clientes pueden entender este modelo con mayor facilidad, pero las empresas portadoras de red dorsal realizan cargos a las redes regionales con base en el volumen de su tráfico. Las diferencias se resumen en la figura 2-40.

Elemento	Conmutación de circuitos	Conmutación de paquetes
Establecimiento de llamada	Requerido	No es necesario
Trayectoria física detallada	Sí	No
Cada paquete puede seguir la misma trayectoria	Sí	No
Los paquetes llegan en orden	Sí	No
Es una falla de conmutación fatal	Sí	No
Ancho de banda disponible	Fijo	Dinámico
Cuándo puede haber congestión	Durante el establecimiento	En cada paquete
Ancho de banda potencialmente desperdiciado	Sí	No
Transmisión de almacenamiento y reenvío	No	Sí
Transparencia	Sí	No
Cargos	Por minuto	Por paquete

Figura 2-40. Comparación de redes de conmutación de circuitos y conmutación de paquetes.

Tanto la conmutación de circuitos como la de paquetes son tan importantes que regresaremos a ellas dentro de poco y describiremos en detalle las diversas tecnologías que se usan.

2.6 EL SISTEMA TELEFÓNICO MÓVIL

El sistema telefónico tradicional (incluso aunque algún día llegará a utilizar la fibra de extremo a extremo de múltiples gigabits) no podrá satisfacer un grupo creciente de usuarios: personas en movimiento. Los usuarios ahora esperan realizar llamadas telefónicas desde aviones, automóviles, albercas y mientras corren en el parque. Dentro de algunos años también esperarán poder enviar correo electrónico y navegar por Web desde cualquiera de las ubicaciones antes mencionadas, entre muchas otras cosas. En consecuencia, hay demasiado interés en la telefonía inalámbrica. En las siguientes secciones estudiaremos este tema con mayor detalle.

Los teléfonos inalámbricos se dividen en dos categorías básicas: **teléfonos inalámbricos** y teléfonos móviles (algunas veces llamados **teléfonos celulares**). Los primeros son dispositivos que consisten en una estación base y un teléfono que se venden en conjunto para utilizarse dentro de una casa. Éstos nunca se utilizan para conectividad de redes, por lo que no los trataremos más. En su lugar nos concentraremos en el sistema móvil, que se utiliza para la comunicación de datos y voz de área amplia.

Los **teléfonos móviles** han pasado por tres generaciones distintas, con tecnologías diferentes:

1. Voz analógica.
2. Voz digital.
3. Voz y datos digitales (Internet, correo electrónico, etcétera).

Aunque la mayor parte de nuestro análisis se concentra en la tecnología de estos sistemas, es interesante mencionar cómo es que las decisiones políticas y de publicidad pueden tener un gran impacto. El primer sistema móvil fue diseñado en Estados Unidos por AT&T y regulado por la FCC. Como resultado, Estados Unidos tenía un solo sistema (analógico), y un teléfono celular comprado en California también funcionaba en Nueva York. En contraste, cuando el sistema móvil apareció en Europa, cada país diseñó su propio sistema, lo cual fue un fracaso.

Europa aprendió de su error y cuando aparecieron los sistemas digitales, los PTTs a cargo del gobierno se unieron y estandarizaron un solo sistema (GSM), por lo que cualquier teléfono móvil europeo funcionaría en cualquier lugar de Europa. En ese entonces, Estados Unidos decidió que el gobierno no debería estar en el negocio de la estandarización, por lo que dejó la cuestión de los sistemas digitales al mercado. Esta decisión resultó en diferentes fabricantes que producían diferentes tipos de teléfonos móviles. En consecuencia, Estados Unidos ahora tiene funcionando dos principales sistemas telefónicos móviles incompatibles (además de otro menor).

A pesar de la ventaja inicial que tenía Estados Unidos, la posesión y el uso de teléfonos móviles en Europa ahora es mayor que en Estados Unidos. El hecho de tener un solo sistema para toda Europa es una de las razones, pero hay más. Una segunda parte en la que Europa y Estados Unidos difieren es en la cuestión de los números telefónicos. En Estados Unidos los teléfonos móviles están mezclados con los teléfonos (fijos) normales. Por lo tanto, no hay forma de que la persona que llama vea si, digamos, (212)234-5678 es el número de un teléfono fijo (barato o de

llamada gratis) o uno de teléfono móvil (llamada costosa). Para que la gente no se asustara de utilizar los teléfonos móviles, las compañías telefónicas decidieron que el dueño de un teléfono móvil pague por las llamadas entrantes. En consecuencia, muchas personas dudaron en comprar un teléfono móvil por miedo a terminar con una gran cuenta por pagar sólo por recibir llamadas. En Europa, los números de los teléfonos móviles tienen un código de área especial (parecido a los números 800 y 900) por lo que se pueden reconocer al instante. Como resultado la regla común de “el que llama paga” también se aplica a los teléfonos en Europa (excepto en las llamadas internacionales en las que el costo se divide).

Un tercer aspecto que ha tenido un gran impacto en la adopción es el amplio uso de los teléfonos móviles prepagados en Europa (hasta de 75% en algunas áreas). Pueden comprarse en muchas tiendas de la misma manera que un radio; simplemente se pagan. Se precargan con, digamos, 20 o 50 euros y pueden recargarse (utilizando un código de PIN secreto) cuando el saldo se acaba. En consecuencia, prácticamente todos los adolescentes y muchos niños de Europa tienen teléfonos móviles (por lo general, prepagados), y de esta manera sus padres pueden localizarlos sin el peligro de que el niño termine con una cuenta enorme. Si el teléfono móvil sólo se utiliza de vez en cuando, su uso es esencialmente libre debido a que no hay un cargo mensual ni uno por llamadas entrantes.

2.6.1 Teléfonos móviles de primera generación: voz analógica

Ya es suficiente sobre los aspectos políticos y de marketing de los teléfonos celulares. Ahora examinemos a la tecnología, comenzando con el sistema más antiguo. Los radioteléfonos móviles se utilizaban de forma esporádica para comunicación marítima y militar durante las primeras décadas del siglo XX. En 1946, el primer sistema de teléfonos instalado en autos se construyó en St. Louis. Este sistema utilizaba un solo transmisor grande colocado en la parte superior de un edificio y tenía un solo canal que servía para enviar y recibir. Para hablar, el usuario tenía que oprimir un botón que habilitaba el transmisor e inhabilitaba el receptor. Tales sistemas, conocidos como **sistemas de oprimir para hablar**, se instalaron en algunas ciudades desde finales de la década de 1950. El radio de banda civil (CB), los taxis y las patrullas policiacas en programas de televisión a veces usan esta tecnología.

En la década de 1960 se instaló el **IMTS (Sistema Mejorado de Telefonía Móvil)**. También utilizaba un transmisor de alta potencia (200 watts), en la cima de una colina, pero tenía dos frecuencias, una para enviar y otra para recibir, por lo que el botón de oprimir para hablar ya no era necesario. Puesto que toda la comunicación desde los teléfonos móviles entraba por un canal diferente del que recibían los teléfonos emisores, los usuarios móviles no podían escucharse unos a otros (a diferencia del sistema de oprimir para hablar empleado en los taxis).

IMTs manejaba 23 canales dispersos desde 150 hasta 450 MHz. Debido al número tan pequeño de canales, los usuarios a veces tenían que esperar bastante tiempo antes de obtener el tono de marcar. Además, debido a la gran potencia del transmisor en la cima de la colina, los sistemas adyacentes tenían que estar alejados varios cientos de kilómetros para evitar la interferencia. Considerando todo, el sistema no era práctico debido a su capacidad limitada.

Sistema Avanzado de Telefonía Móvil

Todo cambió con **AMPS (Sistema Avanzado de Telefonía Móvil)**, inventado por los Laboratorios Bell e instalado por primera vez en Estados Unidos en 1982. Este sistema también se utilizó en Inglaterra, donde se llamó TACS, y en Japón, donde se llamó MCS-L1. Aunque no es lo último en tecnología, lo analizaremos, pues muchas de sus propiedades fundamentales han sido heredadas por su sucesor digital, D-AMPS, con el fin de tener compatibilidad hacia atrás.

En todos los sistemas de telefonía móvil, una región geográfica se divide en **celdas**, razón por la cual los dispositivos se conocen como teléfonos celulares. En AMPS, las celdas normalmente tienen de 10 a 20 km de diámetro; en los sistemas digitales, las celdas son más pequeñas. Cada celda utiliza un conjunto de frecuencias que no es utilizada por ninguno de sus vecinos. La idea clave que confiere a los sistemas celulares más capacidad que todos los sistemas anteriores es el uso de celdas relativamente pequeñas y la reutilización de las frecuencias de transmisión en celdas cercanas (pero no adyacentes). Mientras que un sistema IMTS de 100 km de alcance puede tener una llamada en cada frecuencia, un sistema AMPS podría tener 100 celdas de 10 km en la misma área con 5 a 10 llamadas en cada frecuencia en celdas muy separadas. Por lo tanto, el diseño celular incrementa la capacidad del sistema en un orden de magnitud conforme las celdas se hacen más pequeñas en su área de cobertura. Además, al ser las celdas más pequeñas se necesita menor potencia, lo cual conduce a dispositivos más pequeños y económicos. Los teléfonos de bolsillo tienen una salida de 0.6 watts; los transmisores en los automóviles normalmente son de 3 watts, el máximo permitido por la FCC.

La idea de reutilizar frecuencias se ilustra en la figura 2-41(a). Por lo general, las celdas son casi circulares, pero es más fácil modelarlas como hexágonos. En la figura 2-41(a), las celdas son del mismo tamaño y están agrupadas en unidades de siete celdas. Cada letra indica un grupo de frecuencias. Observe que para cada conjunto de frecuencias hay un espacio de alrededor de dos celdas de ancho en el que esa frecuencia no se reutiliza, proporcionando buena separación y baja interferencia.

Encontrar localidades elevadas para colocar antenas de estación base es un problema importante. Este problema ha llevado a algunas empresas portadoras de telecomunicaciones a forjar alianzas con la Iglesia Católica Romana, puesto que ésta posee un número sustancial de sitios potenciales para antenas en todo el mundo, todos convenientemente bajo una administración única.

En un área en la que la cantidad de usuarios ha crecido tanto que el sistema está sobrecargado, la potencia se reduce y las celdas sobrecargadas se dividen en **microceldas** para permitir una mayor reutilización de las frecuencias, como se muestra en la figura 2-41 (a). Las compañías telefónicas algunas veces crean microceldas temporales, utilizando torres portables con enlaces de satélite, en eventos deportivos, conciertos de rock y otros lugares donde un gran número de usuarios móviles se congrega por algunas horas. Qué tan grandes deben ser las celdas es un tema complejo, y se trata en (Hac, 1995).

En el centro de cada celda está la estación base a la cual transmiten todos los teléfonos de la celda. La estación base consiste en una computadora y un transmisor/receptor conectado a una antena. En un sistema pequeño, todas las estaciones base se conectan a un mismo dispositivo llamado **MTSO (Oficina de Comunicación de Telefonía Móvil)** o **MSC (Centro de Comunicación Móvil)**.

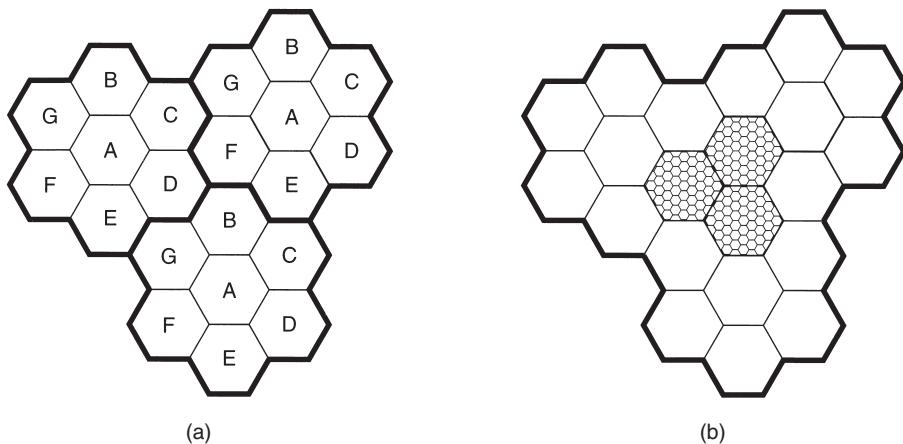


Figura 2-41. (a) Las frecuencias no se reutilizan en celdas adyacentes. (b) Para añadir más usuarios se pueden usar celdas más pequeñas.

En un sistema grande pueden ser necesarias varias MTSOs, las cuales se conectan a una MTSO de segundo nivel, y así sucesivamente. Las MTSOs son esencialmente oficinas centrales como en un sistema telefónico y, de hecho, están conectadas a por lo menos una oficina central del sistema telefónico. Las MTSOs se comunican con las estaciones base, con otras MTSOs y con la PSTN mediante una red de conmutación de paquetes.

En cualquier instante cada teléfono móvil está en una celda específica y bajo el control de la estación base de esa celda. Cuando un teléfono móvil sale de una celda, su estación base nota que la señal telefónica se desvanece o pregunta a todas las estaciones base circundantes cuánta potencia están recibiendo de ella. A continuación, la estación base transfiere la posesión a la celda que está recibiendo la señal más fuerte, esto es, la celda donde se localiza ahora el teléfono. Se informa entonces al teléfono cuál es su nueva base y, si está efectuando una llamada, se le pide que cambie a un nuevo canal (porque el antiguo no se reutiliza en ninguna celda adyacente). Este proceso, llamado **transferencia de celda** (*cell handoff*), tarda cerca de 300 mseg. La asignación de canal es realizada por la MTSO, que es el centro neurálgico del sistema. Las estaciones base sólo son retransmisoras de radio.

Las transferencias de celda pueden realizarse de dos maneras. En la **transferencia suave de celda** (*soft handoff*), el teléfono es adquirido mediante la nueva estación base antes de que las primeras terminen. De esta manera, no hay pérdida de continuidad. La desventaja es que el teléfono necesita estar disponible para sintonizar dos frecuencias al mismo tiempo (la anterior y la nueva). Ni los dispositivos de primera generación ni los de segunda pueden hacer esto.

En la **transferencia dura de celda** (*hard handoff*) la antigua estación base deja el teléfono antes de que la nueva lo adquiera. Si la nueva no puede adquirirlo (por ejemplo, debido a que no hay frecuencia disponible), la llamada se termina de manera abrupta. Los usuarios suelen notar esto, pero con el diseño actual es inevitable que suceda en ocasiones.

Canales

El sistema AMPS emplea 832 canales dúplex, cada uno compuesto por un par de canales simplex. Hay 832 canales de transmisión simplex desde 824 hasta 849 MHz, y 832 canales de recepción simplex desde 869 hasta 894 MHz. Cada uno de estos canales simplex es de 30 kHz de ancho; por lo tanto, AMPS usa FDM para separar los canales.

En la banda de 800 MHz, las ondas de radio son de cerca de 40 cm de largo y viajan en línea recta; son absorbidas por árboles y plantas y rebotan en el suelo y los edificios. Es posible que una señal enviada por un teléfono móvil llegue a la estación base por una trayectoria directa, pero también con un pequeño retardo después de rebotar en el suelo o en un edificio. Esto puede producir un efecto de eco o de distorsión de la señal (desvanecimiento de múltiples trayectorias). A veces es posible incluso oír una conversación distante que ha rebotado varias veces.

Los 832 canales se dividen en cuatro categorías:

1. Control (base a móvil) para administrar el sistema.
2. Localización (base a móvil) para avisar a usuarios móviles que tienen llamadas.
3. Acceso (bidireccional) para establecimiento de llamadas y asignación de canales.
4. Datos (bidireccional) para voz, fax o datos.

Veintiún canales se reservan para control, y están fijos dentro de un PROM en cada teléfono. Puesto que las mismas frecuencias no pueden reutilizarse en celdas cercanas, la cantidad real de canales de voz disponibles por célula es mucho menor que 832, normalmente cerca de 45.

Administración de llamadas

Cada teléfono móvil en AMPS tiene un número de serie de 32 bits y un número telefónico de 10 dígitos en su PROM. El número de teléfono se representa como un código de área de 3 dígitos, en 10 bits, y un número de suscriptor de 7 dígitos, en 24 bits. Cuando un teléfono se enciende, examina una lista preprogramada de 21 canales de control para encontrar la señal más potente.

A continuación, el teléfono difunde su número de serie de 32 bits y su número de teléfono de 34 bits. Al igual que toda la información de control de AMPS, este paquete se envía en forma digital varias veces y con código de corrección de errores, aunque los canales de voz mismos son analógicos.

Cuando la estación base oye el anuncio, avisa a la MTSO, la cual registra la existencia de su nuevo cliente y también informa a la MTSO local del cliente de su ubicación actual. Durante el funcionamiento normal, el teléfono móvil se vuelve a registrar cada 15 minutos aproximadamente.

Para hacer una llamada, un usuario móvil enciende el teléfono, teclea el número al que desea llamar y oprime el botón de Enviar. El teléfono envía entonces el número al que se va a llamar y su propia identidad por el canal de acceso. Si ocurre una colisión, lo intenta nuevamente más tarde.

Cuando la estación base recibe la petición, informa a la MTSO. Si el que llama es un cliente de la compañía de la MTSO (o uno de sus socios), la MTSO busca un canal desocupado para la llamada; si encuentra uno, el número de canal se envía de regreso por el canal de control. A continuación, el teléfono móvil se conecta en forma automática con el canal de voz seleccionado y espera hasta que la persona llamada levante el teléfono.

Las llamadas entrantes funcionan de forma diferente. Para empezar, todos los teléfonos desocupados escuchan continuamente el canal de aviso para detectar mensajes dirigidos a ellos. Cuando se hace una llamada a un teléfono móvil (ya sea desde un teléfono fijo o algún otro teléfono móvil), se envía un paquete a la MTSO local del destinatario de la llamada para averiguar dónde se encuentra. Luego se envía un paquete a la estación base de su celda actual, la cual realiza una difusión por el canal de aviso de la forma: “unidad 14, ¿está ahí?” A continuación el teléfono llamado responde con “Sí” por el canal de control. Enseguida, la base dice algo como: “unidad 14, tiene llamada por el canal 3”. En este punto, el teléfono llamado comuta al canal 3 y empieza a timbrar.

2.6.2 Teléfonos móviles de segunda generación: voz digital

La primera generación de teléfonos móviles fue analógica; la segunda fue digital. Al igual que no hubo estandarización mundial en la primera generación, tampoco la hubo en la segunda. En la actualidad hay cuatro sistemas en uso: D-AMPS, GSM, CDMA y PDC. A continuación analizaremos las primeras tres. PDC sólo se utiliza en Japón y básicamente es un D-AMPS modificado para compatibilidad hacia atrás con el sistema analógico japonés de primera generación. A veces se utiliza el nombre **PCS (Servicios de Comunicaciones Personales)** para indicar el sistema de segunda generación (es decir, el digital). Originalmente denotaba un teléfono móvil que utilizaba la banda de 1900 MHz, pero en la actualidad esa distinción se emplea raramente.

D-AMPS—El Sistema Avanzado de Telefonía Móvil Digital

La segunda generación de los sistemas AMPS es **D-AMPS** y es completamente digital. Se describe en el estándar internacional IS-54 y en su sucesor IS-136. D-AMPS se diseñó con mucho cuidado para que pudiera coexistir con AMPS, a fin de que tanto los teléfonos móviles de primera generación como los de segunda pudieran funcionar de manera simultánea en la misma celda. En particular, D-AMPS utiliza los mismos canales a 30 kHz que AMPS y a las mismas frecuencias a fin de que un canal pueda ser analógico y los adyacentes, digitales. Dependiendo de la mezcla de teléfonos en las celdas, la MTSO de la celda determina cuáles canales son analógicos y cuáles digitales, y puede cambiar tipos de canales de manera dinámica conforme cambie la mezcla de canales en una celda.

Cuando D-AMPS fue introducido como un servicio, se puso disponible una nueva banda de frecuencia para manejar la carga esperada creciente. Los canales ascendentes estaban en el rango de 1850-1910 MHz, y los canales descendentes correspondientes estaban en el rango de 1930-1990 MHz, nuevamente en pares, como en AMPS. En esta banda, las ondas son de 16 cm de

longitud, por lo que una antena de onda estándar de $\frac{1}{4}$ es de sólo 4 cm de longitud, lo que da teléfonos más pequeños. Sin embargo, muchos teléfonos D-AMPS pueden utilizar tanto las bandas de 850-MHz como las de 1900-MHz para obtener una gama más amplia de canales disponibles.

En un teléfono móvil D-AMPS, la señal de voz capturada por el micrófono se digitaliza y comprime utilizando un modelo más refinado que los esquemas de modulación delta y de codificación de predicción que analizamos anteriormente. La compresión toma en cuenta propiedades del sistema de voz humano para obtener el ancho de banda de la codificación PCM estándar de 56 a 8 kbps o menos. La compresión se crea mediante un circuito llamado **vocoder** (Bellamy, 2000). La compresión se realiza en el teléfono, en lugar de en la estación base o en la oficina central, para reducir el número de bits que se envían a través del enlace de aire. Con la telefonía fija, no hay beneficio de hacer que la compresión se realice en el teléfono, debido a que la reducción del tráfico a través del circuito local no incrementa la capacidad del sistema.

Con la telefonía móvil hay una gran ganancia al realizar la digitalización y compresión en el teléfono, tanto que en D-AMPS tres usuarios pueden compartir un solo par de frecuencias que utilicen la multiplexión por división de tiempo. Cada par de frecuencia maneja 25 tramas/seg de 40 mseg cada uno. Además, cada trama se divide en seis ranuras de tiempo de 6.67 mseg cada una, como se muestra en la figura 2-42(a), para el par de frecuencia más bajo.

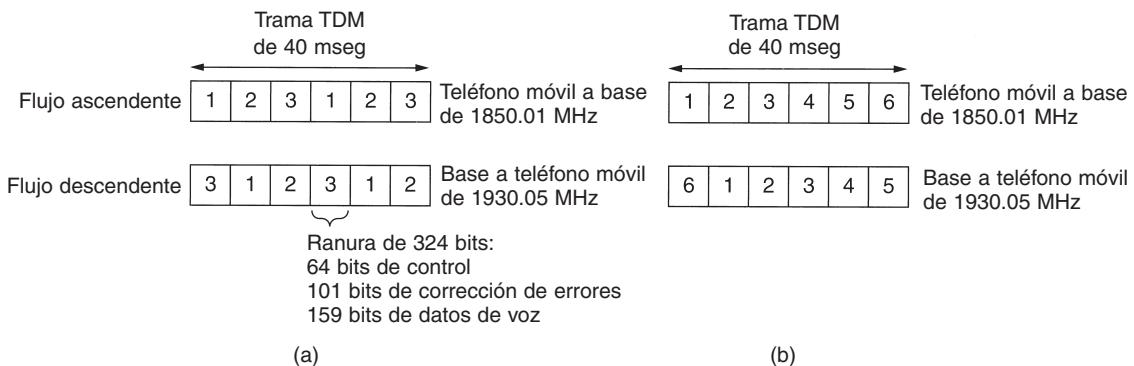


Figura 2-42. (a) Un canal D-AMPS con tres usuarios. (b) Un canal D-AMPS con seis usuarios.

Cada trama mantiene tres usuarios que se turnan para utilizar los enlaces ascendente y descendente. Por ejemplo, durante la ranura 1 de la figura 2-42(a), el usuario 1 podría transmitir a la estación base y el usuario 3 recibir de ella. Cada ranura tiene un tamaño de 324 bits de longitud, de los cuales 64 se utilizan para propósitos de protección, sincronización y control, y los 260 restantes para la carga útil del usuario. De éstos, 101 se utilizan para la corrección de errores a través del enlace de aire con ruido, por lo que a final de cuentas sólo 159 bits se dejan para la voz comprimida. Con 50 ranuras/seg, el ancho de banda disponible para la voz comprimida está por debajo de sólo 8 kbps, que es $1/7$ del ancho de banda estándar PCM.

Al utilizar mejores algoritmos de compresión, es posible obtener la voz por debajo de 4 kbps, en cuyo caso seis usuarios pueden agruparse en una trama, como se ilustra en la figura 2-42(b). Desde el punto de vista del operador, poder comprimir de tres a seis veces tantos usuarios de D-AMPS en el mismo espectro que uno de AMPS es una gran ganancia y explica el porqué de la popularidad de PCS. Por supuesto, la calidad de voz a 4 kbps no se compara con lo que se podría alcanzar a 56 kbps, pero muy pocos operadores de PCS anuncian su calidad de sonido de alta fidelidad. También debe quedar claro que para datos, un canal de 8 kbps no es tan bueno como un módem antiguo de 9600 bps.

La estructura de control de D-AMPS es bastante complicada. En resumen, una supertrama está formada por grupos de 16 tramas y, algunas veces, cada supertrama tiene cierta información de control. Se utilizan seis canales principales de control: configuración del sistema, control en tiempo real y en tiempo no real, localización, respuesta de acceso y mensajes cortos. Pero de manera conceptual, funciona como AMPS. Cuando se enciende un teléfono móvil, hace contacto con la estación base para anunciarle a sí mismo y después escucha un canal de control para llamadas entrantes. Una vez que ha captado un nuevo teléfono móvil, la MTSO informa a la base doméstica del usuario dónde está, y de esta manera las llamadas se pueden enrutar en forma correcta.

Una diferencia entre AMPS y D-AMPS es la manera en que se maneja la transferencia de celdas. En AMPS, la MTSO la maneja por completo sin ayuda de los dispositivos móviles. Como se puede ver en la figura 2-42, en D-AMPS, durante 1/3 del tiempo un teléfono móvil no necesita enviar ni recibir. Utiliza estas ranuras inactivas para medir la calidad de la línea. Cuando descubre que la señal se debilita, se queja con la MTSO, la cual a continuación interrumpe la conexión, en cuyo momento el teléfono móvil trata de sintonizar una señal más fuerte desde otra estación base. Como en AMPS, le toma 300 msec realizar la transferencia de celda. Esta técnica se conoce como **MAHO (Transferencia Asistida Móvil de Celda)**.

GSM—Sistema Global para Comunicaciones Móviles

D-AMPS es ampliamente utilizado en Estados Unidos y (en una forma modificada) en Japón. Casi a nivel mundial, se utiliza un sistema llamado **GSM (Sistema Global para Comunicaciones Móviles)**, e incluso se está comenzando a utilizar en Estados Unidos en una escala limitada. Para una primera aproximación, GSM es similar a D-AMPS. Los dos son sistemas celulares. En ambos se utiliza la multiplexión por división de frecuencia, en el que cada dispositivo móvil transmite en una frecuencia y recibe en una frecuencia mayor (80 MHz más arriba para D-AMPS, 55 MHz más arriba para GSM). Además, en los dos sistemas, se utiliza la multiplexión por división de tiempo para dividir un solo par de frecuencia en ranuras de tiempo compartidas por múltiples teléfonos móviles. Sin embargo, los canales GSM son mucho más anchos que los AMPS (200 kHz en comparación con 30 kHz) y almacenan relativamente pocos usuarios (8 en comparación con 3), lo que da a GSM una tasa de datos mucho más grande por usuario que D-AMPS.

A continuación describiremos brevemente algunas de las propiedades principales de GSM. Sin embargo, el estándar impreso GSM tiene cerca de 5000 páginas. Gran parte de este material se relaciona con los aspectos de ingeniería del sistema, especialmente de los receptores para

manejar la propagación de señal de múltiples trayectorias, y la sincronización de transmisores y receptores. Nada de esto se mencionará en el siguiente análisis.

Cada banda de frecuencia tiene una longitud de 200 kHz, como se muestra en la figura 2-43. Un sistema GSM tiene 124 pares de canales simplex. Cada uno de ellos tiene una longitud de 200 kHz y maneja ocho conexiones por separado, mediante la multiplexión por división de tiempo. A cada estación actualmente activa se le asigna una ranura de tiempo en el par de canal. En teoría, en cada celda se pueden manejar hasta 992 canales, aunque muchos de ellos no están disponibles, para evitar conflictos de frecuencia con las celdas vecinas. En la figura 2-43 las ocho ranuras de tiempo sombreadas pertenecen a la misma conexión, pero en cada dirección hay sólo cuatro. La transmisión y la recepción no suceden en la misma ranura de tiempo porque los radios GSM no pueden transmitir y recibir al mismo tiempo, además de que toma algo de tiempo cambiar de una a otra. Si la estación móvil a la que se le asignó 890.4/935.4 MHz y la ranura de tiempo 2 desea transmitir a la estación base, podría utilizar las cuatro ranuras de tiempo inferiores sombreadas (y las que le sigan), y colocar datos en cada ranura hasta que se hayan enviado todos los datos.

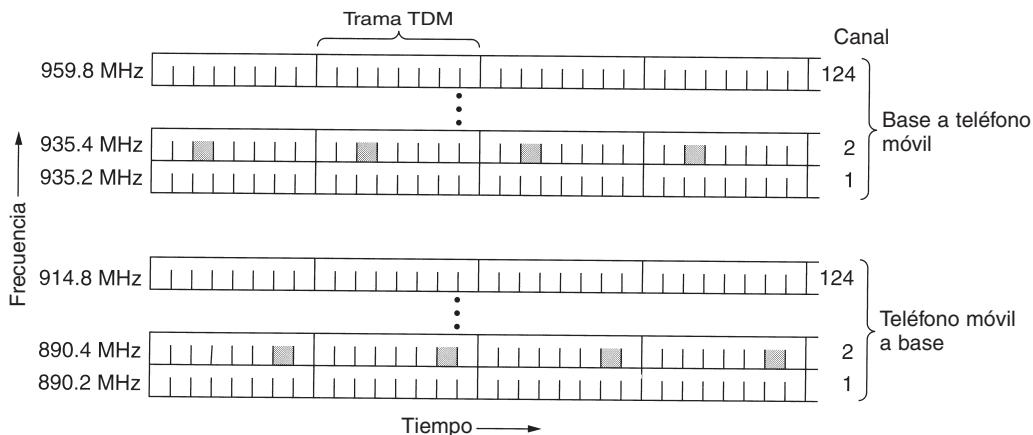


Figura 2-43. GSM utiliza 124 canales de frecuencia, cada uno de los cuales utiliza un sistema TDM de ocho ranuras.

Las ranuras TDM que se muestran en la figura 2-43 son parte de una jerarquía compleja de entramado. Cada ranura TDM tiene una estructura específica, así como grupos de ranuras TDM de multitrrama, que también tienen una estructura específica. En la figura 2-44 se muestra una versión simplificada de esta jerarquía. Observe que una ranura TDM consiste en tramas de datos de 148 bits que ocupan el canal por 577 μ seg (incluyendo un tiempo de protección o guarda de 30 seg después de cada ranura). Cada trama de datos inicia y termina con tres bits 0, para propósitos de delineación de trama. También contiene dos campos de *información* de 57 bits, cada uno de los cuales tiene un bit de control que indica si el siguiente campo de *información* es para voz o para datos. Entre los campos de *información* hay un campo de *sincronización* de 26 bits (entrenamiento) que es utilizado por el receptor para sincronizar los límites de la trama del emisor.

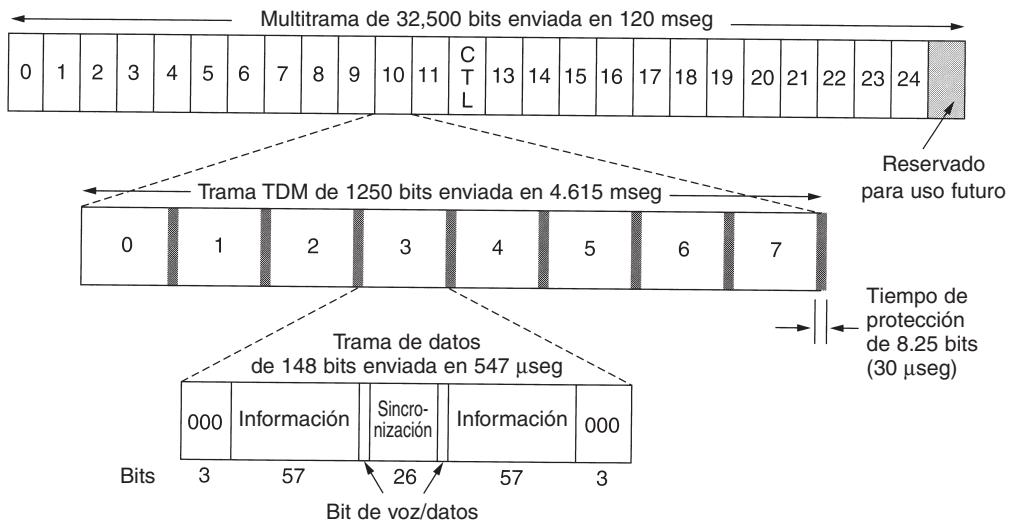


Figura 2-44. Parte de la estructura de entrampado GSM.

Una trama de datos se transmite en 547 µseg, pero un transmisor sólo tiene permitido enviar una trama de datos cada 4.615 mseg, debido a que comparte el canal con otras siete estaciones. La tasa bruta de cada canal es de 270,833 bps, dividida entre ocho usuarios. Esto da un total de 33.854 kbps, más del doble que los 324 bits 50 veces por segundo de 16.2 kbps de D-AMPS. Sin embargo, al igual que con AMPS, la información adicional consume una fracción grande del ancho de banda, lo que finalmente deja 24.7 kbps de carga útil por usuario antes de la corrección de errores. Después de ésta, se dejan 13 kbps para voz, lo que da una calidad de voz sustancialmente mejor que D-AMPS (pero con el costo de utilizar de manera correspondiente más ancho de banda).

Cómo puede ver en la figura 2-44, ocho tramas de datos forman una trama TDM y 26 tramas TDM forman una multitrama de 120 mseg. De las 26 tramas TDM de una multitrama, se utiliza la ranura 12 para el control y la 25 se reserva para uso futuro, de manera que sólo 24 tramas están disponibles para el tráfico del usuario.

Sin embargo, además de la multitrama de 26 ranuras mostrado en la figura 2-44, también se utiliza una multitrama de 51 ranuras (que no se muestra). Algunas de estas ranuras se utilizan para almacenar algunos canales de control utilizados para manejar el sistema. El **canal de control de difusión** es un flujo continuo de salida de la estación base que contiene la identidad de la estación base, así como el estado del canal. Todas las estaciones móviles supervisan su fuerza de señal para ver cuándo se han movido a una nueva celda.

El **canal dedicado de control** se utiliza para actualización de localización, registro y establecimiento de llamada. En particular, cada estación base mantiene una base de datos de las estaciones móviles actualmente bajo su jurisdicción. La información necesaria para mantener esta base de datos se envía en el canal dedicado de control.

Por último, hay un **canal de control común**, que se divide en tres subcanales lógicos. El primero de estos subcanales es el **canal de localización**, que la estación base utiliza para anunciar

llamadas entrantes. Cada estación móvil lo supervisa continuamente en busca de llamadas a las que debería responder. El segundo es el **canal de acceso aleatorio**, que permite que los usuarios soliciten una ranura del canal dedicado de control. Si dos peticiones chocan, se distorsionan y se tienen que volver a realizar más tarde. La estación puede establecer una llamada utilizando la ranura del canal dedicado de control. La ranura asignada es anunciada en el tercer subcanal, el **canal de otorgamiento de acceso**.

CDMA—Acceso Múltiple por División de Código

D-AMPS y GSM son sistemas muy convencionales. Utilizan tanto FDM como TDM para dividir el espectro en canales y éstos en ranuras de tiempo. Sin embargo, hay un tercer sistema, **CDMA (Acceso Múltiple por División de Código)**, que trabaja de una forma completamente diferente. Cuando CDMA fue inicialmente propuesto, la industria tuvo casi la misma reacción que la reina Isabel cuando Colón propuso llegar a la India navegando por una ruta diferente. Sin embargo, debido a la persistencia de una compañía, Qualcomm, CDMA ha madurado al punto en el que no sólo es aceptable, sino que ahora se ve como la mejor solución técnica existente y como la base para los sistemas móviles de la tercera generación. También se utiliza ampliamente en Estados Unidos en los sistemas móviles de segunda generación, y compite de frente con D-AMPS. Por ejemplo, Sprint PCS utiliza CDMA, mientras que AT&T Wireless utiliza D-AMPS. CDMA se describe en el International Standard IS-95 y algunas veces se hace referencia a él mediante ese nombre. También se utiliza el nombre **cdmaOne**.

CDMA es completamente diferente de AMPS, D-AMPS y GSM. En lugar de dividir el rango de frecuencia permitida en algunos cientos de canales estrechos, CDMA permite que cada estación transmita todo el tiempo a través de todo el espectro de frecuencia. Se utiliza la teoría de codificación para separar múltiples transmisiones simultáneas. CDMA no supone que las tramas que colisionan son totalmente distorsionadas. En su lugar, asume que se agregan múltiples señales en forma lineal.

Antes de adentrarnos en el algoritmo, consideremos una analogía: una sala de espera de un aeropuerto con muchas parejas de personas conversando. TDM se compara con todas las personas que están en medio de la sala pero que esperan su turno para hablar. FDM se compara con las personas que están en grupos separados ampliamente, y cada grupo tiene su propia conversación al mismo tiempo, aunque de manera independiente, que los otros. CDMA se compara con el hecho de que todas las personas estén en medio de la sala hablando al mismo tiempo, pero cada pareja hablando en un lenguaje diferente. La pareja que habla francés se concentra en el francés, rechazando todo lo que no sea francés como si fuera ruido. Por lo tanto, la clave de CDMA es tener la capacidad de extraer la señal deseada y rechazar todo lo demás como ruido aleatorio. A continuación se da una descripción algo simplificada de CDMA.

En CDMA, cada tiempo de bit se subdivide en m intervalos cortos llamados **chips**. Por lo general, hay 64 o 128 chips por bit, pero en el ejemplo que se da a continuación por simplicidad utilizaremos 8 chips/bit.

A cada estación se le asigna un código único de m bits llamado **secuencia de chip**. Para transmitir un bit 1, una estación envía su secuencia de chips. Para transmitir un bit 0, envía el comple-

mento de uno de su secuencia de chips. No se permiten otros patrones. Por lo tanto, para $m = 8$, si a la estación A se le asigna la secuencia de chips 00011011, envía un bit 1 mediante el envío de 00011011 y un bit 0 mediante el envío de 11100100.

El incremento de la cantidad de información que se va a enviar de b bits/seg a mb chips/seg sólo puede realizarse si el ancho de banda disponible se incrementa por un factor de m , lo que hace de CDMA una forma de comunicaciones de espectro disperso (suponiendo que no haya cambios en las técnicas de codificación o modulación). Si tenemos una banda de 1 MHz disponible para 100 estaciones, con FDM cada una tendría 10 kHz y podría enviarse a 10 kbps (suponiendo 1 bit por Hz). Con CDMA, cada estación utiliza completamente el megahertzio, por lo que la tasa de chips es de 1 megachip por segundo. Con menos de 100 chips por bit, el ancho de banda efectivo por estación es mayor para CDMA que FDM, y el problema de asignación de canal se resuelve.

Para propósitos de enseñanza, es más conveniente utilizar una notación bipolar, donde el 0 binario es -1 y el 1 es $+1$. Mostraremos las secuencias de chips entre paréntesis, de manera que 1 bit para la estación A ahora se vuelve $(-1-1-1+1+1-1+1+1)$. En la figura 2-45(a) mostramos las secuencias de chips asignadas a cuatro estaciones de ejemplo. En la figura 2-45(b) las mostramos en nuestra notación bipolar.

A: 0 0 0 1 1 0 1 1	A: $(-1-1-1+1+1-1+1+1)$
B: 0 0 1 0 1 1 1 0	B: $(-1-1+1-1+1+1+1-1)$
C: 0 1 0 1 1 1 0 0	C: $(-1+1-1+1+1+1-1-1)$
D: 0 1 0 0 0 0 1 0	D: $(-1+1-1-1-1+1-1)$

(a)

(b)

Seis ejemplos:

$\underline{\underline{-1- C}}$	$S_1 = (-1+1-1+1+1-1-1)$
$\underline{-11- B + C}$	$S_2 = (-2\ 0\ 0\ 0+2+2\ 0-2)$
$\underline{10- A + B}$	$S_3 = (0\ 0-2+2\ 0-2\ 0+2)$
$\underline{101- A + B + C}$	$S_4 = (-1+1-3+3+1-1-1+1)$
$\underline{1111 A + B + C + D}$	$S_5 = (-4\ 0-2\ 0+2\ 0+2-2)$
$\underline{1101 A + B + \bar{C} + D}$	$S_6 = (-2-2\ 0-2\ 0-2+4\ 0)$

(c)

$$\begin{aligned}
 S_1 \bullet C &= (1+1+1+1+1+1+1)/8 = 1 \\
 S_2 \bullet C &= (2+0+0+0+2+2+0+2)/8 = 1 \\
 S_3 \bullet C &= (0+0-2+2+0-2+0-2)/8 = 0 \\
 S_4 \bullet C &= (1+1+3+3+1-1+1-1)/8 = 1 \\
 S_5 \bullet C &= (4+0+2+0+2+0-2+2)/8 = 1 \\
 S_6 \bullet C &= (2-2+0-2+0-2-4+0)/8 = -1
 \end{aligned}$$

(d)

Figura 2-45. (a) Secuencias de chips binarios para cuatro estaciones. (b) Secuencias de chips bipolares. (c) Seis ejemplos de transmisiones. (d) Recuperación de la señal s de la estación C .

Cada estación tiene su propia y única secuencia de bits. Utilicemos el símbolo S para indicar el vector de m chips para la estación S , y \bar{S} para su negación. Todas las secuencias de chips son

ortogonales y apareadas, lo cual quiere decir que el producto interno normalizado de cualquiera de dos secuencias distintas de chips, \mathbf{S} y \mathbf{T} (escritas como $\mathbf{S} \cdot \mathbf{T}$), es 0. Tales secuencias ortogonales de chips se pueden generar utilizando un método conocido como **código de Walsh**. En términos matemáticos, la ortogonalidad de las secuencias de chips se puede expresar como se muestra a continuación:

$$\mathbf{S} \cdot \mathbf{T} = \frac{1}{m} \sum_{i=1}^m S_i T_i = 0 \quad (2-4)$$

En términos simples, entre más parecidos sean los pares, más diferentes serán. Esta propiedad de ortogonalidad será crucial más adelante. Observe que si $\mathbf{S} \cdot \mathbf{T} = 0$, entonces $\mathbf{S} \cdot \bar{\mathbf{T}}$ también es 0. El producto interno normalizado de cualquier secuencia de chips por sí misma es 1:

$$\mathbf{S} \cdot \mathbf{S} = \frac{1}{m} \sum_{i=1}^m S_i S_i = \frac{1}{m} \sum_{i=1}^m S_i^2 = \frac{1}{m} \sum_{i=1}^m (\pm 1)^2 = 1$$

Esto continúa porque cada uno de los términos m del producto interno es 1, por lo que la suma es m . Además, observe que $\mathbf{S} \cdot \bar{\mathbf{S}} = -1$.

Durante cada tiempo de bit, una estación puede transmitir un 1 enviando su secuencia de chips, puede transmitir un 0 enviando el negativo de su secuencia de chips, o puede quedarse en silencio y no transmitir nada. Por el momento, asumimos que todas las estaciones están sincronizadas, por lo que todas las secuencias de chips comienzan al mismo tiempo.

Cuando dos o más estaciones transmiten de manera simultánea, sus señales bipolares se agregan de manera lineal. Por ejemplo, si en un periodo de chips tres estaciones envían +1 y una estación envía -1, el resultado es +2. Uno podría pensar que esto es como agregar voltaje: tres estaciones enviando +1 voltio y una estación enviando -1 voltio da un total de 2 voltios.

En la figura 2-45(c) vemos seis ejemplos de una o más estaciones que transmiten al mismo tiempo. En el primer ejemplo, C transmite un bit 1, por lo que simplemente obtenemos la secuencia de chips de C . En el segundo ejemplo, tanto B como C transmiten bits 1, por lo que obtenemos la suma de sus secuencias de chips bipolares, principalmente:

$$(-1-1+1-1+1+1-1) + (-1+1-1+1+1+1-1) = (-2\ 0\ 0\ 0+2+2\ 0-2)$$

En el tercer ejemplo, la estación A envía un 1 y la estación B envía un 0. Las demás están calladas. En el cuarto ejemplo, A y C envían un 1 mientras que B envía un bit 0. En el quinto ejemplo, todas las estaciones envían un bit 1. Finalmente, en el último ejemplo, A , B y D envían un bit 1, mientras que C envía un bit 0. Observe que cada una de las secuencias S_1 a S_6 que se dan en la figura 2-45(c) sólo representa un tiempo de bit.

Para recuperar el flujo de bits de una estación individual, el receptor debe conocer con anticipación la secuencia de chips de esa estación. Realiza la recuperación calculando el producto interno normalizado de la secuencia de chips recibida (la suma lineal de todas las estaciones que transmitieron) y la secuencia de chips de la estación cuyo flujo de bits se está tratando de recuperar. Si la secuencia de chips recibida es \mathbf{S} y el receptor está tratando de escuchar en una estación cuya secuencia de chips es \mathbf{C} , simplemente calcula el producto interno normalizado, $\mathbf{S} \cdot \mathbf{C}$.

Para ver por qué funciona esto, simplemente imagine que dos estaciones, A y C , transmiten un bit 1 al mismo tiempo que B transmite un bit 0. El receptor ve la suma $\mathbf{S} = \mathbf{A} + \mathbf{B} + \mathbf{C}$ y calcula

$$\mathbf{S} \cdot \mathbf{C} = (\mathbf{A} + \bar{\mathbf{B}} + \mathbf{C}) \cdot \mathbf{C} = \mathbf{A} \cdot \mathbf{C} + \bar{\mathbf{B}} \cdot \mathbf{C} + \mathbf{C} \cdot \mathbf{C} = 0 + 0 + 1 = 1$$

Los primeros dos términos desaparecen porque todos los pares de secuencias de chips han sido cuidadosamente elegidas para ser ortogonales, como se muestra en la ecuación 2-4. Ahora ya debería ser claro por qué esta propiedad debe imponerse en las secuencias de bits.

Un punto de vista alterno acerca de esta situación es imaginar que las tres secuencias vienen por separado, en lugar de sumadas. Después, el receptor podría calcular el producto interno con cada uno por separado y sumar los resultados. Debido a la propiedad ortogonal, todos los productos internos, excepto $\mathbf{C} \cdot \mathbf{C}$ serían 0. Sumar los productos internos y luego calcularlos es lo mismo que calcularlos y luego sumarlos.

Para hacer más concreto el proceso de decodificación, consideremos nuevamente los seis ejemplos de la figura 2-45(c), como se ilustra en la figura 2-45(d). Suponga que el receptor está interesado en extraer el bit enviado por la estación C de cada una de las seis sumas S_1 a S_6 . Calcula el bit sumando los productos apareados de la \mathbf{S} recibida y del vector \mathbf{C} de la figura 2-45(b) y, después, tomando 1/8 del resultado (debido a que aquí $m = 8$). Como se muestra, el bit correcto es decodificado cada vez. Es como hablar francés.

En un sistema CDMA ideal y libre de ruido, es posible hacer que la capacidad (es decir, el número de estaciones) sea arbitrariamente grande, de la misma manera en que la capacidad de un canal Nyquist sin ruido puede hacerse grande de manera arbitraria utilizando más y más bits por muestra. En la práctica, las limitaciones físicas reducen la capacidad en forma considerable. Primero, hemos asumido que todos los chips están sincronizados. En la realidad, tal sincronización es imposible. Lo que puede hacerse es que el emisor y el receptor se sincronicen haciendo que el emisor transmita una secuencia predefinida de chips que sea lo suficientemente grande para que el receptor pueda detectarla. En consecuencia, todas las demás transmisiones (sin sincronizar) se consideran ruido aleatorio. Sin embargo, si no hay muchas de ellas, el algoritmo básico de decodificación aún funciona bien. Existe bastante literatura que describe la superposición de las secuencias de chips a nivel de ruido (Pickholtz y cols., 1982). Como se podría esperar, entre mayor sea la secuencia de chips, mayor será la probabilidad de detectarla de manera correcta en caso de que haya ruido. Para una mayor confiabilidad, la secuencia de bits puede usar un código de corrección de errores. Las secuencias de chips nunca utilizan códigos de corrección de errores.

Una suposición implícita en nuestro análisis es que los niveles de potencia de todas las estaciones son los mismos que el receptor captó. Por lo general, CDMA se utiliza para sistemas inalámbricos con una estación base fija y muchas estaciones móviles a distancias distintas de ella. Los niveles de potencia recibidos en la estación base dependen de qué tan lejos estén los transmisores. Una buena heurística aquí es que cada estación móvil transmita a la estación base con una magnitud de potencia inversa a la que recibe de la estación base. En otras palabras, una estación móvil que reciba una señal débil de la estación base utilizará más potencia que una que reciba una señal fuerte. La estación base también puede dar órdenes explícitas a las estaciones móviles para incrementar o decrementar su potencia de transmisión.

También hemos dado por hecho que el receptor sabe quién es el emisor. Al principio, contando con suficiente capacidad, el receptor puede escuchar al mismo tiempo a todos los emisores mediante la ejecución paralela del algoritmo de decodificación para cada uno de ellos. En la realidad, esto es más fácil de decir que de hacer. CDMA también tiene otros aspectos complicados que se han pasado por alto en esta breve introducción. A pesar de todo, CDMA es un esquema brillante y se está introduciendo rápidamente en la comunicación inalámbrica móvil. Por lo general, funciona en una banda de 1.25 MHz (en comparación con los 30 kHz de D-AMPS y los 200 kHz de GSM), pero maneja muchos más usuarios en esa banda que cualquiera de los otros sistemas. En la práctica, el ancho de banda disponible para cada usuario es tan bueno como el disponible en GSM y, con frecuencia, mejor.

Para un entendimiento muy profundo de CDMA, vea (Lee y Miller, 1998). Un esquema disperso alterno, en donde la dispersión se realiza a través del tiempo en lugar de la frecuencia, se describe en (Crespo y cols., 1995). Otro esquema más se describe en (Sari y cols., 2000). Todas estas referencias requieren conocimientos de ingeniería de comunicaciones.

2.6.3 Teléfonos móviles de tercera generación: voz y datos digitales

¿Cuál es el futuro de la telefonía móvil? Echemos un vistazo. Hay algunos factores que están impulsando a la industria. Primero, el tráfico de datos ya excede el tráfico de voz en la red fija y está creciendo de manera exponencial, mientras que el tráfico de voz es en esencia plano. Muchos expertos de la industria esperan que muy pronto el tráfico de datos domine la voz en dispositivos móviles. Segundo, las industrias telefónica, de entretenimiento y de cómputo han adoptado formatos digitales y están convergiendo rápidamente. Muchas personas están deseosas de un dispositivo portable y ligero que actúe como teléfono, reproductor de CDs, reproductor de DVDs, terminal de correo electrónico, interfaz para Web, máquina de juegos, procesador de texto, etcétera, todo con conectividad inalámbrica a Internet en todo el mundo con un ancho de banda alto. Este dispositivo y cómo conectarlo es de lo que trata la telefonía móvil de tercera generación. Para mayor información, vea (Huber y cols., 2000, y Sarikaya, 2000).

En 1992, la ITU trató de llevar a cabo este sueño y creó un diseño para alcanzarlo, llamado **IMT-2000**. IMT son las siglas de **Telecomunicaciones Móviles Internacionales**. Se le agregó el número 2000 por tres razones: (1) era el año en que se suponía debería funcionar, (2) era a la frecuencia a la que se suponía que trabajaría (en MHz) y (3) era el ancho de banda que el servicio debería tener (en kHz).

No cumplió con nada de lo anterior. En el 2000 no se implementó nada. La ITU recomendó que todos los gobiernos reservaran espectro de 2 GHz a fin de que los dispositivos pudieran llevarse a cualquier país y funcionaran a la perfección. China reservó el ancho de banda requerido pero nadie más lo hizo. Por último, se admitió que 2 Mbps no son factibles para usuarios que se desplazan *mucho* (debido a la dificultad de realizar transferencias de celdas con la rapidez necesaria). Los 2 Mbps son más factibles para usuarios fijos (lo cual podrá competir con ADSL), 384 kbps para usuarios a pie y 144 kbps para conexiones en automóviles. Sin embargo, el área completa de **3G**, como se ha llamado, es un gran caldero de actividad. La tercera generación podría ser menor y llegar más tarde de lo que originalmente se esperaba, pero seguramente sucederá.

Los servicios básicos que se supone que la red IMT-2000 proporcionará a sus usuarios son:

1. Transmisión de voz de alta calidad.
2. Mensajería (lo cual reemplazará al correo electrónico, a los faxes, a SMS, a los salones de conversación, etcétera).
3. Multimedia (reproducir música, ver videos, películas, televisión, etcétera).
4. Acceso a Internet (navegar por Web, incluyendo páginas con audio y vídeo).

Otros servicios adicionales podrían ser la videoconferencia, la telepresencia, los juegos en grupo y el comercio móvil (pasar su teléfono por el cajero para pagar en una tienda). Además, se supone que todos estos servicios estén disponibles a nivel mundial (con conexión automática vía satélite cuando no se encuentre una red terrestre), de manera instantánea (siempre conectado) y con garantía de calidad de servicio.

La ITU visualizó una tecnología sencilla a nivel mundial para IMT-2000, de manera que los fabricantes pudieran construir un solo dispositivo que pudiera venderse y utilizarse en cualquier parte del mundo (similar a un reproductor de CDs y las computadoras, y diferente de los teléfonos y televisiones móviles). Tener una sola tecnología también podría facilitar la vida de los operadores de red y alentaría a más personas a utilizar los servicios. Las guerras de formato, como la batalla entre Beta y VHS cuando aparecieron por primera vez las videograbadoras, no son buenas para los negocios.

Se realizaron varias propuestas y, después de varias selecciones, aparecieron las dos principales. La primera, **W-CDMA (CDMA de Banda Ancha)**, fue propuesta por Ericsson. Este sistema utiliza espectro disperso de secuencia directa del tipo que describimos anteriormente. Se ejecuta en una banda ancha de 5 MHz y se ha diseñado para interactuar con redes GSM aunque no tiene compatibilidad hacia atrás con GSM. Sin embargo, tiene la propiedad de que el invocador puede salir de una celda W-CDMA y entrar a una celda GSM sin perder la llamada. Este sistema fue impulsado por la Unión Europea, que lo llamó **UMTS (Sistema Universal de Telecomunicaciones Móviles)**.

El otro contendiente era **CDMA2000**, propuesto por Qualcomm. Éste también es un diseño de espectro disperso de secuencia directa, básicamente una extensión de IS-95 y es compatible hacia atrás con él. También utiliza un ancho de banda de 5-MHz, pero no ha sido diseñado para interactuar con GSM y no puede entregar llamadas a una celda GSM (ni a una celda DAMPS). Algunas de las diferencias técnicas con respecto a W-CDMA son las siguientes: una tasa de chips diferente, un tiempo de trama diferente, se utiliza un espectro diferente y la sincronización de tiempo se realiza de una manera diferente.

Si los ingenieros de Ericsson y de Qualcomm se reunieran en un cuarto y se les pidiera que crearan un diseño común, tal vez lo podrían hacer. Después de todo, el principio detrás de ambos sistemas es CDMA en un canal de 5 MHz y nadie está dispuesto a morir por esta tasa de chips elegida. El problema real no es la ingeniería, sino las políticas (lo usual). Europa quería un sistema que pudiera interactuar con GSM; Estados Unidos quería un sistema que fuera compatible con uno que ya se distribuía ampliamente en Estados Unidos (IS-95). Cada lado también

mantenía su compañía local (Ericsson se encuentra en Suecia; Qualcomm en California). Por último, Ericsson y Qualcomm se involucraron en numerosas demandas por sus respectivas patentes de CDMA.

En marzo de 1999, las dos compañías resolvieron los problemas legales cuando Ericsson estuvo de acuerdo en comprar la infraestructura de Qualcomm. También estaban de acuerdo en un estándar sencillo 3G, con múltiples opciones incompatibles, que en gran parte simplemente disfraza las diferencias técnicas. A pesar de estas disputas, los dispositivos y servicios de 3G probablemente comiencen a aparecer en los años venideros.

Se ha escrito mucho acerca de los sistemas 3G, la mayor parte los describen como lo mejor desde el pan en rebanadas. Algunas referencias son (Collins y Smith, 2001; De Vriendt y cols., 2002; Harte y cols., 2002; Lu, 2002, y Sarikaya, 2000). Sin embargo, algunos disidentes piensan que la industria está apuntando en la dirección equivocada (Garber, 2002, y Goodman, 2000).

Mientras se espera que termine la batalla por 3G, algunos operadores están dando cautelosamente un pequeño paso en dirección a 3G al ir a lo que algunas veces se llama **2.5G**, aunque 2.1G sería más preciso. Tal sistema es **EDGE (Tasa de Datos Mejorada para la Evolución del GSM)**, que simplemente es GSM con más bits por baudio. El problema es que más bits por baudio también significan más errores por baudio, por lo que EDGE tiene nueve esquemas diferentes para modulación y corrección de errores, que difieren en la cantidad de ancho de banda que se dedica a arreglar los errores introducidos por la velocidad más alta.

Otro esquema de 2.5G es **GPRS (Servicio de Radio de Paquetes Generales)**, que es una red de paquetes superpuestos encima de D-AMPS o GSM. Permite que las estaciones móviles envíen y reciban paquetes IP en una celda que se ejecuta en un sistema de voz. Cuando GPRS está en operación, algunas ranuras de tiempo en algunas frecuencias se reservan para el tráfico de paquetes. La estación base puede manejar de manera dinámica el número y la ubicación de las ranuras de tiempo, dependiendo de la tasa de voz sobre el tráfico de datos de la celda.

Las ranuras de tiempo disponibles se dividen en varios canales lógicos, utilizados para propósitos diferentes. La estación base determina qué canales lógicos se asignan en qué ranuras de tiempo. Un canal lógico se utiliza para bajar paquetes de la estación base a algunas estaciones móviles, y cada paquete indica a quién va destinado. Para enviar un paquete IP, una estación móvil solicita una o más ranuras de tiempo enviando una petición a la estación base. Si la petición llega sin daño alguno, la estación base anuncia la frecuencia y las ranuras de tiempo asignadas al móvil para enviar el paquete. Una vez que el paquete llega a la estación base, se transfiere a Internet mediante una conexión de cable. Puesto que GPRS sólo es una superposición en el sistema de voz existente, es en el mejor de los casos una medida provisional hasta que llegue 3G.

Aunque las redes 3G aún no se han distribuido ampliamente, algunos investigadores consideran a 3G como un asunto terminado y ya no están interesados. Estas personas ahora trabajan en sistemas 4G (Berezdivin y cols., 2002; Guo y Chaskar, 2002; Huang y Zhuang, 2002; Kellerer y cols., 2002, y Misra y cols., 2002). Entre las características propuestas de los sistemas 4G se encuentran un ancho de banda alto, ubicuidad (conectividad en cualquier lado), integración perfecta con redes de cable y especialmente IP, manejo de espectro y de recursos adaptable, radios de software y alta calidad de servicio para multimedia.

Por otro lado, se están estableciendo tantos puntos de acceso a LANs inalámbricas 802.11 por todos lados, que algunas personas piensan que 3G no solamente no es un asunto terminado, sino que está condenado al fracaso. Bajo esta óptica, las personas irán de un punto de acceso 802.11 a otro para mantenerse conectados. Decir que la industria está en medio de un intenso proceso de cambio es poco. Echemos un vistazo en cinco años para ver qué pasa.

2.7 TELEVISIÓN POR CABLE

Hemos estudiado tanto los sistemas inalámbricos como los fijos con suficiente detalle. Ambos jugarán un papel importante en las redes futuras. Sin embargo, hay una alternativa para la conectividad de redes fija que está tomando mucha importancia: las redes de televisión por cable. Muchas personas ya tienen su teléfono y servicio de Internet a través de cable, y los operadores de cable están trabajando arduamente para incrementar su participación de mercado. En las siguientes secciones analizaremos con detalle a la televisión por cable como un sistema de conectividad de redes y lo compararemos con los sistemas telefónicos que acabamos de estudiar. Para mayor información sobre el cable, vea (Laubach y cols., 2001; Louis, 2002; Ovadia, 2001, y Smith, 2002).

2.7.1 Televisión por antena comunal

La televisión por cable se concibió en la última parte de la década de 1940 como una forma de proporcionar mejor recepción a las personas que viven en las áreas rurales o montañosas. El sistema consistió inicialmente en una antena grande en la cima de una colina para captar la señal de televisión, un amplificador, llamado **amplificador head end**, para reforzarla y un cable coaxial para enviarla a las casas de las personas, como se ilustra en la figura 2-46.

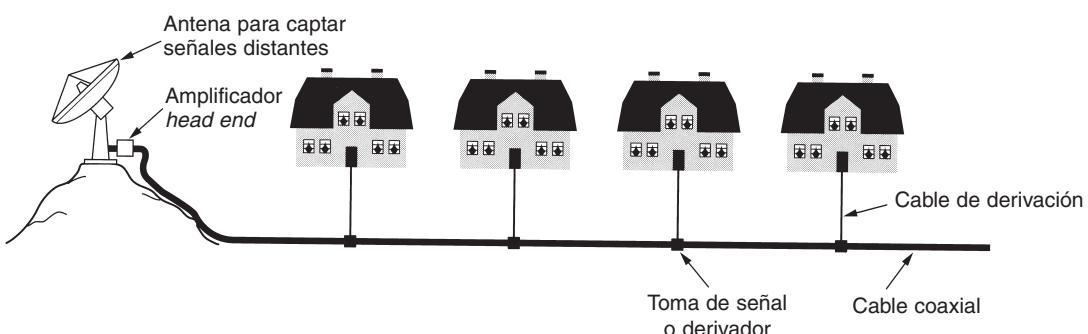


Figura 2-46. Sistema de televisión por cable antiguo.

En sus primeros años, la televisión por cable fue llamada **televisión por antena comunal**. Era un negocio familiar; cualquiera que fuera hábil con la electrónica podía establecer un servicio para su comunidad, y los usuarios podían pagarla en conjunto. Conforme el número de suscriptores crecía, se unían cables adicionales al cable original y se agregaban amplificadores. La transmisión era de una vía, del amplificador *head end* a los usuarios. En 1970 ya existían miles de sistemas independientes.

En 1974, Time, Inc. inició un nuevo canal, Home Box Office, con contenido nuevo (películas) y que se distribuía sólo por cable. Le siguieron otros canales que se transmitían sólo por cable y cuyo contenido eran noticias, deportes, cocina entre muchos otros. Este desarrollo dio origen a dos cambios en la industria. Primero, las grandes compañías comenzaron a comprar sistemas de cable existentes e instalar nuevo cable para adquirir más suscriptores. Segundo, surgió la necesidad de conectar múltiples sistemas, por lo general en ciudades distantes, para distribuir los nuevos canales por cable. Las compañías de cable comenzaron a instalar cable entre ciudades para conectarlas en un solo sistema. Este patrón fue similar a lo que pasó en la industria telefónica 80 años antes con la conexión de las oficinas centrales locales previamente aisladas para hacer posible las llamadas de larga distancia.

2.7.2 Internet a través de cable

A través de los años, el sistema de televisión por cable creció y los cables entre las distintas ciudades se reemplazaron por fibra de ancho de banda alto, de manera similar a lo que sucedió con el sistema telefónico. Un sistema con fibra para distancias considerables y cable coaxial para las casas se conoce como sistema **HFC (Red Híbrida de Fibra Óptica y Cable Coaxial)**. Los convertidores electroópticos que interactúan entre las partes óptica y eléctrica del sistema se llaman **nodos de fibra**. Debido a que el ancho de banda de la fibra es mucho mayor al del cable coaxial, un nodo de fibra puede alimentar múltiples cables coaxiales. En la figura 2-47(a) se muestra parte de un sistema moderno HFC.

En los años recientes, muchos operadores de cable han decidido entrar al negocio de acceso a Internet y con frecuencia también al de la telefonía. Sin embargo, las diferencias técnicas entre la planta de cable y la de telefonía tiene mucho que ver con respecto a lo que se tiene que hacer para alcanzar esas metas. Por un lado, todos los amplificadores de una vía del sistema tienen que reemplazarse por amplificadores de dos vías.

Sin embargo, hay otra diferencia entre el sistema HFC de la figura 2-47(a) y el sistema telefónico de la figura 2-47(b) que es más difícil eliminar. En los vecindarios, muchas casas comparten un solo cable, mientras que en el sistema telefónico, cada casa tiene su propio circuito local privado. Cuando se emplea en la difusión de televisión, esta compartición no tiene importancia. Todos los programas se difunden a través del cable y no importa si hay diez o tres o 10,000 televidentes. Cuando el mismo cable se utiliza para el acceso a Internet, el hecho de que haya 10 o 10,000 usuarios tiene mucha importancia. Si un usuario decide descargar un archivo muy grande, ese ancho de banda se les resta a otros usuarios. Entre más usuarios haya, habrá más competencia por el ancho de banda. El sistema telefónico no tiene esta propiedad particular: descargar un

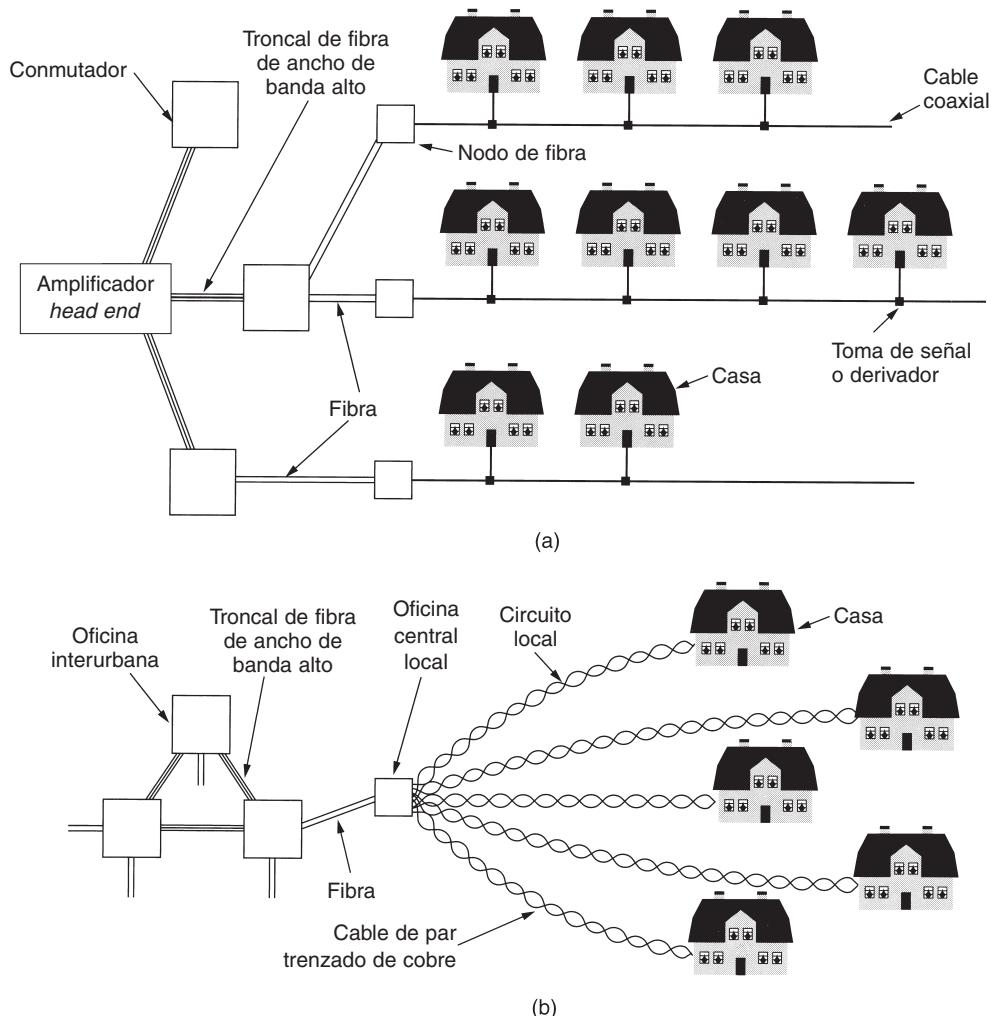


Figura 2-47. (a) Televisión por cable. (b) El sistema telefónico fijo.

archivo grande a través de una línea ADSL no reduce el ancho de banda del vecino. Por otra parte, el ancho de banda del cable coaxial es mucho mayor que el del cable de par trenzado.

La forma en que la industria del cable ha abordado este problema es dividiendo los cables largos y conectándolos directamente al nodo de fibra. El ancho de banda que el amplificador *head end* proporciona a cada nodo de fibra es infinito; por lo tanto, siempre y cuando no haya demasiados suscriptores en cada segmento del cable, la cantidad de tráfico será manejable. En la actualidad, los cables típicos tienen de 500–2000 casas, pero entre más y más gente se suscribe a Internet a través de cable, la carga podría volverse demasiada, lo que requeriría más divisiones y más nodos de fibra.

2.7.3 Asignación de espectro

Deshacerse de todos los canales de TV y utilizar la infraestructura de cable tan sólo para el acceso a Internet tal vez generaría una cantidad considerable de clientes iracundos, por lo que las compañías de cable dudan en hacer esto. Además, la mayoría de las ciudades regulan estrictamente lo que hay en el cable, por lo que podría no permitirse que los operadores de cable hagan esto aunque realmente deseen hacerlo. Como consecuencia, necesitan encontrar una manera de que las televisiones e Internet coexistan en el mismo cable.

Los canales de televisión por cable en Norteamérica normalmente ocupan la región de 54–550 MHz (excepto por la radio FM de 88 a 108 MHz). Estos canales tienen 6 MHz de ancho, incluyendo las bandas de protección. En Europa el extremo inferior por lo general es de 65 MHz y los canales tienen un ancho de 6–8 MHz para la resolución más alta requerida por PAL y SECAM, pero en lo demás el esquema de asignación es similar. La parte baja de la banda no se utiliza. Los cables modernos también pueden operar bien arriba de 550 MHz, con frecuencia a 750 MHz o más. La solución elegida fue introducir canales ascendentes en la banda de 5–42 MHz (un poco más arriba en Europa) y utilizar las frecuencias en el extremo superior para el flujo descendente. El espectro del cable se ilustra en la figura 2-48.

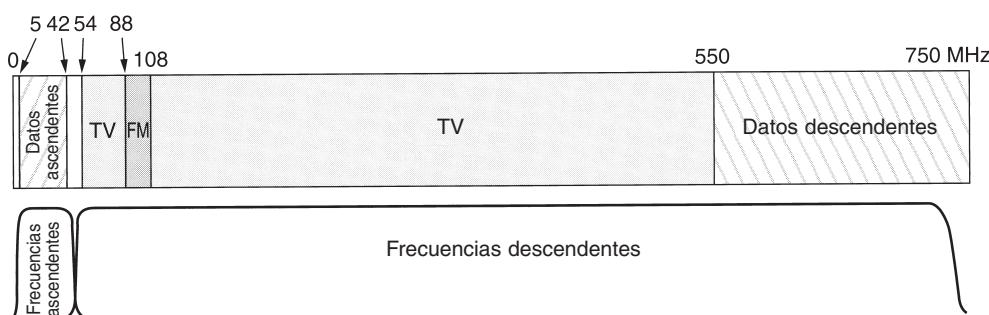


Figura 2-48. Asignación de frecuencia en un sistema típico de cable por TV utilizado para acceso a Internet.

Observe que debido a que todas las señales de televisión son descendentes, es posible utilizar amplificadores ascendentes que funcionen en la región de 5–42 MHz y amplificadores descendentes que sólo funcionen a 54 MHz y mayor, como se muestra en la figura. Por lo tanto, obtenemos una asimetría en los anchos de banda de los flujos ascendente y descendente debido a que hay disponible más espacio arriba del espectro de la televisión que abajo. Por otra parte, la mayor parte del tráfico probablemente sea hacia abajo, por lo que los operadores de cable no están descontentos con este hecho. Como vimos anteriormente, las compañías telefónicas por lo general ofrecen un servicio DSL asimétrico, aunque no tienen ninguna razón técnica para hacerlo.

Los cables coaxiales largos no son mejores para transmitir señales digitales que los circuitos locales largos, por lo que aquí también se necesita la modulación analógica. El esquema usual es tomar cada canal descendente de 6 u 8 MHz y modularlo con QAM-64 o, si la calidad del cable

es muy buena, QAM-256. Con un canal de 6 MHz y QAM-64, obtenemos casi 36 Mbps. Cuando se sustrae la sobrecarga, la carga útil de la red es de aproximadamente 27 Mbps. Con QAM-256, dicha carga es de aproximadamente de 39 Mbps. Los valores europeos son 1/3 más grandes.

Para el flujo ascendente, incluso QAM-64 no funciona bien. Hay mucho ruido proveniente de las microondas, radios CB y otras fuentes, y por esto se utiliza un esquema más conservador —QPSK. Este método (mostrado en la figura 2-25) proporciona 2 bits por baudio en lugar de los 6 u 8 bits que QAM proporciona en los canales descendentes. En consecuencia, la asimetría entre el ancho de banda ascendente y el descendente es mayor de la que se sugiere en la figura 2-48.

Además de actualizar los amplificadores, el operador también tiene que actualizar el amplificador *head end*, de un amplificador tonto a un sistema de cómputo digital inteligente con una interfaz de fibra de ancho de banda alto a un ISP. Por lo general, el nombre también se actualiza, de “amplificador *head end*” a **CMTS (Sistema de Terminación de Módem de Cable)**. En el siguiente texto, nos referiremos a él como “amplificador *head end*”.

2.7.4 Módems de cable

El acceso a Internet requiere un módem de cable, un dispositivo que tiene dos interfaces: una en la computadora y la otra en la red de cable. En los primeros años de Internet por cable, cada operador tenía un módem de cable patentado, que era instalado por un técnico de la compañía de cable. Sin embargo, pronto quedó claro que un estándar abierto podría crear un mercado de módems de cable competitivo y bajar los precios, con lo que se alentaría el uso del servicio. Además, al permitir que los clientes compren los módems de cable en tiendas y que los instalen ellos mismos (como lo hicieron con los módems de teléfono V.9x) se podrían eliminar los temidos camiones de la compañía de cable.

En consecuencia, los operadores de cable más grandes se unieron a una compañía llamada CableLabs para producir un módem de cable estándar y probar la compatibilidad de productos. Este estándar, llamado **DOCSIS (Especificación de Interfaz para Servicio de Datos por Cable)**, está comenzando a reemplazar a los módems patentados. La versión europea se llama **Euro-DOCSIS**. Sin embargo, no a todos los operadores de cable les gusta la idea de un estándar, debido a que muchos de ellos estaban ganando bastante dinero rentando sus módems a sus clientes cautivos. Un estándar abierto con docenas de fabricantes vendiendo módems de cable en tiendas termina con esta práctica tan lucrativa.

La interfaz módem a computadora es directa. En la actualidad, con frecuencia es la Ethernet a 10-Mbps (y en ocasiones es USB). En el futuro, todo el módem podría ser una pequeña tarjeta conectada en la computadora, al igual que con los módems internos V.9x.

El otro extremo es más complicado. Una gran parte del estándar tiene que ver con la ingeniería de radio, tema que está muy alejado del objetivo de este libro. La única parte que vale la pena mencionar aquí es que los módems de cable, al igual que los ADSL, siempre están activos. Establecen una conexión cuando se encienden y la mantienen todo el tiempo que tengan energía, debido a que los operadores de cable no cobran por el tiempo de conexión.

Para entender mejor cómo funcionan, veamos lo que pasa cuando un módem de cable se conecta y activa. El módem explora los canales descendentes en busca de un paquete especial que el

amplificador *head end* transmite periódicamente para proporcionar parámetros del sistema a los módems que se acaban de conectar. Al encontrar este paquete, el nuevo módem anuncia su presencia en uno de los canales ascendentes. El amplificador *head end* responde asignando al módem a sus canales ascendente y descendente. Estas asignaciones pueden cambiarse más tarde si el amplificador *head end* estima que es necesario balancear la carga.

A continuación, el módem determina su distancia con respecto al amplificador *head end* enviándole un paquete especial y tomando el tiempo que tarda en llegar la respuesta. Este proceso se conoce como **alineación** (*ranging*). Es importante que el módem sepa su distancia para reubicar el camino por el que los canales ascendentes funcionan y para obtener la temporización correcta. Dichos canales se dividen en **minirranuras**. Cada paquete ascendente debe ajustarse en una o más minirranuras consecutivas. El amplificador *head end* indica en forma periódica el inicio de una nueva ronda de minirranuras, pero la señal de partida no es escuchada en todos los módems de manera simultánea debido al tiempo de propagación en el cable. Al saber la distancia que separa al módem del amplificador *head end*, cada módem puede calcular el momento en que en realidad se inició la primera minirranura. La longitud de la minirranura depende de la red. Una carga útil típica es de 8 bytes.

Durante la inicialización, el amplificador *head end* también asigna a cada módem una minirranura a fin de utilizarla para solicitar el ancho de banda ascendente. Como regla, la misma minirranura se asignará a múltiples módems, lo que produce contención por las minirranuras. Cuando una computadora necesita enviar un paquete, lo transfiere al módem, el cual a continuación solicita el número necesario de minirranuras para realizar el envío. Si la solicitud es aceptada, el amplificador *head end* coloca una confirmación de recepción en el canal descendente que indica al módem cuáles minirranuras se han reservado para su paquete. A continuación el paquete se envía, comenzando en la minirranura asignada para ese propósito. Es posible solicitar paquetes adicionales mediante el uso de un campo en el encabezado.

Por otro lado, si hay contienda por la minirranura solicitada, no habrá confirmación de recepción y el módem simplemente espera un tiempo aleatorio y vuelve a intentar. Después de cada falla sucesiva, el tiempo aleatorio se duplica. (Para los lectores que ya están familiarizados con la conectividad de redes, este algoritmo sólo es ALOHA ranurado con retroceso exponencial binario. No es posible utilizar Ethernet en el cable porque las estaciones no pueden detectar el medio. En el capítulo 4 retomaremos este tema.)

Los canales descendentes se manejan de manera diferente que los ascendentes. Por un lado, sólo hay un emisor (el amplificador *head end*) por lo que no hay contienda ni necesidad de minirranuras, lo que en realidad es multiplexión estadística por división de tiempo. Por otro lado, el tráfico descendente por lo general es mayor que el ascendente, por lo que se utiliza un tamaño fijo de paquete de 204 bytes. Parte de esto es un código de corrección de errores Reed-Solomon y otra es sobrecarga, lo que deja una carga útil de usuario de 184 bytes. Estos números se eligieron por compatibilidad con la televisión digital que utiliza MPEG-2, por lo que los canales descendente de datos y de TV se formatean de la misma manera. Lógicamente, las conexiones son como se muestra en la figura 2-49.

Volviendo al tema de la inicialización, una vez que el módem ha terminado la alineación y obtenido su canal ascendente, canal descendente y sus asignaciones de minirranuras, puede comenzar

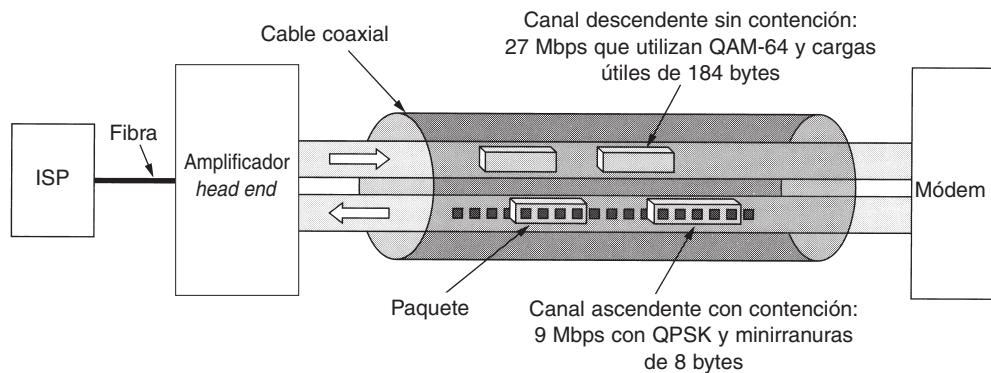


Figura 2-49. Detalles típicos de los canales ascendente y descendente en Norteamérica.

a enviar paquetes. El primer paquete que envía es uno para el ISP pidiéndole una dirección IP, que se asigna de manera dinámica utilizando un protocolo llamado DHCP, el cual analizaremos en el capítulo 5. También solicita y obtiene una hora precisa del amplificador *head end*.

El siguiente paso tiene que ver con la seguridad. Debido a que el cable es un medio compartido, cualquiera que desee utilizarlo puede leer todo el tráfico que pasa a través de él. Para evitar a los curiosos, todo el tráfico se encripta en ambas direcciones. Parte del proceso de inicialización involucra el establecimiento de claves de encriptación. Al principio se podría pensar que es imposible que dos extraños, el amplificador *head end* y el módem, establezcan una clave secreta a plena luz del día con miles de personas observando. Sin embargo, no lo es, pero tendremos que esperar hasta el capítulo 8 para explicar cómo puede hacerse (la respuesta corta sería: mediante el algoritmo de Diffie-Hellman).

Por último, el módem tiene que iniciar sesión y proporcionar su identificador único a través del canal seguro. Hasta este punto la inicialización está completa. A continuación, el usuario puede iniciar sesión en el ISP y comenzar a trabajar.

Hay mucho más por decir acerca de los módems de cable. Algunas referencias son (Adams y Dulchinos, 2001; Donaldson y Jones, 2001, y Dutta-Roy, 2001).

2.7.5 ADSL en comparación con el cable

¿Qué es mejor, ADSL o el cable? Eso es como preguntar cuál sistema operativo es mejor. O qué lenguaje es mejor. O qué religión. La respuesta que obtenga depende de a quién le pregunte. Comparemos ADSL y el cable mediante algunos puntos. Los dos utilizan la fibra óptica en la red dorsal, pero difieren en el extremo. El cable utiliza cable coaxial; ADSL, cable de par trenzado. La capacidad de carga teórica del cable coaxial es de cientos de veces más que el cable de par trenzado. Sin embargo, la capacidad máxima del cable no está disponible para los usuarios de datos porque la mayor parte del ancho de banda del cable se desperdicia en cosas inútiles; por ejemplo, en programas de televisión.

En la práctica, es difícil generalizar acerca de la capacidad efectiva. Los proveedores de ADSL indican específicamente el ancho de banda (por ejemplo, flujo descendente de 1 Mbps, flujo ascendente de 256 kbps) y por lo general logran alrededor de 80% de manera consistente. Los proveedores de cable no dan ninguna indicación pues la capacidad efectiva depende de cuántas personas estén actualmente activas en el segmento de cable del usuario. Algunas veces puede ser mejor que ADSL y otras podría ser peor. Sin embargo, lo que sí puede ser molesto es la incertidumbre. Tener servicio excelente por un minuto no garantiza que al siguiente minuto también lo tendrá, debido a que el ancho de banda más grande de la ciudad ha sido acaparado por otra computadora.

Conforme un sistema ADSL adquiere usuarios, este incremento tiene muy poco efecto en los usuarios existentes, debido a que cada usuario tiene una conexión dedicada. Con el cable, conforme más personas se suscriban al servicio de Internet, el rendimiento de los usuarios existentes disminuirá. El único remedio es que el operador de cable divida los cables ocupados y conecte en forma directa cada uno a un nodo de fibra óptica. Esto cuesta tiempo y dinero, y son presiones de negocios que se deben evitar.

Además, estudiamos otro sistema con un canal compartido como el cable: el sistema telefónico móvil. Aquí un grupo de usuarios también comparte una cantidad fija de ancho de banda. Por lo general, la FDM y la TDM lo dividen estrictamente en porciones fijas entre los usuarios activos pues el tráfico de voz es suave. Pero para el tráfico de datos, esta división rígida es muy inefficiente debido a que los usuarios de datos por lo general están inactivos, en cuyo caso el ancho de banda reservado es un desperdicio. No obstante, en este caso el acceso al cable es más parecido al sistema telefónico móvil que al sistema fijo.

La disponibilidad es un tema en el que ADSL y el cable difieren. Todas las personas tienen teléfono, pero no todos los usuarios están lo suficientemente cerca de su oficina central local para obtener ADSL. Por otro lado, no todos los usuarios tienen cable, pero si usted tiene cable y la compañía proporciona acceso a Internet, puede obtenerlo. Para el nodo de fibra o el amplificador *head end* la distancia no es un problema. También vale la pena mencionar que debido a que el cable inició como un medio de distribución de televisión, pocos negocios cuentan con él.

Debido a que es un medio de punto a punto, ADSL es inherentemente más seguro que el cable. Cualquier usuario de cable puede leer fácilmente todos los paquetes que pasen por el cable. Por esta razón, cualquier proveedor de cable que se precie de serlo encriptará todo el tráfico en ambas direcciones. Sin embargo, el hecho de que su vecino pueda obtener sus mensajes encriptados aún es menos seguro que el hecho de que no obtenga nada.

El sistema telefónico por lo general es más confiable que el cable. Por ejemplo, tiene energía reservada y continúa trabajando de manera normal incluso durante una falla en la energía. Con el cable, si falla la energía de cualquier amplificador de la cadena, todos los usuarios descendentes experimentarán un corte de manera instantánea.

Por último, la mayoría de los proveedores ADSL ofrece una opción de ISPs. Algunas veces la ley los obliga a hacerlo. Éste no siempre es el caso con los operadores de cable.

La conclusión es que ADSL y el cable son tan parecidos como diferentes. Ofrecen servicios comparables y, conforme la competencia entre ellos se avive más, probablemente precios comparables.

2.8 RESUMEN

La capa física es la base de todas las redes. La naturaleza impone dos límites fundamentales a todos los canales, y esto determina su ancho de banda. Estos límites son el de Nyquist, que tiene que ver con los canales sin ruido, y el de Shannon, para canales con ruido.

Los medios de transmisión pueden ser guiados y no guiados. Los principales medios guiados son el cable de par trenzado, el cable coaxial y la fibra óptica. Los medios no guiados incluyen la radio, las microondas, el infrarrojo y los láseres a través del aire. Un sistema de comunicación prometedor es la comunicación por satélite, especialmente los sistemas LEO.

Un elemento clave de la mayor parte de las redes de área amplia es el sistema telefónico. Sus componentes principales son los circuitos locales, troncales y conmutadores. Los circuitos locales son circuitos de cable de par trenzado, analógicos, que requieren módems para transmitir datos digitales. ADSL ofrece velocidades de hasta 50 Mbps dividiendo el circuito local en muchos canales individuales y modulando cada uno por separado. Los ciclos locales inalámbricos son otro nuevo desarrollo que observar, especialmente LMDS.

Las troncales son digitales y se pueden multiplexar de varias formas, incluidas FDM, TDM y WDM. Tanto la conmutación de circuitos como la de paquetes son importantes.

Para las aplicaciones móviles, el sistema de teléfono fijo no es adecuado. En la actualidad los teléfonos móviles se están usando ampliamente para voz y muy pronto se utilizarán ampliamente para datos. La primera generación fue analógica, y dominada por AMPS. La segunda generación fue digital, en la que D-AMPS, GSM y CDMA eran las opciones principales. La tercera generación será digital y se basará en la banda ancha CDMA.

Un sistema alternativo para acceso a red es el sistema de televisión por cable, que ha evolucionado de manera gradual de una antena comunal a una red híbrida de fibra óptica y cable coaxial. Potencialmente, ofrece un ancho de banda muy alto, pero en la práctica, el ancho de banda real disponible depende del número de usuarios activos y de lo que estén haciendo.

PROBLEMAS

1. Calcule los coeficientes de Fourier para la función $f(t) = t$ ($0 \leq t \leq 1$).
2. Un canal sin ruido de 4 kHz se muestrea cada 1 msecg. ¿Cuál es la tasa de datos máxima?
3. Los canales de televisión tienen un ancho de 6 Mhz. ¿Cuántos bits/seg se pueden enviar si se usan señales digitales de cuatro niveles? Suponga que el canal es sin ruido.
4. Si se envía una señal binaria por un canal de 3 kHz cuya relación de señal a ruido es de 20 dB, ¿cuál es la tasa de datos máxima que se puede obtener?
5. ¿Qué relación de señal a ruido se necesita para poner una portadora T1 en una línea de 50 kHz?

6. ¿Qué diferencia hay entre una estrella pasiva y un repetidor activo en una red de fibra óptica?
7. ¿Cuánto ancho de banda existe en 0.1 micras de espectro a una longitud de onda de 1 micra?
8. Se desea enviar una secuencia de imágenes de pantalla de computadora por una fibra óptica. La pantalla es de 480×640 píxeles y cada píxel ocupa 24 bits. Hay 60 imágenes de pantalla por segundo. ¿Cuánto ancho de banda se necesita y cuántas micras de longitud de onda se necesitan para esta banda a 1.30 micras?
9. ¿Se cumple el teorema de Nyquist para la fibra óptica o solamente para el alambre de cobre?
10. En la figura 2-6 la banda de la izquierda es más angosta que las otras. ¿Por qué?
11. A menudo las antenas de radio funcionan mejor cuando el diámetro de la antena es igual a la longitud de la onda de radio. Las antenas prácticas tienen diámetros desde 1 cm hasta 5 m de diámetro. ¿Qué rango de frecuencias cubre esto?
12. El desvanecimiento por múltiples trayectorias alcanza un máximo cuando los dos haces llegan desfasados 180 grados. ¿Qué tan diferentes deben ser las trayectorias para que el desvanecimiento sea máximo para un enlace de microondas de 1 GHz de 50 km de largo?
13. Un rayo láser de 1 mm de diámetro se apunta a un detector de 1 mm de diámetro a 100 m en el techo de un edificio. ¿Cuánta desviación angular deberá tener el láser antes de que pierda al detector?
14. Los 66 satélites de órbita baja en el proyecto Iridium se dividen en seis collares alrededor de la Tierra. A la altitud que están utilizando, el periodo es de 90 minutos. ¿Cuál es el intervalo promedio de transferencias de celdas para un transmisor fijo?
15. Considere un satélite a una altitud de satélites geoestacionarios pero cuyo plan de órbitas se inclina hacia el plano ecuatorial a un ángulo ϕ . Para un usuario fijo en la superficie de la Tierra a una altitud norte ϕ , ¿este satélite da la impresión en el cielo de que no tiene movimiento? De lo contrario, describa su movimiento.
16. Cuántos códigos de oficina central local había antes de 1984, cuando cada oficina central tenía el nombre de los tres dígitos de su código de área y los primeros tres dígitos del número local? Los códigos de área iniciaban con un dígito en el rango de 2–9, tenían un 0 o un 1 como su segundo dígito, y terminaban con cualquier dígito. Los primeros dos dígitos de un número local siempre estaban en el rango de 2–9. El tercer dígito podía ser cualquiera.
17. Utilizando sólo los datos dados en el texto, ¿cuál es la cantidad máxima de teléfonos que el sistema existente de Estados puede manejar sin cambiar el plan de numeración o agregar equipo adicional? ¿Es posible alcanzar esta cantidad de teléfonos? Para propósitos de este problema, una computadora o máquina de fax cuenta como un teléfono. Suponga que sólo hay un dispositivo por línea de suscriptor.
18. Un sistema telefónico simple consiste en dos oficinas centrales locales y una interurbana a la que está conectada cada oficina central por una troncal dúplex de 1 MHz. En promedio, cada teléfono se usa para hacer cuatro llamadas por cada jornada de 8 horas. La duración media de las llamadas es de 6 minutos. El 10% de las llamadas son de larga distancia (esto es, pasan por la oficina interurbana). ¿Cuál es la cantidad máxima de teléfonos que puede manejar una oficina central local? (Suponga que hay 4 kHz por circuito.)
19. Una compañía de teléfonos regional tiene 10 millones de suscriptores. Cada uno de sus teléfonos está conectado a una oficina central local mediante un cable de par trenzado de cobre. La longitud promedio

de estos cables de par trenzado es de 10 km. ¿Cuánto vale el cobre de los circuitos locales? Suponga que la sección transversal de cada filamento es un círculo de 1 mm de diámetro, que el peso específico relativo del cobre es 9.0 y que el cobre se vende a 3 dólares por kilogramo.

20. ¿Un gasoducto es un sistema simplex, uno semidúplex, uno dúplex total, o ninguno de los antes mencionados?
21. El costo de un microprocesador potente se ha reducido a tal grado que ahora es posible incluir uno en cada módem. ¿Cómo afecta esto el manejo de errores en las líneas telefónicas?
22. Un diagrama de constelación de módem, similar al de la figura 2-25, tiene puntos de datos en las siguientes coordenadas: (1, 1), (1, -1), (-1, 1) y (-1, -1). ¿Cuántos bps puede lograr un módem a 1200 baudios con estos parámetros?
23. Un diagrama de constelación de módem, similar al de la figura 2-25, tiene puntos de datos en (0, 1) y (0, 2). ¿El módem usa modulación de fase o modulación de amplitud?
24. En un diagrama de constelación todos los puntos están en un círculo centrado en el origen. ¿Qué tipo de modulación se utiliza?
25. ¿Cuántas frecuencias utiliza un módem QAM-64 de dúplex total?
26. Un sistema ADSL que utiliza DMT asigna 3/4 de los canales de datos disponibles al enlace descendente. Utiliza modulación QAM-64 en cada canal. ¿Cuál es la capacidad del enlace descendente?
27. En el ejemplo de cuatro sectores LMDS de la figura 2-30, cada sector tiene su propio canal de 36 Mbps. De acuerdo con la teoría de encolamiento, si el canal está cargado en 50%, el tiempo de encolamiento será igual que el de descarga. Bajo estas condiciones, ¿cuánto tiempo se tarda en bajar una página Web de 5 KB? ¿Cuánto tiempo se tarda en bajar la página a través de una línea ADSL de 1 Mbps? ¿A través de un módem de 56 kbps?
28. Diez señales, cada una de las cuales requiere 4000 Hz, se multiplexan en un solo canal utilizando FDM. ¿Cuál es el ancho de banda mínimo requerido para el canal multiplexado? Suponga que las bandas de protección tienen un ancho de 400 Hz.
29. ¿Por qué se fijó el tiempo de muestreo de PCM en 125 μ seg?
30. ¿Cuál es el porcentaje de sobrecarga en una portadora T1?; esto es, ¿qué porcentaje de los 1.544 Mbps no se entrega al usuario final?
31. Compare la tasa de datos máxima de un canal sin ruido de 4 kHz que utiliza:
 - (a) Codificación analógica con 2 bits por muestra.
 - (b) El sistema T1 de PCM.
32. Si un sistema de portador T1 pierde la pista de dónde está, trata de resincronizarse con base en el primer bit de cada trama. ¿Cuántas tramas se tendrían que inspeccionar en promedio para resincronizarse con una probabilidad de 0.001 de estar en un error?
33. ¿Cuál es la diferencia, si la hay, entre la parte desmoduladora de un módem y la parte codificadora de un codec? (Después de todo, ambas convierten señales analógicas a digitales.)
34. Una señal se transmite en forma digital por un canal sin ruido de 4 kHz, con una muestra cada 125 μ seg. ¿Cuántos bits por segundo se envían realmente con cada uno de los siguientes métodos de codificación?
 - (a) CCITT, 2.048 Mbps estándar.

- (b) DPCM con un valor de señal relativo de 4 bits.
(c) Modulación delta.
35. Se codifica una onda senoidal pura de amplitud A usando modulación delta, con x muestras/seg. Una salida de +1 corresponde a un cambio de señal de $+A/8$, y una señal de salida de -1 corresponde a un cambio de señal de $-A/8$. ¿Cuál es la frecuencia más alta que se puede rastrear sin error acumulativo?
36. Los relojes de SONET tienen una tasa de arrastre de casi 1 parte en 10^9 . ¿Cuánto tiempo tomará para que el arrastre iguale el ancho de 1 bit? ¿Cuáles son las implicaciones de este cálculo?
37. En la figura 2-37 se establece que la tasa de datos del usuario para OC-3 es de 148.608 Mbps. Demuestre cómo se puede deducir esta cantidad de los parámetros de OC-3 de SONET.
38. Para acomodar tasas de datos menores que STS-1, SONET tiene un sistema de tributarias virtuales (VT). Una VT es una carga útil parcial que se puede insertar en una trama STS-1 y combinar con otras cargas útiles parciales para llenar la trama de datos. VT1.5 utiliza 3 columnas, VT2 utiliza 4, VT3 utiliza 6 y VT6 utiliza 12 columnas de una trama STS-1. ¿Cuál VT puede acomodar
(a) Un servicio DS-1 (1.544 Mbps)?
(b) Un servicio europeo CEPT-1 (2.048 Mbps o E1)?
(c) Un servicio DS-2 (6.312 Mbps)?
39. ¿Cuál es la diferencia esencial entre la conmutación de mensajes y la de paquetes?
40. ¿Cuál es el ancho de banda disponible para el usuario en una conexión OC-12c?
41. Tres redes de conmutación de paquetes contienen n nodos cada una. La primera red tiene una topología de estrella con un commutador central, la segunda es un anillo (bidireccional) y la tercera está interconectada por completo, con una conexión de cada nodo hacia cada uno de los otros nodos. ¿Cuáles son las rutas de transmisión óptima, media y de peor caso en saltos?
42. Compare el retardo al enviar un mensaje de x bits por una trayectoria de k saltos en una red de conmutación de circuitos y en una red de conmutación de paquetes (con carga ligera). El tiempo de establecimiento de circuito es de s segundos, el retardo de propagación es de d segundos por salto, el tamaño del paquete es de p bits y la tasa de datos es de b bps. ¿En qué condiciones tiene un retardo menor la red de paquetes?
43. Suponga que se van a transmitir x bits de datos de usuario por una trayectoria de k saltos en una red de conmutación de paquetes como una serie de paquetes, cada uno contiene p bits de datos y h bits de encabezado, donde $x \geq p + h$. La tasa de bits de las líneas es de b bps y el retardo de propagación es nulo. ¿Qué valor de p minimiza el retardo total?
44. En un sistema de telefónico móvil típico con celdas hexagonales se permite reutilizar una banda de frecuencia en una celda adyacente. Si están disponibles 840 frecuencias, ¿cuántas se pueden utilizar en una celda dada?
45. El diseño real de las celdas rara vez es tan regular como se muestra en la figura 2-41. Incluso la forma de las celdas individuales por lo general es irregular. Dé una posible razón de por qué sucedería esto.
46. Realice una estimación aproximada de la cantidad de microceldas PCS con un diámetro de 100 m que se requerirían para cubrir San Francisco (120 km^2).

47. Algunas veces cuando un usuario móvil cruza el límite de una celda a otra, la llamada actual se termina de manera repentina, aunque todos los transmisores y receptores estén funcionando correctamente. ¿Por qué?
48. D-AMPS tiene evidentemente una calidad de voz menor que GSM. ¿Ésta es la razón por la que D-AMPS necesita tener compatibilidad hacia atrás con AMPS, y GSM no? Si no es así, explique la causa.
49. Calcule el número máximo de usuarios que D-AMPS puede manejar de manera simultánea dentro de una celda. Realice el mismo cálculo para GSM. Explique la diferencia.
50. Suponga que A , B y C , transmiten de manera simultánea bits 0 mediante un sistema CDMA con las secuencias de chips que se muestran en la figura 2-45(b). ¿Cuál es la secuencia de chips resultante?
51. En el análisis acerca de la ortogonalidad de las secuencias de chips CDMA se dijo que si $\mathbf{S} \cdot \mathbf{T} = 0$, entonces $\mathbf{S} \cdot \bar{\mathbf{T}}$ también es 0. Pruebe esto.
52. Considere una manera diferente de mirar la propiedad de ortogonalidad de las secuencias de chips CDMA. Cada bit en un par de secuencias puede o no coincidir. Exprese la propiedad de ortogonalidad en términos de coincidencias y falta de coincidencias.
53. Un receptor CDMA obtiene los siguientes chips: $(-1 +1 -3 +1 -1 -3 +1 +1)$. Suponiendo las secuencias de chips definidas en la figura 2-45(b), ¿cuáles estaciones transmitieron y qué bits envió cada una?
54. En su parte más baja, el sistema telefónico tiene forma de estrella, y todos los circuitos locales de un vecindario convergen en una oficina central local. En contraste, la televisión por cable consiste en un solo cable largo que pasa por todas las casas del mismo vecindario. Suponga que un cable de TV fuera de fibra óptica de 10 Gbps en lugar de cable de cobre. ¿Podría utilizarse para simular un modelo telefónico en el que todo mundo tuviera su propia línea privada a la oficina central local? Si esto fuera posible, ¿cuántas casas con un teléfono podrían conectarse a una sola fibra óptica?
55. Un sistema de TV por cable tiene cien canales comerciales y todos ellos alternan programas con anuncios. ¿Esto es más parecido a TDM o a FDM?
56. Una compañía de cable decide proporcionar acceso a Internet a través de cable en un vecindario que consiste en 5000 casas. La compañía utiliza cable coaxial y asignación de espectro que permite un ancho de banda descendente de 100 Mbps por cable. Para atraer clientes la compañía decide garantizar un ancho de banda descendente de por lo menos 2 Mbps a cada casa en cualquier momento. Describa lo que necesita hacer la compañía de cable para proporcionar esta garantía.
57. Tomando en cuenta la asignación espectral mostrada en la figura 2-48 y la información dada en el texto, ¿cuántos Mbps necesita asignar el sistema por cable al flujo ascendente y cuántos al flujo descendente?
58. ¿Qué tan rápido un usuario de cable puede recibir datos si la red está inactiva?
59. Multiplexar flujos de datos múltiples STS-1, llamados tributarias, tiene un papel importante en SONET. Un multiplexor 3:1 multiplexa tres tributarias STS-1 de entrada en un flujo STS-3 de salida. Esta multiplexión se realiza byte por byte, es decir, los tres primeros bytes de salida son los primeros bytes de las tributarias 1, 2 y 3, respectivamente. Los siguientes tres bytes de salida son los segundos bytes

de las tributarias 1, 2 y 3, respectivamente, etcétera. Escriba un programa que simule este multiplexor 3:1. El programa deberá consistir de cinco procesos. El proceso principal crea cuatro procesos, uno para cada una de las tres tributarias STS-1 y uno para el multiplexor. Cada proceso tributario lee una trama STS-1 de un archivo de entrada como una secuencia de 810 bytes. Tales procesos tributarios envían sus tramas (byte por byte) al proceso multiplexor. Éste recibe los bytes y envía una trama STS-3 (byte por byte) escribiéndola en una salida estándar. Utilice canalizaciones para la comunicación entre procesos.

3

LA CAPA DE ENLACE DE DATOS

En este capítulo estudiaremos los principios de diseño de la capa 2, la capa de enlace de datos. Este estudio tiene que ver con los algoritmos para lograr una comunicación confiable y eficiente entre dos máquinas adyacentes en la capa de enlace de datos. Por adyacente, queremos decir que las dos máquinas están conectadas por un canal de comunicaciones que actúa de manera conceptual como un alambre (por ejemplo, un cable coaxial, una línea telefónica o un canal inalámbrico de punto a punto). La propiedad esencial de un canal que lo hace asemejarse a un alambre es que los bits se entregan con exactitud en el mismo orden en que fueron enviados.

A primera vista podría pensarse que este problema es tan trivial que no hay ningún software que estudiar: la máquina *A* sólo pone los bits en el alambre, y la máquina *B* simplemente los toma. Por desgracia, los circuitos de comunicación cometan errores ocasionales. Además, tienen una tasa de datos finita y hay un retardo de propagación diferente de cero entre el momento en que se envía un bit y el momento en que se recibe. Estas limitaciones tienen implicaciones importantes para la eficiencia de la transferencia de datos. Los protocolos usados para comunicaciones deben considerar todos estos factores. Dichos protocolos son el tema de este capítulo.

Tras una introducción a los aspectos clave de diseño presentes en la capa de enlace de datos, comenzaremos nuestro estudio de sus protocolos observando la naturaleza de los errores, sus causas y la manera en que se pueden detectar y corregir. Después estudiaremos una serie de protocolos de complejidad creciente, cada uno de los cuales resuelve los problemas presentes en esta capa. Por último, concluiremos con un estudio del modelado y la corrección de los protocolos y daremos algunos ejemplos de protocolos de enlace de datos.

3.1 CUESTIONES DE DISEÑO DE LA CAPA DE ENLACE DE DATOS

La capa de enlace de datos tiene que desempeñar varias funciones específicas, entre las que se incluyen:

1. Proporcionar una interfaz de servicio bien definida con la capa de red.
2. Manejar los errores de transmisión.
3. Regular el flujo de datos para que receptores lentos no sean saturados por emisores rápidos.

Para cumplir con estas metas, la capa de enlace de datos toma de la capa de red los paquetes y los encapsula en **tramas** para transmitirlos. Cada trama contiene un encabezado, un campo de carga útil (*payload*) para almacenar el paquete y un terminador o final, como se ilustra en la figura 3-1. El manejo de las tramas es la tarea primordial de la capa de enlace de datos. En las siguientes secciones examinaremos en detalle todos los aspectos mencionados.

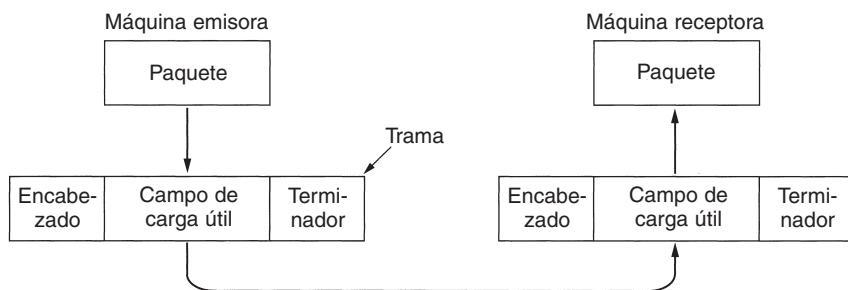


Figura 3-1. Relación entre los paquetes y las tramas.

Aunque este capítulo sólo analiza la capa de enlace de datos y los protocolos de enlace de datos, muchos de los principios que analizaremos aquí, como el control de errores y el de flujo, también se encuentran en la capa de transporte y en otros protocolos. De hecho, en muchas redes, estas funciones se encuentran sólo en las capas superiores y no en la de enlace de datos. Sin embargo, independientemente de donde se encuentren, los principios son casi los mismos, por lo que en realidad no importa en qué parte del libro los analicemos. Por lo general, éstos se muestran en la capa de enlace de datos en sus formas más simples y puras, por lo que dicha capa es un buen lugar para examinarlos en detalle.

3.1.1 Servicios proporcionados a la capa de red

La función de la capa de enlace de datos es suministrar servicios a la capa de red. El servicio principal es transferir datos de la capa de red en la máquina de origen a la capa de red en la máquina de destino. En la capa de red de la máquina de origen hay una entidad, llamada proceso, que entrega algunos bits a la capa de enlace de datos para transmitirlos a la máquina de destino. El tra-

bajo de la capa de enlace de datos es transmitir los bits a la máquina de destino, para que puedan ser entregados a su capa de red, como se muestra en la figura 3-2(a). La transmisión real sigue la trayectoria de la figura 3-2(b), pero es más fácil pensar en términos de dos procesos de capa de enlace de datos que se comunican usando un protocolo de enlace de datos. Por esta razón, a lo largo de este capítulo usaremos de manera implícita el modelo de la figura 3-2(a).

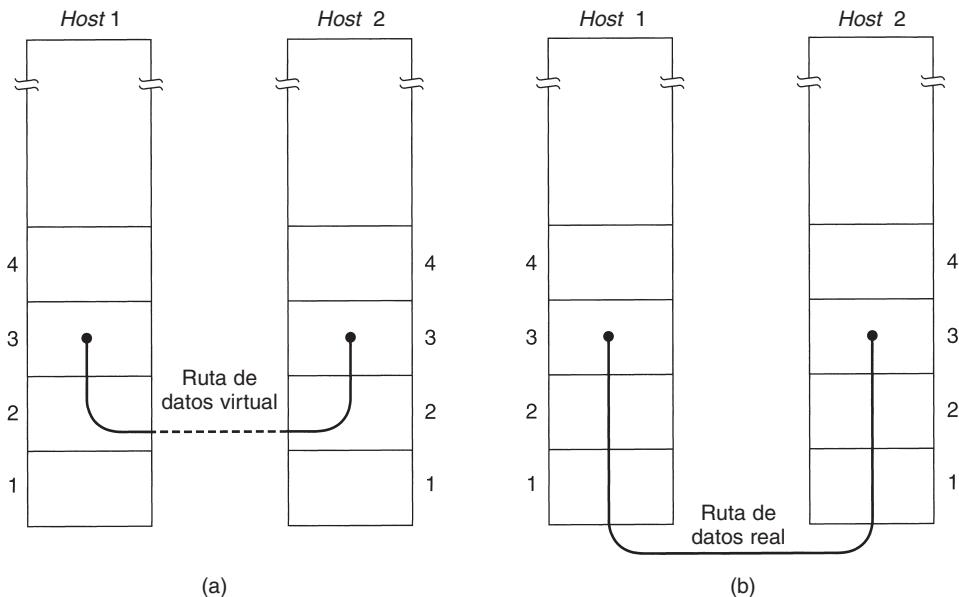


Figura 3-2. (a) Comunicación virtual. (b) Comunicación real.

La capa de enlace de datos puede diseñarse para ofrecer varios servicios. Los servicios reales ofrecidos pueden variar de sistema a sistema. Tres posibilidades razonables que normalmente se proporcionan son:

1. Servicio no orientado a la conexión sin confirmación de recepción.
2. Servicio no orientado a la conexión con confirmación de recepción.
3. Servicio orientado a la conexión con confirmación de recepción.

Consideremos cada uno de ellos por separado.

El servicio no orientado a la conexión sin confirmación de recepción consiste en hacer que la máquina de origen envíe tramas independientes a la máquina de destino sin pedir que ésta confirme la recepción. No se establece conexión de antemano ni se libera después. Si se pierde una trama debido a ruido en la línea, en la capa de enlace de datos no se realiza ningún intento por detectar la pérdida ni por recuperarse de ella. Esta clase de servicio es apropiada cuando la tasa de errores es muy baja, por lo que la recuperación se deja a las capas superiores. También es apropiada para el tráfico en tiempo real, por ejemplo de voz, en el que la llegada retrasada de datos es peor que

los errores de datos. La mayoría de las LANs utilizan servicios no orientados a la conexión sin confirmación de recepción en la capa de enlace de datos.

El siguiente paso hacia adelante en cuanto a confiabilidad es el servicio no orientado a la conexión con confirmación de recepción. Cuando se ofrece este servicio tampoco se utilizan conexiones lógicas, pero se confirma de manera individual la recepción de cada trama enviada. De esta manera, el emisor sabe si la trama ha llegado bien o no. Si no ha llegado en un tiempo especificado, puede enviarse nuevamente. Este servicio es útil en canales inestables, como los de los sistemas inalámbricos.

Tal vez valga la pena poner énfasis en que proporcionar confirmaciones de recepción en la capa de enlace de datos sólo es una optimización, nunca un requisito. La capa de red siempre puede enviar un paquete y esperar que se confirme su recepción. Si la confirmación no llega antes de que expire el temporizador, el emisor puede volver a enviar el mensaje. El problema con esta estrategia es que las tramas tienen una longitud máxima impuesta por el hardware mientras que los paquetes de la capa de red no la tienen. Si el paquete promedio se divide en, digamos, 10 tramas, y se pierde 20% de todas las tramas enviadas, el paquete puede tardar mucho tiempo en pasar. Si las tramas se confirman y retransmiten de manera individual, los paquetes completos pasan con mayor rapidez. En los canales confiables, como la fibra óptica, la sobrecarga que implica el uso de un protocolo de enlace de datos muy robusto puede ser innecesaria, pero en canales inalámbricos bien vale la pena el costo debido a su inestabilidad inherente.

Regresando a nuestros servicios, el servicio más refinado que puede proporcionar la capa de enlace de datos a la capa de red es el servicio orientado a la conexión. Con este servicio, las máquinas de origen y de destino establecen una conexión antes de transferir datos. Cada trama enviada a través de la conexión está numerada, y la capa de enlace de datos garantiza que cada trama enviada llegará a su destino. Es más, garantiza que cada trama será recibida exactamente una vez y que todas las tramas se recibirán en el orden adecuado. En contraste, con el servicio no orientado a la conexión es posible que una confirmación de recepción perdida cause que una trama se envíe varias veces y, por lo tanto, que se reciba varias veces. Por su parte, el servicio orientado a la conexión proporciona a los procesos de la capa de red el equivalente de un flujo de bits confiable.

Cuando se utiliza un servicio orientado a la conexión, las transferencias tienen tres fases distintas. En la primera, la conexión se establece haciendo que ambos lados inicialicen las variables y los contadores necesarios para seguir la pista de las tramas que han sido recibidas y las que no. En la segunda fase se transmiten una o más tramas. En la tercera fase, la conexión se cierra y libera las variables, los búferes y otros recursos utilizados para mantener la conexión.

Considere un ejemplo típico: una subred de WAN que consiste en enrutadores conectados por medio de líneas telefónicas alquiladas de punto a punto. Cuando llega una trama a un enrutador, el hardware la examina para verificar si está libre de errores (mediante una técnica que veremos más adelante en este capítulo), y después la pasa al software de la capa de enlace de datos (que podría estar integrado en un *chip* de la tarjeta de interfaz de red). Dicho software comprueba si ésta es la trama esperada y, de ser así, entrega el paquete contenido en el campo de carga útil al software de enrutamiento. A continuación, este software elige la línea de salida adecuada y reenvía el paquete al software de la capa de enlace de datos, que luego lo transmite. En la figura 3-3 se muestra el flujo a través de dos enrutadores.

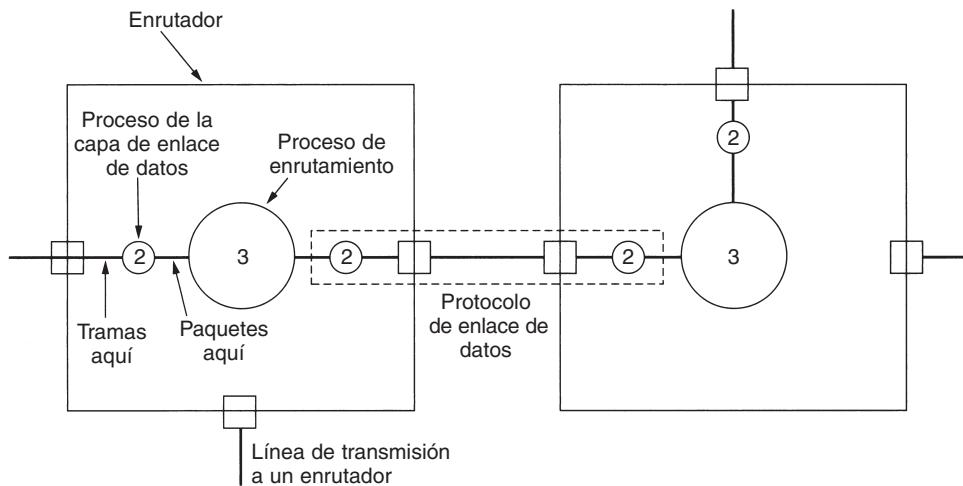


Figura 3-3. Ubicación del protocolo de enlace de datos.

El código de enrutamiento con frecuencia requiere que el trabajo se haga bien, es decir, que haya conexiones estables y ordenadas en cada una de las líneas punto a punto. No quiere que se le moleste frecuentemente con paquetes que se perdieron en el camino. Es responsabilidad del protocolo de enlace de datos, mostrado en el rectángulo punteado, hacer que las líneas de comunicación no estables parezcan perfectas o, cuando menos, bastante buenas. Como información adicional, aunque hemos mostrado múltiples copias del software de la capa de enlace de datos en cada enrutador, de hecho una sola copia maneja todas las líneas, con diferentes tablas y estructuras de datos para cada una.

3.1.2 Entramado

A fin de proporcionar servicios a la capa de red, la de enlace de datos debe utilizar los servicios que la capa física le proporciona. Lo que hace la capa física es aceptar un flujo de bits puros e intentar entregarlo al destino. No se garantiza que este flujo de bits esté libre de errores. La cantidad de bits recibidos puede ser menor, igual o mayor que la cantidad de bits transmitidos, y éstos pueden tener diferentes valores. Es responsabilidad de la capa de enlace de datos detectar y, de ser necesario, corregir los errores.

El método común es que la capa de enlace de datos divida el flujo de bits en tramas separadas y que calcule la suma de verificación de cada trama. (Posteriormente en este capítulo se analizarán los algoritmos de suma de verificación.) Cuando una trama llega al destino, se recalcula la suma de verificación. Si la nueva suma de verificación calculada es distinta de la contenida en la trama, la capa de enlace de datos sabe que ha ocurrido un error y toma medidas para manejarlo (por ejemplo, descartando la trama mala y, posiblemente, regresando un informe de error).

La división en tramas del flujo de bits es más difícil de lo que parece a primera vista. Una manera de lograr esta división en tramas es introducir intervalos de tiempo entre las tramas, de la misma manera que los espacios entre las palabras en el texto común. Sin embargo, las redes pocas veces ofrecen garantías sobre la temporización, por lo que es posible que estos intervalos sean eliminados o que puedan introducirse otros intervalos durante la transmisión.

Puesto que es demasiado riesgoso depender de la temporización para marcar el inicio y el final de cada trama, se han diseñado otros métodos. En esta sección veremos cuatro métodos:

1. Conteo de caracteres.
2. Banderas, con relleno de caracteres.
3. Banderas de inicio y fin, con relleno de bits.
4. Violaciones de codificación de la capa física.

El primer método de entramado se vale de un campo en el encabezado para especificar el número de caracteres en la trama. Cuando la capa de enlace de datos del destino ve la cuenta de caracteres, sabe cuántos caracteres siguen y, por lo tanto, dónde está el fin de la trama. Esta técnica se muestra en la figura 3-4(a) para cuatro tramas de 5, 5, 8 y 8 caracteres de longitud, respectivamente.

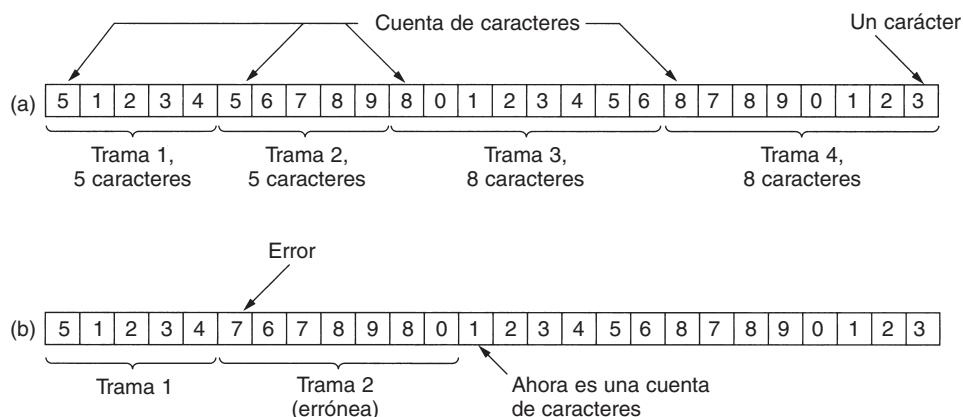


Figura 3-4. Un flujo de caracteres. (a) Sin errores. (b) Con un error.

El problema con este algoritmo es que la cuenta puede alterarse por un error de transmisión. Por ejemplo, si la cuenta de caracteres de 5 en la segunda trama de la figura 3-4(b) se vuelve un 7, el destino perderá la sincronía y será incapaz de localizar el inicio de la siguiente trama. Incluso si el destino sabe que la trama está mal porque la suma de verificación es incorrecta, no tiene forma de saber dónde comienza la siguiente trama. Regresar una trama a la fuente solicitando una retransmisión tampoco ayuda, ya que el destino no sabe cuántos caracteres tiene que saltar para

llegar al inicio de la retransmisión. Por esta razón, en la actualidad casi no se utiliza el método de conteo de caracteres.

El segundo método de entrampado evita el problema de tener que sincronizar nuevamente después de un error, haciendo que cada trama inicie y termine con bytes especiales. En el pasado, los bytes de inicio y final eran diferentes, pero en los años recientes la mayoría de los protocolos han utilizado el mismo byte, llamado **bandera** (o indicador), como delimitador de inicio y final, que en la figura 3-5(a) se muestra como FLAG. De esta manera, si el receptor pierde la sincronía, simplemente puede buscar la bandera para encontrar el final e inicio de la trama actual. Dos banderas consecutivas señalan el final de una trama y el inicio de la siguiente.

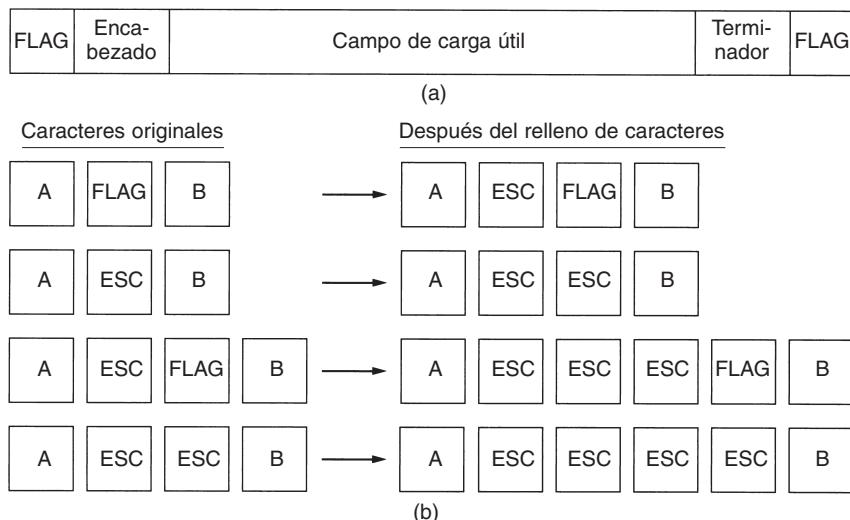


Figura 3-5. (a) Una trama delimitada por banderas. (b) Cuatro ejemplos de secuencias de bytes antes y después del relleno de caracteres.

Cuando se utiliza este método para transmitir datos binarios, como programas objeto o números de punto flotante, surge un problema serio. Se puede dar el caso con mucha facilidad de que el patrón de bits de la bandera aparezca en los datos (*payload*), lo que interferiría en el entrampado. Una forma de resolver este problema es hacer que la capa de enlace de datos del emisor inserte un byte de escape especial (ESC) justo antes de cada bandera “accidental” en los datos. La capa de enlace de datos del lado receptor quita el byte de escape antes de entregar los datos a la capa de red. Esta técnica se llama **relleno de caracteres**. Por lo tanto, una bandera de entrampado se puede distinguir de uno en los datos por la ausencia o presencia de un byte de escape que la antecede.

Por supuesto que surge la pregunta de qué sucede si un byte de escape aparece en medio de los datos. La respuesta es que también se rellena con un byte de escape. Por lo tanto, cualquier byte de escape individual es parte de una secuencia de escape, mientras que uno doble indica que un

escape sencillo apareció de manera natural en los datos. En la figura 3-5(b) se muestran algunos ejemplos. En todos los casos, la secuencia de bytes que se entrega después de la eliminación de los bytes de escape es exactamente la misma que la secuencia de bytes original.

El esquema de relleno de caracteres que se muestra en la figura 3-5 es una ligera simplificación del esquema empleado en el protocolo PPP que la mayoría de las computadoras utiliza para comunicarse con el proveedor de servicios de Internet. Más tarde analizaremos este protocolo.

Una desventaja importante del uso de esta técnica de entramado es que está fuertemente atada a los caracteres de 8 bits. No todos los códigos utilizan caracteres de 8 bits. Por ejemplo, UNICODE utiliza caracteres de 16 bits. A medida que se desarrollaron las redes, las desventajas de incorporar la longitud del código de caracteres en el mecanismo de entramado se volvieron más obvias, por lo que tuvo que desarrollarse una técnica nueva que permitiera caracteres de tamaño arbitrario.

La nueva técnica permite que las tramas de datos contengan un número arbitrario de bits y admite códigos de caracteres con un número arbitrario de bits por carácter. Dicha técnica funciona de la siguiente manera: cada trama comienza y termina con un patrón especial de bits, 01111110 (que es de hecho una bandera). Cada vez que la capa de enlace de datos del emisor encuentra cinco unos consecutivos en los datos, automáticamente inserta un bit 0 en el flujo de bits saliente. Este **relleno de bits** es análogo al relleno de caracteres, en el cual un byte de escape se inserta en el flujo de caracteres saliente antes de un byte igual a la bandera de entramado en los datos.

Cuando el receptor ve cinco bits 1 de entrada consecutivos, seguidos de un bit 0, automáticamente extrae (es decir, borra) el bit 0 de relleno. Así como el relleno de caracteres es completamente transparente para la capa de red en ambas computadoras, también lo es el relleno de bits. Si los datos de usuario contienen el patrón indicador 01111110, éste se transmite como 011111010, pero se almacena en la memoria del receptor como 01111110. En la figura 3-6 se da un ejemplo del relleno de bits.

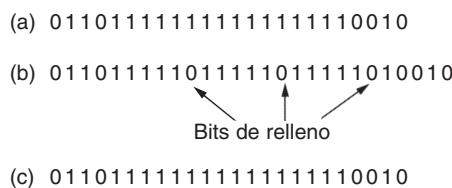


Figura 3-6. Relleno de bits. (a) Los datos originales. (b) Los datos, según aparecen en la línea. (c) Los datos, como se guardan en la memoria del receptor tras eliminar el relleno.

Con el relleno de bits, el límite entre las dos tramas puede ser reconocido sin ambigüedades mediante el patrón de banderas. De esta manera, si el receptor pierde la pista de dónde está, todo lo que tiene que hacer es explorar la entrada en busca de secuencias de banderas, pues sólo pueden ocurrir en los límites de las tramas y nunca en los datos.

El último método de entramado sólo se aplica a las redes en las que la codificación en el medio físico contiene cierta redundancia. Por ejemplo, algunas LANs codifican un bit de datos usando dos bits físicos. Normalmente, un bit 1 es un par alto-bajo, y un bit 0 es un par bajo-alto. El esquema implica que cada bit de datos tiene una transición a medio camino, lo que hace fácil

para el receptor localizar los límites de los bits. Las combinaciones alto-alto y bajo-bajo no se usan para datos, pero en algunos protocolos se utilizan para delimitar tramas.

Como nota final sobre el entrampado, muchos protocolos de enlace de datos usan, por seguridad, una combinación de cuenta de caracteres con uno de los otros métodos. Cuando llega una trama, se usa el campo de cuenta para localizar el final de la trama. Sólo si el delimitador apropiado está presente en esa posición y la suma de verificación es correcta, la trama se acepta como válida. De otra manera, se explora el flujo de entrada en busca del siguiente delimitador.

3.1.3 Control de errores

Una vez resuelto el problema de marcar el inicio y el final de cada trama, llegamos al siguiente problema: cómo asegurar que todas las tramas realmente se entreguen en el orden apropiado a la capa de red del destino. Suponga que el emisor se dedicó a enviar tramas sin importarle si estaban llegando de manera adecuada. Esto podría estar bien para un servicio no orientado a la conexión sin confirmación de recepción, pero no será correcto para un servicio confiable orientado a la conexión.

La manera normal de asegurar la entrega confiable de datos es proporcionar retroalimentación al emisor sobre lo que está ocurriendo en el otro lado de la línea. Por lo general, el protocolo exige que el receptor regrese tramas de control especiales que contengan confirmaciones de recepción positivas o negativas de las tramas que llegan. Si el emisor recibe una confirmación de recepción positiva de una trama, sabe que la trama llegó correctamente. Por otra parte, una confirmación de recepción negativa significa que algo falló y que la trama debe transmitirse otra vez.

Una complicación adicional surge de la posibilidad de que los problemas de hardware causen la desaparición de una trama completa (por ejemplo, por una ráfaga de ruido). En este caso, el receptor no reaccionará en absoluto, ya que no tiene razón para reaccionar. Debe quedar claro que un protocolo en el cual el emisor envía una trama y luego espera una confirmación de recepción, positiva o negativa, se quedaría esperando eternamente si se pierde por completo una trama debido a una falla de hardware.

Esta posibilidad se maneja introduciendo temporizadores en la capa de enlace de datos. Cuando el emisor envía una trama, por lo general también inicia un temporizador. Éste se ajusta de modo que expire cuando haya transcurrido un intervalo suficiente para que la trama llegue a su destino, se procese ahí y la confirmación de recepción se regrese al emisor. Por lo general, la trama se recibirá de manera correcta y la confirmación de recepción llegará antes de que el temporizador expire, en cuyo caso se cancelará.

Sin embargo, si la trama o la confirmación de recepción se pierden, el temporizador expirará, alertando al emisor sobre un problema potencial. La solución obvia es simplemente transmitir de nuevo la trama. Sin embargo, aunque las tramas pueden transmitirse muchas veces, existe el peligro de que el receptor acepte la misma trama dos o más veces y que la pase a la capa de red más de una vez. Para evitar que esto ocurra, generalmente es necesario asignar números de secuencia a las tramas que salen, a fin de que el receptor pueda distinguir las retransmisiones de los originales.

El asunto de la administración de temporizadores y números de secuencia para asegurar que cada trama llegue finalmente a la capa de red en el destino una sola vez, ni más ni menos, es una parte importante de las tareas de la capa de enlace de datos. Posteriormente en este capítulo estudiaremos la manera en que se lleva a cabo esta administración, observando una serie de ejemplos de complejidad creciente.

3.1.4 Control de flujo

Otro tema de diseño importante que se presenta en la capa de enlace de datos (y también en las capas superiores) es qué hacer con un emisor que quiere transmitir tramas de manera sistemática y a mayor velocidad que aquella con que puede aceptarlos el receptor. Esta situación puede ocurrir fácilmente cuando el emisor opera en una computadora rápida (o con baja carga) y el receptor opera en una máquina lenta (o sobrecargada). El emisor envía las tramas a alta velocidad hasta que satura por completo al receptor. Aunque la transmisión esté libre de errores, en cierto punto el receptor simplemente no será capaz de manejar las tramas conforme lleguen y comenzará a perder algunas. Es obvio que tiene que hacerse algo para evitar esta situación.

Por lo general se utilizan dos métodos. En el primero, el **control de flujo basado en retroalimentación**, el receptor regresa información al emisor autorizándolo para enviar más datos o indicándole su estado. En el segundo, el **control de flujo basado en tasa**, el protocolo tiene un mecanismo integrado que limita la tasa a la que el emisor puede transmitir los datos, sin recurrir a retroalimentación por parte del receptor. En este capítulo estudiaremos el método de control de flujo basado en retroalimentación debido a que el método basado en tasa no se utiliza en la capa de enlace de datos. En el capítulo 5 analizaremos el método basado en tasa.

Se conocen varios esquemas de control de flujo basados en retroalimentación, pero la mayoría se fundamenta en el mismo principio. El protocolo contiene reglas bien definidas respecto al momento en que un emisor puede enviar la siguiente trama. Con frecuencia estas reglas prohíben el envío de tramas hasta que el receptor lo autorice, implícita o explícitamente. Por ejemplo, cuando se establece una conexión, el receptor podría decir: “Puedes enviarme n tramas ahora, pero una vez que lo hagas, no envíes nada más hasta que te indique que continúes”. Mas adelante analizaremos los detalles.

3.2 DETECCIÓN Y CORRECCIÓN DE ERRORES

Como vimos en el capítulo 2, el sistema telefónico tiene tres partes: los commutadores, las troncales interoficinas y los circuitos locales. Las primeras dos son ahora casi enteramente digitales en la mayoría de los países desarrollados. Los circuitos locales aún son cables de par trenzado de cobre analógicos en todos lados y continuarán así durante décadas debido al enorme costo de su reemplazo. Aunque los errores son raros en la parte digital, aún son comunes en los circuitos locales. Además, la comunicación inalámbrica se está volviendo más común, y las tasas de errores son de magnitud mucho mayor que en las troncales de fibra interoficinas. La conclusión es: los errores de transmisión van a ser inevitables durante muchos años más. Tendremos que aprender a lidiar con ellos.

Como resultado de los procesos físicos que los generan, los errores en algunos medios (por ejemplo, la radio) tienden a aparecer en ráfagas y no de manera individual. El hecho de que los errores lleguen en ráfaga tiene ventajas y desventajas con respecto a los errores aislados de un solo bit. Por el lado de las ventajas, los datos de computadora siempre se envían en bloques de bits. Suponga que el tamaño de bloque es de 1000 bits y la tasa de errores es de 0.001 por bit. Si los errores fueran independientes, la mayoría de los bloques contendría un error. Sin embargo, si los errores llegan en ráfagas de 100, en promedio sólo uno o dos bloques de cada 100 serán afectados. La desventaja de los errores en ráfaga es que son mucho más difíciles de detectar y corregir que los errores aislados.

3.2.1 Códigos de corrección de errores

Los diseñadores de redes han desarrollado dos estrategias principales para manejar los errores. Una es incluir suficiente información redundante en cada bloque de datos transmitido para que el receptor pueda deducir lo que debió ser el carácter transmitido. La otra estrategia es incluir sólo suficiente redundancia para permitir que el receptor sepa que ha ocurrido un error (pero no qué error) y entonces solicite una retransmisión. La primera estrategia utiliza **códigos de corrección de errores**; la segunda usa **códigos de detección de errores**. El uso de códigos de corrección de errores usualmente se conoce como **corrección de errores hacia adelante**.

Cada una de estas técnicas ocupa un nicho ecológico diferente. En los canales que son altamente confiables, como los de fibra, es más económico utilizar un código de detección de errores y simplemente retransmitir los bloques defectuosos que surgen ocasionalmente. Sin embargo, en los canales que causan muchos errores, como los enlaces inalámbricos, es mejor agregar la redundancia suficiente a cada bloque para que el receptor pueda descubrir cuál era el bloque original transmitido, en lugar de confiar en una retransmisión, que también podría tener errores.

Para entender la manera en que pueden manejarse los errores, es necesario estudiar de cerca lo que es en realidad un error. Por lo general, una trama consiste en m bits de datos (es decir, de mensaje) y r bits redundantes o de verificación. Sea la longitud total n (es decir, $n = m + r$). A una unidad de n bits que contiene datos y bits de verificación se le conoce como **palabra codificada** de n bits.

Dadas dos palabras codificadas cualesquiera, digamos 10001001 y 10110001, es posible determinar cuántos bits correspondientes difieren. En este caso, difieren tres bits. Para determinar la cantidad de bits diferentes, basta aplicar un OR exclusivo a las dos palabras codificadas y contar la cantidad de bits 1 en el resultado, por ejemplo:

$$\begin{array}{r} 10001001 \\ 10110001 \\ \hline 00111000 \end{array}$$

La cantidad de posiciones de bits en la que difieren dos palabras codificadas se llama **distan-
cia de Hamming** (Hamming, 1950). Su significado es que, si dos palabras codificadas están separadas una distancia de Hamming d , se requerirán d errores de un bit para convertir una en la otra.

En la mayoría de las aplicaciones de transmisión de datos, todos los 2^m mensajes de datos posibles son legales, pero debido a la manera en que se calculan los bits de verificación no se usan todas las 2^n palabras codificadas posibles. Dado el algoritmo de cálculo de los bits de verificación, es posible construir una lista completa de palabras codificadas legales y encontrar, en esta lista, las dos palabras codificadas cuya distancia de Hamming es mínima. Ésta es la distancia de Hamming de todo el código.

Las propiedades de detección y corrección de errores de un código dependen de su distancia de Hamming. Para detectar d errores se necesita un código con distancia $d + 1$, pues con tal código no hay manera de que d errores de un bit puedan cambiar una palabra codificada válida a otra. Cuando el receptor ve una palabra codificada no válida, sabe que ha ocurrido un error de transmisión. De manera similar, para corregir d errores se necesita un código de distancia $2d + 1$, pues así las palabras codificadas legales están tan separadas que, aun con d cambios, la palabra codificada original sigue estando más cercana que cualquier otra palabra codificada, por lo que puede determinarse de manera única.

Como ejemplo sencillo de código de detección de errores, considere un código en el que se agrega un solo **bit de paridad** a los datos. Este bit se escoge de manera que la cantidad de bits 1 en la palabra código sea par (o impar). Por ejemplo, cuando se envía 1011010 con paridad par, se agrega un bit al final, y se vuelve 10110100. Con paridad impar, 1011010 se vuelve 10110101. Un código con un solo bit de paridad tiene una distancia de 2, pues cualquier error de un bit produce una palabra codificada con la paridad equivocada. Este sistema puede usarse para detectar errores individuales.

Como ejemplo sencillo de código de corrección de errores, considere un código con sólo cuatro palabras codificadas válidas:

0000000000, 0000011111, 1111100000 y 1111111111

Este código tiene una distancia de 5, lo que significa que puede corregir errores dobles. Si llega la palabra codificada 0000000111, el receptor sabe que el original debió ser 0000011111. Sin embargo, si un error triple cambia 0000000000 a 0000000111, el error no se corregirá de manera adecuada.

Imagine que deseamos diseñar un código con m bits de mensaje y r bits de verificación que permitirá la corrección de todos los errores individuales. Cada uno de los 2^m mensajes legales tiene n palabras codificadas ilegales a una distancia 1 de él. Éstas se forman invirtiendo en forma sistemática cada uno de los n bits de la palabra codificada de n bits que la forman. Por lo tanto, cada uno de los 2^m mensajes legales requiere $n + 1$ patrones de bits dedicados a él. Dado que la cantidad de patrones de bits es 2^n , debemos tener $(n + 1)2^m \leq 2^n$. Usando $n = m + r$, este requisito se vuelve $(m + r + 1) \leq 2^r$. Dado m , esto impone un límite inferior a la cantidad de bits de verificación necesarios para corregir errores individuales.

De hecho, este límite inferior teórico puede lograrse usando un método gracias a Hamming (1950). Los bits de la palabra codificada se numeran en forma consecutiva, comenzando por el bit 1 a la izquierda, el bit 2 a su derecha inmediata, etcétera. Los bits que son potencias de 2 (1, 2, 4, 8, 16, etcétera) son bits de verificación. El resto (3, 5, 6, 7, 9, etcétera) se rellenan con los m bits de datos. Cada bit de verificación obliga a que la paridad de un grupo de bits, incluyéndolo a él

mismo, sea par (o impar). Un bit puede estar incluido en varios cálculos de paridad. Para ver a qué bits de verificación contribuye el bit de datos en la posición k , reescriba k como una suma de potencias de 2. Por ejemplo, $11 = 1 + 2 + 8$ y $29 = 1 + 4 + 8 + 16$. Se comprueba un bit solamente por los bits de verificación que ocurren en su expansión (por ejemplo, el bit 11 es comprobado por los bits 1, 2 y 8).

Cuando llega una palabra codificada, el receptor inicializa a cero un contador y luego examina cada bit de verificación, k ($k = 1, 2, 4, 8, \dots$), para ver si tiene la paridad correcta. Si no, suma k al contador. Si el contador es igual a cero tras haber examinado todos los bits de verificación (es decir, si todos fueron correctos), la palabra codificada se acepta como válida. Si el contador es diferente de cero, contiene el número del bit incorrecto. Por ejemplo, si los bits de verificación 1, 2 y 8 tienen errores, el bit invertido es el 11, pues es el único comprobado por los bits 1, 2 y 8. En la figura 3-7 se muestran algunos caracteres ASCII de 7 bits codificados como palabras codificadas de 11 bits usando un código de Hamming. Recuerde que los datos se encuentran en las posiciones de bit 3, 5, 6, 7, 9, 10 y 11.

Carácter	ASCII	Bits de verificación
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	01111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	10101011111
d	1100100	11111001100
e	1100101	00111000101

↓
Orden de transmisión de bits

Figura 3-7. Uso de un código de Hamming para corregir errores en ráfaga.

Los códigos de Hamming sólo pueden corregir errores individuales. Sin embargo, hay un truco que puede servir para que los códigos de Hamming corrijan errores de ráfaga. Se dispone como matriz una secuencia de k palabras codificadas consecutivas, con una palabra codificada por fila. Normalmente se transmitiría una palabra codificada a la vez, de izquierda a derecha. Para corregir los errores en ráfaga, los datos deben transmitirse una columna a la vez, comenzando por la columna del extremo izquierdo. Cuando todos los bits k han sido enviados, se envía la segunda columna, y así sucesivamente. Cuando la trama llega al receptor, la matriz se reconstruye, una columna a la vez. Si ocurre un error en ráfaga de longitud k , cuando mucho se habrá afectado 1 bit de cada una de las k palabras codificadas; sin embargo, el código de Hamming puede corregir un error por palabra codificada, así que puede restaurarse la totalidad del bloque. Este método usa kr bits de verificación para inmunizar bloques de km bits de datos contra un solo error en ráfaga de longitud k o menos.

3.2.2 Códigos de detección de errores

Los códigos de corrección de errores se utilizan de manera amplia en los enlaces inalámbricos, que son notoriamente más ruidosos y propensos a errores que el alambre de cobre o la fibra óptica. Sin los códigos de corrección de errores sería difícil pasar cualquier cosa. Sin embargo, a través del cable de cobre o de la fibra óptica, la tasa de error es mucho más baja, por lo que la detección de errores y la retransmisión por lo general son más eficientes ahí para manejar un error ocasional.

Como un ejemplo simple, considere un canal en el que los errores son aislados y la tasa de errores es de 10^{-6} por bit. Sea el tamaño de bloque 1000 bits. Para proporcionar corrección de errores en bloques de 1000 bits se requieren 10 bits de verificación; un megabit de datos requerirá 10000 bits de verificación. Para detectar un solo bloque con 1 bit de error, basta con un bit de paridad por bloque. Por cada 1000 bloques se tendrá que transmitir un bloque extra (1001 bits). La sobrecarga total del método de detección de errores + retransmisión es de sólo 2001 bits por megabit de datos, contra 10,000 bits con un código de Hamming.

Si se agrega un solo bit de paridad a un bloque y el bloque viene muy alterado por una ráfaga de errores prolongada, la probabilidad de que se detecte el error es de 0.5, lo que difícilmente es aceptable. Es posible aumentar la probabilidad considerando a cada bloque por enviar como una matriz rectangular de n bits de ancho y k bits de alto, como se describió anteriormente. Se calcula por separado un bit de paridad para cada columna y se agrega a la matriz como última fila. La matriz se transmite entonces fila por fila. Cuando llega el bloque, el receptor comprueba todos los bits de paridad. Si cualquiera de ellos está mal, solicita la retransmisión del bloque. Se solicitan retransmisiones adicionales hasta que un bloque entero se reciba sin ningún error de paridad.

Este método puede detectar una sola ráfaga de longitud n , pues sólo se cambiará un bit por columna. Sin embargo, una ráfaga de longitud $n + 1$ pasará sin ser detectada si se invierten el primero y último bits, y si todos los demás bits están correctos. (Una ráfaga de errores no implica que todos los bits estén mal; sólo implica que cuando menos el primero y el último están mal.) Si el bloque está muy alterado por una ráfaga continua o por múltiples ráfagas más cortas, la probabilidad de que cualquiera de las n columnas tenga, por accidente, la paridad correcta es de 0.5, por lo que la probabilidad de aceptar un bloque alterado cuando no se debe es de 2^{-n} .

Aunque en algunos casos el método anterior puede ser adecuado, en la práctica se usa uno muy definido: el **código polinomial** (también conocido como **código de redundancia cíclica** o **código CRC**). Los códigos polinomiales se basan en el tratamiento de cadenas de bits como representaciones de polinomios con coeficientes de 0 y 1 solamente. Una trama de k bits se considera como la lista de coeficientes de un polinomio con k términos que van de x^{k-1} a x^0 . Se dice que tal polinomio es de grado $k - 1$. El bit de orden mayor (que se encuentra más a la izquierda) es el coeficiente de x^{k-1} , el siguiente bit es el coeficiente de x^{k-2} y así sucesivamente. Por ejemplo, 110001 tiene 6 bits y, por lo tanto, representa un polinomio de seis términos con coeficientes 1, 1, 0, 0, 0 y 1: $x^5 + x^4 + x^0$.

La aritmética polinomial se hace mediante una operación módulo 2, de acuerdo con las reglas de la teoría de campos algebraicos. No hay acarreos para la suma, ni préstamos para la resta. Tanto la suma como la resta son idénticas a un OR exclusivo. Por ejemplo:

$$\begin{array}{r}
 10011011 \\
 +11001010 \\
 \hline
 01010001
 \end{array}
 \quad
 \begin{array}{r}
 00110011 \\
 +11001101 \\
 \hline
 11111110
 \end{array}
 \quad
 \begin{array}{r}
 11110000 \\
 -10100110 \\
 \hline
 01010110
 \end{array}
 \quad
 \begin{array}{r}
 01010101 \\
 -10101111 \\
 \hline
 11111010
 \end{array}$$

La división se lleva a cabo de la misma manera que en binario, excepto que la resta es módulo 2, igual que antes. Se dice que un divisor “cabe” en un dividendo si éste tiene tantos bits como el divisor.

Cuando se emplea el método de código polinomial, el emisor y el receptor deben acordar por adelantado un **polinomio generador**, $G(x)$. Tanto los bits de orden mayor y menor del generador deben ser 1. Para calcular la **suma de verificación** para una trama con m bits, correspondiente al polinomio $M(x)$, la trama debe ser más larga que el polinomio generador. La idea es incluir una suma de verificación al final de la trama de tal manera que el polinomio representado por la trama con suma de verificación sea divisible entre $G(x)$. Cuando el receptor recibe la trama con suma de verificación, intenta dividirla entre $G(x)$. Si hay un residuo, ha habido un error de transmisión.

El algoritmo para calcular la suma de verificación es el siguiente:

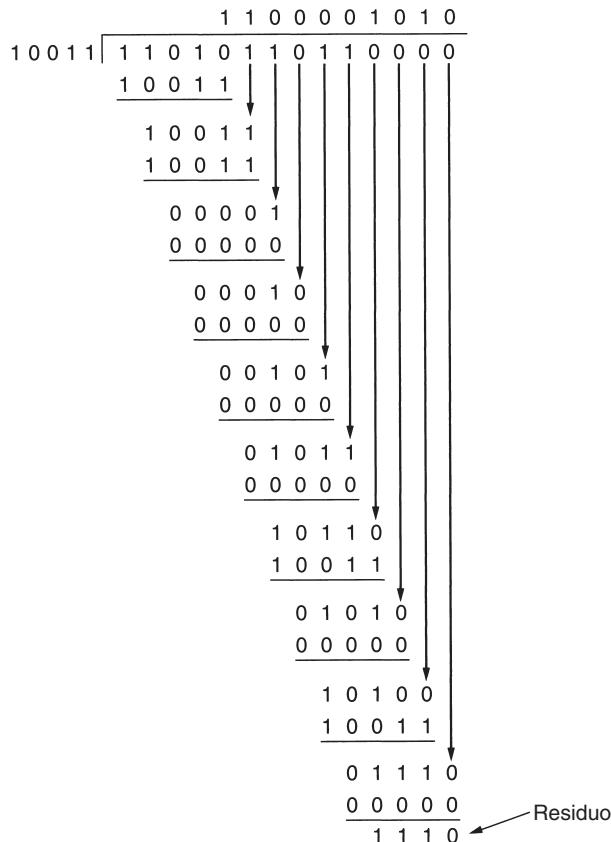
1. Sea r el grado de $G(x)$. Anexe r bits cero al final de la trama, para que ahora contenga $m + r$ bits y corresponda al polinomio $x^r M(x)$.
2. Divida la cadena de bits correspondiente a $G(x)$ entre la correspondiente a $x^r M(x)$ usando una división módulo 2.
3. Reste el residuo (que siempre es de r o menos bits) a la cadena de bits correspondiente a $x^r M(x)$ usando una resta módulo 2. El resultado es la trama con suma de verificación que va a transmitirse. Llame a su polinomio $T(x)$.

En la figura 3-8 se ilustra el cálculo para una trama 1101011011 utilizando el generador $G(x) = x^4 + x + 1$.

Debe quedar claro que $T(x)$ es divisible (módulo 2) entre $G(x)$. En cualquier problema de división, si se resta el residuo del dividendo, lo que queda es divisible entre el divisor. Por ejemplo, en base 10, si se divide 210,278 entre 10,941, el residuo es 2399. Si se resta 2399 a 210,278, lo que queda (207,879) es divisible entre 10,941.

Ahora analizaremos el alcance de este método. ¿Qué tipos de error se detectarán? Imagine que ocurre un error de transmisión tal que en lugar de que llegue la cadena de bits para $T(x)$, llega $T(x) + E(x)$. Cada bit 1 en $E(x)$ corresponde a un bit que ha sido invertido. Si hay k bits 1 en $E(x)$, han ocurrido k errores de un solo bit. Una ráfaga de errores individual se caracteriza por un 1 inicial, una mezcla de ceros y unos, y un 1 final, siendo los demás bits 0.

Trama : 1 1 0 1 0 1 1 0 1 1
 Generador: 1 0 0 1 1
 Mensaje tras anexar 4 bits cero: 1 1 0 1 0 1 1 0 1 1 0 0 0 0



Trama transmitida: 1 1 0 1 0 1 1 0 1 1 1 1 0

Figura 3-8. Cálculo de la suma de verificación de código polinomial.

Al recibir la trama con suma de verificación, el receptor la divide entre $G(x)$; es decir, calcula $[T(x) + E(x)]/G(x)$. $T(x)/G(x)$ es 0, por lo que el resultado del cálculo es simplemente $E(x)/G(x)$. No se detectarán los errores que por casualidad correspondan a polinomios que contengan $G(x)$ como factor; todos los demás errores serán atrapados.

Si ha ocurrido un error de un solo bit, $E(x) = x^i$, donde i determina qué bit es erróneo. Si $G(x)$ contiene dos o más términos, nunca será divisor exacto de $E(x)$, por lo que se detectarán los errores de un solo bit.

Si han ocurrido dos errores de un solo bit aislados, $E(x) = x^i + x^j$, donde $i > j$. Esto también se puede escribir como $E(x) = x^j(x^{i-j} + 1)$. Si suponemos que $G(x)$ no es divisible entre x , una condición suficiente para detectar todos los errores dobles es que $G(x)$ no divida a $x^k + 1$ para ninguna k hasta el valor máximo de $i - j$ (es decir, hasta la longitud máxima de la trama). Se conocen polinomios sencillos de bajo grado que dan protección a tramas largas. Por ejemplo, $x^{15} + x^{14} + 1$ no será divisor exacto de $x^k + 1$ para ningún valor de k menor que 32,768.

Si hay una cantidad impar de bits con error, $E(x)$ contiene un número impar de términos (por ejemplo, $x^5 + x^2 + 1$, pero no $x^2 + 1$). Curiosamente, ningún polinomio con un número impar de términos posee a $x + 1$ como un factor en el sistema de módulo 2. Haciendo $x + 1$ un factor de $G(x)$, podemos atrapar todos los errores consistentes en un número impar de bits invertidos.

Para comprobar que ningún polinomio con una cantidad impar de términos es divisible entre $x + 1$, suponga que $E(x)$ tiene un número impar de términos y que es divisible entre $x + 1$. Factorice $E(x)$ en $(x + 1)Q(x)$. Ahora evalúe $E(1) = (1 + 1)Q(1)$. Dado que $1 + 1 = 0$ (módulo 2), $E(1)$ debe ser cero. Si $E(x)$ tiene un número impar de términos, la sustitución de 1 por x en cualquier lugar siempre dará como resultado un 1. Por lo tanto, ningún polinomio con un número impar de términos es divisible entre $x + 1$.

Por último, y lo que es más importante, un código polinomial con r bits de verificación detectará todos los errores en ráfaga de longitud $\leq r$. Un error en ráfaga de longitud k puede representarse mediante $x^i(x^{k-1} + \dots + 1)$, donde i determina la distancia a la que se encuentra la ráfaga desde el extremo derecho de la trama recibida. Si $G(x)$ contiene un término x^0 , no tendrá x^i como factor, por lo que, si el grado de la expresión entre paréntesis es menor que el grado de $G(x)$, el residuo nunca puede ser cero.

Si la longitud de la ráfaga es de $r + 1$, el residuo de la división entre $G(x)$ será cero si, y sólo si, la ráfaga es idéntica a $G(x)$. Por la definición de ráfaga, el primero y el último bit deben ser 1, así que el que sean iguales o no depende de los $r - 1$ bits intermedios. Si se consideran igualmente probables todas las combinaciones, la probabilidad de que se acepte como válida tal trama incorrecta es de $1/2^{r-1}$.

También puede demostrarse que cuando ocurre una ráfaga de errores mayor que $r + 1$, o cuando ocurren varias ráfagas más cortas, la probabilidad de que una trama incorrecta no sea detectada es de $1/2^r$, suponiendo que todos los patrones de bits sean igualmente probables.

Ciertos polinomios se han vuelto estándares internacionales. El que se utiliza en el IEEE 802 es:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x^1 + 1$$

Entre otras propiedades deseables, tiene la de que detecta todas las ráfagas con una longitud de 32 o menor y todas las ráfagas que afecten un número impar de bits.

Aunque el cálculo requerido para obtener la suma de verificación puede parecer complicado, Peterson y Brown (1961) han demostrado que puede construirse un circuito sencillo con un registro de desplazamiento para calcular y comprobar las sumas de verificación por hardware. En la práctica, casi siempre se usa este hardware. La mayoría de las LANs lo utilizan y, en algunos casos, también lo hacen las líneas punto a punto.

Durante décadas se ha supuesto que las tramas para las que se generan sumas de verificación contienen bits aleatorios. Todos los análisis de algoritmos de suma de verificación se han hecho bajo este supuesto. En fechas más recientes, la inspección de datos reales ha demostrado que este supuesto es equivocado. Como consecuencia, en algunas circunstancias los errores no detectados son mucho más comunes de lo que se pensaba anteriormente (Partridge y cols., 1995).

3.3 PROTOCOLOS ELEMENTALES DE ENLACE DE DATOS

Como introducción al tema de los protocolos, comenzaremos por estudiar tres protocolos de complejidad creciente. Los lectores interesados pueden conseguir un simulador de estos protocolos y otros subsecuentes a través de WWW (vea el prefacio). Antes de estudiar los protocolos, es útil hacer explícitos algunos de los supuestos implícitos del modelo de comunicaciones. Para comenzar, estamos suponiendo que en las capas física, de enlace de datos y de red hay procesos independientes que se comunican pasando mensajes de un lado a otro. En muchos casos, los procesos de las capas física y de enlace de datos se ejecutan en un procesador dentro de un chip especial de E/S y los de la capa de red lo hacen en la CPU principal. Sin embargo, también puede haber otras implementaciones (por ejemplo, tres procesos en un solo chip de E/S o las capas física y de enlace de datos como procedimientos invocados por el proceso de la capa de red). En cualquier caso, el hecho de tratar las tres capas como procesos independientes hace más nítido el análisis en el terreno conceptual y también sirve para subrayar la independencia de las capas.

Otro supuesto clave es que la máquina *A* desea mandar un flujo considerable de datos a la máquina *B* usando un servicio confiable orientado a la conexión. Después consideraremos el caso en que *B* también quiere mandar datos a *A* de manera simultánea. Se ha supuesto que *A* tiene un suministro infinito de datos listos para ser enviados y nunca tiene que esperar a que se produzcan datos. Cuando la capa de enlace de datos de *A* solicita datos, la capa de red siempre es capaz de proporcionarlos de inmediato. (Esta restricción también se desechará posteriormente.)

También supondremos que las máquinas no fallan. Es decir, estos protocolos manejan errores de comunicación, pero no los problemas causados por computadoras que fallan y se reinician.

En lo que concierne a la capa de enlace de datos, el paquete que se le pasa a través de la interfaz desde la capa de red es de datos puros, que deben ser entregados bit por bit a la capa de red del destino. El hecho de que la capa de red del destino pueda interpretar parte del paquete como un encabezado no es de importancia para la capa de enlace de datos.

Cuando la capa de enlace de datos acepta un paquete, lo encapsula en una trama agregándole un encabezado y un terminador de enlace de datos (vea la figura 3-1). Por lo tanto, una trama consiste en un paquete incorporado, cierta información de control (en el encabezado) y una suma de verificación (en el terminador). A continuación la trama se transmite a la capa de enlace de datos de la otra máquina. Supondremos que existen procedimientos de biblioteca adecuados *to_physical_layer* para enviar una trama y *from_physical_layer* para recibir una trama. El hardware emisor calcula y agrega la suma de verificación (y de esta manera crea el terminador) por lo que el software

de la capa de enlace de datos no necesita preocuparse por ella. Por ejemplo, podría utilizarse el algoritmo polinomial analizado antes en este capítulo.

Inicialmente el receptor no tiene nada que hacer. Sólo está esperando que ocurra algo. En los protocolos de ejemplo de este capítulo indicamos que la capa de enlace de datos está en espera de que ocurra algo con la llamada de procedimiento `wait_for_event(&event)`. Este procedimiento sólo regresa cuando ocurre algo (por ejemplo, cuando llega una trama). Al regresar, la variable `event` indica lo que ha ocurrido. El grupo de eventos posibles difiere para cada uno de los diferentes protocolos que describiremos, y se definirán por separado para cada protocolo. Observe que en una situación más realista, la capa de enlace de datos no se quedará en un ciclo cerrado esperando un evento, como hemos sugerido, sino que recibirá una interrupción, la que occasionará que suspenda lo que estaba haciendo y proceda a manejar la trama entrante. Sin embargo, por sencillez ignoraremos todos los detalles de la actividad paralela en la capa de enlace de datos y daremos por hecho que la capa está dedicada de tiempo completo a manejar nuestro canal.

Cuando llega una trama al receptor, el hardware calcula la suma de verificación. Si ésta es incorrecta (es decir, si hubo un error de transmisión), se le informa a la capa de enlace de datos (`event = cksum_err`). Si la trama entrante llega sin daño, también se le informa a la capa de enlace de datos (`event = frame_arrival`) para que pueda adquirir la trama para inspeccionarla usando `from_physical_layer`. Tan pronto como la capa de enlace de datos receptora adquiere una trama sin daños, revisa la información de control del encabezado y, si todo está bien, pasa la parte que corresponde al paquete a la capa de red. En ninguna circunstancia se entrega un encabezado de trama a la capa de red.

Hay una buena razón por la que la capa de red nunca debe recibir ninguna parte del encabezado de trama: para mantener completamente separados el protocolo de red y el de enlace de datos. En tanto la capa de red no sepa nada en absoluto sobre el protocolo de enlace de datos ni el formato de la trama, éstos podrán cambiarse sin requerir cambios en el software de la capa de red. Al proporcionarse una interfaz rígida entre la capa de red y la de enlace de datos se simplifica en gran medida el diseño del software, pues los protocolos de comunicación de las diferentes capas pueden evolucionar en forma independiente.

En la figura 3-9 se muestran algunas declaraciones comunes (en C) para muchos de los protocolos que se analizarán después. Allí se definen cinco estructuras de datos: `boolean`, `seq_nr`, `packet`, `frame_kind` y `frame`. Un `boolean` es un tipo de dato numérico que puede tener los valores `true` y `false`. Un `seq_nr` (número de secuencia) es un entero pequeño que sirve para numerar las tramas, a fin de distinguirlas. Estos números de secuencia van de 0 hasta `MAX_SEQ` (inclusive), que se define en cada protocolo que lo necesita. Un `packet` es la unidad de intercambio de información entre la capa de red y la de enlace de datos en la misma máquina, o entre entidades iguales de la capa de red. En nuestro modelo siempre contiene `MAX_PKT` bytes, pero en la práctica sería de longitud variable.

Un `frame` está compuesto de cuatro campos: `kind`, `Seq`, `ack` e `info`. Los primeros tres contienen información de control y el último puede contener los datos por transferir. Estos campos de control constituyen en conjunto el **encabezado de la trama**.

```

#define MAX_PKT 1024                                /* determina el tamaño del paquete
                                                       en bytes */

typedef enum {false, true} boolean;                /* tipo booleano */
typedef unsigned int seq_nr;                      /* números de secuencia o
                                                       confirmación */

typedef struct {unsigned char data[MAX_PKT];} packet; /* definición de paquete */
typedef enum {data, ack, nak} frame_kind;          /* definición de frame_kind */

typedef struct {                                         /* las tramas se transportan en
   frame_kind kind;                                     esta capa */
   seq_nr seq;                                         /* ¿qué clase de trama es? */
   seq_nr ack;                                         /* número de secuencia */
                                                       /* número de confirmación de
                                                       recepción */
   packet info;                                       /* paquete de la capa de red */
} frame;

/* Espera que ocurra un evento; devuelve el tipo en la variable event. */
void wait_for_event(event_type *event);

/* Obtiene un paquete de la capa de red para transmitirlo por el canal. */
void from_network_layer(packet *p);

/* Entrega información de una trama entrante a la capa de red. */
void to_network_layer(packet *p);

/* Obtiene una trama entrante de la capa física y la copia en r. */
void from_physical_layer(frame *r);

/* Pasa la trama a la capa física para transmitirla. */
void to_physical_layer(frame *s);

/* Arranca el reloj y habilita el evento de expiración de temporizador. */
void start_timer(seq_nr k);

/* Detiene el reloj e inhabilita el evento de expiración de temporizador. */
void stop_timer(seq_nr k);

/* Inicia un temporizador auxiliar y habilita el evento ack_timeout. */
void start_ack_timer(void);

/* Detiene el temporizador auxiliar e inhabilita el evento ack_timeout. */
void stop_ack_timer(void);

/* Permite que la capa de red cause un evento network_layer_ready. */
void enable_network_layer(void);

/* Evita que la capa de red cause un evento network_layer_ready. */
void disable_network_layer(void);

/* La macro inc se expande en línea: incrementa circularmente k. */
#define inc(k) if (k < MAX_SEQ) k = k + 1; else k = 0

```

Figura 3-9. Algunas definiciones necesarias en los protocolos que siguen. Estas definiciones se encuentran en el archivo *protocol.h*.

El campo *kind* indica si hay datos en la trama, porque algunos de los protocolos distinguen entre las tramas que contienen exclusivamente información de control y los que también contienen datos. Los campos *seq* y *ack* se emplean para números de secuencia y confirmaciones de recepción, respectivamente; su uso se describirá posteriormente con mayor detalle. El campo *info* de una trama de datos contiene un solo paquete; el campo *info* de una trama de control no se usa. En una implementación más realista se usaría un campo *info* de longitud variable, omitiéndolo por completo en las tramas de control.

Es importante entender la relación entre un paquete y una trama. La capa de red construye un paquete tomando un mensaje de la capa de transporte y agregándole el encabezado de la capa de red. Este paquete se pasa a la capa de enlace de datos para incluirlo en el campo *info* de una trama saliente. Cuando ésta llega a su destino, la capa de enlace de datos extrae de ella el paquete y a continuación lo pasa a la capa de red. De esta manera, esta capa puede actuar como si las máquinas pudieran intercambiar paquetes directamente.

En la figura 3-9 también se listan varios procedimientos que son rutinas de biblioteca cuyos detalles dependen de la implementación, por lo que no nos ocuparemos de su funcionamiento interno aquí. El procedimiento *wait_for_event* se queda en un ciclo cerrado esperando que algo ocurra, como se mencionó antes. Con los procedimientos *to_network_layer* y *from_network_layer*, la capa de enlace de datos pasa paquetes a la capa de red y acepta paquetes de ella, respectivamente. Observe que *from_physical_layer* y *to_physical_layer* pasan tramas entre la capa de enlace de datos y la capa física, y que los procedimientos *to_network_layer* y *from_network_layer* pasan paquetes entre la capa de enlace de datos y la capa de red. En otras palabras, *to_network_layer* y *from_network_layer* tienen que ver con la interfaz entre las capas 2 y 3, y *from_physical_layer* y *to_physical_layer*, con la interfaz entre las capas 1 y 2.

En la mayoría de los protocolos suponemos un canal inestable que pierde tramas completas ocasionalmente. Para poder recuperarse de tales calamidades, la capa de enlace de datos emisora debe arrancar un temporizador o reloj interno cada vez que envía una trama. Si no obtiene respuesta tras transcurrir cierto intervalo de tiempo predeterminado, el temporizador expira y la capa de enlace de datos recibe una señal de interrupción.

En nuestros protocolos, esto se maneja permitiendo que el procedimiento *wait_for_event* devuelva *event = timeout*. Los procedimientos *start_timer* y *stop_timer* inician y detienen, respectivamente, el temporizador. Las terminaciones del temporizador sólo son posibles cuando éste se encuentra en funcionamiento. Se permite explícitamente llamar a *start_timer* cuando el temporizador está funcionando; tal llamada tan sólo restablece el reloj para hacer que el temporizador termine después de haber transcurrido un intervalo completo de temporización (a menos que se restablezca o apague antes).

Los procedimientos *start_ack_timer* y *stop_ack_timer* controlan un temporizador auxiliar usado para generar confirmaciones de recepción en ciertas condiciones.

Los procedimientos *enable_network_layer* y *disable_network_layer* se usan en los protocolos más complicados, en los que ya no suponemos que la capa de red siempre tiene paquetes que enviar. Cuando la capa de enlace de datos habilita a la capa de red, ésta tiene permitido interrumpir cuando tenga que enviar un paquete. Esto lo indicamos con *event = network_layer_ready*. Cuando una capa de red está inhabilitada, no puede causar tales eventos. Teniendo cuidado respecto a

cuando habilitar e inhabilitar su capa de red, la capa de enlace de datos puede evitar que la capa de red la sature con paquetes para los que no tiene espacio de búfer.

Los números de secuencia de las tramas siempre están en el intervalo de 0 a MAX_SEQ (inclusive), donde MAX_SEQ es diferente para los distintos protocolos. Con frecuencia es necesario avanzar circularmente en 1 un número de secuencia (por ejemplo, MAX_SEQ va seguido de 0). La macro *inc* lleva a cabo este incremento. Esta función se ha definido como macro porque se usa en línea dentro de la ruta crítica. Como veremos después en este libro, el factor que limita el desempeño de una red con frecuencia es el procesamiento del protocolo, por lo que definir como macros las operaciones sencillas como ésta no afecta la legibilidad del código y sí mejora el desempeño. Además, ya que MAX_SEQ tendrá diferentes valores en diferentes protocolos, al hacer que *inc* sea una macro, cabe la posibilidad de incluir todos los protocolos en el mismo binario sin conflictos. Esta capacidad es útil para el simulador.

Las declaraciones de la figura 3-9 son parte de todos los protocolos que siguen. Para ahorrar espacio y proporcionar una referencia práctica, se han extraído y listado juntas, pero conceptualmente deberían estar integradas con los protocolos mismos. En C, esta integración se efectúa poniendo las definiciones en un archivo especial de encabezado, en este caso *protocol.h*, y usando la directiva #include del preprocesador de C para incluirlas en los archivos de protocolos.

3.3.1 Un protocolo simplex sin restricciones

Como ejemplo inicial consideraremos un protocolo que es lo más sencillo posible. Los datos se transmiten sólo en una dirección; las capas de red tanto del emisor como del receptor siempre están listas; el tiempo de procesamiento puede ignorarse; hay un espacio infinito de búfer y, lo mejor de todo, el canal de comunicación entre las capas de enlace de datos nunca tiene problemas ni pierde tramas. Este protocolo completamente irreal, al que apodaremos “utopía”, se muestra en la figura 3-10.

El protocolo consiste en dos procedimientos diferentes, uno emisor y uno receptor. El emisor se ejecuta en la capa de enlace de datos de la máquina de origen y el receptor se ejecuta en la capa de enlace de datos de la máquina de destino. No se usan números de secuencia ni confirmaciones de recepción, por lo que no se necesita MAX_SEQ . El único tipo de evento posible es *frame_arrival* (es decir, la llegada de una trama sin daños).

El emisor está en un ciclo while infinito que sólo envía datos a la línea tan rápidamente como puede. El cuerpo del ciclo consiste en tres acciones: obtener un paquete de la (siempre dispuesta) capa de red, construir una trama de salida usando la variable *s* y enviar la trama a su destino. Este protocolo sólo utiliza el campo *info* de la trama, pues los demás campos tienen que ver con el control de errores y de flujo, y aquí no hay restricciones de control de errores ni de flujo.

El receptor también es sencillo. Inicialmente, espera que algo ocurra, siendo la única posibilidad la llegada de una trama sin daños. En algún momento, la trama llega y el procedimiento *wait_for_event* regresa, conteniendo *event* el valor *frame_arrival* (que de todos modos se ignora). La llamada a *from_physical_layer* elimina la trama recién llegada del búfer de hardware y la

```

/* El protocolo 1 (utopía) provee la transmisión de datos en una sola
dirección, del emisor al receptor. Se supone que el canal de comunicación
está libre de errores, y que el receptor es capaz de procesar toda la entrada
a una rapidez infinita. En consecuencia, el emisor se mantiene en un ciclo,
enviando datos a la línea tan rápidamente como puede. */

typedef enum {frame_arrival} event_type;
#include "protocol.h"

void sender1(void)
{
    frame s;                                /* búfer para una trama de salida */
    packet buffer;                           /* búfer para un paquete de salida */

    while (true) {
        from_network_layer(&buffer); /* consigue algo que enviar */
        s.info = buffer;           /* lo copia en s para transmisión */
        to_physical_layer(&s);   /* lo envía a su destino */
    }                                         /* Mañana, y mañana, y mañana,
                                                Se arrastra a este mísero paso de día
                                                a día
                                                Hasta la última sílaba del tiempo
                                                recordado
                                                -Macbeth, V, v */
}

void receiver1(void)
{
    frame r;
    event_type event;                      /* ocupado por wait, pero no se usa aquí */

    while (true) {
        wait_for_event(&event);          /* la única posibilidad es frame_arrival */
        from_physical_layer(&r);        /* obtiene la trama entrante */
        to_network_layer(&r.info);     /* pasa los datos a la capa de red */
    }
}

```

Figura 3-10. Protocolo simplex sin restricciones.

coloca en la variable *r*, en donde el código receptor pueda obtenerla. Por último, la parte de datos se pasa a la capa de red y la capa de enlace de datos se retira para esperar la siguiente trama, suspendiéndose efectivamente hasta que llega la trama.

3.3.2 Protocolo simplex de parada y espera

Ahora omitiremos el supuesto más irreal hecho en el protocolo 1: la capacidad de la capa de red receptora de procesar datos de entrada con una rapidez infinita (o, lo que es equivalente, la presencia en la capa de enlace de datos receptora de una cantidad infinita de espacio de búfer en el cual almacenar todas las tramas de entrada mientras esperan su respectivo turno). Todavía se supone que el canal de comunicaciones está libre de errores y que el tráfico de datos es simplex.

El problema principal que debemos resolver aquí es cómo evitar que el emisor sature al receptor enviando datos a mayor velocidad de la que este último puede procesarlos. En esencia, si el receptor requiere un tiempo Δt para ejecutar *from_physical_layer* más *to_network_layer*, el emisor debe transmitir a una tasa media menor que una trama por tiempo Δt . Es más, si suponemos que en el hardware del receptor no se realiza de manera automática el almacenamiento en el búfer y el encolamiento, el emisor nunca debe transmitir una trama nueva hasta que la vieja haya sido obtenida por *from_physical_layer*, para que lo nuevo no sobrescriba lo antiguo.

En ciertas circunstancias restringidas (por ejemplo, transmisión síncrona y una capa de enlace de datos receptora dedicada por completo a procesar la línea de entrada única), el emisor podría introducir simplemente un retardo en el protocolo 1 y así reducir su velocidad lo suficiente para evitar que se sature el receptor. Sin embargo, es más común que la capa de enlace de datos tenga varias líneas a las cuales atender, y el intervalo de tiempo entre la llegada de una trama y su procesamiento puede variar en forma considerable. Si los diseñadores de la red pueden calcular el comportamiento de peor caso del receptor, podrán programar al emisor para que transmita con tanta lentitud que, aun si cada trama sufre el retardo máximo, no haya desbordamientos. El problema con este método es que es demasiado conservador. Conduce a un aprovechamiento del ancho de banda muy por debajo del óptimo, a menos que el mejor caso y el peor sean iguales (es decir, la variación en el tiempo de reacción de la capa de enlace de datos sea pequeña).

Una solución más general para este dilema es hacer que el receptor proporcione retroalimentación al emisor. Tras haber pasado un paquete a su capa de red, el receptor regresa al emisor una pequeña trama ficticia que, de hecho, autoriza al emisor para transmitir la siguiente trama. Tras haber enviado una trama, el protocolo exige que el emisor espere hasta que llegue la pequeña trama ficticia (es decir, la confirmación de recepción). Utilizar la retroalimentación del receptor para indicar al emisor cuándo puede enviar más datos es un ejemplo del control de flujo que se mencionó anteriormente.

Los protocolos en los que el emisor envía una trama y luego espera una confirmación de recepción antes de continuar se denominan de **parada y espera**. En la figura 3-11 se da un ejemplo de un protocolo simplex de parada y espera.

Aunque el tráfico de datos en este ejemplo es simplex, y va sólo desde el emisor al receptor, las tramas viajan en ambas direcciones. En consecuencia, el canal de comunicación entre las dos capas de enlace de datos necesita tener capacidad de transferencia de información bidireccional. Sin embargo, este protocolo implica una alternancia estricta de flujo: primero el emisor envía una trama, después el receptor envía una trama, después el emisor envía otra trama, después el receptor envía otra, y así sucesivamente. Aquí sería suficiente un canal físico semidúplex.

```

/* El protocolo 2 (parada y espera) también contempla un flujo unidireccional
de datos del emisor al receptor. Se da por hecho nuevamente que el canal de
comunicación está libre de errores, como en el protocolo 1. Sin embargo, esta
vez el receptor tiene capacidad finita de búfer y capacidad finita de
procesamiento, por lo que el protocolo debe evitar de manera explícita que
el emisor sature al receptor con datos a mayor velocidad de la que puede
manejarse. */

typedef enum {frame_arrival} event_type;
#include "protocol.h"

void sender2(void)
{
    frame s;                                /* búfer para una trama de salida */
    packet buffer;                           /* búfer para un paquete de salida */
    event_type event;                        /* frame_arrival es la única posibilidad */

    while (true) {
        from_network_layer(&buffer);        /* consigue algo que enviar */
        s.info = buffer;                     /* lo copia en s para transmisión */
        to_physical_layer(&s);              /* adiós a la pequeña trama */
        wait_for_event(&event);            /* no procede hasta que recibe la señal de
                                             continuación */
    }
}

void receiver2(void)
{
    frame r, s;                            /* búferes para las tramas */
    event_type event;                      /* frame_arrival es la única posibilidad */
    while (true) {
        wait_for_event(&event);           /* la única posibilidad es frame_arrival */
        from_physical_layer(&r);         /* obtiene la trama entrante */
        to_network_layer(&r.info);       /* pasa los datos a la capa de red */
        to_physical_layer(&s);           /* envía una trama ficticia para informar
                                             al emisor */
    }
}

```

Figura 3-11. Protocolo simplex de parada y espera.

Al igual que en el protocolo 1, el emisor comienza obteniendo un paquete de la capa de red, usándolo para construir una trama y enviarla a su destino. Sólo que ahora, a diferencia del protocolo 1, el emisor debe esperar hasta que llegue una trama de confirmación de recepción antes de reiniciar el ciclo y obtener el siguiente paquete de la capa de red. La capa de enlace de datos emisora no necesita inspeccionar la trama entrante, ya que sólo hay una posibilidad: la trama siempre es de confirmación de recepción.

La única diferencia entre *receiver1* y *receiver2* es que, tras entregar un paquete a la capa de red, *receiver2* regresa al emisor una trama de confirmación de recepción antes de entrar nuevamente en el ciclo de espera. Puesto que sólo es importante la llegada de la trama en el emisor, no su contenido, el receptor no necesita poner ninguna información específica en él.

3.3.3 Protocolo simplex para un canal con ruido

Ahora consideremos la situación normal de un canal de comunicación que comete errores. Las tramas pueden llegar dañadas o perderse por completo. Sin embargo, suponemos que si una trama se daña en tránsito, el hardware del receptor detectará esto cuando calcule la suma de verificación. Si la trama está dañada de tal manera que pese a ello la suma de verificación sea correcta, un caso excesivamente improbable, este protocolo (y todos los demás) puede fallar (es decir, entregar un paquete incorrecto a la capa de red).

A primera vista puede parecer que funcionaría una variación del protocolo 2: agregar un temporizador. El emisor podría enviar una trama, pero el receptor sólo enviaría una trama de confirmación de recepción si los datos llegaran correctamente. Si llegara una trama dañada al receptor, se desecharía. Poco después, el temporizador del emisor expiraría y se enviaría la trama de nuevo. Este proceso se repetiría hasta que la trama por fin llegara intacta.

El esquema anterior tiene un defecto mortal. Medite el problema e intente descubrir lo que podría fallar antes de continuar leyendo.

Para ver lo que puede resultar mal, recuerde que la capa de enlace de datos debe proporcionar una comunicación transparente y libre de errores entre los procesos de las capas de red. La capa de red de la máquina *A* pasa una serie de paquetes a la capa de enlace de datos, que debe asegurar que se entregue una serie de paquetes idéntica a la capa de red de la máquina *B* a través de su capa de enlace de datos. En particular, la capa de red en *B* no tiene manera de saber si el paquete se ha perdido o se ha duplicado, por lo que la capa de enlace de datos debe garantizar que ninguna combinación de errores de transmisión, por improbables que sean, pueda causar la entrega de un paquete duplicado a la capa de red.

Considere el siguiente escenario:

1. La capa de red de *A* entrega el paquete 1 a su capa de enlace de datos. El paquete se recibe correctamente en *B* y se pasa a la capa de red de *B*. *B* regresa a *A* una trama de confirmación de recepción.
2. La trama de confirmación de recepción se pierde por completo. Nunca llega. La vida sería mucho más sencilla si el canal sólo alterara o perdiera tramas de datos y no tramas de control, pero desgraciadamente el canal no hace distinciones.
3. El temporizador de la capa de enlace de datos de *A* expira en algún momento. Al no haber recibido una confirmación de recepción, supone (incorrectamente) que su trama de datos se ha perdido o dañado, y envía otra vez la trama que contiene el paquete 1.

4. La trama duplicada también llega bien a la capa de enlace de datos de B y de ahí se pasa de manera inadvertida a la capa de red. Si A está enviando un archivo a B , parte del archivo se duplicará (es decir, la copia del archivo reconstruida por B será incorrecta y el error no se habrá detectado). En otras palabras, el protocolo fallará.

Es claro que lo que se necesita es alguna manera de que el receptor sea capaz de distinguir entre una trama que está viendo por primera vez y una retransmisión. La forma evidente de lograr esto es hacer que el emisor ponga un número de secuencia en el encabezado de cada trama que envía. A continuación, el receptor puede examinar el número de secuencia de cada trama que llega para ver si es una trama nueva o un duplicado que debe descartarse.

Dado que es deseable que el encabezado de las tramas sea pequeño, surge la pregunta: ¿cuál es la cantidad mínima de bits necesarios para el número de secuencia? La única ambigüedad de este protocolo es entre una trama, m , y su sucesor directo, $m + 1$. Si la trama m se pierde o se daña, el receptor no confirmará su recepción y el emisor seguirá tratando de enviarla. Una vez que la trama se recibe correctamente, el receptor regresa una confirmación de recepción al emisor. Es aquí donde surge el problema potencial. Dependiendo de si el emisor recibe correctamente la trama de confirmación de recepción, tratará de enviar m o $m + 1$.

El evento que indica al emisor que puede enviar $m + 2$ es la llegada de una confirmación de recepción de $m + 1$. Pero esto implica que m se recibió de manera correcta, y además que su confirmación de recepción fue recibida correctamente por el emisor (de otra manera, el emisor no habría comenzado con $m + 1$, y mucho menos con $m + 2$). Como consecuencia, la única ambigüedad es entre una trama y su antecesor o sucesor inmediatos, no entre el antecesor y el sucesor mismos.

Por lo tanto, basta con un número de secuencia de 1 bit (0 o 1). En cada instante, el receptor espera un número de secuencia en particular. Cualquier trama de entrada que contenga un número de secuencia equivocado se rechaza como duplicado. Cuando llega una trama que contiene el número de secuencia correcto, se acepta y se pasa a la capa de red, y el número de secuencia esperado se incrementa módulo 2 (es decir, 0 se vuelve 1 y 1 se vuelve 0).

En la figura 3-12 se muestra un ejemplo de este tipo de protocolo. Los protocolos en los que el emisor espera una confirmación de recepción positiva antes de avanzar al siguiente elemento de datos suelen llamarse **PAR (Confirmación de Recepción Positiva con Retransmisión)** o **ARQ (Solicitud Automática de Repetición)**. Al igual que el protocolo 2, éste también transmite datos en una sola dirección.

El protocolo 3 difiere de sus antecesores en que tanto el emisor como el receptor tienen una variable cuyo valor se recuerda mientras la capa de enlace de datos está en estado de espera. El emisor recuerda el número de secuencia de la siguiente trama a enviar en *next_frame_to_send*; el receptor recuerda el número de secuencia de la siguiente trama esperada en *frame_expected*. Cada protocolo tiene una fase de inicialización corta antes de entrar en el ciclo infinito.

```

/* El protocolo 3 (par) permite el flujo unidireccional de datos por un canal no con-
fiable. */

#define MAX_SEQ 1                                /* debe ser 1 para el protocolo 3 */
typedef enum {frame_arrival, cksum_err, timeout} event_type;
#include "protocol.h"

void sender3(void)
{
    seq_nr next_frame_to_send;                  /* número de secuencia de la siguiente
                                                trama de salida */
    frame s;                                     /* variable de trabajo */
    packet buffer;                               /* búfer para un paquete de salida */
    event_type event;

    next_frame_to_send = 0;                      /* inicializa números de secuencia de
                                                salida */
    from_network_layer(&buffer);                /* obtiene el primer paquete */

    while (true){
        s.info = buffer;                         /* construye una trama para transmisión */
        s.seq = next_frame_to_send;               /* inserta un número de secuencia en la
                                                trama */
        to_physical_layer(&s);                  /* la envía a su destino */
        start_timer(s.seq);                     /* si la respuesta tarda mucho, expira el
                                                temporizador */
        wait_for_event(&event);
        if (event == frame_arrival){            /* frame_arrival, cksum_err, timeout */
            from_physical_layer(&s);           /* obtiene la confirmación de recepción */
            if (s.ack == next_frame_to_send){   /* desactiva el temporizador */
                stop_timer(s.ack);
                from_network_layer(&buffer);    /* obtiene siguiente a enviar */
                inc(next_frame_to_send);        /* invierte next_frame_to_send */
            }
        }
    }
}

void receiver3(void)
{
    seq_nr frame_expected;
    frame r, s;
    event_type event;

    frame_expected = 0;
    while (true){
        wait_for_event(&event);
        if (event == frame_arrival){            /* posibilidades: frame_arrival, cksum_err */
            from_physical_layer(&r);          /* ha llegado una trama válida. */
            if (r.seq == frame_expected){      /* obtiene la trama recién llegada */
                to_network_layer(&r.info);    /* esto es lo que hemos estado esperando. */
                inc(frame_expected);          /* pasa los datos a la capa de red */
                /* para la próxima se espera el otro número
                de secuencia */
            }
            s.ack = 1 - frame_expected;       /* indica la trama cuya recepción se está
                                                confirmando */
            to_physical_layer(&s);           /* envía confirmación de recepción */
        }
    }
}

```

Figura 3-12. Protocolo de confirmación de recepción positiva con retransmisión.

Tras transmitir una trama, el emisor arranca el temporizador. Si éste ya se estaba ejecutando, se restablece para conceder otro intervalo completo de temporización. Dicho intervalo debe escogerse de modo que haya suficiente tiempo para que la trama llegue al receptor, éste la procese en el peor caso y la confirmación de recepción se regrese al emisor. Sólo cuando ha transcurrido ese intervalo de tiempo se puede suponer con seguridad que se ha perdido la trama transmitida o su confirmación de recepción, y que se debe enviar un duplicado. Si el intervalo establecido es muy pequeño, el emisor transmitirá tramas innecesarias. Si bien estas tramas adicionales no afectarán la corrección del protocolo, sí dañarán el rendimiento.

Tras transmitir una trama y arrancar el temporizador, el emisor espera que ocurra algo interesante. Hay tres posibilidades: llega una trama de confirmación de recepción sin daño, llega una trama de confirmación de recepción dañada o expira el temporizador. Si recibe una confirmación de recepción válida, el emisor obtiene el siguiente paquete de la capa de red y lo coloca en el búfer, sobrescribiendo el paquete previo. También avanza el número de secuencia. Si llega una trama dañada o no llega ninguna, ni el búfer ni el número de secuencia cambia, con el fin de que se pueda enviar un duplicado.

Cuando llega una trama válida al receptor, su número de secuencia se verifica para saber si es un duplicado. Si no lo es, se acepta, se pasa a la capa de red y se genera una confirmación de recepción. Los duplicados y las tramas dañadas no se pasan a la capa de red.

3.4 PROTOCOLOS DE VENTANA CORREDIZA

En los protocolos previos, las tramas de datos se transmiten en una sola dirección. En la mayoría de las situaciones prácticas hay necesidad de transmitir datos en ambas direcciones. Una manera de lograr una transmisión de datos dúplex total es tener dos canales de comunicación separados y utilizar cada uno para tráfico de datos simplex (en diferentes direcciones). Si se hace esto, tenemos dos circuitos físicos separados, cada uno con un canal “de ida” (para datos) y un canal “de retorno” (para confirmaciones de recepción). En ambos casos, el ancho de banda del canal usado para confirmaciones de recepción se desperdicia casi por completo. En efecto, el usuario está pagando dos circuitos, pero sólo usa la capacidad de uno.

Una mejor idea es utilizar el mismo circuito para datos en ambas direcciones. Después de todo, en los protocolos 2 y 3 ya se usaba para transmitir tramas en ambos sentidos, y el canal de retorno tiene la misma capacidad que el canal de ida. En este modelo, las tramas de datos de *A* a *B* se mezclan con las tramas de confirmación de recepción de *A* a *B*. Analizando el campo de tipo (*kind*) en el encabezado de una trama de entrada, el receptor puede saber si la trama es de datos o de confirmación de recepción.

Aunque el entrelazado de datos y de tramas de control en el mismo circuito es una mejora respecto al uso de dos circuitos físico separados, se puede lograr otra mejora. Cuando llega una trama de datos, en lugar de enviar inmediatamente una trama de control independiente, el receptor se aguanta y espera hasta que la capa de red le pasa el siguiente paquete. La confirmación de recepción se anexa a la trama de datos de salida (usando el campo *ack* del encabezado de la trama). En efecto, la confirmación de recepción viaja gratuitamente en la siguiente trama de datos de salida.

La técnica de retardar temporalmente las confirmaciones de recepción para que puedan viajar en la siguiente trama de datos de salida se conoce como **superposición** (*piggybacking*).

La ventaja principal de usar la superposición en lugar de tener tramas de confirmación de recepción independientes es un mejor aprovechamiento del ancho de banda disponible del canal. El campo *ack* del encabezado de la trama ocupa sólo unos cuantos bits, mientras que una trama aparte requeriría de un encabezado, la confirmación de recepción y una suma de verificación. Además, el envío de menos tramas implica menos interrupciones de “ha llegado trama” y tal vez menos segmentos de búfer en el receptor, dependiendo de la manera en que esté organizado el software del receptor. En el siguiente protocolo que examinaremos, el campo de superposición ocupa sólo 1 bit en el encabezado de la trama y pocas veces ocupa más de algunos bits.

Sin embargo, la superposición introduce una complicación inexistente en las confirmaciones de recepción independientes. ¿Cuánto tiempo debe esperar la capa de enlace de datos un paquete al cual superponer la confirmación de recepción? Si la capa de enlace de datos espera más tiempo del que tarda en terminar el temporizador del emisor, la trama será retransmitida, frustrando el propósito de enviar confirmaciones de recepción. Si la capa de enlace de datos fuera un oráculo y pudiera predecir el futuro, sabría cuándo se recibiría el siguiente paquete de la capa de red y podría decidir esperarlo o enviar de inmediato una confirmación de recepción independiente, dependiendo del tiempo de espera proyectado. Por supuesto, la capa de enlace de datos no puede predecir el futuro, por lo que debe recurrir a algún esquema particular para el caso, como esperar un número fijo de milisegundos. Si llega rápidamente un nuevo paquete, la confirmación de recepción se superpone a él; de otra manera, si no ha llegado ningún paquete nuevo al final de este periodo, la capa de enlace de datos manda una trama de confirmación de recepción independiente.

Los siguientes tres protocolos son bidireccionales y pertenecen a una clase llamada protocolos de **ventana corrediza**. Los tres difieren entre ellos en la eficiencia, complejidad y requerimientos de búfer, como se analizará más adelante. En ellos, al igual que en todos los protocolos de ventana corrediza, cada trama de salida contiene un número de secuencia, que va desde 0 hasta algún número máximo. Por lo general, éste es $2^n - 1$, por lo que el número de secuencia encaja perfectamente en un campo de n bits. El protocolo de ventana corrediza de parada y espera utiliza $n = 1$, y restringe los números de secuencia de 0 y 1, pero las versiones más refinadas pueden utilizar un n arbitrario.

La esencia de todos los protocolos de ventana corrediza es que, en cualquier instante, el emisor mantiene un grupo de números de secuencia que corresponde a las tramas que tiene permitido enviar. Se dice que estas tramas caen dentro de la **ventana emisora**. De manera semejante, el receptor mantiene una **ventana receptora** correspondiente al grupo de tramas que tiene permitido aceptar. La ventana del emisor y la del receptor no necesitan tener los mismos límites inferior y superior, ni siquiera el mismo tamaño. En algunos protocolos las ventanas son de tamaño fijo, pero en otros pueden crecer y disminuir a medida que se envían y reciben las tramas.

Aunque estos protocolos dan a la capa de enlace de datos mayor libertad en cuanto al orden en que puede enviar y recibir tramas, hemos conservado decididamente el requisito de que el protocolo debe entregar los paquetes a la capa de red del destino en el mismo orden en que se pasaron a la capa de enlace de datos de la máquina emisora. Tampoco hemos cambiado el requisito de que

el canal físico de comunicación es “de tipo alambre”, es decir, que debe entregar todas las tramas en el orden en que fueron enviadas.

Los números de secuencia en la ventana del emisor representan tramas enviadas, o que pueden ser enviadas, pero cuya recepción aún no se ha confirmado. Cuando llega un paquete nuevo de la capa de red, se le da el siguiente número secuencial mayor, y el extremo superior de la ventana avanza en uno. Al llegar una confirmación de recepción, el extremo inferior avanza en uno. De esta manera, la ventana mantiene continuamente una lista de tramas sin confirmación de recepción. En la figura 3-13 se muestra un ejemplo.

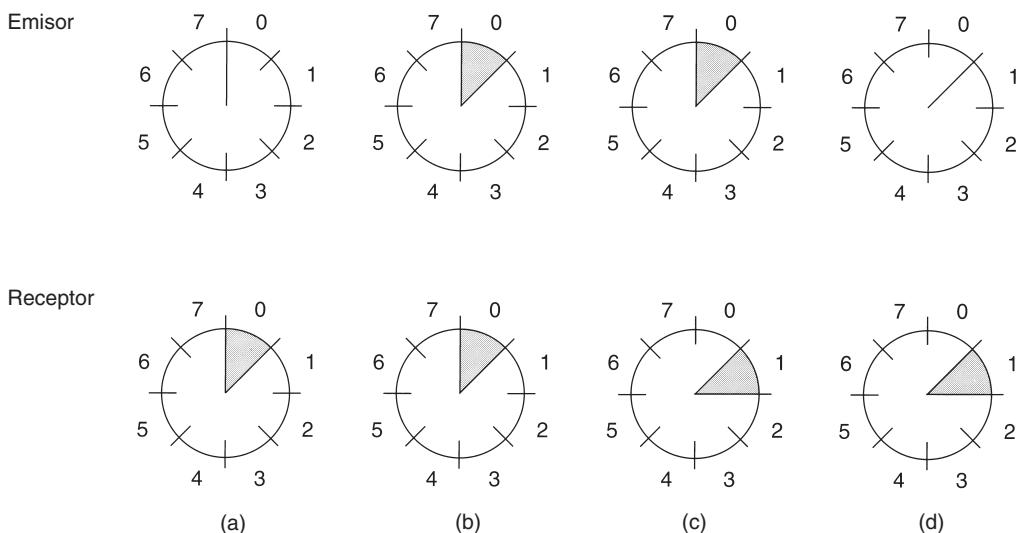


Figura 3-13. Ventana corrediza de tamaño 1, con un número de secuencia de 3 bits. (a) Al inicio. (b) Tras la transmisión de la primera trama. (c) Tras la recepción de la primera trama. (d) Tras recibir la primera confirmación de recepción.

Dado que las tramas que están en la ventana del emisor pueden perderse o dañarse en tránsito, el emisor debe mantener todas estas tramas en su memoria para su posible retransmisión. Por lo tanto, si el tamaño máximo de la ventana es n , el emisor necesita n búferes para contener las tramas sin confirmación de recepción. Si la ventana llega a crecer a su tamaño máximo, la capa de enlace de datos emisora deberá hacer que la capa de red se detenga hasta que se libere otro búfer.

La ventana de la capa de enlace de datos receptor corresponde a las tramas que puede aceptar. Toda trama que caiga fuera de la ventana se descartará sin comentarios. Cuando se recibe la trama cuyo número de secuencia es igual al extremo inferior de la ventana, se pasa a la capa de red, se genera una confirmación de recepción y se avanza la ventana en uno. A diferencia de la ventana del emisor, la ventana del receptor conserva siempre el mismo tamaño inicial. Note que

un tamaño de ventana de 1 significa que la capa de enlace de datos sólo acepta tramas en orden, pero con ventanas más grandes esto no es así. La capa de red, en contraste, siempre recibe los datos en el orden correcto, sin importar el tamaño de la ventana de la capa de enlace de datos.

En la figura 3-13 se muestra un ejemplo con un tamaño máximo de ventana de 1. Inicialmente no hay tramas pendientes, por lo que los extremos de la ventana del emisor son iguales, pero a medida que pasa el tiempo, la situación progresiva como se muestra.

3.4.1 Un protocolo de ventana corrediza de un bit

Antes de lidar con el caso general, examinemos un protocolo de ventana corrediza con un tamaño máximo de ventana de 1. Tal protocolo utiliza parada y espera, ya que el emisor envía una trama y espera su confirmación de recepción antes de transmitir la siguiente.

En la figura 3-14 se presenta tal protocolo. Como los demás, comienza por definir algunas variables. *Next_frame_to_send* indica qué trama está tratando de enviar el emisor. De manera semejante, *frame_expected* indica qué trama espera el receptor. En ambos casos, 0 y 1 son las únicas posibilidades.

Normalmente, una de las dos capas de enlace de datos es la que comienza a transmitir la primera trama. En otras palabras, sólo uno de los programas de capa de enlace de datos debe contener las llamadas de procedimiento *to_physical_layer* y *start_timer* fuera del ciclo principal. Si ambas capas se iniciaran en forma simultánea, surgiría una situación peculiar que se analizará después. La máquina que arranca obtiene el primer paquete de su capa de red, construye una trama a partir de él y la envía. Al llegar esta (o cualquier) trama, la capa de enlace de datos receptora la revisa para saber si es un duplicado, igual que en el protocolo 3. Si la trama es la esperada, se pasa a la capa de red y la ventana del receptor se recorre hacia arriba.

El campo de confirmación de recepción contiene el número de la última trama recibida sin error. Si este número concuerda con el de secuencia de la trama que está tratando de enviar el emisor, éste sabe que ha terminado con la trama almacenada en el búfer y que puede obtener el siguiente paquete de su capa de red. Si el número de secuencia no concuerda, debe continuar intentando enviar la misma trama. Por cada trama que se recibe, se regresa una.

Ahora examinemos el protocolo 4 para ver qué tan flexible es ante circunstancias patológicas. Suponga que *A* está tratando de enviar su trama 0 a *B* y que *B* está tratando de enviar su trama 0 a *A*. Suponga que *A* envía una trama a *B*, pero que el intervalo de temporización de *A* es un poco corto. En consecuencia, *A* podría terminar su temporización repetidamente, enviando una serie de tramas idénticas, todas con *seq* = 0 y *ack* = 1.

Al llegar la primera trama válida a *B*, es aceptada y *frame_expected* se establece en 1. Todas las tramas subsiguientes serán rechazadas, pues *B* ahora espera tramas con el número de secuencia 1, no 0. Además, dado que los duplicados tienen *ack* = 1 y *B* aún está esperando una confirmación de recepción de 0, *B* no extraerá un nuevo paquete de su capa de red.

```

/* El protocolo 4 (de ventana corrediza) es bidireccional. */
#define MAX_SEQ 1                                /* debe ser 1 para el protocolo 4 */
typedef enum {frame_arrival, cksum_err, timeout} event_type;
#include "protocol.h"
void protocol4 (void)
{
    seq_nr next_frame_to_send;                  /* sólo 0 o 1 */
    seq_nr frame_expected;                     /* sólo 0 o 1 */
    frame r, s;                               /* variables de trabajo */
    packet buffer;                           /* paquete actual que se envía */
    event_type event;                         /* siguiente trama del flujo de salida */
    frame_expected = 0;                      /* número de trama de llegada esperada */
    from_network_layer(&buffer);            /* obtiene un paquete de la capa de red */
    s.info = buffer;                          /* se prepara para enviar la trama inicial */
    s.seq = next_frame_to_send;                /* inserta el número de secuencia en la trama */
    s.ack = 1 - frame_expected;              /* confirmación de recepción superpuesta */
    to_physical_layer(&s);                  /* transmite la trama */
    start_timer(s.seq);                     /* inicia el temporizador */

    while (true){
        wait_for_event(&event);

        if (event == frame_arrival){
            from_physical_layer(&r);
            if(r.seq == frame_expected) {
                to_network_layer(&r.info);
                inc(frame_expected);
            }
            if(r.ack == next_frame_to_send){
                stop_timer(r.ack);
                from_network_layer(&buffer);
                inc(next_frame_to_send);
            }
        }
        s.info = buffer;
        s.seq = next_frame_to_send;
        s.ack = 1 - frame_expected;
        to_physical_layer(&s);
        start_timer(s.seq);
    }
}

```

/* frame_arrival, cksum_err o timeout */
 /* ha llegado una trama sin daño. */
 /* lo obtiene */
 /* maneja flujo de tramas de entrada. */
 /* pasa el paquete a la capa de red */
 /* invierte el siguiente número de secuencia esperado */

 /* maneja flujo de tramas de salida. */
 /* desactiva el temporizador */
 /* obtiene paquete nuevo de la capa de red */
 /* invierte el número de secuencia del emisor */

 /* construye trama de salida */
 /* le introduce el número de secuencia */
 /* número de secuencia de la última trama recibida */
 /* transmite una trama */
 /* inicia el temporizador */

Figura 3-14. Protocolo de ventana corrediza de 1 bit.

Cada vez que llega un duplicado rechazado, B envía a A una trama que contiene $seq = 0$ y $ack = 0$. Tarde o temprano una de éstas llegará correctamente a A , haciendo que A comience a enviar el siguiente paquete. Ninguna combinación de tramas perdidas o expiración de temporizadores puede hacer que el protocolo entregue paquetes duplicados a cualquiera de las capas de red, ni que omita un paquete, ni que entre en un bloqueo irreversible.

Sin embargo, si ambos lados envían de manera simultánea un paquete inicial, surge una situación peculiar. En la figura 3-15 se muestra este problema de sincronización. En la parte (a) se muestra la operación normal del protocolo. En (b) se ilustra la peculiaridad. Si B espera la primera trama de A antes de enviar la suya, la secuencia es como se muestra en (a), y todas las tramas son aceptadas. Sin embargo, si A y B inician la comunicación simultáneamente, se cruzan sus primeras tramas y las capas de enlace de datos entran en la situación (b). En (a) cada trama que llega trae un paquete nuevo para la capa de red; no hay duplicados. En (b) la mitad de las tramas contienen duplicados, aun cuando no hay errores de transmisión. Pueden ocurrir situaciones similares como resultado de la expiración prematura de temporizadores, aun cuando un lado comience evidentemente primero. De hecho, si ocurren varias expiraciones prematuras de temporizadores las tramas podrían enviarse tres o más veces.

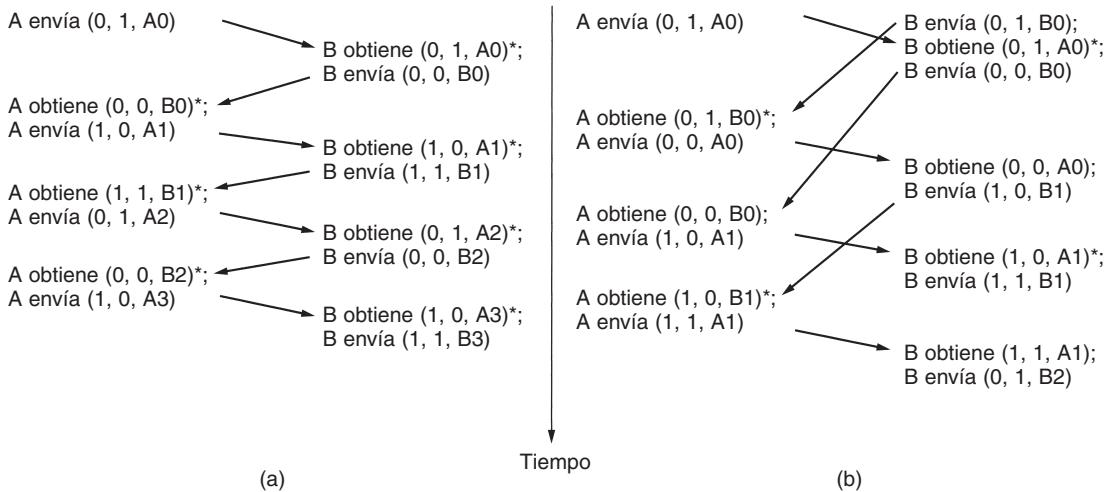


Figura 3-15. Dos escenarios para el protocolo 4. (a) Caso normal. (b) Caso anormal. La notación es (secuencia, confirmación de recepción, número de paquete). Un asterisco indica el lugar en que una capa de red acepta un paquete.

3.4.2 Protocolo que usa retroceso N

Hasta ahora hemos supuesto que el tiempo de transmisión requerido para que una trama llegue al receptor más el necesario para que la confirmación de recepción regrese es insignificante. A veces esta suposición es totalmente falsa. En estas situaciones el tiempo de viaje de ida y vuelta prolongado puede tener implicaciones importantes para la eficiencia del aprovechamiento del ancho

de banda. Por ejemplo, considere un canal de satélite de 50 kbps con un retardo de propagación de ida y vuelta de 500 mseg. Imagine que intentamos utilizar el protocolo 4 para enviar tramas de 1000 bits por medio del satélite. El emisor empieza a enviar la primera trama en $t = 0$. En $t = 20$ mseg la trama ha sido enviada por completo. En las mejores circunstancias (sin esperas en el receptor y una trama de confirmación de recepción corta), no es sino hasta $t = 270$ mseg que la trama ha llegado por completo al receptor, y no es sino hasta $t = 520$ mseg que ha llegado la confirmación de recepción de regreso al emisor. Esto implica que el emisor estuvo bloqueado durante el $500/520 = 96\%$ del tiempo. En otras palabras, sólo se usó el 4% del ancho de banda disponible. Queda claro que la combinación de un tiempo de tránsito grande, un ancho de banda alto y una longitud de tramas corta es desastrosa para la eficiencia.

El problema antes descrito puede verse como una consecuencia de la regla que requiere que el emisor espere una confirmación de recepción antes de enviar otra trama. Si relajamos esa restricción, se puede lograr una mejor eficiencia. Básicamente la solución está en permitir que el emisor envíe hasta w tramas antes de bloquearse, en lugar de sólo 1. Con una selección adecuada de w , el emisor podrá transmitir tramas continuamente durante un tiempo igual al tiempo de tránsito de ida y vuelta sin llenar la ventana. En el ejemplo anterior, w debe ser de cuando menos 26. El emisor comienza por enviar la trama 0, como antes. Para cuando ha terminado de enviar 26 tramas, en $t = 520$, llega la confirmación de recepción de la trama 0. A partir de entonces, las confirmaciones de recepción llegarán cada 20 mseg, por lo que el emisor siempre tendrá permiso de continuar justo cuando lo necesita. En todo momento hay 25 o 26 tramas pendientes de confirmación de recepción. Dicho de otra manera, el tamaño máximo de la ventana del emisor es de 26.

La necesidad de una ventana grande en el lado emisor se presenta cuando el producto del ancho de banda por el retardo del viaje de ida y vuelta es grande. Si el ancho de banda es alto, incluso para un retardo moderado, el emisor agotará su ventana rápidamente a menos que tenga una ventana grande. Si el retardo es grande (por ejemplo, en un canal de satélite geoestacionario), el emisor agotará su ventana incluso con un ancho de banda moderado. El producto de estos dos factores indica básicamente cuál es la capacidad del canal, y el emisor necesita la capacidad de llenarlo sin detenerse para poder funcionar con una eficiencia máxima.

Esta técnica se conoce como **canalización**. Si la capacidad del canal es de b bits/seg, el tamaño de la trama de l bits y el tiempo de propagación de ida y vuelta de R segundos, el tiempo requerido para transmitir una sola trama es de l/b segundos. Una vez que ha sido enviado el último bit de una trama de datos, hay un retardo de $R/2$ antes de que llegue ese bit al receptor y un retardo de cuando menos $R/2$ para que la confirmación de recepción llegue de regreso, lo que da un retardo total de R . En parada y espera, la línea está ocupada durante l/b e inactiva durante R , dando

$$\text{una utilización de la línea de } = l/(l + bR)$$

Si $l < bR$, la eficiencia será menor que 50%. Ya que siempre hay un retardo diferente de cero para que la confirmación de recepción se propague de regreso, en principio la canalización puede servir para mantener ocupada la línea durante este intervalo, pero si el intervalo es pequeño, la complejidad adicional no justifica el esfuerzo.

El envío de tramas en canalización por un canal de comunicación inestable presenta problemas serios. Primero, ¿qué ocurre si una trama a la mitad de una serie larga se daña o pierde? Llegarán grandes cantidades de tramas sucesivas al receptor antes de que el emisor se entere de que algo anda mal. Cuando llega una trama dañada al receptor, obviamente debe descartarse, pero, ¿qué debe hacerse con las tramas correctas que le siguen? Recuerde que la capa de enlace de datos receptora está obligada a entregar paquetes a la capa de red en secuencia. En la figura 3-16 se muestran los efectos de la canalización en la recuperación de un error. A continuación lo analizaremos en detalle.

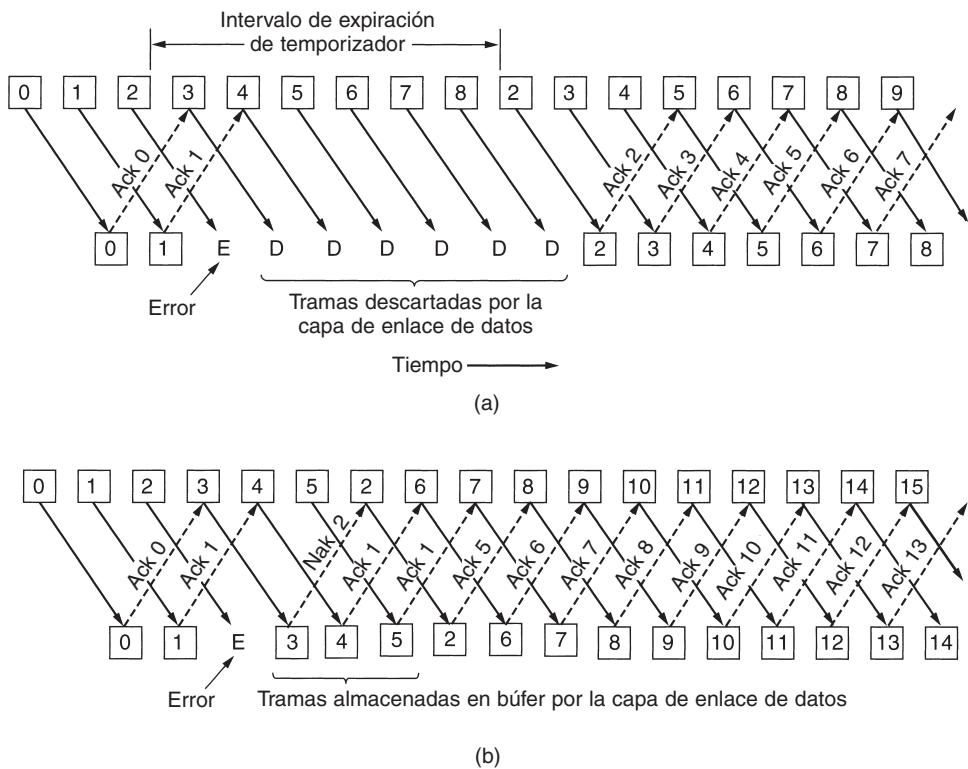


Figura 3-16. Canalización y recuperación de un error. (a) Efecto de un error cuando el tamaño de la ventana del receptor es de 1. (b) Efecto de un error cuando el tamaño de la ventana del receptor es grande.

Hay dos métodos básicos para manejar los errores durante la canalización. Una manera, llamada **retroceso n**, es que el receptor simplemente descarte todas las tramas subsecuentes, sin enviar confirmaciones de recepción para las tramas descartadas. Esta estrategia corresponde a una ventana de recepción de tamaño 1. En otras palabras, la capa de enlace de datos se niega a aceptar cualquier trama excepto la siguiente que debe entregar a la capa de red. Si la ventana del emisor se llena antes de terminar el temporizador, el canal comenzará a vaciarse. En algún momento, el emisor terminará de esperar y retransmitirá en orden todas las tramas cuya recepción aún no se

haya confirmado, comenzando por la dañada o perdida. Esta estrategia puede desperdiciar bastante ancho de banda si la tasa de errores es alta.

En la figura 3-16(a) se muestra el retroceso n en el caso en que la ventana del receptor es grande. Las tramas 0 y 1 se reciben y confirman de manera correcta. Sin embargo, la trama 2 se daña o pierde. El emisor, no consciente de este problema, continúa enviando tramas hasta que expira el temporizador para la trama 2. Después retrocede a la trama 2 y comienza con ella, enviando 2, 3, 4, etcétera, nuevamente.

La otra estrategia general para el manejo de errores cuando las tramas se colocan en canalizaciones se conoce como **repeticIÓN selectiva**. Cuando se utiliza, se descarta una trama dañada recibida, pero las tramas en buen estado recibidas después de ésa se almacenan en el búfer. Cuando el emisor termina, sólo la última trama sin confirmación se retransmite. Si la trama llega correctamente, el receptor puede entregar a la capa de red, en secuencia, todas las tramas que ha almacenado en el búfer. La repetición selectiva con frecuencia se combina con el hecho de que el receptor envíe una confirmación de recepción negativa (NAK) cuando detecta un error, por ejemplo, cuando recibe un error de suma de verificación o una trama en desorden. Las confirmaciones de recepción negativas estimulan la retransmisión antes de que el temporizador correspondiente expire y, por lo tanto, mejoran el rendimiento.

En la figura 3-16(b), las tramas 0 y 1 se vuelven a recibir y confirmar correctamente y la trama 2 se pierde. Cuando la trama 3 llega al receptor, su capa de enlace de datos observa que falta una trama, por lo que regresa una NAK para la trama 2 pero almacena la trama 3. Cuando las tramas 4 y 5 llegan, también son almacenadas por la capa de enlace de datos en lugar de pasarse a la capa de red. En algún momento, la NAK 2 llega al emisor, que inmediatamente reenvía la trama 2. Cuando llega, la capa de enlace de datos ahora tiene 2, 3, 4 y 5 y ya las puede pasar a la capa de red en el orden correcto. También puede confirmar la recepción de todas las tramas hasta, e incluyendo, la 5, como se muestra en la figura. Si la NAK se perdiera, en algún momento el temporizador del emisor expiraría para la trama 2 y la enviaría (sólo a ella), pero eso puede tardar un poco más. En efecto, la NAK acelera la retransmisión de una trama específica.

La repetición selectiva corresponde a una ventana del receptor mayor que 1. Cualquier trama dentro de la ventana puede ser aceptada y mantenida en el búfer hasta que todas las que le preceden hayan sido pasadas a la capa de red. Esta estrategia puede requerir cantidades grandes de memoria en la capa de enlace de datos si la ventana es grande.

Estas dos estrategias alternativas son intercambios entre el ancho de banda y el espacio de búfer en la capa de enlace de datos. Dependiendo de qué recurso sea más valioso, se puede utilizar uno o el otro. En la figura 3-17 se muestra un protocolo de canalización en el que la capa de enlace de datos receptora sólo acepta tramas en orden; las tramas siguientes a un error son descartadas. En este protocolo hemos omitido por primera vez el supuesto de que la capa de red siempre tiene un suministro infinito de paquetes que enviar. Cuando la capa de red tiene un paquete para enviar, puede causar la ocurrencia de un evento *network_layer_ready*. Sin embargo, para poder cumplir la regla de control de flujo de que no debe haber más de *MAX_SEQ* tramas sin confirmación de recepción pendientes en cualquier momento, la capa de enlace de datos debe poder prohibir a la capa de red que la moleste con más trabajo. Los procedimientos de biblioteca *enable_network_layer* y *disable_network_layer* llevan a cabo esta función.

```

/* El protocolo 5 (retroceso n) permite múltiples tramas pendientes. El emisor podría
enviar hasta MAX_SEQ tramas sin esperar una confirmación de recepción. Además, a
diferencia de los primeros protocolos, no se supone que la capa de red debe tener
siempre un paquete nuevo. En vez de ello, la capa de red activa un evento
network_layer_ready cuando hay un paquete por enviar. */

#define MAX_SEQ 7           /* debe ser 2^n - 1 */
typedef enum {frame_arrival, cksum_err, timeout, network_layer_ready} event_type;
#include "protocol.h"

static boolean between(seq_nr a, seq_nr b, seq_nr c)
{
/* Devuelve true si a <= b < c de manera circular, y false en caso contrario. */
    if (((a <= b) && (b < c)) || ((c < a) && (a <= b)) || ((b < c) && (c < a)))
        return(true);
    else
        return(false);
}

static void send_data(seq_nr frame_nr, seq_nr frame_expected, packet buffer[])
{
/* Elabora y envía una trama de datos. */
    frame s;                  /* variable de trabajo */
    s.info = buffer[frame_nr];      /* inserta el paquete en la trama */
    s.seq = frame_nr;            /* inserta un número de secuencia en la trama */
    s.ack = (frame_expected + MAX_SEQ) % (MAX_SEQ + 1);    /* ack superpuesta */
    to_physical_layer(&s);       /* transmite la trama */
    start_timer(frame_nr);      /* inicia la ejecución del temporizador */
}

void protocol5(void)
{
    seq_nr next_frame_to_send;      /* MAX_SEQ > 1; utilizado para flujo de salida */
    seq_nr ack_expected;          /* la trama más vieja hasta el momento no
                                    /* confirmada */
    seq_nr frame_expected;        /* siguiente trama esperada en el flujo de
                                    /* entrada */
    frame r;                     /* variable de trabajo */
    packet buffer[MAX_SEQ + 1];   /* búferes para el flujo de salida */
    seq_nr nbuffered;            /* # de búferes de salida actualmente en uso */
    seq_nr i;                     /* utilizado para indexar en el arreglo de
                                    /* búferes */

    event_type event;

    enable_network_layer();
    ack_expected = 0;

    next_frame_to_send = 0;
    frame_expected = 0;
    nbuffered = 0;
    while (true) {
        wait_for_event(&event);
        /* cuatro posibilidades: vea event_type al principio */

```

```

switch(event) {
    case network_layer_ready:           /* la capa de red tiene un paquete para enviar */
        /* Acepta, guarda y transmite una trama nueva. */
        from_network_layer(&buffer[next_frame_to_send]); /* obtiene un paquete nuevo */
        nbuffed = nbuffed + 1; /* expande la ventana del emisor */
        send_data(next_frame_to_send, frame_expected, buffer); /* transmite la trama */
        inc(next_frame_to_send); /* avanza el límite superior de la ventana del
                                   emisor */
        break;

    case frame_arrival:                /* ha llegado una trama de datos o de control */
        from_physical_layer(&r); /* obtiene una trama entrante de la capa física */

        if (r.seq == frame_expected) {
            /* Las tramas se aceptan sólo en orden. */
            to_network_layer(&r.info); /* pasa el paquete a la capa de red */
            inc(frame_expected); /* avanza el límite inferior de la ventana del
                                   receptor */
        }

        /* Ack n implica n - 1, n - 2, etc. Verificar esto. */
        while (between(ack_expected, r.ack, next_frame_to_send)) {
            /* Maneja la ack superpuesta. */
            nbuffed = nbuffed - 1; /* una trama menos en el búfer */
            stop_timer(ack_expected); /* la trama llegó intacta; detener el
                                         temporizador */
            inc(ack_expected); /* reduce la ventana del emisor */
        }
        break;

    case cksum_err: break;             /* ignora las tramas erróneas */

    case timeout:                     /* problemas; retransmite todas las tramas
                                         pendientes */
        next_frame_to_send = ack_expected; /* inicia aquí la retransmisión */
        for (i = 1; i <= nbuffed; i++) {
            send_data(next_frame_to_send, frame_expected, buffer); /* reenvía la
                           trama 1 */
            inc(next_frame_to_send); /* se prepara para enviar la siguiente */
        }

    }

    if (nbuffed < MAX_SEQ)
        enable_network_layer();
    else
        disable_network_layer();
}
}

```

Figura 3-17. Protocolo de ventana corrediza con retroceso n.

Note que pueden como máximo estar pendientes MAX_SEQ tramas, y no $MAX_SEQ + 1$ en cualquier momento, aun cuando haya $MAX_SEQ + 1$ números de secuencia diferentes: 0, 1, 2,..., MAX_SEQ . Para ver por qué es necesaria esta restricción, considere la siguiente situación con $MAX_SEQ = 7$.

1. El emisor envía las tramas 0 a 7.
2. En algún momento llega al emisor una confirmación de recepción, superpuesta, para la trama 7.
3. El emisor envía otras ocho tramas, nuevamente con los números de secuencia 0 a 7.
4. Ahora llega otra confirmación de recepción, superpuesta, para la trama 7.

La pregunta es: ¿llegaron con éxito las ocho tramas que correspondían al segundo bloque o se perdieron (contando como pérdidas los rechazos siguientes a un error)? En ambos casos el receptor podría estar enviando la trama 7 como confirmación de recepción. El emisor no tiene manera de saberlo. Por esta razón, el número máximo de tramas pendientes debe restringirse a MAX_SEQ .

Aunque el protocolo 5 no pone en el búfer las tramas que llegan tras un error, no escapa del problema de los búferes por completo. Dado que un emisor puede tener que retransmitir en un momento futuro todas las tramas no confirmadas, debe retener todas las tramas retransmitidas hasta saber con certeza que han sido aceptadas por el receptor. Al llegar una confirmación de recepción para la trama n , las tramas $n - 1$, $n - 2$, y demás, se confirman de manera automática. Esta propiedad es especialmente importante cuando algunas tramas previas portadoras de confirmaciones de recepción se perdieron o dañaron. Cuando llega una confirmación de recepción, la capa de enlace de datos revisa si se pueden liberar búferes. Si esto es posible (es decir, hay espacio disponible en la ventana), ya puede permitirse que una capa de red previamente bloqueada produzca más eventos *network_layer_ready*.

Para este protocolo damos por hecho que siempre hay tráfico de regreso en el que se pueden superponer confirmaciones de recepción. Si no lo hay, no es posible enviar confirmaciones de recepción. El protocolo 4 no necesita este supuesto debido a que envía una trama cada vez que recibe una trama, incluso si ya ha enviado la trama. En el siguiente protocolo resolveremos el problema del tráfico de una vía de una forma elegante.

Debido a que este protocolo tiene múltiples tramas pendientes, necesita lógicamente múltiples temporizadores, uno por cada trama pendiente. Cada trama termina de temporizar independientemente de todas las demás. Todos estos temporizadores pueden simularse fácilmente en software, usando un solo reloj de hardware que produzca interrupciones periódicas. Las terminaciones de temporización pendientes forman una lista enlazada, en la que cada nodo de la lista indica la cantidad de pulsos de reloj que faltan para que expire el temporizador, el número de la trama temporizada y un apuntador al siguiente nodo.

Como ilustración de una manera de implementar los temporizadores, considere el ejemplo de la figura 3-18(a). Suponga que el reloj pulsa una vez cada 100 mseg. Inicialmente la hora real es 10:00:00.0 y hay tres terminaciones pendientes, a las 10:00:00.5, 10:00:01.3 y 10:00:01.9.

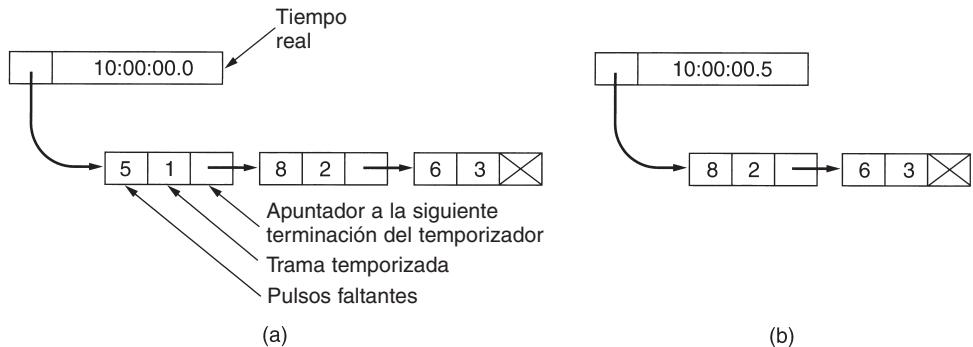


Figura 3-18. Simulación de múltiples temporizadores en software.

Cada vez que pulsa el reloj de hardware, se actualiza el tiempo real y el contador de pulsos a la cabeza de la lista se decrementa. Al llegar a cero el contador de pulsos, se causa una terminación y se retira el nodo de la lista, como se muestra en la figura 3-18(b). Aunque esta organización requiere que la lista se examine al llamar a *start_timer* o a *stop_timer*, no requiere mucho trabajo por pulso. En el protocolo 5 estas dos rutinas tienen un parámetro que indica la trama a temporizar.

3.4.3 Protocolo que utiliza repetición selectiva

El protocolo 5 funciona bien si los errores son poco frecuentes, pero si la línea es mala, se desperdicia mucho ancho de banda en las tramas retransmitidas. Una estrategia alterna para el manejo de errores es permitir que el receptor acepte y coloque en búferes las tramas que siguen a una trama dañada o perdida. Tal protocolo no rechaza tramas simplemente porque se dañó o se perdió una trama anterior.

En este protocolo, tanto el emisor como el receptor mantienen una ventana de números de secuencia aceptables. El tamaño de la ventana del emisor comienza en 0 y crece hasta un máximo predefinido, *MAX_SEQ*. La ventana del receptor, en cambio, siempre es de tamaño fijo e igual a *MAX_SEQ*. El receptor tiene un búfer reservado para cada número de secuencia en su ventana fija. Cada búfer tiene un bit asociado (*arrived*) que indica si el búfer está lleno o vacío. Cuando llega una trama, su número de secuencia es revisado por la función *between* para ver si cae dentro de la ventana. De ser así, y no ha sido recibida aún, se acepta y almacena. Esta acción se lleva a cabo sin importar si la trama contiene el siguiente paquete esperado por la capa de red. Por supuesto, debe mantenerse dentro de la capa de enlace de datos sin entregarse a la capa de red hasta que todas las tramas de número menor hayan sido entregadas a la capa de red en el orden correcto. En la figura 3-19 se presenta un protocolo que usa este algoritmo.

```

/* El protocolo 6 (repetición selectiva) acepta tramas en desorden y pasa paquetes en
orden a la capa de red. Cada trama pendiente tiene un temporizador asociado. Cuando
el temporizador expira, a diferencia de lo que ocurre en el protocolo 5, sólo se
retransmite esa trama, no todas las que están pendientes. */

#define MAX_SEQ 7           /* debe ser 2^n - 1 */
#define NR_BUFS ((MAX_SEQ + 1)/2)
typedef enum {frame_arrival, cksum_err, timeout, network_layer_ready, ack_timeout}
    event_type;
#include "protocol.h"
boolean no_nak = true;          /* aún no se ha enviado un nak */
seq_nr oldest_frame = MAX_SEQ + 1; /* el valor inicial es sólo para el simulador */
static boolean between(seq_nr a, seq_nr b, seq_nr c)
{
/* Parecido a lo que ocurre en el protocolo 5, pero más corto y confuso. */
    return ((a <= b) && (b < c)) || ((c < a) && (a <= b)) || ((b < c) && (c < a));
}
static void send_frame(frame_kind fk, seq_nr frame_nr, seq_nr frame_expected, packet
    buffer[])
{
/* Construye y envía una trama de datos, ack o nak. */
    frame s;                  /* variable de trabajo */
    s.kind = fk;               /* kind == datos, ack o nak */
    if (fk == data) s.info = buffer[frame_nr % NR_BUFS];
    s.seq = frame_nr;          /* sólo tiene importancia para las tramas de datos */
    s.ack = (frame_expected + MAX_SEQ) % (MAX_SEQ + 1);
    if (fk == nak) no_nak = false; /* un nak por trama, por favor */
    to_physical_layer(&s);    /* transmite la trama */
    if (fk == data) start_timer(frame_nr % NR_BUFS);
    stop_ack_timer();          /* no es necesario para tramas ack separadas */
}
void protocol6(void)
{
    seq_nr ack_expected;        /* límite inferior de la ventana del emisor */
    seq_nr next_frame_to_send;  /* límite superior de la ventana del emisor + 1 */
    seq_nr frame_expected;      /* límite inferior de la ventana del receptor */
    seq_nr too_far;             /* límite superior de la ventana del receptor + 1 */
    int i;                      /* índice en el grupo de búferes */
    frame r;                   /* variable de trabajo*/
    packet out_buf[NR_BUFS];    /* búferes para el flujo de salida */
    packet in_buf[NR_BUFS];     /* búferes para el flujo de entrada */
    boolean arrived[NR_BUFS];   /* mapa de bits de entrada */
    seq_nr nbuffered;          /* cuántos búferes de salida se utilizan
                                actualmente */

    event_type event;

    enable_network_layer();     /* inicializar */
    ack_expected = 0;           /* siguiente ack esperada en el flujo de entrada */
    next_frame_to_send = 0;      /* número de la siguiente trama de salida */
    frame_expected = 0;
    too_far = NR_BUFS;
    nbuffered = 0;              /* inicialmente no hay paquetes en el búfer */
    for (i = 0; i < NR_BUFS; i++) arrived[i] = false;

    while (true) {
        wait_for_event(&event);      /* cinco posibilidades: vea event_type al principio */
        switch(event) {
            case network_layer_ready: /* acepta, guarda y transmite una trama nueva */
                nbuffered = nbuffered + 1; /* expande la ventana */

```

```

from_network_layer(&out_buf[next_frame_to_send % NR_BUFS]); /* obtiene un
                                                               paquete nuevo */
send_frame(data, next_frame_to_send, frame_expected, out_buf); /* transmite la
                                                               trama */
inc(next_frame_to_send); /* avanza el límite superior de la ventana */
break;

case frame_arrival:           /* ha llegado una trama de datos o de control */
from_physical_layer(&r); /* obtiene una trama entrante de la capa física */
if (r.kind == data) {
    /* Ha llegado una trama no dañada. */
    if ((r.seq != frame_expected) && no_nak)
        send_frame(nak, 0, frame_expected, out_buf); else start_ack_timer();
    if (between(frame_expected, r.seq, too_far) && (arrived[r.seq%NR_BUFS]
        == false)) {
        /* Las tramas se podrían aceptar en cualquier orden. */
        arrived[r.seq % NR_BUFS] = true; /* marca como lleno el búfer */
        in_buf[r.seq % NR_BUFS] = r.info; /* inserta datos en el búfer */
        while (arrived[frame_expected % NR_BUFS]) {
            /* Pasa tramas y avanza la ventana. */
            to_network_layer(&in_buf[frame_expected % NR_BUFS]);
            no_nak = true;
            arrived[frame_expected % NR_BUFS] = false;
            inc(frame_expected); /* avanza el límite inferior de la
                                   ventana del receptor */
            inc(too_far); /* avanza el límite superior de la
                           ventana del receptor */
            start_ack_timer(); /* para saber si es necesaria una ack
                               separada */
        }
    }
    if((r.kind==nak) && between(ack_expected,(r.ack+1)%(MAX_SEQ+1),
        next_frame_to_send))
        send_frame(data, (r.ack+1) % (MAX_SEQ + 1), frame_expected, out_buf);
    while (between(ack_expected, r.ack, next_frame_to_send)) {
        nbuffered = nbuffered - 1; /* maneja la ack superpuesta */
        stop_timer(ack_expected % NR_BUFS); /* la trama llega intacta */
        inc(ack_expected); /* avanza el límite inferior de la ventana del emisor */
    }
    break;
}

case cksum_err:
    if (no_nak) send_frame(nak, 0, frame_expected, out_buf); /* trama dañada */
    break;

case timeout:
    send_frame(data, oldest_frame, frame_expected, out_buf); /* hacemos que expire
                                                               el temporizador */
    break;

case ack_timeout:
    send_frame(ack,0,frame_expected, out_buf); /* expira el temporizador de ack;
                                               envía ack */
}

if (nbuffered < NR_BUFS) enable_network_layer(); else disable_network_layer();
}
}

```

Figura 3-19. Protocolo de ventana corrediza con repetición selectiva.

La recepción no secuencial introduce ciertos problemas que no se presentan en los protocolos en los que las tramas sólo se aceptan en orden. Podemos ilustrar el problema fácilmente con un ejemplo. Suponga que tenemos un número de secuencia de tres bits, por lo que se permite al emisor enviar hasta siete tramas antes de que se le exija que espere una confirmación de recepción. Inicialmente las ventanas del emisor y del receptor están como se muestra en la figura 3-20(a). El emisor ahora transmite las tramas 0 a 6. La ventana del receptor le permite aceptar cualquier trama con un número de secuencia entre 0 y 6, inclusive. Las siete tramas llegan correctamente, por lo que el receptor confirma su recepción y avanza su ventana para permitir la recepción de 7, 0, 1, 2, 3, 4 o 5, como se muestra en la figura 3-20(b). Los siete búferes se marcan como vacíos.

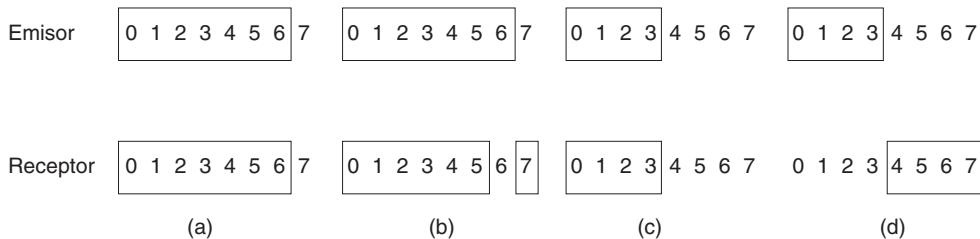


Figura 3-20. (a) Situación original con una ventana de tamaño 7. (b) Después de que se han enviado y recibido siete tramas, pero su recepción no se ha confirmado. (c) Situación inicial con un tamaño de ventana de 4. (d) Despues de que se han enviado y recibido cuatro tramas, pero su recepción no se ha confirmado.

Es en este punto en el que aparece un desastre en la forma de un rayo que cae en el poste telefónico, borrando todas las confirmaciones de recepción. En algún momento termina el temporizador del emisor y retransmite la trama 0. Cuando esta trama llega al receptor, se efectúa una verificación para saber si está dentro de la ventana del receptor. Desgraciadamente, en la figura 3-20(b), la trama 0 está dentro de la nueva ventana, por lo que se acepta. El receptor envía una confirmación de recepción, superpuesta, para la trama 6, ya que se han recibido de la 0 a la 6.

El emisor se entera con beneplácito que todas sus tramas transmitidas han llegado de manera correcta, por lo que avanza su ventana y envía de inmediato las tramas 7, 0, 1, 2, 3, 4 y 5. El receptor aceptará la trama 7 y el paquete de ésta se pasará directamente a la capa de red. Inmediatamente después, la capa de enlace de datos receptora revisa si ya tiene una trama 0 válida, descubre que sí y pasa el paquete que contiene a la capa de red. En consecuencia, la capa de red obtiene un paquete incorrecto, y falla el protocolo.

La esencia del problema es que una vez que el receptor ha avanzado su ventana, el nuevo intervalo de números de secuencia válidos se traslape con el anterior. En consecuencia, el siguiente grupo de tramas podría ser de tramas duplicadas (si se perdieron todas las confirmaciones de recepción) o de nuevas (si se recibieron todas las confirmaciones de recepción). El pobre receptor no tiene manera de distinguir entre estos dos casos.

La salida de este dilema es asegurarse que, una vez que el emisor haya avanzado su ventana, no haya traslape con la ventana original. Para asegurarse de que no haya traslape, el tamaño máximo de la ventana debe ser cuando menos de la mitad del intervalo de los números de secuencia, como en las figuras 3-20(c) y 3-20(d). Por ejemplo, si se utilizan 4 bits para los números de secuencia, éstos tendrán un intervalo de 0 a 15. Sólo ocho tramas sin confirmación de recepción deben estar pendientes en cualquier instante. De esa manera, si el receptor apenas ha aceptado las tramas 0 a 7 y ha avanzado su ventana para permitir la aceptación de las tramas 8 a 15, puede distinguir sin ambigüedades si las tramas subsiguientes son retransmisiones (0 a 7) o nuevas (8 a 15). En general, el tamaño de la ventana para el protocolo 6 será $(MAX_SEQ + 1)/2$. Por lo tanto, para números de secuencia de 3 bits, el tamaño de ventana es 4.

Una pregunta interesante es: ¿cuántos búferes deberá tener el receptor? En ninguna circunstancia puede aceptar tramas cuyos números de secuencia estén por debajo del extremo inferior o por encima del extremo superior de la ventana. En consecuencia, el número de búferes necesarios es igual al tamaño de la ventana, no al intervalo de números de secuencia. En el ejemplo anterior de un número de secuencia de 4 bits, se requieren ocho búferes, numerados del 0 al 7. Cuando llega la trama i , se coloca en el búfer $i \bmod 8$. Observe que, aun cuando i e $(i + 8) \bmod 8$ están “compitiendo” por el mismo búfer, nunca están dentro de la ventana al mismo tiempo, pues ello implicaría un tamaño de ventana de cuando menos 9.

Por la misma razón, el número de temporizadores requeridos es igual al número de búferes, no al tamaño del espacio de secuencia. Efectivamente, hay un temporizador asociado a cada búfer. Cuando termina el temporizador, el contenido del búfer se retransmite.

En el protocolo 5 se supone de manera implícita que el canal está fuertemente cargado. Cuando llega una trama, no se envía de inmediato la confirmación de recepción. En cambio, esta última se superpone en la siguiente trama de datos de salida. Si el tráfico de regreso es ligero, la confirmación de recepción se detendrá durante un periodo largo. Si hay mucho tráfico en una dirección y no hay tráfico en la otra, sólo se envían MAX_SEQ paquetes y luego se bloquea el protocolo, que es por lo cual dimos por hecho que siempre había tráfico de regreso.

En el protocolo 6 se corrige este problema. Tras llegar una trama de datos en secuencia, se arranca un temporizador auxiliar mediante *start_ack_timer*. Si no se ha presentado tráfico de regreso antes de que termine este temporizador, se envía una trama de confirmación de recepción independiente. Una interrupción debida al temporizador auxiliar es conocida como evento *ack_timeout*. Con este arreglo, ahora es posible el flujo de tráfico unidireccional, pues la falta de tramas de datos de regreso a las que puedan superponerse las confirmaciones de recepción ya no es un obstáculo. Sólo existe un temporizador auxiliar, y si *start_ack_timer* se invoca mientras el temporizador se está ejecutando, se restablece a un intervalo completo de temporización de la confirmación de recepción.

Es indispensable que el tiempo asociado al temporizador auxiliar sea notablemente más corto que el del temporizador usado para la terminación de tramas de datos. Esta condición es necesaria para asegurarse de que la confirmación de recepción de una trama correctamente recibida lleve antes de que el emisor termine su temporización y retransmita la trama.

El protocolo 6 utiliza una estrategia más eficiente que el 5 para manejar los errores. Cuando el receptor tiene razones para suponer que ha ocurrido un error, envía al emisor una trama de confirmación de recepción negativa (NAK). Tal trama es una solicitud de retransmisión de la trama especificada en la NAK. Hay dos casos en los que el receptor debe sospechar: cuando llega una trama dañada, o cuando llega una trama diferente de la esperada (pérdida potencial de la trama). Para evitar hacer múltiples solicitudes de retransmisión de la misma trama perdida, el receptor debe saber si ya se ha enviado una NAK para una trama dada. La variable *no_nak* del protocolo 6 es true si no se ha enviado todavía ninguna NAK para *frame_expected*. Si la NAK se altera o se pierde no hay un daño real, pues de todos modos el emisor terminará su temporizador en algún momento y retransmitirá la trama perdida. Si llega la trama equivocada después de haberse enviado y de perderse una NAK, *no_nak* será true y el temporizador auxiliar arrancará. Cuando termine, se enviará una ACK para resincronizar el estado actual del emisor con el del receptor.

En algunas situaciones, el tiempo requerido para que una trama se propague a su destino, sea procesada ahí y regrese la confirmación de recepción es (casi) constante. En estos casos, el emisor puede ajustar su temporizador para que apenas sea mayor que el intervalo de tiempo esperado entre el envío de una trama y la recepción de su confirmación. Sin embargo, si este tiempo es muy variable, el emisor se enfrenta a la decisión de establecer el intervalo en un valor pequeño (y arriesgarse a retransmisiones innecesarias) o de establecerlo en un valor grande (quedándose inactivo por un periodo largo tras un error).

Ambas opciones desperdician ancho de banda. Si el tráfico de regreso es esporádico, el tiempo antes de la confirmación de recepción será irregular, siendo más corto al haber tráfico de regreso y mayor al no haberlo. El tiempo de procesamiento variable en el receptor también puede ser un problema aquí. En general, cuando la desviación estándar del intervalo de confirmación de recepción es pequeña en comparación con el intervalo mismo, el temporizador puede hacerse “justo” y las NAKs no serán útiles. De otra manera, el temporizador deberá hacerse “holgado”, y las NAKs pueden acelerar apreciablemente la retransmisión de tramas perdidas o dañadas.

Un aspecto muy relacionado con el asunto de la terminación de los temporizadores y las NAKs es cómo determinar qué trama causó una terminación del temporizador. En el protocolo 5 siempre es *ack_expected*, pues es la más antigua. En el protocolo 6 no hay ninguna manera sencilla de determinar quién ha terminado el temporizador. Suponga que ya se transmitieron las tramas 0 a 4, de modo que la lista de tramas pendientes es 01234, en orden de la más antigua a la más nueva. Ahora imagine que 0 termina su temporizador, se transmite 5 (una trama nueva), 1 termina su temporizador, 2 termina su temporizador y se transmite 6 (otra trama nueva). En este punto, la lista de tramas pendientes es 3405126, de la más antigua a la más nueva. Si todo el tráfico de entrada se pierde durante un rato (es decir, las tramas que llevan las confirmaciones de recepción), las siete tramas pendientes terminarán su temporizador en ese orden.

Para evitar que el ejemplo se complique aún más, no hemos mostrado la administración de los temporizadores. En cambio, suponemos que la variable *oldest_frame* se establece en el momento de terminación del temporizador para indicar la trama cuyo temporizador ha terminado.

3.5 VERIFICACIÓN DE LOS PROTOCOLOS

Los protocolos que se utilizan en la práctica, y los programas que los implementan, con frecuencia son complicados. En consecuencia, se requiere mucha investigación para encontrar técnicas matemáticas formales con las cuales especificar y verificar los protocolos. En las siguientes secciones veremos algunos modelos y técnicas. Aunque estamos estudiándolos en el contexto de la capa de enlace de datos, también son aplicables a otras capas.

3.5.1 Modelos de máquinas de estado finito

Un concepto clave empleado en muchos modelos de protocolos es el de la **máquina de estados finitos**. Con esta técnica, cada **máquina de protocolo** (es decir, emisor o receptor) siempre está en un estado específico en cualquier instante. Su estado consiste en todos los valores de sus variables, incluido el contador de programa.

En la mayoría de los casos puede agruparse un gran número de estados a fin de analizarlos. Por ejemplo, considerando el receptor del protocolo 3, podemos abstraer dos estados importantes de todos los posibles: en espera de la trama 0 y en espera de la trama 1. Todos los demás estados pueden considerarse como transitorios: pasos en el camino hacia uno de los estados principales. Por lo general, los estados se escogen para que sean aquellos instantes en que la máquina de protocolo está esperando que ocurra el siguiente evento [es decir, la ejecución de la llamada de procedimiento *wait(event)* en nuestros ejemplos]. En este punto, el estado de la máquina de protocolo está determinado por completo por los estados de sus variables. El número de estados es entonces 2^n , donde n es el número de bits necesarios para representar todas las variables combinadas.

El estado del sistema completo es la combinación de todos los estados de las dos máquinas de protocolos y del canal. El estado del canal está determinado por su contenido. Usando nuevamente el protocolo 3 como ejemplo, el canal tiene cuatro posibles estados: una trama cero o una trama 1 viajando del emisor al receptor, una trama de confirmación de recepción que va en el otro sentido o un canal vacío. Si modelamos el emisor y el receptor como si ambos tuvieran dos estados, todo el sistema tiene 16 estados diferentes.

Vale la pena decir algo sobre el estado del canal. El concepto de una trama que está “en el canal” es, por supuesto, una abstracción. Lo que queremos decir en realidad es que es posible que una trama se haya recibido, pero no procesado, en el destino. Una trama permanece “en el canal” hasta que la máquina de protocolo ejecuta *FromPhysicalLayer* y la procesa.

De cada estado hay cero o más **transiciones** posibles a otros estados. Las transiciones ocurren cuando sucede algún evento. Para una máquina de protocolo, podría ocurrir una transición al enviar una trama, al llegar una trama, al terminar un temporizador, al ocurrir una interrupción, etcétera. Para el canal, los eventos típicos son la inserción de una trama nueva en el canal por una máquina de protocolo, la entrega de una trama a una máquina de protocolo o la pérdida de una trama debido a una ráfaga de ruido. Dada una descripción completa de las máquinas de protocolo y las características del canal, es posible dibujar un grafo dirigido que muestre todos los estados como nodos y las transiciones como arcos dirigidos.

Un estado en particular se designa como **estado inicial**. Este estado corresponde a la descripción del sistema cuando comienza a funcionar, o en algún punto conveniente poco después. Desde el estado inicial pueden alcanzarse algunos, o quizás todos, los demás estados mediante una secuencia de transiciones. Usando técnicas bien conocidas de la teoría de grafos (por ejemplo, calculando el cierre transitorio de un grafo), es posible determinar los estados que son alcanzables y los que no. Esta técnica se denomina **análisis de asequibilidad** (Lin y cols., 1987). Este análisis puede ser útil para determinar si un protocolo es correcto o no.

Formalmente, un modelo de máquina de estados finitos de un protocolo se puede considerar como un cuádruple (S, M, I, T), donde:

S es el conjunto de estados en que pueden estar los procesos y el canal.

M es el conjunto de tramas que pueden intercambiarse a través del canal.

I es el conjunto de estados iniciales de los procesos.

T es el conjunto de transiciones entre los estados.

Al principio todos los procesos están en sus estados iniciales. Entonces comienzan a ocurrir eventos, como tramas que se vuelven disponibles para transmisión o temporizadores que expiran. Cada evento puede causar que uno de los procesos o el canal emprenda una acción y cambie a un estado nuevo. Enumerando cuidadosamente cada sucesor posible para cada estado, podemos construir el grafo de asequibilidad y analizar el protocolo.

El análisis de asequibilidad puede servir para detectar una variedad de errores en la especificación del protocolo. Por ejemplo, si es posible que ocurra cierta trama en cierto estado y la máquina de estados finitos no indica la acción a tomar, la especificación es errónea (está incompleta). Si existe un grupo de estados para los que no hay salida y de los que no se puede avanzar (es decir, ya no es posible recibir tramas correctas), tenemos otro error (bloqueo irreversible). Un error menos grave es una especificación de protocolo que indica la manera de manejar un evento en un estado en que el evento no puede ocurrir (transición ajena). También pueden detectarse otros errores.

Como ejemplo de modelo de máquina de estados finitos, considere la figura 3-21(a). Este grafo corresponde al protocolo 3, como se describió antes: cada máquina de protocolo tiene dos estados y el canal tiene cuatro. Hay en total 16 estados, no todos alcanzables desde el inicial. Los inalcanzables no se muestran en la figura. Los errores de suma de verificación se ignoran aquí por simplicidad.

Se utilizan tres caracteres para etiquetar cada estado, SRC , donde S es 0 o 1 y corresponde a la trama que el emisor está tratando de enviar; R también es 0 o 1 y corresponde a la trama que el receptor espera, y C es 0, 1, A o vacío (-) y corresponde al estado del canal. En este ejemplo, el estado inicial se ha elegido como (000). En otras palabras, el emisor sólo ha enviado la trama 0, el receptor espera la trama 0 y ésta está actualmente en el canal.

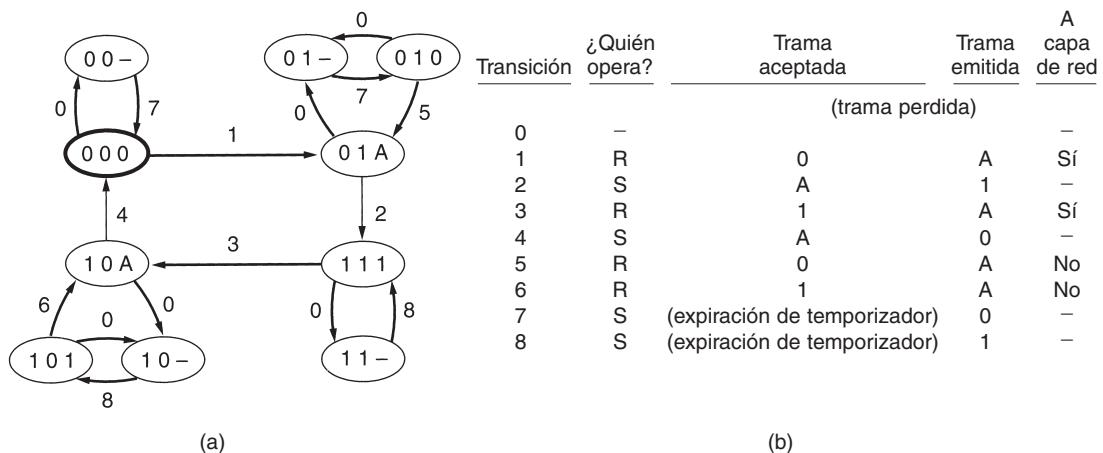


Figura 3-21. (a) Diagrama de estado para el protocolo 3. (b) Transiciones.

En la figura 3-21 se muestran nueve tipos de transiciones. La transición 0 consiste en la pérdida del contenido del canal. La transición 1 consiste en que el canal entregue de manera correcta el paquete 0 al receptor, y que éste cambie a continuación su estado para esperar la trama 1 y emitir una confirmación de recepción. La transición 1 también tiene que ver con que el receptor entregue el paquete 0 a la capa de red. Las otras transiciones se listan en la figura 3-21(b). La llegada de una trama con un error de suma de verificación no se ha mostrado, pues no cambia el estado (en el protocolo 3).

Durante la operación normal, las transiciones 1, 2, 3 y 4 se repiten en orden una y otra vez. En cada ciclo se entregan dos paquetes, regresando el emisor al estado original de tratar de enviar una trama nueva con un número de secuencia 0. Si el canal pierde la trama 0, hace una transición del estado (000) al estado (00-). En algún momento, el temporizador del emisor expira (transición 7) y el sistema regresa a (000). La pérdida de una confirmación de recepción es más complicada, pues requiere dos transiciones, 7 y 5 u 8 y 6, para reparar el daño.

Una de las propiedades que debe tener un protocolo con un número de secuencia de 1 bit es que, sin importar la secuencia de eventos que ocurra, el receptor nunca debe entregar dos paquetes impares sin haber intervenido un paquete par, y viceversa. En el grafo de la figura 3-21 podemos ver que este requisito puede establecerse más formalmente como “no debe existir ninguna ruta desde el estado inicial en la que sucedan dos transiciones 1 sin que ocurra una transición 3 entre ellas, o al revés”. En la figura se puede ver que el protocolo es correcto en este aspecto.

Otro requisito semejante es que no debe haber rutas en las que el emisor pueda cambiar de estado dos veces (por ejemplo, de 0 a 1 y de regreso a 0) mientras el estado del receptor permanezca constante. De existir tal ruta, en la secuencia correspondiente de eventos se perderían dos tramas sin posibilidad de recuperación y sin que el receptor se diera cuenta. La secuencia de paquetes entregados tendrá un hueco no detectado de dos paquetes.

Otra propiedad importante de un protocolo es la ausencia de bloqueos irreversibles. Un **bloqueo irreversible** es una situación en la que el protocolo no puede seguir avanzando (es decir, entregando paquetes a la capa de red), sea cual sea la secuencia de eventos que ocurra. En términos del modelo gráfico, un bloqueo irreversible se caracteriza por la existencia de un subconjunto de estados que es alcanzable desde el estado inicial y que tiene dos propiedades:

1. No hay transición hacia fuera del subconjunto.
2. No hay transiciones en el subconjunto que causen un avance.

Una vez en el estado de bloqueo irreversible, el protocolo permanece ahí eternamente. De nuevo, es fácil ver en el grafo que el protocolo 3 no sufre de bloqueos irreversibles.

3.5.2 Modelos de red de Petri

La máquina de estados finitos no es la única técnica para especificar protocolos formalmente. En esta sección describiremos una técnica completamente diferente, la **red de Petri** (Danthine, 1980). Una red de Petri tiene cuatro elementos básicos: lugares, transiciones, arcos y *tokens*. Un **lugar** representa un estado en el que puede estar parte del sistema. En la figura 3-22 se muestra una red de Petri con dos lugares, *A* y *B*, que aparecen como círculos. El sistema actualmente está en el estado *A*, indicado por el **token** (punto grueso) en el lugar *A*. Se utiliza una barra horizontal o vertical para indicar una **transición**. Cada transición tiene cero o más **arcos de entrada**, que llegan de sus lugares de entrada, y cero o más **arcos de salida**, que van a sus lugares de salida.

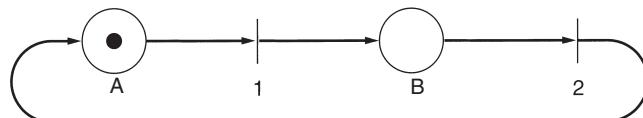


Figura 3-22. Red de Petri con dos lugares y dos transiciones.

Se **habilita** una transición si hay cuando menos un *token* de entrada en cada uno de sus lugares de entrada. Cualquier transición habilitada puede **dispararse** a voluntad, quitando un *token* de cada lugar de entrada y depositando un *token* en cada lugar de salida. Si el número de arcos de entrada no es igual al número de arcos de salida, no se conservarán los *tokens*. Si se habilitan dos o más transiciones, cualquiera de ellas puede dispararse. La decisión de disparo de una transición es indeterminada, por lo que las redes de Petri son útiles para modelar protocolos. La red de Petri de la figura 3-22 es determinista y puede servir para modelar cualquier proceso de dos fases (por ejemplo, el comportamiento de un bebé: comer, dormir, comer, dormir, y así sucesivamente). Como con todas las herramientas de modelado, se omiten los detalles innecesarios.

En la figura 3-23 se presenta el modelo de red de Petri de la figura 3-22. A diferencia del modelo de máquina de estados finitos, aquí no hay estados compuestos; el estado del emisor, el estado del canal y el estado del receptor se representan por separado. Las transiciones 1 y 2 corresponden al

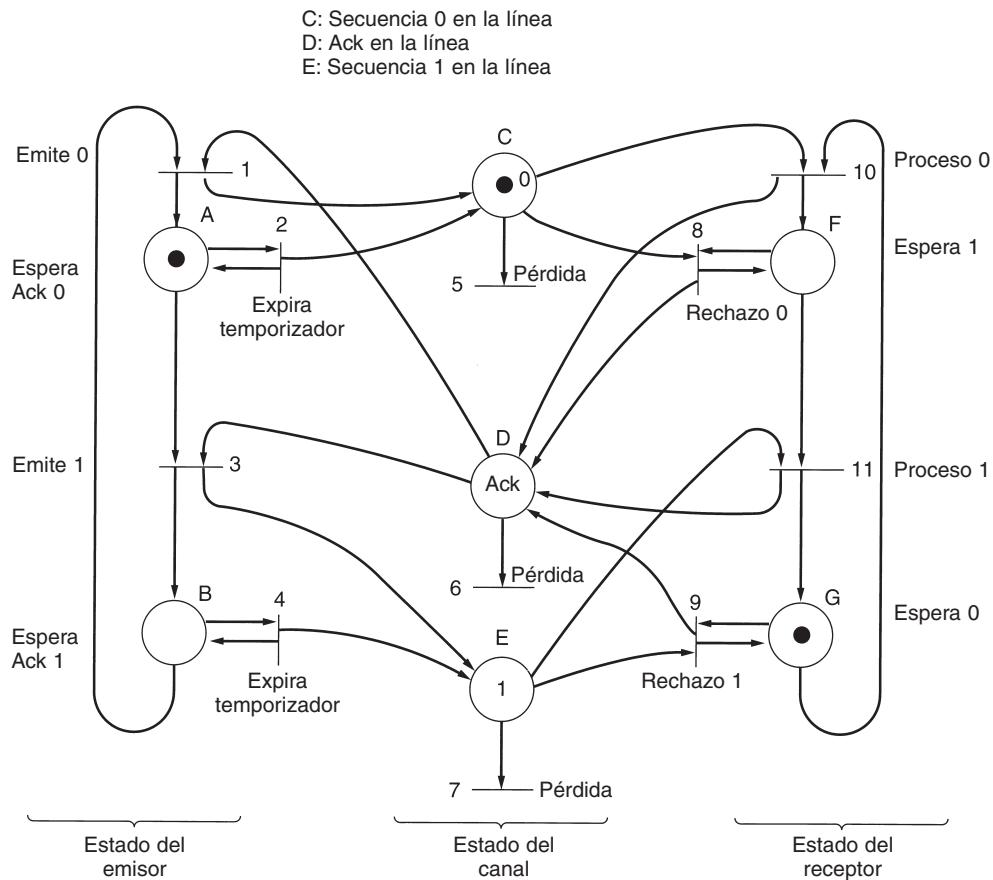


Figura 3-23. Modelo de red de Petri para el protocolo 3.

envío de la trama 0 por el emisor, normalmente, y al momento de una expiración de temporizador, respectivamente. Las transiciones 3 y 4 son análogas para la trama 1. Las transiciones 5, 6 y 7 corresponden a la pérdida de la trama 0, de una confirmación de recepción y de la trama 1, respectivamente. Las transiciones 8 y 9 ocurren cuando una trama de datos con un número de secuencia equivocado llega al receptor. Las transiciones 10 y 11 representan la llegada al receptor de la siguiente trama en la secuencia y su entrega a la capa de red.

Las redes de Petri pueden servir para detectar fallas de protocolo de una manera parecida a como se hace con máquinas de estados finitos. Por ejemplo, si alguna secuencia de disparo incluyera la transición 10 dos veces sin interponerse la transición 11, el protocolo sería incorrecto. El concepto de bloqueo irreversible en una red de Petri es semejante a su contraparte en una máquina de estados finitos.

Las redes de Petri pueden representarse convenientemente en una forma algebraica semejante a una gramática. Cada transición contribuye con una regla a la gramática. Cada regla especifica lugares de entrada y salida de la transición. Debido a que la figura 3-23 tiene 11 transiciones, su

gramática tiene 11 reglas, numeradas del 1 al 11, y cada una corresponde a la transición del mismo número. La gramática de la red de Petri de la figura 3-23 es como se muestra a continuación:

- 1: BD → AC
- 2: A → A
- 3: AD → BE
- 4: B → B
- 5: C →
- 6: D →
- 7: E →
- 8: CF → DF
- 9: EG → DG
- 10: CG → DF
- 11: EF → DG

Es interesante mencionar cómo hemos reducido un protocolo complejo a 11 reglas simples de gramática que pueden ser manipuladas fácilmente por un programa de computadora.

El estado actual de la red de Petri se representa como una colección desordenada de lugares, cada uno representado en la colección dependiendo de los *tokens* que tenga. Cualquier regla con lugares presentes en su izquierda puede ser disparada, removiendo aquellos lugares de su estado actual y agregando sus lugares de salida al estado actual. El marcado de la figura 3-23 es *ACG* (es decir, tanto *A*, como *C* y *G* tienen un *token*). En consecuencia, las reglas 2, 5 y 10 están habilitadas y cualquiera de ellas puede aplicarse, lo que lleva a un nuevo estado (posiblemente con el mismo marcado que el original). En contraste, la regla 3 (*AD* → *BE*) no puede aplicarse porque *D* no está marcada.

3.6 EJEMPLOS DE PROTOCOLOS DE ENLACE DE DATOS

En las siguientes secciones examinaremos varios protocolos de enlace de datos de amplio uso. El primero, HDLC, es un protocolo clásico orientado a bits cuyas variantes se han utilizado durante décadas en muchas aplicaciones. El segundo, PPP, es un protocolo de enlace utilizado para conectar a Internet computadoras domésticas.

3.6.1 HDLC—Control de Enlace de Datos de Alto Nivel

En esta sección examinaremos un grupo de protocolos íntimamente relacionados que, a pesar de ser un poco antiguos, se siguen utilizando ampliamente en redes de todo el mundo. Todas se derivan del primer protocolo de enlace de datos usado en los *mainframes* de IBM: el protocolo **SDLC (Control Síncrono de Enlace de Datos)**. Después de desarrollar SDLC, IBM lo sometió al ANSI y a la ISO para su aceptación como estándar de Estados Unidos e internacional, respectivamente. El ANSI lo modificó convirtiéndolo en **ADCCP (Procedimiento Avanzado de Control de Comunicación de Datos)**, y la ISO lo modificó para convertirlo en **HDLC (Control de Enlace de Datos de Alto Nivel)**. Luego, el CCITT adoptó y modificó HDLC para su **LAP (Procedimiento de Acceso al Enlace)** como parte del estándar de interfaz de red X.25, pero después lo modificó nuevamente a **LAPB** para hacerlo más compatible con una versión posterior de

HDLC. Lo agradable de los estándares es que hay muchos de donde escoger. Es más, si no le gusta ninguno de ellos, simplemente puede sentarse a esperar el modelo del año próximo.

Todos estos protocolo se basan en el mismo principio. Todos son orientados a bits y usan el relleno de bits para lograr la transparencia de los datos. Difieren sólo en aspectos menores, aunque irritantes. El análisis de protocolos orientados a bits que haremos a continuación pretende ser una introducción general. Si desea detalles específicos de cualquier protocolo, consulte la definición adecuada.

Todos los protocolos orientados a bits utilizan la estructura de trama mostrada en la figura 3-24. El campo de *Dirección* es de importancia primordial en las líneas con múltiples terminales, pues sirve para identificar una de las terminales. En líneas punto a punto a veces se usan para distinguir los comandos de las respuestas.

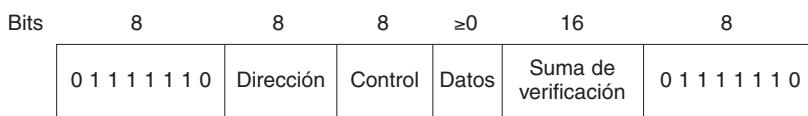


Figura 3-24. Formato de trama para protocolos orientados a bits.

El campo de *Control* se utiliza para números de secuencia, confirmaciones de recepción y otros propósitos, como se explicará más adelante.

El campo de *Datos* puede contener cualquier información y puede tener una longitud arbitraria, aunque la eficiencia de la suma de verificación disminuye conforme el tamaño de la trama aumenta, debido a la mayor probabilidad de múltiples errores en ráfaga.

El campo de *Suma de verificación* es un código de redundancia cíclica que utiliza la técnica que examinamos en la sección 3.2.2.

La trama está delimitada por otra secuencia de bandera (01111110). En líneas punto a punto inactivas se transmiten secuencias de bandera continuamente. La trama mínima contiene tres campos y un total de 32 bits, y excluye a las banderas de ambos lados.

Hay tres tipos de tramas: **de información, de supervisión y no numeradas**. El contenido del campo de *Control* para estos tres tipos se muestra en la figura 3-25. El protocolo emplea una ventana corrediza, con un número de secuencia de 3 bits. En cualquier momento pueden estar pendientes hasta siete tramas sin confirmación de recepción. El campo *Secuencia* de la figura 3-25(a) es el número de secuencia de la trama. El campo *Siguiente* es una confirmación de recepción superpuesta. Sin embargo, todos los protocolos se apegan a la convención de que, en lugar de superponer el número de la última trama recibida correctamente, usan el número de la primera trama no recibida (es decir, la siguiente trama esperada). La decisión de utilizar la última trama recibida o la siguiente trama esperada es arbitraria; no importa la convención que se utilice, siempre y cuando se use con consistencia.

El bit *P/F* (*S/F*) significa *Sondeo/Final (Poll/Final)*. Se utiliza cuando una computadora (o concentrador) está sondeando un grupo de terminales. Cuando se usa como *P*, la computadora está invitando a la terminal a enviar datos. Todas las tramas enviadas por la terminal, excepto la última, tienen el bit *P/F* establecido en *P*. El último se establece en *F*.

Bits	1	3	1	3
(a)	0	Secuencia	P/F	Siguiente
(b)	1	0	Tipo	P/F
(c)	1	1	Tipo	P/F
				Modificado

Figura 3-25. Campo de Control de (a) una trama de información, (b) una trama de supervisión y (c) una trama no numerada.

En algunos de los protocolos el bit *P/F* sirve para obligar a la otra máquina a enviar de inmediato una trama de supervisión, en lugar de esperar tráfico de regreso al cual superponer la información de la ventana. El bit también tiene algunos usos menores en conexión con las tramas sin número.

Los diferentes tipos de tramas de supervisión se distinguen por el campo de *Tipo*. El tipo 0 es una trama de confirmación de recepción (oficialmente llamada RECEIVE READY) que sirve para indicar la siguiente trama esperada. Ésta se usa cuando no hay tráfico de regreso que se pueda aprovechar para superponer confirmaciones de recepción.

El tipo 1 es una trama de confirmación de recepción negativa (oficialmente llamada REJECT); sirve para indicar que se ha detectado un error de transmisión. El campo *siguiente* indica la primera trama en la secuencia que no se ha recibido en forma correcta (es decir, la trama a retransmitir). Se pide al emisor retransmitir todas las tramas pendientes comenzando por *siguiente*. Esta estrategia es semejante a la de nuestro protocolo 5, más que a la de nuestro protocolo 6.

El tipo 2 es RECEIVE NOT READY (receptor no listo); reconoce todas las tramas hasta, pero sin incluir, *siguiente*, al igual que RECEIVE NOT READY, pero le dice al emisor que detenga el envío. El propósito de RECEIVE NOT READY es señalar ciertos problemas temporales en el receptor, como falta de búfer, y no servir como alternativa del control de flujo de ventana corrediza. Cuando el problema se resuelve, el receptor envía RECEIVE READY, REJECT o ciertas tramas de control.

El tipo 3 es SELECTIVE REJECT; solicita la retransmisión de sólo la trama especificada. En este sentido es como nuestro protocolo 6, no como el 5, y por lo tanto es de mayor utilidad cuando el tamaño de la ventana del emisor es la mitad del espacio secuencial, o menor. Por lo tanto, si un receptor desea colocar en búferes tramas fuera de secuencia para un potencial uso futuro, puede reforzar la retransmisión de cualquier trama específica usando SELECTIVE REJECT. HDLC y ADCCP permiten este tipo de tramas, pero SDLC y LAPB no lo permiten (es decir, no hay rechazo selectivo), y las tramas de tipo 3 no están definidas.

La tercera clase de trama es la trama no numerada que a veces se usa para propósitos de control, aunque también puede servir para llevar datos cuando se solicita un servicio no confiable sin conexión. Los diferentes protocolos orientados a bits tienen diferencias considerables aquí, en contraste con los otros dos tipos, donde son casi idénticos. Hay cinco bits disponibles para indicar el tipo de trama enviada, pero no se utilizan las 32 posibilidades.

Todos los protocolos proporcionan un comando, DISC (*DISConnect*), que permite a una máquina anunciar que va a ser desactivada (por ejemplo, para mantenimiento preventivo). También cuentan con un comando que permite a una máquina que acaba de regresar y está en línea anunciar su presencia y obligar el regreso a cero de todos los números de secuencia. Este comando se llama SNRM (Establecer Modo de Respuesta Normal). Desgraciadamente, el “modo de respuesta normal” es todo menos normal. Es un modo desbalanceado (es decir, asimétrico) en el que un extremo de la línea es el maestro y el otro, el subordinado. SNRM se remonta a una época en la que “comunicación de datos” significaba una terminal no inteligente que se comunicaba con una enorme computadora *host*, lo cual evidentemente es asimétrico. Para hacer más adecuado el protocolo cuando los dos interlocutores son iguales, HDLC y LAPB cuentan con un comando adicional, SABM (Establecer Modo Asíncrono Balanceado), que reestablece la línea y declara que ambas partes son iguales. También cuentan con los comandos SABME y SNRME, que son iguales a SABM y a SNRM, respectivamente, excepto que habilitan un formato de trama extendida que maneja números de secuencia de 7 bits en lugar de 3 bits.

Un tercer comando proporcionado por todos los protocolos es FRMR (Rechazo de Trama), que sirve para indicar que ha llegado una trama con suma de verificación correcta pero semántica imposible. Ejemplos de semántica imposible son: una trama de supervisión tipo 3 en LAPB, una trama menor a 32 bits, una trama de control ilegal, la confirmación de recepción de una trama que estaba fuera de la ventana, etcétera. Las tramas FRMR contienen un campo de datos de 24 bits que indica por qué se rechazó la trama. Los datos incluyen el campo de control de la trama rechazada, los parámetros de la ventana y un conjunto de bits que señalan errores específicos.

Las tramas de control pueden perderse o dañarse, igual que las de datos, por lo que también se debe confirmar su recepción. Se proporciona una trama de control especial para este propósito, llamada UA (Confirmación de Recepción no Numerada). Debido a que sólo puede estar pendiente una trama de control, nunca hay ambigüedades sobre la trama de control que está siendo confirmada.

Las tramas de control restantes tienen que ver con la inicialización, sondeo e informe de estado. También hay una trama de control que puede contener información arbitraria, UI (Información no Numerada). Estos datos no se pasan a la capa de red, pues son para uso de la capa de enlace de datos.

A pesar de su uso extendido, HDLC está lejos de ser perfecto. En (Fiorini y cols., 1994) puede encontrar un análisis de los distintos problemas asociados a este protocolo.

3.6.2 La capa de enlace de datos en Internet

Internet consiste en máquinas individuales (*hosts* y enrutadores) y la infraestructura de comunicación que las conecta. Dentro de un solo edificio, las LANs se usan ampliamente para la interconexión, pero la mayor parte de la infraestructura de área amplia está construida a partir de líneas alquiladas punto a punto. En el capítulo 4 veremos las LANs; aquí examinaremos los protocolos de enlace de datos usados en las líneas punto a punto de Internet.

En la práctica, la comunicación punto a punto se utiliza principalmente en dos situaciones. Primero, miles de organizaciones tienen una o más LANs, cada una con cierta cantidad de *hosts*

(computadoras personales, estaciones de trabajo, servidores y otros) junto con un enrutador (o un puente, que funcionalmente es parecido). Con frecuencia, los enrutadores se interconectan mediante una LAN de red dorsal. Por lo general, todas las conexiones al mundo exterior pasan a través de uno o dos enrutadores que tienen líneas alquiladas punto a punto a enrutadores distantes. Son estos enrutadores y sus líneas arrendadas los que conforman las subredes de comunicación sobre las que está construida Internet.

La segunda situación en la que las líneas punto a punto desempeñan un papel principal en Internet son los millones de personas que tienen conexiones domésticas a Internet a través de módems y líneas de acceso telefónico. Generalmente lo que ocurre es que la PC doméstica del usuario llama a un enrutador del proveedor de servicios de Internet y luego actúa como *host* de Internet. Este método de operación no es diferente de tener una línea alquilada entre la PC y el enrutador, excepto que la conexión se termina cuando el usuario termina la sesión. En la figura 3-26 se ilustra una PC doméstica que llama a un proveedor de servicios de Internet. El módem que se muestra es externo a la computadora para enfatizar su papel, pero las computadoras modernas tienen módems internos.

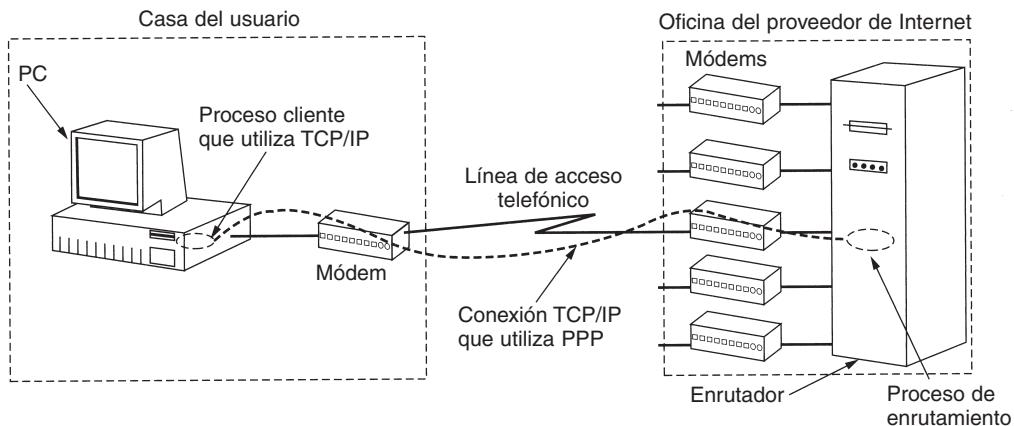


Figura 3-26. Computadora personal doméstica que funciona como *host* de Internet.

Tanto para la conexión por línea alquilada de enrutador a enrutador como para la conexión de acceso telefónico de *host* a enrutador, en la línea se requiere un protocolo de enlace de datos punto a punto para el entramado, el control de errores y las demás funciones de la capa de enlace de datos que hemos estudiado en este capítulo. El que se utiliza en Internet se conoce como PPP. A continuación lo analizaremos.

PPP—Protocolo Punto a Punto

Internet necesita de un protocolo punto a punto para diversos propósitos, entre ellos para el tráfico enrutador a enrutador y tráfico usuario doméstico a ISP. Este protocolo es **PPP (Protocolo**

Punto a Punto), que se define en el RFC 1661 y que se ha desarrollado más en varios otros RFCs (por ejemplo, los RFCs 1662 y 1663). PPP realiza detección de errores, soporta múltiples protocolos, permite la negociación de direcciones de IP en el momento de la conexión, permite la autenticación y tiene muchas otras funciones.

PPP proporciona tres características:

1. Un método de entramado que delinea sin ambigüedades el final de una trama y el inicio de la siguiente. El formato de trama también maneja la detección de errores.
2. Un protocolo de control de enlace para activar líneas, probarlas, negociar opciones y desactivarlas ordenadamente cuando ya no son necesarias. Este protocolo se llama **LCP (Protocolo de Control de Enlace)**. Admite circuitos síncronos y asíncronos y codificaciones orientadas a bits y a caracteres.
3. Un mecanismo para negociar opciones de capa de red con independencia del protocolo de red usado. El método escogido consiste en tener un **NCP (Protocolo de Control de Red)** distinto para cada protocolo de capa de red soportado.

Para ver la manera en que encajan estas piezas, consideremos la situación típica de un usuario doméstico llamando al proveedor de servicios de Internet para convertir una PC doméstica en un *host* temporal de Internet. La PC llama inicialmente al enrutador del proveedor a través de un módem. Una vez que el módem del enrutador ha contestado el teléfono y ha establecido una conexión física, la PC manda al enrutador una serie de paquetes LCP en el campo de carga útil de una o más tramas PPP. Estos paquetes, y sus respuestas, seleccionan los parámetros PPP por usar.

Una vez que se han acordado estos parámetros, se envía una serie de paquetes NCP para configurar la capa de red. Por lo general, la PC requiere ejecutar una pila de protocolos TCP/IP, por lo que necesita una dirección IP. No hay suficientes direcciones IP para todos, por lo que normalmente cada proveedor de Internet tiene un bloque de ellas y asigna de manera dinámica una a cada PC que se acaba de conectar para que la use durante su sesión. Si un proveedor posee n direcciones IP, puede tener hasta n máquinas conectadas en forma simultánea, pero su base total de clientes puede ser muchas veces mayor. Se utiliza el NCP para IP para asignar la dirección IP.

En este momento, la PC es ya un *host* de Internet y puede enviar y recibir paquetes IP, igual que los *host* permanentes. Cuando el usuario ha terminado, se utiliza NCP para desmantelar la conexión de la capa de red y liberar la dirección IP. Luego se usa LCP para cancelar la conexión de la capa de enlace de datos. Por último, la computadora indica al módem que cuelgue el teléfono, liberando la conexión de la capa física.

El formato de trama de PPP se escogió de modo que fuera muy parecido al de HDLC, ya que no había razón para reinventar la rueda. La diferencia principal entre PPP y HDLC es que el primero está orientado a caracteres, no a bits. En particular, PPP usa el relleno de bytes en las líneas de acceso telefónico con módem, por lo que todas las tramas tienen un número entero de bytes. No es posible enviar una trama que conste de 30.25 bytes, como con HDLC. No sólo pueden mandarse tramas PPP a través de líneas de acceso telefónico, sino que también pueden enviarse a través

de SONET o de líneas HDLC auténticas orientadas a bits (por ejemplo, para conexiones enrutador a enrutador). El formato de trama PPP se muestra en la figura 3-27.

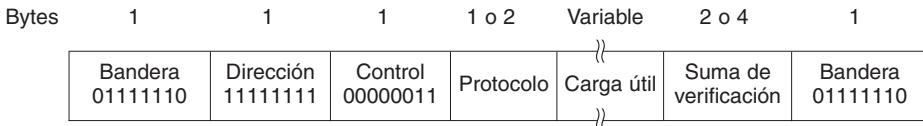


Figura 3-27. Formato de trama completa PPP para el modo de operación no numerado.

Todas las tramas PPP comienzan con la bandera estándar de HDLC (01111110), que se rellena con bytes si ocurre dentro del campo de carga útil. Luego está el campo de *Dirección*, que siempre se establece al valor binario 11111111 para indicar que todas las estaciones deben aceptar la trama. El empleo de este valor evita tener que asignar direcciones de la capa de enlace de datos.

El campo de *Dirección* va seguido del campo de *Control*, cuyo valor predeterminado es 00000011. Este valor indica una trama no numerada. En otras palabras, PPP no proporciona de manera predeterminada transmisión confiable usando números de secuencia y confirmaciones de recepción. En entornos ruidosos, como los de las redes inalámbricas, se puede emplear el modo numerado para transmisión confiable, aunque casi no se utiliza. Los detalles exactos se definen en el RFC 1663.

Dado que los campos de *Dirección* y de *Control* son constantes en la configuración predeterminada, LCP proporciona los mecanismos necesarios para que las dos partes negocien una opción que simplemente los omita por completo y ahorre dos bytes por trama.

El cuarto campo PPP es el de *Protocolo*. Su tarea es indicar la clase de paquete que está en el campo de *Carga útil*. Se definen códigos para LCP, NCP, IP, IPX, AppleTalk, entre otros. Los protocolos que comienzan con un bit 0 son de capa de red como IP, IPX, OSI CLNP, XNS. Los que comienzan con un bit 1 se utilizan para negociar otros protocolos. Entre éstos están LCP y un NCP diferente para cada protocolo de capa de red soportado. El tamaño predeterminado del campo de *protocolo* es de 2 bytes, pero puede negociarse a 1 byte usando LCP.

El campo de *Carga útil* es de longitud variable, hasta algún máximo negociado. Si la longitud no se negocia con LCP durante el establecimiento de la línea, se usa una longitud predeterminada de 1500 bytes. De ser necesario se puede incluir un relleno después de la carga.

Después del campo de *Carga útil* está el campo de *Suma de verificación*, que normalmente es de 2 bytes, pero puede negociarse una suma de verificación de 4 bytes.

En resumen, PPP es un mecanismo de entramado multiprotocolo adecuado para utilizarse a través de módems, líneas seriales de bits HDLC, SONET y otras capas físicas. Soporta detección de errores, negociación de opciones, compresión de encabezados y, opcionalmente, transmisión confiable con formato de tramas similar al de HDLC.

Ahora pasemos del formato de tramas PPP a la manera en que se activan y desactivan las líneas. En el diagrama (simplificado) de la figura 3-28 se muestran las fases por las que pasa una

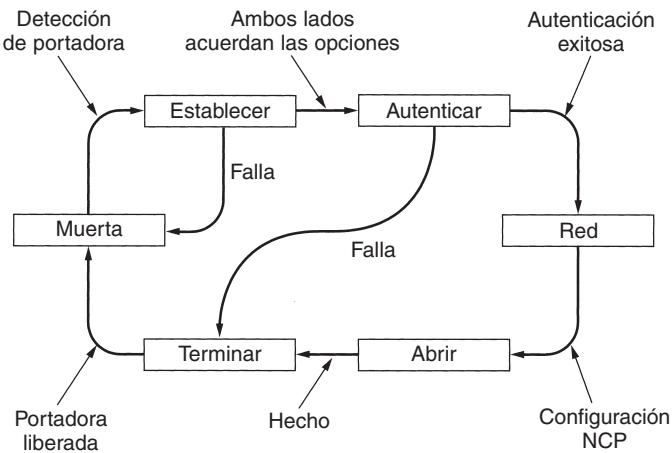


Figura 3-28. Diagrama de fases simplificado para activar y desactivar una línea.

línea cuando es activada, usada y desactivada. Esta secuencia se aplica tanto a las conexiones por módem como a las de enrutador a enrutador.

El protocolo inicia con la línea que tiene el estado *MUERTA*, lo que significa que no hay una portadora de capa física y que no existe una conexión de capa física. Una vez establecida la conexión física, la línea pasa a *ESTABLECER*. En ese punto comienza la negociación de opciones LCP, que, de tener éxito, conduce a *AUTENTICAR*. Ahora las dos partes pueden verificar la identidad del otro, si lo desean. Al entrar en la fase *RED*, se invoca el protocolo NCP apropiado para configurar la capa de red. Si la configuración tiene éxito, se llega a *ABRIR* y puede comenzar el transporte de datos. Al terminar el transporte, la línea pasa a la fase *TERMINAR*, de donde regresa a *MUERTA* al liberarse la portadora.

Se utiliza LCP para negociar las opciones del protocolo de enlace de datos durante la fase *ESTABLECER*. El protocolo LCP no se ocupa realmente de las opciones mismas, sino de los mecanismos de negociación; proporciona una vía para que el proceso iniciador haga una propuesta y para que el proceso que responde la acepte o la rechace, toda o en parte. También proporciona una forma para que los dos procesos prueben la calidad de la línea, y vean si es lo suficientemente buena para establecer una conexión. Por último, el protocolo LCP también permite que las líneas se desactiven cuando ya no se necesitan.

Hay 11 tipos de tramas LCP definidas en el RFC 1661, y se listan en la figura 3-29. Los cuatro tipos de *configuración* permiten que el iniciador (I) proponga valores de opción y que el contestador (R) las acepte o las rechace. En el último caso, el contestador puede hacer una propuesta alterna o anunciar que no está dispuesto a negociar en absoluto. Las opciones negociadas y sus valores propuestos son parte de las tramas LCP.

Los códigos de *terminación* sirven para desactivar una línea cuando ya no se necesita. El contestador utiliza los códigos de *código-rechazo* y *protocolo-rechazo* para indicar que recibió algo que no entiende. Esta situación puede significar que ha ocurrido un error de transmisión no detectado,

Nombre	Dirección	Descripción
Configurar-solicitud	I → R	Lista de opciones y valores propuestos
Configurar-confirmación de recepción	I ← R	Se aceptan todas las opciones
Configurar-confirmación de recepción negativa	I ← R	No se aceptan algunas opciones
Configurar-rechazo	I ← R	Algunas opciones no son negociables
Terminar-solicitud	I → R	Solicitud de desactivación de la línea
Terminar-confirmación de recepción	I ← R	Línea desactivada
Código-rechazo	I ← R	Se recibió una solicitud desconocida
Protocolo-rechazo	I ← R	Se solicitó un protocolo desconocido
Eco-solicitud	I → R	Favor de enviar de regreso esta trama
Eco-respuesta	I ← R	Aquí está la trama de regreso
Descartar-solicitud	I → R	Descartar esta trama (para pruebas)

Figura 3-29. Tipos de tramas LCP.

pero más probablemente significa que el iniciador y el contestador están ejecutando versiones diferentes del protocolo LCP. Los códigos *eco* sirven para probar la calidad de la línea. Por último, *descartar-solicitud* se utiliza para depuración. Si cualquiera de los lados está teniendo problemas para poner bits en la línea (transmitir), el programador puede emplear este tipo para pruebas. Si logra pasar, el receptor simplemente lo descarta en lugar de realizar alguna acción que podría confundir a la persona que efectúa las pruebas.

Las opciones que pueden negociarse incluyen el establecimiento del tamaño máximo de la carga útil para las tramas de datos, la habilitación de la autenticación y la selección del protocolo a emplear, habilitando el monitoreo de la calidad de la línea durante la operación normal y la selección de varias opciones de compresión de encabezados.

Hay poco que decir sobre los protocolos NCP en un sentido general. Cada uno es específico para algún protocolo de capa de red y permite que se hagan solicitudes de configuración específicas de ese protocolo. Por ejemplo, para IP la posibilidad más importante es la asignación dinámica de direcciones.

3.7 RESUMEN

La tarea de la capa de enlace de datos es convertir el flujo de bits en bruto ofrecido por la capa física en un flujo de tramas para que la capa de red lo utilice. Se emplean varios métodos de entramado, incluidos el conteo de caracteres, el relleno de bytes y el relleno de bits. Los protocolos de enlace de datos pueden proporcionar control de errores para retransmitir tramas dañadas o perdidas. Para evitar que un emisor rápido sature a un receptor lento, el protocolo de enlace de datos también puede proporcionar control de flujo. El mecanismo de ventana corrediza se emplea ampliamente para integrar el control de errores y el control de flujo de una manera conveniente.

Los protocolos de ventana corrediza pueden clasificarse por el tamaño de la ventana del emisor y por el tamaño de la ventana del receptor. Cuando ambos son iguales a 1, el protocolo es de parada y espera. Cuando el tamaño de la ventana del emisor es mayor que 1, por ejemplo, para evitar el bloqueo del emisor en un circuito con un retardo de propagación grande, el receptor puede programarse para descartar todas las tramas diferentes a la siguiente de la secuencia o almacenar en el búfer tramas fuera de orden hasta que se necesiten.

En este capítulo examinamos una serie de protocolos. El protocolo 1 se designa para un entorno libre de errores, en el que el receptor puede manejar cualquier flujo que se le haya enviado. El protocolo 2 también da por hecho un entorno libre de errores pero introduce control de flujo. El protocolo 3 maneja los errores con números de secuencia y con el algoritmo de parada y espera. El protocolo 4 permite la comunicación bidireccional e introduce el concepto de superposición. El protocolo 5 utiliza un protocolo de ventana corrediza con retroceso n. Por último, el protocolo 6 utiliza repetición selectiva y confirmaciones de recepción negativas.

Los protocolos pueden modelarse usando varias técnicas para demostrar que son correctos (o no). Los modelos de máquina de estados finitos y de red de Petri se usan frecuentemente para este propósito.

Muchas redes utilizan uno de los protocolos orientados a bits (SDLC, HDLC, ADCCP o LAPB) en la capa de enlace de datos. Todos estos protocolos usan bytes de bandera para delimitar tramas y relleno de bits para evitar que los bytes de bandera ocurran en los datos. Todos ellos también usan ventanas corredizas para el control de flujo. Internet utiliza PPP como el protocolo de enlace de datos primario en líneas punto a punto.

PROBLEMAS

1. Un mensaje de capa superior se divide en 10 tramas, cada una de las cuales tiene 80% de probabilidad de llegar sin daño. Si el protocolo de enlace de datos no lleva a cabo control de errores, ¿cuántas veces debe reenviarse el mensaje en promedio para conseguir que pase todo?
2. La siguiente codificación de caracteres se utiliza en un protocolo de enlace de datos:
A: 01000111; B: 11100011; FLAG: 01111110; ESC: 11100000
Muestre la secuencia de bits transmitida (en binario) para la trama de cuatro caracteres: A B ESC FLAG cuando se utiliza cada uno de los siguientes métodos de entrampado:
 - (a) Conteo de caracteres.
 - (b) Bytes de bandera con relleno de bytes.
 - (c) Bytes de bandera de inicio y final, con relleno de bits.
3. El siguiente fragmento de datos ocurre a la mitad de un flujo de datos para el cual se ha usado el algoritmo de relleno de bytes descrito en el texto: A B ESC C ESC FLAG FLAG D. ¿Cuál es la salida tras el relleno?
4. Uno de sus compañeros ha señalado que es un desperdicio terminar cada trama con un byte de bandera e iniciar la siguiente con otro. Un solo byte de bandera podría hacer el trabajo, por lo que un byte guardado es un byte ganado. ¿Usted está de acuerdo?

5. Una cadena de bits, 011110111110111110, necesita transmitirse en la capa de enlace de datos. ¿Cuál es la cadena que realmente se está transmitiendo después del relleno de bits?
6. Cuando se usa relleno de bits, ¿es posible que la pérdida, inserción o modificación de un solo bit cause un error que la suma de verificación no detecte? Si no, ¿por qué no? Si es así, explique cómo. ¿Desempeña aquí un papel la longitud de la suma de verificación?
7. ¿Puede pensar en alguna circunstancia en la cual podría ser preferible un protocolo de ciclo abierto (por ejemplo, un código de Hamming) a los protocolos tipo realimentación analizados a lo largo de este capítulo?
8. Para proporcionar mayor confiabilidad de la que puede dar un solo bit de paridad, un esquema de codificación de detección de errores usa un bit de paridad para todos los bits de número par. ¿Cuál es la distancia de Hamming de este código?
9. Se utiliza el código de Hamming para transmitir mensajes de 16 bits. ¿Cuántos bits de verificación se necesitan para asegurar que el receptor pueda detectar y corregir errores de un solo bit? Muestre el patrón de bits transmitido para el mensaje 1101001100110101. Suponga que se utiliza paridad par en el código de Hamming.
10. Un byte de 8 bits con un valor binario de 10101111 se va a codificar utilizando código de Hamming de paridad par. ¿Cuál es el valor binario que resulta de la codificación?
11. Un código de Hamming de 12 bits, cuyo valor hexadecimal es 0xE4F, llega al receptor. ¿Cuál era el valor hexadecimal original? Suponga que no más de un bit es erróneo.
12. Una manera de detectar errores es transmitir los datos como un bloque de n filas de k bits por fila y agregar bits de paridad a cada fila y a cada columna. La esquina inferior derecha es un bit de paridad que verifica su fila y su columna. ¿Detectará este esquema todos los errores sencillos? ¿Los errores dobles? ¿Los errores triples?
13. Un bloque de bits con n filas y k columnas usa bits de paridad horizontales y verticales para la detección de errores. Suponga que se invierten exactamente 4 bits debido a errores de transmisión. Deduza una expresión para la probabilidad de que el error no sea detectado.
14. ¿Qué residuo se obtiene al dividir $x^7 + x^5 + 1$ entre el polinomio generador $x^3 + 1$?
15. Un flujo de bits 10011101 se transmite utilizando el método estándar CRC que se describió en el texto. El generador polinomial es $x^3 + 1$. Muestre la cadena de bits real que se transmite. Suponga que el tercer bit, de izquierda a derecha, se invierte durante la transmisión. Muestre que este error se detecta en el lado receptor.
16. Los protocolos de enlace de datos casi siempre ponen el CRC en un terminador, en lugar de un encabezado. ¿Por qué?
17. Un canal tiene una tasa de bits de 4 kbps y un retardo de propagación de 20 mseg. ¿Para qué intervalo de tamaños de trama, la parada y espera da una eficiencia de cuando menos 50%?
18. Una troncal T1 de 3000 km de longitud se usa para transmitir tramas de 64 bytes con el protocolo 5. Si la velocidad de propagación es de 6 μ seg/km, ¿de cuántos bits deben ser los números de secuencia?
19. En el protocolo 3, ¿es posible que el emisor inicie el temporizador cuando éste ya está en ejecución? De ser así, ¿cómo podría ocurrir? De lo contrario, ¿por qué no es posible?

20. Imagine un protocolo de ventana corrediza que utiliza tantos bits para los números de secuencia que nunca ocurre un reinicio. ¿Qué relaciones deben mantenerse entre los cuatro límites de la ventana y el tamaño de la ventana, que es constante y el mismo tanto para el emisor como para el receptor?
21. Si el procedimiento *between* del protocolo 5 revisara la condición $a \leq b \leq c$ en lugar de la condición $a \leq b < c$, ¿tendría esto algún efecto en la corrección o en la eficiencia del protocolo? Explique su respuesta.
22. En el protocolo 6, cuando llega una trama de datos, se hace una revisión para ver si el número de secuencia es diferente del esperado y si *no_nak* es verdadero. Si ambas condiciones se cumplen, se envía una NAK. De otra manera, se arranca el temporizador auxiliar. Suponga que se omite la cláusula *else*. ¿Afectará este cambio la corrección del protocolo?
23. Suponga que el ciclo *while* de tres instrucciones cerca del final del protocolo 6 se elimina del código. ¿Afectará esto la corrección del protocolo o sólo su desempeño? Explique su respuesta.
24. Suponga que el caso para errores de suma de verificación fuera eliminado de la instrucción *switch* del protocolo 6. ¿Cómo afectará este cambio la operación del protocolo?
25. En el protocolo 6, el código de *frame_arrival* tiene una sección que se usa para los NAKs. Dicha sección se invoca si la trama entrante es una NAK y se cumple otra condición. Describa un escenario en el que la presencia de esta otra condición sea esencial.
26. Imagine que está escribiendo el software de la capa de enlace de datos para una línea que se usa para enviar datos a usted, pero no desde usted. El otro extremo usa HDLC, con un número de secuencia de tres bits y un tamaño de ventana de siete tramas. Usted podría querer almacenar en el búfer tantas tramas fuera de secuencia como fuera posible para aumentar la eficiencia, pero no se le permite modificar el software del lado emisor. ¿Es posible tener una ventana de receptor mayor que uno y aun así garantizar que el protocolo nunca fallará? De ser así, ¿cuál es el tamaño de la ventana más grande que se puede usar con seguridad?
27. Considere la operación del protocolo 6 en una línea de 1 Mbps libre de errores. El tamaño máximo de trama es de 1000 bits. Se generan nuevos paquetes a intervalos aproximados de 1 segundo. El tiempo de expiración del temporizador es de 10 mseg. Si se eliminara el temporizador especial de confirmación de recepción ocurrirían terminaciones de temporizador innecesarias. ¿Cuántas veces se transmitiría el mensaje promedio?
28. En el protocolo 6, $\text{MAX_SEQ} = 2^n - 1$. Si bien esta condición es evidentemente deseable para utilizar de manera eficiente los bits de encabezado, no hemos demostrado que sea esencial. ¿Funciona correctamente el protocolo con $\text{MAX_SEQ} = 4$, por ejemplo?
29. Se están enviando tramas de 1000 bits a través de un canal de 1 Mbps utilizando un satélite geoestacionario cuyo tiempo de propagación desde la Tierra es de 270 mseg. Las confirmaciones de recepción siempre se superponen en las tramas de datos. Los encabezados son muy cortos. Se usan números de secuencia de tres bits. ¿Cuál es la utilización máxima de canal que se puede lograr para:
- Parada y espera?
 - El protocolo 5?
 - El protocolo 6?
30. Calcule la fracción del ancho de banda que se desperdicia en sobrecarga (encabezados y retransmisiones) para el protocolo 6 en un canal satelital de 50 kbps con carga pesada, usando tramas de datos consistentes en 40 bits de encabezado y 3960 bits de datos. Asuma que el tiempo de propagación de la señal

de la Tierra al satélite es de 270 mseg. Nunca ocurren tramas ACK. Las tramas NAK son de 40 bits. La tasa de errores de las tramas de datos es de 1%, y la tasa de errores para las tramas NAK es insignificante. Los números de secuencia son de 8 bits.

31. Considere un canal satelital de 64 kbps libre de errores que se usa para enviar tramas de datos de 512 bytes en una dirección y devolver confirmaciones de recepción muy cortas en la otra. ¿Cuál es la velocidad real de transporte máxima con tamaños de ventana de 1, 7, 15 y 127? El tiempo de propagación de la Tierra al satélite es de 270 mseg.
32. Un cable de 100 km de longitud opera con una tasa de datos T1. La velocidad de propagación del cable es 2/3 de la velocidad de la luz en el vacío. ¿Cuántos bits caben en el cable?
33. Suponga que modelamos el protocolo 4 mediante el modelo de máquina de estados finitos. ¿Cuántos estados existen para cada máquina? ¿Cuántos estados existen para el canal de comunicaciones? ¿Cuántos estados existen para todo el sistema (dos máquinas y el canal)? Ignore los errores de suma de verificación.
34. Dé la secuencia de activación para la red de Petri de la figura 3-23 correspondiente a la secuencia de estado (000), (01A), (01—), (010), (01A) de la figura 3-21. Explique con palabras lo que representa la secuencia.
35. Dadas las reglas de transición $AC \rightarrow B$, $B \rightarrow AC$, $CD \rightarrow E$ y $E \rightarrow CD$, dibuje la red de Petri descrita. A partir de esta red, dibuje el grafo de estado finito alcanzable desde el estado inicial ACD . ¿Qué concepto bien conocido modelan estas reglas de transición?
36. PPP se basa estrechamente en HDLC, que utiliza relleno de bits para prevenir que los bytes de banda accidental dentro de la carga útil causen confusión. Dé por lo menos una razón por la cual PPP utiliza relleno de bytes.
37. ¿Cuál es la sobrecarga mínima para enviar un paquete IP mediante PPP? Tome en cuenta sólo la sobrecarga introducida por el PPP mismo, no la del encabezado IP.
38. El objetivo de este ejercicio es implementar un mecanismo de detección de errores utilizando el algoritmo CRC estándar descrito en el texto. Escriba dos programas: el generador y el verificador. El programa generador lee de una entrada estándar un mensaje de n bits como una cadena de 0s y 1s perteneciente a una línea de texto ASCII. La segunda línea es el código polinomial de k bits, también en ASCII. Ésta envía a la salida estándar una línea de texto ASCII con $n + k$ 0s y 1s que representan el mensaje que se va a transmitir. Después envía el código polinomial, justo como lo lee. El programa verificador lee la salida del programa generador y envía a la salida un mensaje que indica si es correcto o no. Por último, escriba un programa, de alteración, que invierta un bit en la primera línea dependiendo de su argumento (el número de bits considerando al bit más a la izquierda como 1), pero que copie el resto de las dos líneas de manera correcta. Si escribe:

```
generator <file | verifier
debe ver que el mensaje es correcto, pero si escribe
generator <file | alter arg | verifier
deberá obtener el mensaje de error.
```
39. Escriba un programa que simule el comportamiento de una red de Petri. El programa deberá leer las reglas de transición, así como una lista de estados que correspondan a la capa de enlace de red que emite o acepta un nuevo paquete. A partir del estado inicial, también de leído, el programa deberá elegir transiciones habilitadas al azar y activarlas, y verificar si el *host* acepta alguna vez 2 paquetes sin que el otro *host* emita uno nuevo en el interin.

4

LA SUBCAPA DE CONTROL DE ACCESO AL MEDIO

Como mencionamos en el capítulo 1, las redes pueden dividirse en dos categorías: las que utilizan conexiones punto a punto y las que utilizan canales de difusión. Este capítulo trata las redes de difusión y sus protocolos.

En cualquier red de difusión, el asunto clave es la manera de determinar quién puede utilizar el canal cuando hay competencia por él. Para aclarar este punto, considere una llamada en conferencia en la que seis personas, en seis teléfonos diferentes, están conectadas de modo que cada una puede oír y hablar con todas las demás. Es muy probable que cuando una de ellas deje de hablar, dos o más comiencen a hacerlo a la misma vez, lo que conducirá al caos. En las reuniones cara a cara, el caos se evita por medios externos. Por ejemplo, en una reunión, la gente levanta la mano para solicitar permiso de hablar. Cuando únicamente hay un canal, determinar quién debería tener el turno es muy complicado. Se conocen muchos protocolos para resolver el problema y constituyen el propósito de este capítulo. En la literatura, los canales de difusión a veces se denominan **canales multiacceso** o **canales de acceso aleatorio**.

Los protocolos usados para determinar quién sigue en un canal multiacceso pertenecen a una subcapa de la capa de enlace de datos llamada subcapa **MAC (Control de Acceso al Medio)**. La subcapa MAC tiene especial importancia en las LANs, casi todas las cuales usan un canal multiacceso como base de su comunicación. Las WANs, en contraste, usan enlaces punto a punto, excepto en las redes satelitales. Debido a que los canales multiacceso y las LANs están estrechamente relacionados, en este capítulo analizaremos las LANs en general, además de algunos aspectos que no son estrictamente parte de la subcapa MAC.

Desde el punto de vista técnico, la subcapa MAC es la parte inferior de la capa de enlace de datos, por lo que lógicamente debimos haberla estudiado antes de examinar los protocolos punto a punto en el capítulo 3. No obstante, para la mayor parte de la gente, la comprensión de los protocolos en los que intervienen muchas partes es más fácil una vez que se han entendido bien los protocolos de dos partes. Por esta razón nos hemos desviado de un orden de presentación estrictamente ascendente.

4.1 EL PROBLEMA DE ASIGNACIÓN DEL CANAL

El tema central de este capítulo es la forma de asignar un solo canal de difusión entre usuarios competidores. Primero veremos los esquemas estático y dinámico en general. Luego examinaremos varios algoritmos específicos.

4.1.1 Asignación estática de canal en LANs y MANs

La manera tradicional de asignar un solo canal, como una troncal telefónica, entre varios usuarios competidores es la FDM (Multiplexión por División de Frecuencia). Si hay N usuarios, el ancho de banda se divide en N partes de igual tamaño (vea la figura 2-31), y a cada usuario se le asigna una parte. Dado que cada usuario tiene una banda de frecuencia privada, no hay interferencia entre los usuarios. Cuando sólo hay una pequeña cantidad fija de usuarios, cada uno de los cuales tiene (en búfer) una carga de tráfico pesada (por ejemplo, las oficinas de conmutación de una empresa portadora), la FDM es un mecanismo de asignación sencillo y eficiente.

Sin embargo, cuando el número de emisores es grande y varía continuamente, o cuando el tráfico se hace en ráfagas, la FDM presenta algunos problemas. Si el espectro se divide en N regiones, y hay menos de N usuarios interesados en comunicarse actualmente, se desperdiciará una buena parte de espectro valioso. Si más de N usuarios quieren comunicarse, a algunos de ellos se les negará el permiso por falta de ancho de banda, aun cuando algunos de los usuarios que tengan asignada una banda de frecuencia apenas transmitan o reciban algo.

Sin embargo, aun suponiendo que el número de usuariosaría, de alguna manera, mantenerse constante en N , dividir el canal disponible en subcanales estáticos es inherentemente ineficiente. El problema básico es que, cuando algunos usuarios están inactivos, su ancho de banda simplemente se pierde. No lo están usando, y a nadie más se le permite usarlo. Es más, en casi todos los sistemas de cómputo el tráfico de datos se hace en ráfagas (son comunes las relaciones de tráfico pico a tráfico medio de 1000:1). En consecuencia, la mayoría de los canales estarán inactivos casi todo el tiempo.

El desempeño pobre de la FDM estática puede verse fácilmente mediante un cálculo sencillo de la teoría de colas. Comencemos por el retardo medio, T , de un canal de C bps de capacidad, con una tasa de llegada de λ tramas/seg, en el que cada trama tiene una longitud que se obtiene de una función exponencial de densidad de probabilidad con una media de $1/\mu$ bits/trama. Con estos parámetros, la tasa de llegada es de λ tramas/seg y la tasa de servicio es de

μC tramas/seg. A partir de la teoría de colas se puede mostrar que para tiempos de llegada y de servicio de Poisson,

$$T = \frac{1}{\mu C - \lambda}$$

Por ejemplo, si C es 100 Mbps, la longitud media de trama, $1/\mu$, es 10,000 bits y la tasa de llegada de tramas, λ , es 5000 tramas/seg, entonces $T = 200 \mu\text{seg}$. Observe que si ignoráramos el retardo de colas y sólo preguntáramos cuánto toma enviar una trama de 10,000 bits en una red de 100 Mbps, podríamos obtener la respuesta (incorrecta) de 100 μseg . El resultado sólo es válido cuando no hay competencia por el canal.

Ahora dividamos el canal en N subcanales independientes, cada uno con capacidad de C/N bps. La tasa media de entrada de cada uno de los subcanales ahora será de λ/N . Recalculando T , obtenemos:

$$T_{\text{FDM}} = \frac{1}{\mu(C/N) - (\lambda/N)} = \frac{N}{\mu C - \lambda} = NT \quad (4-1)$$

El retardo medio al usar FDM es N veces peor que si todas las tramas se acomodaran mágicamente de algún modo en una gran cola central.

Precisamente los mismos argumentos que se aplican a la FDM se aplican a la TDM (Multiplexión por División de Tiempo). A cada usuario se le asigna cada N -ésima ranura de tiempo. Si un usuario no usa la ranura asignada, simplemente se desperdicia. Lo mismo se aplica si dividimos las redes físicamente. Utilizando otra vez el ejemplo anterior, si reemplazamos la red de 100 Mbps con 10 redes de 10 Mbps cada una y asignamos de manera estática un usuario a cada una de ellas, el retardo medio reduciría de 200 μseg a 2 mseg.

Ya que ninguno de los métodos tradicionales de asignación estática de canal funciona muy bien con tráfico en ráfagas, ahora exploraremos los métodos dinámicos.

4.1.2 Asignación dinámica de canales en LANs y MANs

Antes de entrar en el primero de muchos métodos de asignación de canal que se analizarán en este capítulo, vale la pena formular con cuidado el problema de la asignación. Todo el trabajo hecho en esta área se basa en cinco supuestos clave, que se describen a continuación.

1. **Modelo de estación.** El modelo consiste en N estaciones independientes (computadoras, teléfonos, comunicadores personales, etcétera), cada una con un programa o usuario que genera tramas para transmisión. Algunas veces, las estaciones se conocen como **terminales**. La probabilidad de que una trama se genere en un intervalo de longitud Δt es de $\lambda\Delta t$, donde λ es una constante (la tasa de llegada de tramas nuevas). Una vez que se ha generado una trama, la estación se bloquea y no hace nada sino hasta que la trama se ha transmitido con éxito.

2. **Supuesto de canal único.** Hay un solo canal disponible para todas las comunicaciones. Todas las estaciones pueden transmitir en él y pueden recibir de él. En lo referente al hardware, todas las estaciones son equivalentes, aunque el software del protocolo puede asignarles prioridades.
3. **Supuesto de colisión.** Si dos tramas se transmiten en forma simultánea, se traslapan en el tiempo y la señal resultante se altera. Este evento se llama **colisión**. Todas las estaciones pueden detectar colisiones. Una trama en colisión debe transmitirse nuevamente después. No hay otros errores excepto aquellos generados por las colisiones.
- 4a. **Tiempo continuo.** La transmisión de una trama puede comenzar en cualquier momento. No hay reloj maestro que divida el tiempo en intervalos discretos.
- 4b. **Tiempo ranurado.** El tiempo se divide en intervalos discretos (ranuras). La transmisión de las tramas siempre comienza al inicio de una ranura. Una ranura puede contener 0, 1 o más tramas, correspondientes a una ranura inactiva, una transmisión con éxito o una colisión, respectivamente.
- 5a. **Detección de portadora.** Las estaciones pueden saber si el canal está en uso antes de intentar usarlo. Si se detecta que el canal está en uso, ninguna estación intentará utilizarlo sino hasta que regrese a la inactividad.
- 5b. **Sin detección de portadora.** Las estaciones no pueden detectar el canal antes de intentar usarlo. Simplemente transmiten. Sólo después pueden determinar si la transmisión tuvo éxito.

Es importante un análisis de estos supuestos. El primero dice que las estaciones son independientes, y que se genera trabajo a velocidad constante. También supone de manera implícita que cada estación sólo tiene un programa o usuario, así que mientras la estación esté bloqueada no se generará trabajo nuevo. Los modelos más complicados permiten estaciones multiprogramadas que pueden generar trabajo mientras la estación está bloqueada, pero el análisis de estas estaciones es mucho más complejo.

El supuesto del canal único es la esencia del modelo. No hay formas externas de comunicación. Las estaciones no pueden levantar la mano para solicitar que el maestro les ceda la palabra.

El supuesto de colisión también es básico, aunque en algunos sistemas (principalmente los de espectro disperso) este supuesto se suaviza con resultados sorprendentes. Además algunas LANs, como las *token ring*, pasan un *token* especial de estación en estación, y quien lo posea es quien puede transmitir una trama. Sin embargo, en las siguientes secciones nos apegaremos al modelo de canal único con contención y colisiones.

Hay dos supuestos alternativos sobre el tiempo. O es continuo (4a) o ranurado (4b). Algunos sistemas usan uno y otros el otro, por lo que estudiaremos y analizaremos ambos. Obviamente, para un sistema dado, sólo un supuesto es válido.

De manera semejante, una red puede tener detección de portadora (5a) o no (5b). Por lo general, las LANs tienen detección de portadora. Sin embargo, las redes inalámbricas no la pueden

utilizar de manera efectiva porque tal vez no todas las estaciones estén dentro del rango de radio de las demás. Las estaciones en redes alámbricas con detección de portadora pueden terminar su transmisión prematuramente si descubren que está chocando con otra transmisión. Por razones de ingeniería, la detección de colisión se hace muy rara vez en redes inalámbricas. Note que la palabra “portadora” en este sentido se refiere a una señal eléctrica en el cable y no tiene nada que ver con las empresas portadoras (por ejemplo, las compañías telefónicas), que datan de los tiempos del Pony Express.

4.2 PROTOCOLOS DE ACCESO MÚLTIPLE

Se conocen muchos algoritmos para asignar un canal de acceso múltiple. En las siguientes secciones estudiaremos una muestra representativa de los más interesantes y daremos algunos ejemplos de su uso.

4.2.1 ALOHA

En la década de 1970, Norman Abramson y sus colegas de la Universidad de Hawaii inventaron un método novedoso y elegante para resolver el problema de asignación de canal. Desde entonces, su trabajo ha sido extendido por muchos investigadores (Abramson, 1985). Aunque el trabajo de Abramson, llamado sistema ALOHA, usó la radiodifusión basada en tierra, la idea básica es aplicable a cualquier sistema en el que usuarios no coordinados compiten por el uso de un solo canal compartido.

Analizaremos dos versiones de ALOHA: puro y ranurado. Difieren en si se divide o no el tiempo en ranuras discretas en las que deben caber todas las tramas. El ALOHA puro no requiere sincronización global del tiempo; el ALOHA ranurado sí.

ALOHA puro

La idea básica de un sistema ALOHA es sencilla: permitir que los usuarios transmitan cuando tengan datos por enviar. Por supuesto, habrá colisiones y las tramas en colisión se dañarán. Sin embargo, debido a la propiedad de retroalimentación de la difusión, un emisor siempre puede saber si la trama fue destruida o no escuchando el canal, de la misma manera que los demás usuarios. Con una LAN, la retroalimentación es inmediata; vía satélite, hay un retardo de 270 mseg antes de que el emisor sepa si la transmisión tuvo éxito. Si por alguna razón no es posible escuchar mientras se transmite, se necesitan confirmaciones de recepción. Si la trama fue destruida, el emisor simplemente espera un tiempo aleatorio y la envía de nuevo. El tiempo de espera debe ser aleatorio o las mismas tramas chocarán una y otra vez, en sincronía. Los sistemas en los cuales varios usuarios comparten un canal común de modo tal que puede dar pie a conflictos se conocen como sistemas de **contención**.

En la figura 4-1 se presenta un esbozo de la generación de tramas en un sistema ALOHA. Hemos hecho que todas las tramas sean de la misma longitud porque la velocidad real de transporte (*throughput*) de los sistemas ALOHA se maximiza al tener tramas con un tamaño uniforme en lugar de tramas de longitud variable.

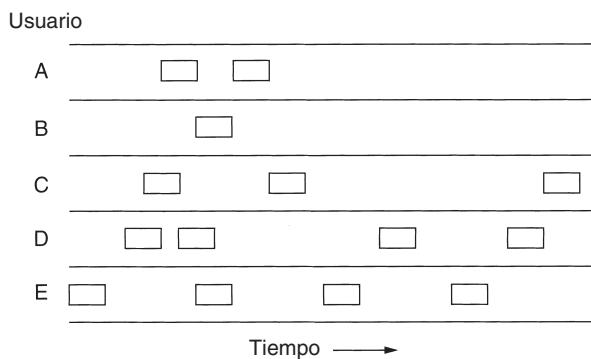


Figura 4-1. En ALOHA puro, las tramas se transmiten en momentos completamente arbitrarios.

Cada vez que dos tramas traten de ocupar el canal al mismo tiempo, habrá una colisión y ambas se dañarán. Si el primer bit de una trama nueva se traslapa con el último bit de una trama casi terminada, ambas tramas se destruirán por completo, y ambas tendrán que volver a transmitirse. La suma de verificación no puede (y no debe) distinguir entre una pérdida total y un error ligero. Lo malo es malo.

Una pregunta por demás interesante es: ¿cuál es la eficiencia de un canal ALOHA? Es decir, ¿qué fracción de todas las tramas transmitidas escapa a las colisiones en estas caóticas circunstancias? Primero consideraremos un conjunto infinito de usuarios interactivos sentados ante sus computadoras (estaciones). Un usuario siempre está en uno de dos estados: escribiendo o esperando. Inicialmente, todos los usuarios están en el estado de escritura. Al terminar una línea, el usuario deja de escribir, en espera de una respuesta. La estación después transmite una trama que contiene la línea y verifica el canal para saber si llegó con éxito. De ser así, el usuario ve la respuesta y continúa escribiendo. Si no, el usuario continúa esperando y la trama se transmite una y otra vez hasta que se envía con éxito.

Denotemos con “tiempo de trama” el tiempo necesario para transmitir una trama estándar de longitud fija (es decir, la longitud de la trama dividida entre la tasa de bits). En este punto, suponemos que la población infinita de usuarios genera tramas nuevas según una distribución de Poisson con una media de N tramas por tiempo de trama. (La suposición de población infinita es necesaria para asegurar que N no disminuya a medida que se bloquean los usuarios.) Si $N > 1$, la comunidad de usuarios está generando tramas a una velocidad mayor que la que puede manejar el canal, y casi cada trama sufre una colisión. Para una velocidad real de transporte razonable esperaríamos que $0 < N < 1$.

Además de tramas nuevas, las estaciones también generan retransmisiones de tramas que previamente sufrieron colisiones. Asimismo, supongamos que la probabilidad de k intentos de

transmisión por tiempo de trama, viejas y nuevas combinadas, también es una distribución de Poisson, con una media de G por tiempo de trama. Claramente, $G \geq N$. Con carga baja (es decir, $N \approx 0$) habrá pocas colisiones y, por lo tanto, pocas retransmisiones, por lo que $G \approx N$. Con carga alta habrá muchas colisiones, por lo que $G > N$. Con todas las cargas, la velocidad real de transporte, S , es sólo la carga ofrecida, G , por la probabilidad, P_0 , de que una transmisión tenga éxito (es decir, $S = GP_0$, donde P_0 es la probabilidad de que una trama no sufra una colisión).

Una trama no sufrirá una colisión si no se envían otras tramas durante un tiempo de trama desde su envío, como se muestra en la figura 4-2. ¿En qué condiciones llegará sin daño la trama sombreada? Sea t el tiempo requerido para enviar una trama. Si cualquier otro usuario generó una trama entre el tiempo t_0 y $t_0 + t$, el final de esa trama chocará con el comienzo de la trama sombreada. De hecho, el destino de la trama sombreada se selló aun antes de enviar el primer bit pero, dado que en ALOHA puro una estación no escucha el canal antes de transmitir, no tiene manera de saber que otra trama ya está en camino. De manera parecida, cualquier otra trama que salga entre $t_0 + t$ y $t_0 + 2t$ chocará con el final de la trama sombreada.

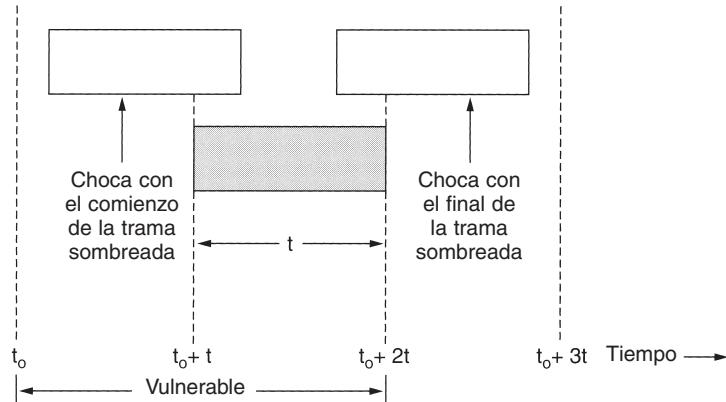


Figura 4-2. Periodo vulnerable para la trama sombreada.

La probabilidad de que k tramas sean generadas durante un tiempo de trama determinado está dada por la distribución de Poisson:

$$\Pr[k] = \frac{G^k e^{-G}}{k!} \quad (4-2)$$

así que la probabilidad de cero tramas es simplemente e^{-G} . En un intervalo de dos tiempos de trama de longitud, el número medio de tramas generadas es de $2G$. La probabilidad de que no se inicie otro tráfico durante todo el periodo vulnerable está dada entonces por $P_0 = e^{-2G}$. Si $S = GP_0$, obtenemos:

$$S = Ge^{-2G}$$

En la figura 4-3 se muestra la relación entre el tráfico ofrecido y la velocidad real de transporte. La velocidad real de transporte máxima ocurre a $G = 0.5$, con $S = 1/2e$, que es aproximadamente

te 0.184. En otras palabras, lo más que podemos esperar es un uso del canal de 18%. Este resultado no es muy alentador, pero con todo mundo transmitiendo al azar difícilmente podríamos esperar una tasa de éxito de 100%.

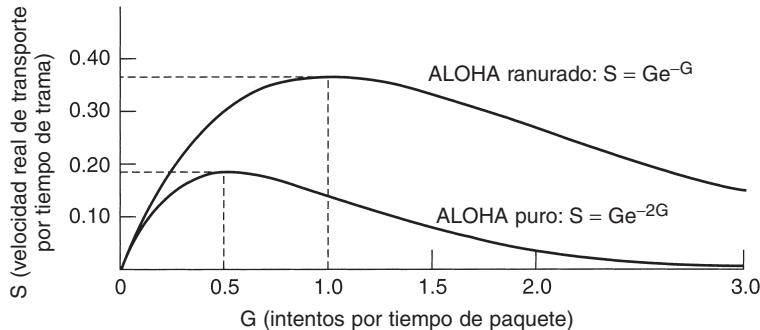


Figura 4-3. Velocidad real de transporte contra tráfico ofrecido en los sistemas ALOHA.

ALOHA ranurado

En 1972, Roberts publicó un método para duplicar la capacidad de un sistema ALOHA (Roberts, 1972). Su propuesta fue dividir el tiempo en intervalos discretos, cada uno de los cuales correspondía a una trama. Este enfoque requiere que los usuarios acuerden límites de ranura. Una manera de lograr la sincronización sería tener una estación especial que emitiera una señal al comienzo de cada intervalo, como un reloj.

En el método de Roberts, que se conoce como **ALOHA ranurado**, en contraste con el **ALOHA puro** de Abramson, no se permite que una computadora envíe cada vez que se pulsa un retorno de carro. En cambio, se le obliga a esperar el comienzo de la siguiente ranura. Por lo tanto, el ALOHA puro continuo se convierte en uno discreto. Dado que el periodo vulnerable ahora es de la mitad, la probabilidad de que no haya más tráfico durante la misma ranura que nuestra trama de prueba es de e^{-G} , lo que conduce a

$$S = Ge^{-G} \quad (4-3)$$

Como se puede ver en la figura 4-3, el ALOHA ranurado alcanza su máximo valor en $G = 1$, con una velocidad real de transporte de $S = 1/e$, o aproximadamente 0.368, el doble que el ALOHA puro. Si el sistema está operando a $G = 1$, la probabilidad de una ranura vacía es de 0.368 (de la ecuación 4-2). Lo mejor que podemos esperar usando ALOHA ranurado es 37% de ranuras vacías, 37% de éxitos y 26% de colisiones. La operación con valores mayores de G reduce el número de ranuras vacías pero aumenta de manera exponencial el número de colisiones. Para ver la manera en que se desarrolla este rápido crecimiento de colisiones con G , considere la transmisión de una trama de prueba. La probabilidad de que se evitará una colisión es de e^{-G} , que es la probabilidad de que los demás usuarios estén en silencio durante esa ranura. La probabilidad de una colisión es

entonces de sólo $1 - e^{-G}$. La probabilidad de que una transmisión requiera exactamente k intentos (es decir, $k - 1$ colisiones seguidas de un éxito) es

$$P_k = e^{-G}(1 - e^{-G})^{k-1}$$

El número esperado de transmisiones, E , por retorno de carro introducido es

$$E = \sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} ke^{-G}(1 - e^{-G})^{k-1} = e^G$$

Como resultado de la dependencia exponencial de E respecto a G , pequeños aumentos en la carga del canal pueden reducir drásticamente su desempeño.

El Aloha ranurado es importante por una razón que al principio tal vez no sea obvia. Se diseñó en 1970 y se utilizó en algunos sistemas experimentales iniciales, después casi se olvidó por completo. Cuando se inventó el acceso a Internet a través de cable, de repente surgió el problema de cómo asignar un canal compartido entre varios usuarios competidores, por lo que el ALOHA ranurado se sacó del cesto de la basura para resolver el problema. Con frecuencia sucede que los protocolos que son perfectamente válidos caen en desuso por razones políticas (por ejemplo, alguna compañía grande desea que todos hagan las cosas a su manera), pero años más tarde alguna persona astuta se da cuenta de que un protocolo descartado por mucho tiempo es el que puede sacarlo de su problema actual. Por esta razón, en este capítulo estudiaremos varios protocolos elegantes que en la actualidad no se utilizan mucho, pero que podrían utilizarse fácilmente en aplicaciones futuras, siempre y cuando suficientes diseñadores de red los conozcan. Por supuesto, también estudiaremos varios protocolos que se utilizan en la actualidad.

4.2.2 Protocolos de acceso múltiple con detección de portadora

Con el ALOHA ranurado, el mejor aprovechamiento de canal que puede lograrse es $1/e$. Esto no es sorprendente pues, con estaciones que transmiten a voluntad propia, sin prestar atención a lo que están haciendo las demás estaciones, es inevitable que haya muchas colisiones. Sin embargo, en las redes de área local es posible que las estaciones detecten lo que están haciendo las demás estaciones y adapten su comportamiento con base en ello. Estas redes pueden lograr una utilización mucho mejor que $1/e$. En esta sección analizaremos algunos protocolos para mejorar el desempeño.

Los protocolos en los que las estaciones escuchan una portadora (es decir, una transmisión) y actúan de acuerdo con ello se llaman **protocolos de detección de portadora**. Se ha propuesto una buena cantidad de ellos. Kleinrock y Tobagi (1975) han analizado en forma detallada algunos de esos protocolos. A continuación mencionaremos varias versiones de los protocolos de detección de portadora.

CSMA persistente y no persistente

El primer protocolo de detección de portadora que estudiaremos aquí se llama **CSMA (Acceso Múltiple con Detección de Portadora) persistente-1**. Cuando una estación tiene datos por

transmitir, primero escucha el canal para saber si otra está transmitiendo en ese momento. Si el canal está ocupado, la estación espera hasta que se desocupa. Cuando la estación detecta un canal inactivo, transmite una trama. Si ocurre una colisión, la estación espera una cantidad aleatoria de tiempo y comienza de nuevo. El protocolo se llama persistente-1 porque la estación transmite con una probabilidad de 1 cuando encuentra que el canal está inactivo.

El retardo de propagación tiene un efecto importante en el desempeño del protocolo. Hay una pequeña posibilidad de que, justo después de que una estación comienza a transmitir, otra estación está lista para enviar y detectar el canal. Si la señal de la primera estación no ha llegado aún a la segunda, esta última detectará un canal inactivo y comenzará a enviar también, lo que dará como resultado una colisión. Cuanto mayor sea el tiempo de propagación, más importante será este efecto, y peor el desempeño del protocolo.

Aun si el retardo de propagación es cero, habrá colisiones. Si dos estaciones quedan listas a la mitad de la transmisión de una tercera, ambas esperarán respetuosamente hasta el fin de la transmisión y entonces comenzarán a transmitir de manera simultánea, lo que dará como resultado una colisión. Si no fueran tan impacientes, habría menos colisiones. Aun así, este protocolo es mucho mejor que el ALOHA puro, ya que ambas estaciones tienen la decencia de dejar de interferir con la trama de la tercera estación. Intuitivamente, esto conducirá a un mejor desempeño que el del ALOHA puro. Con el ALOHA ranurado ocurre exactamente lo mismo.

Un segundo protocolo de detección de portadora es el **CSMA no persistente**. En éste se hace un intento consciente por ser menos egoísta que en el previo. Antes de enviar, una estación escucha el canal. Si nadie más está transmitiendo, la estación comienza a hacerlo. Sin embargo, si el canal ya está en uso, la estación no lo escucha de manera continua a fin de tomarlo de inmediato al detectar el final de la transmisión previa. En cambio, espera un periodo aleatorio y repite el algoritmo. En consecuencia, este algoritmo conduce a un mejor uso del canal pero produce mayores retardos que el CSMA persistente-1.

El último protocolo es el **CSMA persistente-p**, que se aplica a canales ranurados y funciona como se explica a continuación. Cuando una estación está lista para enviar, escucha el canal. Si éste se encuentra inactivo, la estación transmite con una probabilidad p . Con una probabilidad $q = 1 - p$, se espera hasta la siguiente ranura. Si esa ranura también está inactiva, la estación transmite o espera nuevamente, con probabilidades p y q . Este proceso se repite hasta que la trama ha sido transmitida o hasta que otra estación ha comenzado a transmitir. En el segundo caso, la estación actúa como si hubiera habido una colisión (es decir, espera un tiempo aleatorio y comienza de nuevo). Si al inicio la estación detecta que el canal está ocupado, espera hasta la siguiente ranura y aplica el algoritmo anterior. En la figura 4-4 se muestra la velocidad real de transporte calculada contra el tráfico ofrecido para los tres protocolos, así como para el ALOHA puro y el ranurado.

CSMA con detección de colisiones

Los protocolos CSMA persistentes y no persistentes ciertamente son una mejora respecto a ALOHA porque aseguran que ninguna estación comienza a transmitir cuando detecta que el canal está ocupado. Otra mejora es que las estaciones aborten sus transmisiones tan pronto como detecten

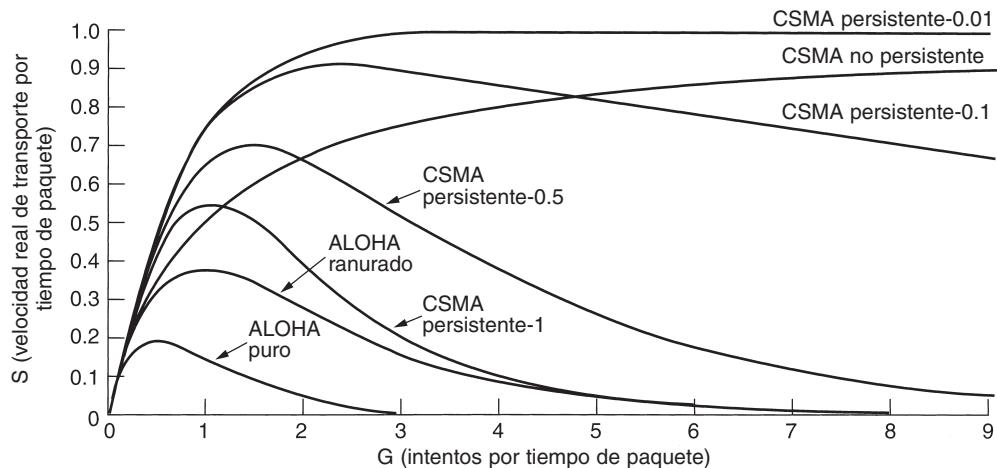


Figura 4-4. Comparación de la utilización del canal contra la carga para varios protocolos de acceso aleatorio.

una colisión. En otras palabras, si dos estaciones detectan que el canal está inactivo y comienzan a transmitir en forma simultánea, ambas detectarán la colisión casi de inmediato. En lugar de terminar de transmitir sus tramas, que de todos modos están irremediablemente alteradas, deben detener de manera abrupta la transmisión tan pronto como detectan la colisión. La terminación pronta de tramas dañadas ahorra tiempo y ancho de banda. Este protocolo, conocido como **CSMA/CD (Acceso Múltiple con Detección de Portadora y Detección de Colisiones)**, se usa ampliamente en las LANs en la subcapa MAC. En particular, es la base de la popular LAN Ethernet, por lo que vale la pena dedicar un poco de tiempo a analizarlo con detalle.

CSMA/CD, al igual que muchos otros protocolos de LAN, utiliza el modelo conceptual de la figura 4-5. En el punto marcado t_0 , una estación ha terminado de transmitir su trama. Cualquier otra estación que tenga una trama por enviar ahora puede intentar hacerlo. Si dos o más estaciones deciden transmitir en forma simultánea, habrá una colisión. Las colisiones pueden detectarse comparando la potencia o el ancho de pulso de la señal recibida con el de la señal transmitida.

Una vez que una estación detecta una colisión, aborta la transmisión, espera un tiempo aleatorio e intenta de nuevo, suponiendo que ninguna otra estación ha comenzado a transmitir durante ese lapso. Por lo tanto, nuestro modelo de CSMA/CD consistirá en períodos alternantes de contención y transmisión, ocurriendo períodos de inactividad cuando todas las estaciones están en reposo (por ejemplo, por falta de trabajo).

Ahora observemos con cuidado los detalles del algoritmo de contención. Suponga que dos estaciones comienzan a transmitir de manera exacta en el momento t_0 . ¿En cuánto tiempo se darán cuenta de que ha habido una colisión? La respuesta a esta pregunta es vital para determinar la longitud del periodo de contención y, por lo tanto, el retardo y la velocidad real de transporte. El tiempo mínimo para detectar la colisión es sólo el tiempo que tarda la señal para propagarse de una estación a otra.

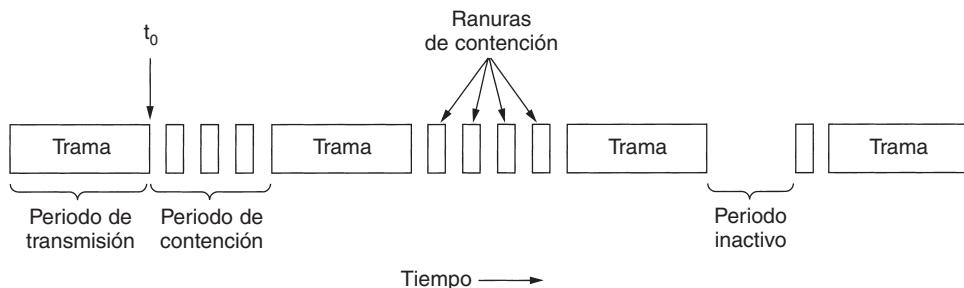


Figura 4-5. El CSMA/CD puede estar en uno de tres estados: contención, transmisión o inactivo.

Con base en este razonamiento, se podría pensar que una estación que después de iniciar su transmisión no detecta una colisión durante un tiempo igual al tiempo completo de propagación del cable podrá estar segura de que ha tomado el cable. Por “tomado” queremos decir que todas las demás estaciones sabían que estaba transmitiendo y no interfirieron. Esta conclusión es equivocada. Considere el siguiente escenario de peor caso. Sea τ el tiempo que tarda una señal en propagarse entre las dos estaciones más lejanas. En t_0 , una estación comienza a transmitir. En $\tau - \varepsilon$, un instante antes de que la señal llegue a la estación más lejana, esa estación también comienza a transmitir. Por supuesto, detecta la colisión casi de inmediato y se detiene, pero la pequeña ráfaga de ruido causada por la colisión no regresa a la estación original hasta $2\tau - \varepsilon$. En otras palabras, en el peor caso una estación no puede estar segura de que ha tomado el canal hasta que ha transmitido durante 2τ sin detectar una colisión. Por esta razón, modelaremos el intervalo de contención como un sistema ALOHA ranurado con un ancho de ranura de 2τ . En un cable coaxial de 1 km de longitud, $\tau \approx 5 \mu\text{seg}$. Por sencillez supondremos que cada ranura contiene sólo 1 bit. Por supuesto, una vez que ha tomado el canal, una estación puede transmitir a cualquier tasa que desee, no sólo a 1 bit cada 2τ seg.

Es importante darse cuenta de que la detección de colisiones es un proceso *analógico*. El hardware de la estación debe escuchar el cable mientras transmite. Si lo que lee es distinto de lo que puso en él, sabe que está ocurriendo una colisión. La implicación es que la codificación de la señal debe permitir que se detecten colisiones (por ejemplo, una colisión de dos señales de 0 voltios bien podría ser imposible de detectar). Por esta razón, por lo general se usa una codificación especial.

También vale la pena mencionar que una estación emisora debe monitorear de manera continua el canal en busca de ráfagas de ruido que puedan indicar una colisión. Por esta razón, CSMA/CD con un solo canal es inherentemente un sistema semidúplex. Es imposible que una estación transmita y reciba tramas al mismo tiempo, debido a que la lógica de recepción está en uso, en busca de colisiones durante cada transmisión.

Para evitar cualquier malentendido, es importante hacer notar que ningún protocolo de subcapa MAC garantiza la entrega confiable. Incluso en ausencia de colisiones, el receptor podría no haber copiado en forma correcta la trama por varias razones (por ejemplo, falta de espacio de búfer o una interrupción no detectada).

4.2.3 Protocolos libres de colisiones

Aunque las colisiones no ocurren en CSMA/CD una vez que una estación ha tomado sin ambigüedades el canal, aún pueden ocurrir durante el periodo de contención. Estas colisiones afectan en forma adversa el desempeño del sistema, especialmente cuando el cable es largo (es decir, τ es grande) y las tramas cortas. Además, CSMA/CD no es aplicable en todo el mundo. En esta sección examinaremos algunos protocolos que resuelven la contención por el canal sin que haya colisiones, ni siquiera durante el periodo de contención. En la actualidad, la mayoría de ellos no se utilizan en los sistemas grandes, pero en un campo en constante cambio, el hecho de tener algunos protocolos con excelentes propiedades disponibles para sistemas futuros con frecuencia es algo bueno.

En los protocolos que describiremos suponemos que hay N estaciones, cada una con una dirección única de 0 a $N - 1$ incorporada en hardware. El hecho de que algunas estaciones puedan estar inactivas parte del tiempo no importa. También damos por hecho que el retardo de propagación no importa. La pregunta básica persiste: ¿qué estación toma el canal tras una transmisión exitosa? Continuamos usando el modelo de la figura 4-5 con sus intervalos de contención discretos.

Un protocolo de mapa de bits

En nuestro primer protocolo libre de colisiones, el **método básico de mapa de bits**, cada periodo de contención consiste en exactamente N ranuras. Si la estación 0 tiene una trama por enviar, transmite un bit 1 durante la ranura 0. No está permitido a ninguna otra estación transmitir durante esta ranura. Sin importar lo que haga la estación 0, la estación 1 tiene la oportunidad de transmitir un 1 durante la ranura 1, pero sólo si tiene en cola una trama. En general, la estación j puede anunciar que tiene una trama por enviar introduciendo un bit 1 en la ranura j . Una vez que han pasado las N ranuras, cada estación sabe cuáles son todas las estaciones que quieren transmitir. En ese punto, las estaciones comienzan a transmitir en orden numérico (vea la figura 4-6).

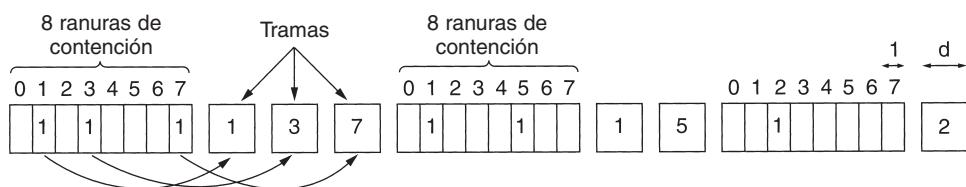


Figura 4-6. Protocolo básico de mapa de bits.

Dado que todos están de acuerdo en quién continúa, nunca habrá colisiones. Una vez que la última estación lista haya transmitido su trama, evento que pueden detectar fácilmente todas las estaciones, comienza otro periodo de contención de N bits. Si una estación queda lista justo después de que ha pasado su ranura de bits, ha tenido mala suerte y deberá permanecer inactiva hasta que cada estación haya tenido su oportunidad y el mapa de bits haya comenzado de nuevo. Los

protocolos como éste en los que el deseo de transmitir se difunde antes de la transmisión se llaman **protocolos de reservación**.

Analicemos con brevedad el desempeño de este protocolo. Por conveniencia, mediremos el tiempo en unidades de la ranura de bits de contención, con tramas de datos consistentes en d unidades de tiempo. En condiciones de carga baja, el mapa de bits simplemente se repetirá una y otra vez, debido a la falta de tramas de datos.

Considere la situación desde el punto de vista de una estación de número bajo, como 0 o 1. Por lo general, cuando la estación queda lista para transmitir, la ranura “actual” estará en algún lugar a la mitad del mapa de bits. En promedio, la estación tendrá que esperar $N/2$ ranuras para que el barrido actual termine, y otras N ranuras para que el siguiente barrido se ejecute hasta su terminación, antes de poder comenzar a transmitir.

Las perspectivas para las estaciones de número alto son mejores. En general, éstas sólo tendrán que esperar la mitad de un barrido ($N/2$ ranuras de bits) antes de comenzar a transmitir. Las estaciones de número alto pocas veces tienen que esperar el siguiente barrido. Dado que las estaciones de número bajo deben esperar un promedio de $1.5N$ ranuras y las estaciones de número alto deben esperar en promedio $0.5N$ ranuras, la media de todas las estaciones es de N ranuras. La eficiencia del canal cuando la carga es baja es fácil de calcular. La sobrecarga por trama es de N bits, y la cantidad de datos es de d bits, dando una eficiencia de $d/(N + d)$.

Si la carga es alta y todas las estaciones tienen algo que enviar todo el tiempo, el periodo de contención de N bits se prorrata entre N tramas, arrojando una sobrecarga de sólo 1 bit por trama, o una eficiencia de $d/(d + 1)$. El retardo medio de una trama es igual a la suma del tiempo que está en cola en su estación más un $N(d + 1)/2$ adicional una vez que llega a la cabeza de su cola interna.

Conteo descendente binario

Un problema con el protocolo básico de mapa de bits es que la sobrecarga es de 1 bit por estación, por lo que no se escala bien en redes con miles de estaciones. Podemos tener mejores resultados usando direcciones de estación binarias. Una estación que quiere utilizar el canal ahora difunde su dirección como una cadena binaria de bits, comenzando por el bit de orden mayor. Se supone que todas las direcciones tienen la misma longitud. A los bits en cada posición de dirección de las diferentes estaciones se les aplica un OR BOOLEANO a todos juntos. A este protocolo lo llamaremos **conteo descendente binario**. Se utilizó en Datakit (Fraser, 1987). Asume de manera implícita que los retardos de transmisión son insignificantes, de manera que todas las estaciones ven los bits instantáneamente.

Para evitar conflictos, debe aplicarse una regla de arbitraje: una vez que una estación ve que una posición de bit de orden alto, que en su dirección es 0, ha sido sobrescrita con un 1, se da por vencida. Por ejemplo, si las estaciones 0010, 0100, 1001 y 1010 están tratando de obtener el canal, en el primer tiempo de bit las estaciones transmiten 0, 0, 1 y 1, respectivamente. A éstos se les aplica el OR para formar un 1. Las estaciones 0010 y 0100 ven el 1 y saben que una estación de número mayor está compitiendo por el canal, por lo que se dan por vencidas durante esta ronda. Las estaciones 1001 y 1010 continúan.

El siguiente bit es 0, y ambas estaciones continúan. El siguiente bit es 1, por lo que la estación 1001 se da por vencida. La ganadora es la estación 1010, debido a que tiene la dirección mayor. Tras ganar la contienda, ahora puede transmitir una trama, después de lo cual comienza otro ciclo de contienda. El protocolo se ilustra en la figura 4-7. Tiene la propiedad de que estaciones con números grandes tienen una prioridad mayor que las estaciones con números pequeños, lo cual puede ser bueno o malo, dependiendo del contexto.

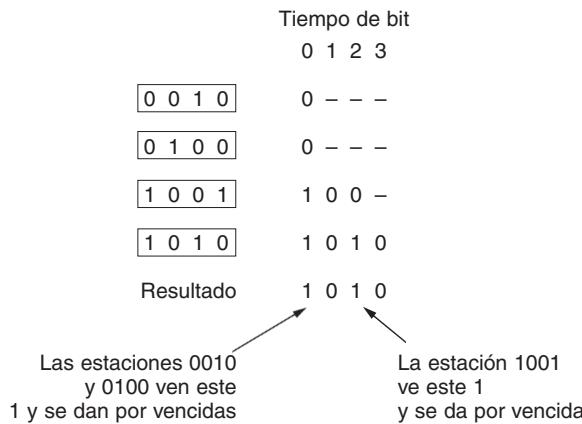


Figura 4-7. Protocolo de conteo descendente binario. Los guiones indican silencios.

La eficiencia de canal de este método es de $d/(d + \log_2 N)$. Sin embargo, si el formato de trama se escoge ingeniosamente de modo que la dirección del emisor sea el primer campo de la trama ni siquiera estos $\log_2 N$ bits se desperdician, y la eficiencia es de 100%.

Mok y Ward (1979) han descrito una variación del conteo descendente binario usando una interfaz paralela en lugar de serial. También sugieren el uso de números virtuales de estación, en el que los números virtuales de estación desde 0 hasta, e incluyendo, el de la estación ganadora se permutan en forma circular tras cada transmisión a fin de darle mayor prioridad a las estaciones que han estado en silencio demasiado tiempo. Por ejemplo, si las estaciones C, H, D, A, G, B, E, F tienen prioridades 7, 6, 5, 4, 3, 2, 1 y 0, respectivamente, entonces una transmisión exitosa de D la pone al final de la lista, dando un orden de prioridad de C, H, A, G, B, E, F, D . Por lo tanto, C permanece como la estación virtual 7, pero A sube de 4 a 5 y D desciende de 5 a 0. La estación D ahora sólo será capaz de adquirir el canal si ninguna otra estación lo quiere.

El conteo descendente binario es un ejemplo de un protocolo sencillo, elegante y eficiente que está esperando a ser redescubierto. Esperemos que algún día encuentre un nuevo hogar.

4.2.4 Protocolos de contención limitada

Hasta ahora hemos considerado dos estrategias básicas de adquisición del canal en una red de cable: los métodos por contención, como el CSMA, y los libres de colisión. Cada estrategia puede ser clasificada según lo bien que funciona en relación con las dos medidas de desempeño

importantes, el retardo con carga baja y la eficiencia del canal con carga alta. En condiciones de carga baja, la contención (es decir, ALOHA puro o ranurado) es preferible debido a su bajo retardo. A medida que aumenta la carga, la contención se vuelve paulatinamente menos atractiva, debido a que la sobrecarga asociada al arbitraje del canal se vuelve mayor. Lo inverso se cumple para los protocolos libres de colisiones. Con carga baja, tienen un retardo alto, pero a medida que aumenta la carga, mejora la eficiencia del canal en lugar de empeorar, como ocurre con los protocolos de contención.

Obviamente, sería agradable si pudiéramos combinar las mejores propiedades de los protocolos de contención y los libres de colisiones, desarrollando un protocolo nuevo que usara contención cuando la carga fuera baja para así tener un retardo bajo, y una técnica libre de colisiones cuando la carga fuera alta para lograr una buena eficiencia de canal. De hecho, existen tales protocolos, a los que llamaremos **protocolos de contención limitada**, y concluirán nuestro estudio de las redes de detección de portadora.

Hasta ahora, los únicos protocolos de contención que hemos estudiado han sido simétricos; es decir, cada estación intenta adquirir el canal con alguna probabilidad, p , y todas las estaciones usan la misma p . Resulta interesante que el desempeño general del sistema a veces puede mejorarse usando un protocolo que asigne diferentes probabilidades a diferentes estaciones.

Antes de ver los protocolos asimétricos, repasemos con rapidez el desempeño en el caso simétrico. Suponga que k estaciones están contendiendo por el acceso al canal. Cada una tiene una probabilidad p de transmitir durante cada ranura. La probabilidad de que una estación adquiera con éxito el canal durante una ranura dada es entonces de $kp(1 - p)^{k-1}$. Para encontrar el valor óptimo de p , diferenciamos con respecto de p , igualamos el resultado a cero y despejamos p . Al hacerlo, encontramos que el mejor valor de p es $1/k$. Sustituyendo $p = 1/k$, obtenemos:

$$\Pr[\text{éxito con } p \text{ óptima}] = \left[\frac{k-1}{k} \right]^{k-1} \quad (4-4)$$

En la figura 4-8 se presenta gráficamente esta probabilidad. Para un número pequeño de estaciones, la probabilidad de éxito es buena, pero tan pronto la cantidad de estaciones alcanza cinco, la probabilidad cae a su valor asintótico de $1/e$.

En la figura 4-8 es bastante evidente que la probabilidad de que una estación adquiera el canal sólo puede aumentar disminuyendo la cantidad de competencia. Los protocolos de contención limitada hacen precisamente eso. Primero dividen las estaciones en grupos (no necesariamente separados). Sólo los miembros del grupo 0 pueden competir por la ranura 0. Si uno de ellos tiene éxito, adquiere el canal y transmite su trama. Si la ranura permanece desocupada o si hay una colisión, los miembros del grupo 1 compiten por la ranura 1, etcétera. Al dividir en forma adecuada y en grupos las estaciones, se puede reducir el grado de contención para cada ranura y, en consecuencia, se puede operar cada ranura cerca de la parte izquierda de la figura 4-8.

El truco está en cómo asignar las estaciones a las ranuras. Antes de ver el caso general, consideremos algunos casos especiales. En un extremo, cada grupo tiene sólo un miembro. Una asignación de este tipo garantiza que nunca habrá colisiones, pues cuando mucho una estación compite

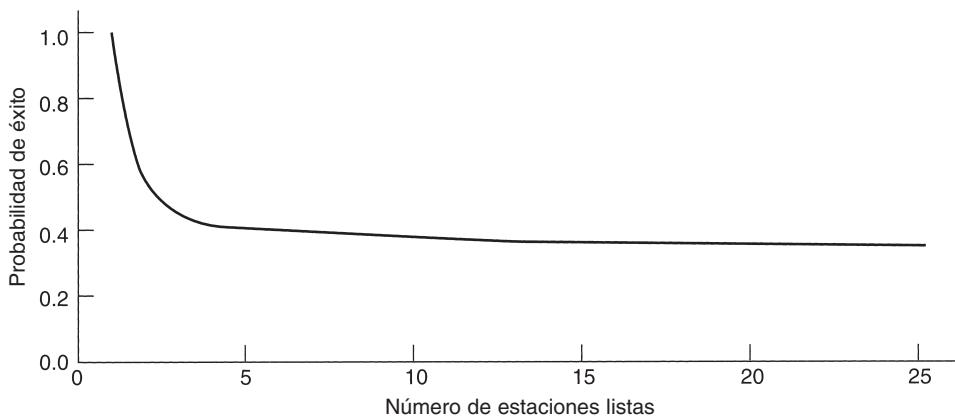


Figura 4-8. Probabilidad de adquisición de un canal de contención simétrica.

por una ranura dada. Antes vimos tales protocolos (por ejemplo, el conteo descendente binario). El siguiente caso especial es asignar dos estaciones por grupo. La probabilidad de que ambos intentarán transmitir durante una ranura es p^2 , que para una p pequeña es insignificante. A medida que se asignan más estaciones a la misma ranura, aumenta la probabilidad de colisión, pero disminuye la longitud del barrido del mapa de bits necesaria para dar a todos una oportunidad. El caso límite es un solo grupo que contiene todas las estaciones (es decir, ALOHA ranurado). Lo que necesitamos es una manera de asignar de manera dinámica las estaciones a las ranuras, con muchas estaciones por ranura cuando la carga es baja y pocas estaciones (o incluso sólo una) por ranura cuando la carga es alta.

Protocolo de recorrido de árbol adaptable

Una manera particularmente sencilla de llevar a cabo la asignación necesaria es usar el algoritmo desarrollado por el ejército de Estados Unidos para hacer pruebas de sífilis a los soldados durante la Segunda Guerra Mundial (Dorfman, 1943). En pocas palabras, el ejército tomaba una muestra de sangre de N soldados. Se vaciaba una parte de cada muestra en un solo tubo de ensayo. Luego, esta muestra mezclada era examinada en busca de anticuerpos. Si no se encontraban, todos los soldados del grupo eran declarados sanos. Si se encontraban anticuerpos, se preparaban dos muestras mezcladas nuevas, una de los soldados 1 a $N/2$ y otra de los demás. El proceso se repetía recursivamente hasta que se determinaban los soldados infectados.

Para la versión de computadora de este algoritmo (Capetanakis, 1979) es conveniente considerar a las estaciones como hojas de un árbol binario, de la manera que se muestra en la figura 4-9. En la primera ranura de contención después de la transmisión satisfactoria de una trama, ranura 0, se permite que todas las estaciones intenten adquirir el canal. Si una de ellas lo logra, qué bueno. Si hay una colisión, entonces, durante la ranura 1, sólo aquellas estaciones que quedan bajo el nodo 2 del árbol podrán competir. Si alguna de ellas adquiere el canal, la ranura que sigue a la trama se reserva para las estaciones que están bajo el nodo 3. Por otra parte, si dos o más

estaciones bajo el nodo 2 quieren transmitir, habrá una colisión durante la ranura 1, en cuyo caso es el turno del nodo 4 durante la ranura 2.

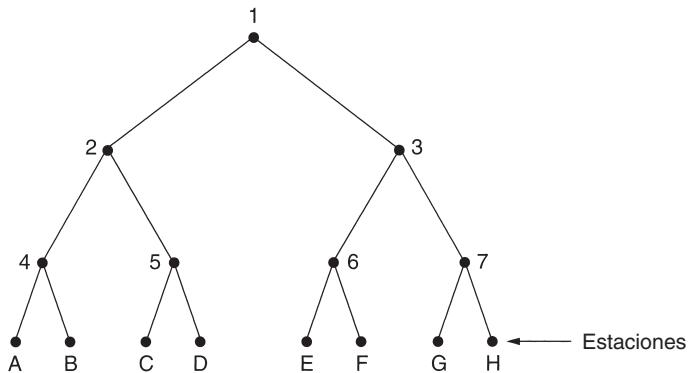


Figura 4-9. El árbol para ocho estaciones.

En esencia, si ocurre una colisión durante la ranura 0, se examina todo el árbol para localizar todas las estaciones listas. Cada ranura de bits está asociada a un nodo en particular del árbol. Si ocurre una colisión, continúa la búsqueda recursivamente con los hijos izquierdo y derecho del nodo. Si una ranura de bits está inactiva, o si sólo hay una estación que transmite en ella, puede detenerse la búsqueda de su nodo, pues se han localizado todas las estaciones listas. (Si hubiera existido más de una, habría ocurrido una colisión.)

Cuando la carga del sistema es pesada, apenas vale la pena dedicarle la ranura 0 al nodo 1, porque eso sólo tiene sentido en el caso poco probable de que precisamente una estación tenga una trama por enviar. De manera similar, se podría argumentar que los nodos 2 y 3 también podrían brincarse por la misma razón. En términos más generales, ¿en qué nivel del árbol debe comenzar la búsqueda? Es obvio que, a mayor carga, la búsqueda debe comenzar más abajo en el árbol. Supondremos que cada estación tiene una buena estimación del número de estaciones listas, q , por ejemplo, por la supervisión del tráfico reciente.

Para proceder, numeraremos los niveles del árbol desde arriba, con el nodo 1 de la figura 4-9 en el nivel 0, los nodos 2 y 3 en el nivel 1, etcétera. Observe que cada nodo del nivel i tiene una fracción 2^{-i} de las estaciones por debajo de él. Si las q estaciones listas se distribuyen de manera uniforme, el número esperado de ellas por debajo de un nodo específico en el nivel i es sólo $2^{-i}q$. Intuitivamente, esperaríamos que el nivel óptimo para comenzar a examinar al árbol fuera aquel cuyo número medio de estaciones contendientes por ranura sea 1, es decir, el nivel en el que $2^{-i}q = 1$. Resolviendo esta ecuación, encontramos que $i = \log_2 q$.

Se han descubierto numerosas mejoras al algoritmo básico, y han sido analizadas con cierto detalle por Bertsekas y Gallager (1992). Por ejemplo, considere el caso en el que las estaciones G y H son las únicas que quieren transmitir. En el nodo 1 ocurrirá una colisión, por lo que se intentará el 2, pero se encontrará inactivo. No tiene caso probar el nodo 3, ya que está garantizado que tendrá una colisión (sabemos que dos o más estaciones bajo 1 están listas y que ninguna de ellas

está bajo 2, por lo que todas deben estar bajo 3). La prueba de 3 puede omitirse para intentar 6. Al no arrojar nada esta prueba, también puede omitirse 7 para intentar el nodo *G* después.

4.2.5 Protocolos de acceso múltiple por división de longitud de onda

Un método diferente para la asignación del canal es dividir el canal en subcanales usando FDM, TDM o ambas, y asignarlos de manera dinámica según se necesite. Por lo general, los esquemas como éste se usan en las LANs de fibra óptica para permitir que diferentes conversaciones usen distintas longitudes de onda (es decir, frecuencias) al mismo tiempo. En esta sección analizaremos uno de tales protocolos (Humblet y cols., 1992).

Una manera sencilla de construir una LAN completamente óptica es utilizar un acoplador pasivo en estrella (vea la figura 2-10). En efecto, se fusionan dos fibras de cada estación a un cilindro de vidrio. Una fibra es para salidas al cilindro y la otra para entradas del cilindro. La salida de luz de cualquier estación ilumina el cilindro y puede ser detectada por todas las demás estaciones. Las estrellas pasivas pueden manejar cientos de estaciones.

Para permitir múltiples transmisiones al mismo tiempo, se divide el espectro en canales (bandas de longitud de onda), como se muestra en la figura 2-31. En este protocolo, **WDMA (Acceso Múltiple por División de Longitud de Onda)**, se asignan dos canales a cada estación. Se proporciona un canal estrecho como canal de control para señalizar la estación, y un canal ancho para que la estación pueda enviar tramas de datos.

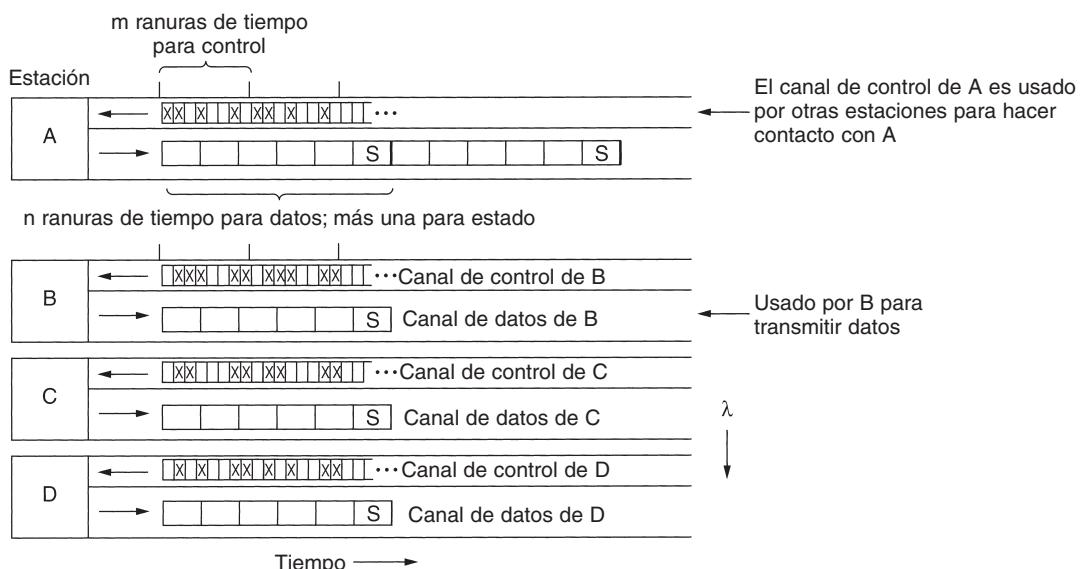


Figura 4-10. Acceso múltiple por división de longitud de onda.

Cada canal se divide en grupos de ranuras de tiempo, como se ilustra en la figura 4-10. Sea m el número de ranuras en el canal de control y $n + 1$ el número de ranuras en el canal de datos,

donde n de éstas son para datos y la última es usada por la estación para informar su estado (específicamente, qué ranuras de ambos canales están libres). En ambos canales, la secuencia de ranuras se repite de manera indefinida, marcándose la ranura 0 de una manera especial para que los que llegan tarde la puedan detectar. Todos los canales se sincronizan con un solo reloj global.

El protocolo reconoce tres clases de tráfico: (1) tráfico orientado a la conexión con tasa de datos constante, como vídeo sin comprimir, (2) tráfico orientado a la conexión con tasa de datos variable, como transferencia de archivos, y (3) tráfico de datagramas, como paquetes UDP. En los dos protocolos orientados a la conexión lo que se pretende es que para que A se comunique con B , primero debe introducir una trama de solicitud de conexión (CONNECTION REQUEST) en una ranura libre del canal de control de B . Si B lo acepta, la comunicación puede llevarse a cabo por el canal de datos de A .

Cada estación tiene dos emisores y dos receptores, como sigue:

1. Un receptor de longitud de onda fija para escuchar su propio canal de control.
2. Un emisor sintonizable para enviar por el canal de control de otra estación.
3. Un emisor de longitud de onda fija para la salida de tramas de datos.
4. Un receptor sintonizable para seleccionar al emisor de datos a escuchar.

En otras palabras, cada estación escucha en su propio canal de control las solicitudes que llegan, pero tiene que sintonizarse a la longitud de onda del emisor para obtener los datos. La sintonización de la longitud de onda se realiza con un interferómetro Fabry-Perot o Mach-Zehnder que filtra todas las longitudes de onda excepto la banda de longitud de onda deseada.

Ahora consideremos la manera en que la estación A establece un canal de comunicación clase 2 con la estación B para, digamos, transferencia de archivos. Primero, A sintoniza su receptor de datos con el canal de datos de B y espera la ranura de estado. Esta ranura indica cuáles ranuras de control están actualmente asignadas y cuáles están libres. Por ejemplo, en la figura 4-10 vemos que de las ocho ranuras de control de B , la 0, la 4 y la 5 están libres. Las demás están ocupadas (lo cual se indica por la "x").

A elige una de las ranuras de control libres, digamos la 4, e introduce ahí su mensaje de solicitud de conexión. Ya que B revisa de manera constante su canal de control, ve la solicitud y la acepta asignando la ranura 4 a A . Esta asignación se anuncia en la ranura de estado del canal de datos de B . Cuando A ve el anuncio, sabe que tiene una conexión unidireccional. Si A solicita una conexión bidireccional, B repite ahora el mismo algoritmo con A .

Es posible que, al mismo tiempo que A intente tomar la ranura de control 4 de B , C haga lo mismo. Ninguno lo conseguirá, y ambos se darán cuenta del fracaso revisando la ranura de estado del canal de control de B . Ahora ambos esperarán una cantidad de tiempo aleatoria y lo reintentarándespués.

En este punto, cada parte tiene un mecanismo libre de conflictos para enviar mensajes de control cortos entre ellas. Para llevar a cabo la transferencia de archivos, A ahora envía a B un mensaje de control que dice, por ejemplo, "observa por favor mi siguiente salida de datos en la ranura 3. Hay

una trama de datos ahí para ti". Cuando B recibe el mensaje de control, sintoniza su receptor al canal de salida de A para leer la trama de datos. Dependiendo del protocolo de la capa superior, B puede utilizar el mismo mecanismo para regresar una confirmación de recepción, si lo desea.

Observe que surge un problema si tanto A como C tienen conexiones con B y cada uno de ellos le indica a B que busque en la ranura 3. B escogerá una de estas solicitudes al azar y la otra transmisión se perderá.

Para tráfico de tasa constante se utiliza una variación de este protocolo. Cuando A solicita una conexión, simultáneamente dice algo como: ¿Está bien si te envío una trama cada vez que ocurra la ranura 3? Si B puede aceptar (es decir, no tiene compromisos previos para la ranura 3), se establece una conexión de ancho de banda garantizado. Si no, A puede intentarlo después con una propuesta distinta, dependiendo de las ranuras de salida que tenga libres.

El tráfico clase 3 (datagramas) usa otra variación. En lugar de escribir un mensaje de solicitud de conexión en la ranura de control que acaba de encontrar (4), escribe un mensaje DATA FOR YOU IN SLOT 3 (HAY DATOS PARA TI EN LA RANURA 3). Si B está libre durante la siguiente ranura de datos 3, la transmisión tendrá éxito. De otra manera, se perderá la trama de datos. En esta forma, nunca se requieren conexiones.

Son posibles diversas variantes del protocolo. Por ejemplo, en lugar de dar a cada estación su propio canal de control, se puede compartir un solo canal de control entre todas las estaciones. Se asigna a cada estación un bloque de ranuras de cada grupo, multiplexando efectivamente varios canales virtuales en uno solo físico.

También es posible arreglárselas con un solo emisor y un solo receptor sintonizables por estación haciendo que el canal de cada estación se divida en m ranuras de control seguidas de $n + 1$ ranuras de datos. La desventaja aquí es que los emisores tienen que esperar más tiempo para capturar una ranura de control, y las tramas de datos consecutivas están más distantes porque se interpone cierta información de control.

Se han propuesto e implementado muchos otros protocolos de control WDMA, los cuales difieren en varios detalles. Algunos tienen sólo un canal de control, otros tienen varios. Algunos toman en cuenta el retardo de propagación, otros no. Algunos hacen del tiempo de sintonización una parte explícita del modelo, otros lo ignoran. Los protocolos también difieren en términos de complejidad de procesamiento, velocidad real de transporte y escalabilidad. Cuando se utiliza una gran cantidad de frecuencias, algunas veces este sistema se llama **DWDM (Multiplexión Densa por División de Longitud de Onda)**. Para mayor información, vea (Bogineni y cols., 1993; Chen, 1994; Goralski, 2001; Kartalopoulos, 1999, y Levine y Akyildiz, 1995).

4.2.6 Protocolos de LANs inalámbricas

A medida que crece la cantidad de dispositivos de cómputo y comunicación portátiles, también crece la demanda de conexión de ellos con el mundo externo. Los primeros teléfonos portátiles ya tenían la capacidad de conectarse con otros teléfonos. Las primeras computadoras portátiles no tenían esta capacidad, pero poco después los módems se volvieron comunes en tales computadoras. Para entrar en línea, estas computadoras tenían que conectarse a un enchufe telefónico. El

requisito de una conexión con cable a la red fija significó que las computadoras eran portátiles, mas no móviles.

Para lograr una auténtica movilidad, las computadoras portátiles necesitan usar señales de radio (o infrarrojas) para comunicarse. De esta manera, los usuarios pueden leer y enviar correo electrónico mientras van de excursión o navegan. Un sistema de computadoras portátiles que se comunican por radio puede considerarse una LAN inalámbrica, como se trató en la sección 1.5.4. Estas LANs tienen propiedades un tanto diferentes que las LANs convencionales y requieren protocolos de subcapa MAC especiales. En esta sección estudiaremos algunos de estos protocolos. Puede encontrar más información sobre las LANs inalámbricas en (Geier, 2002, y O'Hara y Petrick, 1999).

Una configuración común para una LAN inalámbrica es un edificio de oficinas con estaciones base (también conocidas como puntos de acceso) ubicadas estratégicamente en distintas partes del edificio. Todas las estaciones base están interconectadas mediante cobre o fibra. Si la potencia de transmisión de las estaciones base y portátiles se ajusta a un alcance de 3 o 4 metros, entonces cada cuarto se vuelve una celda única, y el edificio entero se vuelve un sistema celular grande, como los sistemas de telefonía celular tradicionales que estudiamos en el capítulo 2. A diferencia de los sistemas telefónicos celulares, cada celda sólo tiene un canal, que cubre todo el ancho de banda disponible. Por lo general, el ancho de banda es de 11 a 54 Mbps.

En nuestros siguientes análisis haremos la suposición de simplificación de que todos los emisores de radio tienen algún alcance fijo. Cuando un receptor está dentro del alcance de dos emisores activos, la señal resultante por lo común se altera y resulta inútil, en otras palabras, ya no consideraremos los sistemas de tipo CDMA en este análisis. Es importante darse cuenta de que en algunas LANs inalámbricas no todas las estaciones están dentro del alcance de todas las demás, lo que conduce a una variedad de complicaciones. Es más, para las LANs inalámbricas interiores, la presencia de paredes entre las estaciones puede tener un impacto importante sobre el alcance efectivo de cada estación.

Un enfoque inocente para usar una LAN inalámbrica podría ser intentar el CSMA; escuchar si hay otras transmisiones y sólo transmitir si nadie más lo está haciendo. El problema radica en que este protocolo no es realmente adecuado porque lo que importa es la interferencia en el receptor, no en el emisor. Para ver la naturaleza de este problema, considere la figura 4-11, en la que se ilustran cuatro estaciones inalámbricas. Para nuestros fines, no importa cuáles son estaciones base ni cuáles son portátiles. El alcance de radio es tal que *A* y *B* están en el mismo alcance y potencialmente pueden interferir entre sí. *C* también podría interferir tanto con *B* como con *D*, pero no con *A*.



Figura 4-11. LAN inalámbrica. (a) *A* transmitiendo. (b) *B* transmitiendo.

Primero considere lo que ocurre cuando A está transmitiendo hacia B , como se muestra en la figura 4-11(a). Si C detecta el medio, no podrá escuchar a A porque está fuera de su alcance y, por tanto, deducirá falsamente que puede transmitir a B . Si C comienza a transmitir, interferirá en B , eliminando la trama de A . El problema de que una estación no puede detectar a un competidor potencial por el medio, puesto que dicho competidor está demasiado lejos, se denomina **problema de estación oculta**.

Ahora consideremos la situación inversa: B transmitiendo a A , como se muestra en la figura 4-11(b). Si C detecta el medio, escuchará una transmisión y concluirá equivocadamente que no puede enviar a D , cuando de hecho tal transmisión causaría una mala recepción sólo en la zona entre B y C , en la que no está localizado ninguno de los receptores pretendidos. Esta situación se denomina **problema de estación expuesta**.

El problema es que antes de comenzar una transmisión, una estación realmente necesita saber si hay actividad o no alrededor del receptor. El CSMA simplemente le indica si hay o no actividad alrededor de la estación que está detectando la portadora. Con un cable, todas las señales se propagan a todas las estaciones, de manera que sólo puede llevarse a cabo una transmisión en un momento dado en cualquier lugar del sistema. En un sistema basado en ondas de radio de corto alcance, pueden ocurrir transmisiones simultáneas si las ondas tienen destinos diferentes y éstos están fuera de alcance entre sí.

Otra forma de visualizar este problema es imaginar un edificio de oficinas en el que cada empleado tiene una computadora portátil inalámbrica. Suponga que Linda quiere enviar un mensaje a Melitón. La computadora de Linda detecta el entorno local y, al no percibir actividad, procede a transmitir. Sin embargo, aún puede hacer colisión en la oficina de Melitón, pues un tercero podría estar transmitiéndole actualmente desde una localidad tan alejada de la de Linda que la computadora de ésta no podría detectarlo.

MACA y MACAW

MACA (Acceso Múltiple con Prevención de Colisiones) (Karn, 1990) es uno de los primeros protocolos diseñados para LANs inalámbricas. El concepto en que se basa es que el emisor estimule al receptor a enviar una trama corta, de manera que las estaciones cercanas puedan detectar esta transmisión y eviten ellas mismas hacerlo durante la siguiente trama de datos (grande). El MACA se ilustra en la figura 4-12.

Consideremos ahora la manera en que A envía una trama a B . A comienza por enviar una trama **RTS (Solicitud de Envío)** a B , como se muestra en la figura 4-12(a). Esta trama corta (30 bytes) contiene la longitud de la trama de datos que seguirá posteriormente. Después B contesta con una trama **CTS (Libre para Envío)**, como se muestra en la figura 4-12(b). La trama CTS contiene la longitud de los datos (copiada de la trama RTS). Una vez que sucede la recepción de la trama CTS, A comienza a transmitir.

Ahora veremos cómo reaccionan las estaciones que escuchan cualquiera de estas tramas. Cualquier estación que escuche el RTS evidentemente está bastante cerca de A y debe permanecer en silencio durante el tiempo suficiente para que el CTS se transmita de regreso a A sin conflicto. Cualquier estación que escuche el CTS está bastante cerca de B y debe permanecer en silencio

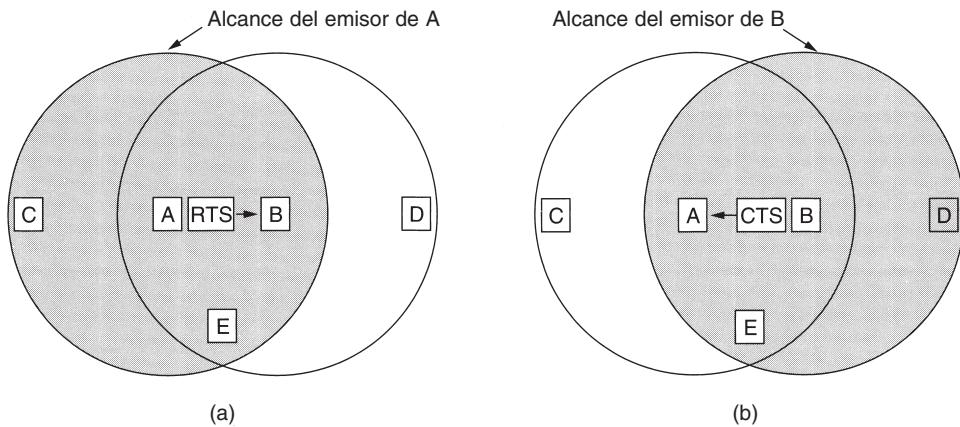


Figura 4-12. El protocolo MACA. (a) *A* enviando a *B* un RTS. (b) *B* respondiendo a *A* con un CTS.

durante la siguiente transmisión de datos, cuya longitud puede determinar examinando la trama CTS.

En la figura 4-12, *C* está en el alcance de *A*, pero no en el de *B*. Por lo tanto, escucha el RTS de *A* pero no el CTS de *B*. Mientras no interfiera con el CTS, está libre para transmitir mientras se está enviando la trama de datos. En contraste, *D* está en el alcance de *B* pero no de *A*. No escucha el RTS pero sí el CTS. Al escuchar el CTS se le indica que está cerca de una estación que está a punto de recibir una trama, por lo que difiere el envío de cualquier cosa hasta el momento en que se espera la terminación de esa trama. La estación *E* escucha ambos mensajes de control y, al igual que *D*, debe permanecer en silencio hasta que se haya completado la trama de datos.

A pesar de estas precauciones, aún pueden ocurrir colisiones. Por ejemplo, *B* y *C* pueden enviar tramas RTS a *A* al mismo tiempo. Éstas chocarán y se perderán. En el caso de una colisión, un emisor sin éxito (es decir, uno que no escucha un CTS en el intervalo de tiempo esperado) espera un tiempo aleatorio y reintenta. El algoritmo empleado es el retroceso exponencial binario, que estudiaremos cuando lleguemos a Ethernet.

Con base en estudios de simulación del MACA, Bharghavan y cols. (1994) afinaron el MACA para mejorar su desempeño y llamaron **MACAW** (**MACA Inalámbrico**) a su nuevo protocolo. Para comenzar, notaron que, sin confirmación de recepción de la capa de enlace de datos, las tramas no eran retransmitidas sino hasta que la capa de transporte notaba su ausencia, mucho después. Resolvieron este problema introduciendo una trama ACK tras cada trama de datos exitosa. También observaron que CSMA puede servir para evitar que una estación transmita un RTS al mismo tiempo y destino que otra estación cercana, por lo que se agregó la detección de portadora. Además, decidieron ejecutar el algoritmo de retroceso por separado para cada flujo de datos (par origen-destino), en lugar de para cada estación. Este cambio mejora la equidad del protocolo. Por último, agregaron un mecanismo para que las estaciones intercambiaran información sobre congestionamientos, y una manera de hacer que el algoritmo de retroceso reaccionara menos violentamente a problemas pasajeros, con lo que mejoraron el desempeño del sistema.

4.3 ETHERNET

Ya terminamos nuestra discusión general sobre los protocolos de asignación de canal, por lo que es tiempo de ver la forma en que estos principios se aplican a sistemas reales, particularmente a LANs. Como lo discutimos en la sección 1.5.3, el IEEE ha estandarizado varias redes de área local y de área metropolitana bajo el nombre de IEEE 802. Algunas han sobrevivido pero muchas no, como lo vimos en la figura 1-38. Quienes creen en la reencarnación piensan que Charles Darwin regresó como miembro de la Asociación de Estándares del IEEE para eliminar a los débiles. Los sobrevivientes más importantes son el 802.3 (Ethernet) y el 802.11 (LAN inalámbrica). Sería muy precipitado decir algo sobre el 802.15 (Bluetooth) y el 802.16 (MAN inalámbrica). Para mayor información, consulte la quinta edición de este libro. Tanto el 802.3 como el 802.11 tienen diferentes capas físicas y diferentes subcapas MAC, pero convergen en la misma subcapa de control lógico del enlace (que se define en el 802.2), por lo que tienen la misma interfaz a la capa de red.

En la sección 1.5.3 presentamos a Ethernet, por lo que no repetiremos aquí la información. En su lugar, nos enfocaremos en los detalles técnicos de Ethernet, en los protocolos y en los desarrollos recientes en la Ethernet de alta velocidad (gigabit). Puesto que Ethernet y el IEEE 802.3 son idénticos, excepto por dos diferencias mínimas que analizaremos pronto, muchas personas utilizan los términos “Ethernet” e “IEEE 802.3” de manera indistinta, y nosotros haremos lo mismo aquí.* Para información sobre Ethernet, vea (Breyer y Riley, 1999; Seifert, 1998, y Spurgeon, 2000).

4.3.1 Cableado Ethernet

Dado que el nombre “Ethernet” se refiere al cable (el éter), comencemos nuestro estudio por ahí. Comúnmente se usan cuatro tipos de cableado, como se muestra en la figura 4-13.

Nombre	Cable	Seg. máx.	Nodos/seg	Ventajas
10Base5	Coaxial grueso	500 m	100	Cable original; ahora obsoleto
10Base2	Coaxial delgado	185 m	30	No se necesita concentrador
10Base-T	Par trenzado	100 m	1024	Sistema más económico
10Base-F	Fibra óptica	2000 m	1024	Mejor entre edificios

Figura 4-13. Los tipos más comunes de cableado Ethernet.

Históricamente llegó primero el cable **10Base5**, llamado popularmente **Ethernet grueso**; se meja una manguera de jardín amarilla, con marcas cada 2.5 metros para indicar los puntos de las derivaciones. (El estándar 802.3 no *requiere* realmente que el cable sea amarillo, pero sí lo *sugiere*.) Por lo general, las conexiones a él se hacen usando **derivaciones vampiro**, en las que se introduce *cuidadosamente* una punta hasta la mitad del núcleo del cable coaxial. La notación

*Cabe mencionar que la comunicación entre estaciones que emplean Ethernet y estaciones que utilizan 802.3 no es posible aun cuando comparten el mismo canal (medio físico). (N. del R.T.)

10Base5 significa que opera a 10 Mbps, utiliza señalización de banda base y puede manejar segmentos de hasta 500 metros. El primer número es la velocidad en Mbps. Después viene la palabra “Base” (o algunas veces “BASE”) para indicar transmisión de banda base. Solía haber una variante para banda ancha, 10Broad36, pero nunca tuvo popularidad en el mercado, por lo que desapareció. Por último, si el medio es coaxial, su longitud se da redondeada a unidades de 100 m después de “Base”.

Históricamente, el segundo tipo de cable fue **10Base2** o **Ethernet delgado** que, a diferencia con el Ethernet grueso parecido a una manguera de jardín, se dobla con facilidad. Las conexiones se hacen usando conectores BNC estándar de la industria para formar uniones T, en lugar de emplear derivaciones vampiro. Los conectores son más fáciles de usar y más confiables. El Ethernet delgado es mucho más económico y fácil de instalar, pero sólo puede extenderse 185 metros por segmento, cada uno de los cuales puede manejar sólo 30 máquinas.

La detección de ruptura de cable, derivaciones malas o conectores flojos puede ser un problema importante en ambos medios. Por esta razón se han desarrollado técnicas para rastrear estos problemas. Básicamente, se inyecta un pulso de forma conocida en el cable. Si el pulso incide en un obstáculo o en el final del cable, se generará un eco que viajará de regreso. Si se cronometra cuidadosamente el intervalo entre el envío del pulso y la recepción del eco, es posible ubicar el origen del eco. Esta técnica se llama **reflectometría en el dominio del tiempo**.

Los problemas asociados con la localización de rupturas de cable han empujado a los sistemas a un tipo de patrón de cableado diferente, en el que todas las estaciones tienen cables que conducen a un **concentrador** (*hub*) central, en el que se conectan de manera eléctrica (como si se soldaran juntas). Por lo general, estos cables son pares trenzados telefónicos, ya que la mayoría de los edificios de oficinas ya están cableados de esta manera y normalmente hay bastantes pares extra disponibles. Este esquema se llama **10Base-T**. Los concentradores no almacenan en el búfer el tráfico de entrada. Más adelante en este capítulo analizaremos una versión mejorada de esta idea (comunicadores), la cual sí almacena en búfer el tráfico entrante.

Estos tres esquemas de cableado se ilustran en la figura 4-14. Para 10Base5, se sujetó firmemente un **transceptor** alrededor del cable, de modo que su derivación haga contacto con el núcleo interno. El transceptor contiene la electrónica que maneja detección de portadora y detección de colisiones. Al detectarse una colisión, el transceptor también pone una señal no válida especial en el cable para asegurar que todos los demás transceptores también se den cuenta de que ha ocurrido una colisión.

Con 10Base5, un **cable de transceptor** o **cable de derivación** conecta el transceptor a una tarjeta de interfaz en la computadora. El cable de transceptor puede tener hasta 50 metros de longitud y contiene cinco pares trenzados aislados individualmente. Dos de los pares son para entrada y salida de datos, respectivamente; dos más son para entrada y salida de señales de control. El quinto par, que no siempre se usa, permite que la computadora energice la electrónica del transceptor. Algunos transceptores permiten la conexión de hasta ocho computadoras cercanas a ellos, a fin de reducir la cantidad de transceptores necesarios.

El cable de transceptor termina en una tarjeta de interfaz en la computadora. La tarjeta de interfaz contiene un *chip* controlador que transmite tramas al transceptor y recibe tramas de él. El controlador se encarga de ensamblar los datos en el formato de trama adecuado, así como de

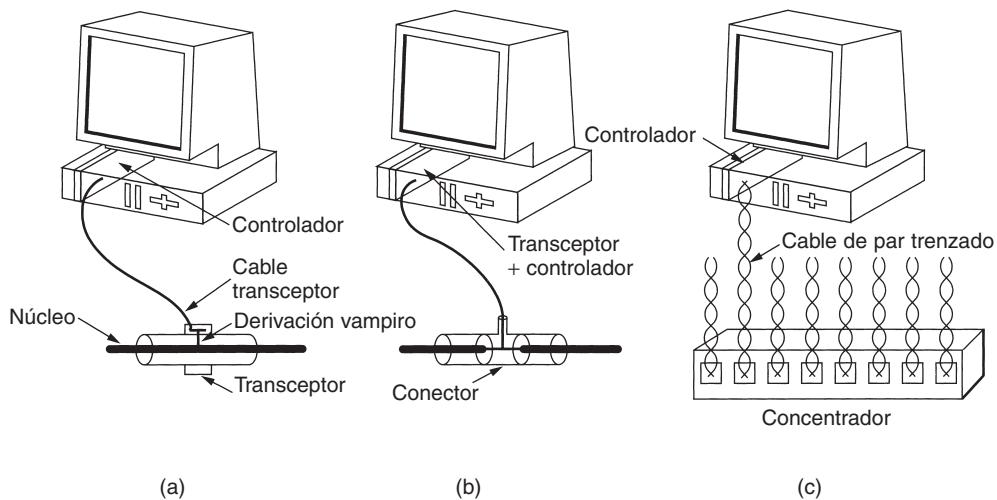


Figura 4-14. Tres tipos de cableado Ethernet. (a) 10Base5. (b) 10Base2. (c) 10Base-T.

calcular las sumas de verificación de las tramas de salida y de comprobarlas en las tramas de entrada. Algunos *chips* controladores también administran un grupo de búferes para las tramas de entrada, una cola para la transmisión de los búferes, transferencias de memoria directa con las computadoras *host* y otros aspectos de administración de la red.

Con 10Base2, la conexión al cable es sólo un conector BNC pasivo de unión T. La electrónica del transceptor está en la tarjeta controladora, y cada estación siempre tiene su propio transceptor.

Con 10Base-T no hay cable en absoluto, sólo el concentrador (una caja llena de circuitos electrónicos) al que cada estación está conectada mediante un cable dedicado (es decir, que no es compartido). Agregar o remover estaciones es más sencillo con esta configuración, y las rupturas de cable pueden detectarse con facilidad. La desventaja de 10Base-T es que la longitud máxima del cable es de sólo 100 metros, tal vez de 200 metros si se usa cable de par trenzado de alta calidad (categoría 5). Aun así, 10Base-T se está volviendo cada vez más común debido a que utiliza el cableado existente y a la facilidad de mantenimiento que ofrece. Se analizará una versión más rápida de 10Base-T (100Base-T) posteriormente en este capítulo.

Una cuarta opción de cableado para Ethernet es **10Base-F**, que usa fibra óptica. Esta alternativa es cara debido al costo de los conectores y los terminadores, pero tiene excelente inmunidad contra el ruido y es el método a usar para conexiones entre edificios o entre concentradores muy separados. Se permiten separaciones de kilómetros entre conexiones. También ofrece buena seguridad debido a que es más difícil intervenir una conexión de fibra que una de cobre.

En la figura 4-15 se muestran diferentes maneras de cablear un edificio. En la figura 4-15(a), un solo cable se pasa entre cuarto y cuarto, y cada estación se conecta a él en el punto más cercano. En la figura 4-15(b) una columna vertebral vertical corre del sótano a la azotea, y en cada piso se conectan cables horizontales a dicha columna mediante amplificadores especiales (repetidores). En algunos edificios los cables horizontales son delgados, y la red dorsal es gruesa. La topología

más general es en árbol, como en la figura 4-15(c), porque una red con dos rutas entre algunos pares de estaciones sufrirá interferencia entre las dos señales.

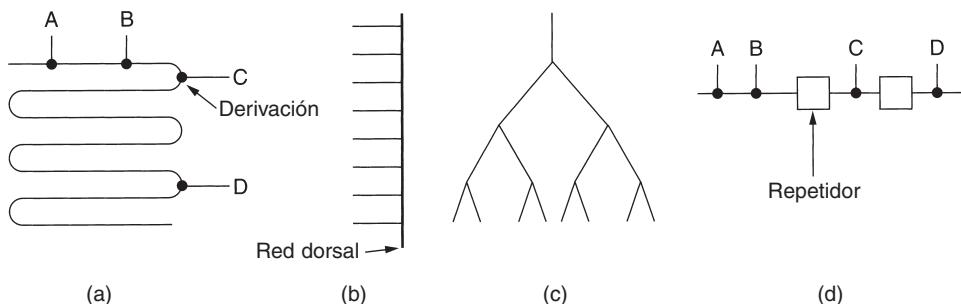


Figura 4-15. Topologías de cableado. (a) Lineal. (b) Columna vertebral. (c) Árbol. (d) Segmentada.

Cada versión de Ethernet tiene una longitud máxima de cable por segmento. Para permitir redes mayores, se pueden conectar múltiples cables mediante **repetidores**, como se muestra en la figura 4-15(d). Un repetidor es un dispositivo de capa física que recibe, amplifica (regenera) y retransmite señales en ambas direcciones. En lo que concierne al software, una serie de segmentos de cable conectados mediante repetidores no es diferente de un solo cable (excepto por el retraso introducido por los repetidores). Un sistema puede contener múltiples segmentos de cable y múltiples repetidores, pero ningún par de transceptores puede estar separado por más de 2.5 km y ninguna ruta entre dos transceptores puede atravesar más de cuatro repetidores.

4.3.2 Codificación Manchester

Ninguna de las versiones de Ethernet utiliza codificación binaria directa con 0 voltios para un bit 0 y 5 voltios para un bit 1, pues conduce a ambigüedades. Si una estación envía la cadena de bits 0001000, otros podrían interpretarla falsamente como 10000000 o 01000000, pues no pueden distinguir entre un emisor inactivo (0 voltios) y un bit 0 (0 voltios). Este problema se puede resolver utilizando +1 voltios para un 1 y -1 voltios para un 0, pero aún está el problema de que un receptor muestree la señal a una frecuencia ligeramente distinta a la que haya utilizado el emisor para generarla. Las diferentes velocidades de reloj pueden causar que el receptor y el emisor pierdan la sincronía respecto a dónde están los límites de bits, especialmente después de una serie larga de 0s consecutivos o una serie larga de 1s consecutivos.

Lo que se necesita es un mecanismo para que los receptores determinen sin ambigüedades el comienzo, el final o la mitad de cada bit sin referencia a un reloj externo. Dos de tales enfoques se llaman **codificación Manchester** y **codificación Manchester diferencial**. En la codificación Manchester, cada periodo de bit se divide en dos intervalos iguales. Un bit 1 binario se envía teniendo el voltaje alto durante el primer intervalo y bajo durante el segundo. Un 0 binario es justo lo inverso: primero bajo y después alto. Este esquema asegura que cada periodo de bit tenga una transición a la mitad, facilitando que el receptor se sincronice con el emisor. Una desventaja de la

codificación Manchester es que requiere el doble de ancho de banda que la codificación binaria directa, pues los pulsos son de la mitad de ancho. Por ejemplo, para enviar datos a 10 Mbps, la señal tiene que cambiar 20 millones de veces/seg. La codificación Manchester se muestra en la figura 4-16(b).

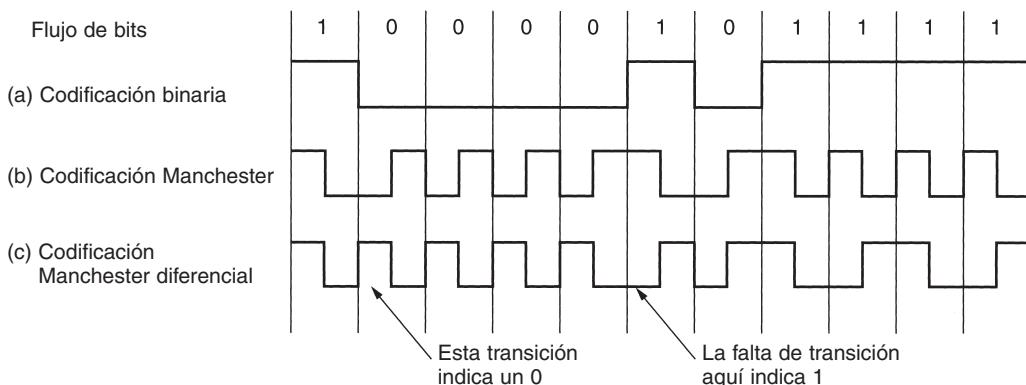


Figura 4-16. (a) Codificación binaria. (b) Codificación Manchester. (c) Codificación Manchester diferencial.

La codificación Manchester diferencial, que se muestra en la figura 4-16(c), es una variación de la codificación Manchester básica. En ella, un bit 1 se indica mediante la ausencia de una transición al inicio del intervalo. Un bit 0 se indica mediante la presencia de una transición al inicio del intervalo. En ambos casos también hay una transición a la mitad. El esquema diferencial requiere equipo más complejo, pero ofrece mejor inmunidad al ruido. Todos los sistemas Ethernet usan codificación Manchester debido a su sencillez. La señal alta es de + 0.85 voltios, y la señal baja es de - 0.85 voltios, dando un valor de DC de 0 voltios. Ethernet no utiliza la codificación Manchester diferencial, pero otras LANs (como la token ring 802.5) sí la utilizan.

4.3.3 El protocolo de subcapa MAC de Ethernet

En la figura 4-17 se muestra la estructura de trama original de DIX (DEC, Intel, Xerox). Cada trama inicia con un *Preámbulo* de 8 bytes, cada uno de los cuales contiene el patrón de bits 10101010. La codificación Manchester de este patrón produce una onda cuadrada de 10 MHz para 6.4 μ seg para permitir que el reloj del receptor se sincronice con el del emisor. Se les pide que permanezcan sincronizados por el resto de la trama, utilizando la codificación Manchester para mantener un registro de los límites de bits.

La trama contiene dos direcciones, una para el destino y una para el origen. El estándar permite direcciones de 2 y 6 bytes, pero los parámetros definidos para el estándar de banda base de 10 Mbps usan sólo direcciones de 6 bytes. El bit de orden mayor de la dirección de destino es 0 para direcciones ordinarias y 1 para direcciones de grupo. Las direcciones de grupo permiten que

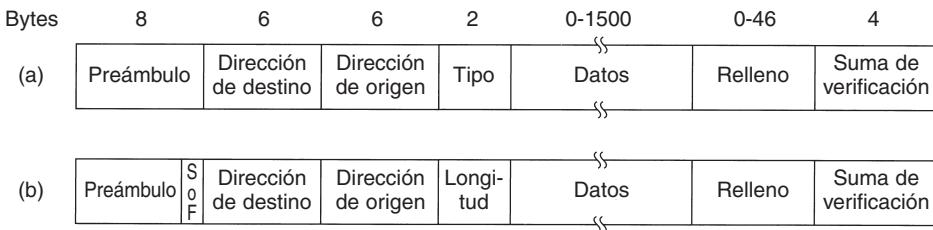


Figura 4-17. Formatos de trama. (a) Ethernet DIX. (b) IEEE 802.3.

varias estaciones escuchen en una sola dirección. Cuando una trama se envía a una dirección de grupo, todas las estaciones del grupo la reciben. El envío a un grupo de estaciones se llama **multidifusión (multicast)**. La dirección que consiste únicamente en bits 1 está reservada para **difusión (broadcast)**. Una trama que contiene sólo bits 1 en el campo de destino se acepta en todas las estaciones de la red. La diferencia entre difusión y multidifusión es lo suficientemente importante para garantizar la repetición. Una trama de multidifusión se envía a un grupo seleccionado de estaciones de la Ethernet; una trama de difusión se envía a todas las estaciones de la Ethernet. La multidifusión es más selectiva, pero involucra el manejo de grupos. La difusión es menos sofisticada pero no requiere manejo de grupos.

Otra característica interesante del direccionamiento es el empleo del bit 46 (adyacente al bit de orden mayor) para distinguir las direcciones locales de las globales. Las direcciones locales son asignadas por cada administrador de la red y no tienen significado fuera de la red local. En contraste, las direcciones globales son asignadas por el IEEE para asegurar que no haya dos estaciones en ningún lugar del mundo que tengan la misma dirección global. Con $48 - 2 = 46$ bits disponibles, hay unas 7×10^{13} direcciones globales. La idea es que cualquier estación pueda dirigir de manera exclusiva cualquier otra estación con sólo dar el número correcto de 48 bits. Es tarea de la capa de red encontrar la manera de localizar al destino.

A continuación está el campo de *Tipo*, que indica al receptor qué hacer con la trama. Es posible utilizar múltiples protocolos de capa de red al mismo tiempo en la misma máquina, por lo que cuando llega una trama de Ethernet, el *kernel* debe saber a cuál entregarle la trama. El campo de *Tipo* especifica a qué proceso darle la trama.

Después están los datos, de hasta 1500 bytes. Este límite fue elegido de manera algo arbitraria cuando se estableció el estándar DIX, principalmente con base en el hecho de que un transceptor necesita suficiente RAM para mantener toda una trama y la RAM era muy costosa en 1978. Un límite mayor podría haber significado más RAM y, por ende, un transceptor más costoso.

Además de haber una longitud de trama máxima, también hay una longitud mínima. Si bien algunas veces un campo de datos de 0 bytes es útil, causa problemas. Cuando un transceptor detecta una colisión, trunca la trama actual, lo que significa que los bits perdidos y las piezas de las tramas aparecen todo el tiempo en el cable. Para que Ethernet pueda distinguir con facilidad las tramas válidas de la basura, necesita que dichas tramas tengan una longitud de por lo menos 64 bytes, de la dirección de destino a la suma de verificación, incluyendo ambas. Si la porción de

datos de una trama es menor que 46 bytes, el campo de *Relleno* se utiliza para llenar la trama al tamaño mínimo.

Otra razón (más importante) para tener una trama de longitud mínima es evitar que una estación complete la transmisión de una trama corta antes de que el primer bit llegue al extremo más alejado del cable, donde podría tener una colisión con otra trama. Este problema se ilustra en la figura 4-18. En el momento 0, la estación *A*, en un extremo de la red, envía una trama. Llámemos τ al tiempo que tarda en llegar esta trama al otro extremo. Justo antes de que la trama llegue al otro extremo (es decir, en el momento $\tau - \epsilon$) la estación más distante, *B*, comienza a transmitir. Cuando *B* detecta que está recibiendo más potencia de la que está enviando, sabe que ha ocurrido una colisión, por lo que aborta su transmisión y genera una ráfaga de ruido de 48 bits para avisar a las demás estaciones. En otras palabras, atiborra el cable para asegurarse de que el emisor no ignore la colisión. En el momento 2τ , aproximadamente, el emisor ve la ráfaga de ruido y aborta también su transmisión; luego espera un tiempo aleatorio antes de reintentar.

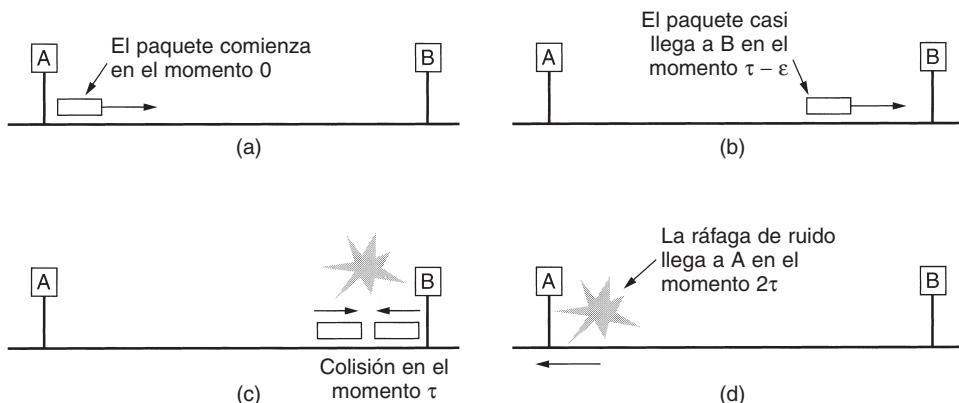


Figura 4-18. La detección de una colisión puede tardar hasta 2τ .

Si una estación intenta transmitir una trama muy corta, es concebible que ocurra una colisión, pero la transmisión se completa antes de que la ráfaga de ruido llegue de regreso, en el momento 2τ . El emisor entonces supondrá incorrectamente que la trama se envió con éxito. Para evitar que esta situación ocurra, todas las tramas deberán tardar más de 2τ para enviarse, de manera que la transmisión aún esté llevándose a cabo cuando la ráfaga de ruido regrese al emisor. Para una LAN de 10 Mbps con una longitud máxima de 2500 metros y cuatro repetidores (de la especificación 802.3), el tiempo de ida y vuelta (incluyendo el tiempo de propagación a través de los cuatro repetidores) se ha determinado a aproximadamente 50 μ seg en el peor caso, incluyendo el tiempo para pasar a través de los repetidores, que con certeza es diferente de cero. Por lo tanto, la trama mínima debe tomar por lo menos este tiempo en transmitir. A 10 Mbps, un bit tarda 100 nseg, por lo que 500 bits es la trama más pequeña que se garantiza funcionará. Para agregar algún margen de seguridad, este número se redondeó a 512 bits o 64 bytes. Las tramas con menos de 64 bytes se llenan con 64 bytes con el campo de *Relleno*.

A medida que aumente la velocidad de la red, la longitud mínima de la trama debe aumentar, o la longitud máxima del cable debe disminuir, de manera proporcional. Para una LAN de 2500 metros operando a 1 Gbps, el tamaño mínimo de trama tendría que ser de 6400 bytes. Como alternativa, el tamaño mínimo de trama podría ser de 640 bytes y la distancia máxima entre dos estaciones de 250 metros. Estas restricciones se vuelven cada vez más dolorosas a medida que progresamos hacia las redes de multigigabits.

El campo final de Ethernet es la *Suma de verificación*. De hecho, ésta es un código de *hash* de 32 bits de los datos. Si algunos bits de datos se reciben erróneamente (debido a ruido en el cable), es casi seguro que la suma de verificación está mal, y se detectará el error. El algoritmo de suma de verificación es una verificación de redundancia cíclica del tipo analizado en el capítulo 3. Simplemente realiza detección de errores, no corrección de errores hacia adelante.

Cuando el IEEE estandarizó Ethernet, el comité realizó dos cambios al formato DIX, como se muestra en la figura 4-17(b). El primero fue reducir el preámbulo a 7 bytes y utilizar el último byte para un delimitador de *Inicio de trama*, por compatibilidad con 802.4 y 802.5. El segundo fue cambiar el campo de *Tipo* en un campo de *Longitud*. Por supuesto, ahora no había forma de que el receptor supiera qué hacer con la trama entrante, pero ese problema se resolvió mediante la adición de un pequeño encabezado a la porción de datos para proporcionar esta información. Analizaremos el formato de la porción de datos cuando veamos el control lógico del enlace más adelante en este capítulo.

Desgraciadamente, en la época en que se publicó el 802.3 ya se utilizaba tanto hardware y software para la Ethernet DIX que muy pocos fabricantes y usuarios tenían deseos de convertir el campo de *Tipo* en uno de *Longitud*. En 1997, el IEEE tiró la toalla y dijo que las dos formas se ajustaban bien. Por fortuna, todos los campos de *Tipo* en uso antes de 1997 eran más grandes que 1500. En consecuencia, cualquier número menor que o igual a 1500 puede interpretarse como *Longitud*, y cualquier número más grande que 1500 puede interpretarse como *Tipo*. Ahora el IEEE puede afirmar que todo el mundo está utilizando su estándar y que cada quién puede seguir haciendo lo que desee sin preocuparse.

4.3.4 Algoritmo de retroceso exponencial binario

Ahora veamos cómo se efectúa el proceso de aleatorización cuando ocurre una colisión. El modelo es el de la figura 4-5. Tras una colisión, el tiempo se divide en ranuras discretas cuya longitud es igual al tiempo de propagación de ida y vuelta de peor caso en el cable (2τ). Tomando en cuenta la ruta más larga permitida por Ethernet, el tiempo de ranura se estableció en 512 tiempos de bit, o 51.2 μ seg.

Tras la primera colisión, cada estación espera 0 o 1 tiempos de ranura antes de intentarlo de nuevo. Si dos estaciones entran en colisión y ambas escogen el mismo número aleatorio, habrá una nueva colisión. Despues de la segunda colisión, cada una escoge 0, 1, 2 o 3 al azar y espera ese número de tiempos de ranura. Si ocurre una tercera colisión (la probabilidad de que esto ocurra es

de 0.25), entonces para la siguiente vez el número de ranuras a esperar se escogerá al azar del intervalo 0 a $2^3 - 1$.

En general, tras i colisiones, se escoge un número aleatorio entre 0 y $2^i - 1$, y se salta ese número de ranuras. Sin embargo, tras haberse alcanzado 10 colisiones, el intervalo de aleatorización se congela en un máximo de 1023 ranuras. Tras 16 colisiones, el controlador tira la toalla e informa de un fracaso a la computadora. La recuperación posterior es responsabilidad de las capas superiores.

Este algoritmo, llamado **retroceso exponencial binario**, se escogió para adaptar en forma dinámica el número de estaciones que intentan transmitir. Si el intervalo de aleatorización para todas las colisiones fuera de 1023, la posibilidad de que chocaran dos estaciones una segunda vez será insignificante, pero la espera promedio tras una colisión será de cientos de tiempos de ranura, lo que introduce un retardo significativo. Por otra parte, si cada estación siempre se retrasa 0 o 1 ranura, entonces, al tratar de transmitir 100 estaciones al mismo tiempo, habría colisiones una y otra vez, hasta que 99 de ellas escogieran 1 y la estación restante escogiera 0. Esto podría tomar años. Haciendo que el intervalo de aleatorización crezca de manera exponencial a medida que ocurren más y más colisiones, el algoritmo asegura un retardo pequeño cuando sólo unas cuantas estaciones entran en colisión, pero también asegura que la colisión se resuelva en un intervalo razonable cuando hay colisiones entre muchas estaciones. Truncar el retroceso a 1023 evita que el límite crezca demasiado.

Como se ha descrito hasta ahora, CSMA/CD no proporciona confirmación de recepción. Ya que la simple ausencia de colisiones no garantiza que los bits no fueron alterados por picos de ruido en el cable, para una comunicación confiable el destino debe verificar la suma de verificación y, de ser correcta, regresar al origen una trama de confirmación de recepción. Por lo general, esta confirmación sería simplemente otra trama, en lo que concierne al protocolo, y tendría que pelear por tiempo de canal de la misma manera que una trama de datos. Sin embargo, una modificación sencilla del algoritmo de contención permite la confirmación rápida de la recepción de una trama (Tokoro y Tamaru, 1977). Todo lo que se necesitará es reservar para la estación de destino la primera ranura de contención que siga a la siguiente transmisión exitosa. Desgraciadamente, el estándar no proporciona esta posibilidad.

4.3.5 Desempeño de Ethernet

Ahora examinaremos brevemente el desempeño de Ethernet en condiciones de carga pesada y constante, es decir, k estaciones siempre listas para transmitir. Es complicado un análisis riguroso del algoritmo de retroceso exponencial binario. En cambio, seguiremos a Metcalfe y Boggs (1976) y supondremos una probabilidad constante de retransmisión en cada ranura. Si cada estación transmite durante una ranura de contención con una probabilidad p , la probabilidad A de que una estación adquiera el canal durante esa ranura es de:

$$A = kp(1-p)^{k-1} \quad (4-5)$$

A se maximiza cuando $p = 1/k$, con $A \rightarrow 1/e$ conforme $k \rightarrow \infty$. La probabilidad de que el intervalo de contención tenga exactamente j ranuras es de $A(1 - A)^{j-1}$, por lo que el número medio de ranuras por contención está dado por

$$\sum_{j=0}^{\infty} jA(1 - A)^{j-1} = \frac{1}{A}$$

Puesto que cada ranura tiene una duración de 2τ , el intervalo medio de contención, w , es $2\tau/A$. Suponiendo una p óptima, el número medio de ranuras de contención nunca es mayor que e , por lo que w es, cuando mucho, $2\tau e \approx 5.4\tau$.

Si la trama media tarda P segundos en transmitirse, cuando muchas estaciones tienen tramas por enviar,

$$\text{Eficiencia del canal} = \frac{P}{P + 2\tau/A} \quad (4.6)$$

Aquí vemos dónde entra la distancia máxima de cable entre dos estaciones en las cifras de desempeño, dando lugar a topologías distintas de las de la figura 4-15(a). Cuanto mayor sea la longitud del cable, mayor será el intervalo de contención. Debido a esta observación el estándar Ethernet especifica una longitud máxima de cable.

Es instructivo formular la ecuación 4-6 en términos de la longitud de trama, F , el ancho de banda de la red, B , la longitud del cable, L , y la velocidad de propagación de la señal, c , para el caso óptimo de e ranuras de contención por trama. Con $P = F/B$, la ecuación 4-6 se convierte en

$$\text{Eficiencia del canal} = \frac{1}{1 + 2BLe/cF} \quad (4.7)$$

Cuando el segundo término del denominador es grande, la eficiencia de la red es baja. Más específicamente, un aumento en el ancho de banda o la distancia de la red (el producto BL) reduce la eficiencia de una trama de tamaño dado. Desgraciadamente, mucha investigación sobre hardware de redes está enfocada precisamente a aumentar este producto. La gente quiere un gran ancho de banda a través de distancias grandes (por ejemplo, en las MANs de fibra óptica), lo que sugiere que Ethernet tal vez no sea el mejor sistema para estas aplicaciones. Veremos otras formas de implementación de Ethernet cuando analicemos la Ethernet conmutada, más adelante en este capítulo.

En la figura 4-19 se presenta gráficamente la eficiencia del canal contra el número de estaciones listas para $2\tau = 51.2 \mu\text{seg}$ y una tasa de datos de 10 Mbps usando la ecuación 4-7. Con un tiempo de ranura de 64 bytes, no es sorprendente que las tramas de 64 bytes no sean eficientes. Por otra parte, con tramas de 1024 bytes y un valor asintótico de e ranuras de 64 bytes por intervalo de contención, el periodo de contención tiene 174 bytes de longitud y la eficiencia es de 0.85.

Para determinar el número medio de estaciones listas para transmitir en condiciones de carga alta, podemos usar la siguiente (y burda) observación. Cada trama atrapa el canal durante un periodo de contención más un tiempo de transmisión de trama, lo que da un total de $P + w$ seg. El número de tramas por segundo, por lo tanto, es $1/(P + w)$. Si cada estación genera tramas a una

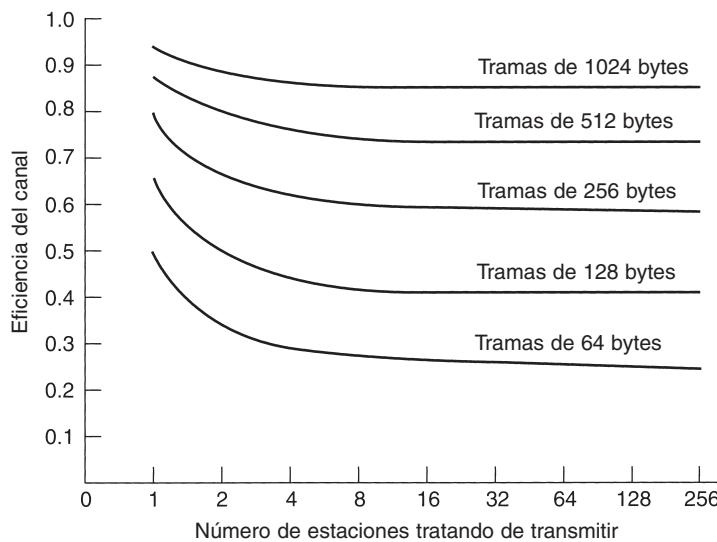


Figura 4-19. Eficiencia de Ethernet a 10 Mbps con tiempos de ranura de 512 bits.

velocidad media de λ tramas/seg, cuando el sistema está en el estado k , la tasa de entrada total de todas las estaciones no bloqueadas combinadas es de $k\lambda$ tramas/seg. Ya que en equilibrio las tasas de entrada y de salida deben ser idénticas, podemos igualar esas dos expresiones y despejar k . (Observe que w es una función de k .) En (Bertsekas y Gallager, 1992) se da un análisis más elaborado.

Probablemente vale la pena mencionar que se ha realizado una gran cantidad de análisis teóricos de desempeño de Ethernet (y otras redes). Prácticamente todos estos trabajos han supuesto que el tráfico es Poisson. A medida que los investigadores han comenzado a examinar datos reales, se ha hecho evidente que el tráfico en redes pocas veces es Poisson, sino autosimilar (Paxson y Floyd, 1994, y Willinger y cols., 1995). Lo que esto significa es que el promedio durante períodos grandes no hace más uniforme el tráfico. La cantidad media de paquetes en cada minuto de una hora tiene tanta variación como la cantidad media de paquetes en cada segundo de un minuto. La consecuencia de este descubrimiento es que la mayoría de los modelos de tráfico de red no se aplican al mundo real y deben tomarse con escepticismo.

4.3.6 Ethernet conmutada

A medida que se agregan más y más estaciones a una Ethernet, aumenta el tráfico. En algún momento, la LAN se saturará. Una solución al problema es utilizar una velocidad mayor, digamos 100 Mbps en lugar de 10 Mbps. Pero con el crecimiento de la multimedia, incluso una Ethernet de 100 Mbps o de 1 Gbps puede saturarse.

Afortunadamente existe una solución diferente para tratar con el aumento de carga: una Ethernet conmutada, como la que se muestra en la figura 4-20. El corazón de este sistema es un **conmutador** (*switch*) que contiene una matriz de conmutación de alta velocidad y espacio

(típicamente) para 4 a 32 tarjetas de línea, cada una de las cuales contiene de uno a ocho conectores. Lo más común es que cada conector tenga una conexión de cable de par trenzado 10Base-T a una sola computadora *host*.

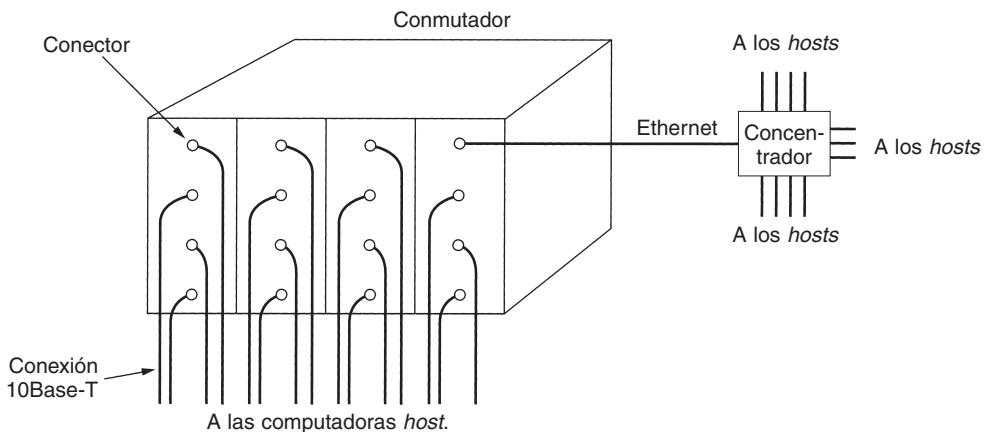


Figura 4-20. Ejemplo sencillo de Ethernet commutada.

Cuando una estación quiere transmitir una trama Ethernet, envía una trama estándar al commutador. La tarjeta que recibe la trama la revisa para ver si está destinada a una de las otras estaciones conectadas a la misma tarjeta. De ser así, la trama se copia ahí. Si no, la trama se envía a través de la matriz de conmutación de alta velocidad a la tarjeta de la estación de destino. Por lo general, dicha matriz de conmutación funciona a más de 1 Gbps usando un protocolo patentado.

¿Qué ocurre si dos máquinas conectadas a la misma tarjeta de conexión transmiten tramas al mismo tiempo? Depende de la manera en que haya sido construida la tarjeta. Una posibilidad es que todos los puertos de la tarjeta estén alambrados entre sí para formar una LAN local dentro de la tarjeta. Las colisiones en esta LAN en tarjeta se detectan y manejan igual que cualquier otra colisión en una red CSMA/CD, en las que las retransmisiones utilizan el algoritmo de retroceso exponencial binario. Con este tipo de tarjeta sólo es posible una transmisión por tarjeta en un momento dado, pero todas las tarjetas pueden estar transmitiendo en paralelo. Con este diseño, cada tarjeta forma su propio **dominio de colisión**, independiente de los demás. Con sólo una estación por dominio de colisión, las colisiones son imposibles y el desempeño se mejora.

Con el otro tipo de tarjeta de conexión, cada puerto de entrada se almacena en un búfer, por lo que las tramas de entrada se almacenan en la RAM de la tarjeta conforme llegan. Este diseño permite que todos los puertos de entrada reciban (y transmitan) tramas al mismo tiempo, para una operación en paralelo completamente dúplex, algo que no es posible con CSMA/CD en un solo canal. Una vez que se ha recibido por completo la trama, la tarjeta puede determinar si la trama está destinada a otro puerto de la misma tarjeta o a un puerto distante. En el primer caso, puede transmitirse directamente al destino. En el segundo, debe transmitirse a través de la matriz de conmutación de alta velocidad a la tarjeta apropiada. Con este diseño, cada puerto es un dominio de colisión independiente, por lo que no ocurren colisiones. En muchos casos, la velocidad real

de transporte total del sistema puede aumentarse en un orden de magnitud respecto al 10Base-5, que tiene un solo dominio de colisión para todo el sistema.

Dado que el conmutador sólo espera tramas Ethernet estándar en cada puerto de entrada, es posible utilizar algunos de los puertos como concentradores. En la figura 4-20, el puerto de la esquina superior derecha no está conectado a una sola estación, sino a un concentrador de 12 puertos. A medida que llegan las tramas al concentrador, luchan por el canal de la manera usual, con colisiones y retroceso binario. Las tramas que tienen éxito llegan al conmutador y ahí se tratan como cualquier otra trama de entrada: se conmutan a la línea de salida correcta a través de la matriz de conmutación de alta velocidad. Los concentradores son más baratos que los conmutadores, pero debido a los precios a la baja de los conmutadores, se están haciendo obsoletos rápidamente. No obstante, los concentradores heredados aún existen.

4.3.7 Fast Ethernet

Al principio, 10 Mbps parecían el cielo, al igual que los módems de 1200 bps parecieron el cielo a los primeros usuarios de módems acústicos de 300 bps. Pero la novedad desapareció rápidamente. Como un tipo de corolario a la Ley de Parkinson (“El trabajo se expande hasta agotar el tiempo destinado a realizarlo”), tal parecía que los datos se expandieron hasta agotar el ancho de banda disponible para su transmisión. Para aumentar la velocidad, varios grupos de la industria propusieron dos nuevas LANs ópticas basadas en anillos. Una se llamó **FDDI (Interfaz de Datos Distribuidos por Fibra)** y la otra se llamó **Canal de fibra**. Para acortar la historia, si bien ambas se utilizaban como redes dorsales, ninguna se popularizó en sistemas de escritorio. En ambos casos, la administración de estaciones era muy complicada, lo que llevó a *chips* complejos y precios altos. La lección que debió aprenderse a partir de aquí fue KISS (*Keep It Simple, Stupid; Manténgalo Simple, Tonto*).

En cualquier caso, al no popularizarse las LANs ópticas quedó un hueco para redes Ethernet de una gran variedad a velocidades superiores a 10 Mbps. Muchas instalaciones necesitaban más ancho de banda y, por lo tanto, tenían numerosas LANs de 10 Mbps conectadas por una maraña de repetidores, puentes, enrutadores y puertas de enlace, aunque los administradores de redes algunas veces sentían que las conexiones parecían estar hechas con goma de mascar y tela metálica, es decir, endebles y poco seguras.

Fue en este entorno que el IEEE convocó al comité 802.3 en 1992 con instrucciones de crear una LAN más rápida. Una propuesta fue mantener 802.3 exactamente como estaba, pero hacerla más rápida. Otra propuesta fue rehacerla en su totalidad para darle muchas características nuevas, como tráfico en tiempo real y voz digitalizada, pero mantener el nombre antiguo (por razones de marketing). Después de algunas discusiones, el comité decidió mantener la Ethernet 802.3 tal como estaba, pero hacerla más rápida. Las personas que apoyaban la propuesta contraria hicieron lo que cualquier persona de la industria de la computación habría hecho bajo estas circunstancias —unieron fuerzas y formaron su propio comité y estandarizaron su LAN de todas maneras (que con el tiempo se llamó 802.12). Ésta fracasó rotundamente.

El comité 802.3 decidió crear una Ethernet mejorada por tres razones principales:

1. La necesidad de compatibilidad hacia atrás con las LANs Ethernet existentes.
2. El miedo de que un nuevo protocolo tuviera problemas no previstos.
3. El deseo de terminar el trabajo antes de que la tecnología cambiara.

El trabajo se terminó rápidamente (mediante las normas de los comités de estándares), y el resultado, **802.3u**, fue aprobado oficialmente por el IEEE en junio de 1995. Técnicamente, 802.3u no es un nuevo estándar, sino un agregado al estándar existente 802.3 (para enfatizar su compatibilidad hacia atrás). Puesto que prácticamente todos lo llaman **Fast Ethernet**, en lugar de 802.3u, nosotros también lo haremos.

La idea básica detrás de Fast Ethernet era sencilla: mantener todos los formatos anteriores, interfaces y reglas de procedimientos, y sólo reducir el tiempo de bits de 100 nseg a 10 nseg. Técnicamente, habría sido posible copiar 10Base-5 o 10Base-2 y aún detectar colisiones a tiempo con sólo reducir la longitud máxima de cable por un factor de diez. Sin embargo, las ventajas del cableado 10Base-T eran tan abrumadoras que Fast Ethernet se basa por completo en este diseño. Por lo tanto, todos los sistemas Fast Ethernet utilizan concentradores y conmutadores; no se permiten cables con múltiples derivaciones vampiro ni conectores BNC.

Sin embargo, aún se tienen que tomar algunas decisiones, la más importante de las cuáles es qué tipos de cable soportar. Un contendiente era el cable de par trenzado categoría 3. El argumento a su favor era que prácticamente todas las oficinas en el mundo occidental tienen por lo menos cuatro cables de par trenzado categoría tres (o mejor) que van de la oficina hacia un gabinete de cableado telefónico dentro de una distancia de 100 metros. Algunas veces existen dos de esos cables. Por lo tanto, el uso del cable de par trenzado categoría 3 hace posible cablear las computadoras de escritorio mediante Fast Ethernet sin tener que volver a cablear el edificio, lo cual es una enorme ventaja para muchas organizaciones.

La principal desventaja del cable de par trenzado categoría 3 es su incapacidad de llevar señales de 200 megabaudios (100 Mbps con codificación Manchester) a una distancia de hasta 100 metros, que es la distancia máxima de computadora a concentrador especificada para 10Base-T (vea la figura 4-13). En contraste, el cable de par trenzado categoría 5 puede manejar 100 metros con facilidad, y la fibra puede ir mucho más rápido. El arreglo elegido fue permitir las tres posibilidades, como se muestra en la figura 4-21, pero fortalecer la solución categoría 3 para darle la capacidad de transmisión adicional necesaria.

Nombre	Cable	Segmento máximo	Ventajas
100Base-T4	Par trenzado	100 m	Utiliza UTP categoría 3
100Base-TX	Par trenzado	100 m	Dúplex total a 100 Mbps (UTP cat 5)
100Base-FX	Fibra óptica	2000 m	Dúplex total a 100 Mbps; distancias largas

Figura 4-21. El cableado original de Fast Ethernet.

El esquema UTP categoría 3, llamado **100Base-T4**, utiliza una velocidad de señalización de 25 MHz, tan sólo 25 por ciento más rápida que los 20 MHz de la Ethernet estándar (recuerde que la codificación Manchester, como se muestra en la figura 4-16, requiere dos periodos de reloj para cada uno de los 10 millones de bits cada segundo). Sin embargo, para alcanzar el ancho de banda necesario, 100Base-T4 requiere cuatro cables de par trenzado. Debido a que el cableado telefónico estándar durante décadas ha tenido cuatro cables de par trenzado por cable, la mayoría de las oficinas puede manejar esto. Por supuesto, esto significa ceder su teléfono de la oficina, pero seguramente eso es un precio pequeño a cambio de correo electrónico más rápido.

De los cuatro cables de par trenzado, uno siempre va al concentrador, uno siempre sale del concentrador y los otros dos son intercambiables a la dirección actual de transmisión. Para obtener el ancho de banda necesario, no se utiliza la codificación Manchester, pero con relojes modernos y distancias cortas, ya no es necesaria. Además, se envían señales ternarias, para que durante un periodo de reloj el cable pueda contener un 0, un 1 o un 2. Con tres cables de par trenzado y la señalización ternaria, se puede transmitir cualquiera de 27 símbolos posibles, con lo que se pueden enviar 4 bits con algo de redundancia. Transmitir 4 bits en cada uno de los 25 millones de ciclos de reloj por segundo da los 100 Mbps necesarios. Además, siempre hay un canal de regreso de 33.3 Mbps que utiliza el resto del cable de par trenzado. No es probable que este esquema, conocido como **8B/6T** (8 bits se convierten en 6 trits), gane un premio por elegancia, pero funciona con la planta de cableado existente.

Para el cableado categoría 5, el diseño **100Base-TX** es más simple porque los cables pueden manejar velocidades de reloj de 125 MHz. Sólo se utilizan dos cables de par trenzado por estación, uno para enviar y otro para recibir. La codificación binaria directa no se utiliza; en su lugar se toma un esquema llamado **4B/5B** tomado de las redes FDDI, y compatible con ellas. Cada grupo de cinco periodos de reloj, cada uno de los cuales contiene uno de dos valores de señal, da 32 combinaciones. Dieciséis de estas combinaciones se utilizan para transmitir los cuatro grupos de bits 0000, 0001, 0010, ..., 1111. Algunos de los 16 restantes se utilizan para propósitos de control, como el marcado de límites de tramas. Las combinaciones utilizadas se han elegido cuidadosamente para proporcionar suficientes transiciones para mantener sincronización de reloj. El sistema 100Base-TX es de dúplex total; las estaciones pueden transmitir a 100 Mbps y recibir a 100 Mbps al mismo tiempo. Con frecuencia, 100Base-TX y 100Base-T4 se llaman en conjunto **100Base-T**.

La última opción, **100Base-FX**, utiliza dos filamentos de fibra multimodo, una para cada dirección, por lo que también es dúplex total con 100 Mbps en cada dirección. Además, la distancia entre una estación y el concentrador puede ser de hasta 2 km.

En respuesta a la demanda popular, en 1997 el comité 802 agregó un nuevo tipo de cableado, 100Base-T2, que permite que la Fast Ethernet se ejecute a través de dos pares de cables existentes de categoría 3. Sin embargo, se necesita un procesador de señales digital sofisticado para manejar el esquema de codificación requerido, lo que hace de esta opción algo muy costoso. Hasta ahora su uso es muy inusual debido a su complejidad y costo, así como al hecho de que muchos edificios de oficinas se han vuelto a cablear con UTP categoría 5.

Con 100Base-T son posibles dos tipos de dispositivos de interconexión: concentradores y commutadores, como se muestra en la figura 4-20. En un concentrador, todas las líneas entrantes (o al menos todas las líneas que llegan a una tarjeta de conexión) se conectan lógicamente, formando

un solo dominio de colisión. Se aplican todas las reglas estándar, entre ellas el algoritmo de retroceso exponencial binario, por lo que el sistema funciona de la misma manera que la Ethernet antigua. En particular, sólo una estación a la vez puede transmitir. En otras palabras, los concentradores requieren comunicación semidúplex.

En un conmutador, cada trama entrante se almacena en el búfer de una tarjeta de conexión y se pasa a través de una matriz de conmutación de alta velocidad de la tarjeta de origen a la de destino, si es necesario. La matriz de conmutación no se ha estandarizado, ni lo necesita, debido a que está completamente oculta dentro del conmutador. Si la experiencia pasada sirve de algo, los fabricantes de conmutadores competirán con ardor para producir matrices de conmutación más veloces para mejorar la velocidad real de transporte del sistema. Debido a que los cables 100Base-FX son muy largos para el algoritmo de colisiones de la Ethernet, deben conectarse a conmutadores, de manera que cada uno sea un dominio de colisión en sí mismo. Los concentradores no están permitidos con 100Base-FX.

Como nota final, casi todos los conmutadores pueden manejar una mezcla de estaciones de 10 y 100 Mbps, para facilitar la actualización. Conforme un sitio adquiera más y más estaciones de trabajo de 100 Mbps, todo lo que tiene que hacer es comprar la cantidad necesaria de tarjetas de línea e insertarlas en el conmutador. De hecho, el estándar mismo proporciona una forma para que dos estaciones negocien de manera automática la velocidad óptima (10 o 100 Mbps) y el tipo de transmisión dúplex (semi o total). La mayoría de los productos de Fast Ethernet utilizan esta característica para autoconfigurarse.

4.3.8 Gigabit Ethernet

La tinta apenas se estaba secando en el estándar de la Fast Ethernet cuando el comité 802 comenzó a trabajar en una Ethernet aún más rápida (1995). Se conoció como **Gigabit Ethernet** y fue aprobada por el IEEE en 1998 bajo el nombre 802.3z. Este identificador sugiere que la Gigabit Ethernet va a ser el final de la línea, a menos que alguien invente rápidamente una letra después de la z. A continuación analizaremos algunas de las características principales de la Gigabit Ethernet. Para mayor información, vea (Seifert, 1998).

Los objetivos del comité 802.3z eran esencialmente los mismos que los del comité 802.3u: hacer que Ethernet fuera 10 veces más rápida y que permaneciera compatible hacia atrás con todos los estándares Ethernet existentes. En particular, Gigabit Ethernet tiene que ofrecer servicio de datagramas sin confirmación de recepción con difusión y multidifusión, utilizar el mismo esquema de direccionamiento de 48 bits que el actual y mantener el mismo formato de trama, incluyendo los tamaños mínimo y máximo de trama. El estándar final cumple con todos estos objetivos.

Todas las configuraciones de Gigabit Ethernet son de punto a punto en lugar de múltiples derivaciones como en el estándar original de 10 Mbps, ahora conocido como **Ethernet clásica**. En la configuración más simple de Gigabit Ethernet, que se muestra en la figura 4-22(a), dos computadoras están conectadas de manera directa entre sí. Sin embargo, el caso más común es tener un conmutador o un concentrador conectado a múltiples computadoras y posiblemente a conmutadores o concentradores adicionales, como se muestra en la figura 4-22(b). En ambas configuraciones cada cable Ethernet individual tiene exactamente dos dispositivos en él, ni más ni menos.

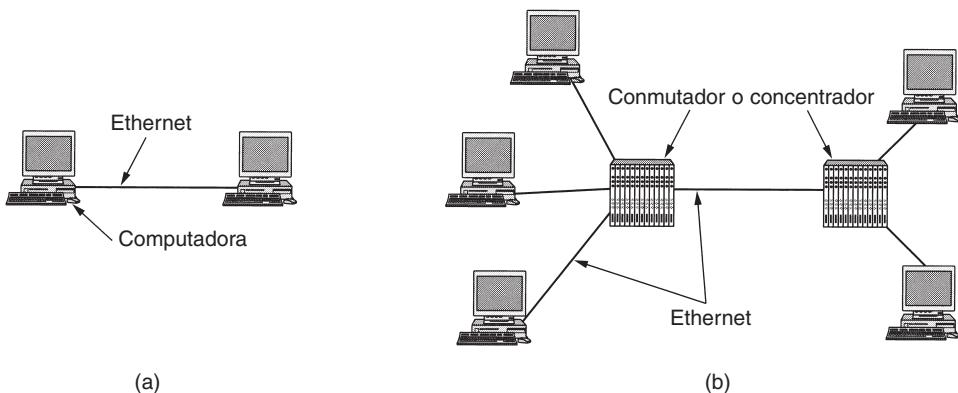


Figura 4-22. (a) Ethernet de dos estaciones. (b) Ethernet con múltiples estaciones.

Gigabit Ethernet soporta dos modos diferentes de funcionamiento: modo de dúplex total y modo de semidúplex. El modo “normal” es el de dúplex total, el cual permite tráfico en ambas direcciones al mismo tiempo. Este modo se utiliza cuando hay un conmutador central conectado a computadoras (o a otros conmutadores) en el periférico. En esta configuración, todas las líneas se almacenan en el búfer a fin de que cada computadora y conmutador pueda enviar tramas siempre que lo deseé. El emisor no tiene que detectar el canal para ver si alguien más lo está utilizando debido a que la contención es imposible. En la línea entre una computadora y un conmutador, la computadora es el único emisor posible en esa línea al conmutador y la transmisión tiene éxito aun cuando el conmutador esté enviado actualmente una trama a la computadora (porque la línea es de dúplex total). Debido a que no hay contención, no se utiliza el protocolo CSMA/CD y la longitud máxima del cable se determina con base en la fuerza de la señal más que en el tiempo que tarda una ráfaga de ruido en regresar al emisor en el peor caso. Los conmutadores son libres de mezclar e igualar velocidades. La autoconfiguración se soporta al igual que en Fast Ethernet.

El otro modo de operación, semidúplex, se utiliza cuando las computadoras están conectadas a un concentrador en lugar de a un conmutador. Un concentrador no almacena en el búfer las tramas entrantes. En su lugar, conecta en forma eléctrica todas las líneas internamente, simulando el cable con múltiples derivaciones que se utiliza en la Ethernet clásica. En este modo las colisiones son posibles, por lo que es necesario el protocolo CSMA/CD estándar. Debido a que una trama mínima (de 64 bytes) ahora puede transmitirse 100 veces más rápido que en la Ethernet clásica, la distancia máxima es 100 veces menor, o 25 metros, para mantener la propiedad esencial de que el emisor aún transmita cuando la ráfaga de ruido vuelva a él, incluso en el peor caso. Con un cable de 2500 metros de longitud, el emisor de una trama de 64 bytes a 1 Gbps podría terminar su transmisión antes de que la trama recorra una décima del camino, y muchísimo antes de que llegue al otro extremo y regrese.

El comité 802.3z consideró un radio de 25 metros como inaceptable y agregó dos características al estándar para incrementar el radio. La primera, llamada **extensión de portadora**, esencialmente indica al hardware que agregue su propio relleno después de la trama normal para extenderla a 512 bytes. Puesto que este relleno es agregado por el hardware emisor y eliminado

por el hardware receptor, el software no toma parte en esto, lo que significa que no es necesario realizar cambios al software existente. Por supuesto, utilizar 512 bytes de ancho de banda para transmitir 46 bytes de datos de usuario (la carga útil de una trama de 64 bytes) tiene una eficiencia de línea de 9%.

La segunda característica, llamada **ráfagas de trama**, permite que un emisor transmita una secuencia concatenada de múltiples tramas en una sola transmisión. Si la ráfaga total es menor que 512 bytes, el hardware la rellena nuevamente. Si suficientes tramas están esperando la transmisión, este esquema es muy eficiente y se prefiere antes que la extensión de portadora. Estas nuevas características amplían el radio de red de 200 metros, que probablemente es suficiente para la mayoría de las oficinas.

Sin duda, una organización difícilmente enfrentará el problema de comprar e instalar tarjetas Gigabit Ethernet para obtener mayor rendimiento y después conectar las computadoras con un concentrador para simular una Ethernet clásica con todas sus colisiones. Si bien los concentradores son un tanto más baratos que los commutadores, las tarjetas de interfaz Gigabit Ethernet aún son relativamente costosas. Por lo tanto, tratar de economizar comprando un concentrador barato y reducir drásticamente el desempeño del nuevo sistema es algo impensable. Además, la compatibilidad hacia atrás es sagrada en la industria de la computación, por lo que se le pidió al comité 802.3z que la añadiera.

Como se lista en la figura 4-23, Gigabit Ethernet soporta tanto el cableado de fibra óptica como el de cobre. Transmitir señales a o aproximadamente a 1 Gbps a través de fibra significa que la fuente de luz debe encenderse y apagarse en 1 nseg. Los LEDs simplemente no pueden funcionar con esta rapidez, por lo que se necesitan láseres. Se permiten dos longitudes de onda: 0.85 micras (Corto) y 1.3 micras (Largo). Los láseres a 0.85 micras son más económicos pero no funcionan en una fibra de modo sencillo.

Nombre	Cable	Segmento máximo	Ventajas
1000Base-SX	Fibra óptica	550 m	Fibra multimodo (50, 62.5 micras)
1000Base-LX	Fibra óptica	5000 m	Sencilla (10 μ) o multimodo (50, 62.5 μ)
1000Base-CX	2 pares de STP	25 m	Cable de par trenzado blindado
1000Base-T	4 Pares de UTP	100 m	UTP categoría 5 estándar

Figura 4-23. Cableado de Gigabit Ethernet.

Se permiten tres diámetros de fibra: 10, 50 y 62.5 micras. El primero es para el modo sencillo y los últimos dos son para multimodo. Sin embargo, no se permiten las seis combinaciones y la distancia máxima depende de la combinación que se utilice. Los números que se dan en la figura 4-23 son para el mejor de los casos. En particular, 5000 metros son alcanzables únicamente con láseres de 1.3 micras que funcionen a través de fibra de 10 micras en modo sencillo, pero ésta es la mejor opción para redes dorsales y se espera que sea popular, a pesar de ser la opción más costosa.

La opción 1000Base-CX utiliza cables de cobre blindados cortos. Su problema es que compite con la fibra de alto desempeño por una parte, y con el UTP más económico por la otra. No es probable que se utilice mucho, si es que se utiliza.

La última opción son paquetes de cuatro cables UTP categoría 5 que trabajan juntos. Debido a que la mayor parte de este cableado ya está instalada, es probable que sea la Gigabit Ethernet de la gente pobre.

Gigabit Ethernet utiliza nuevas reglas de codificación en las fibras. La codificación Manchester a 1 Gbps podría requerir una señal de 2 Gbaudios, lo cual era considerado muy difícil y también un desperdicio de ancho de banda. En su lugar se eligió un nuevo esquema, llamado **8B/10B**, que se basa en un canal de fibra. Cada byte de 8 bits está codificado en la fibra como 10 bits, de aquí el nombre 8B/10B. Debido a que hay 1024 palabras codificadas posibles para cada byte de entrada, hay algo de libertad al elegir cuáles palabras codificadas permitir. Las siguientes dos reglas se utilizaron al realizar las elecciones:

1. Ninguna palabra codificada podría tener más de cuatro bits idénticos en una fila.
2. Ninguna palabra codificada podría tener más de seis bits 0 o seis bits 1.

Estas elecciones se realizaron para mantener suficientes transiciones en el flujo para asegurarse de que el receptor continúe sincronizado con el emisor y también para mantener la cantidad de bits 0 y bits 1 en la fibra tan cerca del equilibrio como sea posible. Además, muchos bytes de entrada tienen dos palabras codificadas posibles asignadas a ellos. Cuando el codificador tiene la opción de palabras codificadas, siempre elige la palabra codificada que tiende a igualar la cantidad de 0s y 1s transmitidos hasta ese momento. Este énfasis en igualar 0s y 1s es necesario para mantener el componente DC de la señal tan bajo como sea posible para permitirle pasar a través de transformadores no modificados. Si bien los científicos de la computación no son afectos a que las propiedades de los transformadores dicten sus esquemas de codificación, algunas veces la vida es así.

Las Gigabit Ethernet que utilizan 1000Base-T emplean un esquema de codificación diferente debido a que cronometrar el tiempo de los datos en el cable de cobre en 1 nseg es muy difícil. Esta solución utiliza cuatro cables de par trenzado categoría 5 para permitir que se transmitan en paralelo cuatro símbolos. Cada uno de ellos se codifica utilizando uno de cinco niveles de voltaje. Este esquema permite que un solo símbolo codifique 00, 01, 10, 11 o un valor especial para propósitos de control. Por lo tanto, hay 2 bits de datos por cable de par trenzado u 8 bits de datos por ciclo de reloj. El reloj se ejecuta a 125 MHz, y permite una operación de 1 Gbps. La razón para permitir cinco niveles de voltaje en lugar de cuatro es tener combinaciones sobrantes para propósitos de entrampado y control.

1 Gbps es una velocidad muy alta. Por ejemplo, si un receptor está ocupado con otra tarea por incluso un 1 msec y no vacía el búfer de entrada en alguna línea, podrían haberse acumulado ahí hasta 1953 tramas en ese espacio de 1 ms. Además, cuando una computadora en una Gigabit Ethernet está enviando datos en la línea a una computadora en una Ethernet clásica, es muy probable que sucedan rebases de búfer. Como consecuencia de estas dos observaciones, Gigabit Ethernet soporta control de flujo (como lo hace la Fast Ethernet, aunque los dos son diferentes).

El control de flujo consiste en que un extremo envíe una trama de control especial al otro extremo indicándole que se detenga por algún tiempo. Las tramas de control son tramas comunes de Ethernet que contienen un tipo de 0x8808. Los primeros dos bytes del campo de datos dan el comando; los bytes exitosos proporcionan los parámetros, si es que hay. Para control de flujo, se utilizan las tramas PAUSE, en las que el parámetro indica cuánto tiempo detenerse, en unidades de tiempo de la trama más pequeña. Para la Gigabit Ethernet, la unidad de tiempo es 512 nseg, lo que permite pausas de 33.6 mseg.

Una vez que la Gigabit Ethernet se estandarizó, el comité 802 se aburrió y quiso volver al trabajo. El IEEE les dijo que iniciaran una Ethernet de 10 gigabits. Después de buscar arduamente una letra que siguiera a la z, el comité abandonó ese enfoque y se concentró en los sufijos de dos letras. Comenzó el trabajo y ese estándar fue aprobado por el IEEE en el 2002 como 802.3ae. ¿Es posible que le siga una Ethernet de 100 gigabits?

4.3.9. Estándar IEEE 802.2: control lógico del enlace

Tal vez ahora sea el momento de dar un paso atrás y comparar lo que hemos aprendido en este capítulo con lo que estudiamos en el anterior. En el capítulo 3 vimos la manera en que dos máquinas se podían comunicar de manera confiable a través de una línea inestable usando varios protocolos de enlace de datos. Estos protocolos proporcionaban control de errores (mediante confirmaciones de recepción) y control de flujo (usando una ventana corrediza).

En contraste, en este capítulo no hemos mencionado las comunicaciones confiables. Todo lo que ofrecen las Ethernet y los protocolos 802 es un servicio de datagramas de mejor esfuerzo. A veces, este servicio es adecuado. Por ejemplo, para transportar paquetes IP no se requieren ni se esperan garantías. Un paquete IP simplemente puede introducirse en un campo de carga 802 y enviarse a su destino; si se pierde, que así sea.

Sin embargo, también hay sistemas en los que se desea un protocolo de enlace de datos con control de errores y control de flujo. El IEEE ha definido uno que puede operar encima de todos los protocolos Ethernet y 802. Además, este protocolo, llamado **LLC (Control Lógico del Enlace)**, esconde las diferencias entre los distintos tipos de redes 802, proporcionando un formato único y una interfaz con la capa de red. Este formato, interfaz y protocolo están basados estrechamente en HDLC que estudiamos en el capítulo 3. El LLC forma la mitad superior de la capa de enlace de datos, con la subcapa de MAC por debajo de él, como se muestra en la figura 4-24.

El uso típico del LLC es el siguiente. La capa de red de la máquina emisora pasa un paquete al LLC usando las primitivas de acceso del LLC. A continuación, la subcapa LLC agrega un encabezado LLC que contiene los números de secuencia y confirmación de recepción. La estructura resultante se introduce entonces en el campo de carga útil de una trama 802 y se transmite. En el receptor ocurre el proceso inverso.

El LLC proporciona tres opciones de servicio: servicio no confiable de datagramas, servicio de datagramas sin confirmación de recepción y servicio confiable orientado a la conexión. El encabezado LLC contiene tres campos: un punto de acceso de destino, un punto de acceso de origen y un campo de control. Los puntos de acceso indican de cuál proceso proviene la trama y en dónde

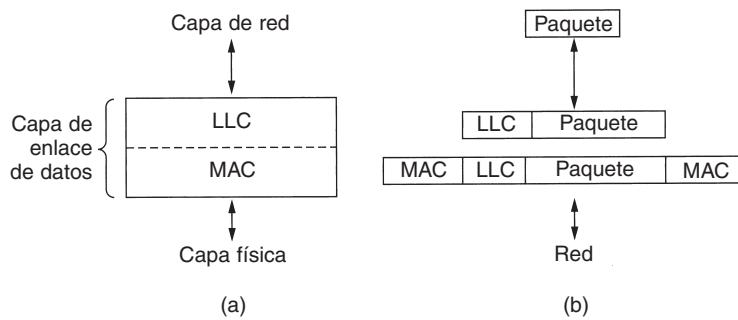


Figura 4-24. (a) Posición del LLC. (b) Formatos de protocolo.

se va a enviar, con lo que reemplazan el campo de *Tipo DIX*. El campo de control contiene números de secuencia y de confirmación de recepción, muy parecido a HDLC (vea la figura 3-24), pero no idéntico. Estos campos se utilizan principalmente cuando se necesita una conexión confiable en el nivel de enlace de datos, en cuyo caso se utilizarían protocolos similares a los que tratamos en el capítulo 3. Para Internet, los intentos de mejor esfuerzo para enviar los paquetes IP son suficientes, por lo que no se requieren confirmaciones de recepción en el nivel LLC.

4.3.10 Retrospectiva de Ethernet

Ethernet ha existido desde hace 20 años y no tiene competidores serios a la vista, por lo que es probable que exista por algunos años más. Pocas arquitecturas de CPU, sistemas operativos o lenguajes de programación han sido los reyes de la montaña por más de dos décadas. Claramente, Ethernet hizo algo bien, pero, ¿qué fue?

Probablemente la razón principal de su longevidad es que Ethernet es simple y flexible. En la práctica, simple se traduce como confiable, barato y fácil de mantener. Una vez que las derivaciones vampiro se reemplazaron con conectores BNC, las fallas eran menos frecuentes. Las personas dudaban en reemplazar algo que funcionaba bien todo el tiempo, especialmente porque sabían que muchas cosas funcionaban pobremente en la industria de la computación, por lo que muchas “actualizaciones” son peores que lo que reemplazan.

Simple también se traduce como barato. El cableado Ethernet delgado y el de par trenzado tienen un costo relativamente bajo. Las tarjetas de interfaz también tienen un costo bajo. Sólo cuando se introdujeron concentradores y conmutadores, se necesitaron inversiones considerables, pero para la época en que entraron en escena, Ethernet ya estaba bien establecida.

Ethernet es fácil de mantener. No hay software que instalar (sólo los controladores) y no hay tablas de configuración que manejar (con las cuales equivocarse). Además, agregar nuevos hosts es tan simple como conectarlos.

Otro punto es que Ethernet interactúa fácilmente con TCP/IP, el cual se ha vuelto dominante. IP es un protocolo sin conexión, porque se ajusta perfectamente con Ethernet, que tampoco es orientado a la conexión. IP no se ajusta tan bien con ATM, que es orientado a la conexión. Esta falta de ajuste afecta definitivamente las posibilidades de ATM.

Por último, Ethernet ha sido capaz de evolucionar en formas importantes. Las velocidades han aumentado en algunos niveles de magnitud y se han introducido los concentradores y commutadores, pero estos cambios no requieren modificaciones en el software. Un vendedor de redes está en un grave problema cuando muestra una instalación grande y dice: "Tengo esta nueva red fantástica para usted. Lo único que tiene que hacer es tirar todo su hardware y reescribir todo su software". Cuando se introdujeron la FDDI, el canal de fibra y ATM, eran más rápidos que Ethernet, pero también eran incompatibles con Ethernet, mucho más complejos y difíciles de manejar. Con el tiempo, Ethernet los igualó en cuanto a velocidad, por lo que ya no tenían ventajas y poco a poco dejaron de utilizarse, excepto ATM, el cual se utiliza en el núcleo del sistema telefónico.

4.4 LANS INALÁMBRICAS

Aunque Ethernet se utiliza ampliamente, está a punto de tener un competidor fuerte. Las LANs inalámbricas se están volviendo muy populares, y más y más edificios de oficinas, aeropuertos y otros lugares públicos se están equipando con ellas. Las LANs inalámbricas pueden funcionar en una de dos configuraciones, como vimos en la figura 1-35: con una estación base y sin ninguna estación base. En consecuencia, el estándar de LAN 802.11 toma en cuenta esto y se previene para ambos arreglos, como veremos más adelante.

En la sección 1.5.4 proporcionamos información sobre 802.11. Ahora es tiempo de ver más de cerca esta tecnología. En las siguientes secciones veremos la pila de protocolos, las técnicas de transmisión de radio de la capa física, el protocolo de la subcapa MAC, la estructura de trama y los servicios. Para mayor información sobre 802.11, vea (Crow y cols., 1997; Geier, 2002; Heegard y cols., 2001; Kapp, 2002; O'Hara y Petrick, 1999, y Severance, 1999). Para obtener información de una fuente fidedigna, consulte el estándar publicado 802.11.

4.4.1 La pila de protocolos del 802.11

Los protocolos utilizados por todas las variantes 802, entre ellas Ethernet, tienen ciertas similitudes de estructura. En la figura 4-25 se muestra una vista parcial de la pila de protocolos del estándar 802.11. La capa física corresponde muy bien con la capa física OSI, pero la capa de enlace de datos de todos los protocolos 802 se divide en dos o más subcapas. En el estándar 802.11, la subcapa MAC determina la forma en que se asigna el canal, es decir, a quién le toca transmitir a continuación. Arriba de dicha subcapa se encuentra la subcapa LLC, cuyo trabajo es ocultar las diferencias entre las variantes 802 con el propósito de que sean imperceptibles para la capa de red. Anteriormente en este capítulo analizamos el LLC, cuando examinamos Ethernet, por lo que no repetiremos ese material aquí.

El estándar 802.11 de 1997 especifica tres técnicas de transmisión permitidas en la capa física. El método de infrarrojos utiliza en su mayor parte la misma tecnología que los controles remotos

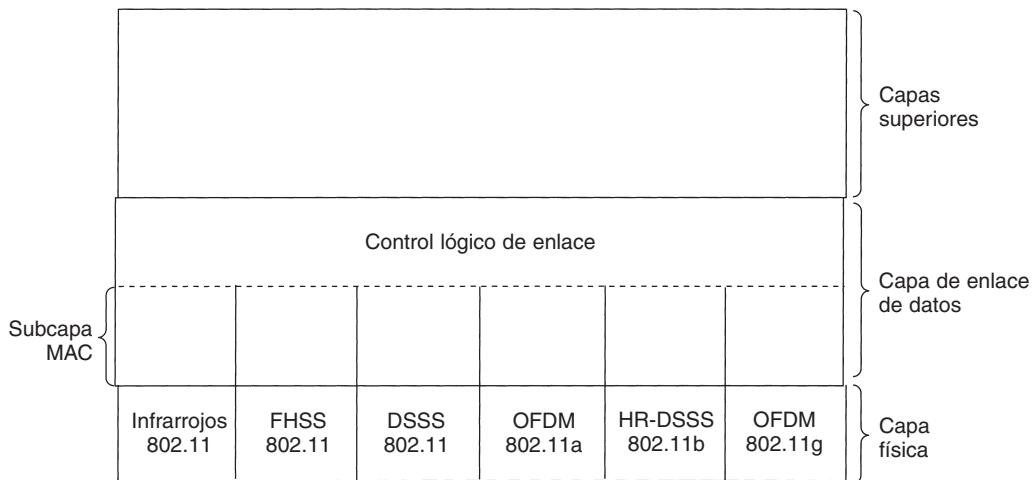


Figura 4-25. Parte de la pila de protocolos del 802.11.

de televisión. Los otros dos métodos utilizan el radio de corto alcance, mediante técnicas conocidas como FHSS y DSSS. Éstas utilizan parte del espectro que no necesita licencia (la banda ISM de 2.4 GHz). Los abridores de puertas de cocheras controlados por radio también utilizan esta parte del espectro, por lo que su computadora portátil podría encontrarse compitiendo con la puerta de la cochera. Los teléfonos inalámbricos y los hornos de microondas también utilizan esta banda. Todas estas técnicas funcionan a 1 o 2 Mbps y con poca energía por lo que no interfieren mucho entre sí. En 1999 se introdujeron dos nuevas técnicas para alcanzar un ancho de banda más alto. Éstas se conocen como OFDM y HRDSSS. Funcionan hasta 54 y 11 Mbps, respectivamente. En 2001 se introdujo una segunda modulación OFDM, pero en una banda de frecuencia diferente respecto a la primera. A continuación examinaremos con brevedad cada una de ellas. Técnicamente, pertenecen a la capa física y debieron examinarse en el capítulo 2, pero las trataremos aquí debido a que están estrechamente enlazadas a las LANs en general y a la subcapa MAC 802.11.

4.4.2 La capa física del 802.11

Cada una de las cinco técnicas permitidas de transmisión posibilitan el envío de una trama MAC de una estación a otra. Sin embargo, difieren en la tecnología utilizada y en las velocidades alcanzables. Un análisis detallado de estas tecnologías está más allá del alcance de este libro, pero es posible que algunas palabras sobre dichas tecnologías, junto con algunos términos clave, proporcionen a los lectores interesados algunos términos con los cuales buscar más información en Internet o en alguna otra parte.

La opción de infrarrojos utiliza transmisión difusa (es decir, no requiere línea visual) a 0.85 o 0.95 micras. Se permiten dos velocidades: 1 y 2 Mbps. A 1 Mbps se utiliza un esquema de codificación en el cual un grupo de 4 bits se codifica como una palabra codificada de 16 bits, que contiene quince 0s y un 1, mediante **código de Gray**. Este código tiene la propiedad de que un pequeño error en la sincronización en el tiempo lleva a un solo error de bits en la salida. A 2 Mbps, la codificación toma 2 bits y produce una palabra codificada de 4 bits, también con un solo 1, que es uno de 0001, 0010, 0100 o 1000. Las señales de infrarrojos no pueden penetrar las paredes, por lo que las celdas en los diferentes cuartos están bien aisladas entre sí. Sin embargo, debido al bajo ancho de banda (y al hecho de que la luz solar afecta las señales de infrarrojos), ésta no es una opción muy popular.

FHSS (Espectro Disperso con Salto de Frecuencia) utiliza 79 canales, cada uno de los cuales tiene un ancho de banda de 1 MHz, iniciando en el extremo más bajo de la banda ISM de 2.4 GHz. Para producir la secuencia de frecuencias a saltar, se utiliza un generador de números pseudoaleatorios. Siempre y cuando todas las estaciones utilicen la misma semilla para el generador de números pseudoaleatorios y permanezcan sincronizadas, saltarán de manera simultánea a la misma frecuencia. El tiempo invertido en cada frecuencia, el **tiempo de permanencia**, es un parámetro ajustable, pero debe ser menor que 400 mseg. La aleatorización de FHSS proporciona una forma justa de asignar espectro en la banda ISM no regulada. También proporciona algo de seguridad pues un intruso que no sepa la secuencia de saltos o el tiempo de permanencia no puede espiar las transmisiones. En distancias más grandes, el desvanecimiento de múltiples rutas puede ser un problema, y FHSS ofrece buena resistencia a ello. También es relativamente insensible a la interferencia de radio, lo que lo hace popular para enlaces de edificio en edificio. Su principal desventaja es su bajo ancho de banda.

El tercer método de modulación, **DSSS (Espectro Disperso de Secuencia Directa)**, también está restringido a 1 o 2 Mbps. El esquema utilizado tiene algunas similitudes con el sistema CDMA que examinamos en la sección 2.6.2, pero difiere en otros aspectos. Cada bit se transmite como 11 *chips*, utilizando lo que se conoce como **secuencia Barker**. Utiliza modulación por desplazamiento de fase a 1 Mbaudio, y transmite 1 bit por baudio cuando opera a 1 Mbps, y 2 bits por baudio cuando opera a 2 Mbps. Durante mucho tiempo, la FCC exigió que todo el equipo de comunicación inalámbrica que operaba en la banda ISM en Estados Unidos utilizaría el espectro disperso, pero en mayo de 2002 esa regla se eliminó conforme apareció nueva tecnología.

La primera de las LANs inalámbricas de alta velocidad, **802.11a**, utiliza **OFDM (Multiplexión por División de Frecuencias Ortogonales)** para enviar hasta 54 Mbps en la banda ISM más ancha de 5 GHz. Como lo sugiere el término FDM, se utilizan frecuencias diferentes —52 en total, 48 para datos y 4 para sincronización— al igual que ADSL. Debido a que las transmisiones están presentes en múltiples frecuencias al mismo tiempo, esta técnica se considera como una forma de espectro disperso, pero es diferente a CDMA y a FHSS. Dividir la señal en bandas más estrechas tiene más ventajas que el uso de una sola banda ancha, entre ellas mejor inmunidad a la interferencia de bandas estrechas y la posibilidad de utilizar bandas no contiguas. Se utiliza un sistema de codificación complejo, con base en la modulación por desplazamiento de fase para velocidades de hasta 18 Mbps, y en QAM para velocidades mayores. A 54 Mbps, se codifican 216 bits

de datos en símbolos de 288 bits. Parte del motivo para utilizar OFDM es la compatibilidad con el sistema europeo HiperLAN/2 (Doufexi y cols., 2002). La técnica tiene buena eficiencia de espectro en términos de bits/Hz y buena inmunidad al desvanecimiento de múltiples rutas.

A continuación analizaremos **HR-DSSS (Espectro Disperso de Secuencia Directa de Alta Velocidad)**, otra técnica de espectro disperso, que utiliza 11 millones de chips/seg para alcanzar 11 Mbps en la banda de 2.4 GHz. Se llama **802.11b** pero no es la continuación de 802.11a. De hecho, su estándar se aprobó primero y apareció primero en el mercado. Las tasas de datos soportadas por 802.11b son 1, 2, 5.5 y 11 Mbps. Las dos tasas bajas se ejecutan a 1 Mbaudio, con 1 y 2 bits por baudio, respectivamente, utilizando modulación por desplazamiento de fase (por compatibilidad con DSSS). Las dos tasas más rápidas se ejecutan a 1.375 Mbaudios, con 4 y 8 bits por baudio, respectivamente, utilizando códigos **Walsh/Hadamard**. La tasa de datos puede ser adaptada de manera dinámica durante la operación para alcanzar la velocidad más óptima posible bajo las condiciones actuales de la carga y el ruido. En la práctica, la velocidad de operación de 802.11b siempre es de aproximadamente 11 Mbps. Aunque 802.11b es más lento que 802.11a, su rango es aproximadamente 7 veces mayor, lo que es más importante en muchas situaciones.

En noviembre de 2001, el IEEE aprobó una versión mejorada de 802.11b, **802.11g**, después de mucho politiquero por cuál tecnología patentada podría utilizar. Utiliza el método de modulación OFDM de 802.11a pero opera en la banda ISM más estrecha 2.4 GHz ISM junto con 802.11b. En teoría, puede operar hasta a 54 Mbps. Aún no se ha decidido si esta velocidad se va a alcanzar en la práctica. Lo que esto significa es que el comité 802.11 ha producido tres LANs inalámbricas diferentes de alta velocidad: 802.11a, 802.11b y 802.11g (sin mencionar las tres LANs inalámbricas de baja velocidad). Uno se puede preguntar si es bueno que el comité de estándares haga esto. Tal vez el tres sea su número de suerte.

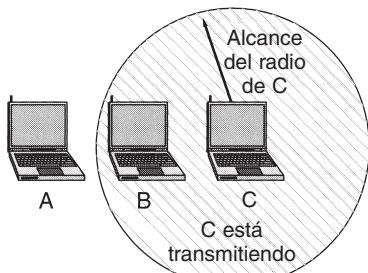
4.4.3 El protocolo de la subcapa MAC del 802.11

Regresemos ahora de la tierra de la ingeniería eléctrica a la de las ciencias de la computación. El protocolo de la subcapa MAC para el estándar 802.11 es muy diferente del de Ethernet debido a la complejidad inherente del entorno inalámbrico en comparación con el de un sistema cableado. Con Ethernet, una estación simplemente espera hasta que el medio queda en silencio y comienza a transmitir. Si no recibe una ráfaga de ruido dentro de los primeros 64 bytes, con seguridad la trama ha sido entregada correctamente. Esta situación no es válida para los sistemas inalámbricos.

Para empezar, existe el problema de la estación oculta mencionado con anterioridad, el cual se ilustra nuevamente en la figura 4-26(a). Puesto que no todas las estaciones están dentro del alcance de radio de cada una, las transmisiones que van en un lado de una celda podrían no recibirse en otro lado de la misma celda. En este ejemplo, la estación *C* transmite a la estación *B*. Si *A* detecta el canal, no escuchará nada y concluirá erróneamente que ahora puede comenzar a transmitir a *B*.

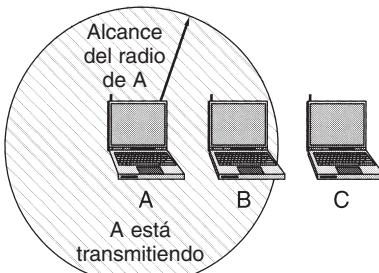
Además, existe el problema inverso, el de la estación expuesta, que se ilustra en la figura 4-26(b). Aquí *B* desea enviar a *C* por lo que escucha el canal. Cuando escucha una transmisión, concluye erróneamente que no debería transmitir a *C*, aunque *A* esté transmitiendo a *D* (lo cual no

A desea enviar a B pero
no puede oír que
B está ocupado



(a)

B desea enviar a C pero
piensa erróneamente que
la transmisión fallará



(b)

Figura 4-26. (a) El problema de la estación oculta. (b) El problema expuesta.

se muestra). Además, la mayoría de los radios son semidúplex, lo que significa que no pueden transmitir y escuchar ráfagas de ruido al mismo tiempo en una sola frecuencia. Como resultado de estos problemas, 802.11 no utiliza CSMA/CD, como lo hace Ethernet.

Para solucionar este problema, 802.11 soporta dos modos de funcionamiento. El primero, llamado **DCF (Función de Coordinación Distribuida)**, no utiliza ningún tipo de control central (en ese aspecto, es similar a Ethernet). El otro, llamado **PCF (Función de Coordinación Puntual)**, utiliza la estación base para controlar toda la actividad en su celda. Todas las implementaciones soportan DCF pero PCF es opcional. A continuación analizaremos estos dos modos a la vez.

Cuando se emplea DCF, 802.11 utiliza un protocolo llamado **CSMA/CA (CSMA con Evitación de Colisiones)**. En este protocolo, se utiliza tanto la detección del canal físico como la del canal virtual. Los dos métodos de funcionamiento son soportados por CSMA/CA. En el primer método, cuando una estación desea transmitir, detecta el canal. Si está inactivo, comienza a transmitir. No detecta el canal mientras transmite pero emite su trama completa, la cual podría ser destruida en el receptor debido a interferencia. Si el canal está ocupado, el emisor espera hasta que esté inactivo para comenzar a transmitir. Si ocurre una colisión, las estaciones involucradas en ella esperan un tiempo aleatorio, mediante el algoritmo de retroceso exponencial binario de Ethernet, y vuelve a intentarlo más tarde.

El otro modo de la operación CSMA/CA se basa en MACAW y utiliza la detección de canal virtual, como se ilustra en la figura 4-27. En este ejemplo, *A* desea enviar a *B*. *C* es una estación que está dentro del alcance de *A* (y posiblemente dentro del alcance de *B*, pero eso no importa). *D* es una estación dentro del alcance de *B* pero no dentro del de *A*.

El protocolo inicia cuando *A* decide enviar datos a *B*. *A* inicia enviándole una trama RTS a *B* en la que le solicita permiso para enviarle una trama. Cuando *B* recibe esta solicitud, podría decidir otorgarle el permiso, en cuyo caso le regresa una trama CTS. Al recibir la CTS, *A* ahora envía su

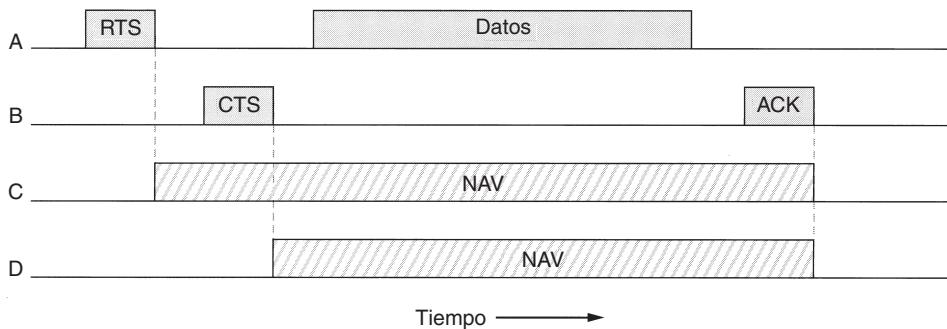


Figura 4-27. El uso de la detección de canal virtual utilizando CSMA/CA.

trama y comienza su temporizador de ACK. Al recibir correctamente la trama de datos, *B* responde con una trama de ACK, con lo que termina el intercambio. Si el temporizador de ACK de *A* termina antes de que el ACK regrese, todo el protocolo se ejecuta de nuevo.

Ahora consideremos este intercambio desde el punto de vista de *C* y *D*. *C* está dentro del alcance de *A*, por lo que podría recibir la trama RTS. Si pasa esto, se da cuenta de que alguien va a enviar datos pronto, así que por el bien de todos desiste de transmitir cualquier cosa hasta que el intercambio esté completo. A partir de la información proporcionada en la solicitud RTS, *C* puede estimar cuánto tardará la secuencia, incluyendo el ACK final, por lo que impone para sí misma un tipo de canal virtual ocupado, indicado por **NAV (Vector de Asignación de Red)** en la figura 4-27. *D* no escucha el RTS, pero sí el CTS, por lo que también impone la señal *NAV* para sí misma. Observe que las señales *NAV* no se transmiten; simplemente son recordatorios internos para mantenerse en silencio durante cierto periodo.

En contraste con las redes cableadas, las inalámbricas son ruidosas e inestables, en gran parte debido a los hornos de microondas, que también utilizan las bandas sin licencia ISM. Como consecuencia, la probabilidad de que una trama llegue a su destino se decremente con la longitud de la trama. Si la probabilidad de que cualquier bit sea erróneo es p , entonces la probabilidad de que una trama de n bits se reciba por completo y correctamente es $(1 - p)^n$. Por ejemplo, para $p = 10^{-4}$, la probabilidad de recibir correctamente una trama Ethernet completa (12,144 bits) es menor que 30%. Si $p = 10^{-5}$, aproximadamente una trama de 9 estará dañada. Incluso si $p = 10^{-6}$, más de 1% de las tramas se dañará, lo que equivale a casi una docena por segundo, y más si se utilizan tramas más cortas que el máximo. En resumen, si una trama es demasiado grande, tiene muy pocas probabilidades de pasar sin daño y probablemente tenga que retransmitirse.

Para solucionar el problema de los canales ruidosos, 802.11 permite dividir las tramas en fragmentos, cada uno con su propia suma de verificación. Cada fragmento se numera de manera individual y su recepción se confirma utilizando un protocolo de parada y espera (es decir, el emisor podría no transmitir fragmentos de $k + 1$ hasta que haya recibido la confirmación de recepción del fragmento k). Una vez que se ha adquirido el canal mediante RTS y CTS, pueden enviarse múltiples

fragmentos en una fila, como se muestra en la figura 4-28. La secuencia de fragmentos se conoce como **ráfaga de fragmentos**.

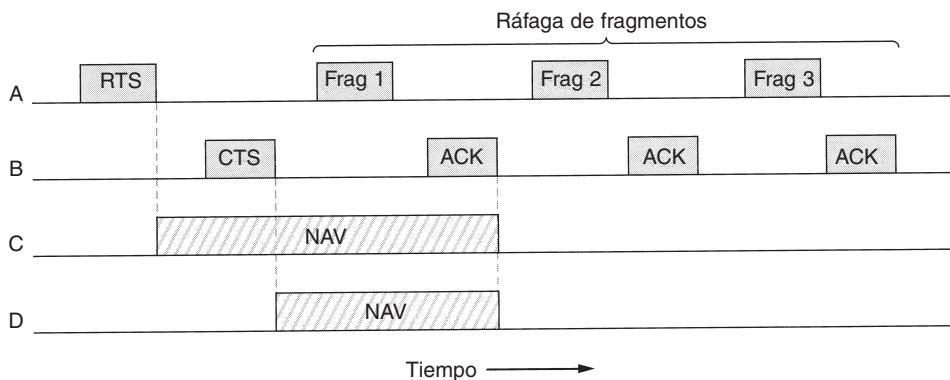


Figura 4-28. Una ráfaga de fragmentos.

La fragmentación incrementa la velocidad real de transporte restringiendo las retransmisiones a los fragmentos erróneos en lugar de la trama completa. El tamaño del fragmento no lo fija el estándar pero es un parámetro de cada celda y la estación base puede ajustarlo. El mecanismo NAV mantiene otras estaciones en silencio sólo hasta la siguiente confirmación de recepción, pero se utiliza otro mecanismo (descrito a continuación) para permitir que otra ráfaga de fragmentos completa se envíe sin interferencia.

Todo el análisis anterior se aplica al modo DCF 802.11. En él, no hay control central y la estación compite por tiempo aire, como en Ethernet. El otro modo permitido es PCF, en el que la estación base sondea las demás estaciones, preguntándoles si tienen tramas que enviar. Puesto que el orden de transmisión se controla por completo por la estación base en el modo PCF, no ocurren colisiones. El estándar prescribe el mecanismo para sondeo, pero no la frecuencia del sondeo, el orden del sondeo, ni el hecho de que las demás estaciones necesiten obtener un servicio igual.

El mecanismo básico consiste en que la estación base difunda una **trama de beacon** (trama guía o faro) de manera periódica (de 10 a 100 veces por segundo). Esta trama contiene parámetros de sistema, como secuencias de salto y tiempos de permanencia (para FHSS), sincronización de reloj, etcétera. También invita a las nuevas estaciones a suscribirse al servicio de sondeo. Una vez que una estación se inscribe para el servicio de sondeo a cierta tasa, se le garantiza de manera efectiva cierta fracción de ancho de banda, y se hace posible proporcionar garantías de calidad de servicio.

La duración de la batería siempre es un problema en los dispositivos inalámbricos móviles, por lo que 802.11 pone atención al asunto de la administración de energía. En particular, una estación base puede conducir una estación móvil al estado de hibernación hasta que dicha estación base o el usuario la saquen de él de manera explícita. Sin embargo, el hecho de indicar a una estación que entre en estado de hibernación significa que la estación base tiene la responsabilidad de almacenar en el búfer las tramas que vayan dirigidas a ella mientras la estación móvil esté hibernando. Posteriormente, esas tramas pueden colectarse.

PCF y DCF pueden coexistir dentro de una celda. Al principio podría parecer imposible tener control central y distribuido funcionando al mismo tiempo, pero 802.11 proporciona una forma de

alcanzar este objetivo. Funciona definiendo cuidadosamente el intervalo de tiempo entre tramas. Después de que se ha enviado una trama, se necesita cierta cantidad de tiempo muerto antes de que cualquier estación pueda enviar una trama. Se definen cuatro intervalos diferentes, cada uno con un propósito específico. Estos intervalos se describen en la figura 4-29.

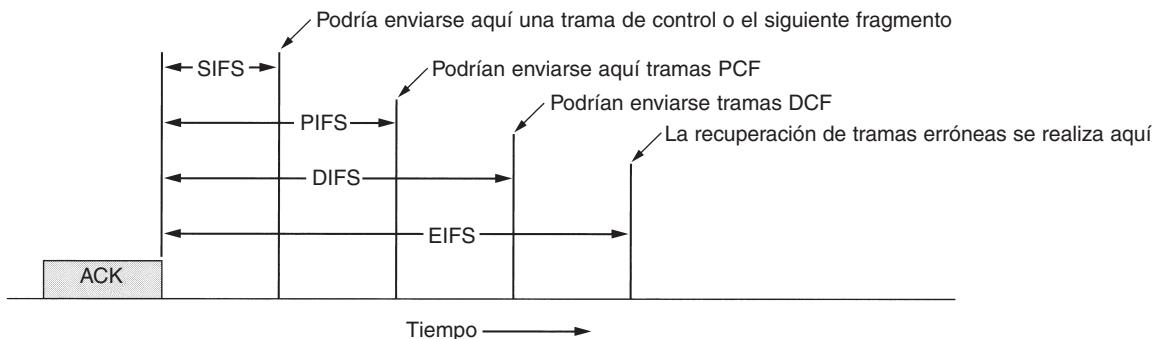


Figura 4-29. Espaciado entre tramas 802.11.

El intervalo más corto es **SIFS (Espaciado Corto Entre Tramas)**. Se utiliza para permitir que las distintas partes de un diálogo transmitan primero. Esto incluye dejar que el receptor envíe un CTS para responder a una RTS, dejar que el receptor envíe un ACK para un fragmento o una trama con todos los datos y dejar que el emisor de una ráfaga de fragmentos transmita el siguiente fragmento sin tener que enviar una RTS nuevamente.

Siempre hay una sola estación que debe responder después de un intervalo SIFS. Si falla al utilizar su oportunidad y transcurre un tiempo **PIFS (Espaciado Entre Tramas PCF)**, la estación base podría enviar una trama de *beacon* o una trama de sondeo. Este mecanismo permite que una estación base envíe una trama de datos o una secuencia de fragmentos para finalizar su trama sin que nadie interfiera, pero le da a la estación base la oportunidad de tomar el canal cuando el emisor anterior haya terminado, sin tener que competir con usuarios ansiosos.

Si la estación base no tiene nada que decir y transcurre un tiempo **DIFS (Espaciado Entre Tramas DCF)**, cualquier estación podría intentar adquirir el canal para enviar una nueva trama. Se aplican las reglas de contención normales, y si ocurre una colisión, podría necesitarse el retroceso exponencial binario.

Sólo una estación que acaba de recibir una trama errónea o desconocida utiliza el último intervalo de tiempo, **EIFS (Espaciado Entre Tramas Extendido)**, para reportar la trama errónea. La idea de dar a este evento la menor prioridad es que debido a que el receptor tal vez no tenga idea de lo que está pasando, debe esperar un tiempo considerable para evitar interferir con un diálogo en curso entre las dos estaciones.

4.4.4 La estructura de trama 802.11

El estándar 802.11 define tres clases diferentes de tramas en el cable: de datos, de control y de administración. Cada una de ellas tiene un encabezado con una variedad de campos utilizados

dentro de la subcapa MAC. Además, hay algunos encabezados utilizados por la capa física, pero éstos tienen que ver en su mayor parte con las técnicas de modulación utilizadas, por lo que no las trataremos aquí.

En la figura 4-30 se muestra el formato de la trama de datos. Primero está el campo de *Control de trama*. Éste tiene 11 subcampos. El primero es la *Versión de protocolo*, que permite que dos versiones del protocolo funcionen al mismo tiempo en la misma celda. Después están los campos de *Tipo* (de datos, de control o de administración) y de *Subtipo* (por ejemplo, RTS o CTS). Los bits *A DS* y *De DS* indican que la trama va hacia o viene del sistema de distribución entre celdas (por ejemplo, Ethernet). El bit *MF* indica que siguen más fragmentos. El bit *Retrans.* marca una retransmisión de una trama que se envió anteriormente. El bit de *Administración de energía* es utilizado por la estación base para poner al receptor en estado de hibernación o sacarlo de tal estado. El bit *Más* indica que el emisor tiene tramas adicionales para el receptor. El bit *W* especifica que el cuerpo de la trama se ha codificado utilizando el algoritmo **WEP (Privacidad Inalámbrica Equivalente)**. Por último, el bit *O* indica al receptor que una secuencia de tramas que tenga este bit encendido debe procesarse en orden estricto.

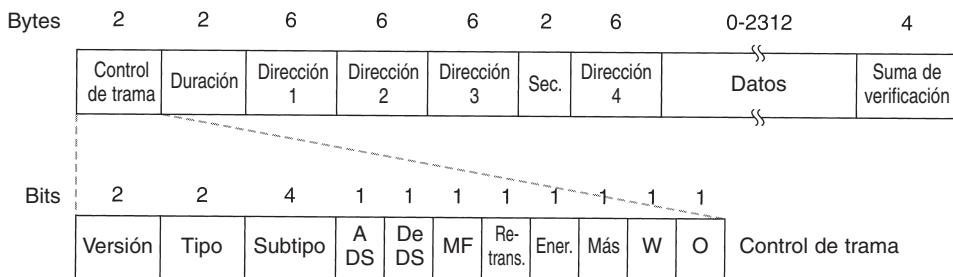


Figura 4-30. La trama de datos 802.11.

El segundo campo de la trama de datos, el de *Duración*, indica cuánto tiempo ocuparán el canal la trama y su confirmación de recepción. Este campo también está presente en las tramas de control y es la forma mediante la cual otras estaciones manejan el mecanismo NAV. El encabezado de trama contiene cuatro direcciones, todas en formato estándar IEEE 802. Obviamente se necesitan el origen y el destino, pero, ¿para qué son las otras dos? Recuerde que las tramas podrían entrar o dejar una celda a través de una estación base. Las otras dos direcciones se utilizan para las estaciones base de origen y destino para el tráfico entre celdas.

El campo de *Secuencia* permite que se numeren los fragmentos. De los 16 bits disponibles, 12 identifican la trama y 4 el fragmento. El campo de *Datos* contiene la carga útil, hasta 2312 bytes, y le sigue el campo común de *Suma de verificación*.

Las tramas de administración tienen un formato similar al de las tramas de datos, excepto que no tienen una de las direcciones de la estación base, debido a que las tramas de administración se restringen a una sola celda. Las tramas de control son más cortas; tienen una o dos direcciones, y no tienen ni campo de *Datos* ni de *Secuencia*. La información clave aquí se encuentra en el campo de *Subtipo*, que por lo general es RTS, CTS o ACK.

4.4.5 Servicios

El estándar 802.11 afirma que cada LAN inalámbrica que se apegue a él debe proporcionar nueve servicios. Éstos se dividen en dos categorías: cinco servicios de distribución y cuatro de estación. Los servicios de distribución se relacionan con la administración de membresías dentro de la celda y con la interacción con estaciones que están fuera de la celda. En contraste, los servicios de estación se relacionan con la actividad dentro de una sola celda.

Los cinco servicios de distribución son proporcionados por las estaciones base y tienen que ver con la movilidad de la estación conforme entran y salen de las celdas, conectándose ellos mismos a las estaciones base y separándose ellos mismos de dichas estaciones. Estos servicios son los siguientes:

1. **Asociación.** Este servicio es utilizado por las estaciones móviles para conectarse ellas mismas a las estaciones base. Por lo general, se utiliza después de que una estación se mueve dentro del alcance de radio de la estación base. Una vez que llega, anuncia su identidad y sus capacidades. Éstas incluyen las tasas de datos soportadas, necesarias para los servicios PCF (es decir, el sondeo), y los requerimientos de administración de energía. La estación base podría aceptar o rechazar la estación móvil. Si se acepta, dicha estación debe autenticarse.
2. **Disociación.** Es posible que la estación o la estación base se disocie, con lo que se rompería la relación. Una estación podría utilizar este servicio antes de apagarse o de salir, pero la estación base también podría utilizarlo antes de su mantenimiento.
3. **Reasociación.** Una estación podría cambiar su estación base preferida mediante este servicio. Esta capacidad es útil para estaciones móviles que se mueven de una celda a otra. Si se utiliza correctamente, no se perderán datos como consecuencia del cambio de estación base (*handover*). (Pero 802.11, al igual que Ethernet, es sólo un servicio de mejor esfuerzo.)
4. **Distribución.** Este servicio determina cómo enrutar tramas enviadas a la estación base. Si el destino es local para la estación base, las tramas pueden enviarse directamente a través del aire. De lo contrario, tendrán que reenviarse a través de la red cableada.
5. **Integración.** Si una trama necesita enviarse a través de una red no 802.11 con un esquema de direccionamiento o formato de trama diferentes, este servicio maneja la traducción del formato 802.11 al requerido por la red de destino.

Los cuatro servicios restantes son dentro de las celdas (es decir, se relacionan con acciones dentro de una sola celda). Se utilizan después de que ha ocurrido la asociación y son las siguientes:

1. **Autenticación.** Debido a que las estaciones no autorizadas pueden recibir o enviar con facilidad la comunicación inalámbrica, una estación debe autenticarse antes de que se le permita enviar datos. Una vez que la estación base asocia una estación móvil (es decir, la ha aceptado en su celda), le envía una trama especial de desafío para ver si dicha estación móvil sabe la clave secreta (contraseña) que se le ha asignado. La estación móvil prueba que sabe la clave secreta codificando la trama de desafío y regresándola a la estación base. Si el resultado es correcto, la estación móvil se vuelve miembro de la celda. En el estándar inicial, la estación base no tiene que probar su identidad a la estación móvil, pero se está realizando trabajo para reparar este defecto en el estándar.
2. **Desautenticación.** Cuando una estación previamente autenticada desea abandonar la red, se desautentica. Después de esto, tal vez ya no utilice la red.
3. **Privacidad.** Para que la información que se envía a través de una LAN inalámbrica se mantenga confidencial, debe codificarse. Este servicio maneja la codificación y la decodificación. El algoritmo de codificación especificado es RC4, inventado por Ronald Rivest del M.I.T.
4. **Entrega de datos.** Por último, la transmisión de datos es la parte esencial, por lo que el 802.11 naturalmente proporciona una forma de transmitir y recibir datos. Puesto que el 802.11 está basado en Ethernet y no se garantiza que la transmisión a través de Ethernet sea 100% confiable, tampoco se garantiza que la transmisión a través del 802.11 sea confiable. Las capas superiores deben tratar con la detección y la corrección de errores.

Una celda 802.11 tiene algunos parámetros que pueden inspeccionarse y, en algunos casos, ajustarse. Se relacionan con la codificación, intervalos de expiración de temporizador, tasas de datos, frecuencia de la trama de *beacon*, etcétera.

Las LANs inalámbricas basadas en 802.11 se están comenzando a distribuir en edificios de oficinas, aeropuertos, hoteles, restaurantes y universidades de todo el mundo. Se espera un crecimiento rápido. Para obtener información adicional acerca de la distribución extendida de 802.11 en CMU, vea (Hills, 2001).

4.5 BANDA ANCHA INALÁMBRICA

Hemos estado en casa mucho tiempo. Salgamos y veamos si hay algo interesante sobre redes por ahí. Pues sucede que hay bastantes novedades, y algunas de ellas tienen que ver con una característica llamada última milla. Con la desregulación del sistema telefónico en muchos países, en la actualidad a los competidores de la compañía telefónica arraigada con frecuencia se les permite ofrecer voz local y servicio de alta velocidad de Internet. Ciertamente hay mucha demanda. El problema es que el tendido de fibra óptica, cable coaxial o incluso cable de par

trenzado categoría 5 a millones de casas y oficinas es extremadamente costoso. ¿Qué debe hacer un competidor?

La respuesta es la banda ancha inalámbrica. Construir una antena enorme en una colina en las afueras del pueblo e instalar antenas que se dirijan a dicha antena en los techos de los clientes es más fácil y barato que cavar zanjas y ensartar cables. Por lo tanto, las compañías de telecomunicación en competencia tienen mucho interés en proporcionar un servicio de comunicación inalámbrica de multimegabits para voz, Internet, películas bajo demanda, etcétera. Como vimos en la figura 2-30, los LMDS se inventaron para este propósito. Sin embargo, hasta hace poco, cada portadora diseñaba su propio sistema. Esta falta de estándares significaba que el hardware y software no se podía producir en masa, por lo que los precios eran altos y la aceptación, baja.

Muchas personas en la industria se dieron cuenta de que tener un estándar de banda ancha inalámbrica era el elemento clave que faltaba, por lo que se le pidió a IEEE que formara un comité compuesto de personas de compañías clave y de academias para redactar el estándar. El siguiente número disponible en el espacio de numeración 802 era **802.16**, por lo que el estándar obtuvo este número. El trabajo se inició en julio de 1999, y el estándar final se aprobó en abril de 2002. Oficialmente el estándar se llama “Air Interface for Fixed Broadband Wireless Access Systems” (Interfaz de Aire para Sistemas Fijos de Acceso Inalámbrico de Banda Ancha). Sin embargo, algunas personas prefieren llamarlo **MAN (red de área metropolitana) inalámbrica o circuito local inalámbrico**. Nos referiremos a estos términos de manera indistinta.

Al igual que los otros estándares 802, el 802.16 estuvo influido fuertemente por el modelo OSI, incluyendo las (sub)capas, terminología, primitivas de servicios y más. Desgraciadamente, al igual que OSI, es muy complicado. En las siguientes secciones daremos una breve descripción de algunos de los puntos de importancia del estándar 802.16, pero este tratado no es completo y no trata muchos detalles. Para mayor información acerca de la banda ancha inalámbrica, vea (Bolcskei y cols., 2001, y Webb, 2001). Para mayor información sobre el estándar 802.16 en particular, vea (Eklund y cols., 2002).

4.5.1 Comparación entre los estándares 802.11 y 802.16

En este punto tal vez piense: ¿Por qué diseñar un nuevo estándar? ¿Por qué no simplemente utilizar 802.11? Hay algunas buenas razones para no utilizar 802.11, principalmente porque 802.11 y 802.16 resuelven diferentes problemas. Antes de introducirnos en la tecnología de 802.16, probablemente valga la pena dar algunos detalles de por qué es necesario un estándar completamente nuevo.

Los entornos en los que funcionan 802.11 y 802.16 son similares en algunas formas, principalmente en que fueron diseñados para proporcionar comunicaciones inalámbricas de alto ancho de banda. Pero también difieren en aspectos muy importantes. Para empezar, el protocolo 802.16 proporciona servicio a edificios, y éstos no son móviles. No migran de celda a celda con frecuencia. La mayor parte de 802.11 tiene que ver con la movilidad, y nada de eso es relevante aquí. Además, los edificios pueden tener más de una computadora en ellos, lo cual no ocurre cuando la estación final es una sola computadora portátil.

Debido a que los dueños de edificios por lo general están dispuestos a gastar mucho más dinero en artículos de comunicación que los dueños de computadoras portátiles, hay mejores radios disponibles. Esta diferencia significa que 802.16 puede utilizar comunicación de dúplex total, algo que 802.11 evita para mantener bajo el costo de los radios.

Puesto que el estándar 802.16 se usa en parte de la ciudad, las distancias involucradas pueden ser de varios kilómetros, lo que significa que la energía detectada en la estación base puede variar considerablemente de estación en estación. Esta variación afecta la relación señal a ruido, que, a su vez, fija múltiples esquemas de modulación. Además, la comunicación abierta a través de la ciudad significa que la seguridad y privacidad son esenciales y obligatorias.

Además, es probable que cada celda tenga muchos más usuarios que una celda 802.11 típica, y se espera que estos usuarios utilicen más ancho de banda que un usuario 802.11 típico. Después de todo es raro que una compañía reúna a 50 empleados con sus computadoras portátiles en un cuarto para ver si pueden saturar la red inalámbrica 802.11 al observar a la vez 50 películas por separado. Por esta razón es necesario más espectro del que las bandas ISM pueden proporcionar, con lo que se obliga al estándar 802.16 a funcionar en el rango de frecuencia más alto de 10 a 66 GHz, el único lugar en el que el espectro no utilizado aún está disponible.

Pero estas ondas milimétricas tienen propiedades físicas diferentes que las ondas más largas en las bandas ISM, que a su vez requieren una capa física completamente diferente. Una propiedad de las ondas milimétricas es que el agua (especialmente la lluvia y, en cierta medida, la nieve, el granizo y, con un poco de mala suerte, la niebla espesa) las absorbe por completo. En consecuencia, el control de errores es más importante que en un entorno interno. Las ondas milimétricas pueden enfocarse en rayos direccionales (802.11 es omnidireccional), por lo que las opciones realizadas en 802.11 relacionadas con la propagación de múltiples rutas son debatibles.

Otro aspecto es la calidad del servicio. Si bien el estándar 802.11 proporciona soporte para el tráfico en tiempo real (utilizando el modo PCF), realmente no se diseñó para uso extenso de telefonía y multimedia. En contraste, se espera que el estándar 802.16 soporte estas aplicaciones por completo porque está diseñado para uso residencial y de negocios.

En resumen, 802.11 se diseñó para ser una Ethernet móvil, mientras que el estándar 802.16 se diseñó para ser televisión por cable inalámbrica, pero estacionaria. Estas diferencias son tan grandes que los estándares resultantes son muy diferentes debido a que tratan de optimizar cosas distintas.

Vale la pena hacer una pequeña comparación con el sistema de teléfonos celulares. Al referirnos a los teléfonos celulares, hablamos de estaciones móviles de banda estrecha, baja potencia y con orientación a voz que se comunican mediante microondas de longitud media. Nadie ve (todavía) películas de 2 horas a alta resolución en teléfonos móviles GSM. Incluso UMTS tiene poca esperanza de cambiar esta situación. En resumen, el mundo de las MANs inalámbricas es más demandante que el de los teléfonos móviles, por lo que se necesita un sistema completamente diferente. El hecho de que en el futuro el estándar 802.16 pueda utilizarse para dispositivos móviles es una pregunta interesante. No fue optimizado para ellos, pero la posibilidad está abierta. Por el momento se enfoca en los sistemas inalámbricos fijos.

4.5.2 La pila de protocolos del estándar 802.16

En la figura 4-31 se ilustra la pila de protocolos del estándar 802.16. La estructura general es similar a la de las otras redes 802, pero con más subcapas. La subcapa inferior tiene que ver con la transmisión. El radio de banda estrecha tradicional se utiliza con esquemas de modulación tradicionales. Arriba de la capa de transmisión física está una subcapa de convergencia para ocultarle las diferentes tecnologías a la capa de enlace de datos. En la actualidad el estándar 802.11 también tiene algo parecido a esto, sólo que el comité eligió no formalizarlo con un nombre de tipo OSI.

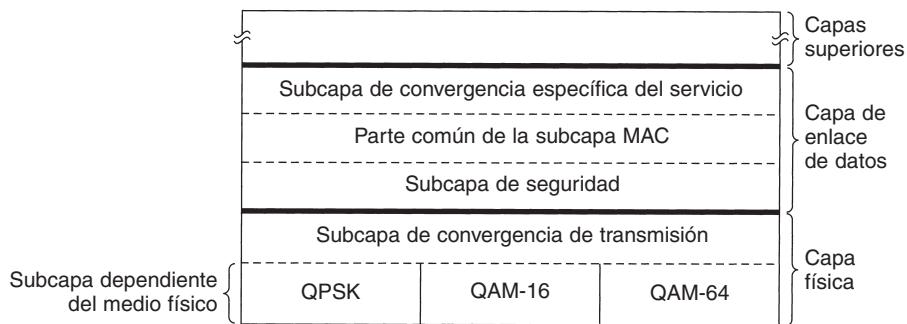


Figura 4-31. La pila de protocolos del 802.16.

Aunque no los mostramos en la figura, se está trabajando para agregar dos nuevos protocolos de capa física. El estándar 802.16a soportará a OFDM en el rango de frecuencia de 2 a 11 GHz. El estándar 802.16b operará en la banda ISM de 5 GHz. Estos dos son intentos para acercarse al estándar 802.11.

La capa de enlace de datos consta de tres subcapas. La inferior tiene que ver con la privacidad y seguridad, lo cual es más importante para las redes externas públicas que para las redes internas privadas. Maneja codificación, decodificación y administración de claves.

A continuación se encuentra la parte común de la subcapa MAC. Es aquí donde se encuentran los principales protocolos, como la administración de canal. El modelo consiste en que la estación base controla el sistema. Puede calendarizar de manera muy eficiente los canales de flujo descendente (es decir, de la estación base al suscriptor) y es muy importante en el manejo de los canales ascendentes (es decir, del suscriptor a la estación base). Una característica no muy común de la subcapa MAC es que, a diferencia de las subcapas de las otras redes 802, es completamente orientada a la conexión, para proporcionar garantías de calidad del servicio para la comunicación de telefonía y multimedia.

En los otros protocolos 802, la subcapa de convergencia específica del servicio toma el lugar de la subcapa de enlace lógico. Su función es interactuar con la capa de red. Un problema aquí es que el estándar 802.16 fue diseñado para integrarse sin ningún problema tanto con los protocolos

de datagramas (por ejemplo, PPP, IP y Ethernet) como con ATM. El problema es que los protocolos de paquetes no son orientados a la conexión y ATM sí lo es. Esto significa que cada conexión ATM se tiene que asignar a una conexión 802.16, que en principio es un asunto directo. ¿Pero en cuál conexión 802.16 debe asignarse un paquete IP entrante? El problema se soluciona en esta subcapa.

4.5.3 La capa física del estándar 802.16

Como se mencionó anteriormente, la banda ancha inalámbrica necesita mucho espectro y el único lugar para encontrarlo es en el rango de 10 a 66 GHz. Estas ondas milimétricas tienen una propiedad interesante que las microondas más largas no tienen: viajan en líneas rectas, a diferencia del sonido, pero en forma similar a la luz. Como consecuencia, la estación base puede tener múltiples antenas, cada una apuntando a un sector diferente del terreno circundante, como se muestra en la figura 4-32. Cada sector tiene sus propios usuarios y es completamente independiente de los sectores adyacentes, algo que no es verdad es el radio celular, el cual es omnidireccional.

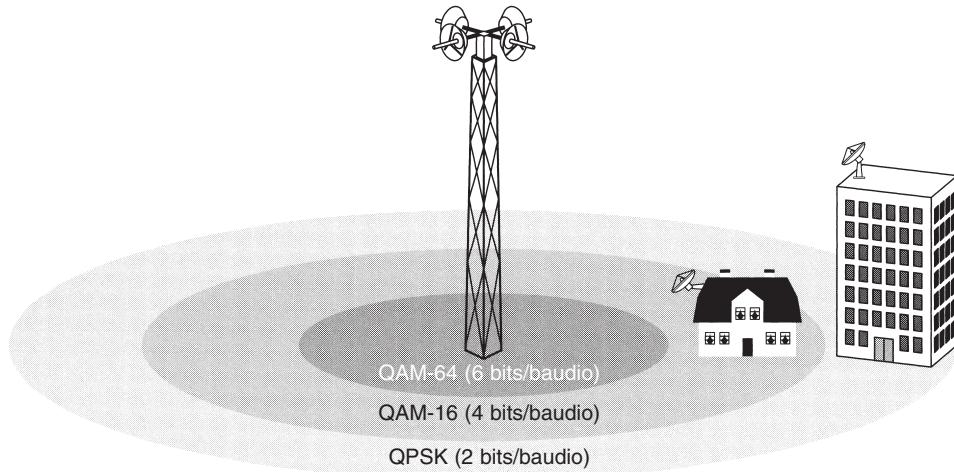


Figura 4-32. El entorno de transmisión 802.16.

Debido a que la fuerza de señal en la banda milimétrica desciende drásticamente con la distancia a partir de la estación base, la relación señal a ruido también desciende con la distancia a partir de la estación base. Por esta razón, el estándar 802.16 emplea tres esquemas de modulación diferentes, dependiendo de la distancia entre la estación suscriptora y la estación base. Para suscriptores cercanos se utiliza QAM-64, con 6 bits/baudio. Para suscriptores a distancias medias se utiliza QAM-16, con 4 bits/baudio. Para suscriptores distantes se utiliza QPSK, con 2 bits/baudio. Por ejemplo, para un valor típico de 25 MHz digno de espectro, QAM-64 da 150 Mbps, QAM-16 da 100 Mbps y QPSK da 50 Mbps. En otras palabras, entre más lejos esté el suscriptor de la estación base, será más baja la tasa de datos (algo similar a lo que vimos con ADSL en la figura 2-27). El diagrama de constelación de estas tres técnicas de modulación se mostró en la figura 2-25.

Dado el objetivo de producir un sistema de banda ancha, sujeto a las limitantes físicas mostradas anteriormente, los diseñadores del protocolo 802.16 trabajaron duro para utilizar eficientemente el espectro disponible. Los que no les gustaba era la forma en que funcionaban GSM y DAMPS. Ambos utilizan bandas de frecuencia diferentes pero iguales para el tráfico ascendente y descendente. Para voz, es probable que el tráfico sea simétrico en su mayor parte, pero para acceso a Internet por lo general hay más tráfico descendente que ascendente. En consecuencia, el estándar 802.16 proporciona una forma más flexible para asignar el ancho de banda. Se utilizan dos esquemas: **FDD (Duplexación por División de Frecuencia)** y **TDD (Duplexación por División de Tiempo)**. Este último se ilustra en la figura 4-33. Aquí la estación base envía tramas periódicamente. Cada trama contiene ranuras de tiempo. Las primeras son para el tráfico descendente. Después se encuentra el tiempo de protección o guarda, el cual es utilizado por las estaciones para cambiar de dirección. Por último, están las ranuras para el tráfico ascendente. El número de ranuras de tiempo dedicadas para cada dirección se puede cambiar de manera dinámica con el fin de que el ancho de banda en cada dirección coincida con el tráfico.

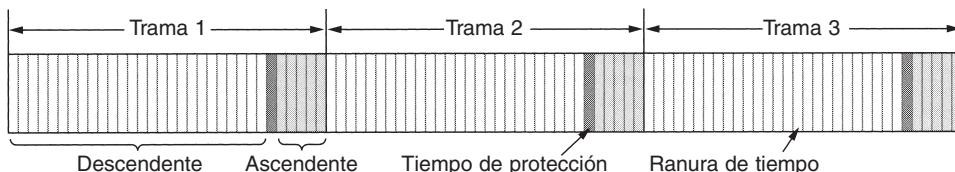


Figura 4-33. Tramas y ranuras de tiempo para duplexación por división de tiempo.

La estación base asigna el tráfico descendente en ranuras de tiempo. Además, controla por completo esta dirección. El tráfico ascendente es más complejo y depende de la calidad del servicio requerido. Más adelante volveremos a la asignación de ranuras, cuando analicemos la subcapa MAC.

Otra característica interesante de la capa física es su capacidad de empaquetar múltiples tramas MAC consecutivas en una sola transmisión física. Esta característica mejora la eficiencia espectral al reducir el número de preámbulos y encabezados de capa física necesarios.

Otro aspecto que vale la pena mencionar es el uso de los códigos de Hamming para realizar corrección de errores hacia delante en la capa física. La mayoría de las otras redes se basa simplemente en sumas de verificación para detectar errores y solicitar retransmisiones cuando se reciben tramas erróneas. Pero en el entorno de banda ancha de área amplia, se esperan tantos errores de transmisión que la corrección de errores se emplea en la capa física, además de sumas de verificación en las capas superiores. El objetivo de la corrección de errores es hacer que el canal luzca mejor de lo que realmente es (de la misma forma en que los CD-ROMs parecen ser muy confiables, pero sólo porque más de la mitad de los bits se destinan para la corrección de errores en la capa física).

4.5.4 El protocolo de la subcapa MAC del 802.16

La capa de enlace de datos se divide en tres subcapas, como vimos en la figura 4-31. Debido a que estudiaremos la criptografía hasta el capítulo 8 es difícil explicar ahora cómo funciona la

subcapa de seguridad. Basta decir que la codificación se utiliza para que todos los datos transmitidos se mantengan en secreto. Sólo las cargas útiles de las tramas se codifican; los encabezados no se codifican. Esta propiedad significa que un fisgón puede ver quién está hablándole a quién pero no puede saber qué se están diciendo.

Si usted ya sabe algo sobre criptografía, a continuación se muestra una explicación de un párrafo acerca de la subcapa de seguridad. Si no sabe nada sobre criptografía, es probable que el siguiente párrafo no sea muy ilustrativo para usted (pero podría considerar volver a leerlo después de terminar el capítulo 8).

Cuando un suscriptor se conecta a una estación base, realiza autenticación mutua con criptografía de clave pública RSA mediante certificados X.509. Las cargas útiles mismas se codifican mediante un sistema de clave simétrica, ya sea DES con cambio de bloques de código o triple DES con dos claves. Es probable que AES (Rijndael) se agregue pronto. La verificación de integridad utiliza SHA-1. Eso no es tan malo, ¿o sí?

Ahora veamos la parte común de la subcapa MAC. Las tramas MAC ocupan un número integral de ranuras de tiempo de la capa física. Cada trama se compone de subtramas, de las cuales las primeras dos son los mapas descendente y ascendente. Éstos indican lo que hay en cada ranura de tiempo y cuáles ranuras de tiempo están libres. El mapa descendente también contiene varios parámetros de sistema para informar de nuevas estaciones conforme entran en línea.

El canal descendente es muy directo. La estación base decide simplemente lo que se va a poner en cada subtrama. El canal ascendente es más complicado debido a que hay suscriptores no coordinados compitiendo por él. Su asignación está estrechamente relacionada con el aspecto de calidad del servicio. Hay cuatro clases de servicio definidas:

1. Servicio de tasa de bits constante.
2. Servicio de tasa de bits variable en tiempo real.
3. Servicio de tasa de bits variable no en tiempo real.
4. Servicio de mejor esfuerzo.

Todos los servicios del estándar 802.16 son orientados a la conexión, y cada conexión toma una de las clases de servicio mostradas anteriormente, que se determina cuando se configura la conexión. Este diseño es muy diferente al de 802.11 o al de Ethernet, los cuales no tienen conexiones en la subcapa MAC.

El servicio de tasa de bits constante está diseñado para transmitir voz descomprimida, como en un canal T1. Este servicio necesita enviar una cantidad predeterminada de datos en intervalos de tiempo predeterminados. Se aloja mediante la dedicación de ciertas ranuras de tiempo a cada conexión de este tipo. Una vez que se ha asignado el ancho de banda, las ranuras de tiempo quedan disponibles automáticamente, sin necesidad de solicitar cada una.

El servicio de tasa de bits variable en tiempo real está destinado para la multimedia comprimida y otras aplicaciones en tiempo real en las que la cantidad de ancho de banda necesaria puede variar en cada instante. Es ajustada por la estación base sondeando al suscriptor a un intervalo fijo para saber cuánto ancho de banda se necesita esta vez.

El servicio de tasa de bits variable en tiempo no real es para las transmisiones pesadas que no son en tiempo real, como transmisiones grandes de archivos. Para este servicio, la estación base sondea al suscriptor con mucha frecuencia. Un cliente de tasa de bits constante puede establecer un bit en una de sus tramas, solicitando un sondeo para enviar tráfico adicional (tasa de bits variable).

Si una estación no responde a un sondeo k veces en una fila, la estación base la coloca en un grupo de multidifusión y elimina su sondeo personal. En su lugar, cuando se sondea el grupo de multidifusión, cualquiera de las estaciones que conforman el grupo puede responder, compitiendo por el servicio. De esta forma, las estaciones con poco tráfico no desperdician sondeos valiosos.

Por último, el servicio de mejor esfuerzo es para todo lo demás. No se realiza sondeo y el suscriptor debe competir por ancho de banda con otros suscriptores de mejor servicio. Las solicitudes por ancho de banda se realizan en ranuras de tiempo que están marcadas en el mapa ascendente como disponibles para competencia. Si una solicitud es exitosa, su éxito se notará en el siguiente mapa de bits descendente. Si no es exitosa, los suscriptores no exitosos deberán tratar más tarde. Para minimizar las colisiones, se utiliza el algoritmo de retroceso exponencial binario.

El estándar define dos formas de asignación de ancho de banda: por estación y por conexión. En el primer caso, la estación suscriptora agrega las necesidades de todos los usuarios del edificio y realiza solicitudes colectivas por ellos. Cuando se le concede el ancho de banda, lo asigna entre sus usuarios como considere necesario. En el último caso, la estación base administra cada conexión de manera directa.

4.5.5 La estructura de trama 802.16

Todas las tramas MAC comienzan con un encabezado genérico. A éste le sigue una carga útil y una suma de verificación (CRC) opcionales, como se ilustra en la figura 4-34. La carga útil no es necesaria en las tramas de control, por ejemplo, en las que solicitan ranuras de canal. La suma de verificación también es (sorprendentemente) opcional, debido a la corrección de errores en la capa física y al hecho de que nunca se realiza un intento por retransmitir tramas en tiempo real. Si estos intentos nunca ocurren, ¿para qué molestarse con una suma de verificación?

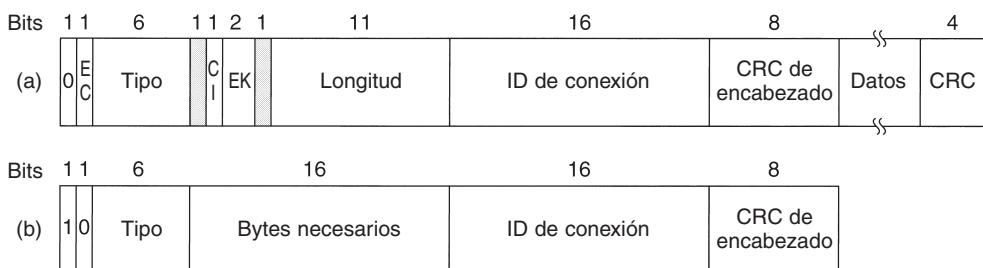


Figura 4-34. (a) Una trama genérica. (b) Una trama de solicitud de ancho de banda.

Un rápido vistazo a los campos de encabezado de la figura 4-34(a) es como sigue: el bit *EC* indica si la carga útil está encriptada. El campo *Tipo* identifica el tipo de la trama e indica principalmente si hay empaquetamiento y fragmentación. El campo *CI* indica la presencia o ausencia de la suma de verificación final. El campo *EK* indica cuál de las claves de encriptación se está utilizando (si es que se está utilizando alguna). El campo *Longitud* proporciona la longitud exacta de la trama, incluyendo la del encabezado. El *Identificador de conexión* indica a cuál conexión pertenece esta trama. Por último, el campo *CRC de encabezado* es la suma de verificación sólo del encabezado, que utiliza el polinomio $x^8 + x^2 + x + 1$.

En la figura 4-34(b) se muestra un segundo tipo de encabezado, para tramas que solicitan ancho de banda. Comienza con un bit 1 en lugar de uno 0 y es similar al encabezado genérico, excepto que el segundo y tercer bytes forman un número de 16 bits, lo que indica la cantidad de ancho de banda necesaria para transmitir el número de bytes especificados. Las tramas de solicitud de ancho de banda no transmiten datos útiles o un CRC de la trama completa.

Es posible decir muchísimas cosas más sobre el estándar 802.16, pero este no es el lugar para decirlo. Para mayor información, consulte el estándar mismo.

4.6 BLUETOOTH

En 1994, la empresa L. M. Ericsson se interesó en conectar sus teléfonos móviles y otros dispositivos (por ejemplo, PDAs,) sin necesidad de cables. En conjunto con otras cuatro empresas (IBM, Intel, Nokia y Toshiba), formó un SIG (grupo de interés especial, es decir, un consorcio) con el propósito de desarrollar un estándar inalámbrico para interconectar computadoras, dispositivos de comunicaciones y accesorios a través de radios inalámbricos de bajo consumo de energía, corto alcance y económicos. Al proyecto se le asignó el nombre **Bluetooth**, en honor de Harald Blaatand (Bluetooth) II (940-981), un rey vikingo que unificó (es decir, conquistó) Dinamarca y Noruega, también sin necesidad de cables. Aunque la idea original eran tan sólo prescindir de cables entre dispositivos, su alcance se expandió rápidamente al área de las LANs inalámbricas. Aunque esta expansión le dio más utilidad al estándar, también provocó el surgimiento de competencia con el 802.11. Para empeorar las cosas, los dos sistemas interfieren entre sí en el ámbito eléctrico. Asimismo, vale la pena hacer notar que Hewlett-Packard introdujo hace algunos años una red infrarroja para conectar periféricos de computadora sin cables, pero en realidad nunca alcanzó popularidad.

Sin desanimarse por esto, el SIG de Bluetooth emitió en julio de 1999 una especificación de 1500 páginas acerca de V1.0. Un poco después, el grupo de estándares del IEEE que se encarga de las redes de área personal inalámbricas, 802.15, adoptó como base el documento sobre Bluetooth y empezó a trabajar en él. A pesar de que podría parecer extraño estandarizar algo que ya cuenta con una especificación bien detallada, sin implementaciones incompatibles que tengan que armonizarse, la historia demuestra que al existir un estándar abierto manejado por un cuerpo neutral como el IEEE con frecuencia se estimula el uso de una tecnología. Para ser un poco más precisos, debe apuntarse que la especificación de Bluetooth está dirigida a un sistema completo, de la capa física a la capa de aplicación. El comité 802.15 del IEEE estandariza solamente las capas física y la de enlace de datos; el resto de la pila de protocolos está fuera de sus estatutos.

Aun cuando el IEEE aprobó en el 2002 el primer estándar para redes de área personal, 802.15.1, el SIG de Bluetooth continúa las mejoras. A pesar de que las versiones del SIG y del IEEE difieren, se espera que en breve coincidirán en un solo estándar.

4.6.1 Arquitectura de Bluetooth

Empecemos nuestro análisis del sistema Bluetooth con un rápido vistazo de sus elementos y de su propósito. La unidad básica de un sistema Bluetooth es una **piconet**, que consta de un nodo maestro y hasta siete nodos esclavos activos a una distancia de 10 metros. En una misma sala (grande) pueden encontrarse varias *piconets* y se pueden conectar mediante un nodo puente, como se muestra en la figura 4-35. Un conjunto de *piconets* interconectadas se denomina **scatternet**.

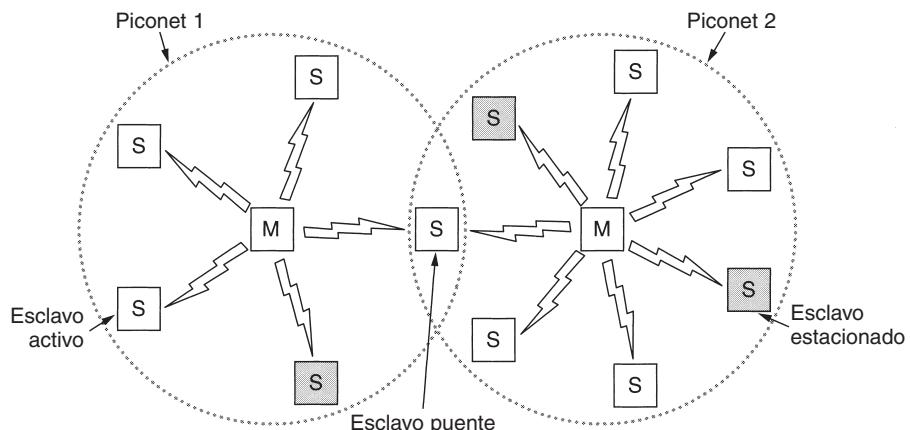


Figura 4-35. Dos piconets se pueden conectar para conformar una scatternet.

Además de los siete nodos esclavos activos de una *piconet*, puede haber hasta 255 nodos estacionados en la red. Éstos son dispositivos que el nodo maestro ha cambiado a un estado de bajo consumo de energía para reducir el desgaste innecesario de sus pilas. Lo único que un dispositivo en estado estacionado puede hacer es responder a una señal de activación por parte del maestro. También hay dos estados intermedios, *hold* y *sniff*, pero no nos ocuparemos de ellos aquí.

La razón para el diseño maestro/esclavo es que los diseñadores pretendían facilitar la implementación de chips Bluetooth completos por debajo de 5 dólares. La consecuencia de esta decisión es que los esclavos son sumamente pasivos y realizan todo lo que los maestros les indican. En esencia, una *piconet* es un sistema TDM centralizado, en el cual el maestro controla el reloj y determina qué dispositivo se comunica en un momento determinado. Todas las comunicaciones se realizan entre el maestro y el esclavo; no existe comunicación directa de esclavo a esclavo.

4.6.2 Aplicaciones de Bluetooth

La mayoría de los protocolos de red sólo proporcionan canales entre las entidades que se comunican y permiten a los diseñadores de aplicaciones averiguar para qué desean utilizarlos. Por ejemplo, el 802.11 no especifica si los usuarios deben utilizar sus computadoras portátiles para leer correo electrónico, navegar por la Red o cualquier otro uso. En contraste, la especificación Bluetooth V1.1 designa el soporte de 13 aplicaciones en particular y proporciona diferentes pilas de protocolos para cada una. Desgraciadamente, este enfoque conlleva una gran complejidad, que aquí omitiremos. En la figura 4-36 se describen las 13 aplicaciones, las cuales se denominan **perfiles**. Al analizarlas brevemente en este momento, veremos con mayor claridad lo que pretende el SIG de Bluetooth.

Nombre	Descripción
Acceso genérico	Procedimientos para el manejo de enlaces
Descubrimiento de servicios	Protocolo para descubrir los servicios que se ofrecen
Puerto serie	Reemplazo para un cable de puerto serie
Intercambio genérico de objetos	Define la relación cliente-servidor para el traslado de objetos
Acceso a LAN	Protocolo entre una computadora móvil y una LAN fija
Acceso telefónico a redes	Permite que una computadora portátil realice una llamada por medio de un teléfono móvil
Fax	Permite que un fax móvil se comunique con un teléfono móvil
Telefonía inalámbrica	Conecta un handset (teléfono) con su estación base local
Intercom (Intercomunicador)	Walkie-talkie digital
Headset (Diadema telefónica)	Posibilita la comunicación de voz sin utilizar las manos
Envío de objetos	Ofrece una manera de intercambiar objetos simples
Transferencia de archivos	Proporciona una característica para transferencia de archivos más general
Sincronización	Permite a un PDA sincronizarse con otra computadora

Figura 4-36. Los perfiles de Bluetooth.

El perfil de acceso genérico no es realmente una aplicación, sino más bien la base sobre la cual se construyen las aplicaciones; su tarea principal es ofrecer una manera para establecer y mantener enlaces (canales) seguros entre el maestro y los esclavos. El perfil de descubrimiento de servicios también es relativamente genérico; los dispositivos lo utilizan para descubrir qué servicios ofrecen otros dispositivos. Se espera que todos los dispositivos Bluetooth implementen estos dos perfiles. Los restantes son opcionales.

El perfil de puerto serie es un protocolo de transporte que la mayoría de los perfiles restantes utiliza. Emula una línea serie y es especialmente útil para aplicaciones heredadas que requieren una línea serie.

El perfil de intercambio genérico define una relación cliente-servidor para el traslado de datos. Los clientes inician operaciones, pero tanto un cliente como un servidor pueden fungir como esclavo. Al igual que el perfil de puerto serie, es la base para otros perfiles.

El siguiente grupo de tres perfiles está destinado a la conectividad. El perfil de acceso a LAN permite a un dispositivo Bluetooth conectarse a una red fija; este perfil es competencia directa del estándar 802.11. El perfil de acceso telefónico a redes fue el propósito original de todo el proyecto; permite a una computadora portátil conectarse a un teléfono móvil que contenga un módem integrado, sin necesidad de cables. El perfil de fax es parecido al de acceso telefónico a redes, excepto que posibilita a máquinas de fax inalámbricas enviar y recibir faxes a través de teléfonos móviles sin que exista una conexión por cable entre ambos.

Los tres perfiles siguientes son para telefonía. El perfil de telefonía inalámbrica proporciona una manera de conectar el *handset* (teléfono) de un teléfono inalámbrico a la estación base. En la actualidad, la mayoría de los teléfonos inalámbricos no se puede utilizar también como teléfonos móviles, pero quizás en el futuro se puedan combinar los teléfonos inalámbricos y los móviles. El perfil intercom hace posible que dos teléfonos se conecten como *walkie-talkies*. Por último, con el perfil *headset* (diadema telefónica) se puede realizar comunicación de voz entre la diadema telefónica y su estación base, por ejemplo, para comunicarse telefónicamente sin necesidad de utilizar las manos al manejar un automóvil.

Los tres perfiles restantes sirven para intercambiar objetos entre dos dispositivos inalámbricos, como tarjetas de presentación, imágenes o archivos de datos. En particular, el propósito del perfil de sincronización es cargar datos en un PDA o en una computadora portátil cuando se está fuera de casa y de recabar estos datos al llegar a casa.

¿Era realmente necesario explicar en detalle todas estas aplicaciones y proporcionar diferentes pilas de protocolos para cada una? Tal vez no, pero esto se debió a que fueron diversos grupos de trabajo los que diseñaron las diferentes partes del estándar y cada uno se enfocó en su problema específico y generó su propio perfil. La ley de Conway podría aplicarse en esta situación. (En el número de abril de 1968 de la revista *Datamation*, Melvin Conway apuntó que si se asignan n personas a escribir un compilador, se obtendrá un compilador de n pasos, o, en forma más general, la estructura del software reflejará la estructura del grupo que la produjo.) Quizás dos pilas de protocolos habrían sido suficientes en lugar de 13, una para la transferencia de archivos y otra para posibilitar el flujo continuo de la comunicación en tiempo real.

4.6.3 La pila de protocolos de Bluetooth

El estándar Bluetooth cuenta con muchos protocolos agrupados con poco orden en capas. La estructura de capas no sigue el modelo OSI, el modelo TCP/IP, el modelo 802 o algún otro modelo conocido. Sin embargo, el IEEE se encuentra modificando actualmente Bluetooth para ajustarlo al modelo 802. En la figura 4-37 se muestra la arquitectura básica de protocolos de Bluetooth tal como la modificó el comité 802.

La capa inferior es la capa de radio física, la cual es bastante similar a la capa física de los modelos OSI y 802. Se ocupa de la transmisión y la modulación de radio. Aquí, gran parte del interés se enfoca en el objetivo de lograr que el sistema tenga un costo bajo para que pueda entrar al mercado masivo.

La capa de banda base tiene algunos puntos en común con la subcapa MAC, aunque también incluye elementos de la capa física. Se encarga de la manera en que el maestro controla las ranuras de tiempo y de que éstas se agrupen en tramas.

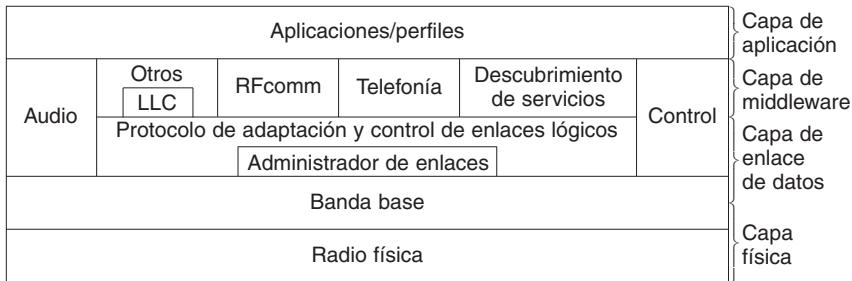


Figura 4-37. Versión 802.15 de la arquitectura de protocolos de Bluetooth.

A continuación se encuentra una capa con un grupo de protocolos un tanto relacionados. El administrador de enlaces se encarga de establecer canales lógicos entre dispositivos, incluyendo administración de energía, autenticación y calidad de servicio. El protocolo de adaptación y control de enlaces lógicos (también conocido como L2CAP) aísla a las capas superiores de los detalles de la transmisión. Es análogo a la subcapa LLC del estándar 802, pero difiere de ésta en el aspecto técnico. Como su nombre indica, los protocolos de audio y control se encargan del audio y el control, respectivamente. Las aplicaciones pueden acceder a ellos de manera directa sin necesidad de pasar por el protocolo L2CAP.

La siguiente capa hacia arriba es la de middleware, que contiene una mezcla de diferentes protocolos. El IEEE incorporó aquí la subcapa LLC del 802 para ofrecer compatibilidad con las redes 802. Los protocolos RFcomm, de telefonía y de descubrimiento de servicios son nativos. RFcomm (comunicación de Radio Frecuencia) es el protocolo que emula el puerto serie estándar de las PCs para la conexión de teclados, ratones y módems, entre otros dispositivos. Su propósito es permitir que dispositivos heredados lo utilicen con facilidad. El protocolo de telefonía es de tiempo real y está destinado a los tres perfiles orientados a voz. También se encarga del establecimiento y terminación de llamadas. Por último, el protocolo de descubrimiento de servicios se emplea para localizar servicios dentro de la red.

En la capa superior es donde se ubican las aplicaciones y los perfiles, que utilizan a los protocolos de las capas inferiores para realizar su trabajo. Cada aplicación tiene su propio subconjunto dedicado de protocolos. Por lo general, los dispositivos específicos, como las diademas telefónicas, contienen únicamente los protocolos necesarios para su aplicación.

En las siguientes secciones examinaremos las tres capas inferiores de la pila de protocolos de Bluetooth, ya que éstas corresponden más o menos con las subcapas física y MAC.

4.6.4 La capa de radio de Bluetooth

La capa de radio traslada los bits del maestro al esclavo, o viceversa. Es un sistema de baja potencia con un rango de 10 metros que opera en la banda ISM de 2.4 GHz. La banda se divide en 79 canales de 1 MHz cada uno. La modulación es por desplazamiento de frecuencia, con 1 bit por Hz, lo cual da una tasa de datos aproximada de 1 Mbps, pero gran parte de este espectro la consume la

sobrecarga. Para asignar los canales de manera equitativa, el espectro de saltos de frecuencia se utiliza a 1600 saltos por segundo y un tiempo de permanencia de 625 µseg. Todos los nodos de una *piconet* saltan de manera simultánea, y el maestro establece la secuencia de salto.

Debido a que tanto el 802.11 como Bluetooth operan en la banda ISM de 2.4 GHz en los mismos 79 canales, interfieren entre sí. Puesto que Bluetooth salta mucho más rápido que el 802.11, es más probable que un dispositivo Bluetooth dañe las transmisiones del 802.11 que en el caso contrario. Como el 802.11 y el 802.15 son estándares del IEEE, éste busca una solución para el problema, aunque no es tan sencilla porque ambos sistemas utilizan la banda ISM por la misma razón: no se requiere licencia para su uso. El estándar 802.11a emplea la otra banda ISM (5 GHz), pero tal estándar tiene un rango mucho más corto que el 802.11b (debido a la física de las ondas de radio), por lo cual el 802.11a no es una solución idónea en todos los casos. Algunas empresas han recurrido a la prohibición total de Bluetooth para solucionar el problema. Una solución de mercado es que la red más potente (en los aspectos político y económico, no eléctrico) solicite que la parte más débil modifique su estándar para dejar de interferir con ella. (Lansford y cols., 2001) dan algunas opiniones al respecto.

4.6.5 La capa de banda base de Bluetooth

La capa de banda base de Bluetooth es lo más parecido a una subcapa MAC. Esta capa convierte el flujo de bits puros en tramas y define algunos formatos clave. En la forma más sencilla, el maestro de cada *piconet* define una serie de ranuras de tiempo de 625 µseg y las transmisiones del maestro empiezan en las ranuras pares, y las de los esclavos, en las ranuras impares. Ésta es la tradicional multiplexión por división de tiempo, en la cual el maestro acapara la mitad de las ranuras y los esclavos comparten la otra mitad. Las tramas pueden tener 1, 3 o 5 ranuras de longitud.

La sincronización de saltos de frecuencia permite un tiempo de asentamiento de 250-260 µseg por salto para que los circuitos de radio se estabilicen. Es posible un asentamiento más rápido, pero a un mayor costo. Para una trama de una sola ranura, después del asentamiento, se desechan 366 de los 625 bits. De éstos, 126 se utilizan para un código de acceso y el encabezado, y 240 para los datos. Cuando se enlazan cinco ranuras, sólo se necesita un periodo de asentamiento y se utiliza uno ligeramente más corto, de tal manera que de los $5 \times 625 = 3125$ bits de las cinco ranuras de tiempo, 2781 se encuentran disponibles para la capa de banda base. Así, las tramas más grandes son mucho más eficientes que las de una sola ranura.

Cada trama se transmite por un canal lógico, llamado **enlace**, entre el maestro y un esclavo. Hay dos tipos de enlaces. El primero es el **ACL (Asíncrono no Orientado a la Conexión)**, que se utiliza para datos commutados en paquetes disponibles a intervalos irregulares. Estos datos provienen de la capa L2CAP en el nodo emisor y se entregan en la capa L2CAP en el nodo receptor. El tráfico ACL se entrega sobre la base de mejor esfuerzo. No hay garantías. Las tramas se pueden perder y tienen que retransmitirse. Un esclavo puede tener sólo un enlace ACL con su maestro.

El otro tipo de enlace es el **SCO (Síncrono Orientado a la Conexión)**, para datos en tiempo real, como ocurre en las conexiones telefónicas. A este tipo de canal se le asigna una ranura fija en cada dirección. Por la importancia del tiempo en los enlaces SCO, las tramas que se envían a través de ellos nunca se retransmiten. En vez de ello, se puede utilizar la corrección de errores hacia delante (o corrección de errores sin canal de retorno) para conferir una confiabilidad alta. Un esclavo puede establecer hasta tres enlaces SCO con su maestro. Cada enlace de este tipo puede transmitir un canal de audio PCM de 64,000 bps.

4.6.6 La capa L2CAP de Bluetooth

La capa L2CAP tiene tres funciones principales. Primera, acepta paquetes de hasta 64 KB provenientes de las capas superiores y los divide en tramas para transmitirlos. Las tramas se reensamblan nuevamente en paquetes en el otro extremo.

Segunda, maneja la multiplexión y desmultiplexión de múltiples fuentes de paquetes. Cuando se reensambla un paquete, la capa L2CAP determina cuál protocolo de las capas superiores lo manejará, por ejemplo, el RFcomm o el de telefonía.

Tercera, la capa L2CAP se encarga de la calidad de los requerimientos de servicio, tanto al establecer los enlaces como durante la operación normal. Asimismo, durante el establecimiento de los enlaces se negocia el tamaño máximo de carga útil permitido, para evitar que un dispositivo que envíe paquetes grandes sature a uno que reciba paquetes pequeños. Esta característica es importante porque no todos los dispositivos pueden manejar paquetes de 64 KB.

4.6.7 Estructura de la trama de Bluetooth

Existen diversos formatos de trama, el más importante de los cuales se muestra en la figura 4-38. Empieza con un código de acceso que identifica al maestro, cuyo propósito es que los esclavos que se encuentren en el rango de alcance de dos maestros sepan cuál tráfico es para ellos. A continuación se encuentra un encabezado de 54 bits que contiene campos comunes de la subcapa MAC. Luego está el campo de datos, de hasta 2744 bits (para una transmisión de cinco ranuras). Para una sola ranura de tiempo, el formato es el mismo excepto que el campo de datos es de 240 bits.

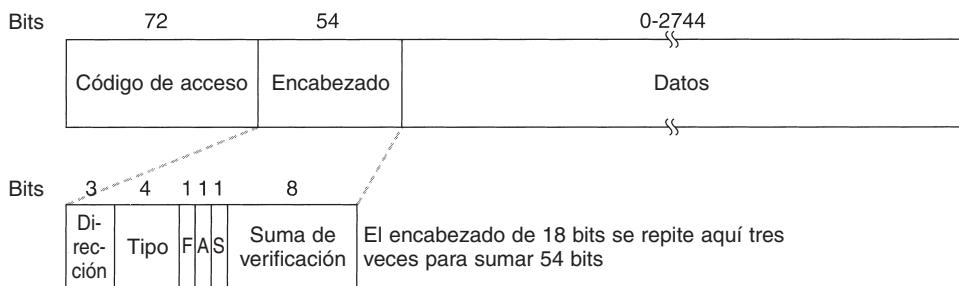


Figura 4-38. Trama de datos típica de Bluetooth.

Demos un rápido vistazo al encabezado. El campo *Dirección* identifica a cuál de los ocho dispositivos activos está destinada la trama. El campo *Tipo* indica el tipo de trama (ACL, SCO, de sondeo o nula), el tipo de corrección de errores que se utiliza en el campo de datos y cuántas ranuras de longitud tiene la trama. Un esclavo establece el bit *F* (de flujo) cuando su búfer está lleno y no puede recibir más datos. Ésta es una forma primitiva de control de flujo. El bit *A* (de confirmación de recepción) se utiliza para incorporar un ACK en una trama. El bit *S* (de secuencia) sirve para numerar las tramas con el propósito de detectar retransmisiones. El protocolo es de parada y espera, por lo que 1 bit es suficiente. A continuación se encuentra el encabezado *Suma de verificación* de 8 bits. Todo el encabezado de 18 bits se repite tres veces para formar el encabezado de 54 bits que se aprecia en la figura 4-38. En el receptor, un circuito sencillo examina las tres copias de cada bit. Si son las mismas, el bit es aceptado. De lo contrario, se impone la opinión de la mayoría. De esta forma, 54 bits de capacidad de transmisión se utilizan para enviar 10 bits de encabezado. Esto se debe a que es necesaria una gran cantidad de redundancia para enviar datos de manera confiable en un entorno con ruido mediante dispositivos de bajo costo y baja potencia (2.5 mW) con poca capacidad de cómputo.

En el campo de datos de las tramas ACL se utilizan varios formatos. Sin embargo, las tramas SCO son más sencillas: el campo de datos siempre es de 240 bits. Se definen tres variantes, que permiten 80, 160 y 240 bits de carga útil real, y el resto se utiliza para corrección de errores. En la versión más confiable (carga útil de 80 bits), el contenido se repite tres veces, al igual que el encabezado.

Debido a que el esclavo podría utilizar solamente las ranuras impares, obtiene 800 ranuras por segundo, de la misma manera que el maestro. Con una carga útil de 80 bits, la capacidad de canal del esclavo es de 64,000 bps y la del maestro también es de 64,000 bps, exactamente la necesaria para un canal de voz PCM dúplex total (razón por la cual se eligió una tasa de saltos de 1600 saltos por segundo). Estas cifras indican que un canal de voz dúplex total con 64,000 bps en cada dirección y el formato más confiable satura totalmente la *piconet* a pesar de un ancho de banda puro de 1 Mbps. En la variante menos confiable (240 bits por ranura sin redundancia a este nivel), se pueden soportar al mismo tiempo tres canales de voz dúplex total, debido a lo cual se permite un máximo de tres enlaces SCO por esclavo.

Hay mucho más por decir acerca de Bluetooth, pero ya no tenemos espacio aquí. Si desea más información, vea (Bhagwat, 2001; Bisdikian, 2001; Bray y Sturman, 2002; Haartsen, 2000; Johansson y cols., 2001; Miller y Bisdikian, 2001, y Sairam y cols., 2002).

4.7 CONMUTACIÓN EN LA CAPA DE ENLACE DE DATOS

Muchas organizaciones tienen varias LANs y desean interconectarlas. Este tipo de redes se puede conectar mediante dispositivos llamados **puentes**, que funcionan en la capa de enlace de datos. Los puentes examinan las direcciones de la capa de enlace de datos para enrutar los datos. Como no tienen que examinar el campo de carga útil de las tramas que enrutan, pueden transportar paquetes IPv4 (que se utilizan actualmente en Internet), IPv6 (que se utilizarán en el futuro en Internet), AppleTalk, ATM, OSI o de otros tipos. En contraste, los *enrutadores* examinan

las direcciones de los paquetes y realizan su trabajo de enrutamiento con base en ellas. Aunque ésta parece una clara división entre los puentes y los enrutadores, algunos desarrollos modernos, como el surgimiento de la Ethernet conmutada, han enturbiado las aguas, como veremos más tarde. En las siguientes secciones analizaremos los puentes y los conmutadores, en especial para conectar diferentes LANs 802. Para un análisis más completo sobre puentes, conmutadores y temas relacionados, vea (Perlman, 2000).

Antes de entrar de lleno a la tecnología de los puentes, vale la pena dar un vistazo a algunas situaciones comunes en las cuales se utilizan los puentes. Mencionaremos seis razones por las cuales una sola organización podría contar con varias LANs.

Primera, muchas universidades y departamentos corporativos tienen sus propias LANs, principalmente para interconectar sus propias computadoras personales, estaciones de trabajo y servidores. Dado que las metas de los distintos departamentos difieren, los departamentos escogen LANs diferentes, sin importarles lo que hagan los demás departamentos. Tarde o temprano surge la necesidad de interacción, y aquí es donde entran los puentes. En este ejemplo surgieron múltiples LANs debido a la autonomía de sus dueños.

Segunda, la organización puede estar distribuida geográficamente en varios edificios, separados por distancias considerables. Puede ser más económico tener LANs independientes en cada edificio y conectarlas mediante puentes y enlaces láser que tender un solo cable por toda la zona.

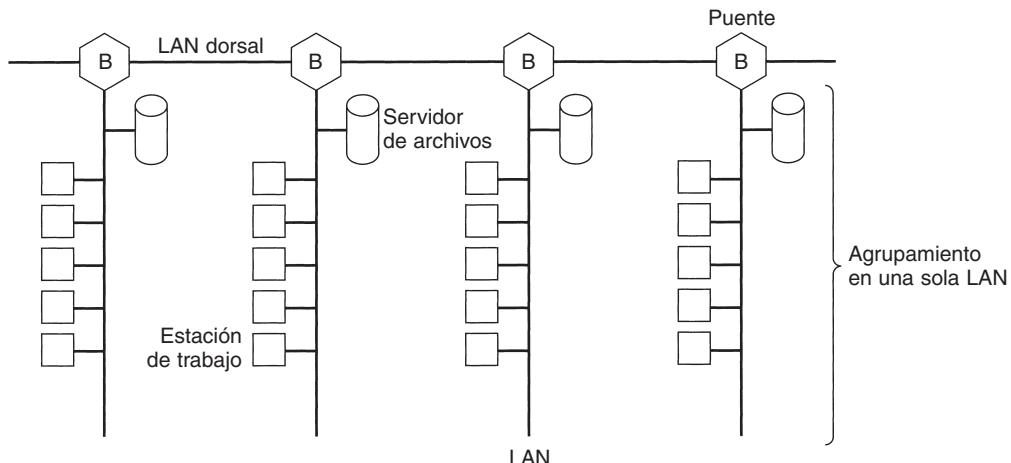


Figura 4-39. Varias LANs conectadas mediante una red dorsal para manejar una carga total mayor que la capacidad de una sola LAN.

Tercera, puede ser necesario dividir lo que lógicamente es una sola LAN en LANs individuales para manejar la carga. Por ejemplo, en muchas universidades miles de estaciones de trabajo están disponibles para los estudiantes y el cuerpo docente. Los archivos normalmente se guardan en máquinas servidoras de archivos, y son descargados a las máquinas de los usuarios a solicitud. La enorme escala de este sistema hace imposible poner todas las estaciones de trabajo en una sola

LAN, pues el ancho de banda total necesario es demasiado grande. En cambio, se usan varias LANs conectadas mediante puentes, como se muestra en la figura 4-39. Cada LAN contiene un grupo de estaciones de trabajo con su propio servidor de archivos, por lo que la mayor parte del tráfico está restringida a una sola LAN y no agrega carga a la red dorsal.

Vale la pena mencionar que aunque por lo general esquematizamos las LANs como cables con múltiples derivaciones como en la figura 4-39 (la apariencia clásica), hoy en día se implementan con mayor frecuencia mediante concentradores o comutadores. Sin embargo, un cable largo con múltiples derivaciones para las máquinas, y un concentrador con las máquinas conectadas a él funcionan de manera idéntica. En ambos casos, todas las máquinas pertenecen al mismo dominio de colisión y todas utilizan el protocolo CSMA/CD para enviar tramas. No obstante, las LANs commutadas son diferentes, como ya vimos y volveremos a ver en breve.

Cuarta, en algunas situaciones una sola LAN sería adecuada en términos de la carga, pero la distancia física entre las máquinas más distantes es demasiado grande (por ejemplo, mayor que 2.5 km para Ethernet). Aun si fuera fácil tender el cable, la red no funcionaría debido al retardo excesivamente grande del viaje de ida y vuelta. La única solución es segmentar la LAN e instalar puentes entre los segmentos. Con puentes, puede aumentarse la distancia física total cubierta.

Quinta, está la cuestión de la confiabilidad. En una sola LAN, un nodo defectuoso que envíe constantemente una cadena de basura echará a perder la LAN. Pueden introducirse puentes en lugares críticos, como puertas para bloquear el fuego en un edificio, y así evitar que un solo nodo enloquecido tire el sistema completo. A diferencia de un repetidor, que sólo copia lo que ve, un puente puede programarse para discriminar lo que envía y lo que no.

Sexta y última, los puentes pueden contribuir a la seguridad de la organización. La mayor parte de las interfaces LAN tienen un **modo promiscuo**, en el que *todas* las tramas se entregan a la computadora, no sólo las dirigidas a ella. Los espías y los intrusos aman esta característica. Al introducir puentes en varios lugares y tener cuidado de no reenviar tráfico delicado, es posible aislar partes de la red de manera que su tráfico no pueda escapar y caer en las manos equivocadas.

En el plano ideal, los puentes deberían ser totalmente transparentes, es decir, debería ser posible cambiar una máquina de un segmento a otro sin necesidad de modificar el hardware, software o tablas de configuración. Asimismo, debería ser posible que las máquinas de un segmento se comunicaran con las de cualquier otro segmento sin que importara el tipo de LAN que se utilizara en ambos segmentos o en los segmentos que hubiera entre ellos. Este objetivo se consigue en ocasiones, pero no siempre.

4.7.1 Puentes de 802.x a 802.y

Después de comprender por qué son necesarios los puentes, pasemos a la cuestión de su funcionamiento. En la figura 4-40 se ilustra la operación de un puente sencillo de dos puertos. El *host A* en una LAN (802.11) inalámbrica tiene un paquete por enviar a un *host fijo, B*, en una Ethernet (802.3) a la cual se encuentra conectada la LAN inalámbrica. El paquete desciende a la subcapa LLC y adquiere un encabezado LLC (aparece en negro en la figura). A continuación el paquete pasa a la subcapa MAC y se le antepone un encabezado 802.11 (también un terminador, que no se

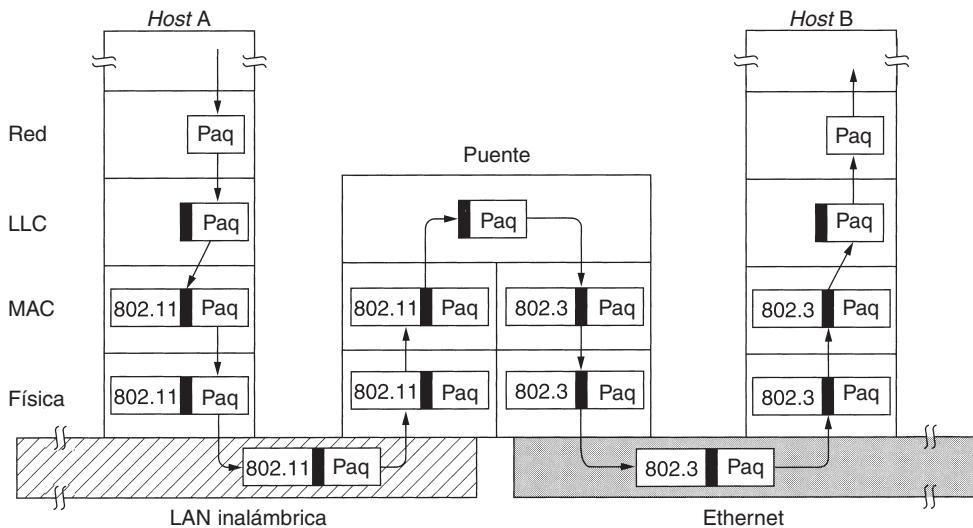


Figura 4-40. Operación de un puente entre una red 802.11 y una 802.3.

muestra en la figura). Esta unidad viaja a través del aire y es captada por la estación base, que se percata de que tiene que pasar por la Ethernet fija. Cuando el paquete llega al puente que conecta la red 802.11 con la 802.3, empieza en la capa física y realiza el recorrido hacia arriba. El encabezado 802.11 se elimina en la subcapa MAC del puente. El paquete recortado (con el encabezado LLC) se pasa a la subcapa LLC del puente. En este ejemplo, el paquete está destinado a una LAN 802.3, por lo que recorre la ruta hacia abajo en el lado 802.3 del puente y al terminar pasa a la red Ethernet. Observe que un puente que conecta k LANs diferentes tendrá k subcapas MAC diferentes y k capas físicas diferentes, una para cada tipo.

Hasta aquí parecía que es muy sencillo desplazar una trama de una LAN a otra. No es así. En esta sección señalaremos algunos de los problemas que se enfrentan al intentar construir un puente entre las diversas LANs (y MANs) 802. Nos concentraremos en 802.3, 802.11 y 802.16, pero existen otras, cada una con sus propios problemas.

Para empezar, cada LAN utiliza un formato de trama distinto (vea la figura 4-41). En contraste con las diferencias entre Ethernet, token bus y token ring, que se originaron por la historia y el ego de las grandes corporaciones, aquí las diferencias son válidas hasta cierto punto. Por ejemplo, el campo *Duración* en 802.11 se justifica por el protocolo MACAW y no tiene sentido en Ethernet. En consecuencia, cualquier copia que se realice entre LANs distintas requiere de volver a dar formato, lo que lleva tiempo de CPU, una nueva suma de verificación y se presenta la posibilidad de errores sin detectar debido a bits erróneos en la memoria del puente.

Un segundo problema es que las LANs interconectadas no necesariamente operan a la misma tasa de datos. Al retransmitir una gran cantidad de tramas una tras otra, provenientes de una LAN rápida a otra más lenta, el puente será incapaz de despachar las tramas con la misma rapidez que

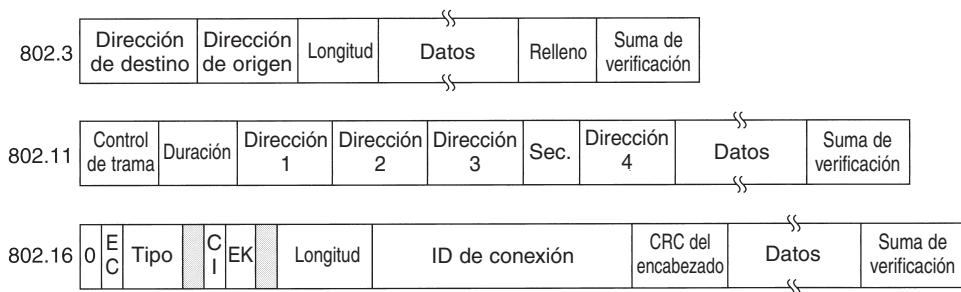


Figura 4-41. Formatos de trama de la redes 802. El dibujo no es a escala.

arriban. Por ejemplo, si una Gigabit Ethernet envía bits a su velocidad máxima a una LAN 802.11b de 11 Mbps, el puente tendrá que almacenarlos en búfer, con la esperanza de no agotar su memoria. Los puentes que conectan tres o más LANs tienen un problema similar cuando varias LANs intentan enviar datos a una misma LAN al mismo tiempo aun cuando todas operen a la misma velocidad.

Un tercer problema, y potencialmente el más grave de todos, es que distintas LANs 802 tienen diferentes longitudes máximas de trama. Un problema obvio surge cuando una trama grande tiene que reenviarse a una LAN que no puede aceptarla. En esta capa no es posible dividir la trama. Todos los protocolos dan por sentado que las tramas llegan o se pierden. No se considera el reensamblado de las tramas a partir de unidades más pequeñas. Lo anterior no significa que no se pueden diseñar tales protocolos. Es posible y se ha hecho. Es sólo que ningún protocolo de enlace de datos confiere esta característica, así que los puentes deben olvidarse de manipular la carga útil de la trama. En esencia, no hay solución. Las tramas demasiado grandes para reenviarse deben descartarse. Es suficiente sobre la transparencia.

Otro punto es la seguridad. Tanto el 802.11 como el 802.16 soportan encriptación en la capa de enlace de datos. Ethernet no. Esto significa que los diversos servicios de encriptación disponibles en las redes inalámbricas se pierden cuando el tráfico pasa sobre una Ethernet. Peor aún, si una estación inalámbrica emplea encriptación en la capa de enlace de datos, no habrá forma de desencriptar los datos cuando lleguen a la red Ethernet. Si la estación inalámbrica no utiliza encriptación, su tráfico quedará expuesto en el enlace aéreo. De cualquier manera hay un problema.

Una solución al problema de la seguridad es realizar la encriptación en una capa superior, pero en este caso la estación 802.11 tiene que saber si se está comunicando con otra estación sobre una red 802.11 (lo que significa que utilizará encriptación en la capa de enlace de datos) o con una distinta (en cuyo caso no utilizará encriptación). Al obligar a la estación a elegir se destruye la transparencia.

Un punto final es la calidad del servicio. Tanto el 802.11 como el 802.16 la ofrecen en diversas formas, el primero con el modo PCF y el último mediante conexiones a tasas de bits constantes. En Ethernet no existe el concepto de calidad del servicio, así que el tráfico proveniente de alguna de las anteriores perderá su calidad de servicio al pasar por una Ethernet.

4.7.2 Interconectividad local

La sección anterior examinó los problemas que surgen al conectar dos LANs IEEE 802 distintas mediante un solo puente. Sin embargo, en grandes organizaciones con muchas LANs, la sola interconexión entre todas da lugar a muchos problemas, aun cuando todas sean Ethernet. En un plano ideal, debería bastar con adquirir puentes diseñados para el estándar IEEE e insertar los conectores en el puente para que todo funcionara perfectamente al instante. No deberían ser necesarios cambios de hardware ni de software, ni configurar conmutadores de direcciones, descargar tablas de enrutamiento ni parámetros, nada. Tan sólo conectar los cables y empezar a trabajar. Más aún, los puentes no deberían afectar de ninguna manera el funcionamiento de LANs existentes. En otras palabras, los puentes deberían ser completamente transparentes (invisibles para todo el hardware y el software). Sorprendentemente, esto es posible. Echemos un vistazo a la manera en que se hace realidad esta magia.

En su forma más sencilla, un puente transparente funciona en modo promiscuo y acepta todas las tramas transmitidas sobre las LANs a las cuales está conectado. Tomemos como ejemplo la configuración de la figura 4-42. El puente B1 está conectado a las LANs 1 y 2, y el puente B2 está conectado a las LANs 2, 3 y 4. Una trama que llega al puente B1 en la LAN 1 con destino a A se puede descartar de inmediato porque se encuentra en la LAN correcta, pero una trama que llega a la LAN 1 con destino a C o F debe reenviarse.

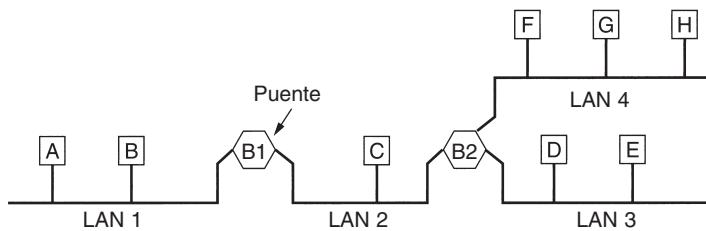


Figura 4-42. Configuración con cuatro LANs y dos puentes.

Cuando llega una trama, un puente debe decidir si la descarta o la reenvía, y si elige lo último, a cuál LAN la mandará. Esta decisión la toma consultando la dirección de destino en una enorme tabla (de *hash*) que se encuentra en su interior. La tabla lista cada posible destino e indica a cuál línea de salida (LAN) pertenece la trama. Por ejemplo, la tabla de B2 podría listar que A pertenece a LAN 2, ya que todo lo que B2 tiene que saber es a cuál LAN enviar las tramas para A. No le preocupa en absoluto el hecho de que posteriormente ocurran más reenvíos.

Cuando los puentes se conectan por primera vez, todas las tablas de *hash* están vacías. Ninguno de los puentes sabe dónde se encuentran los destinos, por lo que utilizan un algoritmo de inundación: todas las tramas que llegan con un destino desconocido se envían a todas las LANs a las cuales está conectado el puente, excepto a aquélla de la cual proceden. Con el paso del tiempo, los puentes aprenden dónde están los destinos, como se describe más adelante. Una vez conocido un destino, las tramas para él se reenvían solamente a la LAN apropiada en lugar de a todas las LANs.

El algoritmo que los puentes transparentes utilizan es **aprendizaje hacia atrás**. Como ya mencionamos, los puentes funcionan en modo promiscuo y de esta manera pueden ver todas las tramas que se envían a cualquiera de sus LANs. Al examinar la dirección del origen, pueden saber cuál máquina está disponible en cuál LAN. Por ejemplo, si el puente B1 de la figura 4-42 ve una trama proveniente de *C* en la LAN 2, sabe que es posible acceder a *C* por medio de la LAN 2, así que registra una entrada en su tabla de *hash* con la observación de que las tramas para *C* deben utilizar la LAN 2. Cualquier trama subsecuente dirigida a *C* que llegue desde la LAN 1 será reenviada, pero una trama para *C* que llegue desde la LAN 2 será descartada.

La topología puede cambiar conforme las máquinas y los puentes se enciendan y apaguen, o cuando se trasladen de un sitio a otro. Para manejar topologías dinámicas, siempre que se realiza una entrada en una tabla de *hash* se registra en la entrada la hora de llegada de una trama. Siempre que llega una trama cuyo origen ya está en la tabla, su entrada se actualiza con la hora actual. Por lo tanto, la hora asociada a cada entrada indica la última vez que se registró una trama proveniente de ese origen.

Un proceso del puente analiza periódicamente la tabla de *hash* y purga todas las entradas que tengan más de algunos minutos. De esta manera, si una computadora se desconecta de su LAN, se traslada a otro lugar del edificio y se vuelve a conectar en algún otro lugar, en pocos minutos volverá a funcionar con normalidad, sin necesidad de intervención manual. Este algoritmo también significa que si una máquina está inactiva durante algunos minutos, el tráfico destinado a ella será inundado hasta que la máquina misma envíe una trama.

El procedimiento de enrutamiento para una trama entrante depende de la LAN de que proceda (la LAN de origen) y de la LAN a la cual está destinada (la LAN de destino), como se puede ver a continuación:

1. Si las LANs de destino y de origen son la misma, descartar la trama.
2. Si las LANs de destino y de origen son diferentes, reenviar la trama.
3. Si se desconoce la LAN de destino, recurrir a la inundación.

Este algoritmo debe aplicarse cada vez que llega una trama. Chips VLSI especiales realizan la consulta y actualización de las entradas de la tabla en tan sólo algunos microsegundos.

4.7.3 Puentes con árbol de expansión

Para incrementar la confiabilidad, algunos sitios utilizan dos o más puentes en paralelo entre pares de LANs, como se muestra en la figura 4-43. Sin embargo, este arreglo también genera algunos problemas adicionales porque produce ciclos en la topología.

Un ejemplo simple de estos problemas lo tenemos al observar cómo se maneja una trama, *F*, con destino desconocido, en la figura 4-43. Cada puente, siguiendo las reglas normales para el manejo de destinos desconocidos, recurre a la inundación, que en este ejemplo es tan sólo copiar la trama a la LAN 2. Poco después, el puente 1 detecta a *F*₂, una trama con destino desconocido, y

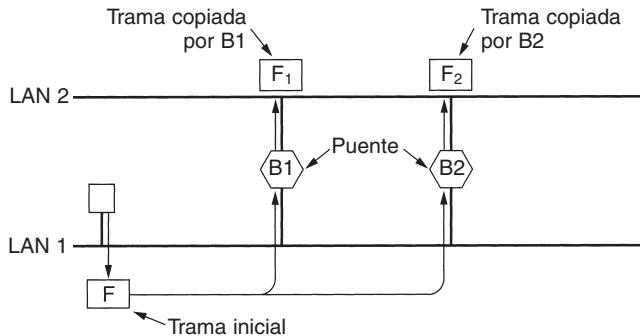


Figura 4-43. Dos puentes paralelos transparentes.

la copia a la LAN 1, lo cual genera a F_3 (no se muestra). De manera similar, el puente 2 copia a F_1 a la LAN 1 y genera a F_4 (tampoco se muestra). El puente 1 reenvía ahora a F_4 y el puente 2 copia a F_3 . Este ciclo se repite una y otra vez.

La solución a este problema es que los puentes se comuniquen entre sí y cubran la topología existente con un árbol de expansión que llegue a todas las LANs. En realidad, algunas conexiones potenciales entre LANs se ignoran en el afán de construir una topología ficticia libre de ciclos. Por ejemplo, en la figura 4-44(a) vemos nueve LANs interconectadas por diez puentes. Esta configuración se puede abstraer en un grafo con las LANs como nodos. Un arco conecta dos LANs que estén unidas por un puente. El grafo puede reducirse a un árbol de expansión eliminando los arcos que se muestran como líneas punteadas en la figura 4-44(b). Con este árbol de expansión existe exactamente una ruta desde cada LAN hasta las demás LANs. Una vez que los puentes se ponen de acuerdo en el árbol de expansión, todos los reenvíos entre LANs se hacen a través del árbol de expansión. Puesto que existe sólo una ruta de cada origen a cada destino, es imposible que se generen ciclos.

Para construir el árbol de expansión, los puentes primero tienen que escoger un puente que funcione como raíz del árbol. Toman esta decisión haciendo que cada uno difunda su número de serie, instalado por el fabricante y con garantía de ser único en el mundo. El puente con el menor número de serie se vuelve la raíz. A continuación, se construye un árbol con las rutas más cortas de la raíz a cada puente y LAN. Éste es el árbol de expansión. Si falla un puente o una LAN, se calcula un árbol nuevo.

El resultado de este algoritmo es que se establece una ruta única de cada LAN hasta la raíz y, por tanto, a todas las demás LANs. Aunque el árbol abarca todas las LANs, no necesariamente están presentes todos los puentes en el árbol (para evitar ciclos). Aun después de que se ha establecido el árbol de expansión, el algoritmo continúa operando a fin de detectar automáticamente cambios de topología y actualizar el árbol. El algoritmo distribuido que se usa para construir el árbol de expansión fue inventado por Radia Perlman y se describe en detalle en (Perlman, 2000). Se estandarizó en el IEEE 802.1D.

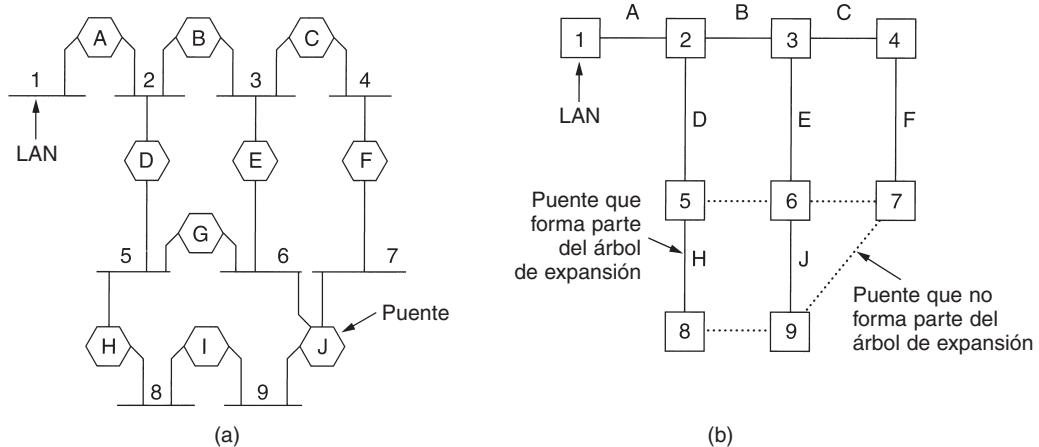


Figura 4-44. (a) LANs interconectadas. (b) Árbol de expansión que abarca las LANs. Las líneas punteadas no son parte del árbol de expansión.

4.7.4 Puentes remotos

Un uso común de los puentes es conectar dos (o más) LANs distantes. Por ejemplo, una empresa podría contar con plantas en varias ciudades, cada una con su propia LAN. En un plano ideal, todas las LANs deberían estar interconectadas de tal forma que funcionaran como una sola LAN grande.

Este objetivo se puede cumplir colocando un puente en cada LAN y conectando los puentes por pares con líneas punto a punto (por ejemplo, líneas alquiladas a una compañía telefónica). En la figura 4-45 se ilustra un sistema sencillo, con tres LANs. Aquí se aplican los algoritmos comunes de enrutamiento. La forma más sencilla de entender esto es considerar las tres líneas punto a punto como LANs sin *hosts*. A continuación tenemos un sistema normal de seis LANs interconectadas mediante cuatro puentes.

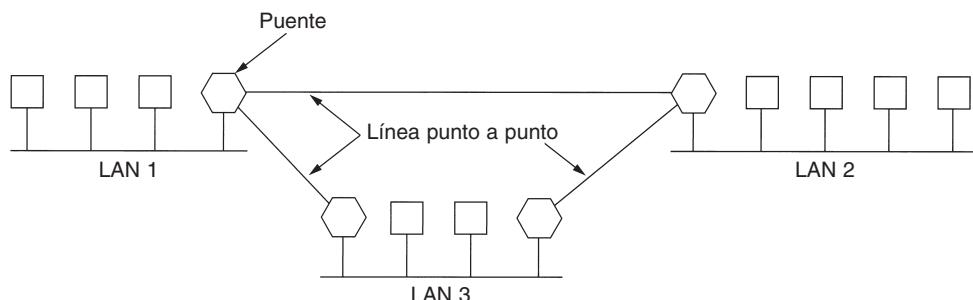


Figura 4-45. Los puentes remotos se pueden utilizar para interconectar LANs distantes.

En las líneas punto a punto se pueden utilizar diversos protocolos. Una opción es elegir algún protocolo de enlace de datos estándar de punto a punto como PPP y colocar tramas MAC completas en el campo de carga útil. Esta estrategia funciona mejor si todas las LANs son idénticas, y el único problema es conseguir que las tramas lleguen a la LAN correcta. Otra opción es eliminar tanto el encabezado como el terminador MAC en el puente de origen y agregar lo que queda en el campo de carga útil del protocolo de punto a punto. A continuación, en el puente de destino se pueden generar un nuevo encabezado y un nuevo terminador MAC. Una desventaja de este método consiste en que la suma de verificación que llega al *host* de destino no es la que se calculó en el *host* de origen, debido a lo cual tal vez no se detecten los errores ocasionados por bits defectuosos en la memoria de un puente.

4.7.5 Repetidores, concentradores, puentes, commutadores, enrutadores y puertas de enlace

Hasta este punto hemos visto una gran variedad de formas para desplazar tramas y paquetes de un segmento de cable a otro. Hemos mencionado repetidores, puentes, commutadores, concentradores, enrutadores y puertas de enlace. Todos estos dispositivos son de uso común, aunque difieren en formas sutiles y no tan sutiles. Puesto que son tantos, tal vez valga la pena analizarlos en conjunto para conocer sus similitudes y diferencias.

Para empezar, estos dispositivos operan en diferentes capas, como se muestra en la figura 4-46(a). La capa es importante porque los distintos dispositivos utilizan diferentes partes de información para decidir su modo de operación. En un escenario común, el usuario genera algunos datos que se enviarán a una máquina remota. Estos datos se pasan a la capa de transporte, que le agrega un encabezado, por ejemplo, un encabezado TCP, y pasa la unidad que resulta a la capa de red. Ésta incorpora su propio encabezado para obtener un paquete de capa de red, por ejemplo, un paquete IP. En la figura 4-46(b) podemos ver el paquete IP con un sombreado gris. A continuación, el paquete pasa a la capa de enlace de datos, que incorpora su propio encabezado y suma de verificación (CRC) y envía la trama resultante a la capa física para que desde ahí sea transmitida, por ejemplo, sobre una LAN.

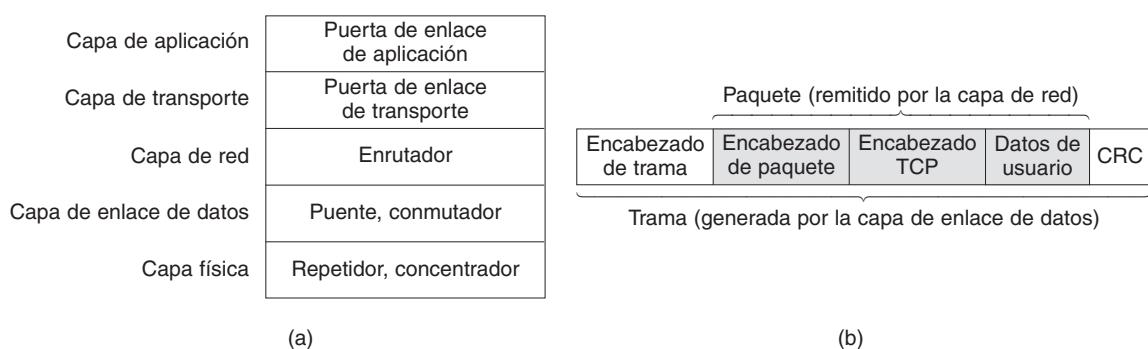


Figura 4-46. (a) Los dispositivos y sus capas correspondientes. (b) Tramas, paquetes y encabezados.

Ahora demos un vistazo a los dispositivos de commutación y veamos cómo se relacionan con los paquetes y las tramas. Al fondo, en la capa física, se encuentran los repetidores. Éstos son dispositivos análogos conectados a dos segmentos de cable. Una señal que aparece en uno de ellos es amplificada y enviada al otro. Los repetidores no distinguen entre tramas, paquetes o encabezados. Manejan voltios. Por ejemplo, la Ethernet tradicional admite cuatro repetidores, con el propósito de extender la longitud máxima de cable de 500 a 2500 metros.

Pasemos ahora a los concentradores. Un concentrador tiene numerosos puertos de entrada que une de manera eléctrica. Las tramas que llegan a cualquiera de las líneas se envían a todas las demás. Si dos tramas llegan al mismo tiempo, chocarán, al igual que en un cable coaxial. En otras palabras, el concentrador constituye un solo dominio de colisión. Todas las líneas que convergen en un concentrador deben operar a la misma velocidad. A diferencia de los repetidores, los concentradores (por lo general) no amplifican las señales entrantes y su diseño les permite contener varias tarjetas de línea con múltiples entradas, aunque las diferencias son ligeras. Al igual que los repetidores, los concentradores no examinan las direcciones 802 ni las utilizan de ninguna manera. En la figura 4-47(a) se muestra un concentrador.

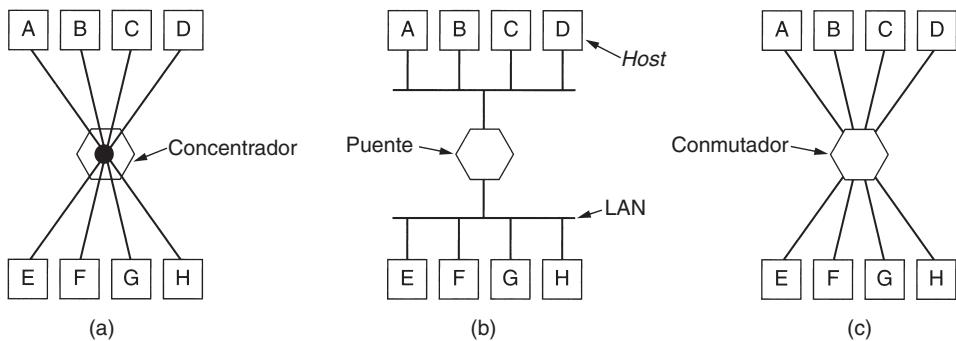


Figura 4-47. (a) Concentrador. (b) Puente. (c) Comutador.

Veamos a continuación la capa de enlace de datos donde operan los puentes y los comutadores. Ya hemos visto algo de los puentes. Un puente conecta dos o más LANs, como se puede ver en la figura 4-47(b). Cuando llega una trama, el software del puente extrae la dirección de destino del encabezado y la busca en una tabla para averiguar a dónde debe enviar la trama. En Ethernet, esta dirección es la dirección de destino de 48 bits que se muestra en la figura 4-17. De la misma manera que un concentrador, un puente moderno tiene tarjetas de línea, por lo general para cuatro u ocho puertos de entrada de un tipo determinado. Una tarjeta de línea para Ethernet no puede manejar tramas token ring debido a que no sabe dónde buscar la dirección de destino que viene en el encabezado de la trama. Sin embargo, un puente podría tener tarjetas de línea para diferentes tipos de red y diferentes velocidades. En contraste con un concentrador, en un puente cada puerto constituye su propio dominio de colisión.

Los comutadores son similares a los puentes en el aspecto de que ambos enrutan tomando como base las direcciones de las tramas. De hecho, mucha gente se refiere a ellos de manera indistinta. La principal diferencia consiste en que un comutador se utiliza con mayor frecuencia

para conectar computadoras individuales, como se puede ver en la figura 4-47(c). En consecuencia, cuando el *host A* de la figura 4-47(b) desea enviar una trama al *host B*, el puente toma la trama pero la descarta. Por el contrario, en la figura 4-47(c), el conmutador debe reenviar activamente la trama de *A* a *B* porque no existe otra forma para que ésta llegue ahí. Puesto que por lo general cada puerto del conmutador va hacia una sola computadora, éstos deben contar con espacio para muchas más tarjetas de línea que los puentes, cuyo propósito es conectar solamente LANs. Cada tarjeta de línea proporciona espacio de búfer para las tramas que llegan a sus puertos. Como cada puerto constituye su propio dominio de colisión, los conmutadores nunca pierden tramas por colisiones. Sin embargo, si las tramas llegan con mayor rapidez de la que pueden retransmitirse, el conmutador podría quedarse sin espacio de búfer y proceder a descartar tramas.

Para aliviar en parte este problema, los conmutadores modernos empiezan el reenvío de tramas tan pronto como llega el campo de encabezado del destino, antes de que el resto de la trama haya llegado (siempre y cuando el puerto de salida esté disponible, por supuesto). Estos conmutadores no utilizan la técnica de conmutación de almacenamiento y reenvío. En ocasiones se les menciona como **conmutadores cut-through**. Por lo general, este tipo de manejo se realiza por completo en hardware, en tanto que los puentes contienen tradicionalmente una CPU que realiza la conmutación de almacenamiento y reenvío en software. No obstante, debido a que todos los puentes y conmutadores modernos contienen circuitos integrados especiales para conmutación, la diferencia entre un conmutador y un puente es más un asunto de mercadotecnia que técnico.

Hasta aquí hemos visto repetidores y concentradores, que son bastante similares, así como puentes y conmutadores, que también son muy semejantes. Ahora pasaremos a los enrutadores, que son diferentes de todos los anteriores. Cuando un paquete llega a un enrutador, el encabezado y el terminador de la trama se eliminan y el paquete contenido en el campo de carga útil de la trama (sombreado en la figura 4-46) se pasa al software de enrutamiento. Este software se vale del encabezado del paquete para elegir un puerto de salida. En un paquete IP, el encabezado contendrá una dirección de 32 bits (IPv4) o 128 bits (IPv6), no una dirección 802 de 48 bits. El software de enrutamiento no analiza las direcciones de las tramas e incluso no sabe si el paquete proviene de una LAN o una línea punto a punto. En el capítulo 5 estudiaremos los enrutadores y el enrutamiento.

Una capa más arriba encontramos puertas de enlace de transporte. Estos dispositivos conectan dos computadoras que utilizan diferentes protocolos de transporte orientados a la conexión. Por ejemplo, imagine que una computadora que utiliza el protocolo TCP/IP orientado a la conexión necesita comunicarse con una computadora que emplea el protocolo de transporte ATM, también orientado a la conexión. La puerta de enlace de transporte puede copiar los paquetes de una conexión a la otra y darles el formato que necesiten.

Por último, las puertas de enlace de aplicación comprenden el formato y contenido de los datos y traducen los mensajes de un formato a otro. Por ejemplo, una puerta de enlace de correo electrónico puede traducir mensajes Internet en mensajes SMS para teléfonos móviles.

4.7.6 LANs virtuales

En los primeros días de las redes de área local, cables amarillos gruesos serpenteaban por los ductos de muchos edificios de oficinas. Conectaban a todas las computadoras por las que pasa-

ban. Con frecuencia había muchos cables, los cuales se conectaban a una red vertebral central (como en la figura 4-39) o a un concentrador central. No importaba cuál computadora pertenecía a cuál LAN. Todos los usuarios de oficinas cercanas se conectaban a la misma LAN aunque no estuvieran relacionados con ella. El aspecto geográfico se imponía al lógico.

Todo cambió con el surgimiento de 10Base-T y los concentradores en la década de 1990. El cableado de los edificios se renovó (a un costo considerable) para desechar todas las mangueras amarillas de jardín e instalar cables de par trenzado desde cada oficina hasta gabinetes centrales al final de cada pasillo o hasta salas centrales de máquinas, como se observa en la figura 4-48. Si el encargado del reemplazo del cableado era un visionario, se instalaba cable de par trenzado categoría 5; si era un simple administrador, se instalaba el cable telefónico (categoría 3) existente (que tenía que reemplazarse algunos años más tarde con la aparición de Fast Ethernet).

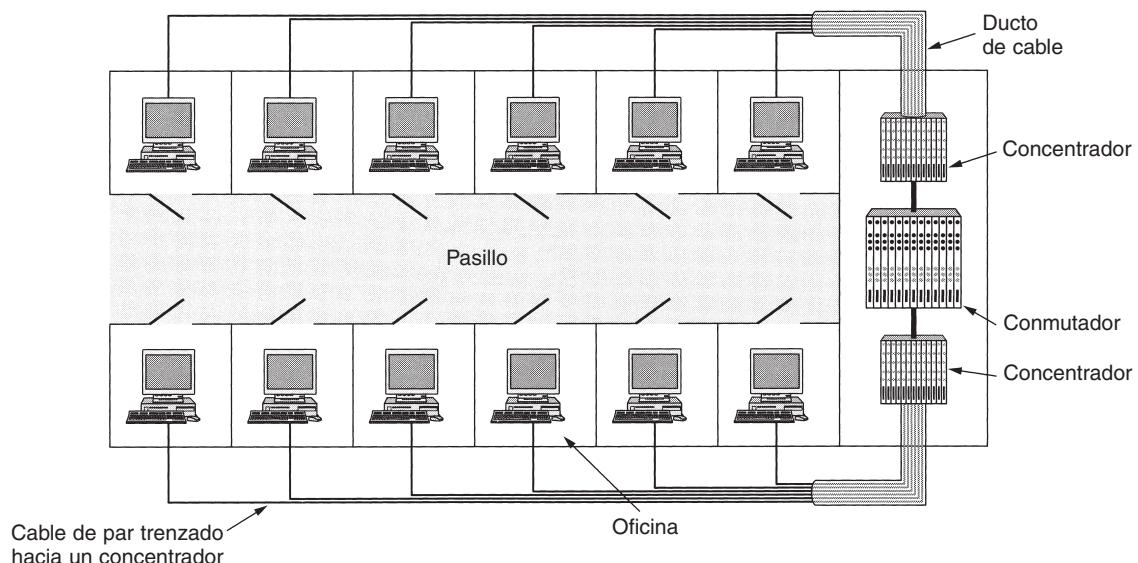


Figura 4-48. Edificio con cableado centralizado que utiliza concentradores y un commutador.

El uso de concentradores (y de commutadores posteriormente) con Ethernet hizo posible configurar las LANs con base en el aspecto lógico más que en el físico. Si una empresa necesita k LANs, compra k concentradores. Al elegir con cuidado qué conectores incorporar en qué concentradores, los usuarios de una LAN se pueden seleccionar de tal manera que tenga sentido para la organización, sin tomar mucho en cuenta el aspecto geográfico. Por supuesto, si dos personas del mismo departamento trabajan en diferentes edificios, es muy probable que estarán en diferentes concentradores y, en consecuencia, en diferentes LANs. No obstante, esto es mucho mejor que tener usuarios de una LAN con base totalmente en el aspecto geográfico.

¿Es importante quién está en qué LAN? Después de todo, en casi todas las organizaciones las LANs están interconectadas. Como veremos en breve, sí es importante. Por diversas razones, a los administradores de red les gusta agrupar a los usuarios en LANs para reflejar la estructura de la or-

ganización más que el diseño físico del edificio. Un aspecto es la seguridad. Cualquier interfaz de red se puede operar en modo promiscuo para que copie todo el tráfico que llegue. Muchos departamentos, como los de investigación, patentes y contabilidad, manejan información que no debe salir de los límites de sus respectivas áreas. En casos como éste se justifica que todos los usuarios de un departamento sean asignados a una sola LAN y que no se permita que el tráfico salga de ésta. A los directivos no les agrada escuchar que un arreglo de este tipo sólo es posible si todos los usuarios de un departamento están en oficinas adyacentes, sin más gente entre ellos.

Un segundo aspecto es la carga. Algunas LANs se utilizan mucho más que otras, y en ocasiones podría ser necesario separarlas. Por ejemplo, si los usuarios de investigaciones realizan toda clase de experimentos que en ocasiones se les van de las manos y saturan su LAN, tal vez a los usuarios de contabilidad no les agrade tener que ceder parte de su capacidad para ayudarles.

Un tercer aspecto es la difusión. La mayoría de las LANs soporta la difusión, y muchos protocolos de la capa superior utilizan ampliamente esta característica. Por ejemplo, cuando un usuario desea enviar un paquete a una dirección IP x , ¿cómo sabe a cuál dirección MAC enviar la trama? En el capítulo 5 estudiaremos este asunto, pero en pocas palabras, la respuesta es que debe difundir una trama con la pregunta: ¿Quién posee la dirección IP x ? y esperar la respuesta. Existen muchos más ejemplos del uso de la difusión. Conforme se interconectan más y más LANs, la cantidad de difusiones (*broadcasts*) que pasan por cada máquina se incrementa de manera lineal con el número de máquinas.

Las difusiones tienen el problema asociado de que de vez en cuando las interfaces de red se averían y empiezan a generar flujos interminables de tramas de difusión. El resultado de una **tormenta de difusión** es que 1) las tramas ocupan toda la capacidad de la LAN, y 2) las máquinas de todas las LANs interconectadas se atascan procesando y descartando las tramas difundidas.

A primera vista parecería que la magnitud de las tormentas de difusión podría limitarse separando las LANs con puentes o comutadores, pero si el objetivo es conseguir transparencia (es decir, que una máquina pueda cambiarse a una LAN distinta al otro lado del puente sin que nadie lo note), entonces los puentes tienen que reenviar las tramas difundidas.

Después de analizar por qué las empresas podrían requerir varias LANs con un alcance limitado, regresemos al problema de desacoplar la topología lógica de la física. Supongamos que un usuario es transferido de un departamento a otro de la misma empresa sin que se le cambie de oficina o que se le cambia de oficina pero no de departamento. En un entorno de concentradores con cables, cambiar al usuario a la LAN correcta implica que el administrador de la red debe desplazarse hasta el gabinete de cableado, quitar de un concentrador el conector de la máquina del usuario e insertar el mismo conector en otro concentrador.

En muchas empresas, los cambios organizacionales ocurren todo el tiempo, lo cual quiere decir que los administradores de sistemas desperdician mucho tiempo quitando y metiendo conectores de un lado a otro. Asimismo, en algunos casos el cambio no se puede realizar de ninguna manera porque el cable de par trenzado de la máquina del usuario está demasiado lejos del concentrador correcto (por ejemplo, en otro edificio).

En respuesta a la demanda de mayor flexibilidad por parte de los usuarios, los fabricantes de redes empezaron a trabajar en una forma de volver a cablear edificios completos mediante software.

El concepto que surgió se denomina **VLAN (LAN Virtual)** e incluso fue estandarizado por el comité 802. Ahora se encuentra funcionando en muchas organizaciones. Analicémoslo brevemente. Si desea información adicional, vea (Breyer y Riley, 1999, y Seifert, 2000).

Las VLANs se fundamentan en conmutadores especialmente diseñados para este propósito, aunque también podrían contar con algunos concentradores, como se muestra en la figura 4-48. Para configurar una red VLAN, el administrador de la red decide cuántas VLANs habrá, qué computadoras habrá en cuál VLAN y cómo se llamarán las VLANs. Es común nombrar mediante colores a las VLANs (de manera informal), ya que de esta manera es posible imprimir diagramas en color que muestren la disposición física de las máquinas, con los miembros de la LAN roja en rojo, los de la LAN verde en verde, etc. De esta forma, tanto el diseño físico como el lógico se pueden reflejar en un solo esquema.

Por ejemplo, considere las cuatro LANs de la figura 4-49(a), en la cual ocho de las máquinas pertenecen a la VLAN G (gris) y siete forman parte de la VLAN W (blanca). Dos puentes, *B1* y *B2*, conectan las cuatro LANs físicas. Si se utiliza cableado de par trenzado centralizado, también podría haber cuatro concentradores (que no se muestran), pero desde el punto de vista lógico un cable con múltiples derivaciones y un concentrador representan lo mismo. Al esquematizar la figura de esta forma se aprecia un poco menos amontonada. Asimismo, el término “puente” se emplea actualmente cuando hay varias máquinas en cada puerto, como es el caso de esta figura, pero de otra manera, “puente” y “comutador” en esencia son indistintos. En la figura 4-49(b) se muestran las mismas máquinas y las mismas VLANs, aunque en esta ocasión con conmutadores y una sola máquina en cada puerto.

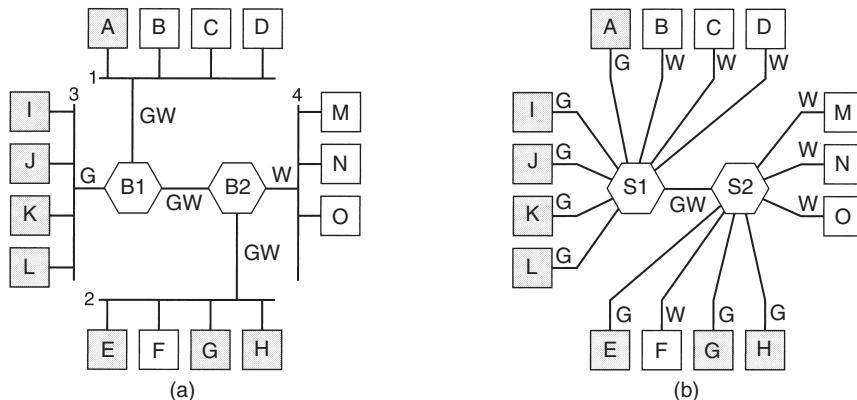


Figura 4-49. (a) Cuatro LANs físicas organizadas en dos VLANs, en gris y blanco, mediante dos puentes. (b) Las mismas 15 máquinas organizadas en dos VLANs mediante conmutadores.

Para que las VLANs funcionen correctamente, las tablas de configuración se deben establecer en los puentes o en los conmutadores. Estas tablas indican cuáles VLANs se pueden acceder a través de qué puertos (líneas). Cuando una trama llega procedente de, digamos, la VLAN gris, debe

reenviarse a todos los puertos G. De este modo se puede enviar tráfico ordinario (es decir, de unidifusión), así como de difusión y multidifusión.

Observe que un puerto puede marcarse con varios colores de VLAN. Esto se aprecia con más claridad en la figura 4-49(a). Suponga que la máquina A difunde una trama. El puente B1 la recibe y detecta que proviene de una máquina de la VLAN gris, por lo cual la reenvía a todos los puertos G (excepto al puerto del que procede). Puesto que B1 tiene sólo otros dos puertos y ambos son G, la trama se envía a ambos.

En B2 la situación es distinta. Aquí el puerto sabe que no hay máquinas grises en la LAN 4, por lo que no envía la trama ahí. Sólo la manda a la LAN 2. Si uno de los usuarios de la LAN 4 debe cambiar de departamento y trasladarse a la VLAN gris, entonces las tablas de B2 deben actualizarse y renombrar el puerto como GW en lugar de W. Si la máquina F cambia a gris, entonces el puerto de la LAN 2 debe renombrarse como G en lugar de GW.

Imaginemos ahora que todas las máquinas de las LANs 2 y 4 cambian a gris. En ese caso, no sólo los puertos de B2 para las LANs 2 y 4 se renombran como G, sino que también el puerto de B1 que va a B2 debe renombrarse como G en lugar de GW porque las tramas blancas que llegan a B1 procedentes de las LANs 1 y 3 ya no tienen que reenviarse a B2. En la figura 4-49(b) permanece la misma situación, sólo que aquí todos los puertos que van hacia una sola máquina se marcan con un solo color porque sólo hay una VLAN.

Hasta aquí hemos dado por sentado que los puentes y los conmutadores saben de alguna forma qué color tienen las tramas que llegan. ¿Cómo lo saben? Por medio de los tres métodos siguientes:

1. A cada puerto se le asigna un color de VLAN.
2. A cada dirección MAC se le asigna un color de VLAN.
3. A cada protocolo de la capa 3 o a cada dirección IP se le asigna un color de VLAN.

En el primer método, cada puerto se marca con un color de VLAN. Sin embargo, este método sólo funciona si todas las máquinas de un puerto pertenecen a la misma VLAN. En la figura 4-49(a), esta propiedad se aplica a B1 para el puerto de la LAN 3 pero no al puerto de la LAN 1.

En el segundo método, el puente o el conmutador tienen una tabla con las direcciones MAC de 48 bits de cada máquina conectada a ellos, junto con la VLAN a la cual pertenece la máquina. Bajo estas condiciones, es factible mezclar VLANs en una LAN física, como en el caso de la LAN 1 de la figura 4-49(a). Cuando llega una trama, todo lo que tienen que hacer el puente o el conmutador es extraer la dirección MAC y buscarla en una tabla para averiguar de qué VLAN proviene.

En el tercer método el puente o el conmutador examinan el campo de carga útil de la trama, por ejemplo, para clasificar todas las máquinas IP en una VLAN y todas las máquinas AppleTalk en otra. En el primer caso, la dirección IP se puede utilizar también para identificar a la máquina. Esta estrategia es más útil cuando varias máquinas son computadoras portátiles que se pueden acoplar en cualquiera de diversos lugares. Puesto que cada estación de acoplamiento tiene su propia dirección MAC, el solo hecho de saber cuál estación de acoplamiento se utilizó no indica en absoluto en cuál VLAN se encuentra la computadora portátil.

El único problema de este enfoque es que transgrede la regla más elemental de la conectividad: independencia de las capas. A la capa de enlace de datos no le incumbe lo que esté en el campo de carga útil. No le corresponde analizar la carga útil ni tomar decisiones con base en el contenido. Una consecuencia del uso de este enfoque es que un cambio en el protocolo de la capa 3 (por ejemplo, una actualización de IPv4 a IPv6) ocasiona que los conmutadores fallen repentinamente. Por desgracia, en el mercado hay conmutadores que funcionan de esta manera.

Por supuesto, no hay nada de malo en enrutar con base en las direcciones IP —casi todo el capítulo 5 está dedicado al enrutamiento IP— pero al mezclar las capas se pueden propiciar problemas. Un fabricante de conmutadores podría minimizar esta situación argumentando que sus conmutadores comprenden tanto IPv4 como IPv6, así que no hay problema. ¿Pero qué pasará cuando surja IPv7? En tal caso, el fabricante tal vez dirá: Compre nuevos conmutadores, ¿cuál es el problema?

El estándar IEEE 802.1Q

Al ahondar un poco más en este asunto salta a la vista que lo importante es la VLAN de la trama, no la VLAN de la máquina emisora. Si hubiera alguna forma de identificar la VLAN en el encabezado de la trama, se desvanecería la necesidad de examinar la carga útil. Para una LAN nueva como 802.11 u 802.16 habría sido bastante fácil tan sólo agregar un campo de VLAN en el encabezado. De hecho, el campo *Identificador de conexión* del estándar 802.16 es muy parecido a un identificador VLAN. ¿Pero qué se puede hacer con Ethernet, que es la LAN dominante y no tiene campos disponibles para el identificador VLAN?

El comité IEEE 802 se enfrentó a este problema en 1995. Después de muchas discusiones, hizo lo impensable y cambió el encabezado de Ethernet. El nuevo formato se publicó en el estándar **802.1Q** del IEEE, emitido en 1998. El nuevo formato contiene una etiqueta VLAN; en breve la examinaremos. No es de sorprender que el cambio de algo ya bien establecido como el encabezado de Ethernet no sea nada sencillo. Algunas de las preguntas que surgen son:

1. ¿Tenemos que tirar a la basura cientos de millones de tarjetas Ethernet existentes?
2. Si no es así, ¿quién generará los nuevos campos?
3. ¿Qué sucederá con las tramas que ya tienen el tamaño máximo?

Por supuesto, el comité 802 estaba consciente de estos problemas y tenía que encontrar soluciones, lo cual hizo.

La clave para la solución consiste en comprender que los campos VLAN sólo son utilizados por los puentes y los conmutadores, no por las máquinas de los usuarios. De ahí que en la figura 4-49 no sea realmente necesario que estén presentes en las líneas que van hacia las estaciones finales siempre y cuando se encuentren en la línea entre los puentes o los conmutadores. Así, para utilizar VLANs, los puentes o los conmutadores deben tener soporte para VLAN, pero ese ya era un requisito. Ahora sólo estamos agregando el requisito adicional de que tengan soporte para 802.1Q, requisito que los nuevos ya cubren.

Respecto a la cuestión de si es necesario desechar todas las tarjetas Ethernet existentes, la respuesta es no. Recuerde que el comité 802.3 no pudo conseguir que la gente cambiara el campo *Tipo* por un campo *Longitud*. Ya podrá imaginar la reacción ante el anuncio de que todas las tarjetas Ethernet existentes tuvieran que desecharse. Sin embargo, se espera que las nuevas tarjetas Ethernet que salgan al mercado tendrán compatibilidad con el 802.1Q y llenarán correctamente el campo VLAN.

Por lo tanto, si el emisor no generará los campos VLAN, ¿quién lo hará? La respuesta es que el primer puente o conmutador con soporte de VLAN en recibir una trama los agregará y el último que los reciba los eliminará. ¿Pero cómo sabrán cuál trama corresponde a cuál VLAN? Bueno, el primer puente o conmutador podría asignar un número de VLAN a un puerto, analizar la dirección MAC o (¡Dios no lo quiera!) examinar la carga útil. Mientras todas las tarjetas Ethernet no se apeguen al estándar 802.1Q, estaremos en donde empezamos. La esperanza real es que todas las tarjetas Gigabit Ethernet se apegarán a 802.1Q desde el principio y que en tanto la gente se actualiza a Gigabit Ethernet, el 802.1Q se introducirá automáticamente. En cuanto al problema de las tramas mayores a 1518 bytes, el 802.1Q tan sólo incrementó el límite a 1522 bytes.

Durante el proceso de transición, muchas instalaciones tendrán algunas máquinas heredadas (en su mayoría, clásicas o Fast Ethernet) que no soportarán VLAN y otras (por lo general, Gigabit Ethernet) que sí lo harán. Esta situación se ilustra en la figura 4-50, en donde los símbolos sombreados representan máquinas que soportan VLAN y los vacíos no las soportan. Por simplicidad, damos por sentado que todos los conmutadores soportan VLAN. Si no es así, el primer conmutador que soporte VLAN puede incorporar las etiquetas con base en las direcciones MAC o IP.

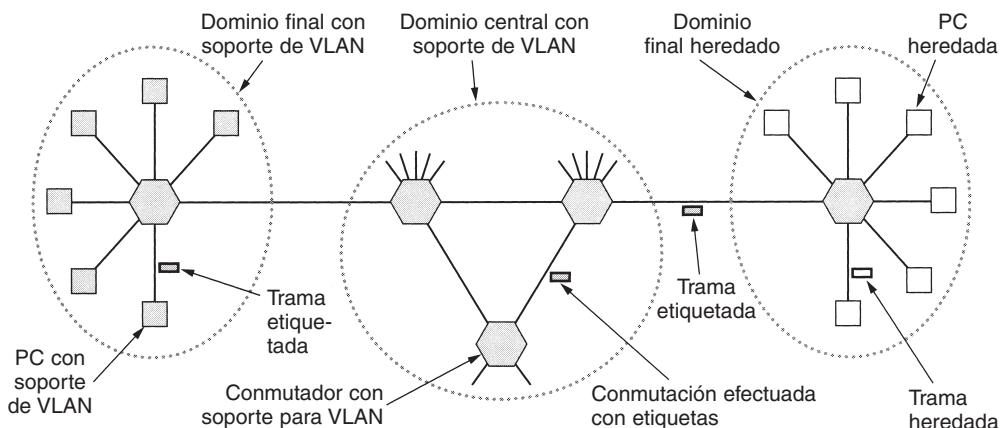


Figura 4-50. Transición de Ethernet heredado a Ethernet con soporte para VLAN. Los símbolos sombreados representan soporte para VLAN, a diferencia de los vacíos.

En esta figura, las tarjetas Ethernet con soporte para VLAN generan directamente tramas etiquetadas (es decir, 802.1Q) y la commutación posterior se vale de estas etiquetas. Para realizar esta commutación, los conmutadores tienen que saber cuáles VLANs están al alcance en cada puerto, lo mismo que antes. El hecho de saber que una trama pertenece a la VLAN gris no es de mucha

ayuda sino hasta que el conmutador sabe cuáles puertos tienen conexión con las máquinas de la VLAN gris. De esta forma, el conmutador necesita una tabla indexada por VLAN que le indique cuáles puertos puede utilizar y si éstos tienen soporte para VLAN o son heredados.

Cuando una PC heredada envía una trama a un conmutador con soporte para VLAN, el conmutador genera una trama etiquetada apoyándose en el conocimiento que tiene de la VLAN del emisor (utilizando el puerto, la dirección MAC o la dirección IP). De ahí en adelante, no importa que el emisor sea una máquina heredada. Asimismo, un conmutador que necesita entregar una trama etiquetada a una máquina heredada tiene que darle a la trama el formato heredado antes de entregarla.

Demos ahora un vistazo al formato de la trama 802.1Q, que se muestra en la figura 4-51. El único cambio es la adición de un par de campos de dos bytes. El primero es la *ID del protocolo de VLAN*. Siempre tiene el valor 0x8100. Como esta cifra es mayor que 1500, todas las tarjetas Ethernet lo interpretan como un tipo más que como una longitud. Lo que una tarjeta heredada hace con una trama como ésta es discutible porque dichas tramas no deberían enviarse a tarjetas heredadas.

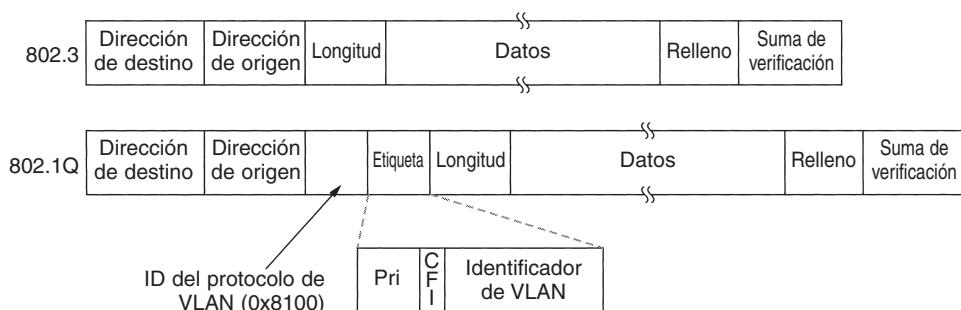


Figura 4-51. Formatos de trama Ethernet 802.3 (heredada) y 802.1Q.

El segundo campo de dos bytes contiene tres subcampos. El principal es el *Identificador de VLAN*, que ocupa los 12 bits de orden menor. Éste es el punto central de la cuestión: ¿a qué VLAN pertenece la trama? El campo *Prioridad* de 3 bits no tiene absolutamente nada que ver con las VLANs, pero como el cambio del encabezado Ethernet es un suceso poco frecuente que tarda tres años y ocupa a un ciento de personas, ¿por qué no incorporarle algunas otras cosas buenas en el proceso? Este campo permite distinguir el tráfico en tiempo real estricto del tráfico en tiempo real flexible y del tráfico no sensible al retardo, con el propósito de ofrecer una mejor calidad de servicio sobre Ethernet. Esto es necesario para el transporte de voz sobre Ethernet (aunque, para ser justos, IP tiene un campo similar desde hace un cuarto de siglo y nadie lo utiliza).

El último bit, *CFI (Indicador del Formato Canónico)*, debió haberse llamado *CEI (Indicador del Ego Corporativo)*. Su propósito original era indicar las direcciones MAC *little endian* en comparación con las *big endian*, pero esto ha cambiado con el tiempo. Actualmente indica que la carga útil contiene una trama 802.5 congelada-seca que se espera encuentre otra LAN 802.5 en el destino cuando se transmita a través de Ethernet. Por supuesto, este arreglo no tiene absolutamente

nada que ver con las VLANs. Pero la política de los comités de estándares no difiere mucho de la política común: si votas por mi bit, votaré por el tuyo.

Como ya mencionamos, cuando una trama etiquetada llega a un conmutador con soporte para VLAN, éste utiliza el ID de la VLAN como índice de tabla para averiguar a cuáles puertos enviar la trama. ¿Pero de dónde proviene la tabla? Si se elabora en forma manual, tenemos que empezar desde cero: la configuración manual de los puentes. La ventaja de los puentes transparentes es que son *plug and play* y no requieren configuración manual. Sería un terrible retroceso perder esa propiedad. Por fortuna, los puentes con soporte para VLAN se pueden autoconfigurar al observar las etiquetas que arriben a ellos. Si una trama etiquetada como VLAN 4 llega al puerto 3, entonces aparentemente una máquina en el puerto 3 se encuentra en la VLAN 4. El estándar 802.1Q explica cómo construir las tablas de manera dinámica, en su mayor parte referenciando porciones apropiadas del algoritmo de Perlman estandarizado en el 802.1D.

Antes de abandonar el tema del enrutamiento para VLAN, vale la pena hacer una última observación. Mucha gente de Internet y Ethernet es fanática de las redes no orientadas a la conexión y se oponen terminantemente a todo lo que huele a conexiones en las capas de enlace de datos y de red. No obstante, las VLANs incluyen un aspecto que es sorprendentemente similar a una conexión. Para utilizar las VLANs de manera apropiada, cada trama lleva un identificador especial nuevo que se utiliza como índice en una tabla dentro del conmutador para averiguar el destino al que se debe enviar la trama. Esto es precisamente lo que se hace en las redes orientadas a la conexión. En las redes no orientadas a la conexión, la dirección de destino es la que se utiliza para el enrutamiento, no un tipo de identificador de conexión. En el capítulo 5 abundaremos en este conexionismo gradual.

4.8 RESUMEN

Algunas redes tienen un solo canal que se usa para todas las comunicaciones. En estas redes, el aspecto clave del diseño es la asignación del canal entre las estaciones competidoras que desean usarlo. Se han desarrollado muchos algoritmos de asignación de canal. En la figura 4-52 se presenta un resumen de algunos de los métodos de asignación de canal más importantes.

Los métodos de asignación más sencillos son la FDM y la TDM; son eficientes con un número de estaciones pequeño y fijo y tráfico continuo. Ambos esquemas se usan ampliamente en estas circunstancias; por ejemplo, para dividir el ancho de banda de las troncales telefónicas.

Con un número grande y variable de estaciones, o con un tráfico en ráfagas, la FDM y la TDM son soluciones pobres. Se ha propuesto como alternativa el protocolo ALOHA, con y sin ranuras y control. El ALOHA y sus muchas variantes y derivaciones han sido ampliamente estudiados, analizados y usados en sistemas reales.

Cuando puede detectarse el estado del canal, las estaciones pueden evitar el comienzo de una transmisión mientras otra estación está transmitiendo. Esta técnica, la detección de portadora, ha producido varios protocolos que pueden usarse en LANs y MANs.

Método	Descripción
FDM	Dedica una banda de frecuencia a cada estación
WDM	Esquema FDM dinámico para fibra
TDM	Dedica una ranura de tiempo a cada estación
ALOHA puro	Transmisión asíncrona en cualquier momento
ALOHA ranurado	Transmisión aleatoria en ranuras de tiempo bien definidas
CSMA persistente-1	Acceso múltiple con detección de portadora estándar
CSMA no persistente	Retardo aleatorio cuando se detecta que el canal está ocupado
CSMA persistente-p	CSMA, pero con una probabilidad de persistencia p
CSMA/CD	CSMA, pero aborta al detectar una colisión
Mapa de bits	Calendarización <i>round robin</i> mediante mapa de bits
Conteo descendente binario	La estación disponible con el número más alto toma el turno
Recorrido de árbol	Contención reducida mediante habilitación selectiva
MACA, MACAW	Protocolos de LAN inalámbrica
Ethernet	CSMA/CD con retraso exponencial binario
FHSS	Espectro disperso con salto de frecuencia
DSSS	Espectro disperso de secuencia directa
CSMA/CA	Acceso múltiple con detección de portadora y evitación de colisiones

Figura 4-52. Métodos de asignación de canal y sistemas para canal común.

Se conoce una clase de protocolos que eliminan por completo la contención, o cuando menos la reducen considerablemente. El conteo binario descendente elimina por completo la contención. El protocolo de recorrido de árbol la reduce dividiendo dinámicamente las estaciones en dos grupos separados, uno que puede transmitir y otro que no. Se intenta hacer la división de tal manera que sólo una estación lista para transmitir pueda hacerlo.

Las LANs inalámbricas tienen sus propios problemas y soluciones. El problema principal lo causan las estaciones ocultas, por lo que el CSMA no funciona. Una clase de soluciones, tipificadas por MACA y MACAW, intenta estimular las transmisiones en las cercanías del destino, para hacer que el CSMA funcione mejor. También se usan el espectro disperso con salto de frecuencia y el espectro disperso de secuencia directa. El IEEE 802.11 combina CSMA y MACAW para producir CSMA/CA.

Ethernet predomina en el campo de las redes de área local. Utiliza CSMA/CD para la asignación de canal. Las primeras versiones empleaban un cable que serpenteaba entre las máquinas, pero en la actualidad son más comunes los cables de par trenzado hacia concentradores y conmutadores. Las velocidades se han incrementado de 10 Mbps a 1 Gbps y siguen en aumento.

Las LANs inalámbricas se están popularizando, y el 802.11 domina el campo. Su capa física permite cinco diferentes modos de transmisión, entre ellos el infrarrojo, diversos esquemas de

espectro disperso y un sistema FDM multicanal con una estación base en cada celda, aunque también puede funcionar sin ninguna. El protocolo es una variante de MACAW, con detección de portadora virtual.

Las MANs inalámbricas están empezando a aparecer. Son sistemas de banda amplia que utilizan radio para reemplazar la última milla en conexiones telefónicas. También utilizan técnicas tradicionales de modulación de banda estrecha. La calidad de servicio es importante, y el estándar 802.16 define cuatro clases (tasa de bits constante, dos tasas variables de bits y una de mejor esfuerzo).

El sistema Bluetooth también es inalámbrico, aunque está más enfocado a los sistemas de escritorio, para conectar diademas telefónicas y otros periféricos a las computadoras sin necesidad de cables. También se utiliza para conectar periféricos, como máquinas de fax, a los teléfonos móviles. Al igual que el 801.11, utiliza espectro disperso con saltos de frecuencia en la banda ISM. Debido al nivel de ruido esperado en muchos entornos y a la necesidad de interacción en tiempo real, sus diversos protocolos incorporan una sofisticada corrección de errores hacia delante.

Con tantas LANs diferentes, es necesaria una forma para interconectarlas. Los puentes y los conmutadores tienen este propósito. El algoritmo de árbol de expansión se utiliza para construir puentes *plug and play*. La VLAN es un nuevo desarrollo del mundo de la interconexión de LANs, que separa la topología lógica de la topología física de las LANs. Se ha introducido un nuevo formato para las tramas Ethernet (802.1Q), cuyo propósito es facilitar la utilización de las VLANs en las organizaciones.

PROBLEMAS

1. Para este problema, utilice una fórmula de este capítulo, pero primero enúnciela. Las tramas arriban de manera aleatoria a un canal de 100 Mbps para su transmisión. Si el canal está ocupado cuando arriba una trama, ésta espera su turno en una cola. La longitud de la trama se distribuye exponencialmente con una media de 10,000 bits/trama. Para cada una de las siguientes tasas de llegada de tramas, dé el retardo experimentado por la trama promedio, incluyendo tanto el tiempo de encolamiento como el de transmisión.
 - (a) 90 tramas/seg.
 - (b) 900 tramas/seg.
 - (c) 9000 tramas/seg.
2. Un grupo de N estaciones comparte un canal ALOHA puro de 56 kbps. La salida de cada estación es una trama de 1000 bits en promedio cada 100 seg aun si la anterior no ha sido enviada (por ejemplo, las estaciones pueden almacenar en búfer las tramas salientes). ¿Cuál es el valor máximo de N ?
3. Considere el retardo del ALOHA puro comparándolo con el ALOHA ranurado cuando la carga es baja. ¿Cuál es menor? Explique su respuesta.
4. Diez mil estaciones de reservaciones de una aerolínea compiten por un solo canal ALOHA ranurado. La estación promedio hace 18 solicitudes/hora. Una ranura dura 125 μ seg. ¿Cuál es la carga aproximada total del canal?

5. Una gran población de usuarios de ALOHA genera 50 solicitudes/seg incluidas tanto originales como retransmisiones. El tiempo se divide en ranuras de 40 mseg.
 - (a) ¿Cuál es la oportunidad de éxito en el primer intento?
 - (b) ¿Cuál es la probabilidad exacta de k colisiones y después tener éxito?
 - (c) ¿Cuál es el número esperado de intentos de transmisión necesarios?
6. Mediciones en un canal ALOHA ranurado con una cantidad infinita de usuarios muestra que 10% de las ranuras están inactivas.
 - (a) ¿Qué carga, G , tiene el canal?
 - (b) ¿Cuál es la velocidad real de transporte?
 - (c) ¿El canal está subcargado o sobrecargado?
7. En un sistema ALOHA ranurado de población infinita, la cantidad media de ranuras que espera una estación entre una colisión y su retransmisión es de 4. Grafique la curva de retardo contra velocidad real de transporte de este sistema.
8. ¿Cuánto debe esperar una estación, s , en el peor de los casos, antes de empezar a transmitir su trama sobre una LAN que utiliza
 - (a) el protocolo básico de mapa de bits?
 - (b) el protocolo de Mok y Ward con cambio de números virtuales de estación?
9. Una LAN usa la versión de Mok y Ward del conteo descendente binario. En cierto momento, las 10 estaciones tienen los números de estación virtual 8, 2, 4, 5, 1, 7, 3, 6, 9 y 0. Las tres estaciones siguientes que van a enviar son: 4, 3 y 9, en ese orden. ¿Cuáles son los nuevos números de estación virtual una vez que las tres han terminado sus transmisiones?
10. Dieciséis estaciones contienen por un canal compartido que usa el protocolo de recorrido de árbol. Si todas las estaciones cuyas direcciones son números primos de pronto quedaran listas al mismo tiempo, ¿cuántas ranuras de bits se necesitan para resolver la contención?
11. Un conjunto de 2^n estaciones usa el protocolo de recorrido de árbol adaptable para arbitrar el acceso a un cable compartido. En cierto momento, dos de ellas quedan listas. ¿Cuál es el número de ranuras mínimo, máximo y medio para recorrer el árbol si $2^n \gg 1$?
12. Las LANs inalámbricas que estudiamos usaban protocolos como MACA en lugar de CSMA/CD. ¿En qué condiciones sería posible usar CSMA/CD?
13. ¿Qué propiedades tienen en común los protocolos de acceso a canal WDMA y GSM? Consulte el GSM en el capítulo 2.
14. Seis estaciones, de A a F , se comunican mediante el protocolo MACA. ¿Es posible que dos transmisiones tengan lugar de manera simultánea? Explique su respuesta.
15. Un edificio de oficinas de siete pisos tiene 15 oficinas adyacentes por piso. Cada oficina contiene un enchufe de pared para una terminal en la pared frontal, por lo que los enchufes forman una retícula triangular en el plano vertical, con una separación de 4 m entre enchufes, tanto vertical como horizontalmente. Suponiendo que es factible tender un cable recto entre cualquier par de enchufes, horizontal, vertical o diagonalmente, ¿cuántos metros de cable se necesitan para conectar todos los enchufes usando
 - (a) una configuración en estrella con un solo enrutador en medio?
 - (b) una LAN 802.3?

16. ¿Cuál es la tasa de baudios de la Ethernet de 10 Mbps estándar?
17. Bosqueje la codificación Manchester para el flujo de bits: 0001110101.
18. Bosqueje la codificación diferencial Manchester para el flujo de bits del problema anterior. Suponga que la línea se encuentra inicialmente en el estado bajo.
19. Una LAN CSMA/CD (no la 802.3) de 10 Mbps y 1 km de largo tiene una velocidad de propagación de 200 m/ μ seg. En este sistema no se permiten los repetidores. Las tramas de datos tienen 256 bits de longitud, incluidos 32 bits de encabezado, suma de verificación y un poco más de sobrecarga. La primera ranura de bits tras una transmisión exitosa se reserva para que el receptor capture el canal y envíe una trama de confirmación de recepción de 32 bits. ¿Cuál es la tasa de datos efectiva, excluyendo la sobrecarga, suponiendo que no hay colisiones?
20. Dos estaciones CSMA/CD intentan transmitir archivos grandes (multitrama). Tras el envío de cada trama, contienden por el canal usando el algoritmo de retroceso exponencial binario. ¿Cuál es la probabilidad de que la contención termine en la ronda k , y cuál es la cantidad media de rondas por periodo de contención?
21. Considere la construcción de una red CSMA/CD que opere a 1 Gbps a través de un cable de 1 km de longitud sin repetidores. La velocidad de la señal en el cable es de 200,000 km/seg. ¿Cuál es el tamaño mínimo de trama?
22. Un paquete IP que se transmitirá a través de Ethernet tiene 60 bytes de longitud, incluyendo todos los encabezados. Si no se utiliza LLC, ¿se necesita relleno en la trama Ethernet, y de ser así, cuántos bytes?
23. Las tramas Ethernet deben tener al menos 64 bytes de longitud para asegurar que el transmisor permanezca en línea en caso de que ocurra una colisión en el extremo más lejano del cable. Fast Ethernet tiene el mismo tamaño mínimo de trama de 64 bytes pero puede recibir los bits diez veces más rápido. ¿Cómo es posible mantener el mismo tamaño mínimo de trama?
24. Algunos libros citan que el tamaño máximo de una trama Ethernet es de 1518 bytes en lugar de 1500. ¿Están en un error? Explique su respuesta.
25. La especificación 1000Base-SX indica que el reloj debe correr a 1250 MHz, aun cuando se supone que Gigabit Ethernet funciona a 1 Gbps. ¿Esta velocidad más alta confiere un margen adicional de seguridad? Si no es así, ¿qué está pasando?
26. ¿Cuántas tramas por segundo puede manejar Gigabit Ethernet? Reflexione con cuidado y tome en cuenta todos los casos relevantes. *Sugerencia:* es importante el hecho de que se trata de *Gigabit* Ethernet.
27. Mencione dos redes que permitan empaquetar tramas una tras otra. ¿Por qué es importante esta característica?
28. En la figura 4.27 se muestran cuatro estaciones, A , B , C y D . ¿Cuál de las dos últimas estaciones cree que está más cerca de A y por qué?
29. Suponga que una LAN 802.11b de 11 Mbps transmite tramas de 64 bytes una tras otra sobre un canal de radio con una tasa de error de 10^{-7} . ¿Cuántas tramas por segundo en promedio resultarán dañadas?
30. Una red 802.16 tiene un ancho de canal de 20 MHz. ¿Cuántos bits/seg se pueden enviar a una estación suscrita?

31. El IEEE 802.16 soporta cuatro clases de servicio. ¿Cuál es la mejor clase de servicio para enviar vídeo sin comprimir?
32. Dé dos razones por las cuales las redes podrían usar un código de corrección de errores en lugar de detección de errores y retransmisión.
33. En la figura 4-35 podemos ver que un dispositivo Bluetooth puede estar en dos *piconets* al mismo tiempo. ¿Hay alguna razón por la cual un dispositivo no pueda fungir como maestro en ambas al mismo tiempo?
34. La figura 4-25 muestra varios protocolos de capa física. ¿Cuál de éstos está más cercano al protocolo de capa física Bluetooth? ¿Cuál es la principal diferencia entre ambos?
35. Bluetooth soporta dos tipos de enlaces entre un maestro y un esclavo. ¿Cuáles son y para qué se utiliza cada uno?
36. Las tramas de *beacon* en el espectro disperso con salto de frecuencia, variante del 802.11, contienen el tiempo de permanencia. ¿Cree que las tramas de *beacon* análogas de Bluetooth también contienen tiempo de permanencia? Explique su respuesta.
37. Considere las LANs interconectadas que se muestran en la figura 4-44. Suponga que los *hosts* *a* y *b* se encuentran en la LAN 1, *c* está en la LAN 2 y *d* está en la LAN 8. En principio, las tablas de *hash* de todos los puentes están vacías y se utiliza el árbol de expansión que se muestra en la figura 4-44(b). Muestre la manera en que cambian las tablas de *hash* de los diversos puentes después de que cada uno de los siguientes sucesos ocurren en secuencia, primero (a) y a continuación (b), y así sucesivamente.
 - (a) *a* envía a *d*.
 - (b) *c* envía a *a*.
 - (c) *d* envía a *c*.
 - (d) *d* pasa a la LAN 6.
 - (e) *d* envía a *a*.
38. Una consecuencia del uso de un árbol de expansión para reenviar tramas en una LAN extendida es que algunos puentes tal vez no participen en absoluto en el reenvío de tramas. Identifique tres puentes que se encuentren en esta situación en la figura 4-44. ¿Hay alguna razón para conservar estos puentes, aun cuando no se utilicen para el reenvío?
39. Suponga que un conmutador tiene tarjetas de línea para cuatro líneas de entrada. Con frecuencia, una trama que llega en una de las líneas tiene que salir en otra línea de la misma tarjeta. ¿A qué decisiones se enfrenta el diseñador del conmutador como resultado de esta situación?
40. Un conmutador diseñado para Fast Ethernet tiene una tarjeta madre que puede transportar 10 Gbps. ¿Cuántas tramas/seg puede manejar en el peor de los casos?
41. Considere la red de la figura 4-49(a). Si la máquina *J* tuviera que volverse blanca repentinamente, ¿serían necesarios cambios para el etiquetado? Si es así, ¿cuáles?
42. Describa brevemente la diferencia entre los conmutadores de almacenamiento y reenvío y los *cut-through*.
43. Los conmutadores de almacenamiento y reenvío tienen una ventaja sobre los *cut-through* en relación con las tramas dañadas. Explique cuál es.

44. Las tablas de configuración son necesarias en los commutadores y los puentes para que las VLANs funcionen. ¿Qué pasaría si las VLANs de la figura 4-49(a) utilizaran concentradores en vez de cables con múltiples derivaciones? ¿Los concentradores también necesitarían tablas de configuración? ¿Por qué sí o por qué no?
45. En la figura 4-50 el commutador del dominio final heredado en la parte derecha tiene soporte para VLAN. ¿Sería posible utilizar ahí un commutador heredado? Si es así, ¿cómo funcionaría? En caso contrario, ¿por qué no?
46. Escriba un programa para simular el comportamiento del protocolo CSMA/CD sobre Ethernet cuando hay N estaciones listas para transmitir en el momento en que se está transmitiendo una trama. El programa deberá informar las veces que cada estación inicia exitosamente el envío de su trama. Suponga que un pulso de reloj ocurre una vez cada ranura de tiempo (51.2 microsegundos) y que la detección de una colisión y el envío de una secuencia atorada tarda una ranura de tiempo. Todas las tramas tienen la longitud máxima permitida.

5

LA CAPA DE RED

La capa de red se encarga de llevar los paquetes desde el origen hasta el destino. Llegar al destino puede requerir muchos saltos por enrutadores intermedios. Esta función ciertamente contrasta con la de la capa de enlace de datos, que sólo tiene la meta modesta de mover tramas de un extremo del cable al otro. Por lo tanto, la capa de red es la capa más baja que maneja la transmisión de extremo a extremo.

Para lograr su cometido, la capa de red debe conocer la topología de la subred de comunicación (es decir, el grupo de enrutadores) y elegir las rutas adecuadas a través de ella; también debe tener cuidado al escoger las rutas para no sobrecargar algunas de las líneas de comunicación y los enrutadores y dejar inactivos a otros. Por último, cuando el origen y el destino están en redes diferentes, ocurren nuevos problemas. La capa de red es la encargada de solucionarlos. En este capítulo estudiaremos todos estos temas y los ilustraremos principalmente valiéndonos de Internet y de su protocolo de capa de red, IP, aunque también veremos las redes inalámbricas.

5.1 ASPECTOS DE DISEÑO DE LA CAPA DE RED

En las siguientes secciones presentaremos una introducción a algunos de los problemas que deben enfrentar los diseñadores de la capa de red. Estos temas incluyen el servicio proporcionado a la capa de transporte y el diseño interno de la subred.

5.1.1 Conmutación de paquetes de almacenamiento y reenvío

Antes de iniciar una explicación sobre los detalles de la capa de red, probablemente valga la pena volver a exponer el contexto en el que operan los protocolos de esta capa. En la figura 5-1 se muestra dicho contexto. Los componentes principales del sistema son el equipo de la empresa portadora (enrutadores conectados mediante líneas de transmisión), que se muestra dentro del óvalo sombreado, y el equipo del cliente, que se muestra fuera del óvalo. El *host* *H1* está conectado de manera directa al enrutador de una empresa portadora, *A*, mediante una línea alquilada. En contraste, *H2* se encuentra en una LAN con un enrutador, *F*, el cual es propiedad de un cliente, quien lo maneja. Este enrutador también tiene una línea alquilada hacia el equipo de la empresa portadora. Mostramos *F* fuera del óvalo porque no pertenece a la empresa portadora, pero en términos de construcción, software y protocolos, tal vez no sea diferente de los enrutadores de la empresa portadora. Si bien es debatible el hecho de que este enrutador pertenezca a la subred, para los propósitos de este capítulo, los enrutadores del cliente son considerados como parte de la subred porque se valen de los mismos algoritmos que los enrutadores de la empresa portadora (y nuestro interés principal aquí son los algoritmos).

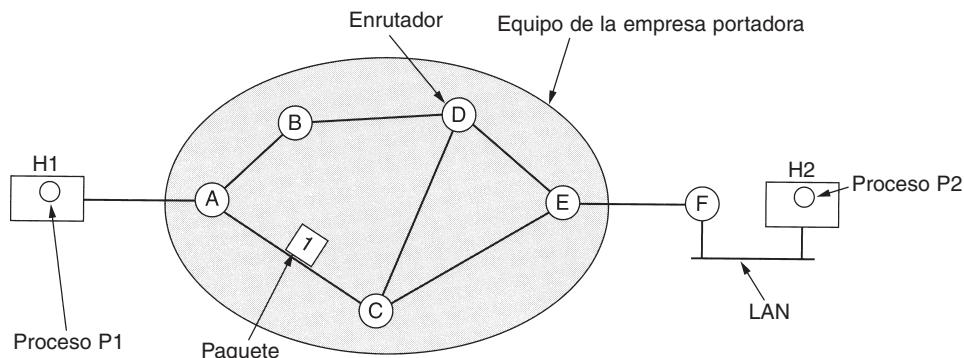


Figura 5-1. El entorno de los protocolos de la capa de red.

Este equipo se utiliza como sigue. Un *host* transmite al enrutador más cercano un paquete que tiene por enviar, ya sea en su propia LAN o a través de un enlace punto a punto con la empresa portadora. El paquete se almacena ahí hasta que haya llegado por completo, a fin de que la suma de verificación pueda comprobarse. Despues se reenvía al siguiente enrutador de la ruta hasta que llegue al *host* de destino, donde se entrega. Este mecanismo se conoce como conmutación de paquetes de almacenamiento y reenvío, como vimos en capítulos anteriores.

5.1.2 Servicios proporcionados a la capa de transporte

La capa de red proporciona servicios a la capa de transporte en la interfaz capa de red/capa de transporte. Una pregunta importante es qué tipo de servicios proporciona la capa de red a la capa de transporte. Los servicios de la capa de red se han diseñado con los siguientes objetivos en mente.

1. Los servicios deben ser independientes de la tecnología del enrutador.
2. La capa de transporte debe estar aislada de la cantidad, tipo y topología de los enrutadores presentes.
3. Las direcciones de red disponibles para la capa de transporte deben seguir un plan de numeración uniforme, aun a través de varias LANs y WANs.

Dadas estas metas, los diseñadores de la capa de red tienen mucha libertad para escribir especificaciones detalladas de los servicios que se ofrecerán a la capa de transporte. Con frecuencia esta libertad degenera en una batalla campal entre dos bandos en conflicto. La discusión se centra en determinar si la capa de red debe proporcionar servicio orientado o no orientado a la conexión.

Un bando (representado por la comunidad de Internet) alega que la tarea del enrutador es mover bits de un lado a otro, y nada más. Desde su punto de vista (basado en casi 30 años de experiencia con una red de computadoras real y operativa), la subred es inherentemente inestable, sin importar su diseño. Por lo tanto, los *hosts* deben aceptar este hecho y efectuar ellos mismos el control de errores (es decir, detección y corrección de errores) y el control de flujo.

Este punto de vista conduce directamente a la conclusión de que el servicio de red no debe ser orientado a la conexión, y debe contar tan sólo con las primitivas SEND PACKET y RECEIVE PACKET. En particular, no debe efectuarse ningún ordenamiento de paquetes ni control de flujo, pues de todos modos los *hosts* lo van a efectuar y probablemente se ganaría poco haciéndolo dos veces. Además, cada paquete debe llevar la dirección de destino completa, porque cada paquete enviado se transporta de manera independiente de sus antecesores, si los hay.

El otro bando (representado por las compañías telefónicas) argumenta que la subred debe proporcionar un servicio confiable, orientado a la conexión. Afirman que una buena guía son 100 años de experiencia exitosa del sistema telefónico mundial. Desde este punto de vista, la calidad del servicio es el factor dominante, y sin conexiones en la subred, tal calidad es muy difícil de alcanzar, especialmente para el tráfico de tiempo real como la voz y el vídeo.

Estas dos posturas se ejemplifican mejor con Internet y ATM. Internet ofrece servicio de capa de red no orientado a la conexión; las redes ATM ofrecen servicio de capa de red orientado a la conexión. Sin embargo, es interesante hacer notar que conforme las garantías de calidad del servicio se están volviendo más y más importantes, Internet está evolucionando. En particular, está empezando a adquirir propiedades que normalmente se asocian con el servicio orientado a la conexión, como veremos más adelante. En el capítulo 4 dimos un breve indicio de esta evolución en nuestro estudio sobre las VLANs.

5.1.3 Implementación del servicio no orientado a la conexión

Puesto que ya vimos las dos clases de servicios que la capa de red puede proporcionar a sus usuarios, es tiempo de analizar la manera en que funciona internamente esta capa. Se pueden realizar dos formas de organización distintas, dependiendo del tipo de servicio que se ofrezca. Si se ofrece el servicio no orientado a la conexión, los paquetes se colocan individualmente en la subred y se enrutan de manera independiente. No se necesita una configuración avanzada. En este

contexto, por lo general los paquetes se conocen como **datagramas** (en analogía con los telegramas) y la subred se conoce como **subred de datagramas**. Si se utiliza el servicio orientado a la conexión, antes de poder enviar cualquier paquete de datos, es necesario establecer una ruta del enrutador de origen al de destino. Esta conexión se conoce como **CV (circuito virtual)**, en analogía con los circuitos físicos establecidos por el sistema telefónico, y la subred se conoce como **subred de circuitos virtuales**. En esta sección examinaremos las subredes de datagramas; en la siguiente analizaremos las subredes de circuitos virtuales.

A continuación veamos cómo funciona una subred de datagramas. Suponga que el proceso *P1* de la figura 5-2 tiene un mensaje largo para *P2*. Dicho proceso entrega el mensaje a la capa de transporte y le indica a ésta que lo envíe al proceso *P2* que se encuentra en el *host H2*. El código de la capa de transporte se ejecuta en *H1*, por lo general dentro del sistema operativo. Dicho código agrega un encabezado de transporte al mensaje y entrega el resultado a la capa de red, quizás otro procedimiento dentro del sistema operativo.

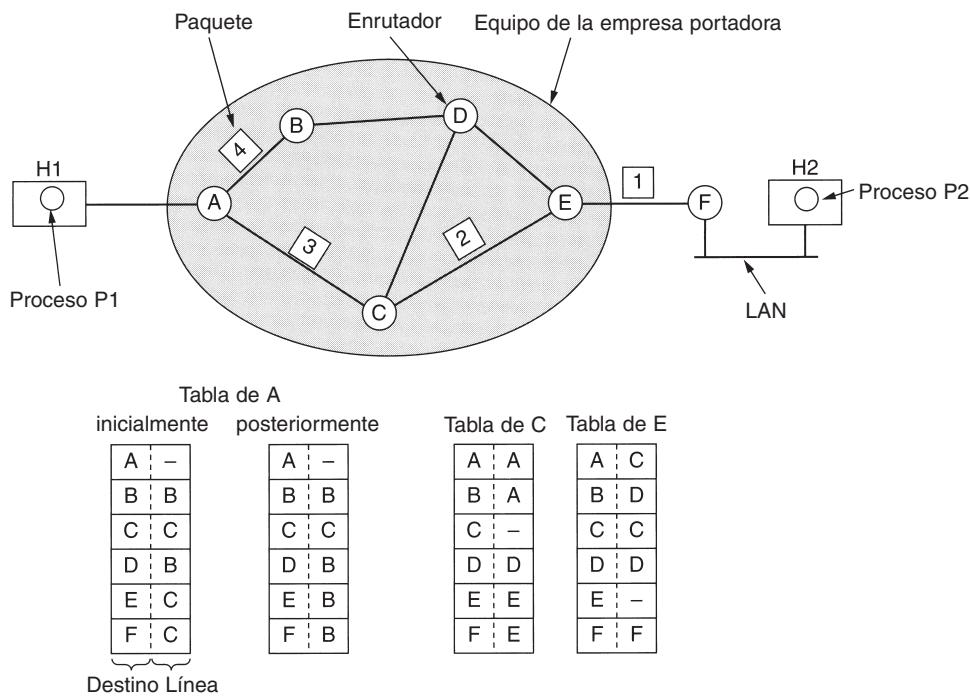


Figura 5-2. Enrutamiento dentro de una subred de datagramas.

Supongamos que el mensaje es cuatro veces más largo que el tamaño máximo de paquete, por lo que la capa de red tiene que dividirlo en cuatro paquetes, 1, 2, 3 y 4, y envía cada uno de ellos a la vez al enrutador *A* mediante algún protocolo punto a punto; por ejemplo, PPP. En este momento entra en acción la empresa portadora. Cada enrutador tiene una tabla interna que le indica a dónde enviar paquetes para cada destino posible. Cada entrada de tabla es un par que consiste en un

destino y la línea de salida que se utilizará para ese destino. Sólo se pueden utilizar líneas conectadas directamente. Por ejemplo, en la figura 5-2, *A* sólo tiene dos líneas de salida —a *B* y *C*—, por lo que cada paquete entrante debe enviarse a uno de estos enrutadores, incluso si el destino final es algún otro enrutador. En la figura, la tabla de enrutamiento inicial de *A* se muestra abajo de la leyenda “inicialmente”.

Conforme los paquetes 1, 2 y 3 llegaron a *A*, se almacenaron unos momentos (para comprobar sus sumas de verificación). Después cada uno se reenvió a *C* de acuerdo con la tabla de *A*. Posteriormente, el paquete 1 se reenvió a *E* y después a *F*. Cuando llegó a *F*, se encapsuló en una trama de capa de enlace de datos y se envió a *H*2 a través de la LAN. Los paquetes 2 y 3 siguieron la misma ruta.

Sin embargo, pasó algo diferente con el paquete 4. Cuando llegó a *A*, se envió al enrutador *B*, aunque también estaba destinado a *F*. Por alguna razón, *A* decidió enviar el paquete 4 por una ruta diferente a la de los primeros tres paquetes. Tal vez se enteró de que había alguna congestión de tráfico en alguna parte de la ruta *ACE* y actualizó su tabla de enrutamiento, como se muestra bajo la leyenda “posteriormente”. El algoritmo que maneja las tablas y que realiza las decisiones de enrutamiento se conoce como **algoritmo de enrutamiento**. Los algoritmos de enrutamiento son uno de los principales temas que estudiaremos en este capítulo.

5.1.4 Implementación del servicio orientado a la conexión

Para servicio orientado a la conexión necesitamos una subred de circuitos virtuales. Veamos cómo funciona. El propósito de los circuitos virtuales es evitar la necesidad de elegir una nueva ruta para cada paquete enviado, como en la figura 5-2. En su lugar, cuando se establece una conexión, se elige una ruta de la máquina de origen a la de destino como parte de la configuración de conexión y se almacena en tablas dentro de los enrutadores. Esa ruta se utiliza para todo el tráfico que fluye a través de la conexión, exactamente de la misma forma en que funciona el sistema telefónico. Cuando se libera la conexión, también se termina el circuito virtual. Con el servicio orientado a la conexión, cada paquete lleva un identificador que indica a cuál circuito virtual pertenece.

Como ejemplo, considere la situación que se muestra en la figura 5-3. En ésta, el *host H*1 ha establecido una conexión 1 con el *host H*2. Se recuerda como la primera entrada de cada una de las tablas de enrutamiento. La primera línea de la tabla *A* indica que si un paquete tiene el identificador de conexión 1 viene de *H*1, se enviará al enrutador *C* y se le dará el identificador de conexión 1. De manera similar, la primera entrada en *C* enruta el paquete a *E*, también con el identificador de conexión 1.

Ahora consideremos lo que sucede si *H*3 también desea establecer una conexión con *H*2. Elije el identificador de conexión 1 (debido a que está iniciando la conexión y a que ésta es su única conexión) y le indica a la subred que establezca el circuito virtual. Esto nos lleva a la segunda fila de las tablas. Observe que aquí surge un problema debido a que aunque *A* sí puede saber con facilidad cuáles paquetes de conexión 1 provienen de *H*1 y cuáles provienen de *H*3, *C* no puede hacerlo. Por esta razón, *A* asigna un identificador de conexión diferente al tráfico saliente para la segunda conexión. Con el propósito de evitar conflictos de este tipo, los enrutadores requieren

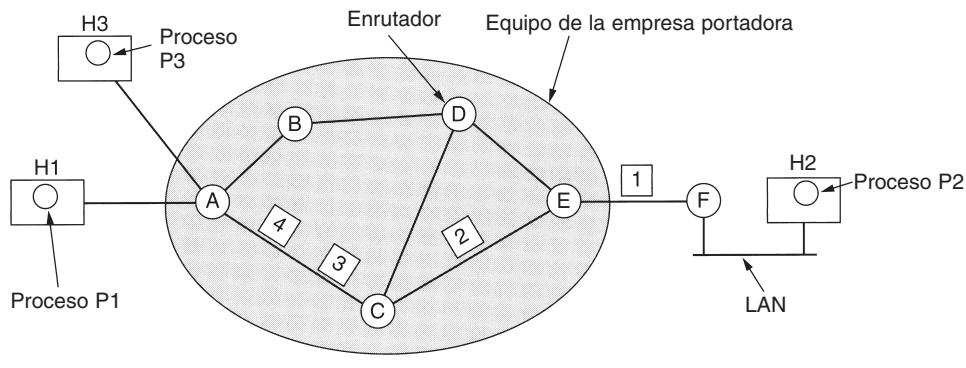


Tabla de A	Tabla de C	Tabla de E
H1 : 1 H3 : 1	C : 1 A : 2	A : 1 E : 1
C : 2 C : 2	E : 2 E : 2	C : 1 C : 2

Dentro Fuera

Figura 5-3. Enrutamiento dentro de una subred de circuitos virtuales.

la capacidad de reemplazar identificadores de conexión en los paquetes salientes. En algunos contextos a esto se le conoce como comutación de etiquetas.

5.1.5 Comparación entre las subredes de circuitos virtuales y las de datagramas

Tanto los circuitos virtuales como los datagramas tienen sus seguidores y sus detractores. Ahora intentaremos resumir los argumentos de ambos bandos. Los aspectos principales se listan en la figura 5-4, aunque los puristas probablemente podrán encontrar ejemplos contrarios para todo lo indicado en la figura.

Dentro de la subred hay varios pros y contras entre los circuitos virtuales y los datagramas. Uno de ellos tiene que ver con el espacio de memoria del enrutador y el ancho de banda. Los circuitos virtuales permiten que los paquetes contengan números de circuito en lugar de direcciones de destino completas. Si los paquetes suelen ser bastante cortos, una dirección de destino completa en cada paquete puede representar una sobrecarga significativa y, por lo tanto, ancho de banda desperdiciado. El precio que se paga por el uso interno de circuitos virtuales es el espacio de tabla en los enrutadores. La mejor elección desde el punto de vista económico depende del costo relativo entre los circuitos de comunicación y la memoria de los enrutadores.

Otro punto por considerar es el del tiempo de configuración contra el tiempo de análisis de la dirección. El uso de circuitos virtuales requiere una fase de configuración, que consume tiempo y recursos. Sin embargo, determinar lo que hay que hacer con un paquete de datos en una subred de

Asunto	Subred de datagramas	Subred de circuitos virtuales
Configuración del circuito	No necesaria	Requerida
Direccionamiento	Cada paquete contiene la dirección de origen y de destino	Cada paquete contiene un número de CV corto
Información de estado	Los enrutadores no contienen información de estado de las conexiones	Cada CV requiere espacio de tabla del enrutador por conexión
Enrutamiento	Cada paquete se enruta de manera independiente	Ruta escogida cuando se establece el CV; todos los paquetes siguen esta ruta
Efecto de fallas del enrutador	Ninguno, excepto para paquetes perdidos durante una caída	Terminan todos los CVs que pasan a través del enrutador
Calidad del servicio	Difícil	Fácil si se pueden asignar suficientes recursos por adelantado para cada CV
Control de congestión	Difícil	Fácil si pueden asignarse por adelantado suficientes recursos a cada CV

Figura 5-4. Comparación de las subredes de datagramas y de circuitos virtuales.

circuitos virtuales es fácil: el enrutador simplemente usa el número de circuito para buscar en una tabla y encontrar hacia dónde va el paquete. En una subred de datagramas se requiere un procedimiento más complicado para localizar el destino del paquete.

Otra cuestión es la cantidad requerida de espacio de tabla en la memoria del enrutador. Una subred de datagramas necesita tener una entrada para cada destino posible, mientras que una subred de circuitos virtuales sólo necesita una entrada por cada circuito virtual. Sin embargo, esta ventaja es engañosa debido a que los paquetes de configuración de conexión también tienen que enrutararse, y a que utilizan direcciones de destino, de la misma forma en que lo hacen los datagramas.

Los circuitos virtuales tienen algunas ventajas en cuanto a la calidad del servicio y a que evitan congestiones en la subred, pues los recursos (por ejemplo, búferes, ancho de banda y ciclos de CPU) pueden reservarse por adelantado al establecerse la conexión. Una vez que comienzan a llegar los paquetes, estarán ahí el ancho de banda y la capacidad de enrutamiento necesarios. En una subred de datagramas es más difícil evitar las congestiones.

En los sistemas de procesamiento de transacciones (por ejemplo, las tiendas que llaman para verificar pagos con tarjeta de crédito), la sobrecarga requerida para establecer y terminar un circuito virtual puede ocupar mucho más tiempo que el uso real del circuito. Si la mayor parte del tráfico esperado es de este tipo, el uso de circuitos virtuales dentro de la subred tiene poco sentido. Por otra parte, aquí pueden ser de utilidad los circuitos virtuales permanentes, establecidos manualmente y con duración de meses o años.

Los circuitos virtuales también tienen un problema de vulnerabilidad. Si se cae un enrutador y se pierde su memoria, todos los circuitos virtuales que pasan por él tendrán que abortarse,

aunque se recupere un segundo después. Por el contrario, si se cae un enrutador de datagramas, sólo sufrirán los usuarios cuyos paquetes estaban encolados en el enrutador en el momento de la falla y, dependiendo de si ya se había confirmado o no su recepción, tal vez ni siquiera todos ellos. La pérdida de una línea de comunicación es fatal para los circuitos virtuales que la usan, pero puede compensarse fácilmente cuando se usan datagramas. Éstos también permiten que los enrutadores equilibren el tráfico a través de la subred, ya que las rutas pueden cambiarse a lo largo de una secuencia larga de transmisiones de paquetes.

5.2 ALGORITMOS DE ENRUTAMIENTO

La función principal de la capa de red es enrutar paquetes de la máquina de origen a la de destino. En la mayoría de las subredes, los paquetes requerirán varios saltos para completar el viaje. La única excepción importante son las redes de difusión, pero aun aquí es importante el enrutamiento si el origen y el destino no están en la misma red. Los algoritmos que eligen las rutas y las estructuras de datos que usan constituyen un aspecto principal del diseño de la capa de red.

El **algoritmo de enrutamiento** es aquella parte del software de la capa de red encargada de decidir la línea de salida por la que se transmitirá un paquete de entrada. Si la subred usa datagramas de manera interna, esta decisión debe tomarse cada vez que llega un paquete de datos, dado que la mejor ruta podría haber cambiado desde la última vez. Si la subred usa circuitos virtuales internamente, las decisiones de enrutamiento se toman sólo al establecerse un circuito virtual nuevo. En lo sucesivo, los paquetes de datos simplemente siguen la ruta previamente establecida. Este último caso a veces se llama **enrutamiento de sesión**, dado que una ruta permanece vigente durante toda la sesión de usuario (por ejemplo, durante una sesión desde una terminal, o durante una transferencia de archivos).

Algunas veces es útil distinguir entre el enrutamiento, que es el proceso consistente en tomar la decisión de cuáles rutas utilizar, y el reenvío, que consiste en la acción que se toma cuando llega un paquete. Se puede considerar que un enrutador realiza dos procesos internos. Uno de ellos maneja cada paquete conforme llega, buscando en las tablas de enrutamiento la línea de salida por la cual se enviará. Este proceso se conoce como **reenvío**. El otro proceso es responsable de llenar y actualizar las tablas de enrutamiento. Es ahí donde entra en acción el algoritmo de enrutamiento.

Sin importar si las rutas para cada paquete se eligen de manera independiente o sólo cuando se establecen nuevas conexiones, hay ciertas propiedades que todo algoritmo de enrutamiento debe poseer: exactitud, sencillez, robustez, estabilidad, equidad y optimización. La exactitud y la sencillez apenas requieren comentarios, pero la necesidad de robustez puede ser menos obvia a primera vista. Una vez que una red principal entra en operación, cabría esperar que funcionara continuamente durante años sin fallas a nivel de sistema. Durante ese periodo habrá fallas de hardware y de software de todo tipo. Los *hosts*, enrutadores y líneas fallarán en forma repetida y la topología cambiará muchas veces. El algoritmo de enrutamiento debe ser capaz de manejar los cambios de topología y tráfico sin requerir el aborto de todas las actividades en todos los *hosts* y el reinicio de la red con cada caída de un enrutador.

La estabilidad también es una meta importante del algoritmo de enrutamiento. Existen algoritmos de enrutamiento que nunca alcanzan el equilibrio, sin importar el tiempo que permanezcan operativos. Un algoritmo estable alcanza el equilibrio y lo conserva. La equidad y la optimización pueden parecer algo obvias (ciertamente nadie se opondrá a ellas), pero resulta que con frecuencia son metas contradictorias. En la figura 5-5 se muestra un ejemplo sencillo de este conflicto. Suponga que hay suficiente tráfico entre A y A' , entre B y B' y entre C y C' para saturar los enlaces horizontales. A fin de aumentar al máximo el flujo total, el tráfico de X a X' debe suspenderse por completo. Por desgracia, X y X' podrían inconformarse. Evidentemente se requiere un punto medio entre la eficiencia global y la equidad hacia las conexiones individuales.

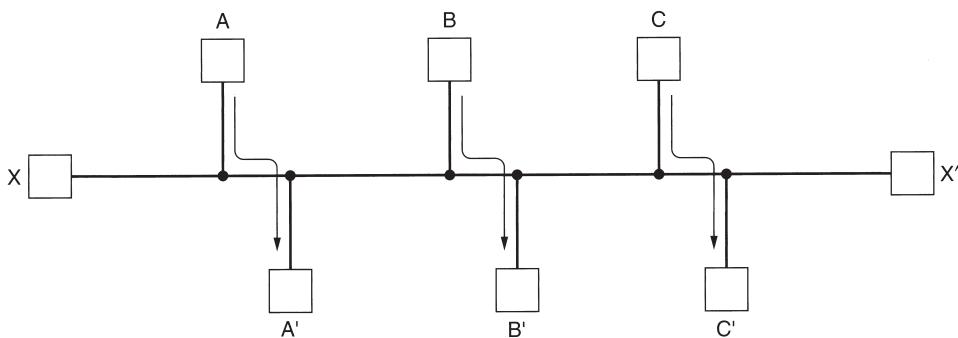


Figura 5-5. El conflicto entre equidad y optimización.

Antes de que podamos siquiera intentar encontrar el punto medio entre la equidad y la optimización, debemos decidir qué es lo que buscamos optimizar. Un candidato obvio es la minimización del retardo medio de los paquetes, pero también lo es el aumento al máximo de la velocidad real de transporte de la red. Además, estas dos metas también están en conflicto, ya que la operación de cualquier sistema de colas cerca de su capacidad máxima implica un retardo de encolamiento grande. Como término medio, muchas redes intentan minimizar el número de saltos que tiene que dar un paquete, puesto que la reducción de la cantidad de saltos reduce el retardo y también el consumo de ancho de banda, lo que a su vez mejora la velocidad real de transporte.

Los algoritmos de enrutamiento pueden agruparse en dos clases principales: no adaptativos y adaptativos. Los **algoritmos no adaptativos** no basan sus decisiones de enrutamiento en mediciones o estimaciones del tráfico y la topología actuales. En cambio, la decisión de qué ruta se usará para llegar de I a J (para todas las I y J) se toma por adelantado, fuera de línea, y se carga en los enrutadores al arrancar la red. Este procedimiento se conoce como **enrutamiento estático**.

En contraste, los **algoritmos adaptativos** cambian sus decisiones de enrutamiento para reflejar los cambios de topología y, por lo general también el tráfico. Los algoritmos adaptativos difieren en el lugar de donde obtienen su información (por ejemplo, localmente, de los enrutadores adyacentes o de todos los enrutadores), el momento de cambio de sus rutas (por ejemplo, cada ΔT segundos, cuando cambia la carga o cuando cambia la topología) y la métrica usada para la optimización (por ejemplo, distancia, número de saltos o tiempo estimado de tránsito). En las siguientes

secciones estudiaremos una variedad de algoritmos de enruteamiento, tanto estáticos como dinámicos.

5.2.1 Principio de optimización

Antes de entrar en algoritmos específicos, puede ser útil señalar que es posible hacer un postulado general sobre las rutas óptimas sin importar la topología o el tráfico de la red. Este postulado se conoce como **principio de optimización**, y establece que si el enruteador J está en ruta óptima del enruteador I al enruteador K , entonces la ruta óptima de J a K también está en la misma ruta. Para ver esto, llamemos r_1 a la parte de la ruta de I a J , y r_2 al resto de la ruta. Si existiera una ruta mejor que r_2 entre J y K , podría conectarse con r_1 para mejorar la ruta entre I y K , contradiciendo nuestra aseveración de que r_1r_2 es óptima.

Como consecuencia directa del principio de optimización, podemos ver que el grupo de rutas óptimas de todos los orígenes a un destino dado forman un árbol con raíz en el destino. Tal árbol se conoce como **árbol sumidero** (o árbol divergente) y se ilustra en la figura 5-6, donde la métrica de distancia es el número de saltos. Observe que un árbol sumidero no necesariamente es único; pueden existir otros árboles con las mismas longitudes de rutas. La meta de todos los algoritmos de enruteamiento es descubrir y utilizar los árboles sumideros de todos los enruteadores.

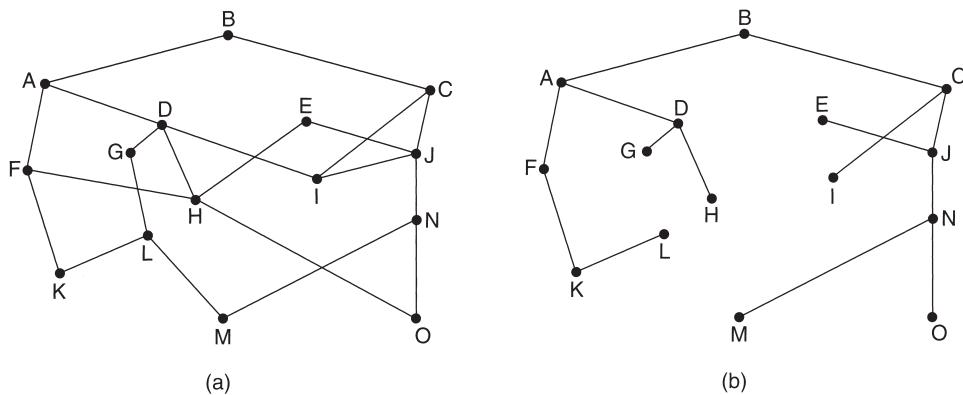


Figura 5-6. (a) Una subred. (b) Árbol sumidero para el enruteador B .

Puesto que un árbol sumidero ciertamente es un árbol, no contiene ciclos, por lo que cada paquete será entregado en un número de saltos finito y limitado. En la práctica, la vida no es tan fácil. Los enlaces y los enruteadores pueden caerse y reactivarse durante la operación, por lo que los diferentes enruteadores pueden tener ideas distintas sobre la topología actual. Además hemos evitado calladamente la cuestión de si cada enruteador tiene que adquirir de manera individual la información en la cual basa su cálculo del árbol sumidero, o si esta información se obtiene por otros

medios. Regresaremos a estos asuntos pronto. Con todo, el principio de optimización y el árbol sumidero proporcionan parámetros contra los que se pueden medir otros algoritmos de enruteamiento.

5.2.2 Enrutamiento por la ruta más corta

Comencemos nuestro estudio de los algoritmos de enruteamiento con una técnica de amplio uso en muchas formas, porque es sencilla y fácil de entender. La idea es armar un grafo de la subred, en el que cada nodo representa un enrutador y cada arco del grafo una línea de comunicación (con frecuencia llamada enlace). Para elegir una ruta entre un par dado de enrutadores, el algoritmo simplemente encuentra en el grafo la ruta más corta entre ellos.

El concepto de **ruta más corta** merece una explicación. Una manera de medir la longitud de una ruta es por la cantidad de saltos. Usando esta métrica, las rutas *ABC* y *ABE* de la figura 5-7 tienen la misma longitud. Otra métrica es la distancia geográfica en kilómetros, en cuyo caso *ABC* es claramente mucho mayor que *ABE* (suponiendo que la figura está dibujada a escala).

Sin embargo, también son posibles muchas otras métricas además de los saltos y la distancia física. Por ejemplo, cada arco podría etiquetarse con el retardo medio de encolamiento y transmisión de un paquete de prueba estándar, determinado por series de prueba cada hora. Con estas etiquetas en el grafo, la ruta más corta es la más rápida, en lugar de la ruta con menos arcos o kilómetros.

En el caso más general, las etiquetas de los arcos podrían calcularse como una función de la distancia, ancho de banda, tráfico medio, costo de comunicación, longitud media de las colas, retardo medio y otros factores. Cambiando la función de ponderación, el algoritmo calcularía la ruta “más corta” de acuerdo con cualquiera de varios criterios, o una combinación de ellos.

Se conocen varios algoritmos de cálculo de la ruta más corta entre dos nodos de un grafo. Éste se debe a Dijkstra (1959). Cada nodo se etiqueta (entre paréntesis) con su distancia al nodo de origen a través de la mejor ruta conocida. Inicialmente no se conocen rutas, por lo que todos los nodos tienen la etiqueta infinito. A medida que avanza el algoritmo y se encuentran rutas, las etiquetas pueden cambiar, reflejando mejores rutas. Una etiqueta puede ser tentativa o permanente. Inicialmente todas las etiquetas son tentativas. Una vez que se descubre que una etiqueta representa la ruta más corta posible del origen a ese nodo, se vuelve permanente y no cambia más.

Para ilustrar el funcionamiento del algoritmo de etiquetado, observe el grafo ponderado no dirigido de la figura 5-7(a), donde las ponderaciones representan, por ejemplo, distancias. Queremos encontrar la ruta más corta posible de *A* a *D*. Comenzamos por marcar como permanente el nodo *A*, indicado por un círculo relleno. Después examinamos, por turno, cada uno de los nodos adyacentes a *A* (el nodo de trabajo), reetiquetando cada uno con la distancia desde *A*. Cada vez que reetiquetamos un nodo, también lo reetiquetamos con el nodo desde el que se hizo la prueba, para poder reconstruir más tarde la ruta final. Una vez que terminamos de examinar cada uno de los nodos adyacentes a *A*, examinamos todos los nodos etiquetados tentativamente en el grafo

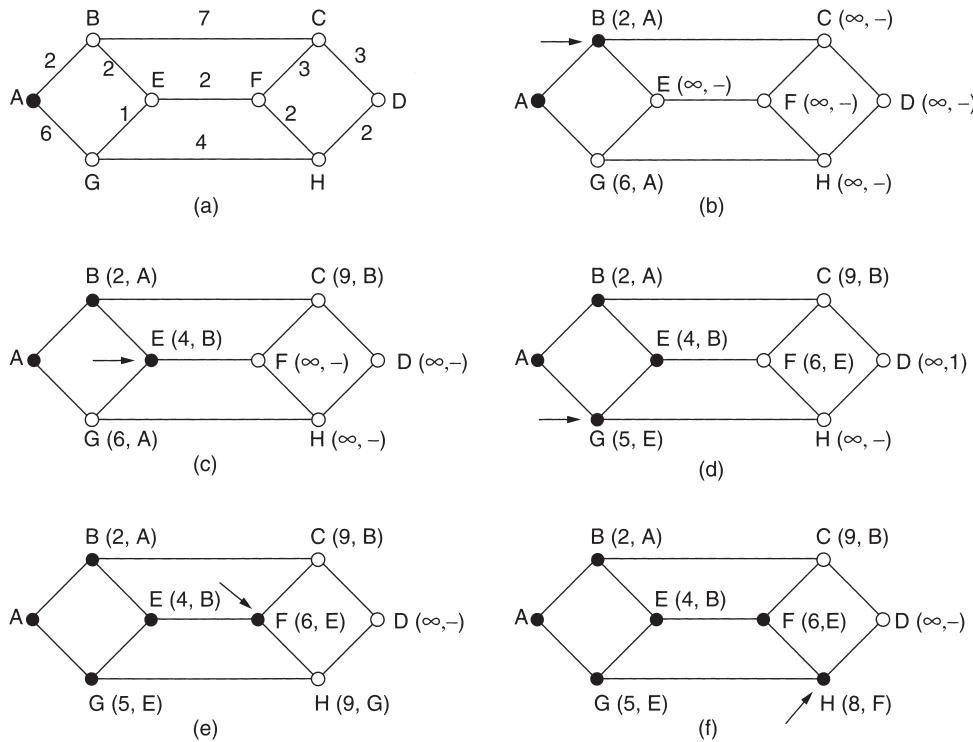


Figura 5-7. Los primeros cinco pasos del cálculo de la ruta más corta de A a D . Las flechas indican el nodo de trabajo.

completo y hacemos permanente el de la etiqueta más pequeña, como se muestra en la figura 5-7(b). Éste se convierte en el nuevo nodo de trabajo.

Ahora comenzamos por B , y examinamos todos los nodos adyacentes a él. Si la suma de la etiqueta de B y la distancia desde B al nodo en consideración es menor que la etiqueta de ese nodo, tenemos una ruta más corta, por lo que reetiquetamos ese nodo.

Tras inspeccionar todos los nodos adyacentes al nodo de trabajo y cambiar las etiquetas tentativas (de ser posible), se busca en el grafo completo el nodo etiquetado tentativamente con el menor valor. Este nodo se hace permanente y se convierte en el nodo de trabajo para la siguiente ronda. En la figura 5-7 se muestran los primeros cinco pasos del algoritmo.

Para ver por qué funciona el algoritmo, vea la figura 5-7(c). En ese punto acabamos de hacer permanente a E . Suponga que hubiera una ruta más corta que ABE , digamos $AXYZE$. Hay dos posibilidades: el nodo Z ya se hizo permanente, o no se ha hecho permanente. Si ya es permanente, entonces E ya se probó (en la ronda que siguió a aquella en la que se hizo permanente Z), por lo que la ruta $AXYZE$ no ha escapado a nuestra atención y, por lo tanto, no puede ser una ruta más corta.

Ahora considere el caso en el que Z aún está etiquetado tentativamente. O bien la etiqueta de Z es mayor o igual que la de E , en cuyo caso $AXYZE$ no puede ser una ruta más corta que ABE , o es menor que la de E , en cuyo caso Z , y no E , se volverá permanente primero, lo que permitirá que E se pruebe desde Z .

Este algoritmo se da en la figura 5-8. Las variables globales n y $dist$ describen el grafo y son inicializadas antes de que se llame a *shortest_path*. La única diferencia entre el programa y el algoritmo antes descrito es que, en la figura 5-8, calculamos la ruta más corta posible comenzando por el nodo terminal, t , en lugar de en el nodo de origen, s . Dado que la ruta más corta posible desde t a s en un grafo no dirigido es igual a la ruta más corta de s a t , no importa el extremo por el que comencemos (a menos que haya varias rutas más cortas posibles, en cuyo caso la inversión de la búsqueda podría descubrir una distinta). La razón de una búsqueda hacia atrás es que cada nodo está etiquetado con su antecesor, en lugar de con su sucesor. Al copiar la ruta final en la variable de salida, $path$, la ruta de salida se invierte. Al invertir la búsqueda, ambos efectos se cancelan y la respuesta se produce en el orden correcto.

5.2.3 Inundación

Otro algoritmo estático es la **inundación**, en la que cada paquete de entrada se envía por cada una de las líneas de salida, excepto aquella por la que llegó. La inundación evidentemente genera grandes cantidades de paquetes duplicados; de hecho, una cantidad infinita a menos que se tomen algunas medidas para limitar el proceso. Una de estas medidas es integrar un contador de saltos en el encabezado de cada paquete, que disminuya con cada salto, y el paquete se descarte cuando el contador llegue a cero. Lo ideal es inicializar el contador de saltos a la longitud de la ruta entre el origen y el destino. Si el emisor desconoce el tamaño de la ruta, puede inicializar el contador al peor caso, es decir, el diámetro total de la subred.

Una técnica alterna para ponerle diques a la inundación es llevar un registro de los paquetes difundidos, para evitar enviarlos una segunda vez. Una manera de lograr este propósito es hacer que el enrutador de origen ponga un número de secuencia en cada paquete que recibe de sus *hosts*. Cada enrutador necesita una lista por cada enrutador de origen que indique los números de secuencia originados en ese enrutador que ya ha visto. Si un paquete de entrada está en la lista, no se difunde.

Para evitar que la lista crezca sin límites, cada lista debe incluir un contador, k , que indique que todos los números de secuencia hasta k ya han sido vistos. Cuando llega un paquete, es fácil comprobar si es un duplicado; de ser así, se descarta. Es más, no se necesita la lista completa por debajo de k , pues k la resume efectivamente.

Una variación de la inundación, un poco más práctica, es la **inundación selectiva**. En este algoritmo, los enrutadores no envían cada paquete de entrada por todas las líneas, sino sólo por aquellas que van aproximadamente en la dirección correcta. Por lo general, no tiene mucho caso enviar un paquete dirigido al oeste a través de una línea dirigida al este, a menos que la topología sea extremadamente peculiar y que el enrutador esté seguro de este hecho.

```

#define MAX_NODES 1024           /* número máximo de nodos */
#define INFINITY 1000000000     /* un número mayor que cualquier ruta
                                máxima */
int n, dist[MAX_NODES][MAX_NODES];    /* dist[i][j] es la distancia entre
                                         i y j */

void shortest_path(int s, int t, int path[])
{ struct state {                         /* la ruta con la que se está
                                         trabajando */
    int predecessor;                    /* nodo previo */
    int length;                        /* longitud del origen a este nodo */
    enum {permanent, tentative} label; /* estado de la etiqueta */
} state[MAX_NODES];}

int i, k, min;
struct state *p;

for (p = &state[0]; p < &state[n]; p++) { /* estado de inicialización */
    p->predecessor = -1;
    p->length = INFINITY;
    p->label = tentative;
}
state[t].length = 0; state[t].label = permanent;
k = t;                                     /* k es el nodo de trabajo inicial */
do{
    for (i = 0; i < n; i++)               /* este grafo tiene n nodos */
        if (dist[k][i] != 0 && state[i].label == tentative) {
            if (state[k].length + dist[k][i] < state[i].length) {
                state[i].predecessor = k;
                state[i].length = state[k].length + dist[k][i];
            }
        }
    /* Encuentra el nodo etiquetado tentativamente con la etiqueta menor. */
    k = 0; min = INFINITY;
    for (i = 0; i < n; i++)
        if (state[i].label == tentative && state[i].length < min) {
            min = state[i].length;
            k = i;
        }
    state[k].label = permanent;
} while (k != s);

/* Copia la ruta en el arreglo de salida. */
i = 0; k = s;
do {path[i++] = k; k = state[k].predecessor;} while (k >= 0);
}

```

Figura 5-8. Algoritmo de Dijkstra para calcular la ruta más corta a través de un grafo.

La inundación no es práctica en la mayoría de las aplicaciones, pero tiene algunos usos. Por ejemplo, en aplicaciones militares, donde grandes cantidades de enrutadores pueden volar en pedazos en cualquier momento, es altamente deseable la excelente robustez de la inundación. En las aplicaciones distribuidas de bases de datos a veces es necesario actualizar concurrentemente todas las bases de datos, en cuyo caso la inundación puede ser útil. En las redes inalámbricas, algunas estaciones que se encuentren dentro del alcance de radio de una estación dada pueden recibir los mensajes que ésta trasmite, lo cual es, de hecho, inundación, y algunos algoritmos utilizan esta propiedad. Un cuarto posible uso de la inundación es como métrica contra la que pueden compararse otros algoritmos de enrutamiento. La inundación siempre escoge la ruta más corta posible, porque escoge en paralelo todas las rutas posibles. En consecuencia, ningún otro algoritmo puede producir un retardo más corto (si ignoramos la sobrecarga generada por el proceso de inundación mismo).

5.2.4 Enrutamiento por vector de distancia

Las redes modernas de computadoras por lo general utilizan algoritmos de enrutamiento dinámico en lugar de los estáticos antes descritos, pues los algoritmos estáticos no toman en cuenta la carga actual de la red. En particular, dos algoritmos dinámicos, el enrutamiento por vector de distancia y el enrutamiento por estado del enlace, son los más comunes. En esta sección veremos el primer algoritmo. En la siguiente estudiaremos el segundo.

Los algoritmos de **enrutamiento por vector de distancia** operan haciendo que cada enrutador mantenga una tabla (es decir, un vector) que da la mejor distancia conocida a cada destino y la línea que se puede usar para llegar ahí. Estas tablas se actualizan intercambiando información con los vecinos.

El algoritmo de enrutamiento por vector de distancia a veces recibe otros nombres, incluido el de algoritmo de enrutamiento **Bellman-Ford** distribuido y el de algoritmo **Ford-Fulkerson**, por los investigadores que los desarrollaron (Bellman, 1957, y Ford y Fulkerson, 1962). Éste fue el algoritmo original de enrutamiento de ARPANET y también se usó en Internet con el nombre RIP.

En el enrutamiento por vector de distancia, cada enrutador mantiene una tabla de enrutamiento indizada por, y conteniendo un registro de, cada enrutador de la subred. Esta entrada comprende dos partes: la línea preferida de salida hacia ese destino y una estimación del tiempo o distancia a ese destino. La métrica usada podría ser la cantidad de saltos, el retardo de tiempo en milisegundos, el número total de paquetes encolados a lo largo de la ruta, o algo parecido.

Se supone que el enrutador conoce la “distancia” a cada uno de sus vecinos. Si la métrica es de saltos, la distancia simplemente es un salto. Si la métrica es la longitud de la cola, el enrutador simplemente examina cada cola. Si la métrica es el retardo, el enrutador puede medirlo en forma directa con paquetes especiales de ECO que el receptor simplemente marca con la hora y lo regresa tan rápido como puede.

Por ejemplo, suponga que el retardo se usa como métrica y que el enrutador conoce el retardo a cada uno de sus vecinos. Una vez cada T msec, cada enrutador envía a todos sus vecinos una

lista de sus retardos estimados a cada destino. También recibe una lista parecida de cada vecino. Imagine que una de estas tablas acaba de llegar del vecino X , siendo X_i la estimación de X respecto al tiempo que le toma llegar al enrutador i . Si el enrutador sabe que el retardo a X es de m mseg, también sabe que puede llegar al enrutador i a través de X en $X_i + m$ mseg. Efectuando este cálculo para cada vecino, un enrutador puede encontrar la estimación que parezca ser la mejor y usar esa estimación, así como la línea correspondiente, en su nueva tabla de enrutamiento. Observe que la vieja tabla de enrutamiento no se usa en este cálculo.

Este proceso de actualización se ilustra en la figura 5-9. En la parte (a) se muestra una subred. En las primeras cuatro columnas de la parte (b) aparecen los vectores de retardo recibidos de los vecinos del enrutador J . A indica tener un retardo de 12 mseg a B , un retardo de 25 mseg a C , un retardo de 40 mseg a D , etc. Suponga que J ha medido o estimado el retardo a sus vecinos A, I, H y K en 8, 10, 12 y 6 mseg, respectivamente.

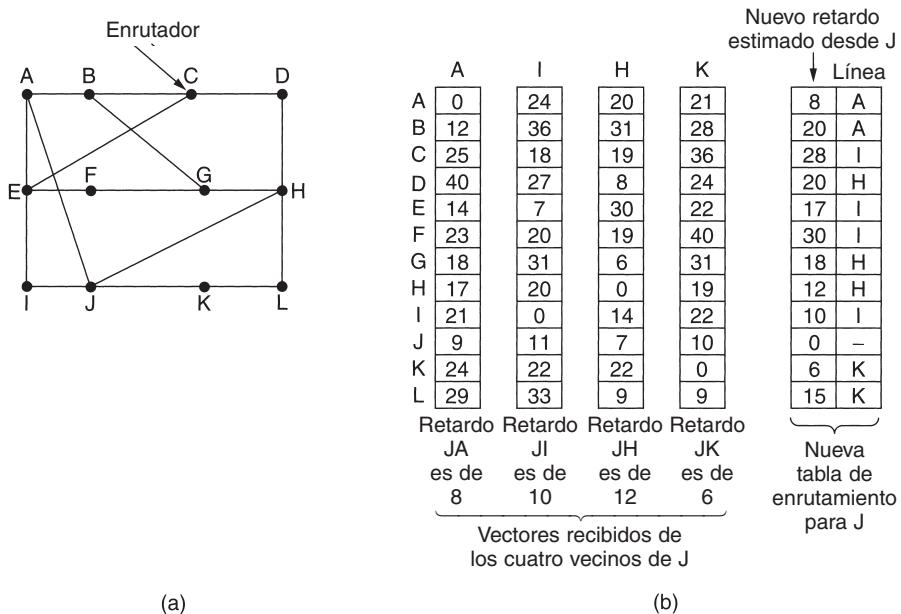


Figura 5-9. (a) Subred. (b) Entrada de A, I, H, K y la nueva tabla de enrutamiento de J.

Considere la manera en que J calcula su nueva ruta al enrutador G . Sabe que puede llegar a A en 8 mseg, y A indica ser capaz de llegar a G en 18 mseg, por lo que J sabe que puede contar con un retardo de 26 mseg a G si reenvía a través de A los paquetes destinados a G . Del mismo modo, J calcula el retardo a G a través de I, H y K en 41 (31 + 10), 18 (6 + 12) y 37 (31 + 6) mseg, respectivamente. El mejor de estos valores es el 18, por lo que escribe una entrada en su tabla de enrutamiento indicando que el retardo a G es de 18 mseg, y que la ruta que se utilizará es vía H . Se lleva a cabo el mismo cálculo para los demás destinos, y la nueva tabla de enrutamiento se muestra en la última columna de la figura.

El problema de la cuenta hasta infinito

El enrutamiento por vector de distancia funciona en teoría, pero tiene un problema serio en la práctica: aunque llega a la respuesta correcta, podría hacerlo lentamente. En particular, reacciona con rapidez a las buenas noticias, pero con lentitud ante las malas. Considere un enrutador cuya mejor ruta al destino X es larga. Si en el siguiente intercambio el vecino A informa repentinamente un retardo corto a X , el enrutador simplemente se comuta a modo de usar la línea a A para enviar tráfico hasta X . En un intercambio de vectores, se procesan las buenas noticias.

Para ver la rapidez de propagación de las buenas noticias, considere la subred de cinco nodos (lineal) de la figura 5-10, en donde la métrica de retardo es el número de saltos. Suponga que A está desactivado inicialmente y que los otros enrutadores lo saben. En otras palabras, habrán registrado como infinito el retardo a A .

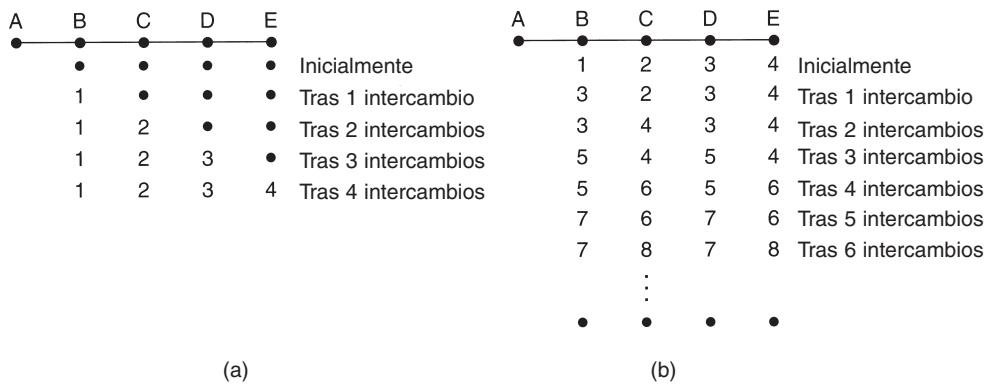


Figura 5-10. El problema de la cuenta hasta infinito.

Al activarse A , los demás enrutadores saben de él gracias a los intercambios de vectores. Por sencillez, supondremos que hay un gong gigantesco en algún lado, golpeando periódicamente para iniciar de manera simultánea un intercambio de vectores entre todos los enrutadores. En el momento del primer intercambio, B se entera de que su vecino de la izquierda tiene un retardo de 0 hacia A . B crea entonces una entrada en su tabla de enrutamiento, indicando que A está a un salto de distancia hacia la izquierda. Los demás enrutadores aún piensan que A está desactivado. En este punto, las entradas de la tabla de enrutamiento de A se muestran en la segunda fila de la figura 5-10(a). Durante el siguiente intercambio, C se entera de que B tiene una ruta a A de longitud 1, por lo que actualiza su tabla de enrutamiento para indicar una ruta de longitud 2, pero D y E no se enteran de las buenas nuevas sino hasta después. Como es evidente, las buenas noticias se difunden a razón de un salto por intercambio. En una subred cuya ruta mayor tiene una longitud de N saltos, en un lapso de N intercambios todo mundo sabrá sobre las líneas y enrutadores recientemente revividos.

Ahora consideremos la situación de la figura 5-10(b), en la que todas las líneas y enrutadores están activos inicialmente. Los enrutadores B , C , D y E tienen distancias a A de 1, 2, 3 y 4, respectivamente. De pronto, A se desactiva, o bien se corta la línea entre A y B , que de hecho es la misma cosa desde el punto de vista de B .

En el primer intercambio de paquetes, B no escucha nada de A . Afortunadamente, C dice: "No te preocunes. Tengo una ruta a A de longitud 2". B no sabe que la ruta de C pasa a través de B mismo. Hasta donde B sabe, C puede tener 10 líneas, todas con rutas independientes a A de longitud 2. Como resultado, B ahora piensa que puede llegar a A por medio de C , con una longitud de ruta de 3. D y E no actualizan sus entradas para A en el primer intercambio.

En el segundo intercambio, C nota que cada uno de sus vecinos indica tener una ruta a A de longitud 3. C escoge una de ellas al azar y hace que su nueva distancia a A sea de 4, como se muestra en la tercera fila de la figura 5-10(b). Los intercambios subsecuentes producen la historia mostrada en el resto de la figura 5-10(b).

A partir de esta figura debe quedar clara la razón por la que las malas noticias viajan con lentitud: ningún enrutador jamás tiene un valor mayor en más de una unidad que el mínimo de todos sus vecinos. Gradualmente, todos los enrutadores elevan cuentas hacia el infinito, pero el número de intercambios requerido depende del valor numérico usado para el infinito. Por esta razón, es prudente hacer que el infinito sea igual a la ruta más larga, más 1. Si la métrica es el retardo de tiempo, no hay un límite superior bien definido, por lo que se necesita un valor alto para evitar que una ruta con un retardo grande sea tratada como si estuviera desactivada. Este problema se conoce como el problema de la **cuenta hasta el infinito**, lo cual no es del todo sorprendente. Se han realizado algunos intentos por resolverlo (como el horizonte dividido con rutas inalcanzables [*poisoned reverse*] en el RFC 1058), pero ninguno funciona bien en general. La esencia del problema consiste en que cuando X indica a Y que tiene una ruta en algún lugar, Y no tiene forma de saber si él mismo está en la ruta.

5.2.5 Enrutamiento por estado del enlace

El enrutamiento por vector de distancia se usó en ARPANET hasta 1979, cuando fue reemplazado por el enrutamiento por estado del enlace. Dos problemas principales causaron su desaparición. Primero, debido a que la métrica de retardo era la longitud de la cola, no tomaba en cuenta el ancho de banda al escoger rutas. Inicialmente, todas las líneas eran de 56 kbps, por lo que el ancho de banda no era importante, pero una vez que se modernizaron algunas líneas a 230 kbps y otras a 1.544 Mbps, el no tomar en cuenta el ancho de banda se volvió un problema importante. Por supuesto, habría sido posible cambiar la métrica de retardo para considerar el ancho de banda, pero también existía un segundo problema: que el algoritmo con frecuencia tardaba demasiado en converger (el problema de la cuenta hasta el infinito). Por estas razones, el algoritmo fue reemplazado por uno completamente nuevo, llamado **enrutamiento por estado del enlace**. Hoy en día se usan bastante algunas variantes del enrutamiento por estado del enlace.

El concepto en que se basa el enrutamiento por estado del enlace es sencillo y puede enunciarse en cinco partes. Cada enrutador debe:

1. Descubrir a sus vecinos y conocer sus direcciones de red.
2. Medir el retardo o costo para cada uno de sus vecinos.
3. Construir un paquete que indique todo lo que acaba de aprender.
4. Enviar este paquete a todos los demás enrutadores.
5. Calcular la ruta más corta a todos los demás enrutadores.

De hecho, toda la topología y todos los retardos se miden experimentalmente y se distribuyen a cada enrutador. Entonces puede usarse el algoritmo de Dijkstra para encontrar la ruta más corta a los demás enrutadores. A continuación veremos con mayor detalle estos cinco pasos.

Conocimiento de los vecinos

Cuando un enrutador se pone en funcionamiento, su primera tarea es averiguar quiénes son sus vecinos; esto lo realiza enviando un paquete HELLO especial a cada línea punto a punto. Se espera que el enrutador del otro extremo regrese una respuesta indicando quién es. Estos nombres deben ser globalmente únicos puesto que, cuando un enrutador distante escucha después que tres enrutadores están conectados a F , es indispensable que pueda determinar si los tres se refieren al mismo F .

Cuando se conectan dos o más enrutadores mediante una LAN, la situación es ligeramente más complicada. En la figura 5-11(a) se ilustra una LAN a la que están conectados directamente tres enrutadores, A , C y F . Cada uno de estos enrutadores está conectado a uno o más enrutadores adicionales.

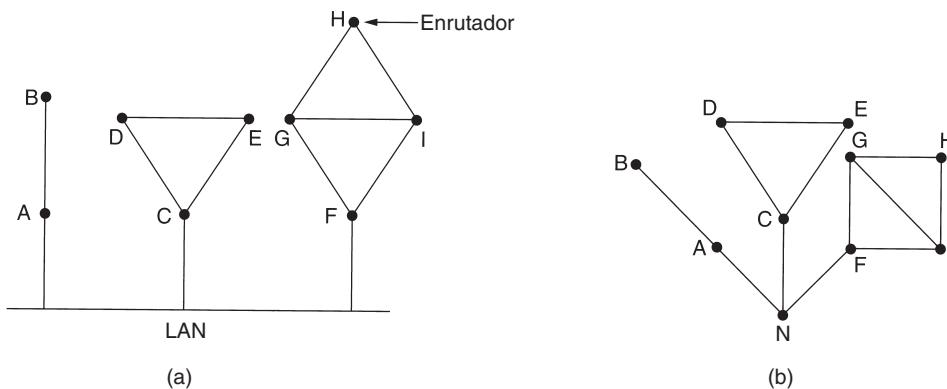


Figura 5-11. (a) Nueve enrutadores y una LAN. (b) Modelo de grafo de (a).

Una manera de modelar la LAN es considerarla como otro nodo, como se muestra en la figura 5-11(b). Aquí hemos introducido un nodo artificial nuevo, N , al que están conectados A , C y F . El hecho de que sea posible ir de A a C a través de la LAN se representa aquí mediante la ruta ANC .

Medición del costo de la línea

El algoritmo de enrutamiento por estado del enlace requiere que cada enrutador sepa, o cuando menos tenga una idea razonable, del retardo a cada uno de sus vecinos. La manera más directa de determinar este retardo es enviar un paquete ECHO especial a través de la línea y una vez que llegue al otro extremo, éste debe regresarlo inmediatamente. Si se mide el tiempo de ida y vuelta y se divide entre dos, el enrutador emisor puede tener una idea razonable del retardo. Para

obtener todavía mejores resultados, la prueba puede llevarse a cabo varias veces y usarse el promedio. Por supuesto que este método asume de manera implícita que los retardos son simétricos, lo cual no siempre es el caso.

Un aspecto interesante es si se debe tomar en cuenta la carga al medir el retardo. Para considerar la carga, el temporizador debe iniciarse cuando el paquete ECHO se ponga en la cola. Para ignorar la carga, el temporizador debe iniciarse cuando el paquete ECHO alcance el frente de la cola.

Pueden citarse argumentos a favor de ambos métodos. La inclusión de los retardos inducidos por el tráfico en las mediciones implica que cuando un enrutador puede escoger entre dos líneas con el mismo ancho de banda, una con carga alta continua y otra sin ella, considerará como ruta más corta la de la línea sin carga. Esta selección resultará en un mejor desempeño.

Desgraciadamente, también hay un argumento en contra de la inclusión de la carga en el cálculo del retardo. Considere la subred de la figura 5-12, dividida en dos partes, este y oeste, conectadas por dos líneas, *CF* y *EI*.

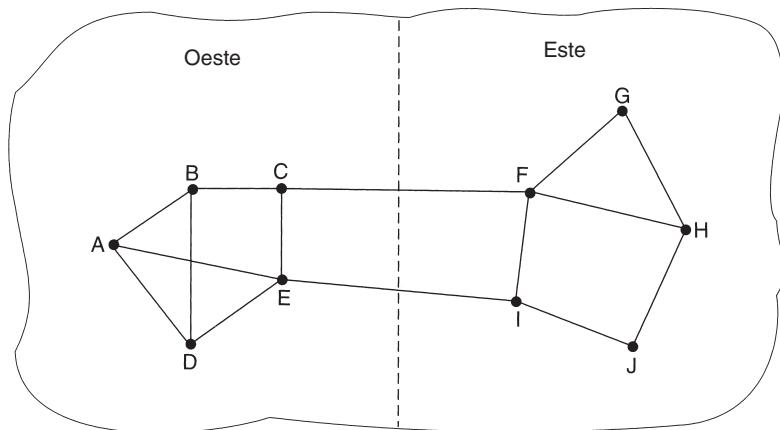


Figura 5-12. Subred en la que las partes este y oeste están conectadas por dos líneas.

Suponga que la mayor parte del tráfico entre el este y el oeste usa la línea *CF* y, como resultado, esta línea tiene tráfico alto con retardos grandes. La inclusión del retardo por encolamiento en el cálculo de la ruta más corta hará más atractiva a *EI*. Una vez instaladas las nuevas tablas de enrutamiento, la mayor parte del tráfico este-oeste pasará ahora por *EI*, sobrecargando esta línea. En consecuencia, en la siguiente actualización, *CF* aparecerá como la ruta más corta. Como resultado, las tablas de enrutamiento pueden oscilar sin control, lo que provocará un enrutamiento errático y muchos problemas potenciales. Si se ignora la carga y sólo se considera el ancho de banda, no ocurre este problema. De manera alterna, puede distribuirse la carga entre ambas líneas, pero esta solución no aprovecha al máximo la mejor ruta. No obstante, para evitar oscilaciones en la selección de la mejor ruta, podría ser adecuado dividir la carga entre múltiples líneas, con una fracción conocida de la carga viajando sobre cada una de ellas.

Construcción de los paquetes de estado del enlace

Una vez que se ha recabado la información necesaria para el intercambio, el siguiente paso es que cada enrutador construya un paquete que contenga todos los datos. El paquete comienza con la identidad del emisor, seguida de un número de secuencia, una edad (que se describirá después) y una lista de vecinos. Se da el retardo de vecino. En la figura 5-13(a) se da un ejemplo de subred, y los retardos se muestran como etiquetas en las líneas. En la figura 5-13(b) se muestran los paquetes de estado del enlace de los seis enrutadores.

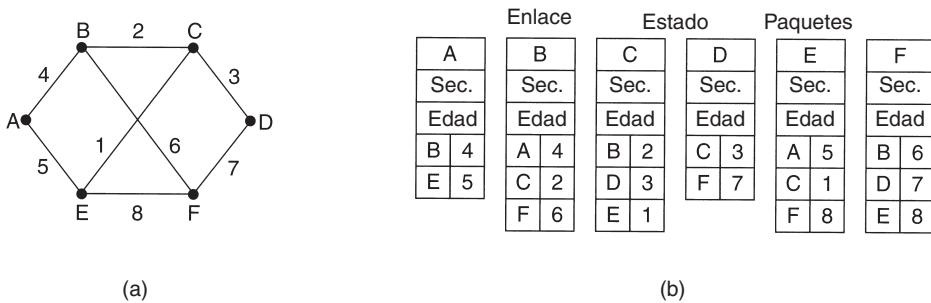


Figura 5-13. (a) Subred. (b) Paquetes de estado del enlace para esta subred.

Es fácil construir los paquetes de estado del enlace. La parte difícil es determinar cuándo construirlos. Una posibilidad es construirlos de manera periódica, es decir, a intervalos regulares. Otra posibilidad es construirlos cuando ocurra un evento significativo, como la caída o la reactivación de una línea o de un vecino, o el cambio apreciable de sus propiedades.

Distribución de los paquetes de estado del enlace

La parte más complicada del algoritmo es la distribución confiable de los paquetes de estado del enlace. A medida que se distribuyen e instalan los paquetes, los enrutadores que reciban los primeros cambiarán sus rutas. En consecuencia, los distintos enrutadores podrían estar usando versiones diferentes de la topología, lo que puede conducir a inconsistencias, ciclos, máquinas inalcanzables y otros problemas.

Primero describiremos el algoritmo básico de distribución y luego lo refinaremos. La idea fundamental es utilizar inundación para distribuir los paquetes de estado del enlace. A fin de mantener controlada la inundación, cada paquete contiene un número de secuencia que se incrementa con cada paquete nuevo enviado. Los enrutadores llevan el registro de todos los pares (enrutador de origen, secuencia) que ven. Cuando llega un paquete de estado del enlace, se verifica contra la lista de paquetes ya vistos. Si es nuevo, se reenvía a través de todas las líneas, excepto aquella por la que llegó. Si es un duplicado, se descarta. Si llega un paquete con número de secuencia menor que el mayor visto hasta el momento, se rechaza como obsoleto debido que el enrutador tiene datos más recientes.

Este algoritmo tiene algunos problemas, pero son manejables. Primero, si los números de secuencia vuelven a comenzar, reinará la confusión. La solución aquí es utilizar un número de secuencia de 32 bits. Con un paquete de estado del enlace por segundo, el tiempo para volver a empezar será de 137 años, por lo que puede ignorarse esta posibilidad.

Segundo, si llega a caerse un enrutador, perderá el registro de su número de secuencia. Si comienza nuevamente en 0, se rechazará como duplicado el siguiente paquete.

Tercero, si llega a corromperse un número de secuencia y se escribe 65,540 en lugar de 4 (un error de 1 bit), los paquetes 5 a 65,540 serán rechazados como obsoletos, dado que se piensa que el número de secuencia actual es 65,540.

La solución a todos estos problemas es incluir la edad de cada paquete después del número de secuencia y disminuirla una vez cada segundo. Cuando la edad llega a cero, se descarta la información de ese enrutador. Generalmente, un paquete nuevo entra, por ejemplo, cada 10 segundos, por lo que la información de los enrutadores sólo expira cuando el enrutador está caído (o cuando se pierden seis paquetes consecutivos, evento poco probable). Los enrutadores también decrementan el campo de *edad* durante el proceso inicial de inundación para asegurar que no pueda perderse ningún paquete y sobrevivir durante un periodo indefinido (se descarta el paquete cuya edad sea cero).

Algunos refinamientos de este algoritmo lo hacen más robusto. Una vez que un paquete de estado del enlace llega a un enrutador para ser inundado, no se encola para transmisión inmediata. En vez de ello, entra en un área de almacenamiento donde espera un tiempo breve. Si antes de transmitirlo entra otro paquete de estado del enlace proveniente del mismo origen, se comparan sus números de secuencia. Si son iguales, se descarta el duplicado. Si son diferentes, se desecha el más viejo. Como protección contra los errores en las líneas enrutador-enrutador, se confirma la recepción de todos los paquetes de estado del enlace. Cuando se desactiva una línea, se examina el área de almacenamiento en orden *round-robin* para seleccionar un paquete o confirmación de recepción a enviar.

En la figura 5-14 se describe la estructura de datos usada por el enrutador *B* para la subred de la figura 5-13(a). Cada fila aquí corresponde a un paquete de estado del enlace recién llegado, pero aún no procesado por completo. La tabla registra dónde se originó el paquete, su número de secuencia y edad, así como los datos. Además, hay banderas de transmisión y de confirmación de recepción para cada una de las tres líneas de *B* (a *A*, *C* y *F*, respectivamente). Las banderas de envío significan que el paquete debe enviarse a través de la línea indicada. Las banderas de confirmación de recepción significan que su confirmación debe suceder ahí.

En la figura 5-14, el paquete de estado del enlace de *A* llegó directamente, por lo que debe enviarse a *C* y *F*, y debe confirmarse la recepción a *A*, como lo muestran los bits de bandera. De la misma manera, el paquete de *F* tiene que reenviarse a *A* y a *C*, y debe enviarse a *F* la confirmación de su recepción.

Sin embargo, la situación del tercer paquete, de *E*, es diferente. Llegó dos veces, la primera a través de *EAB* y la segunda por medio de *EFB*. En consecuencia, este paquete tiene que enviarse sólo a *C*, pero debe confirmarse su recepción tanto a *A* como a *F*, como lo indican los bits.

Si llega un duplicado mientras el original aún está en el búfer, los bits tienen que cambiar. Por ejemplo, si llega una copia del estado de *C* desde *F* antes de que se reenvíe la cuarta entrada de la tabla, cambiarán los seis bits a 100011 para indicar que debe enviarse a *F* una confirmación de recepción del paquete, sin enviarle el paquete mismo.

Origen	Sec.	Edad	Banderas de envío			Banderas de ACK			Datos
			A	C	F	A	C	F	
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Figura 5-14. El búfer de paquetes para el enrutador *B* de la figura 5-13.

Cálculo de las nuevas rutas

Una vez que un enrutador ha acumulado un grupo completo de paquetes de estado del enlace, puede construir el grafo de la subred completa porque todos los enlaces están representados. De hecho, cada enlace se representa dos veces, una para cada dirección. Los dos valores pueden promediarse o usarse por separado.

Ahora puede ejecutar localmente el algoritmo de Dijkstra para construir la ruta más corta a todos los destinos posibles. Los resultados de este algoritmo pueden instalarse en las tablas de enrutamiento, y la operación normal puede reiniciarse.

Para una subred con n enrutadores, cada uno de los cuales tiene k vecinos, la memoria requerida para almacenar los datos de entrada es proporcional a kn . En las subredes grandes éste puede ser un problema. También puede serlo el tiempo de cómputo. Sin embargo, en muchas situaciones prácticas, el enrutamiento por estado del enlace funciona bien.

Sin embargo, problemas con el hardware o el software pueden causar estragos con este algoritmo (lo mismo que con otros). Por ejemplo, si un enrutador afirma tener una línea que no tiene, u olvida una línea que sí tiene, el grafo de la subred será incorrecto. Si un enrutador deja de reenviar paquetes, o los corrompe al hacerlo, surgirán problemas. Por último, si al enrutador se le acaba la memoria o se ejecuta mal el algoritmo de cálculo de enrutamiento, surgirán problemas. A medida que la subred crece en decenas o cientos de miles de nodos, la probabilidad de falla ocasional de un enrutador deja de ser insignificante. Lo importante es tratar de limitar el daño cuando ocurra lo inevitable. Perlman (1988) estudia detalladamente estos problemas y sus soluciones.

El enrutamiento por estado del enlace se usa ampliamente en las redes actuales, por lo que son pertinentes unas pocas palabras sobre algunos protocolos que lo usan. El protocolo OSPF, que se emplea cada vez con mayor frecuencia en Internet, utiliza un algoritmo de estado del enlace. Describiremos OSPF en la sección 5.6.4.

Otro protocolo de estado del enlace importante es el **IS-IS (sistema intermedio-sistema intermedio)**, diseñado por DECnet y más tarde adoptado por la ISO para utilizarlo con su protocolo de capa de red sin conexiones, CLNP. Desde entonces se ha modificado para manejar también

otros protocolos, siendo el más notable el IP. El IS-IS se usa en varias redes dorsales de Internet (entre ellas la vieja red dorsal NSFNET) y en muchos sistemas celulares digitales, como CDPD. El Novell NetWare usa una variante menor del IS-IS (NLSP) para el enrutamiento de paquetes IPX.

Básicamente, el IS-IS distribuye una imagen de la topología de enrutadores, a partir de la cual se calculan las rutas más cortas. Cada enrutador anuncia, en su información de estado del enlace, las direcciones de capa de red que puede alcanzar de manera directa. Estas direcciones pueden ser IP, IPX, AppleTalk o cualquier otra dirección. El IS-IS incluso puede manejar varios protocolos de red al mismo tiempo.

Muchas de las innovaciones diseñadas para el IS-IS fueron adoptadas por el OSPF (el cual se diseñó varios años después que el IS-IS). Entre éstas se encuentran un método autorregulable para inundar actualizaciones de estado del enlace, el concepto de un enrutador designado en una LAN y el método de cálculo y soporte de la división de rutas y múltiples métricas. En consecuencia, hay muy poca diferencia entre el IS-IS y el OSPF. La diferencia más importante es que el IS-IS está codificado de tal manera que es fácil y natural llevar de manera simultánea información sobre varios protocolos de capa de red, característica que no está presente en el OSPF. Esta ventaja es especialmente valiosa en entornos multiprotocolo grandes.

5.2.6 Enrutamiento jerárquico

A medida que crece el tamaño de las redes, también lo hacen, de manera proporcional, las tablas de enrutamiento del enrutador. Las tablas que siempre crecen no sólo consumen memoria del enrutador, sino que también se necesita más tiempo de CPU para examinarlas y más ancho de banda para enviar informes de estado entre enrutadores. En cierto momento, la red puede crecer hasta el punto en que ya no es factible que cada enrutador tenga una entrada para cada uno de los demás enrutadores, por lo que el enrutamiento tendrá que hacerse de manera jerárquica, como ocurre en la red telefónica.

Cuando se utiliza el enrutamiento jerárquico, los enrutadores se dividen en lo que llamaremos **regiones**, donde cada enrutador conoce todos los detalles para enrutar paquetes a destinos dentro de su propia región, pero no sabe nada de la estructura interna de las otras regiones. Cuando se interconectan diferentes redes, es natural considerar cada una como región independiente a fin de liberar a los enrutadores de una red de la necesidad de conocer la estructura topológica de las demás.

En las redes enormes, una jerarquía de dos niveles puede ser insuficiente; tal vez sea necesario agrupar las regiones en clústeres, los clústeres en zonas, las zonas en grupos, etcétera, hasta que se nos agoten los nombres para clasificarlos. Como ejemplo de jerarquía multinivel, considere una posible forma de enrutar un paquete de Berkeley, California, a Malindi, Kenya. El enrutador de Berkeley conocería la topología detallada de California, pero podría enviar todo el tráfico exterior al enrutador de Los Ángeles. El enrutador de Los Ángeles podría enrutar el tráfico a otros enrutadores del país, pero enviaría el tráfico internacional a Nueva York. El enrutador de Nueva York tendría la programación para dirigir todo el tráfico al enrutador del país de destino encargado

del manejo de tráfico internacional, digamos en Nairobi. Por último, el paquete encontraría su camino por el árbol de Kenya hasta llegar a Malindi.

En la figura 5-15 se da un ejemplo cuantitativo de enrutamiento en una jerarquía de dos niveles con cinco regiones. La tabla de enrutamiento completa para el enrutador 1A tiene 17 entradas, como se muestra en la figura 5-15(b). Si el enrutamiento es jerárquico, como en la figura 5-15(c), hay entradas para todos los enrutadores locales, igual que antes, pero las demás regiones se han condensado en un solo enrutador, por lo que todo el tráfico para la región 2 va a través de la línea 1B-2A, pero el resto del tráfico remoto viaja por la línea 1C-3B. El enrutamiento jerárquico redujo la tabla de 17 entradas a 7. A medida que crece la razón entre la cantidad de regiones y el número de enrutadores por región, aumentan los ahorros de espacio de tabla.

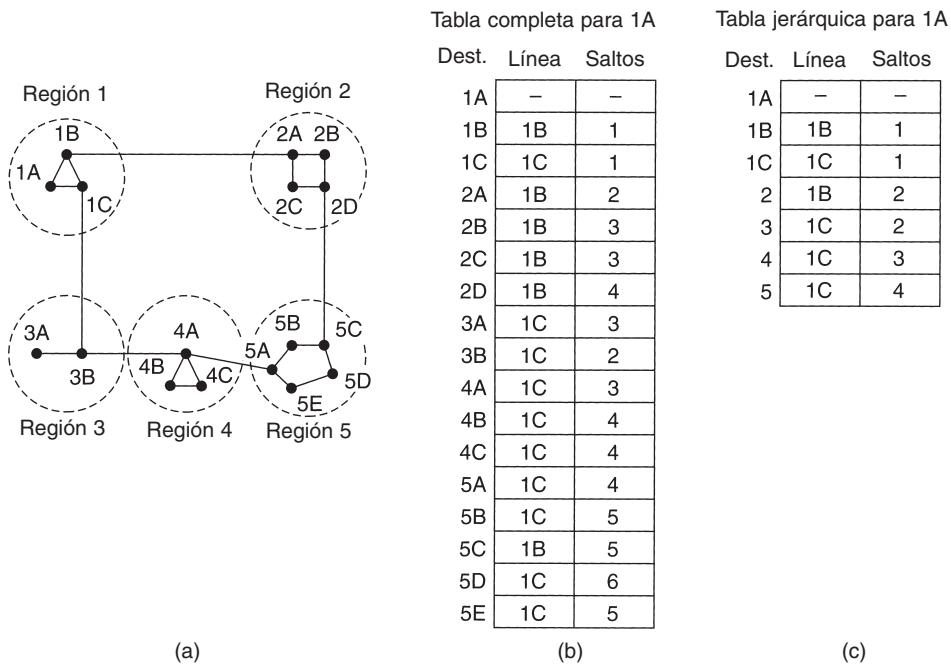


Figura 5-15. Enrutamiento jerárquico.

Desgraciadamente, estas ganancias de espacio no son gratuitas. Se paga un precio, que es una longitud de ruta mayor. Por ejemplo, la mejor ruta de 1A a 5C es a través de la región 2 pero con el enrutamiento jerárquico, todo el tráfico a la región 5 pasa por la región 3, porque eso es mejor para la mayoría de los destinos de la región 5.

Cuando una red se vuelve muy grande, surge una pregunta interesante: ¿cuántos niveles debe tener la jerarquía? Por ejemplo, considere una subred con 720 enrutadores. Si no hay jerarquía, cada enrutador necesita 720 entradas en la tabla de enrutamiento. Si dividimos la subred en 24 regiones de 30 enrutadores cada una, cada enrutador necesitará 30 entradas locales más 23 entradas remotas, lo que da un total de 53 entradas. Si elegimos una jerarquía de tres niveles, con ocho

clústeres, cada uno de los cuales contiene 9 regiones de 10 enrutadores, cada enrutador necesita 10 entradas para los enrutadores locales, 8 entradas para el enrutamiento a otras regiones dentro de su propio clúster y 7 entradas para clústeres distantes, lo que da un total de 25 entradas. Kamoun y Kleinrock (1979) descubrieron que el número óptimo de niveles para una subred de N enrutadores es de $\ln N$, y se requieren un total de $e \ln N$ entradas por enrutador. También han demostrado que el aumento en la longitud media efectiva de ruta causado por el enrutamiento jerárquico es tan pequeño que por lo general es aceptable.

5.2.7 Enrutamiento por difusión

En algunas aplicaciones, los *hosts* necesitan enviar mensajes a varios otros *hosts* o a todos los demás. Por ejemplo, el servicio de distribución de informes ambientales, la actualización de los precios de la bolsa o los programas de radio en vivo podrían funcionar mejor difundiéndolos a todas las máquinas y dejando que las que estén interesadas lean los datos. El envío simultáneo de un paquete a todos los destinos se llama **difusión**; se han propuesto varios métodos para llevarla a cabo.

Un método de difusión que no requiere características especiales de la subred es que el origen simplemente envíe un paquete distinto a todos los destinos. El método no sólo desperdicia ancho de banda, sino que también requiere que el origen tenga una lista completa de todos los destinos. En la práctica, ésta puede ser la única posibilidad, pero es el método menos deseable.

La inundación es otro candidato obvio. Aunque ésta es poco adecuada para la comunicación punto a punto ordinaria, para difusión puede merecer consideración seria, especialmente si no es aplicable ninguno de los métodos descritos a continuación. El problema de la inundación como técnica de difusión es el mismo que tiene como algoritmo de enrutamiento punto a punto: genera demasiados paquetes y consume demasiado ancho de banda.

Un tercer algoritmo es el **enrutamiento multidestino**. Con este método, cada paquete contiene una lista de destinos o un mapa de bits que indica los destinos deseados. Cuando un paquete llega al enrutador, éste revisa todos los destinos para determinar el grupo de líneas de salida que necesitará. (Se necesita una línea de salida si es la mejor ruta a cuando menos uno de los destinos.) El enrutador genera una copia nueva del paquete para cada línea de salida que se utilizará, e incluye en cada paquete sólo aquellos destinos que utilizarán la línea. En efecto, el grupo de destinos se divide entre las líneas de salida. Después de una cantidad suficiente de saltos, cada paquete llevará sólo un destino, así que puede tratarse como un paquete normal. El enrutamiento multidestino es como los paquetes con direccionamiento individual, excepto que, cuando varios paquetes deben seguir la misma ruta, uno de ellos paga la tarifa completa y los demás viajan gratis.

Un cuarto algoritmo de difusión usa explícitamente el árbol sumidero para el enrutador que inicia la difusión, o cualquier otro árbol de expansión adecuado. El **árbol de expansión** es un subgrupo de la subred que incluye todos los enrutadores pero no contiene ciclos. Si cada enrutador sabe cuáles de sus líneas pertenecen al árbol de expansión, puede copiar un paquete de entrada difundido en todas las líneas del árbol de expansión, excepto en aquella por la que llegó. Este método utiliza de manera óptima el ancho de banda, generando la cantidad mínima de paquetes necesarios para llevar a cabo el trabajo. El único problema es que cada enrutador debe tener cono-

cimiento de algún árbol de expansión para que este método pueda funcionar. Algunas veces esta información está disponible (por ejemplo, con el enrutamiento por estado del enlace), pero a veces no (por ejemplo, con el enrutamiento por vector de distancia).

Nuestro último algoritmo de difusión es un intento de aproximar el comportamiento del anterior, aun cuando los enrutadores no saben nada en lo absoluto sobre árboles de expansión. La idea, llamada **reenvío por ruta invertida** (*reverse path forwarding*), es excepcionalmente sencilla una vez planteada. Cuando llega un paquete difundido a un enrutador, éste lo revisa para ver si llegó por la línea normalmente usada para enviar paquetes *al origen* de la difusión. De ser así, hay excelentes posibilidades de que el paquete difundido haya seguido la mejor ruta desde el enrutador y, por lo tanto, sea la primera copia en llegar al enrutador. Si éste es el caso, el enrutador reenvía copias del paquete a todas las líneas, excepto a aquella por la que llegó. Sin embargo, si el paquete difundido llegó por una línea diferente de la preferida, el paquete se descarta como probable duplicado.

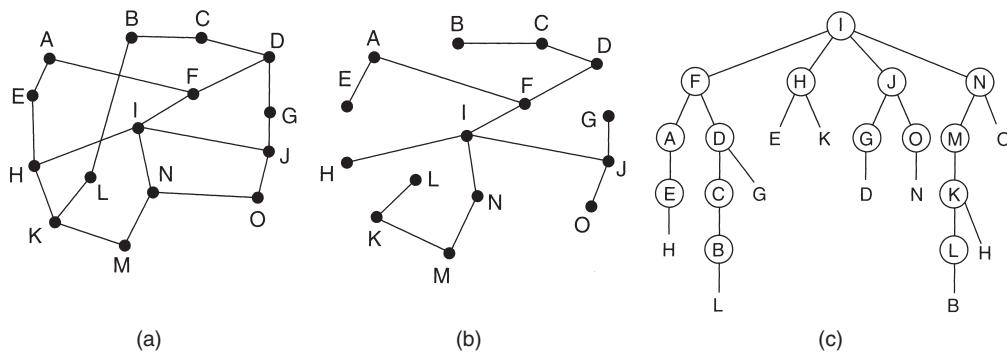


Figura 5-16. Reenvío por ruta invertida. (a) Subred. (b) Árbol sumidero. (c) Árbol construido mediante reenvío por ruta invertida.

En la figura 5-16 se muestra un ejemplo del reenvío por ruta invertida. En la parte (a) se muestra una subred, en la parte (b) se muestra un árbol sumidero para el enrutador *I* de esa subred, y en la parte (c) se muestra el funcionamiento del algoritmo de ruta invertida. En el primer salto, *I* envía paquetes a *F*, *H*, *J* y *N*, como lo indica la segunda fila del árbol. Cada uno de estos paquetes llega a *I* por la ruta preferida (suponiendo que la ruta preferida pasa a través del árbol sumidero), como lo indica el círculo alrededor de la letra. En el segundo salto, se generan ocho paquetes, dos por cada uno de los enrutadores que recibieron un paquete en el primer salto. Como resultado, los ocho llegan a enrutadores no visitados previamente, y cinco llegan a través de la línea preferida. De los seis paquetes generados en el tercer salto, sólo tres llegan por la ruta preferida (a *C*, *E* y *K*); los otros son duplicados. Después de cinco saltos y 24 paquetes, termina la difusión, en comparación con cuatro saltos y 14 paquetes si se hubiera seguido exactamente el árbol sumidero.

La ventaja principal del reenvío por ruta invertida es que es razonablemente eficiente y fácil de implementar. No requiere que los enrutadores conozcan los árboles de expansión ni tiene la sobrecarga de una lista de destinos o de un mapa de bits en cada paquete de difusión, como los

tiene el direccionamiento multidestino. Tampoco requiere mecanismos especiales para detener el proceso, como en el caso de la inundación (ya sea un contador de saltos en cada paquete y un conocimiento previo del diámetro de la subred, o una lista de paquetes ya vistos por origen).

5.2.8 Enrutamiento por multidifusión

Algunas aplicaciones requieren que procesos muy separados trabajen juntos en grupo; por ejemplo, un grupo de procesos que implementan un sistema de base de datos distribuido. En estos casos, con frecuencia es necesario que un proceso envíe un mensaje a todos los demás miembros del grupo. Si el grupo es pequeño, simplemente se puede transmitir a cada uno de los miembros un mensaje punto a punto. Si el grupo es grande, esta estrategia es costosa. A veces puede usarse la difusión, pero su uso para informar a 1000 máquinas de una red que abarca un millón de nodos es ineficiente porque la mayoría de los receptores no están interesados en el mensaje (o, peor aún, definitivamente están interesados pero no deben verlo). Por lo tanto, necesitamos una manera de enviar mensajes a grupos bien definidos de tamaño numéricamente grande, pero pequeños en comparación con la totalidad de la red.

El envío de un mensaje a uno de tales grupos se llama **multidifusión**, y su algoritmo de enrutamiento es el **enrutamiento por multidifusión**. En esta sección describiremos una manera de lograr el enrutamiento por multidifusión. Para información adicional, vea (Chu y cols., 2000; Costa y cols., 2001; Kasera y cols., 2000; Madruga y García-Luna-Aceves, 2001, y Zhang y Ryu, 2001).

Para la multidifusión se requiere administración de grupo. Se necesita alguna manera de crear y destruir grupos, y un mecanismo para que los procesos se unan a los grupos y salgan de ellos. La forma de realizar estas tareas no le concierne al algoritmo de enrutamiento. Lo que sí le concierne es que cuando un proceso se una a un grupo, informe a su *host* este hecho. Es importante que los enrutadores sepan cuáles de sus *hosts* pertenecen a qué grupos. Los *hosts* deben informar a sus enrutadores de los cambios en los miembros del grupo, o los enrutadores deben enviar de manera periódica la lista de sus *hosts*. De cualquier manera, los enrutadores aprenden qué *hosts* pertenecen a cuáles grupos. Los enrutadores les dicen a sus vecinos, de manera que la información se propaga a través de la subred.

Para realizar enrutamiento de multidifusión, cada enrutador calcula un árbol de expansión que cubre a todos los demás enrutadores de la subred. Por ejemplo, en la figura 5-17(a) tenemos una subred con dos grupos, 1 y 2. Algunos enrutadores están conectados a *hosts* que pertenecen a uno o ambos grupos, como se indica en la figura. En la figura 5-17(b) se muestra un árbol de expansión para el enrutador de la izquierda.

Cuando un proceso envía un paquete de multidifusión a un grupo, el primer enrutador examina su árbol de expansión y lo recorta, eliminando todas las líneas que conduzcan a *hosts* que no sean miembros del grupo. En el ejemplo de la figura 5-17(c) se muestra el árbol de expansión recortado del grupo 1. De la misma manera, en la figura 5-17(d) se presenta el árbol de expansión recortado del grupo 2. Los paquetes de multidifusión se reenvían sólo a través del árbol de expansión apropiado.

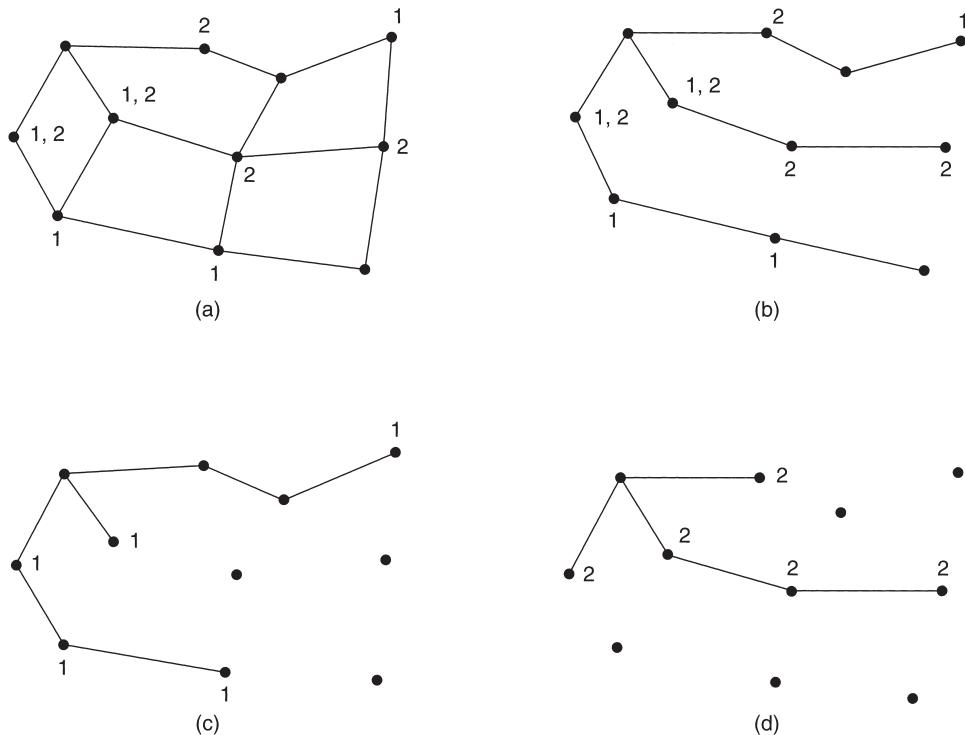


Figura 5-17. (a) Subred. (b) Árbol de expansión del enrutador del extremo izquierdo. (c) Árbol de multidifusión del grupo 1. (d) Árbol de multidifusión para el grupo 2.

Hay varias maneras de recortar el árbol de expansión. La más sencilla puede utilizarse si se maneja enrutamiento por estado del enlace, y cada enrutador está consciente de la topología completa de la subred, incluyendo qué *hosts* pertenecen a cuáles grupos. Después puede recortarse el árbol, comenzando por el final de cada ruta y trabajando hacia la raíz, eliminando todos los enrutadores que no pertenecen al grupo en cuestión.

Con el enrutamiento por vector de distancia se puede seguir una estrategia de recorte diferente. El algoritmo básico es el reenvío por ruta invertida. Sin embargo, cuando un enrutador sin *hosts* interesados en un grupo en particular y sin conexiones con otros enrutadores recibe un mensaje de multidifusión para ese grupo, responde con un mensaje de recorte (PRUNE), indicando al emisor que no envíe más multidifusiones para ese grupo. Cuando un enrutador que no tiene miembros del grupo entre sus propios *hosts* recibe uno de tales mensajes por todas las líneas, también puede responder con un mensaje de recorte. De esta forma, la subred se recorta recursivamente.

Una desventaja potencial de este algoritmo es que no escala bien en redes grandes. Suponga que una red tiene n grupos, cada uno con un promedio de m miembros. Por cada grupo se deben almacenar m árboles de expansión recortados, lo que da un total de mn árboles. Cuando hay muchos grupos grandes, se necesita bastante espacio para almacenar todos estos árboles.

Un diseño alterno utiliza **árboles de núcleo** (*core-based trees*) (Ballardie y cols., 1993). Aquí se calcula un solo árbol de expansión por grupo, con la raíz (el núcleo) cerca de la mitad del grupo. Para enviar un mensaje de multidifusión, un *host* lo envía al núcleo, que entonces hace la multidifusión a través del árbol de expansión. Aunque este árbol no será óptimo para todos los orígenes, la reducción de costos de almacenamiento de m árboles a un árbol por grupo representa un ahorro significativo.

5.2.9 Enrutamiento para *hosts* móviles

En la actualidad millones de personas tienen computadoras portátiles, y generalmente quieren leer su correo electrónico y acceder a sus sistemas de archivos normales desde cualquier lugar del mundo. Estos *hosts* móviles generan una nueva complicación: para enrutar un paquete a un *host* móvil, la red primero tiene que encontrarlo. El tema de la incorporación de *hosts* móviles en una red es muy reciente, pero en esta sección plantearemos algunos de los problemas relacionados y sugeriremos una posible solución.

En la figura 5-18 se muestra el modelo del mundo que usan generalmente los diseñadores de red. Aquí tenemos una WAN que consiste en enrutadores y *hosts*. Conectadas a la WAN hay varias LANs, MANs y celdas inalámbricas del tipo que estudiamos en el capítulo 2.

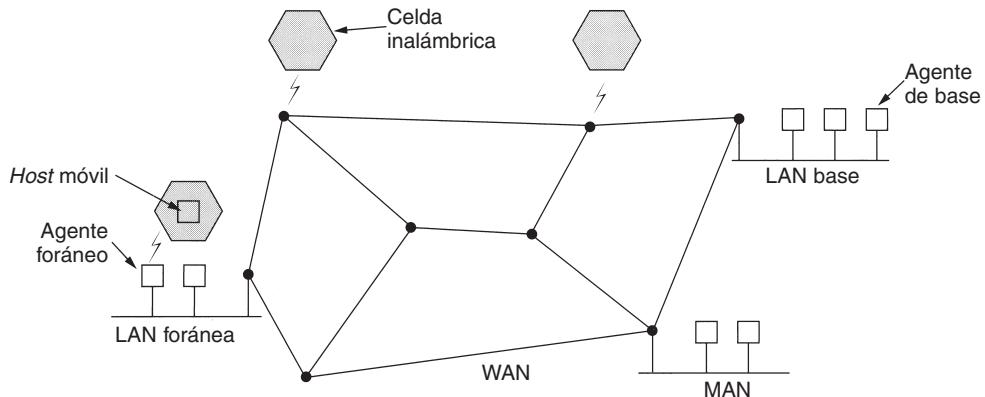


Figura 5-18. WAN a la que están conectadas LANs, MANs y celdas inalámbricas.

Se dice que los *hosts* que nunca se mueven son estacionarios; se conectan a la red mediante cables de cobre o fibra óptica. En contraste, podemos distinguir otros dos tipos de *hosts*. Los *hosts* migratorios básicamente son *hosts* estacionarios que se mueven de un lugar fijo a otro de tiempo en tiempo, pero que usan la red sólo cuando están conectados físicamente a ella. Los *hosts* ambulantes hacen su cómputo en movimiento, y necesitan mantener sus conexiones mientras se trasladan de un lado a otro. Usaremos el término ***hosts* móviles** para referirnos a cualquiera de las dos últimas categorías, es decir, a todos los *hosts* que están lejos de casa y que necesitan seguir conectados.

Se supone que todos los *hosts* tienen una **localidad base** que nunca cambia. Los *hosts* también tienen una dirección base permanente que puede servir para determinar su localidad base, de manera análoga a como el número telefónico 1-212-5551212 indica Estados Unidos (código de país 1) y Manhattan (212). La meta de enrutamiento en los sistemas con *hosts* móviles es posibilitar el envío de paquetes a *hosts* móviles usando su dirección base, y hacer que los paquetes lleguen eficientemente a ellos en cualquier lugar en el que puedan estar. Lo difícil, por supuesto, es encontrarlos.

En el modelo de la figura 5-18, el mundo se divide (geográficamente) en unidades pequeñas, a las que llamaremos áreas. Un área por lo general es una LAN o una celda inalámbrica. Cada área tiene uno o más **agentes foráneos**, los cuales son procesos que llevan el registro de todos los *hosts* móviles que visitan el área. Además, cada área tiene un **agente de base**, que lleva el registro de todos los *hosts* cuya base está en el área, pero que actualmente están visitando otra área.

Cuando un nuevo *host* entra en un área, ya sea al conectarse a ella (por ejemplo, conectándose a la LAN), o simplemente al entrar en la celda, su computadora debe registrarse con el agente foráneo de ese lugar. El procedimiento de registro funciona típicamente de esta manera:

1. Periódicamente, cada agente foráneo difunde un paquete que anuncia su existencia y dirección. Un *host* móvil recién llegado puede esperar uno de estos mensajes, pero si no llega ninguno con suficiente rapidez, el *host* móvil puede difundir un paquete que diga: “¿hay agentes foráneos por ahí?”
2. El *host* móvil se registra con el agente foráneo, dando su dirección base, su dirección actual de capa de enlace de datos y cierta información de seguridad.
3. El agente foráneo se pone en contacto con el agente de base del *host* móvil y le dice: “uno de tus *hosts* está por aquí”. El mensaje del agente foráneo al agente de base contiene la dirección de red del agente foráneo, así como la información de seguridad, para convencer al agente de base de que el *host* móvil en realidad está ahí.
4. El agente de base examina la información de seguridad, que contiene una marca de tiempo, para comprobar que fue generada en los últimos segundos. Si está conforme, indica al agente foráneo que proceda.
5. Cuando el agente foráneo recibe la confirmación de recepción del agente de base, hace una entrada en sus tablas e informa al *host* móvil que ahora está registrado.

Idealmente, cuando un *host* sale de un área, este hecho también se debe anunciar para permitir que se borre el registro, pero muchos usuarios apagan abruptamente sus computadoras cuando terminan.

Cuando un paquete se envía a un *host* móvil, se enruta a la LAN base del *host*, ya que eso es lo indicado en la dirección, como se ilustra en el paso 1 de la figura 5-19. El emisor, que se encuentra en la ciudad noreste de Seattle, desea enviar un paquete a un *host* que se encuentra normalmente en Nueva York. Los paquetes enviados al *host* móvil en su LAN base de Nueva York son interceptados por el agente de base que se encuentra ahí. A continuación, dicho agente busca la

nueva ubicación (temporal) del *host* móvil y encuentra la dirección del agente foráneo que maneja al *host* móvil, en Los Ángeles.

El agente de base entonces hace dos cosas. Primero, encapsula el paquete en el campo de carga útil de un paquete exterior y envía este último al agente foráneo (paso 2 de la figura 5-19). Este mecanismo se llama entunelamiento y lo veremos con mayor detalle después. Tras obtener el paquete encapsulado, el agente foráneo extrae el paquete original del campo de carga útil y lo envía al *host* móvil como trama de datos.

Segundo, el agente de base indica al emisor que en lo futuro envíe paquetes al *host* móvil encapsulándolos en la carga útil de paquetes explícitamente dirigidos al agente foráneo, en lugar de simplemente enviarlos a la dirección base del *host* móvil (paso 3). Los paquetes subsiguientes ahora pueden enrutarse en forma directa al usuario por medio del agente foráneo (paso 4), omitiendo la localidad base por completo.

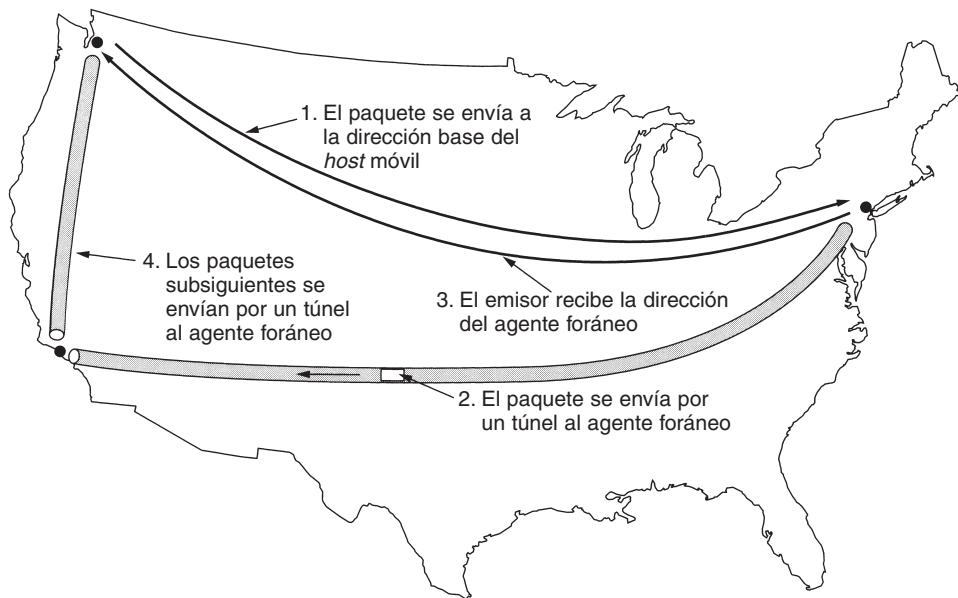


Figura 5-19. Enrutamiento de paquetes para *hosts* móviles.

Los diferentes esquemas propuestos difieren en varios sentidos. Primero está el asunto de qué parte del protocolo es llevada a cabo por los enrutadores y cuál por los *hosts* y, en este último caso, por cuál capa de los *hosts*. Segundo, en unos cuantos esquemas, los enrutadores a lo largo del camino registran las direcciones asignadas, para poder interceptarlas y redirigir el tráfico aún antes de que llegue a la dirección base. Tercero, en algunos esquemas, cada visitante recibe una dirección temporal única; en otros, la dirección temporal se refiere a un agente que maneja el tráfico de todos los visitantes.

Cuarto, los esquemas difieren en la manera en que logran realmente que los paquetes dirigidos a un destino lleguen a uno diferente. Una posibilidad es cambiar la dirección de destino y simplemente retransmitir el paquete modificado. Como alternativa, el paquete completo, con dirección base y todo, puede encapsularse en la carga útil de otro paquete enviado a la dirección temporal. Por último, los esquemas difieren en sus aspectos de seguridad. Por lo general, cuando un *host* o un enrutador recibe un mensaje del tipo “a partir de este momento, envíame por favor todo el correo de Genoveva”, puede tener dudas acerca de con quién está hablando y de si se trata de una buena idea. En (Hac y Guo, 2000; Perkins, 1998a; Snoeren y Balakrishnan, 2000; Solomon, 1998, y Wang y Chen, 2001), se analizan y comparan varios protocolos para *hosts* móviles.

5.2.10 Enrutamiento en redes *ad hoc*

Ya vimos cómo realizar el enrutamiento cuando los *hosts* son móviles y los enrutadores son fijos. Un caso aún más extremo es uno en el que los enrutadores mismos son móviles. Entre las posibilidades se encuentran:

1. Vehículos militares en un campo de batalla sin infraestructura.
2. Una flota de barcos en el mar.
3. Trabajadores de emergencia en un área donde un temblor destruyó la infraestructura.
4. Una reunión de personas con computadoras portátiles en un área que no cuenta con 802.11.

En todos estos casos, y en otros, cada nodo consiste en un enrutador y un *host*, por lo general en la misma computadora. Las redes de nodos que están cerca entre sí se conocen como **redes *ad hoc*** o **MANETs (Redes *ad hoc* Móviles)**. A continuación las examinaremos con brevedad. Para mayor información, vea (Perkins, 2001).

Lo que distingue a las redes *ad hoc* de las redes cableadas es que en las primeras se eliminaron todas las reglas comunes acerca de las topologías fijas, los vecinos fijos y conocidos, la relación fija entre direcciones IP y la ubicación, etcétera. Los enrutadores pueden ir y venir o aparecer en nuevos lugares en cualquier momento. En una red cableada, si un enrutador tiene una ruta válida a algún destino, esa ruta continúa siendo válida de manera indefinida (excepto si ocurre una falla en alguna parte del sistema que afecte a esa ruta). En una red *ad hoc*, la topología podría cambiar todo el tiempo, por lo que la necesidad o la validez de las rutas puede cambiar en cualquier momento, sin previo aviso. No es necesario decir que estas circunstancias hacen del enrutamiento en redes *ad hoc* algo diferente del enrutamiento en sus contrapartes fijas.

Se ha propuesto una variedad de algoritmos de enrutamiento para las redes *ad hoc*. Uno de los más interesantes es el algoritmo de enrutamiento **AODV (Vector de Distancia *ad hoc* bajo Demanda)** (Perkins y Royer, 1999). Es pariente lejano del algoritmo de vector de distancia Bellman-Ford pero está adaptado para funcionar en entornos móviles y toma en cuenta el ancho de banda

limitado y la duración corta de la batería en esos entornos. Otra característica inusual es que es un algoritmo bajo demanda, es decir, determina una ruta a algún destino sólo cuando alguien desea enviar un paquete a ese destino. A continuación veremos lo que eso significa.

Descubrimiento de ruta

En cualquier instante dado, una red *ad hoc* puede describirse mediante un grafo de los nodos (enrutadores + hosts). Dos nodos se conectan (es decir, tienen un arco entre ellos en el grafo) si se pueden comunicar de manera directa mediante sus radios. Debido a que uno de los dos podría tener un emisor más poderoso que el otro, es posible que *A* esté conectado a *B*, pero *B* no está conectado a *A*. Sin embargo, por simplicidad, asumiremos que todas las conexiones son simétricas. También debe hacerse notar que el simple hecho de que dos nodos estén dentro del alcance de radio entre sí no significa que estén conectados. Puede haber edificios, colinas u otros obstáculos que bloquen su comunicación.

Para describir el algoritmo, considere la red *ad hoc* de la figura 5-20, en la que un proceso del nodo *A* desea enviar un paquete al nodo *I*. El algoritmo AODV mantiene una tabla en cada nodo, codificada por destino, que proporciona información acerca de ese destino, incluyendo a cuál vecino enviar los paquetes a fin de llegar al destino. Suponga que *A* busca en sus tablas y no encuentra una entrada para *I*. Ahora tiene que descubrir una ruta a *I*. Debido a esta propiedad de descubrir rutas sólo cuando es necesario este algoritmo se conoce como “bajo demanda”.

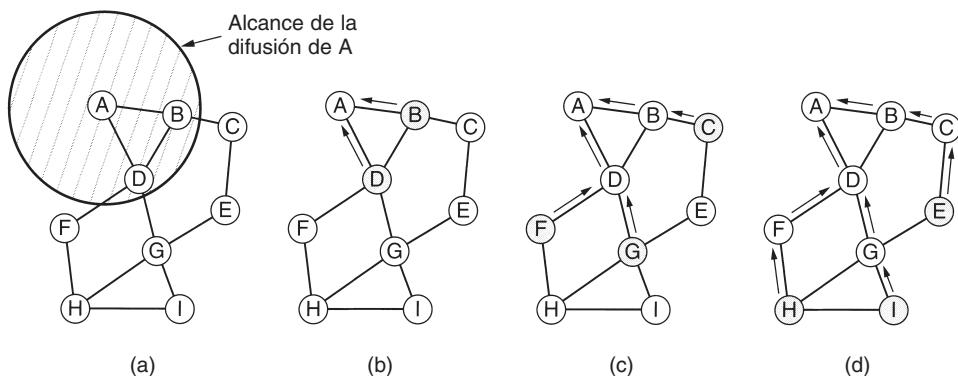


Figura 5-20. (a) Alcance de la difusión de *A*. (b) Despues de que *B* y *D* reciben la difusión de *A*. (c) Despues de que *C*, *F* y *G* reciben la difusión de *A*. (d) Despues de que *E*, *H* e *I* reciben la difusión de *A*. Los nodos sombreados son nuevos receptores. Las flechas muestran las rutas invertidas posibles.

Para localizar a *I*, *A* construye un paquete especial de solicitud de ruta (ROUTE REQUEST) y lo difunde. El paquete llega a *B* y *D*, como se ilustra en la figura 5-20(a). De hecho, la razón por la que *B* y *D* se conectan a *A* en el grafo es que pueden recibir comunicación de *A*. Por ejemplo,

F no se muestra con un arco a A porque no puede recibir la señal de radio de A . Por lo tanto, F no se conecta a A .

En la figura 5-21 se muestra el formato del paquete de solicitud de ruta. Contiene las direcciones de origen y destino, por lo general sus direcciones IP, mediante las cuales se identifica quién está buscando a quién. También contiene un *ID de solicitud*, que es un contador local que se mantiene por separado en cada nodo y se incrementa cada vez que se difunde un paquete de solicitud de ruta. En conjunto, los campos *Dirección de origen* e *ID de solicitud* identifican de manera única el paquete de solicitud de ruta a fin de que los nodos descarten cualquier duplicado que pudieran recibir.

Dirección de origen	ID de solicitud	Dirección de destino	# de secuencia de origen	# de secuencia de destino	Cuenta de saltos
---------------------	-----------------	----------------------	--------------------------	---------------------------	------------------

Figura 5-21. Formato de un paquete ROUTE REQUEST.

Además del contador *ID de solicitud*, cada nodo también mantiene un segundo contador de secuencia que se incrementa cada vez que se envía un paquete de solicitud de ruta (o una respuesta al paquete de solicitud de ruta de alguien más). Funciona de forma muy parecida a un reloj y se utiliza para indicar nuevas rutas a partir de rutas anteriores. El cuarto campo de la figura 5-21 es un contador de secuencia de A ; el quinto campo es el valor más reciente del número de secuencia de I que A ha visto (0 si nunca lo ha visto). El uso de estos campos se aclarará pronto. El último campo, *Cuenta de saltos*, mantendrá un registro de cuántos saltos ha realizado el paquete. Se inicializa a 0.

Cuando un paquete de solicitud de ruta llega a un nodo (B y D en este caso), se procesa mediante los siguientes pasos.

1. El par (*Dirección de origen*, *ID de solicitud*) se busca en una tabla de historia local para ver si esta solicitud ya se había visto y procesado. Si es un duplicado, se descarta y el procesamiento se detiene. Si no es un duplicado, el par se introduce en la tabla de historia a fin de que se puedan rechazar futuros duplicados, y el procesamiento continúa.
2. El receptor busca el destino en su tabla de enrutamiento. Si se conoce una ruta reciente al destino, se regresa un paquete de respuesta de ruta (ROUTE REPLY) al origen que le indica cómo llegar al destino (básicamente: utilízame). Reciente significa que el *Número de secuencia de destino* almacenado en la tabla de enrutamiento es mayor que o igual al *Número de secuencia de destino* del paquete de solicitud de ruta. Si es menor, la ruta almacenada es más antigua que la que el origen tenía para el destino, por lo que se ejecuta el paso 3.
3. Puesto que el receptor no conoce una ruta reciente al destino, incrementa el campo *Cuenta de saltos* y vuelve a difundir el paquete de solicitud de ruta. También extrae los datos del paquete y los almacena como una entrada nueva en su tabla de rutas invertidas. Esta información se utilizará para construir la ruta invertida a fin de que la respuesta pueda

regresar posteriormente al origen. Las flechas que se muestran en la figura 5-20 se utilizan para construir la ruta invertida. También se inicia un temporizador para la nueva entrada de ruta invertida. Si expira, la entrada se borra.

Ni B ni D saben en dónde está I , por lo tanto, cada uno de estos nodos crea una entrada de ruta invertida que apunta a A , como lo muestran las flechas de la figura 5-20, y difunde el paquete con la *Cuenta de saltos* establecida a 1. La difusión de B llega a C y D . C crea una entrada para él en su tabla de rutas invertidas y la vuelve a difundir. En contraste, D la rechaza como un duplicado. De manera similar, B rechaza la difusión de D . Sin embargo, F y G aceptan la difusión de D y la almacenan como se muestra en la figura 5-20(c). Despues de que E , H e I reciben la difusión, el paquete de solicitud de ruta finalmente alcanza un destino que sabe dónde está I , en este caso I mismo, como se ilustra en la figura 5-20(d). Observe que aunque mostramos las difusiones en tres pasos separados, las difusiones de nodos diferentes no se coordinan de ninguna forma.

En respuesta a la solicitud entrante, I construye un paquete de respuesta de ruta, como se muestra en la figura 5-22. La *Dirección de origen*, *Dirección de destino* y *Cuenta de saltos* se copian de la solicitud entrante, pero el *Número de secuencia de destino* se toma de su contador en memoria. El campo *Cuenta de saltos* se establece a 0. El campo *Tiempo de vida* controla cuánto tiempo es válida la ruta. Este paquete se difunde únicamente al nodo de donde proviene la solicitud de ruta, que en este caso es G . Despues sigue la ruta invertida a D y, finalmente, a A . En cada nodo se incrementa *Cuenta de saltos* de modo que el nodo puede ver a qué distancia está del destino (I).

Dirección de origen	Dirección de destino	# de secuencia de origen	Cuenta de saltos	Tiempo de vida
---------------------	----------------------	--------------------------	------------------	----------------

Figura 5-22. Formato de un paquete de respuesta de ruta.

El paquete se inspecciona en cada nodo intermedio del camino de regreso. Se introduce en la tabla de enrutamiento local como una ruta a I si se cumple una o más de las siguientes tres condiciones:

1. No se conoce una ruta a I .
2. El número de secuencia para I en el paquete de respuesta de ruta es mayor que el valor en la tabla de enrutamiento.
3. Los números de secuencia son iguales pero la nueva ruta es más corta.

De esta forma, todos los nodos de la ruta invertida aprenden gratis la ruta a I , gracias al descubrimiento de ruta de A . Los nodos que obtuvieron el paquete original de solicitud de ruta pero que no estaban en la ruta invertida (B , C , E , F y H en este ejemplo) descartan la entrada de la tabla de ruta invertida cuando el temporizador asociado termina.

En una red grande, el algoritmo genera muchas difusiones, incluso para los destinos que están cerca. El número de difusiones se puede reducir como se muestra a continuación. El campo *Tiempo de vida* del paquete IP se inicializa al diámetro esperado de la red y se decremente en cada salto. Si llega a 0, el paquete se descarta en lugar de difundirse.

El proceso de descubrimiento se modifica como se muestra a continuación. Para localizar un destino, el emisor difunde un paquete de solicitud de ruta con el campo *Tiempo de vida* establecido a 1. Si dentro de un tiempo razonable no se obtiene respuesta alguna, se envía otro paquete, pero esta vez con el campo *Tiempo de vida* establecido a 2. Los intentos subsiguientes utilizan 3, 4, 5, etcétera. De esta forma, la búsqueda primero se intenta de manera local y, después, en anillos cada vez más grandes.

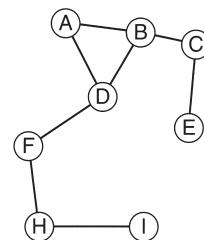
Mantenimiento de rutas

Debido a que es posible mover o apagar los nodos, la topología puede cambiar de manera espontánea. Por ejemplo, en la figura 5-20, si *G* se apaga, *A* no se dará cuenta de que la ruta a *I* (*ADGI*) que estaba utilizando ya no es válida. El algoritmo necesita ser capaz de manejar esto. Cada nodo difunde de manera periódica un mensaje de saludo (*Hello*). Se espera que cada uno de sus vecinos responda a dicho mensaje. Si no se recibe ninguna respuesta, el difusor sabe que el vecino se ha movido del alcance y ya no está conectado a él. De manera similar, si el difusor trata de enviar un paquete a un vecino que no responde, se da cuenta de que el vecino ya no está disponible.

Esta información se utiliza para eliminar rutas que ya no funcionan. Para cada destino posible, cada nodo, *N*, mantiene un registro de sus vecinos que le han proporcionado un paquete para ese destino durante los últimos ΔT segundos. Éstos se llaman **vecinos activos** de *N* para ese destino. *N* hace esto mediante una tabla de enrutamiento codificada por destino y al contener el nodo de salida a utilizar para llegar al destino, la cuenta de saltos al destino, el número de secuencia de destino más reciente y la lista de vecinos activos para ese destino. En la figura 5-23(a) se muestra una tabla de enrutamiento posible para el nodo *D* en nuestra topología de ejemplo.

Dest.	Siguiente salto	Distancia	Vecinos activos	Otros campos
A	A	1	F, G	
B	B	1	F, G	
C	B	2	F	
E	G	2		
F	F	1	A, B	
G	G	1	A, B	
H	F	2	A, B	
I	G	2	A, B	

(a)



(b)

Figura 5-23. (a) La tabla de enrutamiento de *D* antes de que *G* se apague. (b) El grafo después de que *G* se ha apagado.

Cuando cualquiera de los vecinos de N se vuelve inalcanzable, N verifica su tabla de enruteamiento para ver cuáles destinos tienen rutas en las que se incluya al vecino ahora perdido. Para cada una de estas rutas, se les informa a los vecinos activos que su ruta a través de N ahora es inválida y que se debe eliminar de sus tablas de enruteamiento. A continuación los vecinos activos indican este hecho a sus vecinos activos, y así sucesivamente y de manera recursiva, hasta que las rutas que dependen del nodo perdido se eliminan de todas las tablas de enruteamiento.

Como ejemplo del mantenimiento de ruta, considere nuestro ejemplo previo, pero ahora G se apaga de manera repentina. En la figura 5-23(b) se ilustra la topología modificada. Cuando D descubre que G ha desaparecido, busca en su tabla de enruteamiento y ve que G se utilizó en rutas a E , G e I . La unión de los vecinos activos para estos destinos es el conjunto $\{A, B\}$. En otras palabras, A y B dependen de G para algunas de sus rutas, y por esto se les tiene que informar que estas rutas ya no funcionan. D les indica este hecho enviándoles paquetes que causan que actualicen sus propias tablas de enruteamiento de manera acorde. D también elimina las entradas de E , G e I de su tabla de enruteamiento.

En nuestra descripción tal vez no sea obvio, pero una diferencia importante entre AODV y Bellman-Ford es que los nodos no envían difusiones periódicas que contengan su tabla de enruteamiento completa. Esta diferencia ahorra ancho de banda y duración de batería.

AODV también es capaz de realizar enruteamiento de difusión y multidifusión. Para mayores detalles, consulte (Perkins y Royer, 2001). El enruteamiento *ad hoc* es un área de investigación reciente. Se ha escrito bastante sobre este tema. Algunas de las obras son (Chen y cols., 2002; Hu y Johnson, 2001; Li y cols., 2001; Raju y Garcia-Luna-Aceves, 2001; Ramanathan y Redi, 2002; Royer y Toh, 1999; Spohn y Garcia-Luna-Aceves, 2001; Tseng y cols., 2001, y Zadeh y cols., 2002).

5.2.11 Búsqueda de nodos en redes de igual a igual

Las redes de igual a igual son un fenómeno relativamente nuevo, en las cuales una gran cantidad de personas, por lo general con conexiones cableadas permanentes a Internet, están en contacto para compartir recursos. La primera aplicación de uso amplio de la tecnología de igual a igual sirvió para cometer un delito masivo: 50 millones de usuarios de Napster intercambiaron canciones con derechos de autor sin el permiso de los poseedores de tales derechos hasta que las cortes cerraron Napster en medio de una gran controversia. Sin embargo, la tecnología de igual a igual tiene muchos usos interesantes y legales. También tiene algo similar a un problema de enruteamiento, aunque no como los que hemos estudiado hasta el momento. No obstante, vale la pena echarle un vistazo breve.

Lo que hace que los sistemas de igual a igual sean interesantes es que son totalmente distribuidos. Todos los nodos son simétricos y no hay control central o jerarquía. En un sistema típico de igual a igual, cada usuario tiene algo de información que podría ser de interés para otros usuarios. Esta información podría ser software (del dominio público), música, fotografías, etcétera, todos gratuitos. Si hay una gran cantidad de usuarios, éstos no se conocerán entre sí y no sabrán en dónde buscar lo que desean. Una solución es una base de datos central enorme, pero tal vez esto no sea tan factible por alguna razón (por ejemplo, nadie está dispuesto a albergarla ni a mantenerla). Por

lo tanto, el problema se reduce a qué debe hacer el usuario para encontrar un nodo que contiene lo que desea cuando no hay una base de datos centralizada o incluso un índice centralizado.

Supongamos que cada usuario tiene uno o más elementos de datos, como canciones, fotografías, programas, archivos, etcétera, que otros usuarios podrían querer leer. Cada elemento tiene una cadena ASCII que lo nombra. Un usuario potencial sólo conoce la cadena ASCII y desea averiguar si una o más personas tienen copias, y de ser así, cuáles son sus direcciones IP.

Como ejemplo, considere una base de datos genealógica distribuida. Cada genealogista tiene algunos registros en línea de sus predecesores y parientes, posiblemente con fotos, audio o incluso videoclips de las personas. Varias personas pueden tener el mismo bisabuelo, por lo que un predecesor podría tener registros en múltiples nodos. El nombre del registro es el nombre de la persona de alguna forma canónica. En algún punto, un genealogista descubre el testamento de su bisabuelo en un archivo, en el que su bisabuelo hereda su reloj de bolsillo de oro a su sobrino. El genealogista ahora sabe el nombre del sobrino y desea averiguar si otros genealogistas tienen un registro de dicho sobrino. ¿De qué manera sería posible, sin una base de datos central, saber quién, en caso de que lo hubiera, lo tiene?

Se han propuesto varios algoritmos para resolver este problema. El que examinaremos se llama Chord (Dabek y cols., 2001a, y Stoica y cols., 2001). A continuación se proporciona una explicación simplificada de cómo funciona. El sistema Chord consiste de n usuarios participantes, cada uno de los cuales podría contar con algunos registros almacenados y además está preparado para almacenar bits y fragmentos del índice para que otros usuarios los utilicen. Cada nodo de usuario tiene una dirección IP que puede generar un código de *hash* de m bits mediante una función de *hash*. Chord utiliza SHA-1 para *hash*. SHA-1 se utiliza en la criptografía; en el capítulo 8 analizaremos este tema. Por ahora, sólo es una función que toma como argumento una cadena de bytes de longitud variable y produce un número completamente aleatorio de 160 bits. Por lo tanto, podemos convertir cualquier dirección IP a un número de 160 bits llamado **identificador de nodo**.

Conceptualmente, todos los 2^{160} identificadores de nodos están organizados en orden ascendente en un gran círculo. Algunos de ellos corresponden a nodos participantes, pero la mayoría no. En la figura 5-24(a) mostramos el círculo de identificador de nodo de $m = 5$ (por el momento ignore el arco de en medio). En este ejemplo, los nodos con identificadores 1, 4, 7, 12, 15, 20 y 27 corresponden a nodos reales y están sombreados en la figura; el resto no existe.

A continuación definiremos la función *sucesor*(k) como el identificador de nodo del primer nodo real que sigue a k alrededor del círculo en el sentido de las manecillas del reloj. Por ejemplo, *sucesor*(6) = 7, *sucesor*(8) = 12 y *sucesor*(22) = 27.

A los nombres de los registros (nombres de canciones, nombres de los predecesores, etcétera) también se les aplica la función de *hash* (es decir, SHA-1) para generar un número de 160 bits, llamado **clave**. Por lo tanto, para convertir *nombre* (el nombre ASCII del registro) a su clave, utilizamos *clave* = *hash*(*nombre*). Este cálculo es simplemente una llamada de procedimiento local a *hash*. Si una persona que posee un registro genealógico para *nombre* desea ponerlo a disposición de todos, primero construye una tupla que consiste de (*nombre*, *mi-dirección-IP*) y después solicita a *sucesor*(*hash*(*nombre*)) que almacene la tupla. Si existen múltiples registros (en nodos diferentes) para este nombre, su tupla se almacenará en el mismo nodo. De esta forma, el índice se

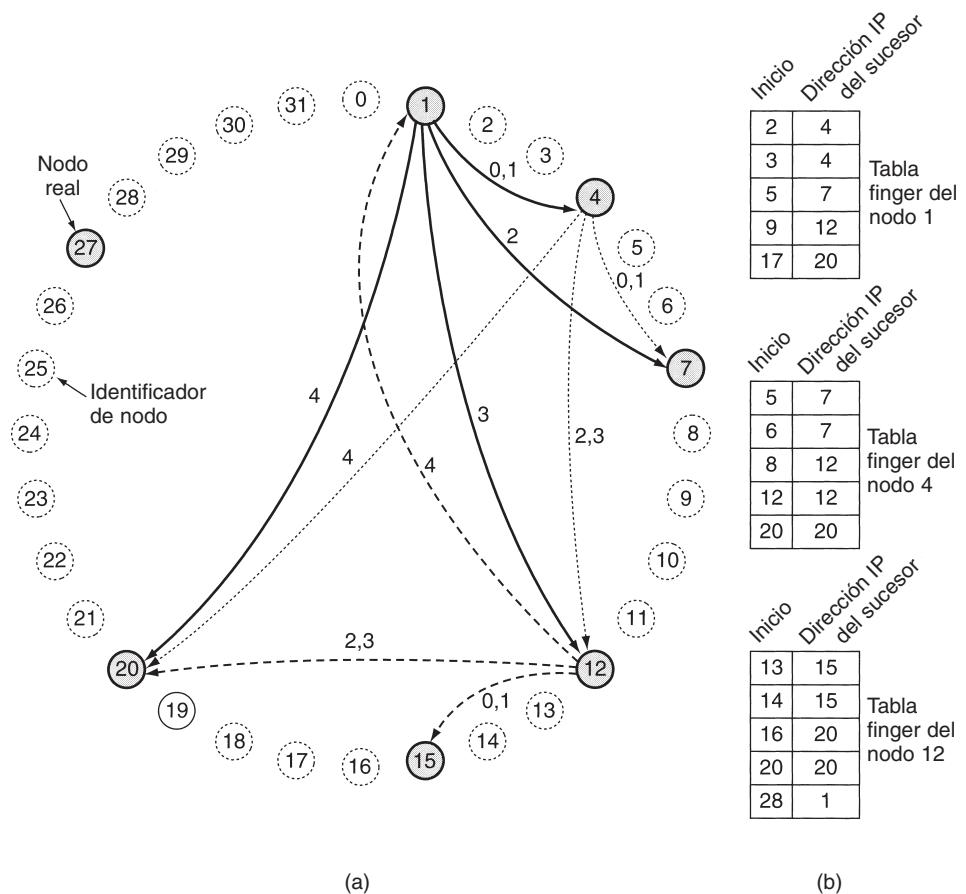


Figura 5-24. (a) Conjunto de 32 identificadores de nodos ordenados en círculo. Los sombreados corresponden a máquinas reales. Los arcos muestran los fingers de los nodos 1, 4 y 12. Las etiquetas de los arcos son los índices de las tablas. (b) Ejemplos de las tablas finger.

distribuye al azar sobre los nodos. Para tolerancia a fallas, podrían utilizarse p funciones de *hash* diferentes para almacenar cada tupla en p nodos, pero esto no lo tomaremos en cuenta aquí.

Si posteriormente algún usuario desea buscar *nombre*, le aplica la función de *hash* para obtener *clave* y después utiliza *sucesor(clave)* para encontrar la dirección IP del nodo que almacena sus tuplas de índice. El primer paso es fácil; el segundo no lo es. Para poder encontrar la dirección IP del nodo que corresponde a cierta clave, cada nodo debe mantener ciertas estructuras de datos administrativas. Una de ellas es la dirección IP de su nodo sucesor a lo largo del círculo identificador de nodo. Por ejemplo, en la figura 5-24, el sucesor del nodo 4 es 7 y el sucesor del nodo 7 es 12.

La búsqueda ahora puede ser como se muestra a continuación. El nodo solicitante envía un paquete a su sucesor, el cual contiene su dirección IP y la clave que está buscando. El paquete se propaga alrededor del anillo hasta que localiza al sucesor del identificador de nodo que se está buscando. Ese nodo verifica si tiene alguna información que corresponda con la clave y, de ser así, la regresa directamente al nodo solicitante, del cual tiene la dirección IP.

Como primera optimización, cada nodo puede contener las direcciones IP tanto de su sucesor como de su predecesor, por lo que las consultas pueden enviarse en dirección de las manecillas del reloj o en dirección contraria, dependiendo de cuál ruta se considere más corta. Por ejemplo, el nodo 7 de la figura 5-24 puede ir en dirección de las manecillas del reloj para encontrar el identificador de nodo 10, pero en dirección contraria para encontrar el identificador de nodo 3.

Incluso con dos opciones de dirección, la búsqueda lineal de todos los nodos es muy ineficiente en un sistema grande de igual a igual debido a que el número promedio de nodos requeridos por búsqueda es $n/2$. Para incrementar en forma considerable la velocidad de la búsqueda, cada nodo también mantiene lo que Chord llama una **tabla finger**. Ésta tiene m entradas, indexadas desde 0 hasta $m - 1$, y cada una apunta a un nodo real diferente. Cada una de estas entradas tiene dos campos: *inicio* y la dirección IP de *sucesor(inicio)*, como se muestra para tres nodos de ejemplo de la figura 5-24(b). Los valores de los campos para la entrada i en el nodo k son:

$$\text{inicio} = k + 2^i \text{ (módulo } 2^m\text{)}$$

Dirección IP de *sucesor(inicio [i])*

Observe que cada nodo almacena las direcciones IP de una cantidad de nodos relativamente pequeña y que la mayoría de éstos están muy cercanos en términos de identificador de nodo.

Mediante la tabla *finger*, la búsqueda de *clave* en el nodo k se realiza como se muestra a continuación. Si *clave* está entre k y *sucesor(k)*, el nodo que contiene información acerca de *clave* es *sucesor (k)* y la búsqueda termina. De lo contrario, en la tabla *finger* se busca la entrada cuyo campo *inicio* sea el predecesor más cercano de *clave*. A continuación se envía una solicitud directamente a la dirección IP de esa entrada de la tabla *finger* para pedirle que continúe la búsqueda. Debido a que dicha dirección está más cerca de la *clave*, aunque todavía está por debajo, es muy probable que pueda regresar la respuesta con tan sólo algunas consultas adicionales. De hecho, debido a que cada búsqueda divide en dos la distancia restante al destino, puede mostrarse que el número promedio de búsquedas es $\log_2 n$.

Como primer ejemplo, considere la búsqueda de *clave* = 3 en el nodo 1. Debido a que el nodo 1 sabe que 3 está entre él y su sucesor, 4, el nodo deseado es 4 y la búsqueda termina, regresando la dirección IP del nodo 4.

Como segundo ejemplo, considere la búsqueda de *clave* = 14 en el nodo 1. Debido a que 14 no está entre 1 y 4, se consulta la tabla *finger*. El predecesor más cercano a 14 es 9, por lo que la solicitud se reenvía a la dirección IP de la entrada 9, es decir, la del nodo 12. Este nodo ve que 14 está entre él y su sucesor (15), por lo que regresa la dirección IP del nodo 15.

Como tercer ejemplo, considere la búsqueda de *clave* = 16 en el nodo 1. Nuevamente, se envía una solicitud al nodo 12, pero esta vez dicho nodo no sabe la respuesta. Busca el nodo más cercano que preceda a 16 y encuentra 14, lo que resulta en la dirección IP del nodo 15. A continuación

se envía una consulta ahí. El nodo 15 observa que 16 está entre él y su sucesor (20), por lo que regresa la dirección IP de 20 al invocador, que en este caso es el nodo 1.

Puesto que los nodos se unen y separan todo el tiempo, Chord necesita una forma de manejar estas operaciones. Suponemos que cuando el sistema comenzó a operar era tan pequeño que los nodos apenas podían intercambiar información directamente para construir el primer círculo y las primeras tablas *finger*. Después de eso se necesita un procedimiento automatizado, como el que se muestra a continuación. Cuando un nuevo nodo, r , desea unirse, debe contactar a algún nodo existente y pedirle que busque la dirección IP de *sucesor(r)*. A continuación, el nuevo nodo solicita a *sucesor(r)* su predecesor. Después pide a ambos que inserten r entre ellos en el círculo. Por ejemplo, si el nodo 24 de la figura 5-24 desea unirse, pide a cualquier nodo que busque *sucesor(24)*, que es 27. A continuación pide a 27 su predecesor (20). Después de que les informa a estos dos de su existencia, 20 utiliza 24 como su sucesor y 27 utiliza 24 como su predecesor. Además, el nodo 27 entrega esas claves en el rango 21–24, que ahora pertenece a 24. En este punto, 24 está completamente insertado.

Sin embargo, ahora muchas tablas *finger* son erróneas. Para corregirlas, cada nodo ejecuta un proceso en segundo plano que recalcula de manera periódica cada *finger* invocando a *sucesor*. Cuando una de estas consultas coincide con un nuevo nodo, se actualiza la entrada correspondiente del *finger*.

Cuando un nodo se separa con éxito, entrega sus claves a su sucesor e informa a su predecesor su partida a fin de que dicho predecesor pueda enlazarse con el sucesor del nodo que se va. Cuando falla un nodo, surge un problema pues su predecesor ya no tiene un sucesor válido. Para solucionarlo, cada nodo lleva un registro no sólo de su sucesor directo sino también de sus s sucesores directos, a fin de saltar hasta $s - 1$ nodos erróneos consecutivos y volver a conectar el círculo.

Chord se ha utilizado para construir un sistema de archivos distribuido (Dabek y cols., 2001b) y otras aplicaciones, y las investigaciones continúan. En (Rowstron y Druschel, 2001a, y Rowstron y Druschel, 2001b), se describe un sistema de igual a igual diferente, Pastry, así como sus aplicaciones. En (Clarke y cols., 2002) se analiza un tercer sistema de igual a igual, Freenet. En (Ratnasamy y cols., 2001) se describe un cuarto sistema de este tipo.

5.3 ALGORITMOS DE CONTROL DE CONGESTIÓN

Cuando hay demasiados paquetes presentes en la subred (o en una parte de ella), hay una degradación del desempeño. Esta situación se llama **congestión**. En la figura 5-25 se muestra este síntoma. Cuando la cantidad de paquetes descargados en la subred por los *hosts* está dentro de su capacidad de conducción, todos se entregan (excepto unos pocos afligidos por errores de transmisión) y la cantidad entregada es proporcional al número enviado. Sin embargo, a medida que aumenta el tráfico, los enrutadores ya no pueden manejarlo y comienzan a perder paquetes. Esto tiende a empeorar las cosas. Con mucho tráfico, el desempeño se desploma por completo y casi no hay entrega de paquetes.

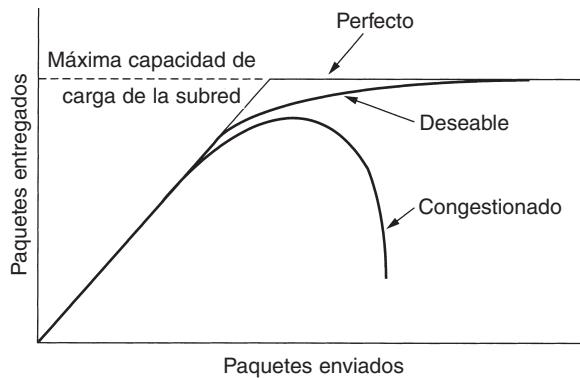


Figura 5-25. Cuando se genera demasiado tráfico, ocurre congestión y se degrada marcadamente el desempeño.

La congestión puede ocurrir por varias razones. Si de manera repentina comienzan a llegar cadenas de paquetes por tres o cuatro líneas de entrada y todas necesitan la misma línea de salida, se generará una cola. Si no hay suficiente memoria para almacenar a todos los paquetes, algunos de ellos se perderán. La adición de memoria puede ayudar hasta cierto punto, pero Nagle (1987) descubrió que si los enrutadores tienen una cantidad infinita de memoria, la congestión empeora en lugar de mejorar, ya que para cuando los paquetes llegan al principio de la cola, su temporizador ha terminado (repetidamente) y se han enviado duplicados. Todos estos paquetes serán debidamente reenviados al siguiente enrutador, aumentando la carga en todo el camino hasta el destino.

Los procesadores lentos también pueden causar congestión. Si las CPUs de los enrutadores son lentas para llevar a cabo las tareas de administración requeridas (búferes de encolamiento, actualización de tablas, etcétera), las colas pueden alargarse, aun cuando haya un exceso de capacidad de línea. De la misma manera, las líneas de poco ancho de banda también pueden causar congestión. La actualización de las líneas sin cambiar los procesadores, o viceversa, por lo general ayuda un poco, pero con frecuencia simplemente desplaza el cuello de botella. Además, actualizar sólo parte de un sistema simplemente mueve el cuello de botella a otra parte. El problema real con frecuencia es un desajuste entre partes del sistema. Este problema persistirá hasta que todos los componentes estén en equilibrio.

Vale la pena indicar de manera explícita la diferencia entre el control de la congestión y el control de flujo, pues la relación es sutil. El control de congestión se ocupa de asegurar que la subred sea capaz de transportar el tráfico ofrecido. Es un asunto global, en el que interviene el comportamiento de todos los *hosts*, todos los enrutadores, el proceso de almacenamiento y reenvío dentro de los enrutadores y todos los demás factores que tienden a disminuir la capacidad de transporte de la subred.

En contraste, el control de flujo se relaciona con el tráfico punto a punto entre un emisor dado y un receptor dado. Su tarea es asegurar que un emisor rápido no pueda transmitir datos de manera continua a una velocidad mayor que la que puede absorber el receptor. El control de flujo casi

siempre implica una retroalimentación directa del receptor al emisor, para indicar al emisor cómo van las cosas en el otro lado.

Para captar la diferencia entre estos dos conceptos, considere una red de fibra óptica con una capacidad de 1000 gigabits/seg en la que una supercomputadora está tratando de transferir un archivo a una computadora personal que opera a 1 Gbps. Aunque no hay congestión (la red misma no está en problemas), se requiere control de flujo para obligar a la supercomputadora a detenerse con frecuencia para darle a la computadora personal un momento de respiro.

En el otro extremo, considere una red de almacenamiento y reenvío con líneas de 1 Mbps y 1000 computadoras grandes, la mitad de las cuales trata de transferir archivos a 100 kbps a la otra mitad. Aquí el problema no es que los emisores rápidos saturen a los receptores lentos, sino simplemente que el tráfico ofrecido total excede lo que la red puede manejar.

La razón por la que se confunden con frecuencia el control de congestión y el control de flujo es que algunos algoritmos de control de congestión operan enviando mensajes de regreso a varios orígenes, indicándoles que reduzcan su velocidad cuando la red se mete en problemas. Por lo tanto, un *host* puede recibir un mensaje de “reducción de velocidad” porque el receptor no puede manejar la carga o porque la red no la puede manejar. Más adelante regresaremos a este punto.

Comenzaremos nuestro estudio del control de congestión examinando un modelo general para manejarlo. Luego veremos métodos generales para prevenirlo. Después de eso, veremos varios algoritmos dinámicos para manejarlo una vez que se ha establecido.

5.3.1 Principios generales del control de congestión

Muchos problemas de los sistemas complejos, como las redes de computadoras, pueden analizarse desde el punto de vista de una teoría de control. Este método conduce a dividir en dos grupos todas las soluciones: de ciclo abierto y de ciclo cerrado. En esencia, las soluciones de ciclo abierto intentan resolver el problema mediante un buen diseño, para asegurarse en primer lugar de que no ocurra. Una vez que el sistema está en funcionamiento, no se hacen correcciones a medio camino.

Las herramientas para llevar a cabo control de ciclo abierto incluyen decidir cuándo aceptar tráfico nuevo, decidir cuándo descartar paquetes, y cuáles, y tomar decisiones de calendarización en varios puntos de la red. Todas tienen en común el hecho de que toman decisiones independientemente del estado actual de la red.

En contraste, las soluciones de ciclo cerrado se basan en el concepto de un ciclo de retroalimentación. Este método tiene tres partes cuando se aplica al control de congestión:

1. Monitorear el sistema para detectar cuándo y dónde ocurren congestiones.
2. Pasar esta información a lugares en los que puedan llevarse a cabo acciones.
3. Ajustar la operación del sistema para corregir el problema.

Es posible utilizar varias métricas para monitorear la subred en busca de congestiones. Las principales son el porcentaje de paquetes descartados debido a falta de espacio de búfer, la longitud

promedio de las colas, la cantidad de paquetes para los cuales termina el temporizador y se transmiten de nueva cuenta, el retardo promedio de los paquetes y la desviación estándar del retardo de paquete. En todos los casos, un aumento en las cifras indica un aumento en la congestión.

El segundo paso del ciclo de retroalimentación es la transferencia de información relativa a la congestión desde el punto en que se detecta hasta el punto en que puede hacerse algo al respecto. La manera más obvia es que el enrutador que detecta la congestión envíe un paquete al origen (u orígenes) del tráfico, anunciando el problema. Por supuesto, estos paquetes adicionales aumentan la carga precisamente en el momento en que no se necesita más carga, es decir, cuando la subred está congestionada.

Por fortuna, existen otras opciones. Por ejemplo, en cada paquete puede reservarse un bit o campo para que los enrutadores lo llenen cuando la congestión rebasa algún umbral. Cuando un enrutador detecta este estado congestionado, llena el campo de todos los paquetes de salida, para avisar a los vecinos.

Otra estrategia es hacer que los *hosts* o enrutadores envíen de manera periódica paquetes de sondeo para preguntar explícitamente sobre la congestión. Esta información puede usarse para enrutar el tráfico fuera de áreas con problemas. Algunas estaciones de radio tienen helicópteros que vuelan sobre la ciudad para informar de la congestión en las calles, con la esperanza de que los escuchas enrutarán sus paquetes (autos) fuera de las zonas conflictivas.

En todos los esquemas de retroalimentación, la esperanza es que el conocimiento sobre la congestión hará que los *hosts* emprendan acciones adecuadas con miras a reducir la congestión. Para operar en forma correcta, la escala de tiempo debe ajustarse con cuidado. Si el enrutador grita ALTO cada vez que llegan dos paquetes seguidos, y SIGA, cada vez que está inactivo durante 20 μ seg, el sistema oscilará sin control y nunca convergerá. Por otra parte, si un enrutador espera 30 minutos para asegurarse antes de tomar una decisión, el mecanismo de control de congestión reaccionará tan lentamente que no será de utilidad. Para funcionar bien se requiere un justo medio, pero encontrar la constante de tiempo correcta no es un asunto trivial.

Se conocen muchos algoritmos de control de congestión. A fin de organizarlos lógicamente, Yang y Reddy (1995) han desarrollado una taxonomía de los algoritmos de control de congestión. Comienzan por dividir todos los algoritmos en ciclo abierto y ciclo cerrado, como se describió anteriormente. Dividen todavía más los algoritmos de ciclo abierto en algoritmos que actúan en el origen y los que actúan en el destino. Los algoritmos de ciclo cerrado también se dividen en dos subcategorías: retroalimentación explícita e implícita. En los algoritmos de retroalimentación explícita, regresan paquetes desde el punto de congestión para avisar al origen. En los algoritmos implícitos, el origen deduce la existencia de una congestión haciendo observaciones locales, como el tiempo necesario para que regresen las confirmaciones de recepción.

La presencia de congestión significa que la carga es (temporalmente) superior (en una parte del sistema) a la que pueden manejar los recursos. Vienen a la mente dos soluciones: aumentar los recursos o disminuir la carga. Por ejemplo, la subred puede comenzar a utilizar líneas de acceso telefónico para aumentar de manera temporal el ancho de banda entre ciertos puntos. En los sistemas satelitales la potencia de transmisión creciente con frecuencia reditúa un ancho de banda más alto. La división del tráfico entre varias rutas en lugar de usar siempre la mejor también aumenta efectivamente el ancho de banda. Por último, a fin de contar con mayor capacidad, los enrutadores

de repuesto que normalmente sirven sólo como respaldo (para hacer que el sistema sea tolerante a fallas), pueden ponerse en línea cuando aparece una congestión severa.

Sin embargo, a veces no es posible aumentar la capacidad, o ya ha sido aumentada al máximo. Entonces, la única forma de combatir la congestión es disminuir la carga. Existen varias maneras de reducir la carga, como negar el servicio a algunos usuarios, degradar el servicio para algunos o todos los usuarios y obligar a los usuarios a programar sus solicitudes de una manera más predecible.

Algunos de estos métodos, que estudiaremos en breve, se aplican mejor a los circuitos virtuales. En las subredes que usan circuitos virtuales de manera interna, estos métodos pueden usarse en la capa de red. En las subredes de datagramas algunas veces se pueden utilizar en las conexiones de capa de transporte. En este capítulo nos enfocaremos en su uso en la capa de red. En el siguiente veremos lo que puede hacerse en la capa de transporte para manejar la congestión.

5.3.2 Políticas de prevención de congestión

Comencemos nuestro estudio de los métodos de control de congestión estudiando los sistemas de ciclo abierto. Estos sistemas están diseñados para reducir al mínimo la congestión desde el inicio, en lugar de permitir que ocurra y reaccionar después del hecho. Tratan de lograr su objetivo usando políticas adecuadas en varios niveles. En la figura 5-26 vemos diferentes políticas para las capas de enlace de datos, red y transporte que pueden afectar a la congestión (Jain, 1990).

Capa	Políticas
Transporte	<ul style="list-style-type: none"> • Política de retransmisión • Política de almacenamiento en caché de paquetes fuera de orden • Política de confirmaciones de recepción • Política de control de flujo • Determinación de terminaciones de temporizador
Red	<ul style="list-style-type: none"> • Circuitos virtuales vs. datagramas en la subred • Política de encolamiento y servicio de paquetes • Política de descarte de paquetes • Algoritmo de enrutamiento • Administración de tiempo de vida del paquete
Enlace de datos	<ul style="list-style-type: none"> • Política de retransmisiones • Política de almacenamiento en caché de paquetes fuera de orden • Política de confirmación de recepción • Política de control de flujo

Figura 5-26. Políticas relacionadas con la congestión.

Comencemos por la capa de enlace de datos y avancemos hacia arriba. La política de retransmisiones tiene que ver con la rapidez con la que un emisor termina de temporizar y con lo que transmite al ocurrir una terminación de temporizador. Un emisor nervioso que a veces termina de temporizar demasiado pronto y retransmite todos los paquetes pendientes usando el protocolo de

retroceso n impondrá una carga más pesada al sistema que un emisor calmado que usa repetición selectiva. La política de almacenamiento en caché está muy relacionada con esto. Si los receptores descartan de manera rutinaria todos los paquetes que llegan fuera de orden, posteriormente se tendrán que enviar otra vez, lo que creará una carga extra. Con respecto al control de congestión, la repetición selectiva es mejor que el retroceso n.

La política de confirmación de recepción también afecta la congestión. Si la recepción de cada paquete se confirma de inmediato, los paquetes de confirmación de recepción generan tráfico extra. Sin embargo, si se guardan las confirmaciones de recepción para sobreponerlas en el tráfico en sentido inverso, pueden resultar en terminaciones de temporizador y retransmisiones extra. Un esquema de control de flujo estricto (por ejemplo, una ventana pequeña) reduce la tasa de datos y permite, por lo tanto, atacar la congestión.

En la capa de red, la decisión entre circuitos virtuales y datagramas afecta la congestión, ya que muchos algoritmos de control de congestión sólo funcionan con subredes de circuitos virtuales. La política de encolamiento y servicio de paquetes se refiere a que los enrutadores tengan una cola por línea de entrada, y una o varias colas por línea de salida. También se relaciona con el orden en que se procesan los paquetes (por ejemplo, en *round robin* o con base en prioridades). La política de descarte es la regla que indica qué paquete se descarta cuando no hay espacio. Una buena política puede ayudar a aliviar la congestión y una mala puede hacerlo peor.

Un buen algoritmo de enrutamiento puede evitar la congestión si distribuye el tráfico entre todas las líneas, pero uno malo puede enviar demasiado tráfico por líneas ya congestionadas. Por último, la administración del tiempo de vida de los paquetes se encarga del tiempo que puede existir un paquete antes de ser descartado. Si este tiempo es demasiado grande, los paquetes perdidos pueden bloquear la operación durante un buen rato, pero si es demasiado corto, los paquetes pueden expirar antes de llegar a su destino, lo que provoca retransmisiones.

En la capa de transporte surgen los mismos problemas que en la capa de enlace de datos, pero además es más difícil la determinación del intervalo de expiración, porque el tiempo de tránsito a través de la red es menos predecible que el tiempo de tránsito por un cable entre dos enrutadores. Si el intervalo es demasiado corto, se enviarán paquetes extra de manera innecesaria. Si es muy largo, se reducirá la congestión, pero el tiempo de respuesta se verá afectado cada vez que se pierda un paquete.

5.3.3 Control de congestión en subredes de circuitos virtuales

Los métodos de control de congestión antes descritos son básicamente de ciclo abierto: tratan de evitar que ocurran las congestiones, en lugar de manejarlas una vez ocurridas. En esta sección describiremos algunos métodos para el control dinámico de la congestión en las subredes de circuitos virtuales. En las siguientes dos veremos técnicas que pueden usarse en cualquier subred.

Una de las técnicas que se usa ampliamente para evitar que empeoren las congestiones que ya han comenzado es el **control de admisión**. La idea es sencilla: una vez que se ha detectado la congestión, no se establecen circuitos virtuales nuevos hasta que ha desaparecido el problema. Por

lo tanto, fallan los intentos por establecer conexiones nuevas de capa de transporte. Permitir el acceso a más personas simplemente empeoraría las cosas. Aunque este método es simple, su implementación es sencilla. En el sistema telefónico, cuando un conmutador se sobrecarga también se pone en práctica el control de admisión, al no darse tonos de marcado.

Un método alterno es permitir el establecimiento de nuevos circuitos virtuales, pero enrutando cuidadosamente los circuitos nuevos por otras rutas que no tengan problemas. Por ejemplo, considere la subred de la figura 5-27(a), en la que dos enrutadores están congestionados.

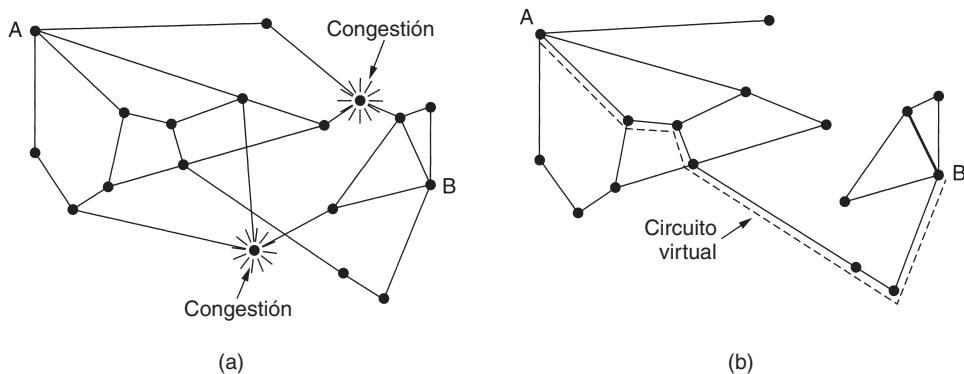


Figura 5-27. (a) Subred congestionada. (b) Subred redibujada que elimina la congestión. También se muestra un circuito virtual de *A* a *B*.

Suponga que un *host* conectado al enrutador *A* quiere establecer una conexión con un *host* conectado al enrutador *B*. Normalmente esta conexión pasaría a través de uno de los enrutadores congestionados. Para evitar esta situación, podemos redibujar la subred como se muestra en la figura 5-27(b), omitiendo los enrutadores congestionados y todas sus líneas. La línea punteada muestra una ruta posible para el circuito virtual que evita los enrutadores congestionados.

Otra estrategia que tiene que ver con los circuitos virtuales es negociar un acuerdo entre el *host* y la subred cuando se establece un circuito virtual. Este arreglo normalmente especifica el volumen y la forma del tráfico, la calidad de servicio requerida y otros parámetros. Para cumplir con su parte del acuerdo, la subred por lo general reservará recursos a lo largo de la ruta cuando se establezca el circuito. Estos recursos pueden incluir espacio en tablas y en búfer en los enrutadores y ancho de banda en las líneas. De esta manera, es poco probable que ocurran congestiones en los circuitos virtuales nuevos, porque está garantizada la disponibilidad de todos los recursos necesarios.

Este tipo de reserva puede llevarse a cabo todo el tiempo como procedimiento operativo estándar, o sólo cuando la subred está congestionada. Una desventaja de hacerlo todo el tiempo es que se tiende a desperdiciar recursos. Si seis circuitos virtuales que podrían usar 1 Mbps pasan por la misma línea física de 6 Mbps, la línea tiene que marcarse como llena, aunque pocas veces ocurrirá que los seis circuitos virtuales transmitan a toda velocidad al mismo tiempo. En consecuencia, el precio del control de congestión es un ancho de banda sin utilizar (es decir, desperdiciado).

5.3.4 Control de congestión en subredes de datagramas

Veamos ahora un enfoque que puede usarse en subredes de datagramas (y también en subredes de circuitos virtuales). Cada enrutador puede monitorear fácilmente el uso de sus líneas de salida y de otros recursos. Por ejemplo, puede asociar cada línea a una variable real, u , cuyo valor, entre 0.0 y 1.0, refleja el uso reciente de esa línea. Para tener una buena estimación de u , puede tomarse periódicamente una muestra del uso instantáneo de la línea, $f(0 \text{ o } 1)$, y actualizar u periódicamente de acuerdo con

$$u_{\text{nvo}} = au_{\text{ant}} + (1 - a)f$$

donde la constante a determina la rapidez con que el enrutador olvida la historia reciente.

Siempre que u rebasa el umbral, la línea de salida entra en un estado de “advertencia”. Cada paquete nuevo que llega se revisa para ver si su línea de salida está en el estado de advertencia. Si es así, se realiza alguna acción. Ésta puede ser una de varias alternativas, que analizaremos a continuación.

El bit de advertencia

La arquitectura DECNET antigua señalaba el estado de advertencia activando un bit especial en el encabezado del paquete. Frame relay también lo hace. Cuando el paquete llegaba a su destino, la entidad transportadora copiaba el bit en la siguiente confirmación de recepción que se regresaba al origen. A continuación el origen reducía el tráfico.

Mientras el enrutador estuviera en el estado de advertencia, continuaba activando el bit de advertencia, lo que significaba que el origen continuaba obteniendo confirmaciones de recepción con dicho bit activado. El origen monitoreaba la fracción de confirmaciones de recepción con el bit activado y ajustaba su tasa de transmisión de manera acorde. En tanto los bits de advertencia continuaran fluyendo, el origen continuaba disminuyendo su tasa de transmisión. Cuando disminuía lo suficiente, el origen incrementaba su tasa de transmisión. Observe que debido a que cada enrutador a lo largo de la ruta podía activar el bit de advertencia, el tráfico se incrementaba sólo cuando no había enrutadores con problemas.

Paquetes reguladores

El algoritmo de control de congestión anterior es muy sutil. Utiliza medios indirectos para indicar al origen que vaya más despacio. ¿Por qué no indicárselo de manera directa? En este método, el enrutador regresa un **paquete regulador** al *host* de origen, proporcionándole el destino encontrado en el paquete. El paquete original se etiqueta (se activa un bit del encabezado) de manera que no genere más paquetes reguladores más adelante en la ruta y después se reenvía de la manera usual.

Cuando el *host* de origen obtiene el paquete regulador, se le pide que reduzca en un porcentaje X el tráfico enviado al destino especificado. Puesto que otros paquetes dirigidos al mismo destino probablemente ya están en camino y generarán más paquetes reguladores, el *host* debe ignorar

los paquetes reguladores que se refieran a ese destino por un intervalo fijo de tiempo. Una vez que haya expirado ese tiempo, el *host* escucha más paquetes reguladores durante otro intervalo. Si llega alguno, la línea todavía está congestionada, por lo que el *host* reduce el flujo aún más y comienza a ignorar nuevamente los paquetes reguladores. Si no llega ningún paquete de este tipo durante el periodo de escucha, el *host* puede incrementar el flujo otra vez. La retroalimentación implícita de este protocolo puede ayudar a evitar la congestión que aún no estrangula ningún flujo a menos que ocurra un problema.

Los *hosts* pueden reducir el tráfico ajustando los parámetros de sus políticas, por ejemplo, su tamaño de ventana. Por lo general, el primer paquete regulador causa que la tasa de datos se reduzca en 0.50 con respecto a su tasa anterior, el siguiente causa una reducción de 0.25, etcétera. Los incrementos se dan en aumentos más pequeños para evitar que la congestión se vuelva a generar rápidamente.

Se han propuesto algunas variaciones de este algoritmo de control de congestión. En una, los enrutadores pueden mantener varios umbrales. Dependiendo de qué umbral se ha rebasado, el paquete regulador puede contener una advertencia suave, una severa o un ultimátum.

Otra variación es utilizar como señal de activación tamaños de colas o utilización de los búferes en lugar de la utilización de la línea. Es posible utilizar la misma ponderación exponencial con esta métrica como con u , por supuesto.

Paquetes reguladores de salto por salto

A altas velocidades o distancias grandes, el envío de un paquete regulador a los *hosts* de origen no funciona bien porque la reacción es muy lenta. Por ejemplo, considere a un *host* en San Francisco (enrutador *A* de la figura 5-28) que está enviando tráfico a un *host* en Nueva York (enrutador *D* de la figura 5-28) a 155 Mbps. Si al *host* de Nueva York se le comienza a terminar el espacio de búfer, un paquete regulador tardará unos 30 msec en regresar a San Francisco para indicarle que disminuya su velocidad. La propagación de paquetes reguladores se muestra en el segundo, tercer y cuarto pasos de la figura 5-28(a). En esos 30 msec se habrán enviado otros 4.6 megabits. Aun si el *host* de San Francisco se desactiva de inmediato, los 4.6 megabits en la línea continuarán fluyendo, y habrá que encargarse de ellos. Sólo hasta el séptimo diagrama de la figura 5-28(a) el enrutador de Nueva York notará un flujo más lento.

Un método alterno es hacer que el paquete regulador ejerza su efecto en cada salto que dé, como se muestra en la secuencia de la figura 5-28(b). Aquí, una vez que el paquete regulador llega a *F*, se obliga a *F* a reducir el flujo a *D*. Hacerlo requerirá que *F* destine más búferes al flujo, ya que el origen aún está transmitiendo a toda velocidad, pero da a *D* un alivio inmediato, como en un mensaje comercial de un remedio contra el dolor de cabeza. En el siguiente paso, el paquete regulador llega a *E*, e indica a éste que reduzca el flujo a *F*. Esta acción impone una mayor carga a los búferes de *E*, pero da un alivio inmediato a *F*. Por último, el paquete regulador llega a *A* y efectivamente se reduce el flujo.

El efecto neto de este esquema de salto por salto es proporcionar un alivio rápido al punto de congestión, a expensas de usar más búferes ascendentes. De esta manera puede cortarse la congestión

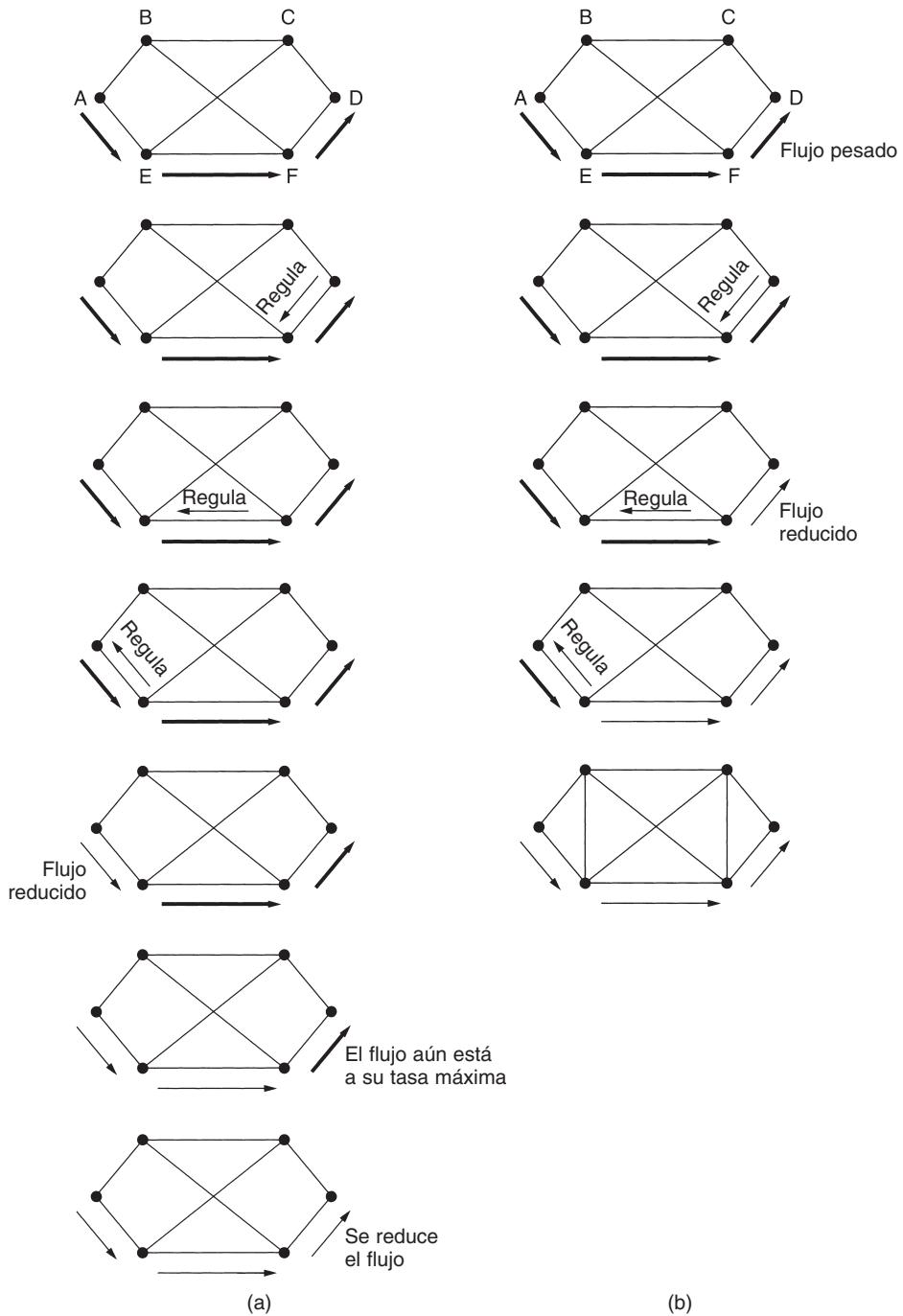


Figura 5-28. (a) Paquete regulador que afecta sólo al origen. (b) Paquete regulador que afecta cada salto por el que pasa.

en la raíz, sin que se pierdan paquetes. La idea se estudia con mayor detalle en Mishra y Kanakia, 1992.

5.3.5 Desprendimiento de carga

Cuando ninguno de los métodos anteriores elimina la congestión, los enrutadores pueden sacar la artillería pesada: el **desprendimiento de carga**, que es una manera rebuscada de decir que, cuando se inunda a los enrutadores con paquetes que no pueden manejar, simplemente los tiran. El término viene del mundo de la generación de energía eléctrica, donde se refiere a la práctica de instalaciones que intencionalmente producen apagones en ciertas áreas para salvar a la red completa de venirse abajo en días calurosos de verano en los que la demanda de energía eléctrica excede por mucho el suministro.

Un enrutador abrumado por paquetes simplemente puede escoger paquetes al azar para desprenderse de ellos, pero normalmente puede hacer algo mejor. El paquete a descartar puede depender de las aplicaciones que se estén ejecutando. En la transferencia de archivos vale más un paquete viejo que uno nuevo, pues el deshacerse del paquete 6 y mantener los paquetes 7 a 10 causará un hueco en el receptor que podría obligar a que se retransmitan los paquetes 6 a 10 (si el receptor descarta de manera rutinaria los paquetes en desorden). En un archivo de 12 paquetes, deshacerse del paquete 6 podría requerir la retransmisión de los paquetes 7 a 12, y deshacerse del 10 podría requerir la retransmisión sólo del 10 al 12. En contraste, en multimedia es más importante un paquete nuevo que uno viejo. La política anterior (más viejo es mejor que más nuevo), con frecuencia se llama **vino**, y la última (más nuevo es mejor que más viejo) con frecuencia se llama **leche**.

Un paso adelante de esto en cuanto a inteligencia requiere la cooperación de los emisores. En muchas aplicaciones, algunos paquetes son más importantes que otros. Por ejemplo, ciertos algoritmos de compresión de vídeo transmiten periódicamente una trama entera y sus tramas subsiguientes como diferencias respecto a la última trama completa. En este caso, es preferible desprenderse de un paquete que es parte de una diferencia que desprenderse de uno que es parte de una trama completa. Como otro ejemplo, considere la transmisión de un documento que contiene texto ASCII e imágenes. La pérdida de una línea de píxeles de una imagen es mucho menos dañina que la pérdida de una línea de texto legible.

Para poner en práctica una política inteligente de descarte, las aplicaciones deben marcar sus paquetes con clases de prioridades para indicar su importancia. Si lo hacen, al tener que descartar paquetes, los enrutadores pueden descartar primero los paquetes de clase más baja, luego los de la siguiente clase más baja, etcétera. Por supuesto, a menos que haya una razón poderosa para marcar los paquetes como MUY IMPORTANTE–NUNCA DESCARTAR, nadie lo hará.

La razón podría ser monetaria, siendo más barato el envío de paquetes de baja prioridad que el de los de alta prioridad. Como alternativa, los emisores podrían tener permitido enviar paquetes de alta prioridad bajo condiciones de carga ligera, pero a medida que aumente la carga, los paquetes podrían descartarse, lo que haría que los usuarios ya no siguieran enviándolos.

Otra opción es permitir que los *hosts* excedan los límites especificados en el acuerdo negociado al establecer el circuito virtual (por ejemplo, usar un ancho de banda mayor que el permitido),

pero sujetos a la condición de que el exceso de tráfico se marque con prioridad baja. Tal estrategia de hecho no es mala idea, porque utiliza con mayor eficiencia los recursos inactivos, permitiendo que los *hosts* los utilicen siempre y cuando nadie más esté interesado, pero sin establecer un derecho sobre ellos cuando los tiempos se vuelven difíciles.

Detección temprana aleatoria

Es bien sabido que tratar con la congestión después de que se detecta por primera vez es más efectivo que dejar que dañe el trabajo y luego tratar de solucionarlo. Esta observación conduce a la idea de descartar paquetes antes de que se ocupe todo el espacio de búfer. Un algoritmo popular para realizar esto se conoce como **RED (detección temprana aleatoria)** (Floyd y Jacobson, 1993). En algunos protocolos de transporte (entre ellos TCP), la respuesta a paquetes perdidos es que el origen disminuya su velocidad. El razonamiento detrás de esta lógica es que TCP fue diseñado para redes cableadas, y éstas son muy confiables, por lo tanto, la pérdida de paquetes se debe principalmente a desbordamientos de búfer y no a errores de transmisiones. Este hecho puede aprovecharse para reducir la congestión.

El objetivo de hacer que los enrutadores se deshagan de los paquetes antes de que la situación sea irremediable (de aquí el término “temprana” en el nombre), es que haya tiempo para hacer algo antes de que sea demasiado tarde. Para determinar cuándo comenzar a descartarlos, los enrutadores mantienen un promedio móvil de sus longitudes de cola. Cuando la longitud de cola promedio en algunas líneas sobrepasa un umbral, se dice que la línea está congestionada y se toma alguna medida.

Debido a que tal vez el enrutador no puede saber cuál origen está causando la mayoría de los problemas, probablemente lo mejor que se puede hacer es elegir un paquete al azar de la cola que puso en marcha la acción.

¿Cómo puede el enrutador informar al origen sobre el problema? Una forma es enviarle un paquete regulador, como describimos anteriormente. Sin embargo, con ese método surge un problema ya que coloca todavía más carga en la ya congestionada red. Una estrategia diferente es descartar el paquete seleccionado y no reportarlo. El origen notará en algún momento la falta de confirmación de recepción y tomará medidas. Debido a que sabe que los paquetes perdidos por lo general son causados por la congestión y las eliminaciones, responderá reduciendo la velocidad en lugar de aumentarla. Esta forma implícita de retroalimentación sólo funciona cuando los orígenes responden a la pérdida de paquetes reduciendo su tasa de transmisión. En las redes inalámbricas, en las que la mayoría de las pérdidas se debe al ruido en el enlace de radio, no se puede utilizar este método.

5.3.6 Control de fluctuación

En aplicaciones como la transmisión continua de audio y vídeo no importa gran cosa si los paquetes tardan 20 o 30 msec en ser entregados, siempre y cuando el tiempo de tránsito (retardo) sea constante. La variación (es decir, la desviación estándar) en el retardo de los paquetes se conoce como **fluctuación**. Una fluctuación alta, por ejemplo cuando unos paquetes tardan en llegar a su

destino 20 mseg y otros 30 mseg, resultará en una calidad desigual del sonido o la imagen. En la figura 5-29 se ilustra la fluctuación. Por tanto, el arreglo de que el 99% de los paquetes se entregue con un retardo que esté entre 24.5 y 25.5 mseg puede ser aceptable.

Por supuesto, el rango escogido debe ser factible. Debe tomar en cuenta el retardo causado por la velocidad de la luz y el retardo mínimo a través de los enrutadores y tal vez dejar un periodo corto para algunos retardos inevitables.

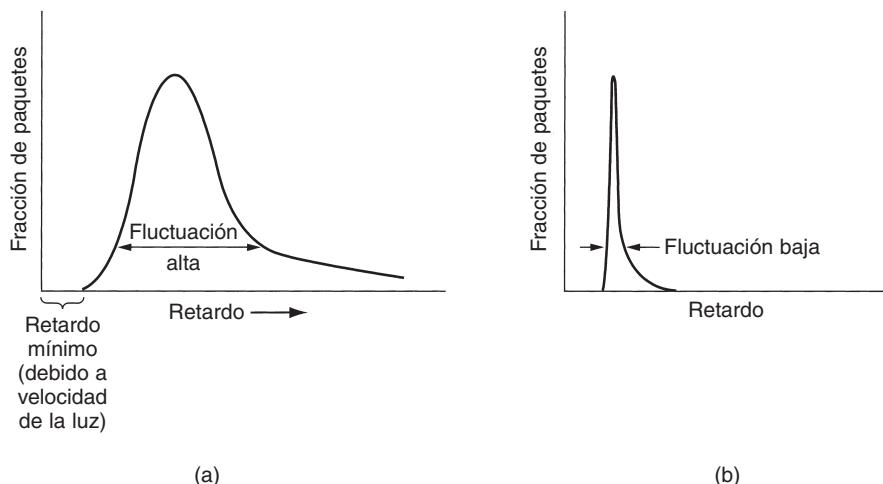


Figura 5-29. (a) Fluctuación alta. (b) Fluctuación baja.

La fluctuación puede limitarse calculando el tiempo de tránsito esperado para cada salto en la ruta. Cuando un paquete llega a un enrutador, éste lo examina para saber qué tan adelantado o retrasado está respecto a lo programado. Esta información se almacena en el paquete y se actualiza en cada salto. Si el paquete está adelantado, se retiene durante el tiempo suficiente para regresarlo a lo programado; si está retrasado, el enrutador trata de sacarlo rápidamente.

De hecho, el algoritmo para determinar cuál de varios paquetes que compiten por una línea de salida debe seguir siempre puede escoger el paquete más retrasado. De esta manera, los paquetes adelantados se frenan y los retrasados se aceleran, reduciendo en ambos casos la cantidad de fluctuación.

En algunas aplicaciones, como el vídeo bajo demanda, la fluctuación puede eliminarse almacenando los datos en el búfer del receptor y después obteniéndolos de dicho búfer en lugar de utilizar la red en tiempo real. Sin embargo, para otras aplicaciones, especialmente aquellas que requieren interacción en tiempo real entre personas como la telefonía y videoconferencia en Internet, el retardo inherente del almacenamiento en el búfer no es aceptable.

El control de congestión es un área activa de investigación. El estado presente se resume en (Gevros y cols., 2001).

5.4 CALIDAD DEL SERVICIO

Las técnicas que observamos en las secciones previas se diseñaron para reducir la congestión y mejorar el rendimiento de la red. Sin embargo, con el crecimiento de las redes de multimedia, con frecuencia estas medidas *ad hoc* no son suficientes. Se necesitan intentos serios para garantizar la calidad del servicio a través del diseño de redes y protocolos. En las siguientes secciones continuaremos nuestro estudio del rendimiento de la red, pero por ahora nos enfocaremos en las formas de proporcionar una calidad de servicio que se ajuste a las necesidades de las aplicaciones. Sin embargo, se debe dejar claro desde el principio que muchas de estas ideas están empezando y sujetas a cambios.

5.4.1 Requerimientos

Un **flujo** es un conjunto de paquetes que van de un origen a un destino. En una red orientada a la conexión, todos los paquetes que pertenezcan a un flujo siguen la misma ruta; en una red sin conexión, pueden seguir diferentes rutas. La necesidad de cada flujo se puede caracterizar por cuatro parámetros principales: confiabilidad, retardo, fluctuación y ancho de banda. Estos parámetros en conjunto determinan la **QoS (calidad del servicio)** que el flujo requiere. En la figura 5-30 se listan varias aplicaciones y el nivel de sus requerimientos.

Aplicación	Confiabilidad	Retardo	Fluctuación	Ancho de banda
Correo electrónico	Alta	Bajo	Baja	Bajo
Transferencia de archivos	Alta	Bajo	Baja	Medio
Acceso a Web	Alta	Medio	Baja	Medio
Inicio de sesión remoto	Alta	Medio	Media	Bajo
Audio bajo demanda	Baja	Bajo	Alta	Medio
Vídeo bajo demanda	Baja	Bajo	Alta	Alto
Telefonía	Baja	Alto	Alta	Bajo
Videoconferencia	Baja	Alto	Alta	Alto

Figura 5-30. Qué tan rigurosos son los requerimientos de calidad del servicio.

Las primeras cuatro aplicaciones tienen requerimientos rigurosos en cuanto a confiabilidad. No sería posible enviar bits de manera incorrecta. Este objetivo por lo general se alcanza al realizar una suma de verificación de cada paquete y al verificar dicha suma en el destino. Si se daña un paquete en el tránsito, no se confirma su recepción y se volverá a transmitir posteriormente. Esta estrategia proporciona una alta confiabilidad. Las cuatro aplicaciones finales (audio/vídeo) pueden tolerar errores, por lo que ni se realizan ni comprueban sumas de verificación.

Las aplicaciones de transferencia de archivos, incluyendo correo electrónico y vídeo, no son sensibles al retardo. Si todos los paquetes se retrasan unos segundos de manera uniforme, no hay daño. Las aplicaciones interactivas, como la navegación en Web y el inicio de sesión remoto,

son más sensibles a los retardos. Las aplicaciones en tiempo real, como la telefonía y la videoconferencia, tienen requerimientos estrictos de retardo. Si cada una de las palabras de una llamada telefónica se retrasa exactamente por 2.000 seg, los usuarios hallarán la conexión inaceptable. Por otra parte, la reproducción de archivos de audio o vídeo desde un servidor no requiere un retardo bajo.

Las primeras tres aplicaciones no son sensibles a los paquetes que llegan con intervalos de tiempo irregulares entre ellos. El inicio de sesión remoto es algo sensible a esto, debido a que los caracteres en la pantalla aparecerán en pequeñas ráfagas si una conexión tiene mucha fluctuación. El vídeo y especialmente el audio son en extremo sensibles a la fluctuación. Si un usuario está observando vídeo a través de la red y todos los cuadros se retrasan exactamente 2.000 seg, no hay daño. Pero si el tiempo de transmisión varía de manera aleatoria entre 1 y 2 seg, el resultado sería terrible. En el audio, una fluctuación de incluso unos cuantos milisegundos es claramente audible.

Por último, las aplicaciones difieren en sus anchos de banda; el correo electrónico y el inicio de sesión remoto no necesitan mucho, pero el vídeo en todas sus formas sí lo necesita.

Las redes ATM clasifican los flujos en cuatro categorías amplias con respecto a sus demandas de QoS, como se muestra a continuación:

1. Tasa de bits constante (por ejemplo, telefonía).
2. Tasa de bits variable en tiempo real (por ejemplo, videoconferencia comprimida).
3. Tasa de bits variable no constante (por ejemplo, ver una película a través de Internet).
4. Tasa de bits disponible (por ejemplo, transferencia de archivos).

Estas categorías también son útiles para otros propósitos y otras redes. La tasa de bits constante es un intento por simular un cable al proporcionar un ancho de banda uniforme y un retardo uniforme. La tasa de bits variable ocurre cuando el vídeo está comprimido, algunos cuadros están más comprimidos que otros. Por lo tanto, el envío de un cuadro con muchos detalles podría requerir enviar muchos bits en tanto que el envío de una foto de una pared blanca podría comprimirse muy bien. La tasa de bits disponible es para las aplicaciones, como el correo electrónico, que no son sensibles al retardo o a la fluctuación.

5.4.2 Técnicas para alcanzar buena calidad de servicio

Ahora que sabemos algo sobre los requerimientos de QoS, ¿cómo cumplimos con ellos? Bueno, para empezar, no hay una bola mágica. Ninguna técnica proporciona QoS eficiente y confiable de una manera óptima. En su lugar, se ha desarrollado una variedad de técnicas, con soluciones prácticas que con frecuencia se combinan múltiples técnicas. A continuación examinaremos algunas de las técnicas que los diseñadores de sistemas utilizan para alcanzar la QoS.

Sobreaprovisionamiento

Una solución fácil es proporcionar la suficiente capacidad de enrutador, espacio en búfer y ancho de banda como para que los paquetes fluyan con facilidad. El problema con esta solución es que

es costosa. Conforme pasa el tiempo y los diseñadores tienen una mejor idea de cuánto es suficiente, esta técnica puede ser práctica. En cierta medida, el sistema telefónico tiene un sobreaprovisionamiento. Es raro levantar un auricular telefónico y no obtener un tono de marcado instantáneo. Simplemente hay mucha capacidad disponible ahí que la demanda siempre se puede satisfacer.

Almacenamiento en búfer

Los flujos pueden almacenarse en el búfer en el lado receptor antes de ser entregados. Almacenarlos en el búfer no afecta la confiabilidad o el ancho de banda, e incrementa el retardo, pero atenúa la fluctuación. Para el vídeo o audio bajo demanda, la fluctuación es el problema principal, por lo tanto, esta técnica es muy útil.

En la figura 5-29 vimos la diferencia entre fluctuación alta y fluctuación baja. En la figura 5-31 vemos un flujo de paquetes que se entregan con una fluctuación considerable. El paquete 1 se envía desde el servidor a $t = 0$ seg y llega al cliente a $t = 1$ seg. El paquete 2 tiene un retardo mayor; tarda 2 seg en llegar. Conforme llegan los paquetes, se almacenan en el búfer en la máquina cliente.

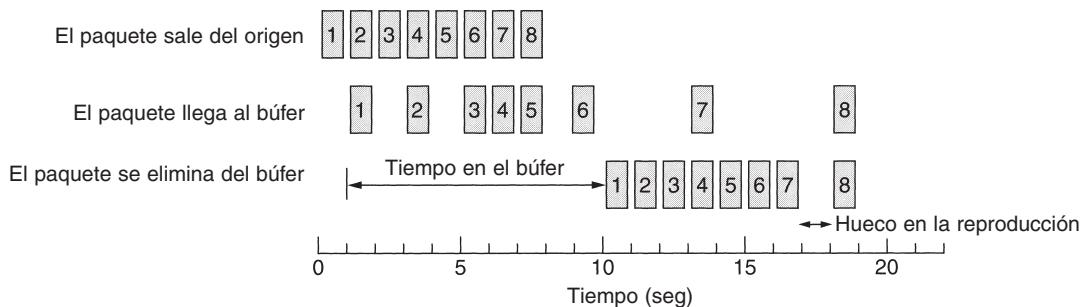


Figura 5-31. Refinamiento del flujo de paquetes almacenándolos en el búfer.

En el seg $t = 10$, la reproducción continúa. En este momento, los paquetes 1 a 6 se han almacenado en el búfer de manera que pueden eliminarse de él en intervalos uniformes para una reproducción suave. Desafortunadamente, el paquete 8 se ha retrasado tanto que no está disponible cuando le toca el turno a su ranura de reproducción, por lo que ésta debe parar hasta que llegue dicho paquete, creando un molesto hueco en la música o película. Este problema se puede atenuar retrasando el tiempo de inicio aún más, aunque hacer eso también requiere un búfer más grande. Los sitios Web comerciales que contienen transmisión continua de vídeo o audio utilizan reproductores que almacenan en el búfer por aproximadamente 10 seg antes de comenzar a reproducir.

Modelado de tráfico

En el ejemplo anterior, el origen envía los paquetes con un espacio uniforme entre ellos, pero en otros casos, podrían emitirse de manera regular, lo cual puede causar congestión en la red. El envío no uniforme es común si el servidor está manejando muchos flujos al mismo tiempo, y también permite otras acciones, como avance rápido y rebobinado, autenticación de usuario,

etcétera. Además, el enfoque que utilizamos aquí (almacenamiento en el búfer) no siempre es posible, por ejemplo, en la videoconferencia. Sin embargo, si pudiera hacerse algo para hacer que el servidor (y los *hosts* en general) transmita a una tasa uniforme, la calidad del servicio mejoraría. A continuación examinaremos una técnica, el **modelado de tráfico**, que modera el tráfico en el servidor, en lugar de en el cliente.

El modelado de tráfico consiste en regular la *tasa* promedio (y las ráfagas) de la transmisión de los datos. En contraste, los protocolos de ventana corrediza que estudiamos anteriormente limitan la cantidad de datos en tránsito de una vez, no la tasa a la que se envían. Cuando se establece una conexión, el usuario y la subred (es decir, el cliente y la empresa portadora) acuerdan cierto patrón de tráfico (es decir, forma) para ese circuito. Algunas veces esto se llama **acuerdo de nivel de servicio**. En tanto el cliente cumpla con su parte del contrato y sólo envíe los paquetes acordados, la empresa portadora promete entregarlos de manera oportuna. El modelado de tráfico reduce la congestión y, por lo tanto, ayuda a la empresa portadora a cumplir con su promesa. Tales acuerdos no son tan importantes para las transferencias de archivos pero sí para los datos en tiempo real, como conexiones de vídeo y audio, lo cual tiene requerimientos rigurosos de calidad de servicio.

En efecto, con el modelado de tráfico, el cliente le dice a la empresa portadora: Mi patrón de transmisión se parecerá a esto, ¿puedes manejarlo? Si la empresa portadora acepta, surge la cuestión de cómo puede saber ésta si el cliente está cumpliendo con el acuerdo y cómo debe proceder si el cliente no lo está haciendo. La supervisión de un flujo de tráfico se conoce como **supervisión de tráfico** (*traffic policing*). Aceptar una forma de tráfico y supervisarlo más tarde es más fácil en las subredes de circuitos virtuales que en las de datagramas. Sin embargo, incluso en las subredes de datagramas se pueden aplicar las mismas ideas a las conexiones de la capa de transporte.

Algoritmo de cubeta con goteo

Imagínese una cubeta con un pequeño agujero en el fondo, como se ilustra en la figura 5-32(a). Sin importar la rapidez con que entra agua en la cubeta, el flujo de salida tiene una tasa constante, ρ , cuando hay agua en la cubeta, y una tasa de cero cuando la cubeta está vacía. Además, una vez que se llena la cubeta, cualquier agua adicional que entra se derrama por los costados y se pierde (es decir, no aparece en el flujo por debajo del agujero).

Puede aplicarse el mismo concepto a los paquetes, como se muestra en la figura 5-32(b). De manera conceptual, cada *host* está conectado a la red mediante una interfaz que contiene una cubeta con goteo, es decir, una cola interna infinita. Si llega un paquete cuando la cola está llena, éste se descarta. En otras palabras, si uno o más procesos del *host* tratan de enviar paquetes cuando la cola ya tiene la cantidad máxima de paquetes, dicho paquete se descarta sin más. Este arreglo puede incorporarse en la interfaz del hardware, o simularse a través del sistema operativo del *host*. El esquema fue propuesto inicialmente por Turner (1986), y se llama **algoritmo de cubeta con goteo**. De hecho, no es otra cosa que un sistema de encolamiento de un solo servidor con un tiempo de servicio constante.

El *host* puede poner en la red un paquete por pulso de reloj. Nuevamente, esto puede forzarse desde la tarjeta de interfaz o desde el sistema operativo. Este mecanismo convierte un flujo desi-

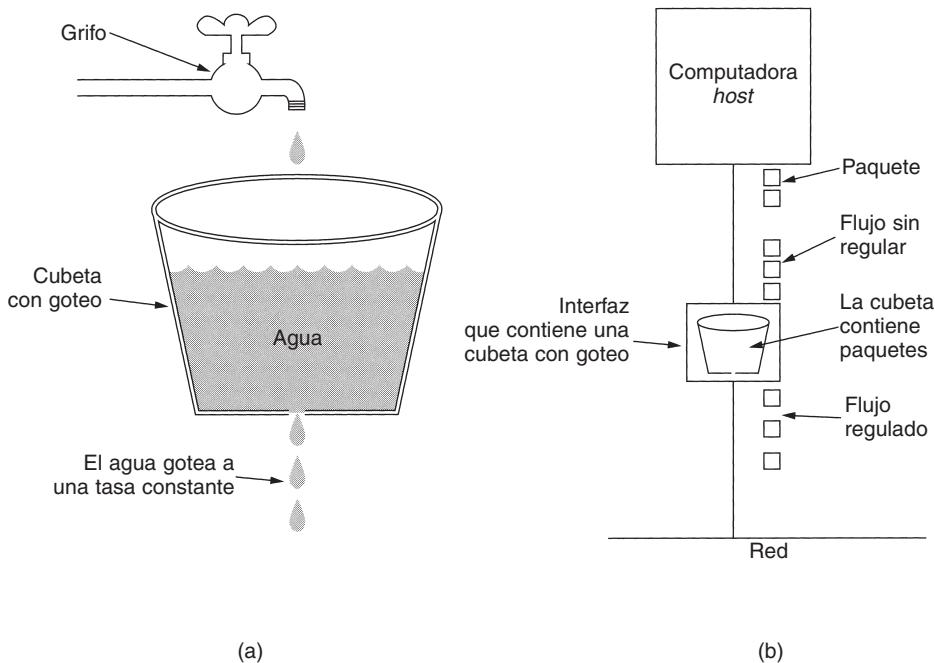


Figura 5-32. (a) Una cubeta con goteo, llena de agua. (b) Cubeta con goteo, llena de paquetes.

gual de paquetes de los procesos de usuario dentro del *host* en un flujo continuo de paquetes hacia la red, moderando las ráfagas y reduciendo en una buena medida las posibilidades de congestión.

Cuando todos los paquetes son del mismo tamaño (por ejemplo, celdas ATM), este algoritmo puede usarse como se describe. Sin embargo, cuando se utilizan paquetes de tamaño variable, con frecuencia es mejor permitir un número fijo de bytes por pulso, en lugar de un solo paquete. Por lo tanto, si la regla es de 1024 bytes por pulso, sólo pueden recibirse por pulso un paquete de 1024 bytes, dos paquetes de 512 bytes, cuatro paquetes de 256 bytes, etcétera. Si el conteo de bytes residuales es demasiado bajo, el siguiente paquete debe esperar hasta el siguiente pulso.

La implementación del algoritmo de cubeta con goteo es fácil. La cubeta con goteo consiste en una cola finita. Si cuando llega un paquete hay espacio en la cola, éste se agrega a ella; de otro modo, se descarta. En cada pulso de reloj se transmite un paquete (a menos que la cola esté vacía).

La cubeta con goteo que usa conteo de bits se implementa casi de la misma manera. En cada pulso un contador se inicializa en n . Si el primer paquete de la cola tiene menos bytes que el valor actual del contador, se transmite y se disminuye el contador en esa cantidad de bytes. Pueden enviarse paquetes adicionales en tanto el contador sea lo suficientemente grande. Cuando el contador está por debajo de la longitud del siguiente paquete de la cola, la transmisión se detiene hasta el siguiente pulso, en cuyo momento se restablece el conteo de bytes residuales y el flujo puede continuar.

Como ejemplo de cubeta con goteo, imagine que una computadora puede producir datos a razón de 25 millones de bytes/seg (200 Mbps) y que la red también opera a esta velocidad. Sin embargo, los enrutadores pueden manejar esta tasa de datos sólo durante intervalos cortos (básicamente, hasta que sus búferes se llenen). Durante intervalos grandes, dichos enrutadores funcionan mejor con tasas que no exceden 2 millones de bytes/seg. Ahora suponga que los datos llegan en ráfagas de un millón de bytes, con una ráfaga de 40 mseg cada segundo. Para reducir la tasa promedio a 2 MB/seg, podemos usar una cubeta con goteo de $\rho = 2$ MB/seg y una capacidad, C , de 1 MB. Esto significa que las ráfagas de hasta 1 MB pueden manejarse sin pérdidas de datos, ya que se distribuyen a través de 500 mseg, sin importar la velocidad a la que lleguen.

En la figura 5-33(a) vemos la entrada de la cubeta con goteo operando a 25 MB/seg durante 40 mseg. En la figura 5-33(b) vemos la salida drenándose a una velocidad uniforme de 2 MB/seg durante 500 mseg.

Algoritmo de cubeta con *tokens*

El algoritmo de cubeta con goteo impone un patrón de salida rígido a la tasa promedio, sin importar la cantidad de ráfagas que tenga el tráfico. En muchas aplicaciones es mejor permitir que la salida se acelere un poco cuando llegan ráfagas grandes, por lo que se necesita un algoritmo más flexible, de preferencia uno que nunca pierda datos. El **algoritmo de cubeta con *tokens*** es uno de tales algoritmos. En este algoritmo, la cubeta con goteo contiene *tokens*, generados por un reloj a razón de un *token* cada ΔT seg. En la figura 5-34(a) se muestra una cubeta que contiene tres *tokens* y cinco paquetes esperando a ser transmitidos. Para que se transmita un paquete, éste debe capturar y destruir un *token*. En la figura 5-34(b) vemos que han pasado tres de los cinco paquetes, pero los otros dos están atorados, esperando la generación de dos o más *tokens*.

El algoritmo de cubeta con *tokens* ofrece una forma diferente de modelado de tráfico que el algoritmo de cubeta con goteo. Este último no permite que los *hosts* inactivos acumulen permisos para enviar posteriormente ráfagas grandes. El algoritmo de cubeta con *tokens* sí permite el ahorro, hasta el tamaño máximo de la cubeta, n . Esta propiedad significa que pueden enviarse a la vez ráfagas de hasta n paquetes, permitiendo cierta irregularidad en el flujo de salida y dando una respuesta más rápida a las ráfagas de entrada repentinas.

Otra diferencia entre los dos algoritmos es que el algoritmo de cubeta con *tokens* descarta los *tokens* (es decir, la capacidad de transmisión) cuando se llena la cubeta, pero nunca descarta los paquetes. En contraste, el algoritmo de cubeta con goteo descarta los paquetes cuando se llena la cubeta.

Aquí también es posible una variación menor, en la que cada *token* representa el derecho de transmitir no un paquete, sino k bytes. Sólo puede transmitirse un paquete si hay suficientes *tokens* disponibles para cubrir su longitud en bytes. Los *tokens* fraccionarios se guardan para uso futuro.

Los algoritmos de cubeta con goteo y cubeta con *tokens* también pueden servir para regular el tráfico entre los enrutadores, así como para regular la salida de un *host*, como en nuestros ejemplos. Sin embargo, una diferencia clara es que una cubeta con *tokens* que regula a un *host* puede hacer que éste detenga el envío cuando las reglas dicen que debe hacerlo. Indicar a un enrutador que detenga la transmisión mientras sigue recibiendo entradas puede dar como resultado una pérdida de datos.

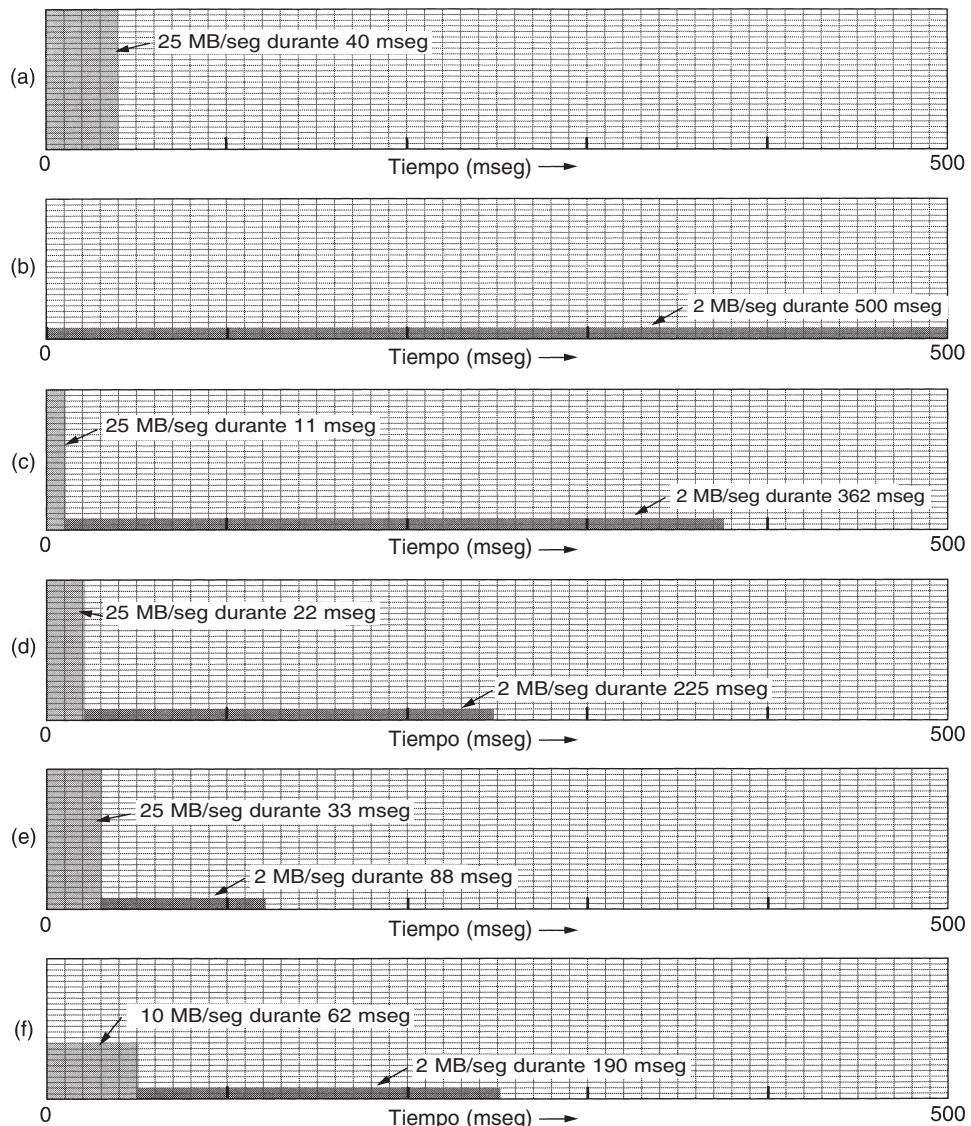


Figura 5-33. (a) Entrada a una cubeta con goteo. (b) Salida de una cubeta con goteo. (c)-(e) Salida de una cubeta con *tokens* con capacidades de 250 KB, 500 KB y 750 KB. (f) Salida de una cubeta con *tokens* de 500 KB que alimenta a una cubeta con goteo de 10 MB/seg.

La implementación del algoritmo básico de cubeta con *tokens* simplemente es sólo una variable que cuenta *tokens*. El contador se incrementa en uno cada ΔT y se decrementa en uno cada vez que se envía un paquete. Cuando el contador llega a cero, ya no pueden enviarse paquetes. En la variante de conteo de bits, el contador se incrementa en k bytes cada ΔT y se decrementa en la longitud de cada paquete enviado.

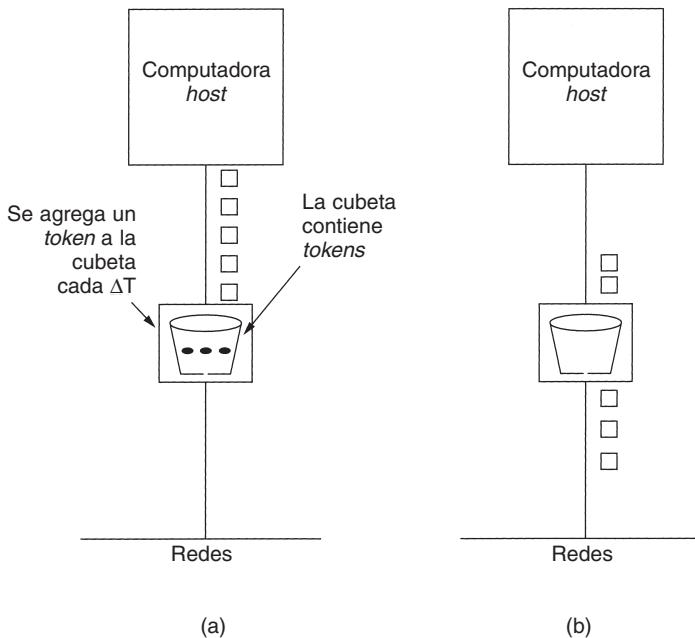


Figura 3-34. Algoritmo de cubeta con *tokens*. (a) Antes. (b) Despu s.

En esencia, lo que hace la cubeta con *tokens* es permitir ráfagas, pero limitadas a una longitud máxima regulada. Por ejemplo, vea la figura 5-33(c). Ahí tenemos una cubeta de *tokens* con 250 KB de capacidad. Los *tokens* llegan a una tasa que permite un flujo de salida de 2 MB/seg. Suponiendo que la cubeta con *tokens* está llena cuando llega la ráfaga de 1 MB, la cubeta puede drenarse a la velocidad máxima de 25 MB/seg durante unos 11 mseg. Entonces tiene que desacelerarse hasta 2 MB/seg hasta que se ha enviado toda la ráfaga de entrada.

El cálculo de la longitud de ráfaga con tasa máxima es un tanto complicado. No es sólo la división de 1 MB entre 25 MB/seg, ya que, mientras se está enviando la ráfaga, llegan más *tokens*. Si S seg es la longitud de la ráfaga, C bytes es la capacidad de la cubeta con *tokens*, ρ bytes/seg es la tasa de llegada de *tokens* y M bytes/seg es la tasa máxima de salida, podemos ver que una ráfaga de salida contiene un máximo de $C + \rho S$ bytes. También sabemos que la cantidad de bytes en una ráfaga a velocidad máxima con longitud de S segundos es MS . Por lo tanto, tenemos

$$C + \rho S = MS$$

Podemos resolver esta ecuación para obtener $S = C/(M - \rho)$. Para nuestros parámetros de $C = 250$ KB, $M = 25$ MB/seg y $\rho = 2$ MB/seg, tenemos un tiempo de ráfaga de aproximadamente 11 mseg. En la figura 5-33(d) y en la figura 5-33(e) se muestra la cubeta con *tokens* para capacidades de 500 y 750 KB, respectivamente.

Un problema potencial con el algoritmo de cubeta con *tokens* es que permite ráfagas largas, aunque puede regularse el intervalo máximo de ráfaga mediante una selección cuidadosa de ρ y M . Con frecuencia es deseable reducir la tasa pico, pero sin regresar al valor mínimo de la cubeta con goteo original.

Una manera de lograr tráfico más uniforme es poner una cubeta con goteo después de la cubeta con *tokens*. La tasa de la cubeta con goteo deberá ser mayor que la ρ de la cubeta con *tokens*, pero menor que la tasa máxima de la red. En la figura 5-33(f) se muestra la salida de una cubeta con *tokens* de 500 KB seguida de una cubeta con goteo de 10 MB/seg.

La supervisión de estos esquemas puede ser un tanto complicada. En esencia, la red tiene que simular el algoritmo y asegurarse de que no se envíen más paquetes o bytes de lo permitido. Sin embargo, estas herramientas proporcionan métodos para modelar el tráfico de la red de formas más manejables para ayudar a cumplir con los requerimientos de calidad del servicio.

Reservación de recursos

El hecho de tener la capacidad de regular la forma del tráfico ofrecido es un buen inicio para garantizar la calidad del servicio. Sin embargo, utilizar efectivamente esta información significa de manera implícita obligar a todos los paquetes de un flujo a que sigan la misma ruta. Su envío a través de enrutadores aleatorios dificulta garantizar algo. Como consecuencia, se debe configurar algo similar a un circuito virtual del origen al destino, y todos los paquetes que pertenecen al flujo deben seguir esta ruta.

Una vez que se tiene una ruta específica para un flujo, es posible reservar recursos a lo largo de esa ruta para asegurar que la capacidad necesaria esté disponible. Se pueden reservar tres tipos de recursos:

1. Ancho de banda.
2. Espacio de búfer.
3. Ciclos de CPU.

El primero, ancho de banda, es el más obvio. Si un flujo requiere 1 Mbps y la línea saliente tiene una capacidad de 2 Mbps, tratar de dirigir tres flujos a través de esa línea no va a funcionar. Por lo tanto, reservar ancho de banda significa no sobrecargar ninguna línea de salida.

Un segundo recurso que por lo general es escaso es el espacio en búfer. Cuando llega un paquete, por lo general el hardware mismo lo deposita en la tarjeta de interfaz de red. A continuación, el software enrutador tiene que copiarlo en un búfer en RAM y colocar en la cola ese búfer para transmitirlo en la línea saliente elegida. Si no hay búfer disponible, el paquete se tiene que descartar debido a que no hay lugar para colocarlo. Para una buena calidad de servicio, es posible reservar algunos búferes para un flujo específico de manera que éste no tenga que competir con otros flujos para obtener espacio en búfer. Siempre que ese flujo necesite un búfer, se le proporcionará uno mientras existan disponibles.

Por último, los ciclos de CPU también son un recurso escaso. Para procesar un paquete se necesita tiempo de CPU del enrutador, por lo que un enrutador sólo puede procesar cierta cantidad de paquetes por segundo. Para asegurar el procesamiento oportuno de cada paquete, es necesario verificar que la CPU no esté sobrecargada.

A primera vista podría parecer que si un enrutador tarda, digamos, 1 µseg, en procesar un paquete, entonces puede procesar 1 millón de paquetes/seg. Esta observación no es verdadera porque siempre habrá periodos inactivos debido a fluctuaciones estadísticas en la carga. Si la CPU necesita cada ciclo para poder realizar su trabajo, la pérdida incluso de algunos ciclos debido a periodos inactivos ocasionales crea un atraso del que nunca se podrá deshacer.

Sin embargo, incluso con una carga que esté ligeramente por debajo de la capacidad teórica, se pueden generar colas y pueden ocurrir retardos. Considere una situación en la que los paquetes llegan de manera aleatoria con una tasa promedio de llegada de λ paquetes/seg. El tiempo de CPU requerido por cada uno también es aleatorio, con una capacidad media de procesamiento de μ paquetes/seg. Bajo el supuesto de que las distribuciones de arribo y de servicio son distribuciones de Poisson, es posible probar, mediante la teoría de encolamiento, que el retardo promedio experimentado por un paquete, T , es

$$T = \frac{1}{\mu} \times \frac{1}{1 - \lambda/\mu} = \frac{1}{\mu} \times \frac{1}{1 - \rho}$$

donde $\rho = \lambda/\mu$ es el uso de CPU. El primer factor, $1/\mu$, sería el tiempo de servicio si no hubiera competencia. El segundo factor es la reducción de velocidad debido a la competencia con otros flujos. Por ejemplo, si $\lambda = 950,000$ paquetes/seg y $\mu = 1,000,000$ paquetes/seg, entonces $\rho = 0.95$ y el retardo promedio experimentado por cada paquete será de 20 µseg en lugar de 1 µseg. Este tiempo cuenta tanto para el tiempo de encolamiento como para el de servicio, como puede verse cuando la carga es muy baja ($\lambda/\mu \approx 0$). Si hay, digamos, 30 enrutadores a lo largo de la ruta del flujo, el retardo de encolamiento será de alrededor de 600 µseg.

Control de admisión

Ahora nos encontramos en el punto en que el tráfico entrante de algún flujo está bien modelado y puede seguir una sola ruta cuya capacidad puede reservarse por adelantado en los enrutadores a lo largo de la ruta. Cuando un flujo de este tipo se ofrece a un enrutador, éste tiene que decidir, con base en su capacidad y en cuántos compromisos tiene con otros flujos, si lo admite o lo rechaza.

La decisión de aceptar o rechazar un flujo no se trata simplemente de comparar el ancho de banda, los búferes o los ciclos requeridos por el flujo con la capacidad excedida del enrutador en esas tres dimensiones. Es más complicado que eso. Para empezar, aunque algunas aplicaciones podrían saber sobre sus requerimientos de ancho de banda, saben poco acerca de búferes o ciclos de CPU y, por esto, se necesita por lo menos una forma diferente de describir los flujos. Además, algunas aplicaciones son mucho más tolerantes con el incumplimiento ocasional de plazos que otras. Por último, algunas aplicaciones podrían estar dispuestas a negociar los parámetros del flujo y otras no. Por ejemplo, un visor de películas que por lo general se ejecuta a 30 cuadros/seg podría

estar dispuesto a ejecutar a 25 cuadros/seg si no hay suficiente ancho de banda para soportar 30 cuadros/seg. De manera similar, la cantidad de píxeles por cuadro y de ancho de banda de audio, entre otras propiedades, podría ser ajustable.

Debido a que muchas partes pueden estar involucradas en la negociación del flujo (el emisor, el receptor y todos los enrutadores a lo largo de la ruta), los flujos deben describirse de manera precisa en términos de parámetros específicos que se puedan negociar. Un conjunto de tales parámetros se conoce como **especificación de flujo**. Por lo general, el emisor (por ejemplo, el servidor de vídeo) produce una especificación de flujo que propone los parámetros que le gustaría utilizar. Conforme la especificación se propague por la ruta, cada enrutador la examina y modifica los parámetros conforme sea necesario. Las modificaciones sólo pueden reducir el flujo, no incrementarlo (por ejemplo, una tasa más baja de datos, no una más grande). Cuando llega al otro extremo, se pueden establecer los parámetros.

Como ejemplo de lo que puede estar en una especificación de flujo, considere el de la figura 5-35, que se basa en los RFCs 2210 y 2211. Tiene cinco parámetros, el primero de los cuales, la *Tasa de la cubeta con tokens*, es la cantidad de bytes por segundo que se colocan en la cubeta. Ésta es la tasa máxima que el emisor puede transmitir, promediada con respecto a un intervalo de tiempo largo.

Parámetro	Unidad
Tasa de la cubeta con <i>tokens</i>	Bytes/seg
Tamaño de la cubeta con <i>tokens</i>	Bytes
Tasa pico de datos	Bytes/seg
Tamaño mínimo de paquete	Bytes
Tamaño máximo de paquete	Bytes

Figura 5-35. Ejemplo de especificación de flujo.

El segundo parámetro es el tamaño de la cubeta en bytes. Por ejemplo, si la *Tasa de la cubeta con tokens* es de 1 Mbps y el *Tamaño de la cubeta con tokens* es de 500 KB, la cubeta se puede llenar de manera continua durante 4 seg antes de llenarse por completo (en caso de que no haya transmisiones). Cualesquier *tokens* enviados después de eso, se pierden.

El tercer parámetro, la *Tasa pico de datos*, es la tasa máxima de transmisiones tolerada, incluso durante intervalos de tiempo breves. El emisor nunca debe sobrepasar esta tasa.

Los últimos dos parámetros especifican los tamaños mínimo y máximo de paquetes, incluyendo los encabezados de la capa de red y de transporte (por ejemplo, TCP e IP). El tamaño mínimo es importante porque procesar cada paquete toma un tiempo fijo, aunque sea breve. Un enrutador debe estar preparado para manejar 10,000 paquetes/seg de 1 KB cada uno, pero no para manejar 100,000 paquetes/seg de 50 bytes cada uno, aunque esto represente una tasa de datos menor. El tamaño máximo de paquete es importante debido a las limitaciones internas de la red que no deben sobreponerse. Por ejemplo, si parte de la ruta es a través de una Ethernet, el tamaño máximo del paquete se restringirá a no más de 1500 bytes, sin importar lo que el resto de la red puede manejar.

Una pregunta interesante es cómo convierte un enrutador una especificación de flujo en un conjunto de reservaciones de recursos específicos. Esta conversión es específica de la implementación y no está estandarizada. Supongamos que un enrutador puede procesar 100,000 paquetes/seg. Si se le ofrece un flujo de 1 MB/seg con tamaños de paquete mínimo y máximo de 512 bytes, el enrutador puede calcular que puede transmitir 2048 paquetes/seg de ese flujo. En ese caso, debe reservar 2% de su CPU para ese flujo, o de preferencia más para evitar retardos largos de encolamiento. Si la política de un enrutador es nunca asignar más de 50% de su CPU (lo que implica un retardo con factor de dos, y ya está lleno el 49%, entonces ese flujo debe rechazarse). Se necesitan cálculos similares para los otros recursos.

Entre más rigurosa sea la especificación de flujo, más útil será para los enrutadores. Si una especificación de flujo especifica que necesita una *tasa de cubeta con tokens* de 5 MB/seg pero los paquetes pueden variar de 50 bytes a 1500 bytes, entonces la tasa de paquetes variará aproximadamente de 3500 a 105,000 paquetes/seg. El enrutador podría asustarse por este último número y rechazar el flujo, mientras que con un tamaño mínimo de paquete de 1000 bytes, el flujo de 5 MB/seg podría aceptarse.

Enrutamiento proporcional

La mayoría de los algoritmos de enrutamiento tratan de encontrar la mejor ruta para cada destino y envían a través de ella todo el tráfico a ese destino. Un método diferente que se ha propuesto para proporcionar una calidad de servicio más alta es dividir el tráfico para cada destino a través de diferentes rutas. Puesto que generalmente los enrutadores no tienen un panorama completo del tráfico de toda la red, la única forma factible de dividir el tráfico a través de múltiples rutas es utilizar la información disponible localmente. Un método simple es dividir el tráfico en fracciones iguales o en proporción a la capacidad de los enlaces salientes. Sin embargo, hay disponibles otros algoritmos más refinados (Nelakuditi y Zhang, 2002).

Calendarización de paquetes

Si un enrutador maneja múltiples flujos, existe el peligro de que un flujo acapare mucha de su capacidad y limite a los otros flujos. El procesamiento de paquetes en el orden de arriba significa que un emisor agresivo puede acaparar la mayor parte de la capacidad de los enrutadores por los que pasan sus paquetes, lo que reduce la calidad del servicio para otros. Para hacer fracasar esos intentos, se han diseñado varios algoritmos de programación de paquetes (Bhatti y Crowcroft, 2000).

Uno de los primeros fue el de **encolamiento justo** (*fair queueing*) (Nagle, 1987). La esencia del algoritmo es que los enrutadores tienen colas separadas para cada línea de salida, una por flujo. Cuando una línea se queda inactiva, el enrutador explora las diferentes colas de manera circular, y toma el primer paquete de la siguiente cola. De esta forma, con n hosts compitiendo por una línea de salida dada, cada host obtiene la oportunidad de enviar uno de n paquetes. El envío de más paquetes no mejorará esta fracción.

Aunque al principio este algoritmo tiene un problema: proporciona más ancho de banda a los hosts que utilizan paquetes más grandes que a los que utilizan paquetes más pequeños. Demers y cols. (1990) sugirieron una mejora en la que la exploración circular (*round robin*) se realiza de tal

manera que se simule una exploración circular byte por byte, en lugar de paquete por paquete. En efecto, explora las colas de manera repetida, byte por byte, hasta que encuentra el instante en el que finalizará cada paquete. A continuación, los paquetes se ordenan conforme a su tiempo de terminación y se envían en ese orden. En la figura 5-36 se ilustra este algoritmo.

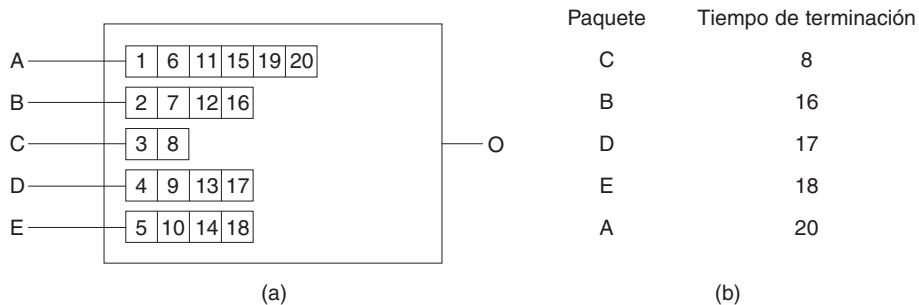


Figura 5-36. (a) Un enrutador con cinco paquetes encolados en la línea O . (b) Tiempos de terminación de los cinco paquetes.

En la figura 5-36(a) se muestran paquetes con una longitud de 2 hasta 6 bytes. En el pulso de reloj (virtual) 1, se envía el primer byte del paquete de la línea A . Después le toca el turno al primer byte del paquete de la línea B , y así sucesivamente. El primer paquete en terminar es C , después de ocho pulsos. El orden se muestra en la figura 5-36(b). Debido a que ya no hay más llegadas, los paquetes se enviarán en el orden listado, de C a A .

Un problema con este algoritmo es que da la misma prioridad a todos los *hosts*. En muchas situaciones, es necesario dar a los servidores de vídeo más ancho de banda que a los servidores de archivos regulares, a fin de que puedan proporcionárseles dos o más bytes por pulso. Este algoritmo modificado se conoce como **encolamiento justo ponderado** (*weighted fair queueing*) y se utiliza ampliamente. Algunas veces el peso es igual a la cantidad de flujos provenientes de una máquina, de manera que el proceso obtiene un ancho de banda igual. Una implementación eficiente del algoritmo se analiza en (Shreedhar y Varghese, 1995). El reenvío real de paquetes a través de un enrutador o commutador se está realizando cada vez más en el hardware (Elhanany y cols., 2001).

5.4.3 Servicios integrados

Entre 1995 y 1997, la IETF se esforzó mucho en diseñar una arquitectura para la multimedia de flujos continuos. Este trabajo resultó en cerca de dos docenas de RFCs, empezando con los RFCs 2205–2210. El nombre genérico para este trabajo es **algoritmos basados en flujo** o **servicios integrados**. Se diseñó tanto para aplicaciones de unidifusión como para multidifusión. Un ejemplo de la primera es un solo usuario difundiendo un clip de vídeo de un sitio de noticias. Un ejemplo del segundo es una colección de estaciones de televisión digital difundiendo sus programas como flujos de paquetes IP a muchos receptores de diferentes ubicaciones. A continuación nos concentraremos en la multidifusión, debido a que la transmisión por unidifusión es un caso especial de multidifusión.

En muchas aplicaciones de multidifusión, los grupos pueden cambiar su membresía de manera dinámica, por ejemplo, conforme las personas entran a una videoconferencia y se aburren y cambian a una telenovela o al canal del juego de croquet. Bajo estas condiciones, el método de hacer que los emisores reserven ancho de banda por adelantado no funciona bien, debido a que requeriría que cada emisor rastreara todas las entradas y salidas de su audiencia. Para un sistema diseñado para transmitir televisión con millones de suscriptores, ni siquiera funcionaría.

RSVP—Protocolo de reservación de recursos

El principal protocolo IETF para la arquitectura de servicios integrados es **RSVP**. Se describe en el RFC 2205 y en otros. Este protocolo se utiliza para marcar las reservas; para el envío de datos se utilizan otros protocolos. RSVP permite que varios emisores transmitan a múltiples grupos de receptores, permite que receptores individuales cambien de canal libremente, optimiza el uso de ancho de banda y elimina la congestión.

En su forma más sencilla, el protocolo usa enrutamiento de multidifusión con árboles de expansión, como se vio antes. A cada grupo se le asigna un grupo de direcciones. Para enviar a un grupo, un emisor pone la dirección del grupo en sus paquetes. El algoritmo estándar de multidifusión construye entonces un árbol de expansión que cubre a todos los miembros del grupo. El algoritmo de enrutamiento no es parte del RSVP. La única diferencia con la multidifusión normal es un poco de información extra multidifundida al grupo periódicamente para indicarle a los enruteadores a lo largo del árbol que mantengan ciertas estructuras de datos en sus memorias.

Como ejemplo, considere la red de la figura 5-37(a). Los *hosts* 1 y 2 son emisores multidifusión, y los *hosts* 3, 4 y 5 son receptores multidifusión. En este ejemplo, los emisores y los receptores son distintos pero, en general, los dos grupos pueden traslaparse. Los árboles de multidifusión de los *hosts* 1 y 2 se muestran en las figuras 5-37(b) y 5-37(c), respectivamente.

Para obtener mejor recepción y eliminar la congestión, cualquiera de los receptores de un grupo puede enviar un mensaje de reservación por el árbol al emisor. El mensaje se propaga usando el algoritmo de reenvío por ruta invertida estudiado antes. En cada salto, el enrutador nota la reservación y aparta el ancho de banda necesario; si no hay suficiente ancho de banda disponible, informa de una falla. En el momento que el mensaje llega de regreso al origen, se ha reservado el ancho de banda desde el emisor hasta el receptor que hace la solicitud de reservación a lo largo del árbol de expansión.

En la figura 5-38(a) se muestra un ejemplo de tales reservaciones. Aquí el *host* 3 ha solicitado un canal al *host* 1. Una vez establecido el canal, los paquetes pueden fluir de 1 a 3 sin congestiones. Ahora considere lo que sucede si el *host* 3 reserva a continuación un canal hacia otro emisor, el *host* 2, para que el usuario pueda ver dos programas de televisión a la vez. Se reserva una segunda ruta, como se muestra en la figura 5-38(b). Observe que se requieren dos canales individuales del *host* 3 al enrutador *E*, porque se están transmitiendo dos flujos independientes.

Por último, en la figura 5-38(c), el *host* 5 decide observar el programa transmitido por el *host* 1 y también hace una reservación. Primero se reserva ancho de banda dedicado hasta el enrutador *H*.

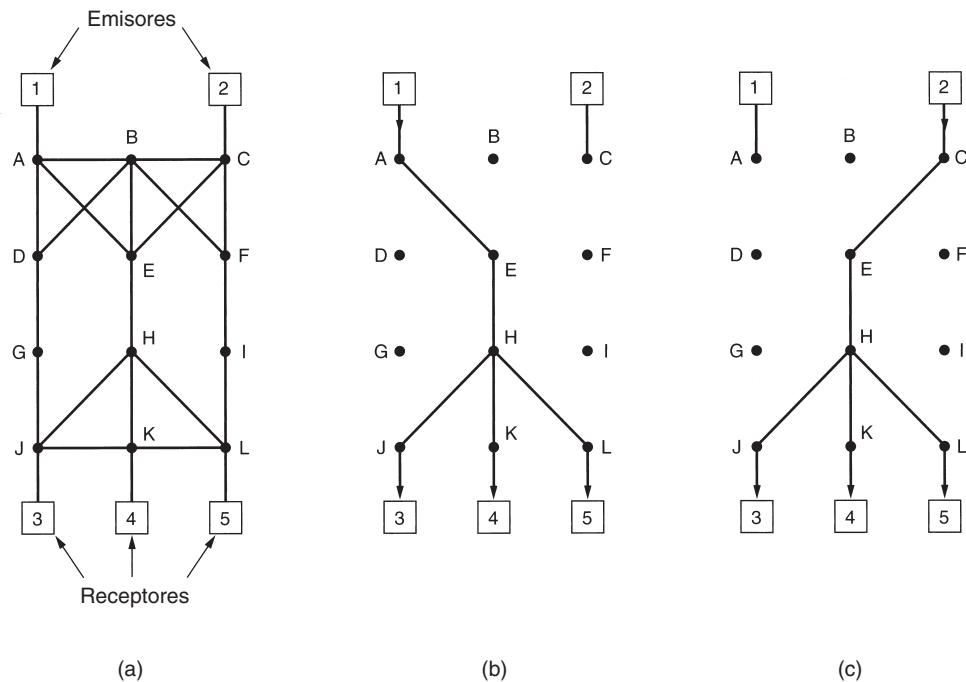


Figura 5-37. (a) Red. (b) Árbol de expansión de multidifusión para el *host* 1. (c) Árbol de expansión de multidifusión para el *host* 2.

Sin embargo, éste ve que ya tiene una alimentación del *host* 1, por lo que, si ya se ha reservado el ancho de banda necesario, no necesita reservar nuevamente. Observe que los *hosts* 3 y 5 podrían haber solicitado diferentes cantidades de ancho de banda (por ejemplo, el 3 tiene una televisión de blanco y negro, por lo que no quiere la información de color), así que la capacidad reservada debe ser lo bastante grande para satisfacer al receptor más voraz.

Al hacer una reserva, un receptor puede especificar (opcionalmente) uno o más orígenes de los que quiere recibir. También puede especificar si estas selecciones quedarán fijas durante toda la reserva, o si el receptor quiere mantener abierta la opción de cambiar los orígenes después. Los enrutadores usan esta información para optimizar la planeación del ancho de banda. En particular, sólo se establece que dos receptores van a compartir una ruta si ambos están de acuerdo en no cambiar los orígenes posteriormente.

La razón de esta estrategia en el caso totalmente dinámico es que el ancho de banda reservado está desacoplado de la selección del origen. Una vez que un receptor ha reservado ancho de banda, puede conmutarse a otro origen y conservar la parte de la ruta existente que es válida para el nuevo origen. Por ejemplo, si el *host* 2 está transmitiendo varios flujos de vídeo, el *host* 3 puede conmutarse entre ellos a voluntad sin cambiar su reserva: a los enrutadores no les importa el programa que está viendo el receptor.

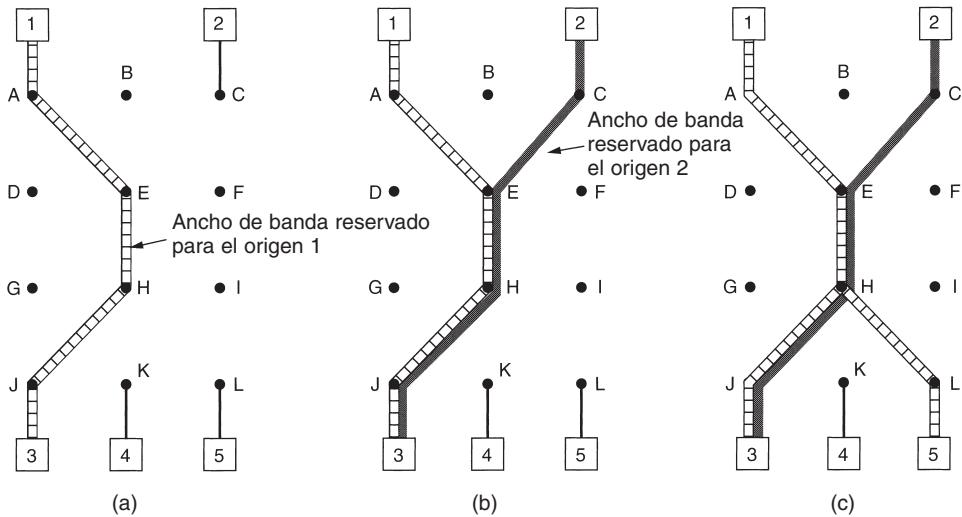


Figura 5-38. (a) El host 3 solicita un canal al host 1. (b) El host 3 solicita entonces un segundo canal al host 2. (c) El host 5 solicita un canal al host 1.

5.4.4 Servicios diferenciados

Los algoritmos basados en flujo tienen el potencial de ofrecer buena calidad de servicio a uno o más flujos debido a que reservan los recursos que son necesarios a lo largo de la ruta. Sin embargo, también tienen una desventaja. Requieren una configuración avanzada para establecer cada flujo, algo que no se escala bien cuando hay miles o millones de flujos. Además, mantienen estado por flujo interno en los enrutadores, haciéndolos vulnerables a las caídas de enrutadores. Por último, los cambios requeridos al código de enrutador son sustanciales e involucran intercambios complejos de enrutador a enrutador para establecer los flujos. Como consecuencia, existen pocas implementaciones de RSVP o algo parecido.

Por estas razones, la IETF también ha diseñado un método más simple para la calidad del servicio, uno que puede implementarse ampliamente de manera local en cada enrutador sin una configuración avanzada y sin que toda la ruta esté involucrada. Este método se conoce como calidad de servicio **basada en clase** (contraria a basada en flujo). La IETF ha estandarizado una arquitectura para él, llamada **servicios diferenciados**, que se describe en los RFCs 2474, 2475, entre otros. A continuación lo describiremos.

Un conjunto de enrutadores que forman un dominio administrativo (por ejemplo, un ISP o una compañía telefónica) puede ofrecer los servicios diferenciados (DS). La administración define un conjunto de clases de servicios con reglas de reenvío correspondientes. Si un cliente firma para un DS, los paquetes del cliente que entran en el dominio podrían contener un campo *Tipo de servicio*, con un mejor servicio proporcionado a algunas clases (por ejemplo, un servicio premium) que a otras. Al tráfico dentro de una clase se le podría requerir que se apegue a algún modelo específico, como a una cubeta con goteo con una tasa especificada de drenado. Un operador con intuición para los negocios

podría cargar una cantidad extra por cada paquete premium transportado o podría permitir hasta N paquetes premium por una mensualidad adicional fija. Observe que este esquema no requiere una configuración avanzada, ni reserva de recursos ni negociación extremo a extremo que consume tiempo para cada flujo, como sucede con los servicios integrados. Esto hace de DS relativamente fácil de implementar.

El servicio basado en clase también ocurre en otras industrias. Por ejemplo, las compañías de envío de paquetes con frecuencia ofrecen servicio de tres días, de dos días, y servicio de un día para otro. Las aerolíneas ofrecen servicio de primera clase, de clase de negocios y de tercera clase. Los trenes que recorren largas distancias con frecuencia tienen múltiples clases de servicios. Incluso el metro de París tiene dos clases de servicios. Para los paquetes, las clases pueden diferir en términos de retardo, fluctuación y probabilidad de ser descartado en caso de congestión, entre otras posibilidades (pero probablemente sin tramas Ethernet más amplias).

Para hacer que la diferencia entre la calidad basada en el servicio y la basada en clase de servicio sea más clara, considere un ejemplo: la telefonía de Internet. Con un esquema basado en flujo, cada llamada telefónica obtiene sus propios recursos y garantías. Con un esquema basado en clase, todas las llamadas telefónicas obtienen los recursos reservados para la telefonía de clase. Estos recursos no pueden ser tomados por paquetes de la clase de transferencia de archivos u otras clases, pero ninguna llamada telefónica obtiene ningún recurso privado reservado sólo para ella.

Reenvío expedito o acelerado

Cada operador debe realizar la selección de clases de servicios, pero debido a que los paquetes con frecuencia se reenvían entre subredes ejecutadas por diferentes operadores, la IETF está trabajando para definir las clases de servicios independientes de la red. La clase más simple es el **reenvío expedito**, por lo tanto, iniciemos con ella. Se describe en el RFC 3246.

La idea detrás del reenvío expedito es muy simple. Dos clases de servicios están disponibles: regular y expedita. Se espera que la mayor parte del tráfico sea regular, pero una pequeña fracción de los paquetes son expeditos. Los paquetes expeditos deben tener la capacidad de transitar la subred como si no hubieran otros paquetes. En la figura 5-39 se muestra una representación simbólica de este sistema de “dos tubos”. Observe que todavía hay una línea física. Los dos conductos lógicos que se muestran en la figura representan una forma de reservar ancho de banda, no una segunda línea física.

Una forma de implementar esta estrategia es programar los enruteadores para que tengan dos colas de salida por cada línea de salida, una para los paquetes expeditos y una para los regulares. Cuando llega un paquete, se coloca en la cola de manera acorde. La programación de paquetes debe utilizar algo parecido al encolamiento justo ponderado. Por ejemplo, si 10% del tráfico es expedito y 90% es regular, 20% del ancho de banda podría dedicarse al tráfico expedito y el resto al tráfico regular. Al hacer esto se daría al tráfico expedito dos veces más ancho de banda del que necesita a fin de que dicho tráfico tenga un retardo bajo. Esta asignación se puede alcanzar transmitiendo un paquete expedito por cada cuatro paquetes regulares (suponiendo que el tamaño de la distribución para ambas clases es similar). De esta forma, se espera que los paquetes expeditos vean una red descargada, incluso cuando hay, de hecho, una carga pesada.

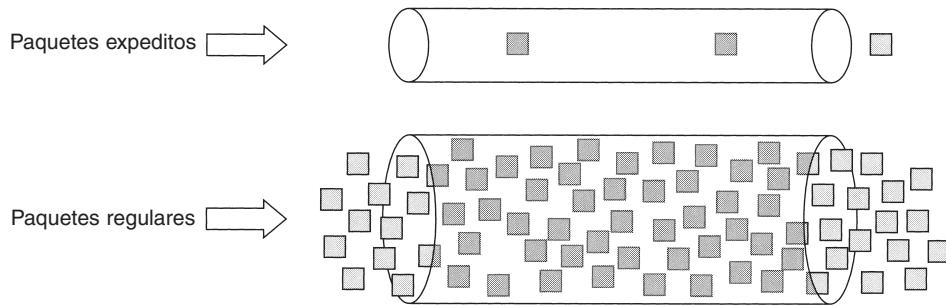


Figura 5-39. Los paquetes expeditos viajan por una red libre de tráfico.

Reenvío asegurado

Un esquema un poco más elaborado para el manejo de las clases de servicios se conoce como **reenvío asegurado**. Se describe en el RFC 2597. Especifica que deberán haber cuatro clases de prioridades, y cada una tendrá sus propios recursos. Además, define tres probabilidades de descarte para paquetes que están en congestión: baja, media y alta. En conjunto, estos dos factores definen 12 clases de servicios.

La figura 5-40 muestra una forma en que los paquetes pueden ser procesados bajo reenvío asegurado. El paso 1 es clasificar los paquetes en una de cuatro clases de prioridades. Este paso podría realizarse en el *host* emisor (como se muestra en la figura) o en el enrutador de ingreso. La ventaja de realizar la clasificación en el *host* emisor es que hay más información disponible acerca de cuáles paquetes pertenecen a qué flujos.

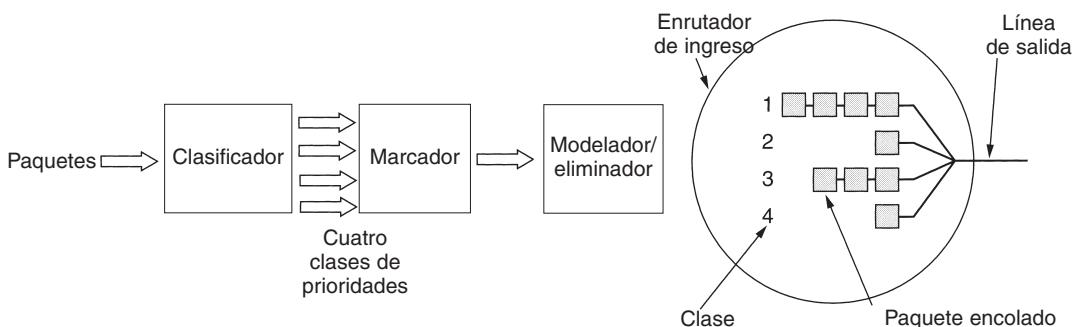


Figura 5-40. Una posible implementación del flujo de datos para el reenvío asegurado.

El paso 2 es marcar los paquetes de acuerdo con su clase. Para este propósito se necesita un campo de encabezado. Por fortuna, en el encabezado IP está disponible un campo *Tipo de servicio* de 8 bits, como veremos un poco más adelante. El RFC 2597 especifica que seis de estos bits se van a utilizar para la clase de servicio, dejando espacio de codificación para clases de servicio históricas y para futuras.

El paso 3 es pasar los paquetes a través de un filtro modelador/eliminador que podría retardar o descartar algunos de ellos para dar una forma aceptable a los cuatro flujos, por ejemplo, mediante cubetas con goteo o con *tokens*. Si hay muchos paquetes, algunos de ellos podrían descartarse aquí, mediante una categoría de eliminación. También son posibles esquemas elaborados que involucren la medición o la retroalimentación.

En este ejemplo, estos tres pasos se realizan en el *host* emisor, por lo que el flujo de salida ahora se introduce en el enrutador de ingreso. Vale la pena mencionar que estos pasos pueden ser realizados por software especial de conectividad de redes o incluso por el sistema operativo, a fin de no tener que cambiar las aplicaciones existentes.

5.4.5 Comutación de etiquetas y MPLS

Mientras la IETF estaba desarrollando servicios integrados y diferenciados, varios fabricantes de enrutadores estaban desarrollando mejores métodos de reenvío. Este trabajo se enfocó en agregar una etiqueta en frente de cada paquete y realizar el enrutamiento con base en ella y no con base en la dirección de destino. Hacer que la etiqueta sea un índice de una tabla provoca que encontrar la línea correcta de salida sea una simple cuestión de buscar en una tabla. Al utilizar esta técnica, el enrutamiento puede llevarse a cabo de manera muy rápida y cualesquier recursos necesarios pueden reservarse a lo largo de la ruta.

Por supuesto, etiquetar los flujos de esta manera se acerca peligrosamente a los circuitos virtuales. X.25, ATM, frame relay, y otras redes con una subred de circuitos virtuales colocan una etiqueta (es decir, un identificador de circuitos virtuales) en cada paquete, la buscan en una tabla y enrutan con base en la entrada de la tabla. A pesar del hecho de que muchas personas en la comunidad de Internet tienen una aversión intensa por las redes orientadas a la conexión, la idea parece surgir nuevamente, pero esta vez para proporcionar un enrutamiento rápido y calidad de servicio. Sin embargo, hay diferencias esenciales entre la forma en que Internet maneja la construcción de la ruta y la forma en que lo hacen las redes orientadas a la conexión, por lo que esta técnica no utiliza la comutación de circuitos tradicional.

Esta “nueva” idea de comutación ha pasado por varios nombres (propietarios), entre ellos **comutación de etiquetas**. En algún momento, la IETF comenzó a estandarizar la idea bajo el nombre **MPLS (comutación de etiquetas multiprotocolo)**. De aquí en adelante lo llamaremos MPLS. Se describe en el RFC 3031, entre muchos otros.

Además, algunas personas hacen una distinción entre *enrutamiento* y *comutación*. El enrutamiento es el proceso de buscar una dirección de destino en una tabla para saber a dónde enviar los paquetes hacia ese destino. En contraste, la comutación utiliza una etiqueta que se toma de un paquete como un índice en una tabla de reenvío. Sin embargo, estas definiciones están lejos de ser universales.

El primer problema es en dónde colocar la etiqueta. Debido a que los paquetes IP no fueron diseñados para circuitos virtuales, en el encabezado IP no hay ningún campo disponible para los números de tales circuitos. Por esta razón, se tuvo que agregar un nuevo encabezado MPLS en frente del encabezado IP. En una línea de enrutador a enrutador que utiliza PPP como protocolo

de tramas, el formato de trama, incluyendo los encabezados PPP, MPLS, IP y TCP, es como se muestra en la figura 5-41. De cierta forma, MPLS es, por lo tanto, la capa 2.5.

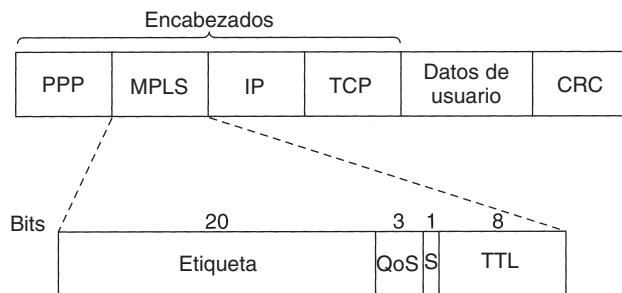


Figura 5-41. Transmisión de un segmento TCP que utiliza IP, MPLS y PPP.

El encabezado MPLS genérico tiene cuatro campos, el más importante de los cuales es el de *Etiqueta*, el cual contiene el índice. El campo *QoS (bits experimentales)* indica la clase de servicio. El campo *S* se relaciona con colocar en una pila múltiples etiquetas en redes jerárquicas (que se analizan más adelante). Si tiene el valor de 1 indica que es la última etiqueta añadida al paquete IP, si es un 0 indica que hay más etiquetas añadidas al paquete. El campo evita el ciclo infinito en caso de que haya inestabilidad en el enrutamiento, ya que se decrementa en cada enrutador y al llegar al valor de 0, el paquete es descartado.

Debido a que los encabezados MPLS no son parte del paquete de la capa de red o de la trama del enlace de datos, MPLS es en gran medida independiente de ambas capas. Entre otras cosas, esta propiedad significa que es posible construir conmutadores MPLS que pueden reenviar tanto paquetes IP como celdas ATM, dependiendo de lo que aparezca. De esta característica proviene la parte “multiprotocolo” del nombre MPLS.

Cuando un paquete mejorado con MPLS (o celda) llega a un enrutador con capacidad MPLS, la etiqueta se utiliza como un índice en una tabla para determinar la línea de salida y la nueva etiqueta a utilizar. Esta conmutación de etiquetas se utiliza en todas las subredes de circuitos virtuales, debido a que las etiquetas sólo tienen importancia local y dos enrutadores diferentes pueden asignar la misma etiqueta a paquetes hacia diferentes destinos, es decir, la etiqueta es reasignada a la salida de cada enrutador, por lo que no se mantiene la misma etiqueta en toda la ruta. En la figura 5-3 vimos en acción este mecanismo. MPLS utiliza la misma técnica.

Una diferencia con respecto a los circuitos virtuales tradicionales es el nivel de agregación. Ciertamente es posible que cada flujo tenga su propio conjunto de etiquetas a través de la subred. Sin embargo, es más común que los enrutadores agrupen múltiples flujos que terminan en un enrutador o una LAN particulares y utilicen una sola etiqueta de ellos. Se dice que los flujos que están agrupados en una sola etiqueta pertenecen a la misma **FEC (clase de equivalencia de reenvío)**. Esta clase cubre no sólo a dónde van los paquetes, sino también su clase de servicio (en el sentido de los servicios diferenciados), debido a que todos sus paquetes se tratan de la misma forma para propósitos de reenvío.

Con el enrutamiento de circuitos virtuales tradicional no es posible agrupar en el mismo identificador de circuitos virtuales varias rutas diferentes con diferentes puntos finales, debido a que podría no haber forma de distinguirlas en el destino final. Con MPLS, los paquetes aún contienen su dirección de destino final, además de la etiqueta, a fin de que al final de la red de MPLS pueda eliminarse la etiqueta y que el reenvío pueda continuar de la forma normal, utilizando la dirección de destino de la capa de red.

Una diferencia principal entre MPLS y los diseños de circuitos virtuales convencionales es la forma en que está construida la tabla de reenvío. En las redes de circuitos virtuales tradicionales, cuando un usuario desea establecer una conexión, se inicia un paquete de configuración en la red para crear la ruta y crear las entradas de la tabla de reenvío. MPLS no funciona de esa forma porque no hay fase de configuración para cada conexión (pues eso podría romper con la operación de mucho software existente en Internet).

En su lugar, hay dos formas de crear las entradas de la tabla de reenvío. En el método **orientado a datos**, cuando un paquete llega, el primer enrutador que encuentra contacta al siguiente enrutador en el sentido descendente del flujo a donde tiene que ir el paquete y le pide que genere una etiqueta para el flujo. Este método se aplica de manera recursiva. En efecto, ésta es una creación de circuitos virtuales por petición.

Los protocolos que hacen esta propagación son muy cuidadosos para evitar los ciclos cerrados (loops). Por lo general, utilizan una técnica llamada **subprocesos con color** (*colored threads*). La propagación en reversa de una FEC se puede comparar con extraer un subproceso de un color único en la subred. Si un enrutador ve un color que ya tiene, sabe que hay un ciclo y toma una medida para solucionarlo. El método dirigido por datos se utiliza principalmente en redes en las que el transporte subyacente es ATM (como sucede en la mayor parte del sistema telefónico).

La otra forma, que se utiliza en las redes que no se basan en ATM, es el método **dirigido por control**. Tiene algunas variantes. Una de ellas funciona de la siguiente manera. Cuando se inicia un enrutador, verifica para cuáles rutas es el último salto (por ejemplo, qué *hosts* están en su LAN). Después crea una o más FECs para ellas, asigna una etiqueta para cada una y pasa las etiquetas a sus vecinos. Éstos, a su vez, introducen las etiquetas en sus tablas de reenvío y envían nuevas etiquetas a sus vecinos, hasta que todos los enrutadores han adquirido la ruta. También es posible reservar recursos conforme la ruta está construida para garantizar una calidad de servicio apropiada.

MPLS puede operar a múltiples niveles al mismo tiempo. En el nivel más alto, cada empresa portadora puede considerarse como un tipo de metaenrutador, con una ruta a través de los metaenrutadores del origen al destino. Esta ruta puede utilizar MPLS. Sin embargo, MPLS también puede utilizarse dentro de la red de cada empresa portadora, lo que resulta en un segundo nivel de etiquetado. De hecho, un paquete puede llevar consigo una pila entera de etiquetas. El bit *S* de la figura 5-41 permite que un enrutador elimine una etiqueta para saber si quedaron etiquetas adicionales. Se establece a 1 para la etiqueta inferior y 0 para las otras etiquetas. En la práctica, esta característica se utiliza principalmente para implementar redes privadas virtuales y túneles recursivos.

Aunque las ideas básicas detrás de MPLS son directas, los detalles son extremadamente complicados, y tienen muchas variaciones y optimizaciones, por lo que ya no trataremos más ese tema. Para mayor información, vea (Davie y Rekhter, 2000; Lin y cols., 2002; Pepelnjak y Gui-chard, 2001, y Wang, 2001).

5.5 INTERCONECTIVIDAD

Hasta ahora hemos supuesto de manera implícita que hay una sola red homogénea y que cada máquina usa el mismo protocolo en cada capa. Por desgracia, este supuesto es demasiado optimista. Existen muchas redes diferentes, entre ellas LANs, MANs y WANs. En cada capa hay numerosos protocolos de uso muy difundido. En las siguientes secciones estudiaremos con cuidado los problemas que surgen cuando dos o más redes se juntan, formando una **interred**.

Existe una controversia considerable sobre si la abundancia actual de tipos de red es una condición temporal que desaparecerá tan pronto como todo mundo se dé cuenta de lo maravilloso que es [indique aquí su red favorita], o si es una característica inevitable pero permanente del mundo, que está aquí para quedarse. Tener diferentes redes invariablemente implica tener diferentes protocolos.

Creemos que siempre habrá una variedad de redes (y, por lo tanto, de protocolos) diferentes, por las siguientes razones. Antes que nada, la base instalada de redes diferentes es grande. Casi todas las instalaciones UNIX ejecutan TCP/IP. Muchos negocios grandes aún tienen *mainframes* que ejecutan SNA de IBM. Una cantidad considerable de compañías telefónicas operan redes ATM. Algunas LANs de computadoras personales aún usan Novell NCP/IPX o AppleTalk. Por último, las redes inalámbricas constituyen un área nueva y en desarrollo con una variedad de protocolos. Esta tendencia continuará por años, debido a problemas de herencia, tecnología nueva y al hecho de que no todos los fabricantes se interesan en que sus clientes puedan migrar fácilmente al sistema de otro fabricante.

Segundo, a medida que las computadoras y las redes se vuelven más baratas, el lugar de la toma de decisiones se desplaza hacia abajo. Muchas compañías tienen políticas en el sentido de que las compras de más de un millón de dólares tienen que ser aprobadas por la gerencia general, las compras de más de 100,000 dólares tienen que ser aprobadas por la gerencia media, pero las compras por debajo de 100,000 dólares pueden ser hechas por los jefes de los departamentos sin aprobación de los superiores. Esto puede dar como resultado fácilmente a que el departamento de ingeniería instale estaciones de trabajo de UNIX que ejecuten TCP/IP y a que el departamento de marketing instale Macs con AppleTalk.

Tercero, las distintas redes (por ejemplo, ATM e inalámbricas) tienen tecnologías radicalmente diferentes, por lo que no debe sorprendernos que, a medida que haya avances nuevos en hardware, también se cree software nuevo adaptado al nuevo hardware. Por ejemplo, la casa típica ahora es como la oficina típica de hace 10 años: está llena de computadoras que no se hablan entre ellas. En el futuro, podría ser común que el teléfono, la televisión y otros aparatos estuvieran en red, para controlarlos de manera remota. Esta nueva tecnología sin duda generará nuevos protocolos y redes.

Como ejemplo de cómo se pueden conectar redes diferentes, considere la figura 5-42. Ahí vemos una red corporativa con múltiples ubicaciones enlazadas por una red ATM de área amplia. En una de las ubicaciones, se utiliza una red dorsal óptica FDDI para conectar una Ethernet, una LAN inalámbrica 802.11 y la red de *mainframe* SNA del centro de datos corporativo.

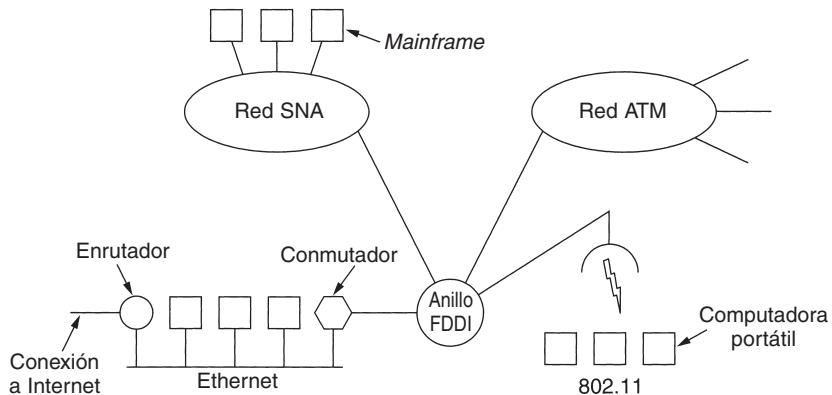


Figura 5-42. Una colección de redes interconectadas.

El propósito de interconectar todas estas redes es permitir que los usuarios de cualquiera de ellas se comuniquen con los usuarios de las demás, así como permitir que los usuarios de cualquiera de ellas accedan los datos de las demás. Lograr este objetivo significa enviar paquetes de una red a otra. Debido a que las redes, por lo general, difieren de formas considerables, obtener paquetes de una red a otra no siempre es tan fácil, como veremos a continuación.

5.5.1 Cómo difieren las redes

Las redes pueden diferir de muchas maneras. Algunas de las diferencias, como técnicas de modulación o formatos de tramas diferentes, se encuentran en las capas de enlace de datos y en la física. No trataremos esas diferencias aquí. En su lugar, en la figura 5-43 listamos algunas diferencias que pueden ocurrir en la capa de red. La conciliación de estas diferencias es lo que hace más difícil la interconexión de redes que la operación con una sola red.

Cuando los paquetes enviados por un origen en una red deben transitar a través de una o más redes foráneas antes de llegar a la red de destino (que también puede ser diferente de la red de origen), pueden ocurrir muchos problemas en las interfaces entre las redes. Para comenzar, cuando los paquetes de una red orientada a la conexión deben transitar a una red sin conexiones, deben reordenarse, algo que el emisor no espera y que el receptor no está preparado para manejar. Con frecuencia se necesitarán conversiones de protocolo, que pueden ser difíciles si la funcionalidad requerida no puede expresarse. También se necesitarán conversiones de direcciones, lo que podría requerir algún tipo de sistema de directorio. El paso de paquetes multidifusión a través de una red que no reconoce la multidifusión requiere la generación de paquetes individuales para cada destino.

Los diferentes tamaños máximos de paquete usados por las diferentes redes son un dolor de cabeza importante. ¿Cómo se pasa un paquete de 8000 bytes a través de una red cuyo tamaño máximo de paquete es de 1500 bytes? Es importante la diferencia en la calidad de servicio cuando

Aspecto	Algunas posibilidades
Servicio ofrecido	Sin conexiones, orientado a conexiones
Protocolos	IP, IPX, SNA, ATM, MPLS, AppleTalk, etc.
Direccionamiento	Plano (802) o jerárquico (IP)
Multidifusión	Presente o ausente (también difusión)
Tamaño de paquete	Cada red tiene su propio máximo
Calidad del servicio	Puede estar presente o ausente; muchos tipos diferentes
Manejo de errores	Entrega confiable, ordenada y desordenada
Control de flujo	Ventana corrediza, control de tasa, otros o ninguno
Control de congestión	Cubeta con goteo, paquetes reguladores, etc.
Seguridad	Reglas de confidencialidad, encriptación, etc.
Parámetros	Diferentes terminaciones de temporizador, especificaciones de flujo, etc.
Contabilidad	Por tiempo de conexión, por paquete, por byte, o sin ella

Figura 5-43. Algunas de las muchas maneras en que pueden diferir las redes.

un paquete que tiene restricciones de tiempo real pasa a través de una red que no ofrece garantías de tiempo real.

El control de errores, de flujo y de congestión suele ser diferente entre las diferentes redes. Si tanto el origen como el destino esperan la entrega de paquetes en secuencia y sin errores, pero una red intermedia simplemente descarta paquetes cuando huele congestión en el horizonte, o los paquetes vagan sin sentido durante un rato y emergen repentinamente para ser entregados, muchas aplicaciones fallarán. Existen diferentes mecanismos de seguridad, ajustes de parámetros y reglas de contabilidad, e incluso leyes de confidencialidad internacionales, que pueden causar problemas.

5.5.2 Conexión de redes

Las redes pueden interconectarse mediante diversos dispositivos, como vimos en el capítulo 4. Revisemos brevemente ese material. En la capa física, las redes se pueden conectar mediante repetidores o concentradores, los cuales mueven los bits de una red a otra idéntica. Éstos son en su mayoría dispositivos analógicos y no comprenden nada sobre protocolos digitales (simplemente re-generan señales).

En la capa de enlace de datos encontramos puentes y commutadores. Pueden aceptar tramas, examinar las direcciones MAC y reenviar las tramas a una red diferente mientras realizan una traducción menor de protocolos en el proceso, por ejemplo, de Ethernet a FDDI o a 802.11.

En la capa de red hay enrutadores que pueden conectar dos redes. Si éstas tienen capas de red diferentes, el enrutador puede tener la capacidad de traducir entre los formatos de paquetes, aunque la traducción de paquetes ahora es cada vez menos común. Un enrutador que puede manejar múltiples protocolos se conoce como **enrutador multiprotocolo**.

En la capa de transporte se encuentran puertas de enlace de transporte, que pueden interactuar entre dos conexiones de transporte. Por ejemplo, una puerta de enlace de transporte podría permitir que los paquetes fluyeran entre una red TCP y una SNA, las cuales tienen protocolos de transporte diferentes, fijando esencialmente una conexión TCP con una SNA.

Por último, en la capa de aplicación, las puertas de enlace de aplicación traducen semánticas de mensaje. Como ejemplo, las puertas de enlace entre el correo electrónico de Internet (RFC 822) y el correo electrónico X.400 deben analizar los mensajes de correo electrónico y cambiar varios campos de encabezado.

En este capítulo nos enfocaremos en la interconectividad en la capa de red. Para ver cómo difiere esto de la conmutación en la capa de enlace de datos, examine la figura 5-44. En la figura 5-44(a), la máquina de origen, *S*, desea enviar un paquete a la máquina de destino, *D*. Estas máquinas se encuentran en Ethernets diferentes, conectadas mediante un conmutador. *S* encapsula el paquete en una trama y lo envía a su destino. La trama llega al conmutador, el cual ve la dirección MAC de dicha trama y determina que ésta tiene que ir a la LAN 2. El conmutador elimina la trama de la LAN 1 y la coloca en la LAN 2.

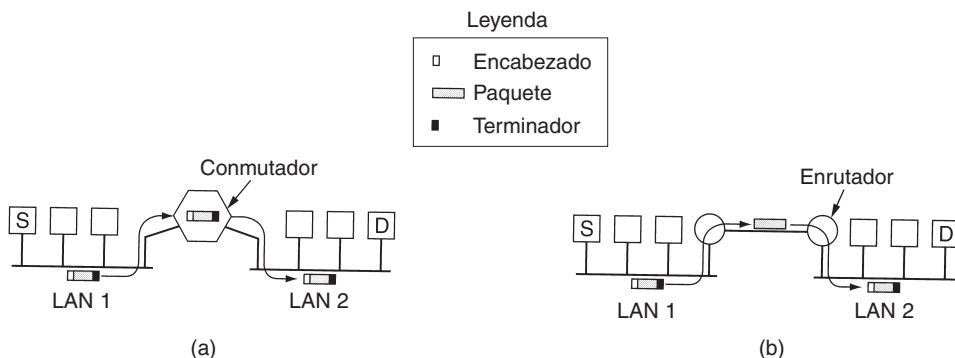


Figura 5-44. (a) Dos Ethernets conectadas mediante un conmutador. (b) Dos Ethernets conectadas mediante enrutadores.

Ahora consideremos la misma situación pero con las dos Ethernets conectadas mediante un par de enrutadores en lugar de un conmutador. Los enrutadores se conectan mediante una línea punto a punto, posiblemente una línea rentada de miles de kilómetros de longitud. Ahora el enrutador recoge la trama y el paquete se elimina del campo de datos de dicha trama. El enrutador examina la dirección del paquete (por ejemplo, una dirección IP) y la busca en su tabla de enrutamiento. Con base en esta dirección, decide enviar el paquete al enrutador remoto, encapsulado en un tipo diferente de trama, dependiendo del protocolo de línea. En el otro extremo el paquete se coloca en el campo de datos de una trama Ethernet y se deposita en la LAN 2.

Lo anterior es la diferencia esencial entre el caso de conmutación (o puenteo) y el caso enrulado. Con un conmutador (o puente), toda la trama se transporta con base en su dirección MAC. Con un enrutador, el paquete se extrae de la trama y la dirección del paquete se utiliza para decidir

a dónde enviarlo. Los conmutadores no tienen que entender el protocolo de capa de red que se está utilizando para conmutar los paquetes. Los enrutadores sí tienen que hacerlo.

5.5.3 Circuitos virtuales concatenados

Dos estilos posibles de interconectividad: la concatenación orientada a la conexión de subredes de circuitos virtuales, y los datagramas estilo Internet. A continuación los examinaremos por separado, pero primero una advertencia. En el pasado, la mayoría de las redes (públicas) eran orientadas a la conexión (frame relay, SNA, 802.16 y ATM aún lo son). Posteriormente, con la aceptación rápida de Internet, los datagramas se pusieron de moda. Sin embargo, sería un error pensar que los datagramas son para siempre. En este negocio, lo único que es para siempre es el cambio. Con la importancia creciente de las redes de multimedia, es probable que la orientación a la conexión regrese de una forma o de otra, puesto que es más fácil garantizar la calidad de servicio con conexiones que sin ellas. Por lo tanto, dedicaremos algún tiempo para estudiar las redes orientadas a la conexión.

En el modelo de circuitos virtuales concatenados, que se muestra en la figura 5-45, se establece una conexión con un *host* de una red distante de un modo parecido a la manera en que se establecen normalmente las conexiones. La subred ve que el destino es remoto y construye un circuito virtual al enrutador más cercano a la red de destino; luego construye un circuito virtual de ese enrutador a una **puerta de enlace** externa (enrutador multiprotocolo). Ésta registra la existencia del circuito virtual en sus tablas y procede a construir otro circuito virtual a un enrutador de la siguiente subred. Este proceso continúa hasta llegar al *host* de destino.

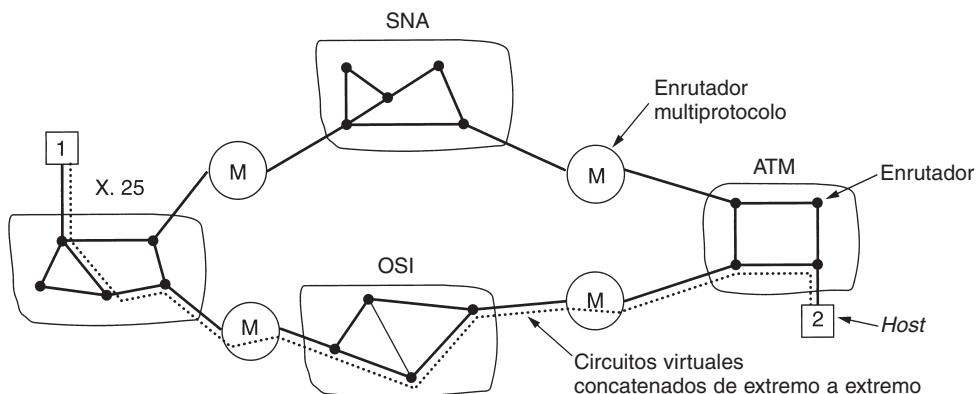


Figura 5-45. Interconectividad mediante circuitos virtuales concatenados.

Una vez que comienzan a fluir paquetes de datos por la ruta, cada puerta de enlace retransmite los paquetes de entrada y hace las conversiones entre los formatos de paquete y los números de circuito virtual, según sea necesario. Obviamente, todos los paquetes de datos deben atravesar la misma secuencia de puertas de enlace. En consecuencia, la red nunca reordena los paquetes de un flujo.

La característica esencial de este enfoque es que se establece una secuencia de circuitos virtuales desde el origen, a través de una o más puertas de enlace, hasta el destino. Cada puerta de enlace mantiene tablas que indican los circuitos virtuales que pasan a través suyo, a dónde se deben enrutar y el nuevo número de circuito virtual.

Este esquema funciona mejor cuando todas las redes tienen aproximadamente las mismas propiedades. Por ejemplo, si todas garantizan la entrega confiable de paquetes de capa de red, entonces, salvo una caída a lo largo de la ruta, el flujo del origen al destino también será confiable. De igual modo, si ninguna de ellas garantiza la entrega confiable, entonces la concatenación de los circuitos virtuales tampoco será confiable. Por otra parte, si la máquina de origen está en una red que garantiza la entrega confiable, pero una de las redes intermedias puede perder paquetes, la concatenación habrá cambiado fundamentalmente la naturaleza del servicio.

Los circuitos virtuales concatenados también son comunes en la capa de transporte. En particular, es posible construir un conducto de bits usando, digamos, SNA, que termine en una puerta de enlace, y luego tener una conexión TCP de esa puerta de enlace a la siguiente. De este modo, puede construirse un circuito virtual de extremo a extremo que abarque diferentes redes y protocolos.

5.5.4 Interconectividad no orientada a la conexión

El modelo alterno de interred es el modelo de datagramas, mostrado en la figura 5-46. En este modelo, el único servicio que ofrece la capa de red a la capa de transporte es la capacidad de inyectar datagramas en la subred y esperar que todo funcione bien. En la capa de red no hay noción en lo absoluto de un circuito virtual, y mucho menos de una concatenación de éstos. Este modelo no requiere que todos los paquetes que pertenecen a una conexión atraviesen la misma secuencia de puertas de enlace. En la figura 5-46 los datagramas del *host* 1 al *host* 2 toman diferentes rutas a través de la interred. Para cada paquete se toma una decisión de enrutamiento independiente, posiblemente dependiendo del tráfico en el momento del envío de dicho paquete. Esta estrategia puede utilizar múltiples rutas y lograr de esta manera un ancho de banda mayor que el modelo de circuitos virtuales concatenados. Por otra parte, no hay garantía de que los paquetes llegarán al destino en orden, suponiendo que lleguen.

El modelo de la figura 5-46 no es tan sencillo como parece. Por una parte, si cada red tiene su propio protocolo de capa de red, no es posible que un paquete de una red transite por otra. Podríamos imaginar a los enrutadores multiprotocolo tratando de traducir de un formato a otro, pero a menos que los dos formatos sean parientes cercanos con los mismos campos de información, tales conversiones siempre serán incompletas, y frecuentemente destinadas al fracaso. Por esta razón, pocas veces se intentan las conversiones.

Un segundo problema, más serio, es el direccionamiento. Imagine un caso sencillo: un *host* de Internet está tratando de enviar un paquete IP a un *host* en una red SNA adyacente. Se podría pensar en una conversión entre direcciones IP y SNA en ambas direcciones. Además, el concepto de lo que es direccionable es diferente. En IP, los *hosts* (en realidad las tarjetas de red) tienen direcciones. En SNA, entidades diferentes a los *hosts* (por ejemplo dispositivos de hardware) pueden también tener direcciones. En el mejor de los casos, alguien tendría que mantener una base de

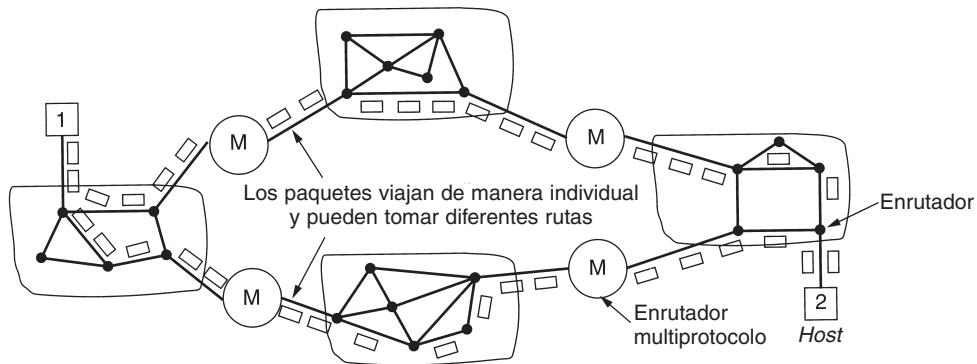


Figura 5-46. Una interred no orientada a la conexión.

datos de las conversiones de todo a todo en la medida de lo posible, pero esto sería constantemente una fuente de problemas.

Otra idea es diseñar un paquete universal de “interred” y hacer que todos los enrutadores lo reconozcan. Este enfoque es, precisamente, el que tiene IP: un paquete diseñado para llevarse por muchas redes. Por supuesto, podría suceder que IPv4 (el protocolo actual de Internet) expulse del mercado a todos los formatos, que Ipv6 (el protocolo futuro de Internet) no se vuelva popular y que no se invente nada más, pero la historia sugiere otra cosa. Hacer que todos se pongan de acuerdo en un solo formato es difícil, más aún cuando las empresas consideran que es una ventaja para ellas contar con un formato patentado bajo su control.

Recapitulemos ahora brevemente las dos maneras en que puede abordarse la interconectividad de redes. El modelo de circuitos virtuales concatenados tiene en esencia las mismas ventajas que el uso de circuitos virtuales en una sola subred: pueden reservarse búferes por adelantado, puede garantizarse la secuencia, pueden usarse encabezados cortos y pueden evitarse los problemas causados por paquetes duplicados retrasados.

El modelo también tiene las mismas desventajas: el espacio de tablas requerido en los enrutadores para cada conexión abierta, la falta de enrutamiento alterno para evitar áreas congestionadas y la vulnerabilidad a fallas de los enrutadores a lo largo de la ruta. También tiene la desventaja de que su implementación es difícil, si no imposible, si una de las redes que intervienen es una red no confiable de datagramas.

Las propiedades del enfoque por datagramas para la interconectividad son las mismas que las de las subredes de datagramas: un mayor potencial de congestión, pero también mayor potencial para adaptarse a él, la robustez ante fallas de los enrutadores y la necesidad de encabezados más grandes. En una interred son posibles varios algoritmos de enrutamiento adaptativo, igual que en una sola red de datagramas.

Una ventaja principal del enfoque por datagramas para la interconectividad es que puede usarse en subredes que no usan circuitos virtuales. Muchas LANs, redes móviles (por ejemplo, flotas

aéreas y navales) e incluso algunas WANs caen en esta categoría. Cuando una interred incluye una de éstas, surgen serios problemas si la estrategia de interredes se basa en circuitos virtuales.

5.5.5 Entunelamiento

El manejo del caso general de lograr la interacción de dos redes diferentes es en extremo difícil. Sin embargo, hay un caso especial común que puede manejarse. Este caso es cuando el *host* de origen y el de destino están en la misma clase de red, pero hay una red diferente en medio. Como ejemplo, piense en un banco internacional con una Ethernet basada en TCP/IP en París, una Ethernet basada en TCP/IP en Londres y una WAN no IP (por ejemplo, ATM) en medio, como se muestra en la figura 5-47.

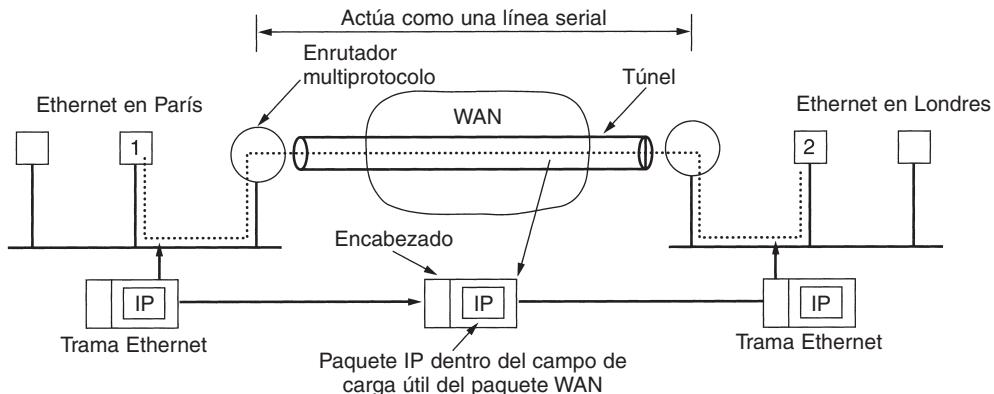


Figura 5-47. Entunelamiento de un paquete de París a Londres.

La solución a este problema es una técnica llamada **entunelamiento**. Para enviar un paquete IP al *host* 2, el *host* 1 construye el paquete que contiene la dirección IP del *host* 2, a continuación lo inserta en una trama Ethernet dirigida al enrutador multiprotocolo de París y, por último, lo pone en la línea Ethernet. Cuando el enrutador multiprotocolo recibe la trama, retira el paquete IP, lo inserta en el campo de carga útil del paquete de capa de red de la WAN y dirige este último a la dirección de la WAN del enrutador multiprotocolo de Londres. Al llegar ahí, el enrutador de Londres retira el paquete IP y lo envía al *host* 2 en una trama Ethernet.

La WAN puede visualizarse como un gran túnel que se extiende de un enrutador multiprotocolo al otro. El paquete IP simplemente viaja de un extremo del túnel al otro, bien acomodado en una caja bonita. No tiene que preocuparse por lidiar con la WAN. Tampoco tienen que hacerlo los *hosts* de cualquiera de las Ethernets. Sólo el enrutador multiprotocolo tiene que entender los paquetes IP y WAN. De hecho, la distancia completa entre la mitad de un enrutador multiprotocolo y la mitad del otro actúa como una línea serial.

El entunelamiento puede aclararse mediante una analogía. Considere una persona que maneja su auto de París a Londres. En Francia, el auto se mueve con su propia energía, pero al llegar al Canal de la Mancha, se carga en un tren de alta velocidad y se transporta a Inglaterra a través del Chunnel (los autos no pueden conducirse a través del Chunnel). En efecto, el auto se transporta como carga, como se muestra en la figura 5-48. En el otro extremo, se libera el auto en las carreteras inglesas y nuevamente continúa moviéndose con sus propios medios. El entunelamiento de paquetes a través de una red foránea funciona de la misma manera.

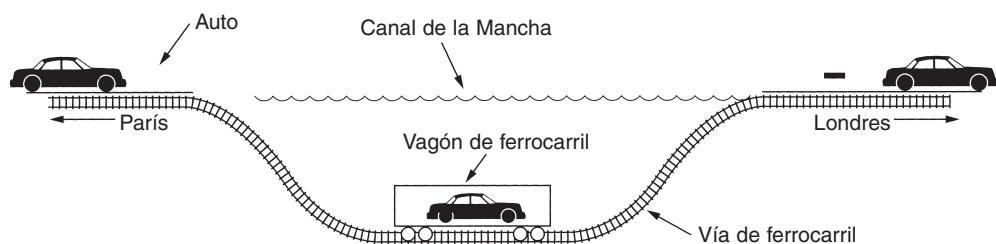


Figura 5-48. Paso de un auto de Francia a Inglaterra a través de un túnel.

5.5.6 Enrutamiento entre redes

El enrutamiento a través de una interred es parecido al enrutamiento en una sola subred, pero con algunas complicaciones adicionales. Por ejemplo, considere la interred de la figura 5-49(a) en la que cinco redes están conectadas mediante seis enrutadores (posiblemente multiprotocolo). Realizar un modelo de grafo de esta situación es complicado por el hecho de que cada enrutador multiprotocolo puede acceder (es decir, enviar paquetes) de manera directa a todos los demás enrutadores conectados a cualquier red a la que esté conectado. Por ejemplo, *B* en la figura 5-49(a) puede acceder directamente a *A* y a *C* a través de la red 2, y también a *D* a través de la red 3. Esto conduce al grafo de la figura 5-49(b).

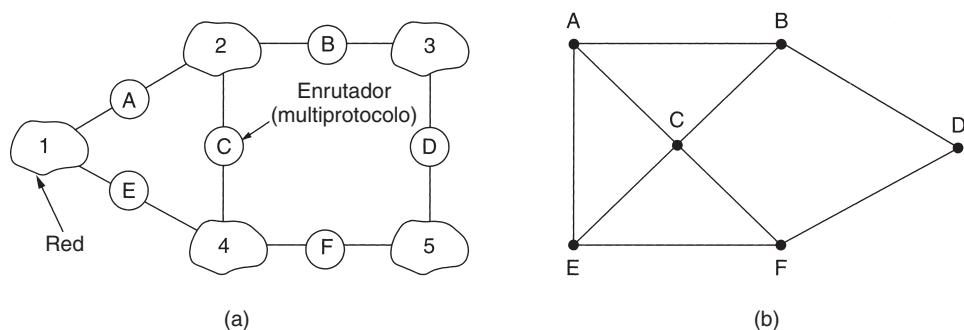


Figura 5-49. (a) Una interred. (b) Grafo de la interred.

Una vez construido el grafo, pueden aplicarse algoritmos de enrutamiento conocidos, como el algoritmo de vector de distancia y el de estado del enlace, al grupo de enrutadores multiprotocolo. Esto da un algoritmo de enrutamiento de dos niveles: en cada red se utiliza un **protocolo de puerta de enlace interior (IGP)**, pero entre ellas se usa un **protocolo de puerta de enlace exterior (EGP)** (“puerta de enlace” es un término antiguo para “enrutador”). De hecho, debido a que estas redes son independientes, cada una puede utilizar un algoritmo diferente del de la otra. Puesto que cada red de una interred es independiente de las demás, con frecuencia se le llama **sistema autónomo (AS)**.

Un paquete de interred típico parte de su LAN hacia el enrutador multiprotocolo local (en el encabezado de la capa de MAC). Al llegar ahí, el código de la capa de red decide por cuál enrutador multiprotocolo reenviará el paquete, usando sus propias tablas de enrutamiento. Si ese enrutador puede alcanzarse usando el protocolo de la red nativa del paquete, éste se reenvía directamente ahí. De otra manera, se envía por túnel, encapsulado en el protocolo requerido por la red que interviene. Este proceso se repite hasta que el paquete llega a la red de destino.

Una de las diferencias del enrutamiento entre las redes y el enrutamiento dentro de las redes es que el primero con frecuencia requiere el cruce de fronteras internacionales. De pronto, entran en escena varias leyes, como las estrictas leyes suecas de confidencialidad sobre la exportación de datos personales de ciudadanos suecos. Otro ejemplo es la ley canadiense que indica que el tráfico de datos que se origina en Canadá y llega a un destino en Canadá no puede dejar el país. Esto significa que el tráfico de Windsor, Ontario a Vancouver no puede enrutararse a través de Detroit, Estado Unidos, incluso si esta ruta es más rápida y barata.

Otra diferencia entre el enrutamiento interior y el exterior es el costo. Dentro de una sola red, normalmente se aplica un solo algoritmo de cargo. Sin embargo, redes diferentes pueden estar bajo administraciones diferentes, un una ruta puede ser menos cara que otra. Del mismo modo, la calidad de servicio ofrecida por diferentes redes puede ser distinta, y ésta puede ser una razón para escoger una ruta y no otra.

5.5.7 Fragmentación

Cada red impone un tamaño máximo a sus paquetes. Estos límites tienen varias razones, entre ellas:

1. El hardware (por ejemplo, el tamaño de una trama Ethernet).
2. El sistema operativo (por ejemplo, todos los búferes son de 512 bytes).
3. Los protocolos (por ejemplo, la cantidad de bits en el campo de longitud de paquete).
4. El cumplimiento de algún estándar (inter)nacional.
5. El deseo de reducir hasta cierto nivel las retransmisiones inducidas por errores.
6. El deseo de evitar que un paquete ocupe el canal demasiado tiempo.

El resultado de estos factores es que los diseñadores de redes no están en libertad de escoger cualquier tamaño máximo de paquetes que deseen. Las cargas útiles máximas van desde 48 bytes (celadas ATM) hasta 65,515 bytes (paquetes IP), aunque el tamaño de la carga útil en las capas superiores con frecuencia es más grande.

Surge un problema obvio cuando un paquete grande quiere viajar a través de una red cuyo tamaño máximo de paquete es demasiado pequeño. Una solución es asegurar que no ocurra el problema. En otras palabras, la interred debe usar un algoritmo de enrutamiento que evite el envío de paquetes a través de redes que no pueden manejarlos. Sin embargo, esta solución en realidad no es una solución. ¿Qué ocurre si el paquete original es demasiado grande para ser manejado por la red de destino? El algoritmo de enrutamiento no puede pasar por alto el destino.

Básicamente, la única solución al problema es permitir que las puertas de enlace dividan los paquetes en **fragmentos**, enviando cada paquete como paquete de interred individual. Sin embargo, como lo sabe cualquier padre de un niño pequeño, la conversión de un objeto grande en fragmentos pequeños es significativamente más fácil que el proceso inverso. (Los físicos incluso le han dado un nombre a este efecto: segunda ley de la termodinámica.) Las redes de comutación de paquetes también tienen problemas al unir nuevamente los fragmentos.

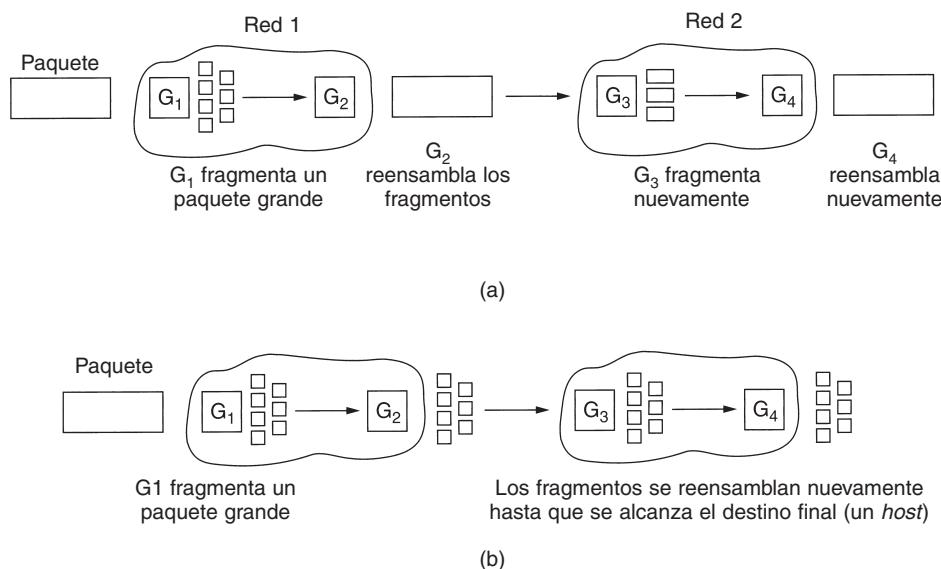


Figura 5-50. (a) Fragmentación transparente. (b) Fragmentación no transparente.

Existen dos estrategias opuestas para recombinar los fragmentos y recuperar el paquete original. La primera es hacer transparente la fragmentación causada por una red de “paquete pequeño” a las demás redes subsiguientes por las que debe pasar el paquete para llegar a su destino final. Esta opción se muestra en la figura 5-50(a). Con este método, la red de paquete pequeño tiene puertas de enlace (lo más probable es que sean enrutadores especializados) que interactúan con