

Redes

UNIVERSIDAD DE LA PUNTA



Tema 3 |
Tema 4 |

Contenidos

Bases teóricas para la comunicación de datos	5
Análisis de Fourier	5
Señales de ancho de banda limitado.....	5
La tasa de datos máxima de un canal	8
Medios de transmisión	9
Medios de transmisión guiados.....	9
Medios magnéticos.....	9
Par trenzado.....	9
Cable coaxial	11
Líneas eléctricas	11
Fibra óptica.....	12
Transmisión inalámbrica	15
El espectro electromagnético	16
Radiotransmisión.....	18
Transmisión por microondas.....	19
Transmisión infrarroja	21
Transmisión por ondas de luz	21
Satélites de comunicación.....	21
Satélites geoestacionarios	22
Satélites de Órbita Terrestre Media (MEO).....	24
Satélites de Órbita Terrestre Baja (LEO).....	24
Comparación de los satélites y la fibra óptica.....	25
Modulación digital y multiplexación	25
Transmisión en banda base.....	26
Transmisión pasa-banda	29
Multiplexación por división de frecuencia	31
Multiplexación por división de tiempo	33
Multiplexación por división de código	33
Principales protocolos de capa física.....	35
Ethernet.....	35
Objetivos de Ethernet.....	36
Características de Ethernet.....	36
Principios de operación de Ethernet	36
Direccionamiento.....	37

Tiempo de señales.....	37
WiFi	37
Historia.....	37
Estándares que certifica la Alianza Wi-Fi	38
Seguridad y fiabilidad	38
Dispositivos.....	39
Ventajas y desventajas.....	39
Capa de enlace de datos.....	40
Cuestiones de diseño de la capa de enlace de datos.....	40
Servicios proporcionados a la capa de red.....	41
Entramado	42
Control de errores	45
Control de flujo	45
Detección y corrección de errores.....	46
Códigos de corrección de errores	47
Códigos de detección de errores	51
Protocolos elementales de enlace de datos	56
Un protocolo simplex utópico.....	59
Protocolo simplex de parada y espera para un canal libre de errores	60
Protocolo simplex de parada y espera para un canal ruidoso	62
Protocolos de ventana deslizante	65
Un protocolo de ventana deslizante de un bit	66
Un protocolo que utiliza retroceso N.....	68
Un protocolo que usa repetición selectiva	74
Ejemplos de protocolos de enlace de datos.....	78
HDL.....	78
Características básicas del HDLC.....	78
Estructura	79
Campo de control.....	80
Campo de información.....	80
Campo para la secuencia de comprobación de la trama.....	80
Funcionamiento del HDLC	80
PPP.....	81
Descripción.....	81
Configuración automática	82
Múltiples protocolos de la capa de red	82

Opciones de configuración de PPP	82
Funcionamiento.....	82
Fases del PPP y activación de la línea.....	83
La subcapa de acceso al medio	84
El problema de asignación del canal.....	84
Asignación estática de canal.....	84
Supuestos para la asignación dinámica de canales.....	85
Protocolos de acceso múltiple.....	86
ALOHA	86
Protocolos de acceso múltiple con detección de portadora.....	89
Protocolos libres de colisiones.....	92
Protocolos de contención limitada	95
Protocolos de LAN inalámbrica.....	97
ETHERNET	99
Capa física de Ethernet clásica	100
El protocolo de subcapa MAC de la Ethernet clásica.....	101
Desempeño de Ethernet.....	103
Ethernet commutada.....	104
FastEthernet.....	106
GigabitEthernet.....	108
10 GigabitEthernet	110
Redes LAN inalámbricas.....	111
La arquitectura de 802.11 y la pila de protocolos.....	111
La capa física del estándar 802.11	112
El protocolo de la subcapa MAC del 802.11	114
La estructura de trama 802.11	118
Servicios.....	119
Commutación en la capa de enlace de datos	120
Usos de los puentes.....	120
Puentes de aprendizaje	121
Puentes con árbol de expansión	124
Repetidores, hubs, puentes, switchs, enrutadores y puertas de enlace (gateways).....	125
Redes LAN virtuales.....	127
Referencias	132

Análisis de equipos 133

Capa física	133
Concentrador.....	133
Repetidor.....	135
Par trenzado – Tipos de cableado	136
Capa de enlace de datos	138
NIC.....	138
Puente.....	140
Comutador	142
Capas superiores.....	144
Router.....	144
Gateway	146
Qué son el dominio de Colisión y el Dominio de Difusión	147
Dominio de colisión	147
Dominio de difusión (Broadcast).....	147
Cómo actúan los dispositivos de red dividiendo los dominios de colisión y difusión	147
Ejemplo	147
Control de acceso al medio	149
Funciones realizadas en la subcapa MAC	149
Mecanismo de direccionamiento	149
Mecanismo de control de acceso al canal	150
Análisis de protocolos.....	151
Capa de enlace de datos	151
Spanning Tree Protocol.....	151
CSMA/CS.....	154
Bibliografía y referencias	158

Bases teóricas para la comunicación de datos

En esta unidad empezaremos analizando la capa más baja en nuestro modelo de protocolos: la capa física. Ésta define las interfaces eléctricas, de temporización y demás interfaces mediante las cuales se envían los bits como señales a través de los canales. La capa física es la base sobre la cual se construye la red. Las propiedades de distintos tipos de canales físicos determinan el desempeño (rendimiento, latencia, tasa de error, etc.).

Mediante la variación de alguna propiedad física, como el voltaje o la corriente, es posible transmitir información a través de cables. Si representamos 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.

Análisis de Fourier

A principios del siglo XIX, el matemático francés *Jean-Baptiste Fourier* demostró que cualquier función periódica de comportamiento razonable, $g(t)$ con un periodo T , se puede construir como la suma de un número (posiblemente infinito) de funciones senos y cosenos:

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

en donde $f=1/T$ es la frecuencia fundamental, a_n y b_n son las amplitudes de seno y coseno del ***n*-ésimo armónico** (término) y c es una constante. A dicha descomposición se le denomina serie de Fourier. Podemos reconstruir la función a partir de la serie de Fourier, es decir, si se conoce el periodo T y se dan las amplitudes, podemos encontrar la función original del tiempo realizando las sumas de la ecuación anterior.

Es posible pensar una señal de datos con una duración finita (todas la tienen) como una señal periódica, con sólo imaginar que el patrón completo se repite de manera indefinida (es decir, el intervalo de T a $2T$ es el mismo que de 0 a T , etc.).

Podemos calcular las amplitudes a_n para cualquier $g(t)$ si multiplicamos ambos lados de la ecuación por $\sin(2\pi kft)$ y después integramos de 0 a T . Dado que

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

sólo sobrevive un término de la sumatoria: a_n . La sumatoria b_n se desvanece por completo. De forma similar, si multiplicamos la ecuación por $\cos(2\pi kft)$ e integramos 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:

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

Señales de ancho de banda limitado

La relevancia de todo esto para la comunicación de datos es que los canales reales afectan a las señales de distintas frecuencia de manera diferente. Consideremos un ejemplo específico: la transmisión del carácter ASCII "b" codificado en un byte de 8 bits. El patrón de bits que transmitirá es 01100010. La parte izquierda de la Ilustración 1 muestra la salida de tensión eléctrica producido por la computadora transmisora. El análisis de Fourier de esta señal produce los siguientes coeficientes:

$$\begin{aligned} a_n &= \frac{1}{\pi n} \left[\cos\left(\frac{\pi n}{4}\right) - \cos\left(\frac{3\pi n}{4}\right) + \cos\left(\frac{6\pi n}{4}\right) - \cos\left(\frac{7\pi n}{4}\right) \right] \\ b_n &= \frac{1}{\pi n} \left[\sin\left(\frac{3\pi n}{4}\right) - \sin\left(\frac{\pi n}{4}\right) + \sin\left(\frac{7\pi n}{4}\right) - \sin\left(\frac{6\pi n}{4}\right) \right] \\ c &= \frac{3}{4} \end{aligned}$$

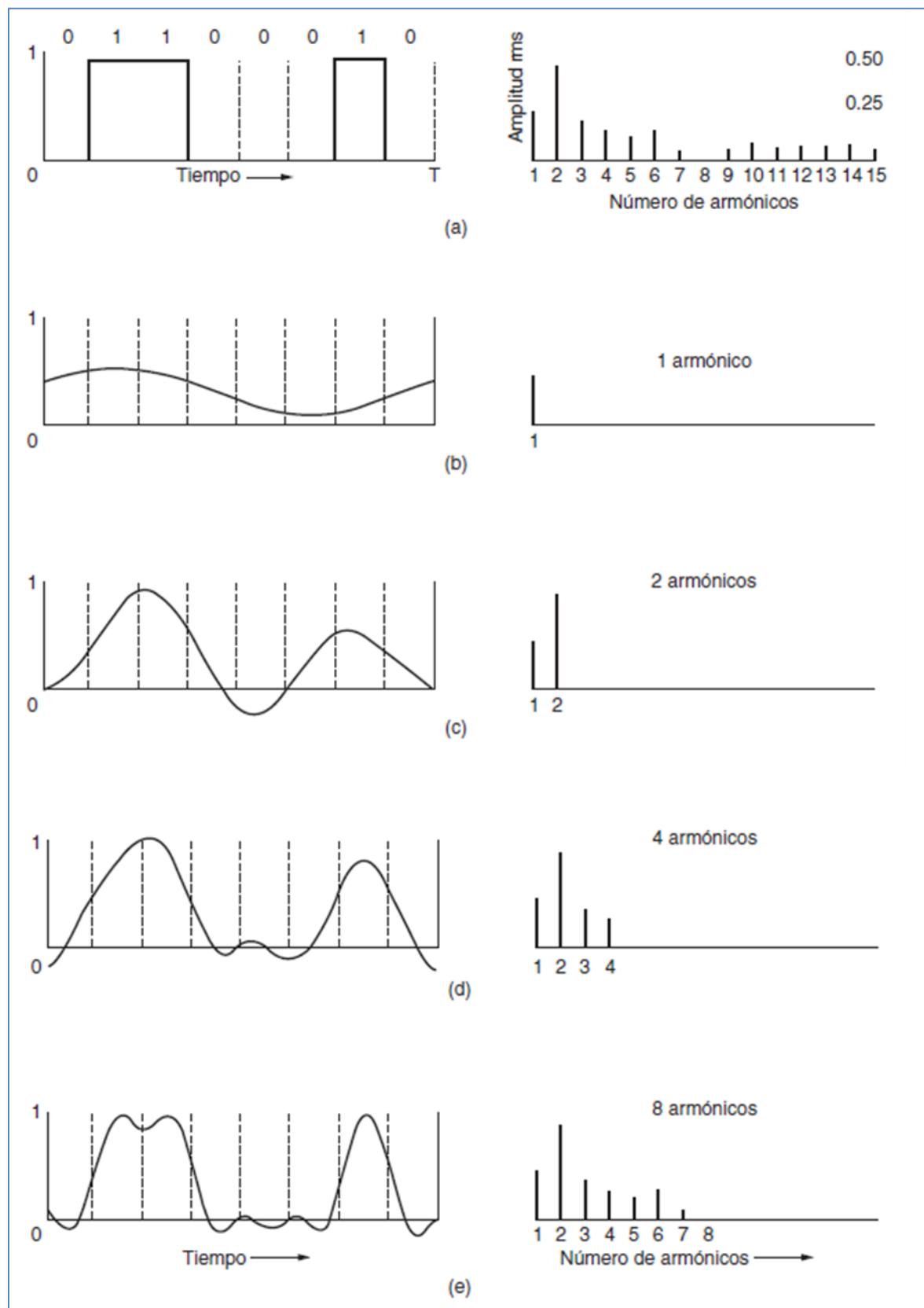


Ilustración 1 - Señal binaria y su aproximación

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

Ninguna instalación transmisora puede enviar 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 de la Ilustración 1(a)). Por desgracia, todas las instalaciones transmisoras disminuyen los componentes de Fourier en distinto grado y, en consecuencia, introducen distorsión. Por lo general, las amplitudes se transmiten en su mayoría sin ninguna disminución en un cable, desde cero hasta cierta frecuencia f_c (se mide en ciclos/segundo o Hertz (Hz)), y se atenúan todas las frecuencias que están por encima de esta frecuencia de corte. El rango de frecuencia que se transmite sin una atenuación considerable se denomina **ancho de banda**. En la práctica, el corte en realidad no es muy abrupto, por lo que a menudo el ancho de banda referido es desde cero hasta la frecuencia a la que disminuyó la potencia recibida a la mitad.

El ancho de banda es una propiedad física del medio de transmisión que depende; por ejemplo, de la construcción, el grosor y la longitud de un cable o fibra óptica. A menudo se utilizan filtros para limitar el ancho de banda de una señal. Por ejemplo, los canales inalámbricos 802.11 pueden utilizar aproximadamente 20 MHz, por lo que los radios 802.11 filtran el ancho de banda de la señal con base en este tamaño. Otro ejemplo, los canales de televisión tradicionales (analógicos) ocupan 6 MHz cada uno, en un cable o a través del aire. Este filtrado permite que más señales compartan una región dada de un espectro, lo cual mejora la eficiencia del sistema en general. Lo que significa que el rango de frecuencia para ciertas señales no empezará en cero, pero no importa. El ancho de banda sigue siendo el rango de la banda de frecuencias que se transmiten, y la información que se puede transportar depende sólo de este ancho de banda, no de su frecuencia inicial ni final. Las señales que van desde cero hasta una frecuencia máxima se llaman señales de **banda base**. Las que se desplazan para ocupar un rango de frecuencias más altas, como es el caso de todas las transmisiones inalámbricas, se llaman señales de **pasa-banda**.

Ahora consideremos cómo luciría la señal de la Ilustración 1(a) si el ancho de banda fuera tan pequeño que sólo se transmitieran las frecuencias más bajas (es decir, que la función se aproxima mediante los primeros términos de la primera ecuación). La Ilustración 1(b) muestra la señal que resulta de un canal que sólo permite el paso del primer armónico (la frecuencia fundamental, f). De manera similar, las Ilustración 1(c)-(e) muestran los espectros y las funciones reconstruidas para canales de mayor ancho de banda. Para la transmisión digital, el objetivo es recibir una señal con la suficiente fidelidad como para poder reconstruir la secuencia de bits que se envió. Se puede hacer esto con facilidad en la Ilustración 1(e), por lo que sería un desperdicio usar más armónicos para recibir una réplica más precisa.

Si tenemos una tasa de bits de b bits/seg, el tiempo requerido para enviar los 8 bits en nuestro ejemplo, 1 bit a la vez, es de $8/b$ segundos, por lo que la frecuencia del primer armónico de esta señal es $b/8$ Hz. Una línea telefónica común, a menudo llamada **línea con calidad de voz**, tiene una frecuencia de corte introducida en forma artificial ligeramente mayor a 3000 Hz. Esta restricción significa que el número de armónicos más altos que puede pasar es de aproximadamente $3000/(b/8)$, o $24000/b$ (el corte no es muy abrupto).

Para algunas tasas de datos, los números resultan como se muestra en la Ilustración 2. De estos números queda claro que tratar de transmitir a 9600 bps a través de una línea telefónica con calidad de voz, transformará la Ilustración 1(a) en algo parecido a la Ilustración 1(c), lo cual dificultará la recepción precisa del flujo de bits original. Debe ser obvio que a tasas de datos mucho mayores que 38.4 kbs no hay esperanza alguna para las señales binarias, aun cuando la instalación transmisora se encuentre totalmente libre de ruidos. En otras palabras, al limitar el ancho de banda se limita la tasa de datos, incluso en canales perfectos. Sin embargo, existen esquemas de codificación que utilizan diferentes niveles de voltaje y logran tasas de datos más altas. Más adelante en este capítulo veremos estos esquemas.

Bps	T (mseg)	Primer armónico (Hz)	Núm. de armónico transmitidos
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

Ilustración 2 - Relación Tasa de datos - Armónicos para el ejemplo dado

Hay mucha confusión en cuanto al ancho de banda, ya que tiene distintos significados para los ingenieros eléctricos y para los científicos de computadoras. Para los ingenieros eléctricos, el ancho de banda (analógico) es (como lo describimos antes) una cantidad que se mide en *Hz*. Para los científicos de computadora, el ancho de banda (digital) es la tasa de datos máxima de un canal, una cantidad que se mide en *bits/segundo*. Esta tasa de datos es el resultado final de usar el ancho de banda analógico de un canal físico para transmisión digital, y ambos están relacionados, como veremos a continuación.

La tasa de datos máxima de un canal

En 1924, un ingeniero de AT&T llamado Henry Nyquist se dio cuenta de que incluso un canal perfecto tiene una capacidad de transmisión finita y dedujo una ecuación para expresar la tasa de datos máxima para un canal sin ruido con un ancho de banda finito. En 1948, Claude Shannon retomó el trabajo de Nyquist y lo extendió al caso de un canal sujeto a ruido aleatorio (es decir, termodinámico) (Shannon, 1948). Este documento es el más importante en toda la teoría de la información. Aquí sólo resumiremos brevemente sus resultados, que ahora son clásicos.

Nyquist demostró que si se pasa una señal cualquiera a través de un filtro pasa-bajas con un ancho de banda *B*, la señal filtrada se puede reconstruir por completo tomando sólo $2B$ muestras (exactas) por segundo. No tiene caso muestrear la línea más de $2B$ veces por segundo, ya que los componentes de mayor frecuencia que dicho muestreo pudiera recuperar ya se han filtrado. Si la señal consiste en *V* niveles discretos, el teorema de Nyquist establece lo siguiente:

$$\text{Tasa de datos máxima} = 2B \log_2 V \left[\frac{\text{bits}}{\text{seg}} \right]$$

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

Hasta ahora hemos considerado sólo los canales sin ruido. Si hay ruido aleatorio presente, la situación se deteriora con rapidez. Y siempre hay ruido aleatorio (térmico) presente debido al movimiento de las moléculas en el sistema. La cantidad de ruido térmico presente se mide con base en la relación entre la potencia de la señal y la potencia del ruido; llamada SNR (Relación Señal a Ruido, del inglés *Signal-to-Noise Ratio*). Si denotamos la potencia de la señal mediante *S* y la potencia del ruido mediante *N*, la relación señal a ruido es *S/N*. Por lo general la relación se expresa en una escala de logaritmo como la cantidad $10\log_{10} S/N$, ya que puede variar sobre un gran rango. Las unidades de esta escala logarítmica se llaman **decibeles (dB)**, en donde “deci” significa 10 y “bel” se eligió en honor a Alexander Graham Bell, inventor del teléfono. Una relación *S/N* de 10 es igual a 10 dB, una relación de 100 es igual a 20 dB, una relación de 1000 es igual a 30 dB y así sucesivamente. A menudo los fabricantes de amplificadores caracterizan el ancho de banda (rango de frecuencia) en el cual sus productos son lineales dando la frecuencia de 3 dB en cada extremo. Éstos son los puntos en los que el factor de amplificación se ha dividido de manera aproximada a la mitad (puesto que $10\log_{10} 0.5 \approx -3$).

El principal resultado de Shannon es que la tasa de datos máxima (o **capacidad**) de un canal ruidoso, cuyo ancho de banda es B Hz y cuya relación señal a ruido es S/N , está dada por:

$$\text{Número máximo de bits/seg} = B \log_2 \left(1 + \frac{S}{N} \right)$$

Esto nos indica las mejores capacidades que pueden tener los canales reales. Por ejemplo, la **ADSL (Línea Asimétrica de Suscriptor Digital)** que provee acceso a Internet a través de líneas telefónicas comunes, utiliza un ancho de banda de aproximadamente 1 MHz. La SNR depende en gran parte de la distancia entre el hogar y la central telefónica; una SNR de alrededor de 40 dB para líneas cortas de 1 a 2 km es algo muy bueno. Con estas características, el canal nunca podrá transmitir a más de 13 Mbps, sin importar cuántos niveles de señal se utilicen ni con qué frecuencia se tomen las muestras. En la práctica, el servicio ADSL se especifica hasta 12 Mbps, aunque es frecuente que los usuarios vean tasas más bajas.

Medios de transmisión

El propósito de la capa física es transportar bits de una máquina a otra. Se pueden utilizar varios medios físicos para la transmisión real. Cada medio tiene sus propias ventajas y desventajas en términos de ancho de banda, retardo, costo y facilidad de instalación y mantenimiento, siendo cada uno más conveniente que otro dependiendo de la red a construir. A grandes rasgos, los medios se agrupan en medios guiados (como el cable de cobre y la fibra óptica) y en medios no guiados (como la transmisión inalámbrica terrestre, los satélites y los láseres a través del aire).

Medios de transmisión guiados

Medios magnéticos

Una de las formas más comunes para transportar datos de una computadora a otra es almacenarlos en cinta magnética o medios removibles (por ejemplo, DVD regrabables, discos duros), transportar físicamente la cinta o los discos a la máquina de destino y leerlos de nuevo. Aunque este método no es tan sofisticado como usar un satélite de comunicación geosíncrono, a menudo es mucho más rentable, en especial para las aplicaciones en las que el ancho de banda alto o el costo por bit transportado es el factor clave.

Para aclarar este punto veremos un simple cálculo. Un disco duro externo con una capacidad de 4 Tb tiene una dimensión 10x8x3 cm aproximadamente. Una caja de 60x60x60 cm puede contener aproximadamente 850 de estos discos, para una capacidad total de 3400 Terabytes, o 27200 Terabits (27.2 petabits). Una caja de discos se puede entregar en cualquier parte del país en un plazo de 48 horas a través de cualquier empresa de mensajería. El ancho de banda efectivo de esta transmisión es de 27200 Terabits/172800 seg, o de 157 Gbps. Si el destino sólo está a una hora de camino, el ancho de banda se incrementa a cerca de 7500 Gbps. Ninguna red de computadoras puede siquiera acercarse a esta velocidad. Claro que las redes se están haciendo más veloces, pero las densidades de los discos también están aumentando.

Si ahora analizamos el costo, obtenemos una imagen similar. El costo aproximado de un disco es de \$6000 si se compra al mayoreo. Si agregamos \$2000 de envío (probablemente sea menos), el costo aproximado de enviar un gigabyte es aproximadamente \$1,5 (Considerando que los discos pueden reutilizarse un gran número de veces, proyectando en el tiempo, el costo sería menor a un centavo). Ninguna red puede vencer eso. La moraleja de la historia es:

Nunca subestime el ancho de banda de una camioneta repleta de discos duros portátiles a toda velocidad por la autopista.

Par trenzado

Aunque las características de ancho de banda de los medios magnéticos son excelentes, las características 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 antiguos y todavía el

más común es el par trenzado. Un par trenzado consta de dos cables de cobre aislados, por lo general de 1mm de grosor. Los cables están trenzados en forma helicoidal, justo igual que una molécula de ADN. El trenzado se debe a que dos cables paralelos constituyen una antena simple. Cuando se trenzan los cables, las ondas de distintos trenzados se cancelan y el cable irradia con menos efectividad. Por lo general una señal se transmite como la diferencia en el voltaje entre los dos cables en el par. Esto ofrece una mejor inmunidad al ruido externo, ya que éste tiende a afectar ambos cables en la misma proporción y en consecuencia, el diferencial queda sin modificación.

La aplicación más común del par trenzado es el sistema telefónico. Casi todos los teléfonos se conectan a la central telefónica mediante un par trenzado. Tanto las llamadas telefónicas como el acceso ADSL a Internet se llevan a cabo mediante estas líneas. Se pueden tender varios kilómetros de par trenzado sin necesidad de amplificación, pero en distancias mayores la señal se atenúa demasiado y se requieren repetidores. Cuando muchos pares trenzados se tienden en paralelo a una distancia considerable, como los cables que van de un edificio de departamentos a la central telefónica, se agrupan en un haz y se cubren con una funda protectora. Los pares en estos haces interferirían unos con otros si no estuvieran trenzados. En algunas partes del mundo en donde las líneas telefónicas penden de postes sobre la tierra, es común ver haces de varios centímetros de diámetro.

Los pares trenzados se pueden usar para transmitir la información analógica o digital. El ancho de banda depende del grosor del cable y de la distancia que recorre, pero en muchos casos se pueden lograr varios megabits/seg durante pocos kilómetros. Debido a su adecuado desempeño y bajo costo, los pares trenzados se utilizan mucho y es probable que se sigan utilizando durante varios años más.

Existen diversos tipos de cableado de par trenzado. El que se utiliza con mayor frecuencia en muchos edificios de oficinas se llama cable de categoría 5, o “cat 5”. Un par trenzado de categoría 5 consta de dos cables aislados que se trenzan de manera delicada. Por lo general se agrupan cuatro de esos pares en una funda de plástico para protegerlos y mantenerlos juntos. Este arreglo se muestra en la Ilustración 5.

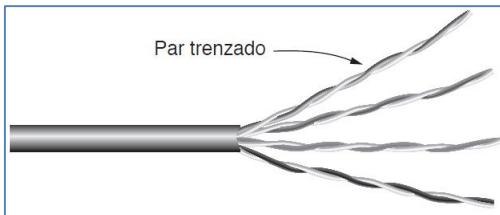


Ilustración 3 - Cable UTP categoría 5 con cuatro pares trenzados.

Los distintos estándares de LAN pueden utilizar los pares trenzados de manera diferente. Por ejemplo, el estándar Ethernet de 100 Mbps utiliza dos (de los cuatro) pares, uno para cada dirección. Para llegar a velocidades más altas, el estándar Ethernet de 1 Gbps utiliza los cuatro pares en ambas direcciones al mismo tiempo; para ello el receptor debe eliminar la señal que se transmite en forma local.

Los cables de categoría 5 reemplazaron a los cables de categoría 3 con un cable similar que utiliza el mismo conector, pero tiene más trenzas por metro. Entre más trenzas, hay menos diafonía (*def.* Perturbación electromagnética producida en un canal de comunicación por el acoplamiento de este con otro u otros vecinos) y se logra una señal de mejor calidad a distancias más largas, lo que hace a los cables más adecuados para la comunicación de computadoras de alta velocidad, en especial para las redes LAN de 100 Mbps y de 1 Gbps.

Es muy probable que el nuevo cableado sea de categoría 6 o incluso de categoría 7. Estas categorías tienen especificaciones más estrictas para manejar señales con mayores anchos de banda. Algunos cables de categoría 6 y superiores están estimados para señales de 500 MHz y pueden soportar los enlaces de 10 Gbps que se implementarán en un futuro cercano.

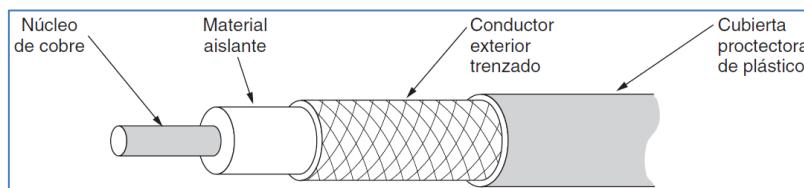
A los tipos de cables hasta la categoría 6 se les conoce como UTP (Par Trenzado sin Blindaje, del inglés *Unshielded Twisted Pair*), ya que están constituidos tan sólo de alambres y aislantes. En contraste, los cables de categoría 7 y superior, tienen blindaje en cada uno de los pares trenzados por separado, así como alrededor

de todo el cable (pero dentro de la funda protectora de plástico). El blindaje reduce la susceptibilidad a interferencias externas y la diafonía con otros cables cercanos para cumplir con las especificaciones más exigentes de rendimiento.

Cable coaxial

El cable coaxial es otro medio de transmisión. Este cable tiene mejor blindaje y mayor ancho de banda que los pares trenzados sin blindaje, por lo que puede abarcar mayores distancias a velocidades más altas. Hay dos tipos de cable coaxial que se utilizan ampliamente. El de 50 ohms es uno de ellos y se utiliza por lo general cuando se tiene pensado emplear una transmisión digital desde el inicio. El otro tipo es el de 75 ohms y se utiliza para la transmisión analógica y la televisión por cable. Esta distinción se basa en factores históricos más que técnicos. A partir de la década de 1990, los operadores de TV por cable empezaron a proveer acceso a Internet por cable, de modo que el cable de 75 ohms se ha vuelto más importante para la comunicación de datos.

Un cable coaxial consiste en alambre de cobre rígido como núcleo, rodeado por un material aislante. El aislante está forrado de un conductor cilíndrico, que por lo general es una malla de tejido fuertemente trenzado. El conductor externo está cubierto con una funda protectora de plástico. En la Ilustración 4 se muestra una vista seccionada de un cable coaxial.



Gracias a su construcción y blindaje, el cable coaxial tiene una buena combinación de un alto ancho de banda y una excelente inmunidad al ruido. El ancho de banda posible depende de la calidad y la longitud del cable. Los cables modernos tienen un ancho de banda de hasta unos cuantos GHz. Los cables coaxiales solían utilizarse mucho dentro del sistema telefónico para las líneas de larga distancia, pero ahora se reemplazaron en su mayoría por fibra óptica en las rutas de largo recorrido. Sin embargo, el cable coaxial se sigue utilizando mucho para la televisión por cable y las redes de área metropolitana.

Líneas eléctricas

Las redes de telefonía y de televisión por cable no son las únicas fuentes de cableado que se pueden reutilizar para la comunicación de datos. Hay otro tipo más común de cableado: las líneas de energía eléctrica. Estas líneas transportan energía eléctrica a las casas, y el cableado eléctrico dentro de las casas distribuye la energía a las tomas de corriente.

El uso de las líneas eléctricas para la comunicación de datos es una idea antigua. Las compañías de electricidad han utilizado las líneas eléctricas para la comunicación de baja velocidad durante varios años (por ejemplo, la medición remota), así como también en el hogar para controlar dispositivos (por ejemplo, el estándar X10). En años recientes surgió un interés renovado por la comunicación de alta velocidad a través de estas líneas, tanto dentro del hogar con una LAN, como fuera de éste para el acceso a Internet de banda ancha. Nos concentraremos en el escenario más común: el uso de los cables eléctricos dentro del hogar.

La conveniencia de usar líneas eléctricas para el trabajo en red debe quedar claro. Simplemente se debe conectar una TV y un receptor a la toma de pared, lo cual debe hacer de todas formas, ya que necesitan electricidad, para que puedan enviar y recibir películas a través del cableado eléctrico. En la Ilustración 5 se muestra esta configuración. No hay otra clavija o radio transmisor. La señal de datos está sobrepuerta en la señal eléctrica de baja frecuencia (en el cable activo o "caliente"), ya que ambas señales usan el cableado al mismo tiempo.

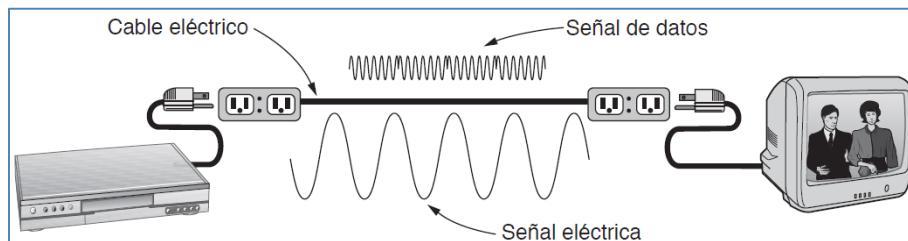


Ilustración 5 - Una red que utiliza el cableado eléctrico en el hogar.

La dificultad al utilizar el cableado eléctrico en el hogar para una red es que éste se diseñó para distribuir señales eléctricas. Esta tarea es muy distinta a la de distribuir señales de datos, en donde el cableado del hogar hace un mal trabajo. Las señales eléctricas se envían a 50-60 Hz y el cableado atenúa las señales de frecuencia mucho más altas (MHz) que se necesitan para la comunicación de datos de alta velocidad. Las propiedades eléctricas del cableado varían de una casa a la otra y cambian a medida que los electrodomésticos se encienden y apagan, lo cual hace que las señales de datos reboden alrededor del cableado. Las corrientes transitorias que se producen al encender y apagar los electrodomésticos crean ruido eléctrico a través de un amplio rango de frecuencias. Y sin el cuidadoso trenzado de los pares trenzados, el cableado eléctrico actúa como una antena simple que recoge las señales externas e irradia sus propias señales. Este comportamiento significa que para cumplir con los requerimientos regulatorios, la señal de datos debe excluir las frecuencias que se otorgan bajo licencia, como las bandas de radioaficionados.

A pesar de estas dificultades, es práctico enviar por lo menos 100 Mbps a través del cableado eléctrico en el hogar mediante el uso de esquemas de comunicación que resisten las frecuencias dañadas y las ráfagas de errores. Muchos productos utilizan varios estándares propietarios para las redes de la línea eléctrica, así que los estándares internacionales se encuentran activamente en desarrollo.

Fibra óptica

La fibra óptica se utiliza para la transmisión de larga distancia en las redes troncales, las redes LAN de alta velocidad (aunque hasta ahora el cobre siempre ha logrado ponerse a la par) y el acceso a Internet de alta velocidad como **FTTH** (Fibra para el Hogar, del inglés *Fiber To The Home*). Un sistema de transmisión óptico tiene tres componentes clave: la fuente de luz, el medio de transmisión y el detector. Por convención, 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 conectar una fuente de luz a un extremo de una fibra óptica y un detector al otro extremo, tenemos un sistema de transmisión de datos unidireccional que acepta una señal eléctrica, la convierte y la transmite mediante pulsos de luz, y después 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 si no fuera por un interesante principio de la física. Cuando un rayo de luz pasa de un medio a otro (por ejemplo, de sílice fundida al aire), el rayo se refracta (dobra) en el límite entre el aire y la sílice, como se muestra en la Ilustración 6(a). Aquí vemos un rayo de luz que incide en el límite a un ángulo α_1 que emerge con un ángulo β_1 . El grado de refracción depende de las propiedades de los dos medios (en especial, de sus índices de refracción). Para ángulos de incidencia por encima de cierto valor crítico, la luz se refracta de regreso a la sílice; nada de ella escapa al aire. Por ende, un rayo de luz incidente con un ángulo igual o mayor al crítico queda atrapado dentro de la fibra, como se muestra en la Ilustración 6(b), y se puede propagar por muchos kilómetros prácticamente sin pérdidas.

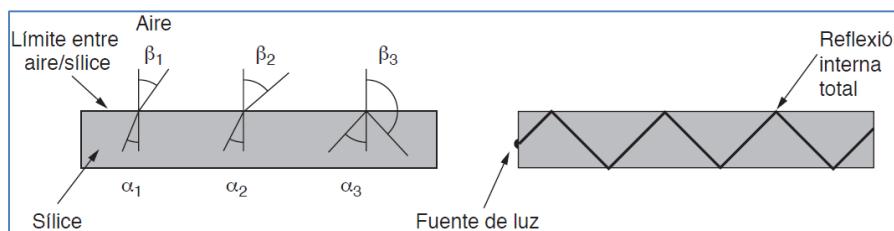


Ilustración 6 - (a) Tres ejemplos de un rayo de luz desde el interior de una fibra de sílice que incide sobre el límite entre aire y sílice a distintos ángulos. (b) Luz atrapada por reflexión interna total.

El bosquejo de la Ilustración 6(b) sólo muestra un rayo atrapado, pero como cualquier rayo de luz que incida en la frontera por encima del ángulo crítico se reflejará de manera interna, habrá muchos rayos distintos rebotando con ángulos diferentes. Se dice que cada rayo tiene un modo distinto, por lo que una fibra con esta propiedad se llama fibra **multimodal**.

Pero 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, con lo que se obtiene una fibra **monomodo**. Estas fibras son más costosas pero se utilizan mucho para distancias más largas. Las fibras monomodo disponibles en la actualidad pueden transmitir datos a 100 Gbps por 100 km sin necesidad de amplificación. Incluso se han logrado tasas de datos más altas en el laboratorio, para distancias más cortas.

Transmisión de luz a través de fibras

Las fibras ópticas están hechas de vidrio, que a su vez se fabrica a partir de la arena, una materia prima de bajo costo disponible en cantidades ilimitadas. La fabricación del vidrio era conocida por los antiguos egipcios, pero su vidrio no podía ser mayor de 1 mm de grosor para que la luz pudiera atravesarlo. Durante el Renacimiento se desarrolló un vidrio lo bastante transparente como para usarlo en las ventanas. El vidrio utilizado para las fibras ópticas modernas es tan transparente que si los océanos estuvieran llenos de él en vez de agua, el lecho marino sería tan visible desde la superficie como lo es el suelo desde un avión en un día claro (Tanembaum, 2009).

La atenuación de la luz que pasa por el vidrio depende de la longitud de onda de la luz (así como de algunas propiedades físicas del vidrio). Se define como la relación entre la potencia de la señal de entrada y la de salida. Para el tipo de vidrio que se utiliza en las fibras ópticas, la atenuación se muestra en la Ilustración 7 en unidades de decibeles por kilómetro lineal de fibra. La figura muestra la parte cercana al infrarrojo del espectro, que es lo 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 equivale a 10^{-6} metros) o 400 nm a 700 nm.

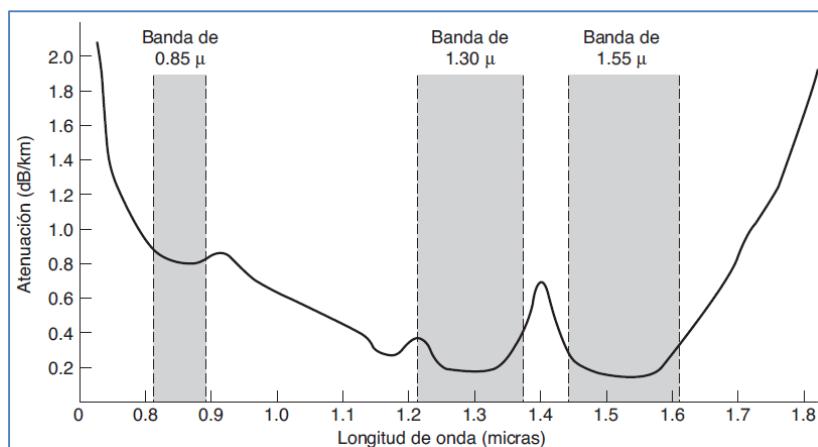


Ilustración 7 - Atenuación de la luz dentro de una fibra en la región de infrarrojo.

En la actualidad se utilizan mucho tres bandas de longitud de onda para la comunicación óptica. Estas tres bandas se centran en 0.85, 1.30 y 1.55 micras, respectivamente. Las tres bandas tienen de 25000 a 30000 GHz de amplitud. La banda de 0.85 micras se utilizó primero. Tiene una mayor atenuación y, por lo tanto, se utiliza para distancias más cortas, pero a esa longitud de onda se pueden fabricar láseres y componentes electrónicos con el mismo material (arseniuro de galio). Las últimas dos bandas tienen buenas propiedades de atenuación (una pérdida de menos de 5% por cada kilómetro). Hoy en día, la banda de 1.55 micrones se utiliza mucho en los amplificadores dopados con erbio que trabajan directamente en el dominio óptico.

La longitud de los pulsos de luz que se transmiten por una fibra aumenta conforme se propagan. A este fenómeno se le conoce como **dispersión cromática**. Su magnitud depende de la longitud de onda. Una forma de evitar que se traslapen estos pulsos dispersos es aumentar la distancia entre ellos, pero esto se puede hacer sólo si se reduce la tasa de transmisión. Por fortuna se descubrió que si se da a los pulsos una forma especial relacionada con el recíproco del coseno hiperbólico, se cancelan casi todos los efectos de la dispersión y es

posible enviar pulsos a miles de kilómetros sin una distorsión apreciable de la forma. Estos pulsos se llaman solitones. Se está realizando una cantidad considerable de investigaciones para sacar los solitones del laboratorio y llevarlos al campo.

Cables de fibras

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

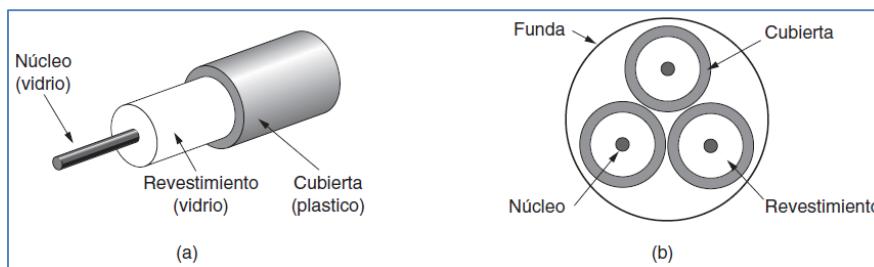


Ilustración 8 - (a) Vista lateral de una sola fibra. (b) Vista de extremo de una envoltura con tres fibras

El núcleo está rodeado de un revestimiento de vidrio con un índice de refracción más bajo que el del núcleo, con el fin de mantener toda la luz en el núcleo. Después viene una cubierta delgada de plástico para proteger el revestimiento. Por lo general las fibras se agrupan en haces, protegidas por una funda exterior. La Ilustración 8(b) muestra una funda con tres fibras.

Por lo general, las fundas de fibras terrestres se colocan un metro debajo de la superficie, en donde en ocasiones están sujetas a los ataques de retroexcavadoras o topos. Cerca de la costa, las fundas de fibras transoceánicas se entierran en zanjas mediante una especie de arado marino. En aguas profundas, simplemente se colocan en el fondo, donde pueden ser enganchadas por pesqueros.

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

Segunda, se pueden empalmar en forma mecánica. Los empalmes mecánicos simplemente acomodan los dos extremos cortados con cuidado, uno junto a otro en una manga especial y los sujetan en su lugar. Para mejorar la alineación se puede pasar luz a través de la unión para después realizar pequeños ajustes de modo que se maximice la señal. El personal capacitado tarda cerca de cinco minutos en crear empalmes mecánicos y se produce una pérdida de luz de 10%.

Tercera, se pueden fusionar (fundir) dos piezas de fibra para formar una conexión sólida. Un empalme por fusión es casi tan bueno como una sola fibra, pero incluso en este caso se produce una pequeña cantidad de atenuación.

En los tres tipos de empalmes se pueden producir reflejos en el punto del empalme; además la energía reflejada puede interferir con la señal.

Por lo general se utilizan dos tipos de fuentes de luz para producir las señales: LED (Diodos Emisores de Luz, del inglés *Light Emitting Diodes*) y láseres semiconductores. Estas fuentes de luz tienen distintas propiedades, como se muestra en la Ilustración 9. Se pueden optimizar en cuanto a la longitud de onda, para lo cual se insertan interferómetros *Fabry-Perot* o *Mach-Zehnder* entre la fuente y la fibra. Los interferómetros *Fabry-Perot* son simples cavidades resonantes que consisten en dos espejos paralelos. La luz incide en los espejos en forma perpendicular. La longitud de la cavidad separa las longitudes de onda que caben dentro de un número entero de veces. Los interferómetros *Mach-Zehnder* separan la luz en dos haces, los cuales viajan distancias ligeramente distintas. Se vuelven a combinar en el extremo y están en fase sólo para ciertas longitudes de onda.

Característica	LED	Láser semiconductor
Tasa de datos	Baja	Alta
Tipo de fibra	Multimodo	Multimodo o monomodo
Distancia	Corta	Larga
Tiempo de vida	Vida larga	Vida corta
Sensibilidad a la temperatura	Poca	Considerable
Costo	Bajo	Elevado

Ilustración 9 - Comparación de los diodos semiconductores y los LED 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 de los fotodiodos, que convierten la señal óptica en eléctrica, limita la tasa de datos a cerca de 100 Gbps. El ruido térmico es otro inconveniente, por lo que un pulso de luz debe llevar suficiente potencia para detectarlo. Cuando los pulsos tienen la potencia suficiente, la tasa de error se puede reducir de manera considerable.

Comparación entre la fibra óptica y el alambre de cobre

Es ilustrativo 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 sería indispensable en las redes de alto rendimiento. Debido a la baja atenuación, sólo se necesitan repetidores aproximadamente cada 50 km en líneas extensas, mientras que el cobre requiere repetidores cada 5 km, lo cual implica un ahorro considerable en el costo. La fibra también tiene la ventaja de que no le afectan las sobrecargas de energía, la interferencia electromagnética ni los cortes en el suministro de energía. Tampoco le afectan las sustancias corrosivas en el aire, lo cual es importante en los ambientes industriales pesados.

Por extraño que parezca, a las compañías telefónicas les gusta la fibra por una razón distinta: es delgada y ligera. Muchos conductos de cables existentes están llenos por completo, de modo que no hay espacio para agregar más capacidad. Si se quita todo el cobre y se sustituye por fibra se vacían los conductos, además de que el cobre tiene un excelente valor de reventa para las refinerías de cobre, quienes lo ven como materia prima de alta calidad. Asimismo, la fibra es mucho más ligera que el cobre. Mil pares trenzados de 1 km de longitud pesan 8000 kg. Dos fibras tienen más capacidad y sólo pesan 100 kg, lo cual reduce la necesidad de costosos sistemas mecánicos de apoyo a los que se debe dar mantenimiento. Para las nuevas rutas, la fibra es la mejor opción debido a que su costo de instalación es mucho más bajo. Por último, las fibras no tienen fugas de luz y son difíciles de intervenir. Estas propiedades les confieren una excelente seguridad contra los potenciales espías.

Sin embargo, la fibra es una tecnología poco familiar, además se pueden dañar con facilidad si se les dobla demasiado. Como la transmisión óptica es unidireccional por naturaleza, para la comunicación en ambos sentidos se requieren ya sea dos fibras o dos bandas de frecuencia en una fibra. Por último, las interfaces de las fibras cuestan más que las interfaces eléctricas. Sin embargo, el futuro de todas las comunicaciones fijas de datos a distancias, de algo más que unos cuantos metros, en definitiva está en la fibra. Para un análisis detallado de todos los aspectos de la fibra óptica y sus redes, consulte a Hecht (2005).

Transmisión inalámbrica

Nuestra Era ha dado origen a los adictos a la información: personas que necesitan estar todo el tiempo en línea. Para estos usuarios móviles no son de utilidad el par trenzado, el cable coaxial ni la fibra óptica. Necesitan obtener datos para sus computadoras laptop, notebook, de bolsillo, de mano o de reloj de pulsera sin tener que estar atados a la infraestructura de comunicación terrestre. Para estos usuarios, la comunicación inalámbrica es la respuesta.

En las siguientes secciones analizaremos la comunicación inalámbrica en general, la cual tiene muchas otras aplicaciones importantes además de proveer conectividad a los usuarios que desean navegar en la Web desde la playa. La tecnología inalámbrica ofrece ventajas incluso para dispositivos fijos en ciertos casos. Por ejemplo,

si es difícil tender una fibra hasta un edificio debido al terreno (montañas, junglas, pantanos, etc.), tal vez sea mejor usar tecnología inalámbrica.

El espectro electromagnético

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

Al conectar una antena del tamaño apropiado a un circuito eléctrico, las ondas electromagnéticas se pueden difundir de manera eficiente y un receptor las puede captar a cierta distancia. Toda la comunicación inalámbrica se basa en este principio.

En el vacío, todas las ondas electromagnéticas viajan a la misma velocidad sin importar cuál sea su frecuencia. Esta velocidad se conoce como **velocidad de la luz**, c , y es de aproximadamente 3×10^8 m/seg. En el cobre o la fibra, 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 máximo límite de velocidad. Ningún objeto o señal puede llegar a ser más rápido que la luz.

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

$$\lambda f = c$$

Dado que c es una constante, si conocemos el valor f podemos encontrar λ y viceversa. Como regla práctica, cuando λ se da en metros y f en MHz, $\lambda f \approx 300$. Por ejemplo, las ondas de 100 MHz tienen una longitud aproximada de 3 metros, las ondas de 1000 MHz tienen una longitud de 0.3 metros y las ondas de 0.1 metros tienen una frecuencia de 3000 MHz.

En la Ilustración 10 se muestra el **espectro electromagnético**. Las porciones de radio, microondas, infrarrojo y luz visible del espectro se pueden utilizar para transmitir información mediante la modulación de 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 de 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 Ilustración 10 son los nombres oficiales de la ITU (Unión Internacional de Telecomunicaciones) y se basan en las longitudes de onda, por lo que la banda LF va de 1 a 10 km (aproximadamente de 30 a 300 kHz). Los términos LF, MF y HF se refieren a las frecuencias baja, media y alta, respectivamente. Está claro que al asignar los nombres nadie esperaba rebasar los 10 MHz, por lo que las bandas más altas se denominaron después como bandas de muy, ultra, súper, extrema y tremadamente alta frecuencia.

Sabemos de la ecuación desarrollada por Shannon que la cantidad de información que puede transportar una señal como una onda electromagnética depende de la potencia recibida y es proporcional a su ancho de banda. De la Ilustración 10 debe quedar ahora claro por qué a las personas que trabajan en redes les gusta tanto la fibra óptica. Hay muchos GHz de ancho de banda disponibles que se pueden aprovechar para la transmisión de datos en la banda de microondas, e incluso más en la fibra debido a que está más a la derecha en nuestra escala logarítmica.

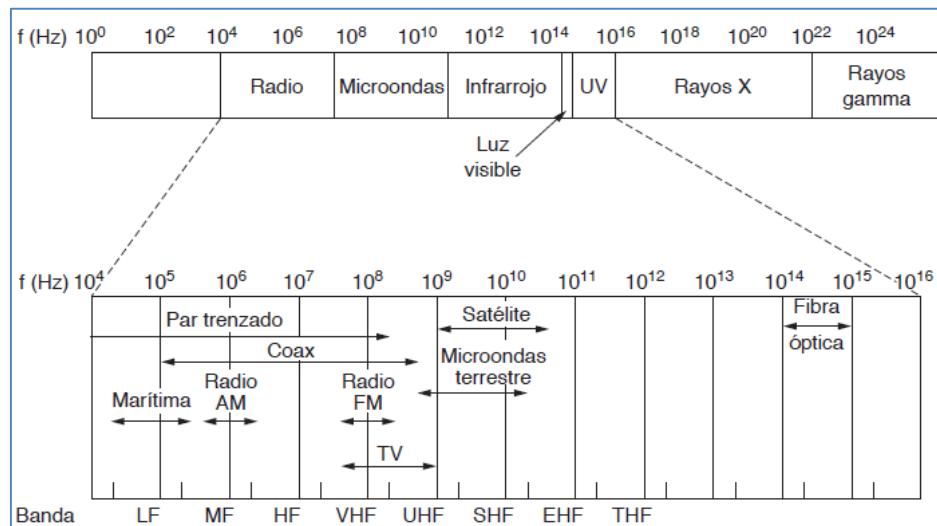


Ilustración 10 -El espectro electromagnético y sus usos para comunicaciones.

La mayoría de las transmisiones utilizan una banda de frecuencia relativamente estrecha (por decir, $\Delta f/f \ll 1$). Concentran sus señales en esta banda estrecha para usar el espectro con eficiencia y obtienen tasas de datos razonables al transmitir con suficiente potencia. Pero en algunos casos se utiliza una banda ancha, con tres variaciones. En el **espectro expandido con salto de frecuencia**, el transmisor salta de frecuencia en frecuencia cientos de veces por segundo. Es popular en la comunicación militar, ya que hace a las transmisiones difíciles de detectar y casi imposibles de bloquear. También ofrece una buena resistencia al desvanecimiento multirayectoria y a la interferencia de banda estrecha, ya que el receptor no se detendrá en una frecuencia dañada el tiempo suficiente como para detener la comunicación. Esta robustez la hace útil para las partes atestadas del espectro, como las bandas ISM que veremos más adelante. Bluetooth y las versiones anteriores de 802.11 son ejemplos del uso comercial de esta técnica.

Hay una segunda forma de espectro disperso conocida como **espectro expandido de secuencia directa**, la cual utiliza una secuencia de códigos para dispersar los datos sobre una banda de frecuencia ancha. Su uso comercial es muy popular como una forma espectralmente eficiente de permitir que múltiples señales comparten la misma banda de frecuencia. Estas señales pueden recibir distintos códigos, un método conocido como CDMA (Acceso Múltiple por División de Código, del inglés *Code Division Multiple Access*) que veremos más adelante. En la Ilustración 11 se muestra este método en contraste con el salto de frecuencias. CDMA forma la base de las redes de telefonía móvil 3G y también se utiliza en sistemas GPS (Sistema de Posicionamiento Global, del inglés *Global Positioning System*). Incluso con códigos distintos, el espectro expandido de secuencia directa (al igual que el espectro expandido de salto de frecuencia) puede tolerar la interferencia de banda estrecha y el desvanecimiento multirayectoria, ya que sólo se pierde una fracción de la señal deseada. Se utiliza para desempeñar esta función en las redes LAN inalámbricas 802.11b anteriores. Si desea leer una historia fascinante y detallada sobre las comunicaciones de espectro disperso, consulte a Scholtz (1982).

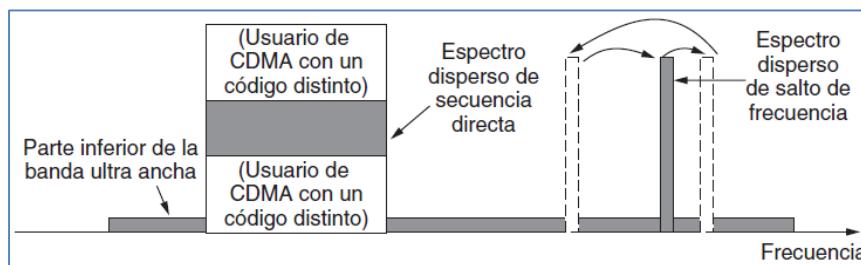


Ilustración 11 - Comunicaciones de espectro expandido y Banda Ultra-Ancha (UWB).

Un tercer método de comunicación con una banda más ancha es la comunicación **UWB (Banda Ultra-Ancha)**. La UWB envía una serie de pulsos rápidos, los cuales varían sus posiciones para comunicar la información. Las

transiciones rápidas conducen a una señal que se dispersa finamente sobre una banda de frecuencia muy amplia. UWB se define como señales que tienen un ancho de banda de por lo menos 500 MHz o de al menos 20% de la frecuencia central de su banda de frecuencia. En la Ilustración 11 también se muestra la UWB. Con todo este ancho de banda, la UWB tiene el potencial para comunicarse a tasas altas. Como se dispersa a través de una banda amplia de frecuencias, puede tolerar una cantidad considerable de interferencia relativamente fuerte que provenga de otras señales de banda estrecha. Otra cuestión de igual importancia es que como la UWB tiene muy poca energía en cualquier frecuencia dada cuando se utiliza para la transmisión de corto rango, no provoca una interferencia dañina a todas esas otras señales de radio de banda estrecha. Se dice que subyace debajo de las otras señales. Gracias a esta pacífica coexistencia se utiliza en redes PAN inalámbricas que operan hasta a 1 Gbps, aunque el éxito comercial ha sido mezclado. También se puede usar para tomar imágenes a través de objetos sólidos (suelo, paredes y cuerpos) o como parte de los sistemas de ubicación precisos.

Ahora veremos cómo se utilizan las diversas partes del espectro electromagnético de la Ilustración 10, empezando por la radio. Supongamos que todas las transmisiones utilizan una banda estrecha de frecuencia, a menos que se indique lo contrario.

Radiotransmisión

Las ondas de radio frecuencia (RF) son fáciles de generar, pueden recorrer distancias largas y penetrar edificios con facilidad, de modo que son muy utilizados en la comunicación, tanto en interiores como en exteriores. Las ondas de radio también son omnidireccionales, lo cual significa que viajan en todas direcciones desde la fuente, por lo que el transmisor y el receptor no tienen que estar alineados físicamente.

Algunas veces la radio omnidireccional es buena, pero otras no lo es tanto. En la década de 1970, General Motors decidió equipar a todos sus Cadillacs nuevos con frenos antibloqueo controlados por computadora. Cuando el conductor pisaba el pedal del freno, la computadora accionaba los frenos para activarlos y desactivarlos en vez de bloquearlos con firmeza. Un buen día, un patrullero de las carreteras de Ohio encendió su nuevo radio móvil para llamar a la estación de policía, cuando de repente el Cadillac que iba junto a él comenzó a comportarse como un potro salvaje. Cuando el oficial detuvo el auto, el conductor alegó que no había hecho nada y que el auto se había vuelto loco. Con el tiempo comenzó a surgir un patrón: algunas veces los Cadillacs se volvían locos, 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 que emprendieron una búsqueda extensa descubrieron que el cableado de los Cadillacs formaba una excelente antena para la frecuencia utilizada por 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, las ondas de radio cruzan bien los obstáculos, pero la potencia se reduce drásticamente a medida que se aleja de la fuente (por lo menos tan rápido como $1/r^2$ en el aire, donde r es la distancia hasta la fuente). A esta atenuación se le conoce como pérdida de trayectoria. A frecuencias altas, las ondas de radio tienden a viajar en línea recta y rebotan en los obstáculos. La pérdida de trayectoria reduce aún más la potencia, aunque la señal recibida también puede depender en gran parte de las reflexiones. Las ondas de radio de alta frecuencia también son absorbidas por la lluvia y otros obstáculos en mayor grado que las de baja frecuencia. En todas las frecuencias las ondas de radio están sujetas a interferencia de los motores y demás equipos eléctricos.

Es interesante comparar la atenuación de las ondas de radio con la de las señales en los medios guiados. Con la fibra, el cable coaxial y el par trenzado, la señal se reduce en la misma fracción por distancia de unidad, por ejemplo 20 dB por cada 100 m para el par trenzado. Con la radio, la señal se reduce en la misma fracción a medida que se duplica la distancia, por ejemplo 6 dB por cada vez que se duplique la distancia en el espacio libre. Este comportamiento indica que las ondas de radio pueden recorrer grandes distancias y, en consecuencia, la interferencia entre usuarios es un problema. Por esta razón, es común que los gobiernos regulen estrictamente el uso de los radiotransmisores, con algunas excepciones notables que veremos más adelante.

En las bandas VLF, LF y MF las ondas de radio siguen la curvatura de la Tierra, como se muestra en la Ilustración 12(a). Estas ondas se pueden detectar quizás a 1000 km en las frecuencias más bajas, y a menos distancia en las frecuencias más altas. Las ondas de radio en estas bandas pasan por los edificios fácilmente, razón por la cual los radios portátiles funcionan en interiores. El principal problema al usar estas bandas para la comunicación de datos es su bajo ancho de banda.

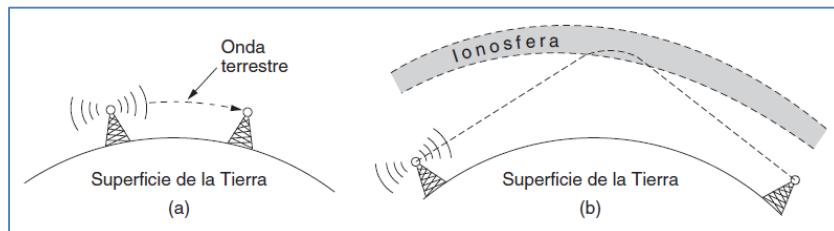


Ilustración 12 - (a) En las bandas VLF, LF y MF, las ondas de radio siguen la curvatura de la Tierra. (b) En la banda HF, rebotan en la ionosfera.

En las bandas HF y VHF, las ondas terrestres tienden a ser absorbidas por la Tierra. Sin embargo, las ondas que llegan a la ionosfera (una capa de partículas cargadas que rodean la Tierra a una altura de 100 a 500 km) se refractan y se envían de vuelta a nuestro planeta, como se muestra en la Ilustración 12(b). Bajo ciertas condiciones atmosféricas, las señales pueden rebotar varias veces. Los operadores de las bandas de radio aficionados utilizan estas bandas para conversar a larga distancia. El ejército también se comunica en las bandas HF y VHF.

Transmisión por microondas

Por encima de los 100 MHz, las ondas viajan en línea recta y en consecuencia, se pueden enfocar en un haz estrecho. Al concentrar toda la energía en un pequeño haz por medio de una antena parabólica (como el tan conocido plato de TV por satélite) se obtiene una relación señal-ruido mucho más alta, pero las antenas transmisora y receptora deben estar alineadas entre sí con precisión. Además, esta direccionalidad permite que varios transmisores alineados en fila se comuniquen con varios receptores sin interferencia, siempre y cuando se sigan ciertas reglas de espacio mínimo. Antes de la fibra óptica, estas microondas formaron durante décadas el corazón del sistema de transmisión telefónica de larga distancia. De hecho, la empresa MCI (uno de los primeros competidores de AT&T después de su liberación) construyó todo su sistema a partir de comunicaciones por microondas que iban de torre en torre ubicadas a decenas de kilómetros una de la otra. Incluso el nombre de la empresa reflejaba esta cuestión (MCI representa a *Microwave Communications, Inc.*). Después MCI cambió a la fibra óptica y por medio de una extensa serie de fusiones corporativas y bancarrotas en la reestructuración de las telecomunicaciones, se volvió parte de Verizon.

Puesto que las microondas viajan en línea recta, si las torres están demasiado separadas, la Tierra se interpondrá en el camino. Por ende se necesitan repetidores periódicos. Entre más altas sean las torres, más separadas pueden estar. La distancia entre repetidores se eleva en forma muy aproximada a la raíz cuadrada de la altura de la torre. Si tenemos torres de 100 metros 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 pueden atravesar 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. Estas ondas retrasadas pueden llegar desfasadas con la onda directa y cancelar así la señal. A este efecto se le llama **desvanecimiento por multirayectorias** y representa a menudo un problema grave que depende del clima y de la frecuencia. Algunos operadores mantienen el 10% de sus canales inactivos como repuestos para activarlos cuando el desvanecimiento por multirayectorias cancela en forma temporal alguna banda de frecuencia.

La creciente demanda de espectro obliga a los operadores a usar frecuencias aún más altas. Ahora las bandas de hasta 10 GHz son de uso rutinario, pero con las de casi 4 GHz se presenta un nuevo problema: la **absorción por el agua**. Estas ondas tienen sólo unos centímetros de longitud y la lluvia las absorbe. Al igual que con el

desvanecimiento por multirayectorias, la única solución es interrumpir los enlaces afectados por la lluvia y encaminar las señales a su alrededor.

En resumen, la comunicación por microondas se utiliza tanto para la comunicación telefónica de larga distancia, los teléfonos móviles, la distribución de la televisión y otros usos, que ha provocado una escasez de espectro. Esta tecnología tiene varias ventajas clave respecto a la fibra. La principal es que no se necesita derecho de paso para tender los cables. Con sólo comprar un pequeño terreno cada 50 km y construir en él una torre de microondas, se puede pasar por alto el sistema telefónico en su totalidad.

Las microondas son también relativamente económicas. Puede ser más barato erigir dos torres sencillas (que pueden ser tan sólo unos postes grandes con cuatro cables de retén) y poner antenas en cada una de ellas, 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 telefónica, en especial si ésta no ha recuperado por completo la inversión por el cobre que quitó al instalar la fibra.

Las políticas del espectro electromagnético

Para evitar el caos total, existen acuerdos nacionales e internacionales en cuanto a quién puede usar ciertas frecuencias. Como todos quieren una tasa más alta de transferencia de datos, todos quieren más espectro. Los gobiernos nacionales asignan el espectro para la radio AM y FM, la televisión y los teléfonos móviles, así como para las compañías telefónicas, la policía, las comunicaciones marítimas, la navegación, el ejército, el gobierno y muchos otros usuarios competidores. A nivel mundial, una agencia de la ITU-R (WRC) trata de coordinar esta asignación de modo que se puedan fabricar dispositivos que funcionen en varios países. Sin embargo, los países no están obligados a seguir las recomendaciones de la ITU-R.

Incluso cuando se haya asignado una parte del espectro para cierto uso, como los teléfonos móviles, se debe determinar qué empresa portadora puede utilizar qué frecuencias. En el pasado se utilizaban tres algoritmos. El más viejo se conoce como **concurso de méritos** (*beauty contest*); en este algoritmo cada empresa portadora tiene que explicar por qué su propósito es más útil para el interés público. Después, los funcionarios de gobierno deciden cuál de todas esas historias los convence más. El hecho de que un funcionario de gobierno pueda otorgar una propiedad con valor de miles de millones de dólares a su compañía favorita conduce con frecuencia al soborno, la corrupción, el nepotismo y cosas peores.

Esta observación condujo al algoritmo 2: llevar a cabo un **sorteo** entre las compañías interesadas. El problema con esta idea es que pueden entrar al sorteo empresas que no tengan interés en utilizar el espectro. Por decir, si un restaurante de comida rápida o una cadena de tiendas de zapatos ganan, puede revender el espectro a una portadora para obtener una enorme ganancia sin ningún riesgo.

Este proceso ha sido criticado con severidad por muchos, lo cual condujo al algoritmo 3: **subastar** el ancho de banda al mejor postor. Cuando el gobierno británico subastó las frecuencias necesarias para los sistemas móviles de tercera generación en el año 2000, esperaba obtener cerca de u\$s4 mil millones. En realidad recibió cerca de u\$s40 mil millones debido a que las empresas portadoras cayeron en la desesperación, muertas de miedo de dejar pasar la oportunidad. Este suceso despertó la avaricia de los gobiernos vecinos y los inspiró a realizar sus propias subastas. Esto funcionó, pero a la vez algunas de las empresas portadoras quedaron con deudas enormes que las llevaron cerca de la bancarrota. Aun en los mejores casos, se requerirán muchos años para recuperar la inversión en la licencia.

Una metodología completamente distinta a la asignación de frecuencias es la de no asignarlas en lo absoluto. En vez de ello hay que dejar que todos transmitan a voluntad, pero que regulen la potencia utilizada de modo que las estaciones tengan un rango tan corto que no interfieran entre sí. En consecuencia, la mayoría de los gobiernos han separado ciertas bandas de frecuencia, llamadas bandas **ISM** (Industriales, Científicas, Médicas, del inglés *Industrial, Scientific, Medical*), para un uso sin necesidad de licencia. Los dispositivos para abrir puertas de cocheras, los teléfonos inalámbricos, los juguetes de radiocontrol, los ratones inalámbricos y muchos otros electrodomésticos inalámbricos utilizan las bandas ISM. Para minimizar la interferencia entre estos dispositivos no coordinados, se exige que todos los dispositivos en las bandas ISM limiten su potencia de transmisión (por ejemplo, a 1 watt) y utilicen técnicas para dispersar sus señales a través de un rango de frecuencias. La ubicación de estas bandas varía de un país a otro.

Transmisión infrarroja

Las ondas infrarrojas no guiadas se usan mucho para la comunicación de corto alcance. El control remoto de los televisores y otros equipos utilizan comunicación infrarroja. Son relativamente direccionales, económicos y fáciles de construir, pero tienen un gran inconveniente: no atraviesan objetos sólidos (pruebe pararse entre el control remoto y su televisión, y vea si aún 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 o edificios adyacentes; no podrá controlar la televisión de su vecino con su control remoto. Además, la seguridad de los sistemas infrarrojos contra el espionaje es mejor que la de los sistemas de radio, precisamente por esta razón. Por ende, no se necesita licencia gubernamental para operar un sistema infrarrojo, en contraste con los sistemas de radio, que deben contar con licencia excepto las bandas ISM.

Transmisión por ondas de luz

La señalización óptica sin guías, también conocida como **óptica de espacio libre**, se ha utilizado durante siglos. Paul Revere utilizó señalización óptica binaria desde la vieja Iglesia del Norte justo antes de su famoso viaje. Una aplicación más moderna es conectar las redes LAN de dos edificios mediante láser montados en sus azoteas. La señalización óptica mediante láser es de naturaleza unidireccional, por lo que cada extremo necesita su propio láser y su propio fotodetector. Este esquema ofrece un ancho de banda muy alto a un costo muy bajo, además de ser relativamente seguro debido a que es difícil intervenir un haz tan estrecho. También es relativamente fácil de instalar y, a diferencia de las microondas, no requiere una licencia.

La ventaja del láser, un haz muy estrecho, es también su debilidad en este caso. Para apuntar un rayo láser de 1 mm de anchura a un blanco del tamaño de la punta de un alfiler a 500 metros de distancia, se requiere una precisión enorme. Por lo general se añaden lentes al sistema para desenfocar ligeramente el rayo. Para dificultar aún más las cosas, los cambios en el viento y la temperatura pueden distorsionar el rayo, además de que los rayos láser no pueden penetrar la lluvia o la niebla densa, aunque por lo general funcionan bien en días soleados.

La comunicación óptica sin guía puede parecer una tecnología de redes exótica en la actualidad, pero pronto puede llegar a ser más frecuente. Estamos rodeados por cámaras (que detectan la luz) y pantallas (que emiten luz mediante el uso de LED y otras tecnologías). La comunicación de datos se puede disponer en capas encima de estas pantallas si se codifica la información en el patrón que hace que los LED se enciendan y apaguen, y que está por debajo del umbral de la percepción humana. Esta forma de comunicarse con luz visible es segura por naturaleza, además de que crea una red de baja velocidad en los alrededores inmediatos de la pantalla. Esto podría permitir todo tipo de escenarios computacionales elegantes y ubicuos. Las luces destellantes en los vehículos de emergencia podrían alertar a los semáforos cercanos y a los vehículos para ayudar a dejar libre el camino. Los anuncios informativos podrían difundir mapas. Incluso las luces de las fiestas podrían difundir canciones sincronizadas con su pantalla.

La tecnología LiFi es un ejemplo moderno de este tipo de comunicaciones. Este es un tipo de conexión a Internet que usa tecnología que se caracteriza por transmitir información a través de la luz led que podría llegar a los 10 Gbps de velocidad. Esto porque la luz se enciende y apaga hasta 10 mil millones de veces por segundo, lo que hace que se transforme la información en forma binaria (0 y 1); se aprovecha esta característica para poder enviar la información a través de la onda de la luz.

Satélites de comunicación

En la década de 1950 y a principios de la década de 1960, las personas trataban de establecer sistemas de comunicación mediante el rebote de señales sobre globos meteorológicos. Por desgracia, las señales que se recibían eran demasiado débiles como para darles un uso práctico. Después, la marina de Estados Unidos observó un tipo de globo meteorológico permanente en el cielo (la Luna), de modo que construyó un sistema operacional para la comunicación de barcos con la costa mediante señales que rebocaban de la Luna.

El avance en el campo de la comunicación celestial tuvo que esperar hasta que se lanzó el primer satélite de comunicaciones. La diferencia clave entre un satélite artificial y uno real es que el primero puede amplificar las señales antes de enviarlas de regreso, convirtiendo una extraña curiosidad en un poderoso sistema de comunicaciones.

Los satélites de comunicaciones tienen ciertas propiedades interesantes que los hacen atractivos para muchas aplicaciones. En su forma más simple, podemos considerar un satélite de comunicaciones como un enorme repetidor de microondas en el cielo que contiene varios **transpondedores**, cada uno de los cuales escucha en cierta porción del espectro, amplifica la señal entrante y después la retransmite en otra frecuencia para evitar interferencia con la señal entrante. Este modo de operación se llama **tubo doblado (u-bend)**. Se puede agregar un procesamiento digital para manipular o redirigir por separado los flujos de datos en toda la banda, o el satélite puede recibir información digital y retransmitirla. Esta forma de regeneración de señales mejora el desempeño si se le compara con un u-bend, ya que el satélite no amplifica el ruido en la señal que va hacia arriba. Los haces que descenden pueden ser amplios y cubrir una fracción considerable de la superficie de la Tierra, o pueden ser estrechos y cubrir un área de unos cuantos cientos de kilómetros de diámetro.

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, mayor será el periodo. Cerca de la superficie de la Tierra, el periodo es de aproximadamente 90 minutos. En consecuencia, los satélites con órbitas bajas salen del rango de visión muy rápido, de modo que muchos de ellos deben proveer una cobertura continua y las antenas terrestres deben rastrearlos. A una altitud aproximada de 35800 km, el periodo es de 24 horas. A una altitud de 384000 km el periodo es de cerca de un mes, como puede atestiguar cualquiera que haya observado la Luna con regularidad.

El periodo de un satélite es importante, pero no es la única razón para determinar en dónde colocarlo. Otra cuestión es la presencia de los cinturones de Van Allen: capas de partículas altamente cargadas, atrapadas por el campo magnético de la Tierra. Cualquier satélite que volara dentro de los cinturones quedaría destruido casi al instante debido a las partículas. Estos factores condujeron a tres regiones en las que se pueden colocar los satélites de forma segura. En la Ilustración 13 se muestran estas regiones con algunas de sus propiedades.

Satélites geoestacionarios

En 1945, el escritor de ciencia ficción Arthur C. Clarke calculó que un satélite con una altitud de 35800 km en una órbita ecuatorial circular parecería estar inmóvil en el cielo, por lo que no habría la necesidad de rastrearlo (Clarke, 1945). Pasó a describir un sistema completo de comunicaciones que utilizaba estos satélites geoestacionarios (tripulados), incluyendo las órbitas, los paneles solares, las frecuencias de radio y los procedimientos de lanzamiento. Por desgracia concluyó que los satélites no eran prácticos debido a la imposibilidad de poner en órbita amplificadores de tubos de vacío frágiles y que consumían una gran cantidad de energía, por lo que nunca profundizó sobre esta idea, aunque escribió algunas historias de ciencia ficción sobre el tema.

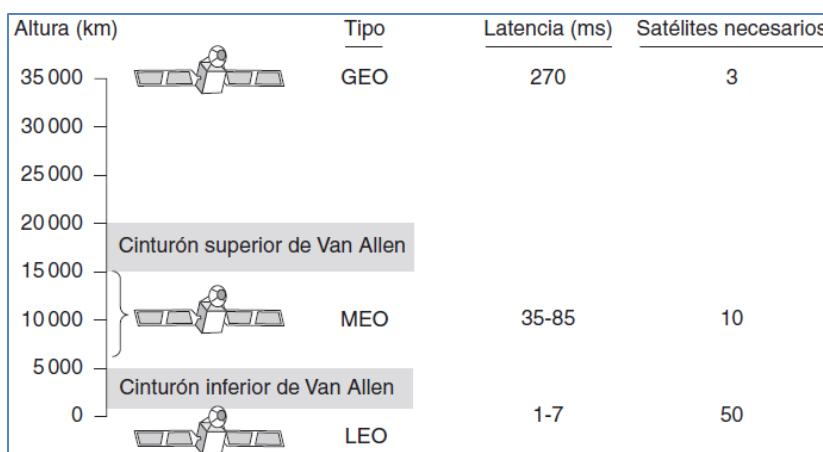


Ilustración 13 - Satélites de comunicaciones y algunas de sus propiedades, incluyendo la altitud sobre la Tierra, el tiempo de retardo de viaje redondo y la cantidad de satélites necesarios para una cobertura global.

La invención del transistor cambió todo eso; el primer satélite de comunicación artificial llamado *Telstar* se lanzó en julio de 1962. Desde entonces, los satélites de comunicación se convirtieron en un negocio multimillonario y el único aspecto del espacio exterior que se ha vuelto muy rentable. Estos satélites que vuelan a grandes alturas se conocen comúnmente como satélites **GEO** (Órbita Terrestre Geoestacionaria, del inglés *Geostationary Earth Orbit*).

Con la tecnología actual, es poco sensato tener satélites geoestacionarios separados a menos de 2 grados en el plano ecuatorial dado que se quieren evitar interferencias. Con una separación de 2 grados sólo puede haber $360/2=180$ de estos satélites en el cielo a la vez. Sin embargo, cada transpondedor puede usar múltiples frecuencias y polarizaciones para incrementar el ancho de banda disponible.

Para evitar un caos total en el cielo, la ITU se encarga de la asignación de espacio orbital. Este proceso es altamente político. Sin embargo, otros países afirman que los derechos de propiedad nacional no se extienden hasta la Luna y que ningún país tiene el derecho legal sobre los espacios orbitales encima de su territorio. Para hacer más grande la pelea, la telecomunicación comercial no es la única aplicación. Las difusoras de televisión, los gobiernos y el ejército también desean una parte del espacio orbital.

Los satélites modernos pueden ser muy grandes y pesar más de 5000 kg, además de que consumen varios kilowatts de energía eléctrica producida por los paneles solares. Los efectos de la gravedad solar, lunar y planetaria tienden a alejarlos de sus espacios orbitales y orientaciones asignadas, un efecto que se contrarresta mediante motores de cohete integrados. Esta actividad de ajuste se conoce como **control de la posición orbital** (*station keeping*). Sin embargo, cuando se agota el combustible de los motores (por lo general después de casi 10 años), el satélite queda a la deriva y cae sin que se pueda hacer nada, de modo que debe ser desactivado. En un momento dado la órbita se deteriora, el satélite vuelve a entrar en la atmósfera y se quema (o en muy raras ocasiones, se estrella en la Tierra).

Los espacios orbitales no son el único motivo de discordia. Las frecuencias también son otro problema debido a que las transmisiones de los enlaces descendentes interfieren con los usuarios existentes de microondas. En consecuencia, la ITU ha asignado bandas de frecuencia específicas a los usuarios de satélites. Las principales se muestran en la Ilustración 14. La banda C fue la primera en ser designada para el tráfico comercial por satélites. Hay dos rangos de frecuencia asignados a esta banda, el inferior para el tráfico de enlace descendente (proveniente del satélite) y el superior para el tráfico del enlace ascendente (que va al satélite). Para permitir que el tráfico viaje en ambos sentidos al mismo tiempo se requieren dos canales. Estos canales ya están sobresaturados debido a que también son utilizados por las portadoras comunes para los enlaces terrestres de microondas. Las bandas L y S se agregaron con base en un acuerdo internacional en el año 2000. Sin embargo, son estrechas y también están 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.

Ilustración 14 - Las principales bandas de satélites.

La siguiente banda más ancha disponible para las portadoras de telecomunicaciones comerciales es la banda Ku (K inferior, del inglés *K under*). Esta banda (aún) no está saturada; a sus frecuencias más altas los satélites pueden tener una separación mínima de 1 grado. Sin embargo, existe otro problema: la lluvia. El agua absorbe bien estas microondas cortas. Por fortuna, las tormentas fuertes por lo general son localizables, de modo que para resolver el problema se pueden usar varias estaciones terrestres separadas a grandes distancias en vez de usar sólo una, pero a costa de requerir antenas, cables y componentes electrónicos adicionales para permitir una conmutación rápida entre las estaciones. También se asignó ancho de banda en la banda Ka (K superior, del inglés *K above*) para el tráfico comercial de satélites, pero el equipo necesario para utilizarla es costoso. Además de estas bandas comerciales, también existen muchas bandas gubernamentales y militares.

Un satélite moderno tiene cerca de 40 transpondedores, cada uno con un ancho de banda de 36 MHz. Por lo general cada transpondedor opera como un tubo doblado, pero los satélites recientes cuentan con capacidad de procesamiento integrada, lo cual les permite una operación más sofisticada. En los primeros satélites, la división de los transpondedores en canales era estática: el ancho de banda simplemente se dividía en bandas fijas de frecuencia. Hoy en día el haz de cada transpondedor se divide en ranuras de tiempo, en donde varios usuarios toman turnos. Más adelante estudiaremos 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 aproximadamente 1/3 de la superficie de la Tierra, a lo cual se le conoce como **huella o pisada**. Con la enorme reducción en el precio, tamaño y requerimientos de energía de los componentes microelectrónicos, se ha hecho posible una estrategia de difusión mucho más sofisticada. Cada satélite está equipado con múltiples antenas y múltiples transpondedores. Cada haz descendente se puede enfocar en una pequeña área geográfica, de manera que se pueden llevar a cabo varias transmisiones ascendentes y descendentes simultáneamente. Por lo general, estos denominados haces puntuales tienen una forma elíptica y pueden ser tan pequeños como de algunos cientos de kilómetros de diámetro.

Los satélites de comunicación tienen varias propiedades que son 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 300000 km/seg), la larga distancia de viaje redondo introduce un retardo considerable para los satélites GEO. Dependiendo de la distancia entre el usuario y la estación terrestre, y de la elevación del satélite sobre el horizonte, el tiempo de tránsito de un extremo a otro está entre 250 y 300 ms. Un valor común es 270 ms.

Para fines de comparación, los enlaces terrestres de microondas tienen un retardo de propagación de aproximadamente 3 μ s/km, y los enlaces de cable coaxial o fibra óptica tienen un retardo de casi 5 ms/km. Los últimos son más lentos que los primeros debido a que las señales electromagnéticas viajan con más rapidez en el aire que en los materiales sólidos.

Otra propiedad importante de los satélites es que son medios de difusión por naturaleza. Cuesta lo mismo enviar un mensaje a miles de estaciones dentro de la huella de un transpondedor que enviarlo a una sola. Para algunas aplicaciones esta propiedad es muy útil. Por ejemplo, podríamos imaginar un satélite difundiendo páginas web populares a las cachés de una gran cantidad de computadoras dispersas sobre un área amplia. Aun cuando podemos simular la difusión mediante líneas de punto a punto, es probable que la difusión vía satélite sea más económica. Por otro lado, desde el punto de vista de la privacidad, los satélites son un completo desastre: todos pueden escucharlo todo. Es esencial el cifrado cuando se requiere seguridad.

Los satélites también tienen la propiedad de que el costo de transmitir un mensaje es independiente de la distancia a recorrer. Es lo mismo dar servicio a una llamada de un extremo a otro del océano que una llamada de un extremo a otro de la calle. Los satélites también tienen excelentes tasas de error y pueden implementarse casi al instante, una buena ventaja para las comunicaciones militares y de respuesta a los desastres.

Satélites de Órbita Terrestre Media (MEO)

En altitudes mucho más bajas entre los dos cinturones de Van Allen se encuentran los satélites **MEO** (Orbita Terrestre Media, del inglés *Medium-Earth Orbit*). Vistos desde la Tierra, se desvían lentamente en longitud y tardan cerca de seis horas en dar vuelta a la Tierra. Por ende, hay que rastrearlos a medida que se mueven por el cielo. Como tienen menor altura que los satélites GEO, producen una huella más pequeña en la Tierra y requieren transmisores menos poderosos para comunicarse. En la actualidad se utilizan para sistemas de navegación en vez de las telecomunicaciones, por lo que no daremos más detalles sobre ellos. La constelación de alrededor de 30 satélites **GPS** (Sistema de Posicionamiento Global, del inglés *Global Positioning System*) que giran a una distancia aproximada de 20200 km son ejemplos de satélites MEO.

Satélites de Órbita Terrestre Baja (LEO)

Los satélites **LEO** (Órbita Terrestre Baja, del inglés *Low-Earth Orbit*) se encuentran a una altitud todavía más baja. Debido a su rápido movimiento, se necesita un gran número de ellos para un sistema completo. Por otro

lado, como los satélites están tan cerca de la Tierra, las estaciones terrestres no necesitan mucha potencia y el retardo de viaje redondo es de sólo unos cuantos milisegundos. El costo de lanzamiento es más económico.

Comparación de los satélites y la fibra óptica

Una comparación entre la comunicación vía satélite y la comunicación terrestre es algo instructivo. Apenas hace 25 años podríamos argumentar que el futuro de las comunicaciones recaería en los satélites de comunicación. Después de todo, el sistema telefónico había cambiado muy poco en los 100 años anteriores y no mostraba signos de cambio en los siguientes 100. Este movimiento glacial era provocado en gran parte por el entorno regulatorio en el que se exigía a las compañías telefónicas proveer un buen servicio de voz a precios razonables (lo cual cumplían), y a cambio obtenían ganancias garantizadas sobre su inversión. Para las personas que necesitaban transmitir datos, había módems de 1200 bps disponibles. Eso era prácticamente todo lo que se tenía.

La introducción de la competencia en 1984 en Estados Unidos y poco después en Europa cambió todo eso de manera radical. Las compañías telefónicas empezaron a reemplazar sus redes de larga distancia con fibra óptica e introdujeron servicios con ancho de banda alto, como **ADSL** (Línea asimétrica de suscriptor digital, del inglés *Asymmetric Digital Subscriber Line*). También dejaron su antigua práctica de cobrar precios estratosféricos a los usuarios de larga distancia para subsidiar el servicio local. De repente, las conexiones terrestres de fibra óptica se perfilaban como el ganador.

Sin embargo, los satélites de comunicación tienen algunos mercados de nichos importantes que la fibra óptica no maneja (y en ciertos casos porque no puede hacerlo). En primer lugar, cuando es imprescindible un despliegue rápido, los satélites ganan fácilmente. Una respuesta rápida es útil para los sistemas de comunicaciones militares en tiempos de guerra y para la respuesta al desastre en tiempos de paz.

Un segundo nicho es para la comunicación en lugares en donde la infraestructura terrestre está mal desarrollada. En la actualidad muchas personas desean comunicarse desde cualquier parte a donde vayan. Las redes de telefonía móvil cubren esas ubicaciones con buena densidad de población, pero no realizan un trabajo adecuado en otros lugares (por ejemplo, en el mar o en el desierto). Además, la instalación de la infraestructura terrestre puede ser costosa, dependiendo del terreno y de los derechos de paso necesarios.

En el tercer nicho la difusión es imprescindible. El mensaje que envía un satélite lo pueden recibir miles de estaciones terrestres a la vez. Por esta razón, los satélites se utilizan para distribuir gran parte de la programación de TV a las estaciones locales. Ahora hay un extenso mercado para las difusiones vía satélite de TV y radio digital directamente a los usuarios finales, que cuentan con receptores de satélite en sus hogares y autos. También se pueden difundir otros tipos de contenido.

En resumen, parece ser que las comunicaciones dominantes en el futuro serán a través de la fibra óptica terrestre combinada con la radio celular, pero para ciertos usos especializados son mejores los satélites. Sin embargo, hay una advertencia que se aplica a todo esto: la economía. Aunque la fibra óptica ofrece más ancho de banda, es probable que la comunicación terrestre y la comunicación vía satélite puedan competir de manera agresiva en cuanto al precio. Si los avances en la tecnología reducen de manera radical el costo de desplegar un satélite (por ejemplo, si algún vehículo espacial en el futuro puede lanzar docenas de satélites a la vez) o los satélites de órbita baja presentan avances considerables, no es seguro que la fibra óptica vaya a ganar en todos los mercados.

Modulación digital y multiplexación

Ahora que hemos estudiado las propiedades de los canales alámbricos e inalámbricos, nos enfocaremos en el problema de cómo enviar información digital. Los cables y los canales inalámbricos transportan señales analógicas, como el voltaje, la intensidad de la luz o del sonido que varían de forma continua. Para enviar información digital debemos idear señales analógicas que representen bits. Al proceso de realizar la conversión entre los bits y las señales que los representan se le conoce como **modulación digital**.

Empezaremos con esquemas que convierten directamente los bits en una señal. Estos esquemas resultan en una **transmisión en banda base**, en donde la señal ocupa frecuencias desde cero hasta un valor máximo que

depende de la tasa de señalización. Este tipo de transmisión es común para los cables. Después consideraremos esquemas que varían la amplitud, fase o frecuencia de una señal portadora para transmitir los bits. Estos esquemas resultan en una **transmisión pasa-banda**, en donde la señal ocupa una banda de frecuencias alrededor de la frecuencia de la señal portadora. Es común para los canales inalámbricos y ópticos, en donde las señales deben residir en una banda de frecuencia dada.

A menudo los canales se comparten entre varias señales. Después de todo, es mucho más conveniente utilizar un solo cable para transportar varias señales que instalar un cable para cada señal. A este tipo de compartición se le denomina **multiplexación** y se puede lograr de varias formas. Presentaremos los métodos para la multiplexación por división de tiempo, de frecuencia y de código.

Las técnicas de modulación y multiplexación que describiremos en esta sección son muy usados en los cables, la fibra óptica, los canales inalámbricos terrestres y los canales de satélite.

Transmisión en banda base

La forma más simple de modulación digital es utilizar un voltaje positivo para representar un 1 y un voltaje negativo para representar un 0. Para una fibra óptica, la presencia de luz podría representar un 1 y la ausencia de luz podría representar un 0. Este esquema se denomina **NRZ** (No Retorno a Cero, del inglés *Non-Return-to-Zero*). El nombre extraño es por cuestiones históricas; simplemente significa que la señal sigue a los datos. En la Ilustración 15(b) se muestra un ejemplo.

Una vez enviada, la señal NRZ se propaga por el cable. En el otro extremo, el receptor la convierte en bits al muestrear la señal a intervalos de tiempo regulares. Esta señal no se verá exactamente igual que la señal que se envió. El canal y el ruido en el receptor la atenuarán y distorsionarán. Para decodificar los bits, el receptor asocia las muestras de la señal con los símbolos más cercanos. Para NRZ, se tomará un voltaje positivo para indicar que se envió un 1 y un voltaje negativo para indicar que se envió un 0.

El esquema NRZ es un buen punto de inicio para nuestros estudios, ya que es simple pero se utiliza pocas veces por sí solo en la práctica. Los esquemas más complejos pueden convertir bits en señales que cumplen mejor con las consideraciones de ingeniería. Estos esquemas se denominan **códigos de línea**. A continuación describiremos códigos de línea que ayudan con la eficiencia del ancho de banda, la recuperación del reloj y el balance de CD.

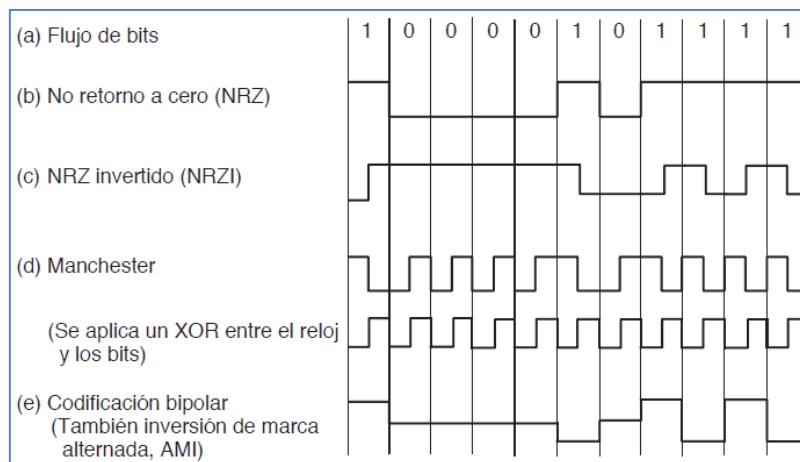


Ilustración 15 - Códigos de línea: (a) Bits, (b) NRZ, (c) NRZI, (d) Manchester, (e) Bipolar o AMI.

Eficiencia del ancho de banda

Con NRZ, la señal puede alternar entre los niveles positivo y negativo hasta cada 2 bits (en caso de alternar 1 s y 0 s). Esto significa que necesitamos un ancho de banda de por lo menos $B/2$ Hz cuando la tasa de bits es de B bits/seg. Esta relación proviene de la tasa de Nyquist. Es un límite fundamental, por lo que no podemos operar el esquema NRZ a una mayor velocidad sin usar más ancho de banda. Por lo general el ancho de banda es un recurso limitado, incluso para los canales con cables. Entre más altas sean las frecuencias de las señales

su atenuación es cada vez mayor, lo que las hace menos útiles; además las señales de frecuencias más altas también requieren componentes electrónicos más rápidos.

Una estrategia para utilizar el ancho de banda limitado con más eficiencia es usar más de dos niveles de señalización. Por ejemplo, si utilizamos cuatro voltajes podemos enviar 2 bits a la vez como un solo **símbolo**. Este diseño funcionará siempre y cuando la señal en el receptor sea lo bastante fuerte como para diferenciar los cuatro niveles. La tasa a la que cambia la señal es entonces la mitad de la tasa de bits, por lo que se reduce el ancho de banda necesario.

La tasa a la que cambia la señal se denomina **tasa de símbolo** para diferenciarla de la **tasa de bits**. La tasa de bits es la tasa de símbolo multiplicada por el número de bits por símbolo. Un nombre antiguo para la tasa de símbolo, en especial dentro del contexto de los dispositivos conocidos como módems telefónicos que transmiten datos digitales a través de las líneas telefónicas, es la **tasa de baudios**. En la literatura es frecuente que los términos “tasa de bits” y “tasa de baudios” se usen en forma incorrecta.

Hay que tener en cuenta que el número de niveles de la señal no necesita ser una potencia de dos. A menudo no lo es, ya que algunos de los niveles se utilizan como protección contra errores y para simplificar el diseño del receptor.

Recuperación del reloj

En todos los esquemas que codifican bits en símbolos, el receptor debe saber cuándo termina un símbolo y empieza el siguiente para decodificar los bits en forma correcta. En el esquema NRZ, en donde los símbolos son sólo niveles de voltaje, una larga sucesión de 0s o 1s deja la señal sin cambios. Después de un rato es difícil diferenciar unos bits de otros, puesto que 15 ceros se ven muy parecidos a 16 ceros, a menos que usted cuente con un reloj muy exacto.

Los relojes exactos serían útiles para resolver este problema, pero son una solución costosa para un equipo básico. Recuerde que vamos a sincronizar bits en enlaces que operan a muchos megabits/seg, por lo que el reloj tendría que variar menos de una fracción de un microsegundo durante la sucesión más larga permitida. Esto podría ser razonable para los enlaces lentos o mensajes cortos, pero no es una solución general.

Una estrategia es enviar una señal de reloj separada al receptor. Otra línea de reloj no representa mucho para los buses de computadora o los cables cortos en donde hay muchas líneas en paralelo, pero sería un desperdicio para la mayoría de los enlaces de red, ya que si tuviéramos otra línea para enviar una señal, la podríamos usar para enviar datos. Un astuto truco que se usa es mezclar la señal de reloj con la señal de datos, mediante la aplicación de una XOR a ambas señales de manera que no se requiera una línea adicional. En la Ilustración 15(d) se muestran los resultados. El reloj realiza una transición en cada tiempo de bit, por lo que opera al doble de la tasa de bits. Al aplicar una XOR con el nivel 0 se produce una transición de nivel bajo a nivel alto, que viene siendo simplemente el reloj. Esta transición es un 0 lógico. Cuando aplica una XOR con el nivel 1, se invierte y produce una transición de nivel alto a nivel bajo. Esta transición es un 1 lógico. Este esquema se llama codificación **Manchester** y se utilizaba en la Ethernet clásica.

La desventaja de la codificación Manchester es que requiere el doble de ancho de banda que NRZ debido al reloj, y hemos aprendido que el ancho de banda es muy importante. Una estrategia distinta se basa en la idea de que deberíamos codificar los datos para asegurar que haya suficientes transiciones en la señal. Consideremos que NRZ tendrá problemas de recuperación del reloj sólo para largas sucesiones de 0s y 1s. Si hay transiciones frecuentes, será fácil para el receptor permanecer sincronizado con el flujo entrante de símbolos.

Para dar un paso en la dirección correcta, podemos simplificar la situación al codificar un 1 como una transición y un 0 como una no transición, o viceversa. A esta codificación se le conoce como **NRZI** (No Retorno a Cero Invertido, del inglés *Non-Return-to-Zero Inverted*), un giro sobre el NRZ. En la Ilustración 15(c) se muestra un ejemplo. El popular estándar USB (Bus Serie Universal, del inglés *Universal Serial Bus*) para conectar periféricos de computadora utiliza NZRI. Con él, las largas sucesiones de 1s no provocan problemas.

Desde luego que las largas sucesiones de 0s siguen provocando un problema que debemos corregir. Si fuéramos una compañía telefónica, tal vez sólo tendríamos que requerir que el emisor no transmitiera

demasiados Os. Las primeras líneas telefónicas digitales en Estados Unidos, conocidas como líneas T1, de hecho requerían que se enviaran como máximo 15 Os consecutivos para funcionar de forma correcta. Para corregir de verdad este problema, podemos descomponer las sucesiones de Os y asociar pequeños grupos de bits, para transmitirlos de forma que los grupos con Os sucesivos se asocien con patrones ligeramente más largos que no tengan muchos Os consecutivos.

Hay un código muy conocido para hacer esto, el cual se llama **4B/5B**. Aquí se asocian grupos de 4 bits a un patrón de 5 bits con una tabla de traducción fija. Los patrones de 5 bits se eligen de tal forma que nunca haya una sucesión de más de tres Os consecutivos. La asociación se muestra en la Ilustración 16. Este esquema agrega un 25% de sobrecarga, lo cual es mejor que la sobrecarga de 100% de la codificación Manchester. Como hay 16 combinaciones de entrada y 32 de salida, algunas de las combinaciones de salida no se utilizan. Haciendo a un lado las combinaciones con demasiados Os sucesivos, aún quedan algunos códigos pendientes. Como bono adicional, podemos usar estos códigos sin datos para representar señales de control de la capa física. Por ejemplo, en algunos casos el patrón “11111” representa una línea inactiva y “11000” representa el inicio de una trama.

Una propuesta alternativa es la aleatorización, o *scrambling*, que consiste en hacer que los datos parezcan aleatorios. En este caso es muy probable que haya transiciones frecuentes. La función del *aleatorizador* o *scrambler* es aplicar una XOR entre los datos y una secuencia pseudoaleatoria antes de transmitirlos. Esta mezcla hará que los datos sean tan aleatorios como la secuencia pseudoaleatoria (suponiendo que sean independientes de la secuencia pseudoaleatoria). Después el receptor aplica una XOR a los bits entrantes con la misma secuencia pseudoaleatoria para recuperar los datos reales. Para que esto sea práctico, la secuencia pseudoaleatoria debe ser fácil de crear. Por lo general se proporciona como la semilla para un generador simple de números aleatorios.

Datos (4B)	Palabra de código (5B)	Datos (4B)	Palabra de código (5B)
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

Ilustración 16 - Asociaciones del esquema 4B/5B.

La aleatorización es atractiva, porque no añade sobrecarga en el ancho de banda ni en el tiempo. De hecho, a menudo ayuda a acondicionar la señal de manera que no tenga su energía en los componentes de frecuencia dominantes (producidos por los patrones de datos repetitivos) que podrían irradiar interferencia electromagnética. La aleatorización es útil debido a que las señales aleatorias tienden a ser “*blancas*” o tienen su energía dispersa a través de los componentes de frecuencia.

Sin embargo, la aleatorización no garantiza que no habrá sucesiones largas. Es posible tener mala suerte en algunas ocasiones. Si los datos son iguales que la secuencia pseudoaleatoria, al aplicar una XOR todos los bits se convertirán en Os. Por lo general este resultado no ocurre con una secuencia pseudoaleatoria larga que sea difícil de predecir. No obstante, con una secuencia corta o predecible podría darse el caso de que usuarios maliciosos enviaran patrones de bits que provocaran largas sucesiones de Os después de la aleatorización y causarán fallas en los enlaces. Las primeras versiones de los estándares para enviar paquetes IP a través de enlaces SONET en el sistema telefónico tenían este defecto (Malis y Simpson, 1999). Los usuarios podían enviar ciertos “paquetes asesinos” que garantizaban provocar problemas.

Señales balanceadas

Las señales que tienen la misma cantidad de voltaje positivo y negativo, incluso durante periodos cortos, se conocen como **señales balanceadas**. Su promedio es cero, lo cual significa que no tienen componente eléctrico de CD (corriente directa o continua). La falta de un componente de CD es una ventaja, ya que algunos canales (como el cable coaxial o las líneas con transformadores) atenúan de manera considerable un componente de CD debido a sus propiedades físicas. Además, un método para conectar el receptor al canal, conocido como **acoplamiento capacitivo**, sólo pasa la porción de CA (corriente alterna) de la señal. En cualquier caso, si enviamos una señal cuyo promedio no sea cero desperdiciaremos energía, puesto que se filtrará el componente de CD.

El balanceo ayuda a proveer transiciones para la recuperación del reloj, ya que hay una mezcla de voltajes positivos y negativos. Además proporciona una forma simple de calibrar los receptores, debido a que se puede medir el promedio de la señal y usarlo como un umbral de decisión para decodificar los símbolos. Con las señales no balanceadas, el promedio puede variar del verdadero nivel de decisión (por ejemplo, debido a una densidad de 1s), lo cual provocaría que se decodificaran más símbolos con errores.

Una manera simple de construir un código balanceado es mediante el uso de dos niveles de voltaje para representar un 1 lógico (por decir, +1 V o -1 V), en donde 0 V representan un cero lógico. Para enviar un 1, el transmisor alterna entre los niveles de +1 V y -1 V de manera que siempre se promedien y eliminen. A este esquema se le llama **codificación bipolar**. En las redes telefónicas se llama **AMI** (Inversión de Marca Alternada, del inglés *Alternate Mark Inversion*), con base en la antigua terminología en donde a un 1 se le llama “*marca*” y a un 0 se le llama “*espacio*”. En la Ilustración 15(e) se muestra un ejemplo.

La codificación bipolar agrega un nivel de voltaje para lograr un balance. También podemos usar una asociación como 4B/5B para lograr un balance (así como transiciones para la recuperación del reloj). El código de línea **8B/10B** es un ejemplo de este tipo de código balanceado, en el cual se asocian 8 bits de entrada a 10 bits de salida, por lo cual tiene una eficiencia de 80%, justo igual que el código de línea 4B/5B. Los 8 bits se dividen en un grupo de 5 bits (el cual se asocia a 6 bits) y un grupo de 3 bits (que se asocia a 4 bits). Después se concatenan los símbolos de 6 y 4 bits. En cada grupo se pueden asociar ciertos patrones de entrada a los patrones de salida balanceados que tengan el mismo número de 0s y 1s. Por ejemplo, “001” se asocia con “1001”, el cual está balanceado. Pero no hay suficientes combinaciones para que todos los patrones de salida estén balanceados. En estos casos, cada patrón de entrada se asocia a dos patrones de salida. Uno tendrá un 1 extra y el otro tendrá un 0 adicional. Por ejemplo, “000” se asocia a “1011” y a su complemento “0100”. A medida que los bits de entrada se asocian a los bits de salida, el codificador recuerda la **disparidad** del símbolo anterior. La disparidad es el número total de 0s o 1s por los que la señal está desbalanceada. Después, el codificador selecciona un patrón de salida o su patrón alterno para reducir la disparidad. Con el código 8B/10B, la disparidad será cuando mucho de 2 bits. Así, la señal nunca estará lejos de ser balanceada. Además, nunca habrá más de cinco 1s o 0s consecutivos para ayudar con la recuperación del reloj.

Transmisión pasa-banda

A menudo es conveniente usar un rango de frecuencias que no empiece en cero para enviar información a través de un canal. En los canales inalámbricos no es práctico enviar señales de muy baja frecuencia, ya que el tamaño de la antena necesita ser de una fracción de la longitud de onda de la señal, por lo que llega a ser grande. En cualquier caso, por lo general la elección de frecuencias se dicta con base en las restricciones regulatorias y a la necesidad de evitar interferencias. Incluso para los cables, es útil colocar una señal en una banda de frecuencias específica para dejar que coexistan distintos tipos de señales en el canal. A este tipo de transmisión se le conoce como **transmisión pasa-banda**, debido a que se utiliza una banda arbitraria de frecuencias para pasar la señal.

Por fortuna, los resultados fundamentales que obtuvimos antes en este capítulo están en términos de ancho de banda, o la anchura de la banda de frecuencias. Los valores absolutos de la frecuencia no importan en cuanto a la capacidad. Esto significa que podemos tomar una señal de **banda base** que ocupe de 0 a B Hz y desplazarla para que ocupe una **banda de paso** de S a S+B Hz sin cambiar la cantidad de información que puede transportar, aun cuando la señal se vea diferente. Para procesar una señal en el receptor, la podemos desplazar de vuelta a la banda base, en donde es más conveniente detectar símbolos.

Para lograr la modulación digital mediante la transmisión pasa-banda, se regula o modula una señal portadora que se sitúa en la banda de paso. Podemos modular la amplitud, frecuencia o fase de la señal portadora. Cada uno de estos métodos tiene su correspondiente nombre. En la **ASK** (Modulación por Desplazamiento de Amplitud, del inglés *Amplitude Shift Keying*) se utilizan dos amplitudes distintas para representar el 0 y 1. En la Ilustración 17(b) se muestra un ejemplo con un nivel distinto de cero y un nivel 0. Se pueden usar más de dos niveles para representar más símbolos. De manera similar, en la **FSK** (Modulación por Desplazamiento de Frecuencia, del inglés *Frequency Shift Keying*) se utilizan dos o más tonos (frecuencias) distintos. El ejemplo en la Ilustración 17(c) utiliza sólo dos frecuencias. En la forma más simple de **PSK** (Modulación por Desplazamiento de Fase, del inglés *Phase Shift Keying*), la onda portadora se desplaza de manera sistemática 0 o 180 grados en cada periodo de símbolo. Como hay dos fases, se llama **BPSK** (Modulación por Desplazamiento de Fase Binaria, del inglés *Binary Phase Shift Keying*). Aquí, la palabra “binaria” se refiere a los dos símbolos, no que los símbolos representan 2 bits. En la Ilustración 17(d) se muestra un ejemplo. Un esquema más conveniente en el que se utiliza el ancho de banda del canal con más eficiencia es el que utiliza cuatro desplazamientos (por ejemplo: 45, 135, 225 o 315 grados) para transmitir 2 bits de información por símbolo. Esta versión se llama **QPSK** (Modulación por Desplazamiento de Fase en Cuadratura, del inglés *Quadrature Phase Shift Keying*).

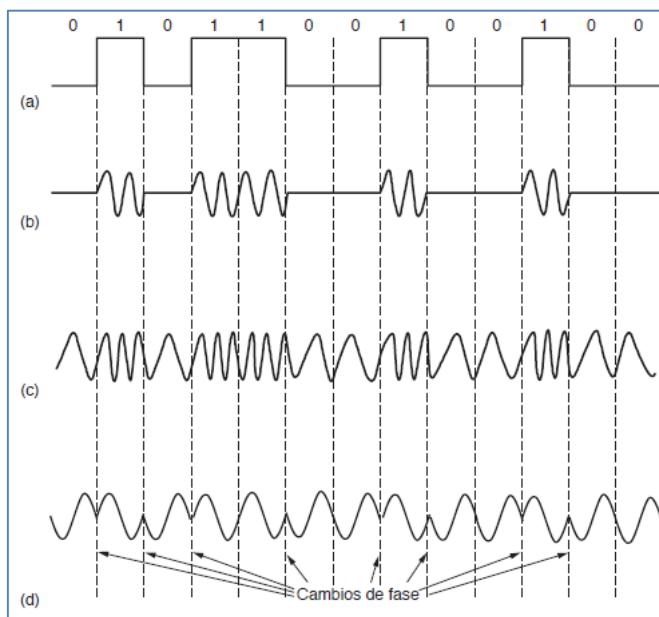


Ilustración 17 - (a) Una señal binaria. (b) Modulación por desplazamiento de amplitud. (c) Modulación por desplazamiento de frecuencia. (d) Modulación por desplazamiento

Podemos combinar estos esquemas y usar más niveles para transmitir más bits por símbolo. Sólo se puede modular la frecuencia o la fase a la vez, ya que están relacionadas; la frecuencia es la tasa de cambio de la fase a través del tiempo. Por lo común, la amplitud y la fase se modulan en combinación. En la Ilustración 18 se muestran tres ejemplos. En cada ejemplo, los puntos proporcionan las combinaciones legales de amplitud y fase de cada símbolo. En la Ilustración 18(a) podemos ver puntos equidistantes a 45, 135, 225 y 315 grados. La fase de un punto se indica mediante el ángulo que hace una línea (que va desde el punto hasta el origen) con el eje x positivo. La amplitud de un punto es la distancia a partir del origen. Esta figura es una representación de QPSK.

A este tipo de diagrama se le conoce como **diagrama de constelación**. En la Ilustración 18(b) podemos ver un esquema de modulación con una constelación más densa. Se utilizan 16 combinaciones de amplitudes y fases, por lo que el esquema de modulación se puede usar para transmitir 4 bits por símbolo. Se denomina **16QAM**, en donde QAM significa Modulación de Amplitud en Cuadratura (en inglés *Quadrature Amplitude Modulation*). La Ilustración 18(c) es un esquema de modulación todavía más denso con 64 combinaciones distintas, por lo que se pueden transmitir 6 bits por símbolo. Se denomina **64QAM**. También se utilizan esquemas QAM más altos. Como podría sospechar de estas constelaciones, es más fácil construir componentes electrónicos para

producir símbolos como una combinación de valores en cada eje, que como una combinación de valores de amplitud y fase. Ésta es la razón por la cual los patrones se ven como cuadros en vez de círculos concéntricos.

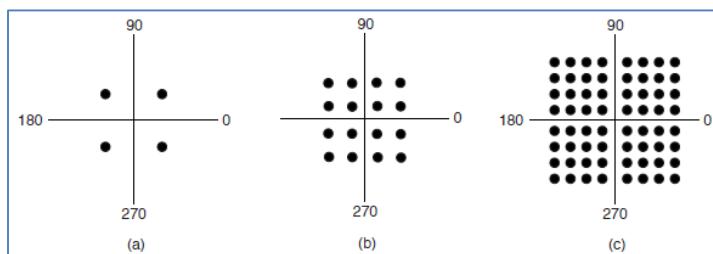


Ilustración 18 - (a) QPSK. (b) QAM-16. (c) QAM-64.

Las constelaciones que hemos visto hasta ahora no muestran cómo se asignan los bits a los símbolos. Es importante considerar que al hacer la asignación una pequeña ráfaga de ruido en el receptor no provoque muchos errores de bits. Esto podría ocurrir si asignáramos valores de bits consecutivos a símbolos adyacentes. Con el 16QAM por ejemplo, si un símbolo representara 0111 y el símbolo adyacente representara 1000, y si el receptor eligiera por error el símbolo adyacente todos los bits estarían incorrectos. Una mejor solución es asociar bits con símbolos de manera que los símbolos adyacentes sólo difieran en 1 posición de bit. A esta asociación se le conoce como **código Gray**. La Ilustración 19 muestra una constelación 16QAM que se ha codificado mediante el **código Gray**. Ahora, si el receptor decodifica un símbolo por error, sólo cometerá un error de un solo bit en el caso esperado en que el símbolo decodificado esté cerca del símbolo transmitido.

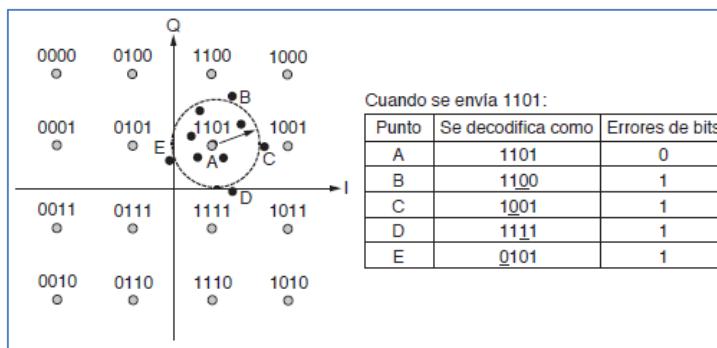


Ilustración 19 - QAM-16 con código Gray.

Multiplexación por división de frecuencia

Los esquemas de modulación que hemos visto nos permiten enviar una señal para transmitir bits a través de un enlace alámbrico o inalámbrico. Sin embargo, la economía de escala desempeña un importante papel en cuanto a la forma en que utilizamos las redes. En esencia, es igual de costoso instalar y mantener una línea de transmisión con un alto ancho de banda que una línea con un bajo ancho de banda entre dos oficinas distintas (es decir, los costos provienen de tener que cavar la zanja y no del tipo de cable o fibra óptica que se va a instalar). Por ende, se han desarrollado esquemas de multiplexación para compartir líneas entre varias señales.

FDM (Multiplexación por División de Frecuencia, del inglés *Frequency Division Multiplexing*) aprovecha la ventaja de la transmisión pasa-banda para compartir un canal. Divide el espectro en bandas de frecuencia, en donde cada usuario tiene posesión exclusiva de cierta banda en la que puede enviar su señal. La difusión de radio AM ilustra el uso del FDM. El espectro asignado es alrededor de 1 MHz, aproximadamente de 500 a 1500 KHz. Las distintas frecuencias se asignan a distintos canales lógicos (estaciones), cada uno de los cuales opera en una parte del espectro y la separación entre canales es lo bastante grande como para evitar interferencias.

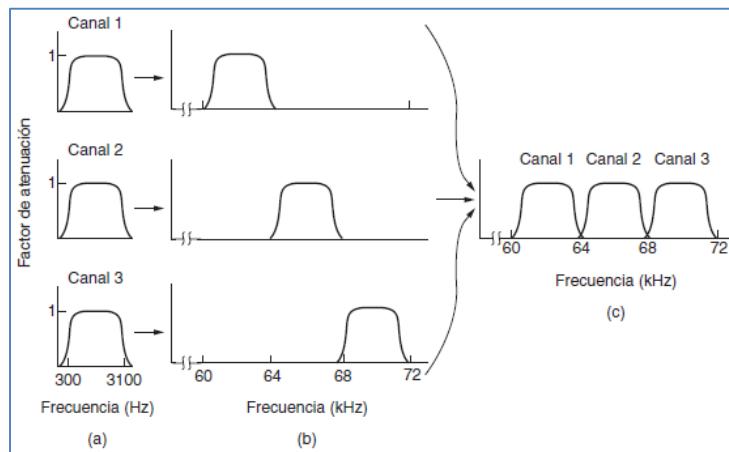


Ilustración 20 - Multiplexión por división de frecuencia. (a) Los anchos de banda originales. (b) Los anchos de banda elevados en frecuencia. (c) El canal multiplexado.

Para un ejemplo más detallado, en la Ilustración 20 mostramos tres canales telefónicos de calidad de voz, multiplexados mediante FDM. Los filtros limitan el ancho de banda útil a cerca de 3100 Hz por cada canal de calidad de voz. Cuando se multiplexan muchos canales juntos, se asignan 4000 Hz por canal. Al exceso se le denomina **banda de guarda**, la cual mantiene los canales bien separados. Primero, los canales de voz se elevan en frecuencia, cada uno en distinto grado. Después se pueden combinar debido a que no hay dos canales que ocupen la misma porción del espectro. Hay que tener en cuenta que, aun cuando hay vacíos entre los canales gracias a las bandas de guarda, existe cierto traslape entre los canales adyacentes. El traslape se debe a que los filtros reales no tienen bordes ideales que sean muy definidos. Esto significa que un pico fuerte en el borde de un canal se detectará en el canal adyacente como ruido no térmico.

Este esquema se ha utilizado para multiplexar llamadas en el sistema telefónico durante muchos años, pero ahora se prefiere más la multiplexación en el tiempo. Sin embargo, FDM se sigue utilizando en las redes telefónicas, así como en las redes celulares, redes inalámbricas terrestres y redes de satélites con un mayor nivel de granularidad.

Al enviar datos digitales, es posible dividir el espectro de manera eficiente sin usar bandas de guarda. En **OFDM** (Multiplexación por División de Frecuencia Ortogonal, del inglés *Orthogonal Frequency Division Multiplexing*), el ancho de banda del canal se divide en muchas subportadoras que envían datos de manera independiente (por ejemplo, mediante QAM). Las subportadoras están empaquetadas estrechamente en el dominio de la frecuencia. Por lo tanto, las señales de cada subportadora se extienden a las subportadoras adyacentes. Pero como podemos ver en la Ilustración 21, la respuesta en frecuencia de cada subportadora está diseñada de manera que sea cero en el centro de las subportadoras adyacentes. Por lo tanto, las subportadoras se pueden muestrear en sus frecuencias centrales sin interferencia de sus vecinas. Para que esto funcione, se necesita un tiempo de guarda para repetir una parte de las señales de los símbolos a tiempo, de manera que tengan la respuesta en frecuencia deseada. Sin embargo, esta sobrecarga es mucho menor de lo que se necesita para muchas bandas de guarda.

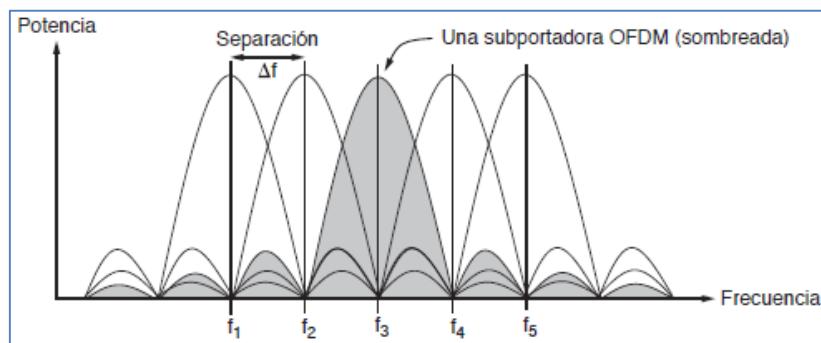


Ilustración 21 - Multiplexación por División de Frecuencia Ortogonal (OFDM).

La idea de OFDM ha estado presente por mucho tiempo, pero sólo a partir de esta último tiempo se empezó a adoptar en muchas aplicaciones, después de haberse dado cuenta de que es posible implementar OFDM con eficiencia en términos de una transformada de Fourier de datos digitales sobre todas las subportadoras (en vez de modular por separado cada subportadora).

Multiplexación por división de tiempo

TDM (Multiplexación por División de Tiempo, del inglés *Time Division Multiplexing*) es una alternativa a FDM. Aquí, los usuarios toman turnos (rotatorios tipo *round-robin*) y cada uno recibe periódicamente todo el ancho de banda durante una pequeña ráfaga de tiempo. En la Ilustración 22 se muestra un ejemplo de tres flujos multiplexados mediante TDM. Se toman bits de cada flujo de entrada en una ranura de tiempo fija y se envían al flujo agregado. Este flujo opera a una velocidad equivalente a la suma de los flujos individuales. Para que esto funcione, los flujos se deben estar sincronizados en tiempo. Se pueden agregar pequeños intervalos de **tiempo de guarda**, los cuales son análogos a una banda de guarda de frecuencia, para tener en cuenta las pequeñas variaciones de sincronización.

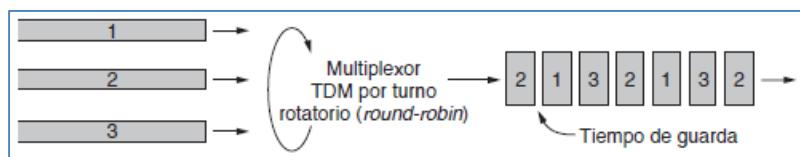


Ilustración 22 - Multiplexación por División de Tiempo (TDM).

El TDM se utiliza mucho como parte de las redes telefónicas y celulares. Para evitar un punto de confusión, dejemos claro que es muy distinto a la **STDIM** (Multiplexación Estadística por División de Tiempo, del inglés *Statistical Time Division Multiplexing*). El prefijo “estadística” es para indicar que los flujos individuales contribuyen al flujo multiplexado no en un itinerario fijo, sino con base en la estadística de su demanda. En sí, STDIM es otro nombre para la conmutación de paquetes.

Multiplexación por división de código

Hay un tercer tipo de multiplexación que funciona de una manera muy distinta a FDM y a TDM. **CDM** (Multiplexación por División de Código, del inglés *Code Division Multiplexing*) es una forma de comunicación de espectro disperso en la que una señal de banda estrecha se dispersa sobre una banda de frecuencia más amplia. Esto puede hacerla más tolerante a la interferencia, al tiempo que permite que varias señales de distintos usuarios comparten la misma banda de frecuencia. Como la multiplexación por división de código se utiliza la mayoría de las veces para este último propósito, se le conoce comúnmente como **CDMA** (Acceso Múltiple por División de Código, del inglés *Code Division Multiple Access*).

CDMA permite que cada estación transmita en todo el espectro de frecuencia todo el tiempo. Las múltiples transmisiones simultáneas se separan mediante el uso de la teoría de codificación. Antes de entrar en detalles del algoritmo, consideremos una analogía: una sala de espera en un aeropuerto con muchas parejas conversando. Podemos comparar a TDM con parejas de personas en el cuarto que toman turnos para hablar. FDM es comparable a las parejas de personas que hablan en distintos tonos, algunas en tonos agudos y otras en tonos bajos, de tal forma que cada pareja puede sostener su propia conversación al mismo tiempo, pero de manera independiente a los demás. CDMA se puede comparar con cada pareja de personas que habla a la vez, pero en un lenguaje distinto. La pareja que habla francés sólo se concentra en el francés y rechaza todo lo que no sea francés, pues lo considera ruido. Así, la clave del CDMA es extraer la señal deseada mientras todo lo demás se rechaza como ruido aleatorio. A continuación veremos 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 cada bit, pero en el ejemplo que veremos aquí utilizamos 8 chips/bit por cuestión de simplicidad. A cada estación se le asigna un código único de m bits, o **secuencia de chip**. Para fines pedagógicos, es conveniente usar una notación bipolar para escribir estos códigos como secuencias de -1 y +1. Mostraremos las secuencias de chip entre paréntesis.

Para transmitir un bit 1, una estación envía su secuencia de chip. Para transmitir un bit 0, envía la negación de su secuencia de chip. No se permite ningún otro patrón. Así, para $m=8$, si se asigna a la estación A la secuencia de chip $(-1 -1 -1 +1 +1 -1 +1 +1)$, para enviar un bit 1 transmite la secuencia de chip y para enviar un 0 transmite $(+1 +1 +1 -1 -1 +1 -1 -1)$. En realidad lo que se envía son señales con estos niveles de voltaje, pero es suficiente para nosotros pensar en términos de las secuencias.

La acción de incrementar la cantidad de información a enviar de b bits/seg a mb chips/seg para cada estación significa que el ancho de banda necesario para CDMA es mayor por un factor de m que el ancho de banda necesario para una estación que no utilice CDMA (suponiendo que no haya cambios en las técnicas de modulación o de codificación). Si tenemos una banda de 1 MHz disponible para 100 estaciones, con FDM cada estación tendría 10 kHz y podría enviar a 10 kbps (suponiendo 1 bit por Hz). Con CDMA, cada estación utiliza el 1 MHz completo, por lo que la tasa de chip es de 100 chips por bit para dispersar la tasa de bits de la estación de 10 kbps a través del canal.

En las Ilustración 23(a) y (b) se muestran las secuencias de chip asignadas a cuatro estaciones de ejemplo y las señales que representan. Cada estación tiene su propia secuencia de chip única. Utilizaremos el símbolo S para indicar el vector de m chips para la estación S , y \bar{S} para su negación. Todas las secuencias de chip son **ortogonales** por pares, lo que quiere decir que el producto interno normalizado de dos distintas secuencias de chip cualesquiera, S y T (lo que se escribe como $S \bullet T$), es 0. Se sabe cómo generar dichas secuencias de chip ortogonales mediante un método conocido como **códigos de Walsh**. En términos matemáticos, la ortogonalidad de las secuencias de chip se puede expresar de la siguiente manera:

$$S \bullet T \equiv \frac{1}{m} \sum_{i=1}^m S_i T_i = 0$$

En español simple, los pares son tan iguales como distintos. Esta propiedad de ortogonalidad demostrará ser imprescindible más adelante. Observe que si $S \bullet T = 0$, entonces $S \bullet T$ también es 0. El producto interno normalizado de cualquier secuencia de chip consigo misma es 1:

$$S \bullet S \equiv \frac{1}{m} \sum_{i=1}^m S_i S_i = \frac{1}{m} \sum_{i=1}^m S_i^2 = \frac{1}{m} \sum_{i=1}^m (\pm 1)^2 = 1$$

Se deduce esto debido a que cada uno de los m términos en el producto interno es 1, por lo que la suma es m . Observe además que $S \bullet \bar{S} = -1$.

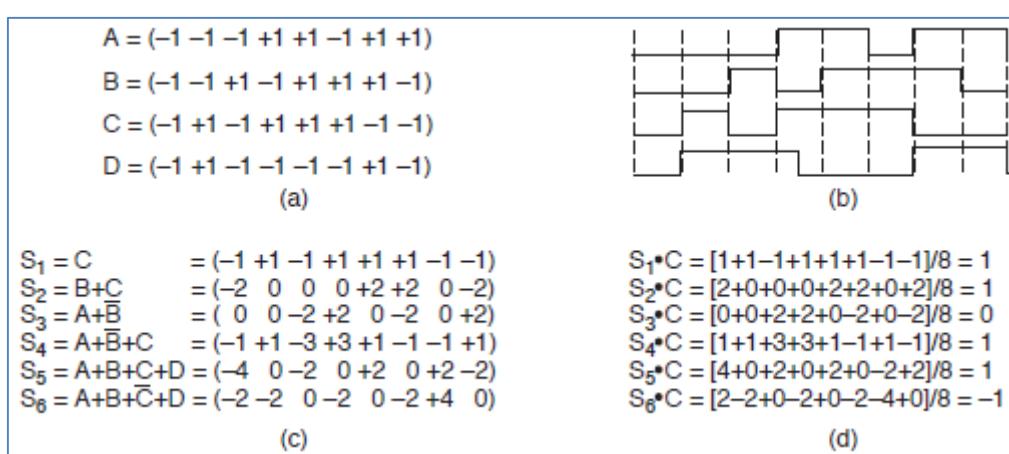


Ilustración 23 - (a) Secuencias de chip para cuatro estaciones. (b) Las señales que representan las secuencias. (c) Seis ejemplos de transmisiones. (d) Recuperación de la señal de la estación C.

Durante cada tiempo de bit, una estación puede transmitir un 1 (si envía su secuencia de chip), puede transmitir un 0 (si envía el negativo de su secuencia de chip) o puede permanecer en silencio y no transmitir nada. Por ahora supongamos que todas las estaciones están sincronizadas en el tiempo, por lo que todas las secuencias de chip empiezan en el mismo instante. Cuando dos o más estaciones transmiten de manera simultánea, sus secuencias bipolares se suman en forma lineal. Por ejemplo, si en un periodo de chip tres

estaciones envían +1 y una estación envía -1, se recibirá +2. Podemos considerar esto como señales que se suman a medida que se sobreponen voltajes en el canal: tres estaciones envían +1 V y una estación envía -1 V, de modo que se reciben 2 V. Por ejemplo, en la Ilustración 23(c) vemos seis ejemplos de una o más estaciones que transmiten 1 bit al mismo tiempo. En el primer ejemplo, C transmite un bit 1, así que sólo recibimos la secuencia de chip de C. En el segundo ejemplo, tanto B como C transmiten bits 1, por lo que obtenemos la suma de sus secuencias de chip bipolares, es decir:

$$\begin{array}{r}
 & (-1 & -1 & +1 & -1 & +1 & +1 & +1 & -1) \\
 + & (-1 & +1 & -1 & +1 & +1 & +1 & -1 & -1) \\
 \hline
 & (-2 & 0 & 0 & 0 & +2 & +2 & 0 & -2)
 \end{array}$$

Para recuperar el flujo de bits de una estación individual, el receptor debe conocer de antemano la secuencia de chip de esa estación. Para llevar a cabo la recuperación, calcula el producto interno normalizado de la secuencia de chip recibida y de la secuencia de chip de la estación cuyo flujo de bits está tratando de recuperar. Si la secuencia de chip recibida es S y el receptor trata de escuchar una estación cuya secuencia de chip sea C, sólo calcula el producto interno normalizado, $S \bullet C$.

Para ver por qué funciona esto, sólo imagine que dos estaciones A y C transmiten un bit 1 al mismo tiempo que B transmite un bit 0, como se da el caso en el tercer ejemplo. El receptor ve la suma, $S = A + \bar{B} + C$, y calcula lo siguiente:

$$S \bullet C = (A + \bar{B} + C) \bullet C = A \bullet C + \bar{B} \bullet C + C \bullet C = 0 + 0 + 1 = 1$$

Los primeros dos términos se desvanecen debido a que todos los pares de secuencias de chip se han elegido con cuidado para que sean ortogonales. Ahora debe quedar claro por qué se debe imponer esta propiedad en las secuencias de chip.

Para que el proceso de decodificación sea más concreto, en la Ilustración 23(d) se muestran seis ejemplos. Suponga que el receptor está interesado en extraer el bit enviado por la estación C de cada una de las seis señales S_1 a S_6 . Para calcular el bit, suma los productos por parejas de la S recibida y el vector C de la Ilustración 23(a), y después toma 1/8 del resultado (ya que $m=8$ en este caso). Los ejemplos incluyen casos en donde C está en silencio, envía un bit 1 y envía un bit 0, por separado y en combinación con otras transmisiones. Como se muestra, se decodifica el bit correcto cada vez.

En principio, dada la suficiente capacidad de cómputo, el receptor puede escuchar a todas las emisoras a la vez si ejecuta el algoritmo de decodificación para cada una de ellas en paralelo. En la vida real basta señalar que es más fácil decirlo que hacerlo, además de que es conveniente saber qué emisoras podrían estar transmitiendo. En el sistema CDMA ideal sin ruido que hemos estudiado aquí, la cantidad de estaciones que envían datos en forma concurrente puede ser arbitrariamente grande si utilizamos secuencias de chip más largas. Para 2^n estaciones, los códigos de Walsh pueden proveer 2^n secuencias de chip ortogonales de longitud 2^n . No obstante, una limitación considerable es que hemos supuesto que todos los chips están sincronizados en el tiempo en el receptor. Esta sincronización ni siquiera está cerca de ser verdad en algunas aplicaciones, como las redes celulares (en donde se empezó a implementar CDMA en muchos casos desde la década de 1990).

Al igual que en las redes celulares, CDMA se utiliza en las redes de satélites y de cable. En esta breve introducción pasamos por alto muchos factores que complicarían el tema.

Principales protocolos de capa física

Ethernet

Ethernet es un estándar de redes de computadoras de área local. Define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI.

Ethernet se tomó como base para la redacción del estándar internacional **IEEE 802.3**. Usualmente se toman Ethernet e IEEE 802.3 como sinónimos. Ambas se diferencian en uno de los campos de la trama de datos. Las tramas Ethernet e IEEE 802.3 pueden coexistir en la misma red.

Objetivos de Ethernet

Los objetivos principales de Ethernet son consistentes con los que se han convertido en los requerimientos básicos para el desarrollo y uso de redes LAN.

- Simplicidad
- Bajo Costo
- Compatibilidad
- Direccionamiento flexible
- Equidad
- Progreso
- Alta velocidad
- Bajo retardo
- Estabilidad
- Mantenimiento

Características de Ethernet

Ethernet usa un método de acceso al medio por disputa (*contention*). Las transmisiones son difundidas en el canal compartido para ser escuchadas por todos los dispositivos conectados, solo el dispositivo de destino previsto va a aceptar la transmisión. Este tipo de acceso es conocido como CSMA/CD.

Ethernet ha evolucionado para operar sobre una variedad de medios, cable coaxial, par trenzado y fibra óptica, a múltiples tasas de transferencia. Todas las implementaciones son interoperables, lo que simplifica el proceso de migración a nuevas versiones de Ethernet.

Múltiples segmentos de Ethernet pueden ser conectados para formar una gran red LAN Ethernet utilizando repetidores. La correcta operación de una LAN Ethernet depende en que los segmentos del medio sean construidos de acuerdo a las reglas para ese tipo de medio. Redes LAN complejas construidas con múltiples tipos de medio deben ser diseñadas de acuerdo a las pautas de configuración para multisegmentos provistas en el estándar Ethernet. Las reglas incluyen límites en el número total de segmentos y repetidores que pueden ser utilizados en la construcción de una LAN.

Ethernet fue diseñado para ser expandido fácilmente. El uso de dispositivos de interconexión tales como puente (*bridges*), ruteadores (*routers*), y conmutadores (*switches*) permiten que redes LAN individuales se conecten entre sí. Cada LAN continúa operando en forma independiente pero es capaz de comunicarse fácilmente con las otras LAN conectadas.

Principios de operación de Ethernet

Cada dispositivo equipado con Ethernet opera en forma independiente del resto de los dispositivos de la red, las redes Ethernet no hacen uso de un dispositivo central de control. Todos los dispositivos son conectados a un canal de comunicaciones de señales compartidas.

Las señales Ethernet son transmitidas en serie, se transmite un bit a la vez. Las transmisiones se realizan a través del canal de señales compartidas donde todos los dispositivos conectados pueden escuchar la transmisión. Antes de comenzar una transmisión, un dispositivo escucha el canal de transmisión para ver si se encuentra libre de transmisiones. Si el canal se encuentra libre, el dispositivo puede transmitir sus datos en la forma de una trama Ethernet.

Después de que es transmitida una trama, todos los dispositivos de la red compiten por la siguiente oportunidad de transmitir una trama. La disputa por la oportunidad de transmitir entre los dispositivos es pareja, para asegurar que el acceso al canal de comunicaciones sea justo, ningún dispositivo puede bloquear a otros dispositivos.

El acceso al canal de comunicaciones compartido es determinado por la subcapa **MAC**. Este control de acceso al medio es conocido como **CSMA/CS**.

Direccionamiento

Los campos de direcciones en una trama Ethernet llevan direcciones de 48 bits, tanto para la dirección de destino como la de origen. El estándar IEEE administra parte del campo de las direcciones mediante el control de la asignación de un identificador de 24 bits conocido como **OUI** (*Organizationally Unique Identifier*, identificador único de organización). A cada organización que desee construir interfaces de red (NIC) Ethernet, se le asigna un OUI de 24 bits único, el cual es utilizado como los primeros 24 bits de la dirección de 48 bits del NIC. La dirección de 48 bits es referida como dirección física, dirección de hardware, o **dirección MAC**.

El uso de direcciones únicas preasignadas, simplifica el montaje y crecimiento de una red Ethernet.

La topología lógica de una red determina como las señales son transferidas en la red. La topología lógica de una red Ethernet provee un único canal de comunicaciones que transporta señales de todos los dispositivos conectados. Esta topología lógica puede ser diferente de la topología física o de la disposición real del medio.

Múltiples segmentos Ethernet pueden ser interconectados utilizando repetidores para formar una red LAN más grande. Cada segmento de medio es parte del sistema de señales completo. Este sistema de segmentos interconectados nunca es conectado en forma de bucle, es decir, cada segmento debe tener dos extremos.

La señal generada por un dispositivo es puesta en el segmento de medio al cual está conectado. La señal es repetida en todos los otros segmentos conectados de forma que sea escuchada por todos las demás estaciones. Sin importar cual sea la topología física, solo existe un canal de señales para entregar tramas a través de todos los segmentos a todos los dispositivos conectados.

Tiempo de señales

Para que el método de control de acceso al medio funcione correctamente, todas las interfaces de red Ethernet deben poder responder a las señales dentro de una cantidad de tiempo especificada. El **tiempo de la señal** está basado en la cantidad de tiempo que le toma a una señal ir de un extremo de la red al otro y regresar (*Round Trip Time*).

El límite del *Round Trip Time* debe alcanzar a pesar de que combinación de segmento de medio se utilicen en la construcción de la red. Las pautas de configuración proveen las reglas para la combinación de segmentos con repetidores de forma que el tiempo de las señales se mantenga. Si estas reglas no son seguidas, las estaciones podrían no llegar a escuchar las transmisiones a tiempo y las señales de estas estaciones pondrían interferirse entre sí, causando colisiones tardías y congestionamiento en la red.

Los segmentos del medio deben ser construidos de acuerdo a las pautas de configuración para el tipo de medio elegido y la velocidad de transmisión de la red (las redes de mayor velocidad exigen un tamaño de red de menor). Las redes locales Ethernet construidas por múltiples tipos de medios deben ser diseñadas siguiendo las pautas para configuraciones multi-segmento del estándar Ethernet.

WiFi

Es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos. Los dispositivos habilitados con WiFi (tales como ordenadores personales, teléfonos, televisores, reproductores de música, etc.) pueden conectarse entre sí o a internet a través de un punto de acceso de red inalámbrica.

WiFi es una marca de la **Alianza Wi-Fi**, la organización comercial que cumple con los estándares 802.11 relacionados con redes inalámbricas de área local. Su primera denominación fue, en inglés, *Wireless Ethernet Compatibility Alliance*.

Historia

Esta nueva tecnología surgió por la necesidad de establecer un mecanismo de conexión inalámbrica que fuese compatible entre distintos dispositivos. Buscando esa compatibilidad, en 1999 las empresas 3Com, Airones, Intersil, Lucent Technologies, Nokia y Symbol Technologies se unieron para crear la *Wireless Ethernet Compatibility Alliance*, o **WECA**, actualmente llamada Alianza Wi-Fi. El objetivo de la misma fue designar una

marca que permitiese fomentar más fácilmente la tecnología inalámbrica y asegurar la compatibilidad de equipos.

De esta forma, en abril de 2000 WECA certifica la interoperabilidad de equipos según la norma IEEE 802.11b, bajo la marca Wi-Fi. Esto quiere decir que el usuario tiene la garantía de que todos los equipos que tengan el sello Wi-Fi pueden trabajar juntos sin problemas, independientemente del fabricante de cada uno de ellos.

En el año 2002, la asociación WECA estaba formada ya por casi 150 miembros en su totalidad. La familia de estándares 802.11 ha ido naturalmente evolucionando desde su creación, mejorando el rango y velocidad de la transferencia de información, su seguridad, entre otras cosas.

La norma IEEE 802.11 fue diseñada para sustituir el equivalente a las capas físicas y MAC de la norma 802.3 (Ethernet). Esto quiere decir que en lo único que se diferencia una red wifi de una red Ethernet es en cómo se transmiten las tramas o paquetes de datos; el resto es idéntico. Por tanto, una red local inalámbrica 802.11 es completamente compatible con todos los servicios de las redes locales (LAN) de cable 802.3 (Ethernet).

Estándares que certifica la Alianza Wi-Fi

Existen diversos tipos de wifi, basados cada uno de ellos en un estándar IEEE 802.11. Son los siguientes:

Los estándares IEEE 802.11b, IEEE 802.11g e IEEE 802.11n disfrutan de una aceptación internacional debido a que la banda de 2,4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente. El problema es que existen otras tecnologías inalámbricas que también funcionan a una frecuencia de 2,4 GHz, como Bluetooth, por lo que pueden presentar interferencias con la tecnología wifi. Debido a esto, en la versión 1.2 del estándar Bluetooth, por ejemplo, se actualizó su especificación para que no existieran interferencias con la utilización simultánea de ambas tecnologías.

Desde 2013 existe también el estándar IEEE 802.11ac, conocido como WIFI 5, que opera en la banda de 5 GHz y que disfruta de una operatividad con canales relativamente limpios. La banda de 5 GHz ha sido habilitada con posterioridad a las usadas por versiones anteriores y, al no existir otras tecnologías (Bluetooth, microondas, ZigBee, WUSB) que la utilicen, se producen muy pocas interferencias. Su alcance es algo menor que el de los estándares que trabajan a 2,4 GHz, debido a que la frecuencia es mayor (a mayor frecuencia, menor alcance).

Seguridad y fiabilidad

Uno de los problemas a los cuales se enfrenta actualmente la tecnología wifi es la progresiva saturación del espectro radioeléctrico, debido a la masificación de usuarios; esto afecta especialmente en las conexiones de larga distancia (mayor de 100 metros). En realidad el estándar wifi está diseñado para conectar computadoras a la red a distancias reducidas, cualquier uso de mayor alcance está expuesto a un excesivo riesgo de interferencias.

Un elevado porcentaje de redes se instalan sin tener en consideración la seguridad, convirtiéndose así en redes abiertas (completamente accesible a terceras personas), sin proteger la información que por ellas circulan. De hecho, la configuración por defecto de muchos dispositivos wifi es muy insegura (routers, por ejemplo) dado que a partir del identificador del dispositivo se puede conocer la contraseña de acceso de este y, por tanto, se puede conseguir fácilmente acceder y controlar el dispositivo .

El acceso no autorizado a un dispositivo wifi es muy peligroso para el propietario por varios motivos. El más obvio es que pueden utilizar la conexión. Pero, además, accediendo al wifi se puede supervisar y registrar toda la información que se transmite a través de él (incluyendo información personal, contraseñas, etc.). Existen formas de hacerlo seguro:

- Cambios frecuentes de la contraseña de acceso, utilizando diversos caracteres, minúsculas, mayúsculas y números.
- Modificación del SSID que viene predeterminado.
- Desactivación de la difusión de SSID y DHCP.

- Configuración de los dispositivos conectados con su dirección MAC (indicando específicamente qué dispositivos están autorizados para conectarse).
- Utilización de cifrado: WPA2.

Existen varias alternativas para garantizar la seguridad de estas redes. Las más comunes son la utilización de protocolos de cifrado de datos para los estándares wifi como el WEP, el WPA, o el WPA2 que se encargan de codificar la información transmitida para proteger su confidencialidad, proporcionados por los propios dispositivos inalámbricos.

Dispositivos

Existen varios dispositivos wifi, los cuales se pueden dividir en dos grupos: dispositivos de distribución o de red, entre los que destacan los routers, puntos de acceso y repetidores; y dispositivos terminales que en general son las tarjetas receptoras para conectar a la computadora personal, ya sean internas (tarjetas PCI) o bien USB.

- Dispositivos de distribución o de red:
 - ✓ Los **puntos de acceso** son dispositivos que generan un set de servicio que podría definirse como una red wifi a la que se pueden conectar otros dispositivos. Los puntos de acceso permiten conectar dispositivos de forma inalámbrica a una red existente.
 - ✓ Los **repetidores inalámbricos** son equipos que se utilizan para extender la cobertura de una red inalámbrica. Se conectan a una red existente que tiene señal más débil y crean una señal más fuerte a la que se pueden conectar los equipos dentro de su alcance.
 - ✓ Los **routers inalámbricos** son dispositivos compuestos especialmente diseñados para redes pequeñas (hogar o pequeña oficina). Estos dispositivos incluyen un router (encargado de interconectar redes; por ejemplo, nuestra red del hogar con Internet), un punto de acceso (explicado más arriba) y generalmente un conmutador que permite conectar algunos equipos vía cable (Ethernet y USB). Su tarea es tomar la conexión a Internet y brindar a través de ella acceso a todos los equipos que conectemos, sea por cable o en forma inalámbrica.
- Los dispositivos terminales abarcan tres tipos mayoritarios: tarjetas PCI, tarjetas PCMCIA y tarjetas USB.
- También existen impresoras, cámaras Web y otros periféricos que funcionan con la tecnología wifi, permitiendo un ahorro de mucho cableado en las instalaciones de redes y especialmente gran movilidad de equipo.

Ventajas y desventajas

Las redes wifi poseen una serie de ventajas, entre las cuales podemos destacar:

- La comodidad que ofrecen es muy superior a las redes cableadas porque cualquiera que tenga acceso a la red puede conectarse desde distintos puntos dentro de un espacio lo bastante amplio.
- Una vez configuradas, las redes wifi permiten el acceso de múltiples dispositivos sin ningún problema ni gasto en infraestructura, ni gran cantidad de cables.
- La Alianza Wi-Fi asegura que la compatibilidad entre dispositivos con la marca Wi-Fi es total, con lo que en cualquier parte del mundo podremos utilizar la tecnología wifi con una compatibilidad absoluta.

Pero como red inalámbrica, la tecnología wifi presenta los problemas intrínsecos de cualquier tecnología inalámbrica. Algunos de ellos son:

- Menor velocidad en comparación a una conexión cableada, debido a las interferencias y pérdidas de señal que el ambiente puede acarrear.
- La seguridad. Existen algunos programas capaces de capturar paquetes, trabajando con su tarjeta wifi en modo promiscuo, de forma que puedan calcular la contraseña de la red y de esta forma acceder a ella.
- No es compatible con otros tipos de conexiones sin cables como Bluetooth, GPRS, UMTS, etc.

- La potencia de la conexión del wifi se verá afectada por los agentes físicos que se encuentran a nuestro alrededor, tales como: árboles, paredes, arroyos, una montaña, etc. Dichos factores afectan la potencia de compartimiento de la conexión wifi con otros dispositivos.

Capa de enlace de datos

En esta sección nos enfocaremos en el estudio de los algoritmos para lograr una comunicación confiable y eficiente de unidades completas de información llamadas tramas (en vez de bits individuales, como en la capa física) entre dos máquinas adyacentes. Por adyacente, queremos decir que las dos máquinas están conectadas mediante un canal de comunicaciones que actúa de manera conceptual como un alambre (por ejemplo, un cable coaxial, una línea telefónica o un canal inalámbrico). La propiedad esencial de un canal que lo hace asemejarse a un “alambre” es que los bits se entregan exactamente en el mismo orden en que se enviaron.

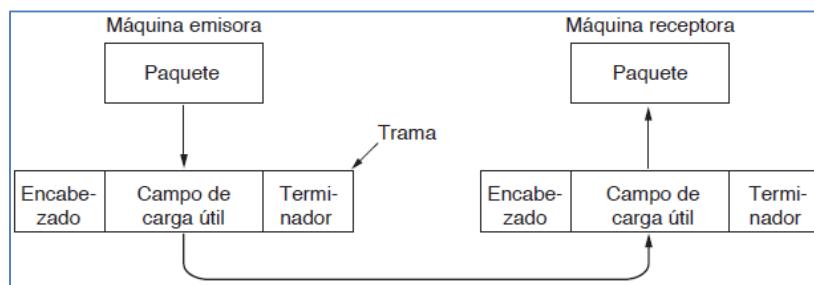
A primera vista se podría pensar que este problema es tan trivial que no hay nada que estudiar: la máquina A sólo pone los bits en el alambre y la máquina B simplemente los toma. Por desgracia, en ocasiones los canales de comunicación cometan errores. Además, sólo tienen una tasa de transmisión de datos finita y hay un retardo de propagación distinto 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.

Cuestiones de diseño de la capa de enlace de datos

La capa de enlace de datos utiliza los servicios de la capa física para enviar y recibir bits a través de los canales de comunicación. Tiene varias funciones específicas, entre las que se incluyen:

1. Proporcionar a la capa de red una interfaz de servicio bien definida.
2. Manejar los errores de transmisión.
3. Regular el flujo de datos para que los emisores rápidos no saturen a los receptores lentos.

Para cumplir con estas metas, la capa de enlace de datos toma los paquetes que obtiene de la capa de red 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, como se muestra en la Ilustración 24. El manejo de las tramas es la tarea más importante de la capa de enlace de datos.



Aunque hablaremos de manera explícita sobre la capa de enlace de datos y sus protocolos, muchos de los principios que se detallan en adelante, como el control de errores y el control de flujo, están presentes también en la capa de transporte y en otros protocolos. Esto se debe a que la confiabilidad es una meta general que se logra cuando todas las capas trabajan en conjunto. De hecho, en muchas redes estas funciones se encuentran casi siempre en las capas superiores y la capa de enlace de datos sólo realiza la tarea mínima que es “suficiente”. No obstante y sin importar en dónde se encuentren, estos principios son básicamente los mismos. A menudo aparecen en sus formas más simples y puras en la capa de enlace de datos, lo que la convierte en un buen lugar para examinarlos a detalle.

Servicios proporcionados a la capa de red

La función de la capa de enlace de datos es proveer servicios a la capa de red. El servicio principal es la transferencia de 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 está una entidad, llamada proceso, que entrega algunos bits a la capa de enlace de datos para que los transmita al destino. La tarea de la capa de enlace de datos es transmitir los bits a la máquina de destino, de modo que se puedan entregar a la capa de red de esa máquina, como se muestra en la Ilustración 25(a). La transmisión real sigue la trayectoria de la Ilustración 25(b), pero es más fácil pensar en términos de dos procesos de la capa de enlace de datos que se comunican mediante un protocolo de enlace de datos. Por esta razón utilizaremos de manera implícita el modelo de la Ilustración 25(a).

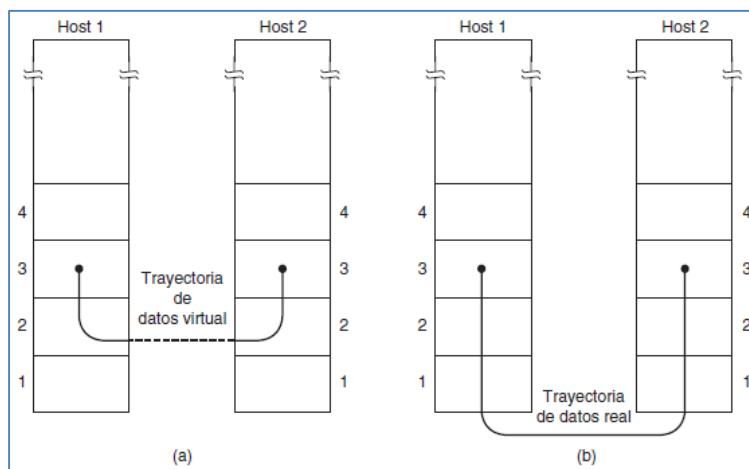


Ilustración 25 - (a) Comunicación virtual. (b) Comunicación real.

La capa de enlace de datos puede diseñarse para ofrecer varios servicios. Los servicios reales ofrecidos varían de un protocolo a otro. Tres posibilidades razonables que normalmente se proporcionan son:

1. Servicio sin conexión ni confirmación de recepción.
2. Servicio sin conexión con confirmación de recepción.
3. Servicio orientado a conexión con confirmación de recepción.

El servicio sin conexión ni confirmación de recepción consiste en hacer que la máquina de origen envíe tramas independientes a la máquina de destino sin que ésta confirme la recepción. Ethernet es un buen ejemplo de una capa de enlace de datos que provee esta clase de servicio. No se establece una conexión lógica de antemano ni se libera después. Si se pierde una trama debido a ruido en la línea, en la capa de enlace no se realiza ningún intento por detectar la pérdida o recuperarse de ella. Esta clase de servicio es apropiada cuando la tasa de error es muy baja, de modo que la recuperación se deja a las capas superiores. También es apropiada para el tráfico en tiempo real, como el de voz, en donde es peor tener retraso en los datos que errores en ellos.

El siguiente paso en términos de confiabilidad es el servicio sin 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 llegó bien o se perdió. Si no ha llegado en un intervalo especificado, se puede enviar de nuevo. Este servicio es útil en canales no confiables, como los de los sistemas inalámbricos. 802.11 (WiFi) es un buen ejemplo de esta clase de servicio.

Tal vez valga la pena enfatizar que el hecho de proporcionar confirmaciones de recepción en la capa de enlace de datos es tan sólo una optimización, nunca un requerimiento. La capa de red siempre puede enviar un paquete y esperar a que su igual en la máquina remota confirme su recepción. Si la confirmación no llega antes de que expire el temporizador, el emisor puede volver a enviar el mensaje completo. El problema con esta estrategia es que puede ser ineficiente. Por lo general los enlaces tienen una estricta longitud máxima para la trama, la cual es impuesta por el hardware, además de los retardos de propagación conocidos. La capa de red no conoce estos parámetros. Podría enviar un paquete largo que se divida, por ejemplo, en 10 tramas, de las cuales pudieran perderse dos en promedio. Entonces se requeriría mucho tiempo para que el paquete pudiera

llegar al otro extremo. Por el contrario, si las tramas se confirman y retransmiten de manera individual, entonces los errores pueden corregirse de una manera más directa y rápida. 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 (no confiables por naturaleza) bien vale la pena el costo.

Recapitulando el servicio más sofisticado que puede proveer la capa de enlace de datos a la capa de red es el servicio orientado a 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 se recibirá exactamente una vez y que todas las tramas se recibirán en el orden correcto. Así, el servicio orientado a conexión ofrece a los procesos de la capa de red el equivalente a un flujo de bits confiable. Es apropiado usarlo en enlaces largos y no confiables, como un canal de satélite o un circuito telefónico de larga distancia. Si se utilizara el servicio no orientado a conexión con confirmación de recepción, es posible que las confirmaciones de recepción perdidas ocasionaran que una trama se enviara y recibiera varias veces, desperdimando ancho de banda.

Cuando se utiliza un servicio orientado a conexión, las transferencias pasan por 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 se recibieron y las que no. En la segunda fase se transmiten una o más tramas. En la tercera y última fase, la conexión se libera al igual que las variables, los búferes y otros recursos utilizados para mantener la conexión.

Entramado

Para proveer servicio a la capa de red, la capa de enlace de datos debe usar el servicio que la capa física le proporciona. Lo que hace la capa física es aceptar un flujo de bits puros y tratar de entregarlo al destino. Si el canal es ruidoso, como en la mayoría de los enlaces inalámbricos y en algunos alámbricos, la capa física agregará cierta redundancia a sus señales para reducir la tasa de error de bits a un nivel tolerable. Sin embargo, no se garantiza que el flujo de bits recibido por la capa de enlace de datos esté libre de errores. Algunos bits pueden tener distintos valores y la cantidad de bits recibidos puede ser menor, igual o mayor que la cantidad de bits transmitidos. 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 discretas, calcule un *token* corto conocido como suma de verificación para cada trama, e incluya esa suma de verificación en la trama al momento de transmitirla. 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 las medidas necesarias para manejarlo (por ejemplo, desecha la trama errónea y es posible que también devuelva un informe de error).

Es más difícil dividir el flujo de bits en tramas de lo que parece a simple vista. Un buen diseño debe facilitar a un receptor el proceso de encontrar el inicio de las nuevas tramas al tiempo que utiliza una pequeña parte del ancho de banda del canal. En esta sección veremos cuatro métodos:

1. Conteo de bytes.
2. Bytes bandera con relleno de bytes.
3. Bits bandera 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 bytes en la trama. Cuando la capa de enlace de datos del destino ve el conteo de bytes, sabe cuántos bytes siguen y, por lo tanto, dónde concluye la trama. Esta técnica se muestra en la Ilustración 26(a) para cuatro tramas pequeñas de ejemplo con 5, 5, 8 y 8 bytes de longitud, respectivamente.

El problema con este algoritmo es que el conteo se puede alterar debido a un error de transmisión. Por ejemplo, si el conteo de bytes de 5 en la segunda trama de la Ilustración 26(b) se vuelve un 7 debido a que cambió un solo bit, el destino perderá la sincronía y entonces será incapaz de localizar el inicio correcto de la siguiente trama. Incluso si el destino sabe que la trama está mal dado que la suma de verificación es incorrecta,

no tiene forma de saber dónde comienza la siguiente trama. Tampoco es útil enviar una trama de vuelta a la fuente para solicitar una retransmisión, ya que el destino no sabe cuántos bytes tiene que saltar para llegar al inicio de la retransmisión. Por esta razón raras veces se utiliza el método de conteo de bytes por sí solo.

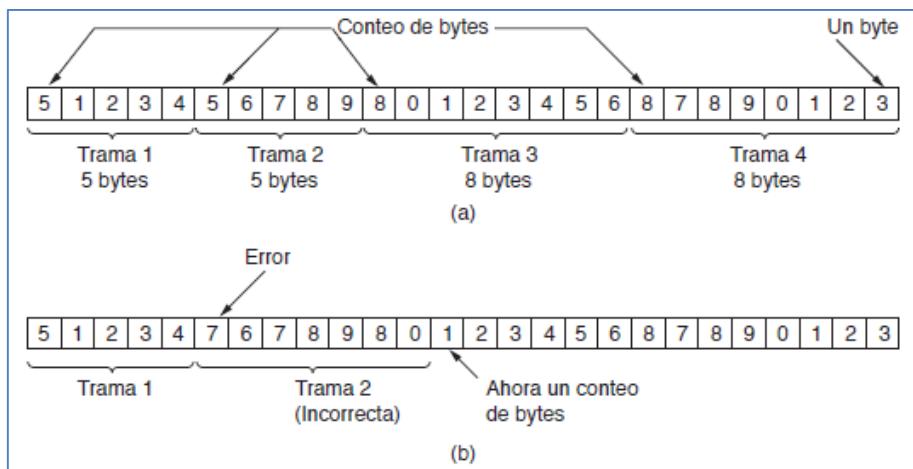


Ilustración 26 - Un flujo de bytes. (a) Sin errores. (b) Con un error.

El segundo método de entramado evita el problema de volver a sincronizar nuevamente después de un error al hacer que cada trama inicie y termine con bytes especiales. Con frecuencia se utiliza el mismo byte, denominado **byte bandera**, como delimitador inicial y final. Este byte se muestra en la Ilustración 27(a) como *FLAG*. Dos bytes bandera consecutivos señalan el final de una trama y el inicio de la siguiente. De esta forma, si el receptor pierde alguna vez la sincronización, todo lo que tiene que hacer es buscar dos bytes bandera para encontrar el fin de la trama actual y el inicio de la siguiente.

Sin embargo, aún queda un problema que tenemos que resolver. Se puede dar el caso de que el byte bandera aparezca en los datos, en especial cuando se transmiten datos binarios como fotografías o canciones. Esta situación interferiría con el entramado. 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 byte bandera “accidental” en los datos. De esta forma es posible diferenciar un byte bandera del entramado de uno en los datos mediante la ausencia o presencia de un byte de escape antes del byte bandera. 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 bytes**.

Ahora bien, la siguiente pregunta es: ¿qué sucede si aparece un byte de escape en medio de los datos? La respuesta es que también se rellena con un byte de escape. En el receptor se quita el primer byte de escape y se deja el byte de datos que le sigue (el cual podría ser otro byte de escape, o incluso el byte bandera). En la Ilustración 27(b) se muestran algunos ejemplos. En todos los casos, la secuencia de bytes que se entrega después de quitar los bytes de relleno es exactamente la misma que la secuencia de bytes original. Así todavía podemos encontrar un límite de trama si buscamos dos bytes bandera seguidos, sin molestarnos por eliminar los escapes.

El esquema de relleno de bytes que se muestra en la Ilustración 27 es una ligera simplificación del esquema empleado en el protocolo **PPP** (Protocolo Punto a Punto, del inglés *Point-to-Point Protocol*), que se utiliza para transmitir paquetes a través de los enlaces de comunicación.

El tercer método de delimitar el flujo de bits resuelve una desventaja del relleno de bytes: que está obligado a usar bytes de 8 bits. También se puede realizar el entramado a nivel de bit, de modo que las tramas puedan contener un número arbitrario de bits compuesto por unidades de cualquier tamaño. Esto se desarrolló para el protocolo **HDLC** (Control de Enlace de Datos de Alto Nivel, del inglés *High-level Data Link Control*), que alguna vez fue muy popular. Cada trama empieza y termina con un patrón de bits especial, 01111110 o *0x7E* en hexadecimal. Este patrón es un byte bandera. Cada vez que la capa de enlace de datos del emisor encuentra cinco bits 1 consecutivos en los datos, inserta automáticamente un 0 como relleno en el flujo de bits de salida. Este relleno de bits es análogo al relleno de bytes, en el cual se inserta un byte de escape en el flujo de

caracteres de salida antes de un byte bandera en los datos. Además asegura una densidad mínima de transiciones que ayudan a la capa física a mantener la sincronización. La tecnología **USB** (Bus Serie Universal, del inglés *Universal Serial Bus*) usa relleno de bits por esta razón.

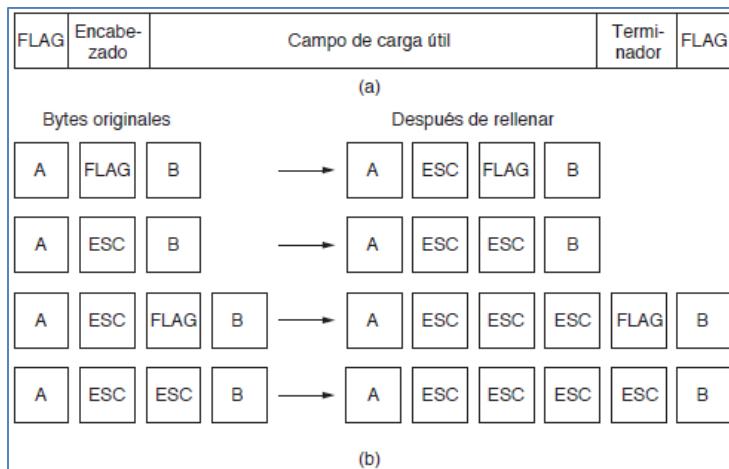


Ilustración 27 - (a) Una trama delimitada por bytes bandera. (b) Cuatro ejemplos de secuencias de bytes antes y después del relleno de bytes.

Cuando el receptor ve cinco bits 1 de entrada consecutivos, seguidos de un bit 0, extrae (es decir, borra) de manera automática el bit 0 de relleno. Así como el relleno de bytes 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 bandera 01111110, éste se transmite como 011111010, pero se almacena en la memoria del receptor como 01111110. En la Ilustración 38 se muestra un ejemplo del relleno de bits.

Con el relleno de bits, el límite entre las dos tramas puede ser reconocido sin ambigüedades mediante el patrón bandera. 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 dentro de los datos.

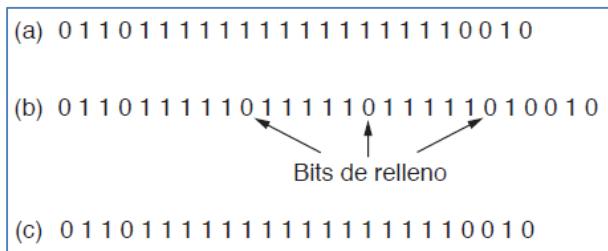


Ilustración 28 - Relleno de bits. (a) Los datos originales. (b) Los datos, según aparecen en la línea. (c) Los datos, como se almacenan en la memoria del receptor después de quitar el relleno.

Un efecto secundario del relleno de bits y de bytes es que la longitud de una trama depende ahora del contenido de los datos que lleva. Por ejemplo, si no hay bytes bandera en los datos, se podrían llevar 100 bytes en una trama de aproximadamente 100 bytes. No obstante, si los datos consisten sólo de bytes bandera, habrá que incluir un byte escape para cada uno de estos bytes y la trama será de cerca de 200 bytes de longitud. Con el relleno de bits, el aumento sería cerca del 12.5%, ya que se agrega 1 bit a cada byte.

El último método de entramado es utilizar un atajo desde la capa física. Ya vimos que la codificación de bits como señales incluye a menudo redundancia para ayudar al receptor. Esta redundancia significa que algunas señales no ocurrirán en los datos regulares. Por ejemplo, en el código de línea 4B/5B se asignan 4 bits de datos a 5 bits de señal para asegurar suficientes transiciones de bits. Esto significa que no se utilizan 16 de las 32 posibles señales. Podemos usar algunas señales reservadas para indicar el inicio y el fin de las tramas. En efecto, estamos usando “*violaciones de código*” para delimitar tramas. La belleza de este esquema es que, como hay señales reservadas, es fácil encontrar el inicio y final de las tramas y no hay necesidad de llenar los datos.

Muchos protocolos de enlace de datos usan una combinación de estos métodos por seguridad. Un patrón común utilizado para Ethernet y 802.11 es hacer que una trama inicie con un patrón bien definido, conocido como **preámbulo**. Este patrón podría ser bastante largo (es común que cuente con 72 bits para 802.11) de modo que el receptor se pueda preparar para un paquete entrante. El preámbulo va seguido de un campo de longitud (cuenta) en el encabezado, que se utiliza para localizar el final de la trama.

Control de errores

Una vez resuelto el problema de marcar el inicio y el fin de cada trama, llegamos al siguiente dilema: cómo asegurar que todas las tramas realmente se entreguen en el orden apropiado a la capa de red del destino. Suponga por un momento que el receptor puede saber si una trama que recibe contiene la información correcta o errónea. Para un servicio sin conexión ni confirmación de recepción sería ideal si el emisor siguiera enviando tramas sin importarle si llegan en forma adecuada. Pero para un servicio confiable orientado a conexión no sería nada bueno.

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 devuelva 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 de manera correcta. Por otra parte, una confirmación de recepción negativa significa que algo falló y que se debe transmitir la trama 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. De manera similar, si se pierde la trama de confirmación de recepción, el emisor no sabrá cómo proceder. Debe quedar claro que en un protocolo en el cual el emisor envía una trama y luego espera una confirmación de recepción, positiva o negativa, éste se quedaría esperando eternamente si se perdiera por completo una trama debido a, por ejemplo, una falla de hardware o un canal de comunicación defectuoso.

Para manejar esta posibilidad se introducen 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 propague de vuelta 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.

No obstante, si la trama o la confirmación de recepción se pierde, el temporizador expirará, alertando al emisor sobre un problema potencial. La solución obvia es simplemente transmitir de nuevo la trama. Sin embargo, aunque éstas pueden transmitirse muchas veces, existe el peligro de que el receptor acepte la misma trama en dos o más ocasiones 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 de salida, con el fin de que el receptor pueda distinguir las retransmisiones de las 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 (y de las capas superiores).

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 cuando el emisor opera en una computadora rápida y el receptor trabaja en una máquina lenta. Una situación común es cuando un teléfono inteligente solicita una página web de un servidor mucho más poderoso, que a su vez dispara el chorro de datos al pobre e indefenso teléfono hasta que queda totalmente saturado. 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 para autorizarle que envíe más datos o por lo menos indicarle 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 la retroalimentación por parte del receptor.

Veremos los esquemas de control de flujo basados en retroalimentación, principalmente debido a que los esquemas basados en tasa sólo se ven como parte de la capa de transporte. Los esquemas basados en retroalimentación se ven tanto en la capa de enlace como en las capas superiores. En realidad, es más común esto último, en cuyo caso el hardware de la capa de enlace se diseña para operar con la rapidez suficiente como para no producir pérdidas. Por ejemplo, se dice algunas veces que las implementaciones de hardware de la capa de enlace como **NIC** (Tarjetas de Interfaz de Red, del inglés *Network Interface Cards*) operan a “*velocidad de alambre*”, lo cual significa que pueden manejar las tramas con la misma rapidez con que pueden llegar al enlace. De esta forma, los excesos no son problema del enlace, por lo que se manejan en las capas superiores.

Se conocen varios esquemas de control de flujo basados en retroalimentación, pero la mayoría se basa en el mismo principio. El protocolo contiene reglas bien definidas respecto al momento en que un emisor puede transmitir la siguiente trama. Con frecuencia estas reglas prohíben el envío de tramas hasta que el receptor lo autorice, ya sea en forma implícita o explícita. 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”.

Detección y corrección de errores

Hemos visto que los canales de comunicación tienen una variedad de características. Algunos de ellos, como la fibra óptica en las redes de telecomunicaciones, tienen tasas de error pequeñas de modo que los errores de transmisión son una rara ocurrencia. Pero otros canales, en especial los enlaces inalámbricos y los viejos lazos locales, tienen tasas de error más grandes. Para estos enlaces, los errores de transmisión son la norma. No se pueden evitar a un costo razonable en términos de rendimiento. La conclusión es que los errores de transmisión están aquí para quedarse. Tenemos que aprender a lidiar con ellos.

Los diseñadores de redes han desarrollado dos estrategias básicas para manejar los errores. Ambas añaden información redundante a los datos que se envían. Una es incluir suficiente información redundante para que el receptor pueda deducir cuáles debieron ser los datos transmitidos. 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 por lo regular se conoce como **FEC** (Corrección de Errores hacia Adelante, del inglés *Forward Error Correction*).

Cada una de estas técnicas ocupa un nicho diferente. En los canales que son altamente confiables, como los de fibra, es más económico utilizar un código de detección de errores y sólo 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 que se transmitió. Los FEC se utilizan en canales ruidosos puesto que las retransmisiones tienen la misma probabilidad de ser tan erróneas como la primera transmisión.

Una consideración clave para estos códigos es el tipo de errores que pueden llegar a ocurrir. Ni los códigos de corrección de errores ni los de detección de errores pueden manejar todos los posibles errores, puesto que los bits redundantes que ofrecen protección tienen la misma probabilidad de ser recibidos con errores que los bits de datos (lo cual puede comprometer su protección). Sería ideal que el canal tratara a los bits redundantes de una manera distinta a los bits de datos, pero no es así. Para el canal todos son sólo bits. Esto significa que para evitar errores no detectados, el código debe ser lo bastante robusto como para manejar los errores esperados.

Un modelo es que los errores son producidos por valores extremos de ruido térmico que saturan la señal breve y ocasionalmente, lo cual produce errores aislados de un solo bit. Otro modelo es que los errores tienden a

producirse en ráfagas en vez de hacerlo en forma individual. Este modelo se deriva de los procesos físicos que los generan (como un desvanecimiento pronunciado en un canal inalámbrico o una interferencia eléctrica transitoria en un canal alámbrico).

Ambos modelos importan en la práctica, además de que tienen distintas concesiones. El hecho de que los errores lleguen en ráfagas tiene tanto ventajas como desventajas en comparación con los errores aislados de un solo bit. Por el lado positivo, los datos de computadora siempre se envían en bloques de bits. Suponga que el tamaño de bloque es de 1000 bits y que la tasa de error es de 0.001 por bit. Si los errores fueran independientes, la mayoría de los bloques contendría un error. Pero si los errores llegaran en ráfagas de 100, sólo un bloque de esos 100 sería afectado en promedio. La desventaja de los errores en ráfaga es que cuando ocurren son mucho más difíciles de corregir que los errores aislados.

También existen otros tipos de errores. Algunas veces se conocerá la ubicación de un error, tal vez debido a que la capa física recibió una señal analógica que estaba muy alejada del valor esperado para un 0 o un 1 y declaró el bit como perdido. A esta situación se le conoce como **canal de borrado**. Es más fácil corregir los errores en los canales de borrado que en canales que voltean bits, ya que incluso si se pierde el valor del bit, por lo menos sabemos cuál tiene el error. Sin embargo, algunas veces no tenemos el beneficio de los canales de borrado.

A continuación examinaremos los códigos de corrección de errores y los códigos de detección de errores. Pero debe tener en cuenta dos puntos. Primero, cubrimos estos códigos en la capa de enlace debido a que es el primer lugar en el que nos topamos con el problema de transmitir grupos de bits de manera confiable. Sin embargo, los códigos se utilizan ampliamente debido a que la confiabilidad es una preocupación general. Los códigos de corrección de errores se ven también en la capa física, en especial con los canales ruidosos, y en capas más altas, en especial con los medios de tiempo real y la distribución de contenido. Los códigos de detección de errores se utilizan con frecuencia en las capas de enlace, red y transporte.

El segundo punto a considerar es que los códigos de error son matemáticas aplicadas. A menos que usted sea muy adepto a los campos de Galois o a las propiedades de las matrices dispersas, es más conveniente que obtenga códigos con buenas propiedades de una fuente confiable en vez de crear sus propios códigos. De hecho, esto es lo que hacen muchos estándares de protocolos, en donde los mismos códigos se utilizan una y otra vez. En el material que veremos a continuación, estudiaremos con detalle un código simple y después describiremos brevemente los códigos avanzados. De esta forma podremos comprender las concesiones a través del código simple y hablaremos sobre los códigos que se utilizan en la práctica a través de los códigos avanzados.

Códigos de corrección de errores

Analizaremos cuatro códigos de corrección de errores:

1. Códigos de Hamming.
2. Códigos convolucionales binarios.
3. Códigos de Reed-Solomon.
4. Códigos de verificación de paridad de baja densidad.

Todos estos códigos agregan redundancia a la información que se envía. Una trama consiste en m bits de datos (mensaje) y r bits redundantes (verificación). En un **código de bloque**, los r bits de verificación se calculan únicamente en función de los m bits de datos con los que se asocian, como si los m bits se buscaran en una gran tabla para encontrar sus correspondientes r bits de verificación. En un **código sistemático**, los m bits de datos se envían directamente, junto con los bits de verificación, en vez de que se codifiquen por sí mismos antes de enviarlos. En un **código lineal**, los r bits de verificación se calculan como una función lineal de los m bits de datos. El OR exclusivo (*XOR*) o la suma de módulo 2 es una elección popular. Esto significa que la codificación se puede llevar a cabo con operaciones como multiplicaciones de matrices o circuitos lógicos simples. Los códigos que analizaremos en esta sección son códigos de bloque lineales sistemáticos, a menos que se indique otra cosa.

Sea la longitud total de un bloque n (es decir, $n=m+r$). Describiremos esto como un código (n, m) . Una unidad de n bits que contiene bits de datos y de verificación se conoce como **palabra codificada** de n bits. La **tasa de**

código, o simplemente tasa, es la fracción de la palabra codificada que lleva información no redundante, o m/n . Las tasas que se utilizan en la práctica varían mucho. Podrían ser $1/2$ para un canal ruidoso, en cuyo caso la mitad de la información recibida es redundante, o podrían estar cerca de 1 para un canal de alta calidad, en donde sólo se agrega un pequeño número de bits de verificación a un mensaje extenso.

Para entender la manera en que pueden manejarse los errores, es necesario estudiar de cerca lo que es en realidad un error. Dadas dos palabras codificadas cualesquiera que se pueden transmitir o recibir, digamos 10001001 y 10110001 , es posible determinar cuántos bits correspondientes difieren. En este caso, difieren 3 bits. Para determinar la cantidad de bits diferentes, basta aplicar un *XOR* a las dos palabras codificadas (se compara bit a bit cada palabra, si son iguales el resultado es 0, si son distintos 1) 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 **distancia 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 solo bit para convertir una en la otra.

Dado el algoritmo para calcular los bits de verificación, es posible construir una lista completa de las palabras codificadas válidas, y a partir de esta lista se pueden encontrar las dos palabras codificadas con la menor distancia de Hamming. Esta distancia es la distancia de Hamming del código completo. En la mayoría de las aplicaciones de transmisión de datos, todos los 2^m mensajes de datos posibles son válidos, pero debido a la manera en que se calculan los bits de verificación no se usan todas las 2^n palabras codificadas posibles. De hecho, cuando hay r bits de verificación sólo la pequeña fracción de $2^m/2^n$ o $1/2^r$ de los posibles mensajes serán palabras codificadas válidas. Esta dispersión con la que se incrusta el mensaje en el espacio de las palabras codificadas es la que permite que el receptor detecte y corrija los errores.

Las propiedades de detección y corrección de errores de un código de bloque dependen de su distancia de Hamming. Para detectar d errores de manera confiable 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 inválida, 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 válidas están tan separadas que, aun con d cambios, la palabra codificada original sigue estando más cercana que cualquier otra. Esto significa que la palabra codificada original se puede determinar en forma única con base en la suposición de que es menos probable un mayor número de errores.

Como ejemplo sencillo de un 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 cual significa que puede corregir errores dobles o detectar errores cuádruples. Si llega la palabra codificada 0000000111 y esperamos sólo errores de uno o dos bits, el receptor sabrá que la palabra original debió haber sido 0000011111. Pero si un error triple cambia 0000000000 a 0000000111, el error no se corregirá en forma apropiada. Por otro lado, si esperamos todos estos errores, podremos detectarlos. Ninguna de las palabras codificadas recibidas es válida, por lo que debe haber ocurrido un error. Debe ser aparente que en este ejemplo no podemos corregir errores dobles y detectar al mismo tiempo errores cuádruples, ya que tendríamos que interpretar de dos maneras distintas una palabra codificada recibida.

En nuestro ejemplo, la tarea de decodificar mediante el proceso de buscar la palabra codificada válida que se asemeje más a la palabra codificada recibida se puede llevar a cabo mediante inspección. Por desgracia, en el caso más general en donde hay que evaluar todas las palabras codificadas como candidatas, esta tarea puede requerir de mucho tiempo. Como alternativa se han diseñado códigos prácticos que nos permiten usar atajos para buscar cuál tendría más probabilidades de ser la palabra codificada original.

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 válidos tiene n palabras codificadas inválidas a una distancia 1 de él. Éstas se forman invirtiendo de manera sistemática cada uno de los n bits de la palabra codificada de n bits que la conforman. Por lo tanto, cada uno de los 2^m mensajes válidos requiere $n+1$ patrones de bits dedicados a él. Dado que la cantidad total de patrones de bits es 2^n , debemos tener $(n+1)2^m \leq 2^n$. Si usamos $n+m+r$, este requisito se vuelve

$$(m + r + 1) \leq 2^n$$

Dado el valor de 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 mediante el uso de un método desarrollado por Hamming (1950). En los **códigos de Hamming**, 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, etc. Los bits que son potencias de 2 (1, 2, 4, 8, 16, etc.) son bits de verificación. El resto (3, 5, 6, 7, 9, etc.) se llenan con los m bits de datos. El patrón se muestra para un código de Hamming (11,7) con 7 bits de datos y 4 bits de verificación en la Ilustración 29. Cada bit de verificación obliga a que la suma módulo 2, o paridad de un grupo de bits, incluyéndolo a él mismo, sea par (o impar). Un bit puede estar incluido en varios cálculos de bits de verificación. 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 $29=1+4+8+16$. Un bit es verificado solamente por los bits de verificación que ocurren en su expansión (por ejemplo, el bit 11 es verificado por los bits 1, 2 y 8). En el ejemplo se calculan los bits de verificación para las sumas con paridad par de un mensaje que contiene la letra "A" del código ASCII.

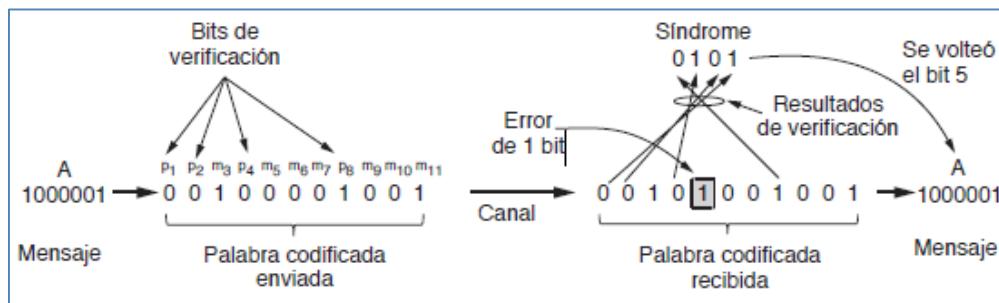


Ilustración 29 - Ejemplo de un código de Hamming (11,7) para corregir errores de un solo bit.

Esta construcción produce un código con una distancia de Hamming de 3, lo cual significa que puede corregir errores individuales (o detectar errores dobles). La razón de enumerar con mucho cuidado el mensaje y los bits de verificación se vuelve aparente en el proceso de decodificación. Cuando llega una palabra codificada, el receptor vuelve a realizar los cálculos del bit de verificación, incluyendo los valores de los bits de verificación recibidos. A estos les llamamos resultados de verificación. Si los bits de verificación están correctos, entonces cada resultado de verificación debe ser cero para las sumas de paridad par. En este caso la palabra codificada se acepta como válida.

Pero si no todos los resultados de verificación son cero, entonces se ha detectado un error. El conjunto de resultados de verificación forma el **síndrome de error** que se utiliza para señalarlo y corregirlo el error. En la Ilustración 29 ocurrió un error de un solo bit en el canal, de modo que los resultados de verificación son 0, 1, 0 y 1 para $k=8, 4, 2$ y 1, respectivamente. Esto nos da un síndrome de 0101, o $4+1=5$. Según el diseño del esquema, esto significa que el quinto bit es un error. Al voltear el bit incorrecto (que podría ser un bit de verificación o uno de datos) y desechar los bits de verificación obtenemos el mensaje correcto de una letra "A" en código ASCII.

Las distancias de Hamming son valiosas para comprender los códigos de bloque, y los códigos de Hamming se utilizan en la memoria de corrección de errores. Sin embargo, la mayoría de las redes utilizan códigos más robustos. El segundo código que veremos es el **código convolucional**. Este código es el único que analizaremos que no es código de bloque. En un código convolucional, un codificador procesa una secuencia de bits de entrada y genera una secuencia de bits de salida. No hay un tamaño de mensaje natural, o límite de

codificación, como en un código de bloque. La salida depende de los bits de entrada actual y previa. Es decir, el codificador tiene memoria. El número de bits previos de los que depende la salida se denomina **longitud de restricción** del código. Los códigos convolucionales se especifican en términos de su tasa de transmisión y su longitud de restricción.

Los códigos convolucionales se utilizan mucho en las redes implementadas; por ejemplo, como parte del sistema de telefonía móvil GSM, en las comunicaciones de satélite y en 802.11. Como ejemplo, en la Ilustración 30 se muestra un código convolucional popular. Este código se conoce como código convolucional NASA de $r=1/2$ y $k=7$, ya que se utilizó por primera vez en las misiones espaciales del Voyager a partir de 1977. Desde entonces se ha reutilizado libremente, por ejemplo, como parte de las redes 802.11.

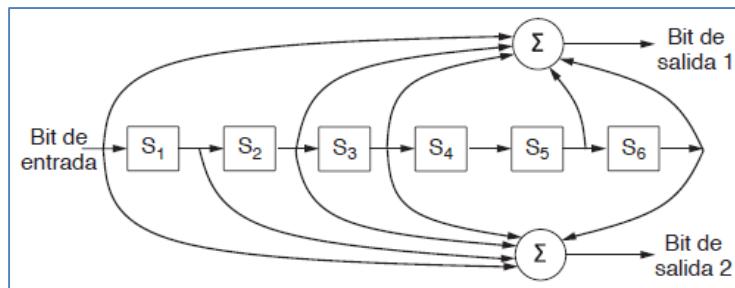


Ilustración 30 - El código convolucional binario NASA utilizado en redes 802.11.

En la Ilustración 30, cada bit de entrada del lado izquierdo produce dos bits de salida del lado derecho, los cuales son las sumas XOR de la entrada y el estado interno. Ya que se trata con bits y se realizan operaciones lineales, es un código convolucional binario lineal. Puesto que 1 bit de entrada produce 2 bits de salida, la tasa de código es de 1/2. No es sistemático, ya que ninguno de los bits de salida es simplemente el bit de entrada.

El estado interno se mantiene en seis registros de memoria. Cada vez que se introduce otro bit, los valores de los registros se desplazan a la derecha. Por ejemplo, si se introduce 111 como entrada y el estado inicial está compuesto sólo de ceros, entonces el estado interno (que se escribe de izquierda a derecha) se convertirá en 100000, 110000 y 111000 después de haber introducido el primer, segundo y tercer bits, respectivamente. Los bits de salida serán 11, seguidos de 10 y después de 01. Se requieren siete desplazamientos para vaciar una entrada por completo, de modo que no afecte la salida. Por lo tanto, la longitud de restricción de este código es $k=7$.

Para decodificar un código convolucional es necesario buscar la secuencia de bits de entrada que tenga la mayor probabilidad de haber producido la secuencia observada de bits de salida (incluyendo los errores). Para valores pequeños de k esto se hace mediante un algoritmo muy popular desarrollado por Viterbi (*Forney, 1973*). El algoritmo recorre la secuencia observada y guarda para cada paso y cada posible estado interno la secuencia de entrada que hubiera producido la secuencia observada con la menor cantidad posible de errores. Al final, la secuencia de entrada que requiera la menor cantidad de errores es el mensaje más probable.

Los códigos convolucionales han sido populares en la práctica, ya que es fácil ignorar la incertidumbre de que un bit sea 0 o 1 en la decodificación. Por ejemplo, suponga que -1V es el nivel 0 lógico y +1V es el nivel 1 lógico, y que podríamos recibir 0.9 V y -0.1V para 2 bits. En vez de asignar estas señales a 1 y 0 de inmediato, podría ser conveniente considerar que 0.9V “*muy probablemente sea un 1*” y que -0.1 V “*puede ser un 0*”, y corregir la secuencia en su totalidad. Las extensiones del algoritmo de Viterbi pueden trabajar con estas incertidumbres para ofrecer una corrección de errores más poderosa. Este método de trabajar con la incertidumbre de un bit se conoce como **decodificación de decisión suave**. Por el contrario, decidir si cada bit es un 0 o un 1 antes de la subsiguiente corrección de errores se conoce como **decodificación de decisión dura**.

El tercer tipo de código de corrección de errores que describiremos es el **código de Reed-Solomon**. Al igual que los códigos de Hamming, los códigos de Reed-Solomon son códigos de bloques lineales y con frecuencia también son sistemáticos. A diferencia de los códigos de Hamming, que operan sobre bits individuales, los códigos de Reed-Solomon operan sobre símbolos de m bits. Naturalmente las matemáticas son más complejas, por lo que describiremos su operación mediante una analogía.

Los códigos de Reed-Solomon se basan en el hecho de que todo polinomio de n grados se determina en forma única mediante $n+1$ puntos. Por ejemplo, una línea con la forma $ax+b$ se determina mediante dos puntos. Los puntos extra en la misma línea son redundantes, lo cual es útil para la corrección de errores. Imagine que tenemos dos puntos de datos que representan una línea y que enviamos esos dos puntos de datos junto con dos puntos de verificación seleccionados sobre la misma línea. Si uno de los puntos se recibe con error, de todas formas podemos recuperar los puntos de datos si ajustamos una línea a los puntos recibidos. Tres de los puntos estarán en la línea y el otro punto (el del error) no. Al encontrar la línea hemos corregido el error.

En realidad los códigos de Reed-Solomon se definen como polinomios que operan sobre campos finitos, pero trabajan de una manera similar. Para símbolos de m bits, las palabras codificadas son de 2^m+1 símbolos de longitud. Una elección popular es hacer a $m=8$, de modo que los símbolos sean bytes. Así, una palabra codificada tiene una longitud de 256 bytes.

Los códigos de Reed-Solomon se utilizan mucho en la práctica debido a sus buenas propiedades de corrección de errores, en especial para los errores de ráfagas. Puesto que se basan en símbolos de m bits, tanto un error de un solo bit como un error de ráfaga de m bits se tratan simplemente como error de un símbolo. Cuando se agregan $2t$ símbolos redundantes, un código de Reed-Solomon es capaz de corregir hasta t errores en cualquiera de los símbolos transmitidos. Esto significa que, por ejemplo, el código (255, 233) que tiene 32 símbolos redundantes puede corregir errores de hasta 16 símbolos. Como los símbolos pueden ser consecutivos y cada uno de ellos es de 8 bits, se puede corregir una ráfaga de errores de hasta 128 bits. La situación es aún mejor si el modelo de error es el de borrado (por ejemplo, una rayadura en un CD que borra algunos símbolos). En este caso se pueden corregir hasta $2t$ errores.

A menudo los códigos de Reed-Solomon se utilizan en combinación con otros códigos, como el convolucional. El razonamiento es el siguiente. Los códigos convolucionales son efectivos a la hora de manejar errores de bits aislados, pero es probable que fallen con una ráfaga de errores, si hay demasiados errores en el flujo de bits recibido. Al agregar un código de Reed-Solomon dentro del código convolucional, la decodificación de Reed-Solomon puede limpiar las ráfagas de errores, una tarea que realiza con mucha eficiencia. Así, el código en general provee una buena protección contra los errores individuales y los errores de ráfaga.

El último código de corrección de errores que estudiaremos es el código LDPC (Verificación de Paridad de Baja Densidad, del inglés *Low-Density Parity Check*). Los códigos LDPC son códigos de bloques lineales inventados por Robert Gallager en su tesis para doctorado (Gallager, 1962). Al igual que la mayoría de las tesis, estos códigos pronto fueron olvidados, y no fue sino hasta 1995 que se reinventaron gracias a los avances en el poder de las computadoras, pues ya era práctico utilizarlos.

En un código LDPC, cada bit de salida se forma sólo a partir de una fracción de los bits de entrada. Esto conduce a una representación matricial del código con una densidad baja de 1s, razón por la cual tiene ese nombre. Las palabras codificadas recibidas se decodifican con un algoritmo de aproximación que mejora de manera reiterativa con base en el mejor ajuste de los datos recibidos con una palabra codificada válida. Esto corrige los errores.

Los códigos LDPC son prácticos para tamaños grandes de bloques y tienen excelentes habilidades de corrección de errores que superan a las de muchos otros códigos (incluyendo los que vimos antes) en la práctica. Por esta razón se están incluyendo rápidamente en los nuevos protocolos. Forman parte del estándar para la difusión de video digital, la Ethernet de 10 Gbps, las redes de líneas eléctricas y la versión más reciente de 802.11.

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 si se les compara con la fibra óptica. Sin los códigos de corrección de errores sería difícil hacer pasar cualquier cosa. Sin embargo, a través de la fibra óptica o del cable de cobre de alta calidad, la tasa de error es mucho más baja, por lo que la detección de errores y la retransmisión por lo general son más eficientes para manejar un error ocasional.

En esta sección examinaremos tres códigos de detección de errores distintos. Todos son códigos de bloques sistemáticos lineales:

1. Paridad.
2. Sumas de verificación.
3. Pruebas de Redundancia Cíclica (CRC).

Para ver cómo pueden ser más eficientes que los códigos de corrección de errores, considere el primer código de detección de errores en el que se adjunta un solo **bit de paridad** a los datos. El bit de paridad se elige de manera que el número de bits 1 en la palabra codificada sea par (o impar). Hacer esto es equivalente a calcular el bit de paridad (par) como la suma módulo 2 o el resultado de un XOR en los bits de datos. Por ejemplo, cuando se envía la secuencia 1011010 en paridad par, se agrega un bit al final para convertirla en 10110100. Con paridad impar, 1011010 se convierte en 10110101. Un código con un solo bit de paridad tiene una distancia de 2, ya que cualquier error de un solo bit produce una palabra codificada con la paridad incorrecta. Esto significa que puede detectar errores de un solo bit.

Considere un canal en el que los errores son aislados y la tasa de error es de 10^{-6} por bit. Ésta puede parecer una tasa de error pequeña, pero a lo más es una tasa equitativa para un cable de cobre extenso que desafía a la detección de errores. Los enlaces de LAN comunes proveen tasas de error de bits de 10^{-10} . Sea el tamaño de bloque 1000 bits. Para proporcionar corrección de errores en bloques de 1000 bits, sabemos que se requieren 10 bits de verificación. Así, un megabit de datos requeriría 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 encontrará un bloque con error y se tendrá que transmitir un bloque extra (1001 bits) para reparar ese error. La sobrecarga total del método de detección de errores y retransmisión es de sólo 2001 bits por megabit de datos, en comparación con los 10000 bits en un código de Hamming.

Un problema con este esquema es que un bit de paridad sólo puede detectar de manera confiable un error de un solo bit en el bloque. Si el bloque está muy confuso debido a una ráfaga de errores larga, la probabilidad de detectar ese error es sólo de 0.5, lo cual es difícilmente aceptable. Es posible aumentar la probabilidad de manera considerable si cada bloque a enviar se trata como una matriz rectangular de n bits de ancho por k bits de alto. Ahora, si calculamos y enviamos un bit de paridad para cada fila, se detectarán de manera confiable errores de hasta k bits siempre y cuando haya a lo mucho un error por fila.

Pero hay algo más que podemos hacer para ofrecer una mejor protección contra los errores: calcular los bits de paridad sobre los datos en un orden distinto al que se transmiten los bits de datos. A este proceso se le denomina **intercalado**. En este caso, calculamos un bit de paridad para cada una de las n columnas y enviamos todos los bits de datos como k filas; para ello enviamos las filas de arriba hacia abajo y los bits en cada fila de izquierda a derecha de la manera usual. En la última fila enviamos los n bits de paridad. Este orden de transmisión se muestra en la Ilustración 31 para $n=7$ y $k=7$.

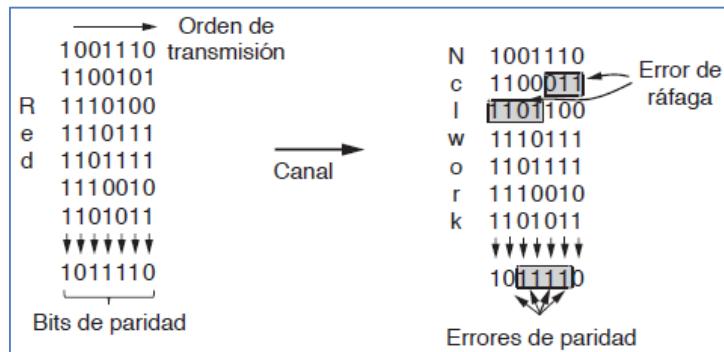


Ilustración 31 - Intercalado de bits de paridad para detectar un error de ráfaga.

El intercalado es una técnica general para convertir un código que detecta (o corrige) los errores aislados en uno que detecta (o corrige) los errores de ráfaga. Cuando en la Ilustración 31 ocurre un error de ráfaga de longitud $n=7$, los bits con error se espacian a través de distintas columnas (un error de ráfaga no implica que todos esos bits estén mal; sólo implica que por lo menos el primero y el último están mal). En la Ilustración 31, se voltearon 4 bits en un rango de 7 bits). Por lo menos 1 bit en cada una de las n columnas se verá afectado,

de modo que los bits de paridad en esas columnas detectarán el error. Este método usa n bits de paridad en bloques de kn bits de datos para detectar un solo error de ráfaga de longitud n o menor.

Sin embargo, una ráfaga de longitud $n+1$ pasará sin ser detectada si el primer bit está invertido, el último bit está invertido y todos los demás bits son correctos. 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} .

El segundo tipo de código de detección de errores, la **suma de verificación**, está muy relacionado con los grupos de bits de paridad. La palabra “*suma de verificación*” se utiliza con frecuencia para indicar un grupo de bits de verificación asociados con un mensaje, sin importar cómo se calculen. Un grupo de bits de paridad es un ejemplo de una suma de verificación. Sin embargo, hay otras sumas de verificación más poderosas basadas en la suma acumulada de los bits de datos del mensaje. Por lo general la suma de verificación se coloca al final del mensaje, como el complemento de la función de suma. De esta forma, los errores se pueden detectar al sumar toda la palabra codificada recibida, tanto los bits de datos como la suma de verificación. Si el resultado es cero, no se ha detectado ningún error.

Un ejemplo es la suma de verificación de Internet de 16 bits que se utiliza en todos los paquetes de Internet como parte del protocolo IP (Braden y colaboradores, 1988). Esta suma de verificación es una suma de los bits del mensaje divididos en palabras de 16 bits. Como este método opera sobre palabras en vez de bits, como en la paridad, los errores que no modifican la paridad de todas formas pueden alterar la suma y ser detectados. Por ejemplo, si el bit de menor orden en dos palabras distintas se volteá de 0 a 1, una verificación de paridad a lo largo de estos bits no podría detectar un error. Sin embargo, se agregarán dos 1s a la suma de verificación de 16 bits para producir un resultado distinto. Entonces se podrá detectar el error.

La suma de verificación de Internet se calcula en aritmética de complemento a uno, en vez de la suma módulo 2^{16} . En aritmética de complemento a uno, un número negativo es el complemento a nivel de bits de su contraparte positiva. La mayoría de las computadoras modernas emplean aritmética de complemento a dos, en donde un número negativo es el complemento a uno más uno. En una computadora con complemento a dos, la suma de complemento a uno es equivalente a obtener la suma módulo 2^{16} y añadir cualquier desbordamiento de los bits de mayor orden de vuelta a los bits de menor orden. Este algoritmo nos da una cobertura más uniforme de los datos mediante los bits de suma de verificación. En caso contrario, se podrían sumar dos bits de mayor orden, desbordarse y perderse sin cambiar la suma. También hay otro beneficio. El complemento a uno tiene dos representaciones de cero, todos los bits en 0 y todos los bits en 1. Esto permite un valor (por ejemplo, todos los bits en 0) para indicar que no hay suma de verificación sin necesidad de otro campo.

En especial, la suma de verificación de Internet es eficiente y simple, pero ofrece una protección débil en algunos casos, precisamente debido a que es una suma simple. No detecta la eliminación o adición de datos cero, ni el intercambio de partes del mensaje y ofrece una protección débil contra los empalmes de mensajes, en donde se juntan partes de dos paquetes. Estos errores pueden parecer muy poco probables de ocurrir mediante procesos aleatorios, pero son justo el tipo de errores que pueden ocurrir con hardware defectuoso.

Hay una mejor opción: la **suma de verificación de Fletcher** (Fletcher, 1982). Ésta incluye un componente posicional, en donde se suma el producto de los datos y su posición con la suma acumulada. Este método ofrece una detección más robusta de los cambios en la posición de los datos.

Aunque los dos esquemas anteriores pueden ser adecuados algunas veces en capas superiores, en la práctica se utiliza mucho un tercer tipo de código de detección de errores más útil en la capa de enlace: el **CRC** (Comprobación de Redundancia Cíclica, del inglés *Cyclic Redundancy Check*), también conocido como **código polinomial**. 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: $1x^5+1x^4+0x^3+0x^2+0x^1+1x^0$.

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:

$$\begin{array}{r}
 10011011 & 00110011 & 11110000 & 01010101 \\
 + 11001010 & + 11001101 & - 10100110 & - 10101111 \\
 \hline
 01010001 & 11111110 & 01010110 & 11111010
 \end{array}$$

La división larga 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 el CRC 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 un CRC 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 la suma de verificación, intenta dividirla entre $G(x)$. Si hay un residuo, ha ocurrido un error de transmisión.

El algoritmo para calcular el CRC 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 $x^rM(x)$.
2. Divida la cadena de bits correspondiente a $G(x)$ entre la correspondiente a $x^rM(x)$; usando una división módulo 2.
3. Reste el residuo (que siempre es de r o menos bits) a la cadena de bits correspondiente a $x^rM(x)$; usando una resta módulo 2. El resultado es la trama con suma de verificación que va a transmitirse. Llame a su polinomio $T(x)$.

La Ilustración 32 ilustra el cálculo para una trama 1101011111 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 210278 entre 10941, el residuo es 2399. Si se resta 2399 a 210278, lo que queda (207879) es divisible entre 10941, por lo que el residuo será cero.

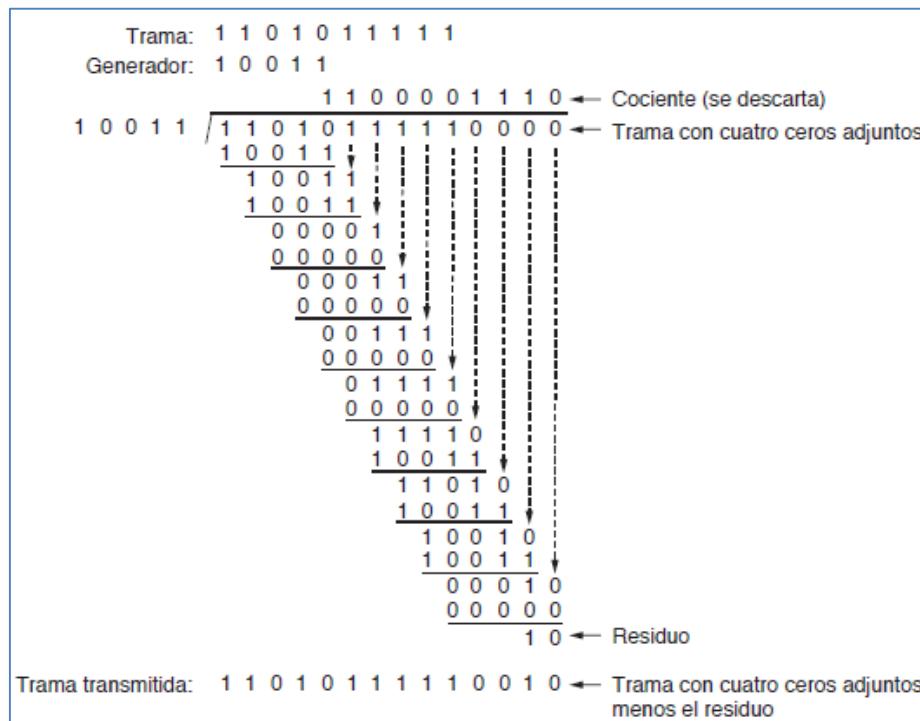


Ilustración 32 - Ejemplo del cálculo de CRC.

Ahora analizaremos el alcance de este método. ¿Qué tipos de errores 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.

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 detectados.

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 $E(x)=x^i(x^{i-j}+1)$. 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 32768.

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. Al hacer de $x+1$ un factor de $G(x)$, podemos detectar todos los errores con un número impar de bits invertidos.

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 se puede representar 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 a 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 podrá 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 $G(x)$. Segundo 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 podemos demostrar que cuando ocurre una ráfaga de errores mayor que $r+1$ bits, 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 tengan la misma probabilidad.

Ciertos polinomios se han vuelto estándares internacionales. El que se utiliza en el IEEE 802 se basa en el ejemplo de Ethernet y 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, detecta todas las ráfagas con una longitud de 32 o menor y todas las ráfagas que afecten a un número impar de bits. Se ha utilizado en muchas partes desde la década de 1980. Sin embargo, esto no significa que sea la mejor opción. Mediante el uso de una búsqueda computacional exhaustiva, Castagnoli y colaboradores (1993), junto con Koopman (2002), encontraron los mejores CRC. Estos CRC tienen una distancia de Hamming de 6 para los tamaños de mensajes típicos, mientras que el estándar de IEEE CRC-32 tiene una distancia de Hamming de sólo 4.

Aunque el cálculo requerido para obtener el CRC puede parecer complicado, es fácil calcular y verificar CRC en el hardware mediante circuitos simples de registros de desplazamiento (Peterson y Brown, 1961). En la práctica, casi siempre se usa este hardware. La mayoría de los estándares de red incluyen varios CRC, entre los cuales están casi todas las redes LAN (por ejemplo, Ethernet, 802.11) y los enlaces punto a punto.

Protocolos elementales de enlace de datos

Para introducir el tema de los protocolos, comenzaremos por estudiar tres protocolos de complejidad creciente.

Para comenzar, supongamos que las capas física, de enlace de datos y de red son procesos independientes que se comunican pasando mensajes de un lado a otro. En la Ilustración 33 se muestra una implementación común. El proceso de la capa física y una parte del proceso de la capa de enlace de datos se ejecutan en hardware dedicado, conocido como **NIC** (Tarjeta de Interfaz de Red, del inglés *Network Interface Card*). El resto del proceso de la capa de enlace y el proceso de la capa de red se ejecutan en la CPU principal como parte del sistema operativo, en donde el software para el proceso de la capa de enlace a menudo toma la forma de un **controlador de dispositivo**. Sin embargo, también puede haber otras implementaciones (por ejemplo, tres procesos descargados a un dispositivo de hardware dedicado, conocido como **acelerador de red**, o tres procesos ejecutándose en la CPU principal en un radio definido por software). En realidad, la implementación preferida cambia de una década a otra con las concesiones tecnológicas. 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 enfatizar la independencia de las capas.

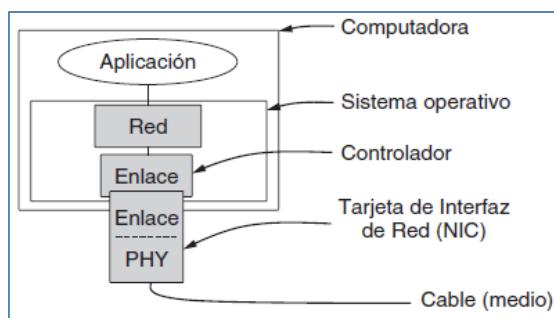


Ilustración 33 - Implementación de las capas física, de enlace de datos y de red.

Otro supuesto clave es que la máquina A desea mandar un flujo considerable de datos a la máquina B mediante el uso de un servicio confiable orientado a 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 éstos se produzcan, sino que cuando la capa de enlace de datos de A los solicita, la capa de red siempre es capaz de proporcionarlos de inmediato (posteriormente también

descartaremos esta restricción). Por otro lado, 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 recibe a través de la interfaz desde la capa de red sólo es de datos, los cuales 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. Por lo tanto, una trama consiste en un paquete incrustado, con 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 los procedimientos de biblioteca adecuados *to_physical_layer* para enviar una trama y *from_physical_layer* para recibir una trama. Estos procedimientos calculan y anexan o verifican la suma de verificación (que por lo general se realiza en el hardware) de modo que no tengamos qué preocuparnos por ella como parte de los protocolos que desarrollaremos en esta sección. Podrían usar el algoritmo de CRC que vimos en la sección anterior, por ejemplo.

En un principio, el receptor no tiene nada que hacer. Sólo está esperando a que ocurra algo. En los protocolos de ejemplo de este capítulo indicaremos que la capa de enlace de datos está en espera de que ocurra algo mediante la llamada al 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 conjunto de eventos posibles es distinto para cada uno de los diferentes protocolos que describiremos, y se definirán por separado para cada protocolo. Tenga en cuenta 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 ocasionará que suspenda lo que estaba haciendo y proceda a manejar la trama entrante. Sin embargo, por simplicidad 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 único canal.

Cuando llega una trama al receptor, se vuelve a calcular la suma de verificación. Si la suma de verificación en la trama es incorrecta (es decir, si hubo un error de transmisión), se informa esto 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*) de modo que pueda adquirir la trama para inspeccionarla mediante el uso de *from_physical_layer*. Tan pronto como la capa de enlace de datos receptora adquiere una trama sin daños, verifica 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 una 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. Esto ocurre cada vez que se instala una nueva NIC en una computadora. Al proporcionarse una interfaz rígida entre la capa de red y la de enlace de datos se simplifica en gran medida la tarea de diseño, pues los protocolos de comunicación de las diferentes capas pueden evolucionar en forma independiente.

En la Ilustración 34 se muestran algunas declaraciones comunes (en C) para muchos de los protocolos que analizaremos después. Allí se definen cinco estructuras de datos: *boolean*, *seq_nr*, *packet*, *frame_kind* y *frame*. Un *boolean* es un tipo enumerado que puede tener los valores *true* y *false*. Un *seq_nr* (número de secuencia) es un entero pequeño que sirve para numerar las tramas, con el fin de distinguirlas. Estos números de secuencia van de 0 hasta *MAX_SEQ* (inclusive), el cual es definido en cada protocolo según sus necesidades. 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**.

El campo *kind* indica si hay datos en la trama, ya que algunos de los protocolos distinguen entre las tramas que contienen sólo información de control y las que también contienen datos. Los campos *seq* y *ack* se emplean para números de secuencia y confirmaciones de recepción, respectivamente; describiremos su uso más adelante 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, el cual se omitiría por completo en las tramas de control.

```
#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 del paquete */
typedef enum {data, ack, nak} frame_kind;              /* definición de frame_kind */

typedef struct {                                         /* las tramas se transportan en esta capa */
    frame_kind kind;                                    /* ¿qué tipo de trama es? */
    seq_nr seq;                                       /* número de secuencia */
    seq_nr ack;                                       /* número de confirmación de recepción */
    packet info;                                      /* paquete de la capa de red */
} frame;

/* Espera a 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 y deshabilita 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 y deshabilita 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 a k. */
#define inc(k) if (k < MAX_SEQ) k = k + 1; else k = 0
```

Ilustración 34 - Algunas definiciones necesarias en los protocolos que siguen. Estas definiciones se encuentran en el archivo protocol.h.

Es importante entender la relación entre un paquete y una trama. Para construir un paquete, la capa de red toma un mensaje de la capa de transporte y le agrega el encabezado de la capa de red. Después 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, la capa de red puede actuar como si las máquinas pudieran intercambiar paquetes directamente.

En la Ilustración 34 también se listan varios procedimientos, los cuales son rutinas de biblioteca cuyos detalles dependen de la implementación, por lo que no nos ocuparemos de su funcionamiento interno. El

procedimiento *wait_for_event* se queda en un ciclo cerrado esperando que algo ocurra, como se mencionó antes. La capa de enlace de datos usa los procedimientos *to_network_layer* y *from_network_layer* para pasar paquetes a la capa de red y aceptar paquetes de ella, respectivamente. Cabe mencionar que *from_physical_layer* y *to_physical_layer* pasan tramas entre la capa de enlace de datos y la capa física. En otras palabras, las primeras dos tienen que ver con la interfaz entre las capas 2 y 3, mientras que las últimas tratan con la interfaz entre las capas 1 y 2.

En la mayoría de los protocolos suponemos que el canal es no confiable y en ocasiones pierde tramas completas. Para poder recuperarse de tales calamidades, la capa de enlace de datos emisora debe iniciar un temporizador o reloj interno cada vez que envía una trama. Si no obtiene respuesta después de cierto intervalo predeterminado, el reloj expira y la capa de enlace de datos recibe una señal de interrupción.

En nuestros protocolos, para manejar esto es necesario permitir que el procedimiento *wait_for_event* devuelva *event=timeout*. Los procedimientos *start_timer* y *stop_timer* inician y detienen el temporizador, respectivamente. Los eventos de expiración del temporizador sólo son posibles cuando éste se encuentra en funcionamiento y antes de llamar a *stop_timer*. Se permite de manera explícita llamar a *start_timer* cuando el temporizador está en operación; 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 que se utiliza para generar confirmaciones de recepción en ciertas condiciones.

Los procedimientos *enable_network_layer* y *disable_network_layer* se usan en los protocolos más sofisticados, en los que ya no suponemos que la capa de red siempre tiene paquetes para enviar. Cuando la capa de enlace de datos habilita a la capa de red, ésta tiene permitido interrumpir cada vez que tenga que enviar un paquete. Esto lo indicamos con *event=network_layer_ready*. Cuando la capa de red está deshabilitada, no puede causar dichos eventos. Si tiene cuidado respecto a cuándo debe habilitar y deshabilitar 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), en 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, el factor que limita con frecuencia el desempeño de una red 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.

Las declaraciones de la Ilustración 34 son parte de todos los protocolos que estudiaremos en breve. Para ahorrar espacio y proveer una referencia conveniente, se han extraído y listado juntas, pero conceptualmente deberían estar integradas con los protocolos mismos. En C, para llevar a cabo esta integración se colocan las definiciones en un archivo especial de encabezado, en este caso *protocol.h*, y se utiliza la directiva *#include* del preprocesador de C para incluirlas en los archivos de protocolos.

Un protocolo simplex utópico

Como ejemplo inicial consideraremos un protocolo que es lo más sencillo posible, por la posibilidad de que algo salga mal. Los datos son transmitidos en una sola dirección; las capas de red tanto del emisor como del receptor siempre están listas. Podemos ignorar el tiempo de procesamiento. Hay un espacio infinito de búfer disponible. Y lo mejor de todo, el canal de comunicación entre las capas de enlace de datos nunca daña ni pierde las tramas. Este protocolo completamente irreal, al que apodaremos “*utopía*”, es simplemente para mostrar la estructura básica en la que nos basaremos. Su implementación se muestra en la Ilustración 35.

```

/* 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 todas las entradas a una rapidez infinita. En consecuencia, el emisor se mantiene en un ciclo, enviando datos a la línea tan
rápido 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);
        s.info = buffer;
        to_physical_layer(&s);               /* consigue algo qué enviar */
                                                /* lo copia en s para transmitirlo */
                                                /* 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);
            from_physical_layer(&r);
            to_network_layer(&r.info);      /* la única posibilidad es frame_arrival */
                                                /* obtiene la trama entrante */
                                                /* pasa los datos a la capa de red */
        }
    }
}

```

Ilustración 35 - Un protocolo simplex utópico.

El protocolo consiste en dos procedimientos diferentes, un emisor y un 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 rápido como puede. El cuerpo del ciclo consiste en tres acciones: obtener un paquete de la (siempre disponible) 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 este tipo.

El receptor también es sencillo. Al principio espera que algo ocurra, siendo la única posibilidad la llegada de una trama sin daños. En algún momento llega la trama, el procedimiento *wait_for_event* regresa y *event* contiene 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 coloca en la variable *r*, en donde el código receptor pueda obtenerla. Por último, la parte de los datos se pasa a la capa de red y la capa de enlace de datos se retira para esperar la siguiente trama, para lo cual se suspende efectivamente hasta que ésta llega.

El protocolo utópico es irreal, ya que no maneja el control de flujo ni la corrección de errores. Su procesamiento se asemeja al de un servicio sin conexión ni confirmación de recepción que depende de las capas más altas para resolver estos problemas, aun cuando un servicio sin conexión ni confirmación de recepción realizaría cierta detección de errores.

Protocolo simplex de parada y espera para un canal libre de errores

Ahora debemos lidiar con el problema principal de evitar que el emisor saturé al receptor enviando tramas a una mayor velocidad de la que este último puede procesarlas. Esta situación puede ocurrir con facilidad en la

práctica, por lo que es de extrema importancia evitarla. Sin embargo, aún existe el supuesto de que el canal está libre de errores y el tráfico de datos sigue siendo simplex.

```

/* El protocolo 2 (parada y espera) también provee 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 una mayor velocidad
de la que pueda manejar. */

typedef enum {frame_arrival} event_type;
#include "protocol.h"

void sender2(void)
{
    frame s;
    packet buffer;
    event_type event;

    while (true) {
        from_network_layer(&buffer);
        s.info = buffer;
        to_physical_layer(&s);
        wait_for_event(&event);
    }
}

void receiver2(void)
{
    frame r, s;
    event_type event;
    while (true) {
        wait_for_event(&event);
        from_physical_layer(&r);
        to_network_layer(&r.info);
        to_physical_layer(&s);
        /* envía una trama ficticia para despertar al emisor */
    }
}

```

Ilustración 36 - Un protocolo simplex de parada y espera.

Una solución es construir un receptor lo suficientemente poderoso como para procesar un flujo continuo de tramas, una tras otra sin interrupción (lo equivalente sería definir la capa de enlace de modo que fuera lo bastante lento como para que el receptor pudiera mantenerse a la par). Debe tener suficiente capacidad en el búfer y de procesamiento como para operar a la tasa de transmisión de la línea; asimismo debe ser capaz de pasar las tramas que se reciben en la capa de red con la rapidez suficiente. Sin embargo, ésta es una solución para el peor de los casos. Requiere hardware dedicado y se pueden desperdiciar recursos si el enlace se usa poco. Además, sólo cambia el problema de tratar con un emisor demasiado rápido a otra parte; en este caso, a la capa de red.

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 que transmita la siguiente trama. Después de enviar una trama, el protocolo exige que el emisor espere hasta que llegue la pequeña trama ficticia (es decir, la confirmación de recepción). Este retraso es un ejemplo simple de un protocolo de control de flujo.

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 Ilustración 36 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, luego el receptor envía otra, y así sucesivamente. Aquí sería suficiente un canal físico semi-dúplex.

Al igual que en el protocolo 1, el emisor comienza obteniendo un paquete de la capa de red, lo usa 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 ni siquiera necesita inspeccionar la trama entrante, ya que sólo hay una posibilidad. La trama entrante 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 al emisor y no su contenido, el receptor no necesita poner ninguna información específica en la trama.

Protocolo simplex de parada y espera para un canal ruidoso

Ahora consideremos la situación normal de un canal de comunicación que comete errores. Las tramas pueden llegar dañadas o se pueden perder 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 (una ocurrencia muy poco probable), este protocolo (y todos los demás) puede fallar (es decir, tal vez entregue 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. Después de un tiempo el temporizador del emisor expiraría y éste enviaría la trama otra vez. Este proceso se repetiría hasta que la trama por fin llegara intacta.

Pero el esquema anterior tiene un defecto fatal. Para ver lo que puede resultar mal, recuerde que el objetivo de la capa de enlace de datos es 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 su capa de enlace de datos, la cual debe asegurar que se entregue una serie de paquetes idénticos 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 perdió o duplicó, por lo que la capa de enlace de datos debe garantizar que ninguna combinación de errores de transmisión, por improbables que sean, pudiera 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. La máquina B recibe correctamente el paquete y lo pasa a su capa de red. 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.

Sin duda, 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 verificar el número de secuencia de cada trama que llega para ver si es una trama nueva o un duplicado que debe descartarse.

Como el protocolo debe ser correcto y es probable que el campo de número de secuencia en el encabezado sea pequeño como para poder usar el enlace en forma eficiente, surge la pregunta: ¿cuál es la cantidad mínima de bits necesarios para el número de secuencia? El encabezado podría proveer 1 bit, unos cuantos bits, 1 byte o varios bytes para un número de secuencia, dependiendo del protocolo. El punto importante es que debe transportar números de secuencia que sean lo bastante grandes como para que el protocolo funcione de manera correcta, o de lo contrario no se podrá considerar un verdadero protocolo.

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 reciba correctamente, el receptor regresará 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 o no, tratará de enviar m o $m+1$.

En el emisor, el evento que desencadena la transmisión de la trama $m+1$ es la llegada de una confirmación de recepción de la trama m . Pero esto implica que $m-1$ se recibió correctamente; también implica que el emisor recibió correctamente su confirmación de recepción. De otra manera, el emisor no habría comenzado con m y mucho menos hubiera considerado $m+1$. Como consecuencia, la única ambigüedad es entre una trama y su antecesor o sucesor inmediato, no entre el antecesor y el sucesor.

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. Cuando llega una trama que contiene el número de secuencia correcto, se acepta y se pasa a la capa de red, para después confirmar su recepción. Luego, el número de secuencia esperado se incrementa módulo 2 (es decir, 0 se vuelve 1 y 1 se vuelve 0). Cualquier trama que llegue y que contenga el número de secuencia incorrecto se rechaza como duplicada. Sin embargo, la última confirmación de recepción válida se repite de modo que el emisor pueda descubrir en un momento dado que se recibió la trama.

En la Ilustración 37 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 se conocen comúnmente como **ARQ** (Solicitud Automática de Repetición, del inglés *Automatic Repeat reQuest*) o **PAR** (Confirmación de Recepción Positiva con Retransmisión, del inglés *Positive Acknowledgement with Retransmission*). Al igual que el protocolo 2, éste también transmite datos en una sola dirección.

El protocolo 3 difiere de sus antecesores en cuando a 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 pequeña fase de inicialización antes de entrar en el ciclo infinito.

Después de transmitir una trama, el emisor inicia el temporizador. Si ya estaba en operación, se restablece para conceder otro intervalo completo de temporización. Hay que elegir dicho intervalo 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 propague de vuelta al emisor. Sólo hasta que haya transcurrido ese intervalo podremos suponer con seguridad que se ha perdido la trama transmitida o su confirmación de recepción, y se debe enviar un duplicado. Si el intervalo establecido es muy pequeño, el emisor transmitirá tramas innecesarias. Si bien estas tramas adicionales no afectarán el funcionamiento correcto del protocolo, sí afectarán el desempeño.

```

/* El protocolo 3 (PAR) permite el flujo unidireccional de datos por un canal no confiable. */

#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 */

    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 */

        next_frame_to_send = 0;                     /* inicializa números de secuencia de salida */
        from_network_layer(&buffer);             /* obtiene el primer paquete */

        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 */
                                                /* frame_arrival, cksum_err, timeout */

        wait_for_event(&event);
        if (event == frame_arrival){
            from_physical_layer(&s);
            if (s.ack == next_frame_to_send){
                stop_timer(s.ack);
                from_network_layer(&buffer);
                inc(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){
                from_physical_layer(&r);
                if (r.seq == frame_expected){
                    to_network_layer(&r.info);
                    inc(frame_expected);
                }
                s.ack = 1 - frame_expected;
                to_physical_layer(&s);
            }
        }
    }
}

```

Ilustración 37 - Un protocolo de confirmación de recepción positiva con retransmisión.

Después de transmitir una trama e iniciar el temporizador, el emisor espera a que ocurra algo interesante. Sólo hay tres posibilidades: que una trama de confirmación de recepción llegue sin daño, que llegue una trama de confirmación de recepción dañada o que expire el temporizador. Si entra una confirmación de recepción válida, el emisor obtiene el siguiente paquete de su capa de red y lo coloca en el búfer, sobre escribiendo el paquete anterior. También incrementa el número de secuencia. Si llega una trama dañada o expira el temporizador, no cambia ni el búfer ni el número de secuencia de modo que se pueda enviar un duplicado. En todos los casos se envía a continuación el contenido del búfer (ya sea el siguiente paquete o 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, pero hacen que se confirme la recepción de la última trama que se recibió correctamente para avisar al emisor de modo que avance a la siguiente trama o retransmita la trama dañada.

Protocolos de ventana deslizante

En los protocolos anteriores, las tramas de datos se transmitían en una sola dirección. En la mayoría de las situaciones prácticas existe la necesidad de transmitir datos en ambas direcciones. Una manera de lograr una transmisión de datos full-dúplex es tener dos instancias de uno de los protocolos anteriores, cada uno de los cuales debe usar un enlace separado para el tráfico de datos simplex (en distintas direcciones). A su vez, cada enlace se compone de un canal de “ida” (para los datos) y de un canal de “retorno” (para las confirmaciones de recepción). En ambos casos se desperdicia la capacidad del canal de retorno casi por completo.

Una mejor idea es utilizar el mismo enlace para datos en ambas direcciones. Después de todo, en los protocolos 2 y 3 ya se usaba para transmitir tramas en ambos sentidos, y por lo general 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 entremezclan con las tramas de confirmación de recepción de *A* a *B*. Si analizamos el campo *kind* en el encabezado de una trama entrante, el receptor puede saber si la trama es de datos o de confirmación de recepción.

Aunque intercalar datos y tramas de control en el mismo enlace es una gran mejora respecto al uso de dos enlaces físicos separados, es posible realizar otra mejora. Cuando llega una trama de datos, en lugar de enviar de inmediato 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 (mediante el uso del 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 salientes para que puedan viajar en la siguiente trama de datos de salida se conoce como **superposición** (*piggybacking*).

La principal ventaja 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 separada requeriría de un encabezado, la confirmación de recepción y una suma de verificación. Además, el envío de menos tramas casi siempre representa una carga de procesamiento más ligera en el receptor. En el siguiente protocolo que vamos a examinar, el campo de superposición sólo ocupa 1 bit en el encabezado de trama. Pocas veces ocupa más de unos cuantos 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 pueda superponer la confirmación de recepción? Si la capa de enlace de datos espera más tiempo del que tarda en terminar el temporizador del emisor, se volverá a transmitir la trama y se frustrará el propósito de enviar confirmaciones de recepción. Dado que la capa de enlace de datos no puede predecir cuando la capa de red tendrá un nuevo dato disponible, se 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 simplemente envía una trama de confirmación de recepción independiente.

Los siguientes tres protocolos son bidireccionales y pertenecen a una clase llamada protocolos de **ventana deslizante**. Los tres difieren entre ellos en términos de eficiencia, complejidad y requerimientos de búfer, como veremos más adelante. En ellos, al igual que en todos los protocolos de ventana deslizante, cada trama de salida contiene un número de secuencia que va desde 0 hasta algún número máximo. Por lo general este valor máximo es $2^n - 1$, por lo que el número de secuencia encaja perfectamente en un campo de *n* bits. El protocolo de ventana deslizante de parada y espera utiliza *n*=1 y restringe los números de secuencia de 0 y 1, pero las versiones más sofisticadas pueden utilizar un *n* arbitrario.

La esencia de todos los protocolos de ventana deslizante es que, en cualquier instante, el emisor mantiene un conjunto 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 similar, el receptor mantiene una **ventana receptora** correspondiente al conjunto 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 aumentar o reducir su tamaño con el transcurso del tiempo, 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 requerimiento 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 requerimiento de que el canal físico de comunicación sea 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 las tramas que se han enviado, o que se pueden enviar pero aún no se ha confirmado su recepción. Cada vez que llega un paquete nuevo de la capa de red, se le asigna el siguiente número secuencial más alto y el extremo superior de la ventana avanza en uno. Cuando llega una confirmación de recepción, el extremo inferior avanza en uno. De esta manera, la ventana mantiene en forma continua una lista de tramas sin confirmación de recepción. En la Ilustración 38 se muestra un ejemplo.

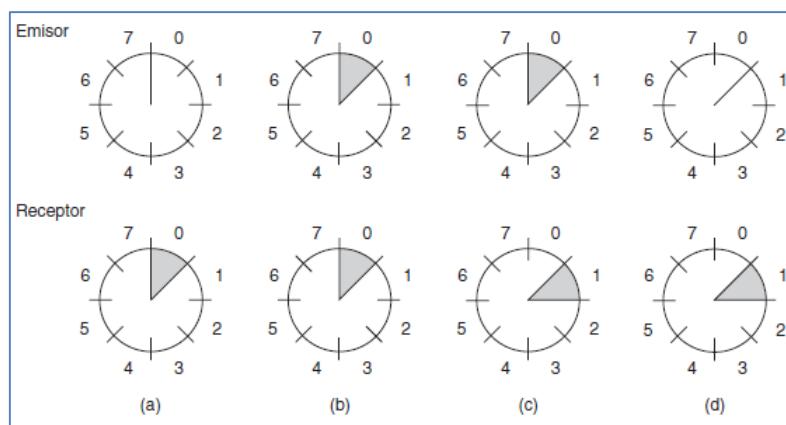


Ilustración 38 - Una ventana deslizante de tamaño 1, con un número de secuencia de 3 bits. (a) Al inicio. (b) Despues de enviar la primera trama. (c) Despues de recibir la primera trama. (d) Despues de recibir la primera confirmación de recepción.

Debido a que las tramas que están en la ventana del emisor se pueden perder o dañar 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 receptora corresponde a las tramas que puede aceptar. Toda trama que caiga dentro de la ventana se colocará en el búfer del receptor. Cuando se reciba una trama cuyo número de secuencia sea igual al extremo inferior de la ventana, se pasará a la capa de red y la ventana se desplazará una posición. Cualquier trama que caiga fuera de la ventana se desechará. En todos estos casos se genera una confirmación de recepción subsiguiente, de manera que el emisor pueda averiguar cómo proceder. Cabe mencionar 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. En contraste, la capa de red siempre recibe los datos en el orden correcto, sin importar el tamaño de la ventana de la capa de enlace de datos.

En la Ilustración 38 se muestra un ejemplo con un tamaño máximo de ventana de 1. En un principio no hay tramas pendientes, por lo que los extremos inferior y superior de la ventana del emisor son iguales, pero a medida que pasa el tiempo, la situación progresiva como se muestra. A diferencia de la ventana del emisor, la ventana del receptor siempre permanece en su tamaño inicial, y se desplaza a medida que se acepta la siguiente trama y se entrega a la capa de red.

Un protocolo de ventana deslizante de un bit

Antes de tratar el caso general, examinemos un protocolo de ventana deslizante 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 Ilustración 39 se describe dicho protocolo. Como los demás, comienza por definir algunas variables. *next_frame_to_send* indica qué trama está tratando de enviar el emisor. Asimismo, *frame_expected* indica qué trama espera el receptor. En ambos casos, 0 y 1 son las únicas posibilidades.

```

/* El protocolo 4 (ventana deslizante) 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 */

    next_frame_to_send = 0;            /* siguiente trama del flujo de salida */
    frame_expected = 0;              /* próxima trama 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 */
    to_physical_layer(&s);          /* confirmación de recepción superpuesta */
    /* transmite la trama */
    start_timer(s.seq);             /* inicia el temporizador */

    while (true){
        wait_for_event(&event);
        if (event == frame_arrival){  /* frame_arrival, cksum_err o timeout */
            from_physical_layer(&r);
            if(r.seq == frame_expected){ /* ha llegado una trama sin daño. */
                to_network_layer(&r.info);
                inc(frame_expected);   /* la obtiene */
                /* maneja flujo de tramas de entrada. */
                /* pasa el paquete a la capa de red */
                /* invierte el siguiente número de secuencia esperado */
            }
            if(r.ack == next_frame_to_send){ /* maneja flujo de tramas de salida. */
                stop_timer(r.ack);
                from_network_layer(&buffer);
                inc(next_frame_to_send);  /* desactiva el temporizador */
                /* obtiene un nuevo paquete de la capa de red */
                /* invierte el número de secuencia del emisor */
            }
        }
        s.info = buffer;               /* construye trama de salida */
        s.seq = next_frame_to_send;    /* le inserta el número de secuencia */
        s.ack = 1 - frame_expected;   /* número de secuencia de la última trama recibida */
        to_physical_layer(&s);       /* transmite una trama */
        start_timer(s.seq);          /* inicia el temporizador */
    }
}

```

Ilustración 39 - Protocolo de ventana deslizante de 1 bit.

En circunstancias normales, 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. La máquina que inicia 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 del receptor la verifica 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 (*buffer*) y que puede obtener el siguiente paquete de su capa de red. Si el número de secuencia no concuerda, debe seguir tratando de enviar la misma trama. Cada vez que se recibe una trama, también se regresa una.

Ahora examinemos el protocolo 4 para ver qué tan flexible es ante circunstancias problemáticas. Suponga que la computadora A trata de enviar su trama 0 a la computadora B y que B trata 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, el

temporizador de *A* se podría agotar repetidamente, enviando una serie de tramas idénticas, todas con *seq=0* y *ack=1*.

Cuando llegue la primera trama válida a la computadora *B*, se aceptará y *frame_expected* se establecerá en 1. Todas las tramas subsiguientes serán rechazadas, puesto que ahora *B* 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.

Cada vez que llegue un duplicado rechazado, *B* enviará a *A* una trama que contenga *seq=0* y *ack=0*. Tarde o temprano una de éstas llegará correctamente a *A* y hará que empiece a enviar el siguiente paquete. Ninguna combinación de tramas perdidas o temporizadores que expiren antes de tiempo 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. El protocolo es correcto.

Sin embargo, para demostrar qué tan sutiles pueden ser las interacciones entre los protocolos, cabe mencionar que si ambos lados envían de manera simultánea un paquete inicial, surge una situación peculiar. En la Ilustración 40 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 se aceptan todas las tramas.

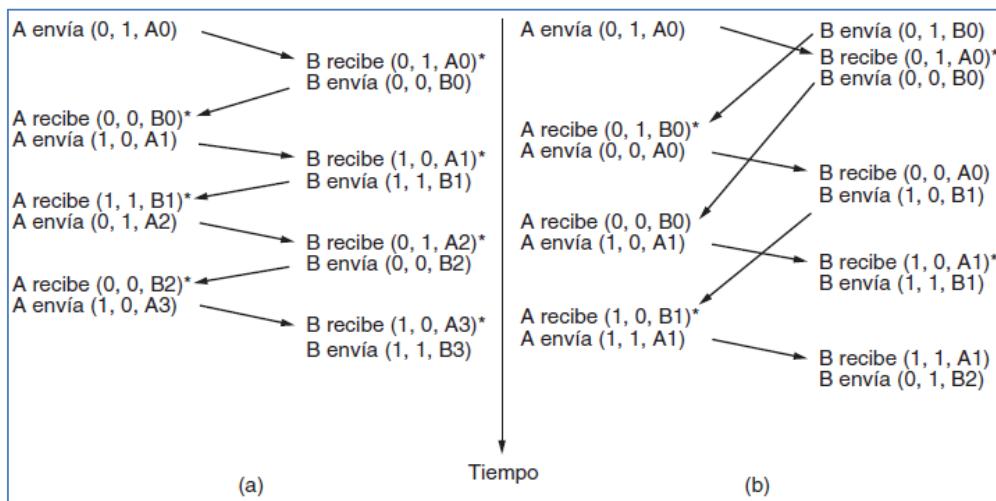


Ilustración 40 - Dos escenarios para el protocolo 4. (a) Caso normal. (b) Caso anormal. La notación es (secuencia, confirmación de recepción, número de paquete). Un asterisco indica el lugar en que una capa de red acepta un paquete.

Pero 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, incluso cuando sea evidente que un lado empezó primero. De hecho, si ocurren varias expiraciones prematuras de temporizadores, tal vez las tramas se envíen dos o tres veces, lo cual representa un desperdicio del valioso ancho de banda.

Un protocolo que utiliza 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 de la utilización del ancho de banda. Por ejemplo, considere un canal de satélite de 50kbps con un retardo de propagación de ida y vuelta de 500ms. Imagine que intentamos utilizar el protocolo 4 para enviar tramas de 1000 bits por medio del satélite. En $t=0$, el emisor empieza a enviar la primera trama. En $t=20ms$ la trama se ha enviado por completo. En el mejor de los casos (sin esperas en el receptor y con una trama de confirmación de recepción corta), no es sino hasta $t=270ms$ que la trama ha llegado por completo al receptor, y no es sino hasta $t=520ms$ que ha llegado la confirmación de recepción de

regreso al emisor. Esto implica que el emisor estuvo bloqueado durante 500/520 o 96% del tiempo. En otras palabras, sólo se usó 4% del ancho de banda disponible. Sin duda, la combinación de un tiempo de tránsito grande, un alto ancho de banda y una longitud de tramas corta es desastrosa en términos de eficiencia.

El problema antes descrito puede considerarse 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, podremos obtener una mejor eficiencia. Básicamente la solución está en permitir que el emisor envíe hasta w tramas antes de bloquearse, en lugar de que sólo envíe 1. Con una selección suficientemente grande de w , el emisor podrá transmitir tramas en forma continua, ya que las confirmaciones de recepción llegarán para las tramas anteriores antes de que la ventana se llene, lo cual evitara que el emisor se bloquee.

Para encontrar un valor apropiado para w necesitamos saber cuántas tramas pueden caber dentro del canal mientras se propagan del emisor al receptor. Esta capacidad se determina mediante el ancho de banda en *bits/s*, multiplicado por el tiempo de tránsito en un sentido, mejor conocido como **producto de ancho de banda-retardo** del enlace. Podemos dividir esta cantidad entre el número de bits en una trama para expresar el producto como un número de tramas. Llámemos a esta cantidad *BD*. Entonces, w debe ser establecido a $2BD+1$. El doble del producto ancho de banda-retardo es el número de tramas que pueden quedar pendientes si el emisor envía en forma continua tramas cuando se considera el tiempo de ida y vuelta para recibir una confirmación de recepción. El “+1” se debe a que no se enviará una trama de confirmación de recepción sino hasta recibir una trama completa.

Para el enlace de ejemplo con un ancho de banda de 50kbps y un tiempo de tránsito en un sentido de 250ms, el producto de ancho de banda-retardo es de 12.5 kbit o 12.5 tramas de 1000 bits cada una. Entonces, $2BD+1$ es 26 tramas. Suponga que el emisor empieza a enviar la trama 0 como antes y que envía una nueva trama cada 20ms. Para cuando termine de enviar 26 tramas en $t=520ms$, apenas sí llegará la confirmación de recepción de la trama 0. A partir de entonces, las confirmaciones de recepción llegarán cada 20ms, por lo que el emisor siempre tendrá permiso de continuar justo cuando lo necesite. De aquí en adelante siempre habrá 25 o 26 tramas pendientes de confirmación de recepción. Dicho de otra manera, el tamaño máximo de la ventana del emisor es de 26.

Para tamaños de ventana más pequeños, el uso del enlace será de menos de 100% debido a que el emisor estará bloqueado algunas veces. Podemos escribir la utilización como la fracción de tiempo que el emisor no está bloqueado:

$$\text{utilización del enlace} \leq \frac{w}{1 + 2BD}$$

Este valor es un límite superior ya que no considera ningún tiempo de procesamiento de tramas y supone que la trama de confirmación de recepción tiene una longitud de cero, puesto que generalmente es corta. La ecuación muestra la necesidad de tener una ventana w grande siempre que el producto ancho de banda-retardo también lo sea. Si el retardo es alto, el emisor agotará rápidamente su ventana incluso para un ancho de banda moderado, como en el ejemplo del satélite. Si el ancho de banda es alto, incluso para un retardo moderado, el emisor agotará su ventana con rapidez, a menos que tenga una ventana grande (por ejemplo, un enlace de 1Gbps con un retraso de 1ms contiene 1 megabit). Con el protocolo de parada y espera en el cual $w=1$, si hay incluso una trama equivalente al retardo de propagación, la eficiencia será menor a 50%.

Esta técnica de mantener varias tramas en movimiento es un ejemplo de **canalización** (pipeling). La canalización de tramas a través de un canal de comunicación no confiable presenta problemas serios. Primero, ¿qué ocurre si una trama a la mitad de un flujo extenso 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 es obvio que debe descartarse pero, ¿qué debe hacer el receptor con todas las tramas correctas que le siguen? Recuerde que la capa de enlace de datos del receptor está obligada a entregar paquetes a la capa de red en secuencia.

Hay dos métodos básicos disponibles para tratar con los errores en presencia de la canalización, ambos se muestran en la Ilustración 41.

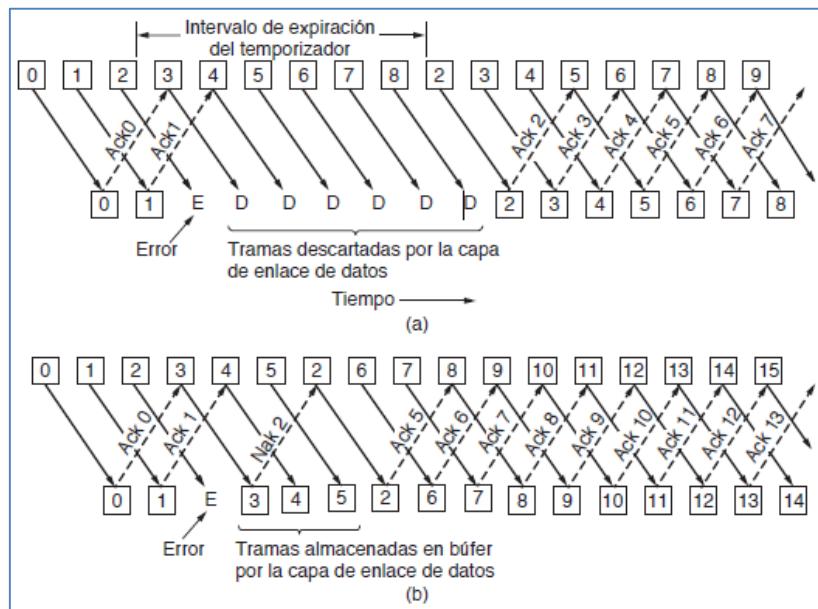


Ilustración 41 - Canalización y recuperación de errores. Efecto de un error cuando (a) el tamaño de la ventana del receptor es 1 y (b) el tamaño de la ventana del receptor es grande.

Una de las opciones, 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 que expire 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 trama dañada o perdida. Esta estrategia puede desperdiciar bastante ancho de banda si la tasa de error es alta.

En la Ilustración 41(a) podemos ver el retroceso n para 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, sin saber sobre 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 nuevamente las tramas 2, 3, 4, etcétera.

La otra estrategia general para el manejo de errores cuando las tramas se colocan en canalizaciones se conoce como **repeticIÓN selectiva**. Cuando se utiliza, si se recibe una trama dañada se descarta, pero las tramas en buen estado que se reciben después de ésa se aceptan y almacenan en el búfer. Cuando expira el temporizador del emisor, sólo se retransmite la última trama sin confirmación de recepción. 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 corresponde a una ventana del receptor mayor a 1. Este método puede requerir grandes cantidades de memoria en la capa de enlace de datos, si la ventana es grande.

A menudo, la repetición selectiva se combina con el hecho de que el receptor envíe una confirmación de recepción negativa **NAK** (del inglés *negative acknowledgement*) al detectar un error; por ejemplo, cuando recibe un error de suma de verificación o una trama fuera de secuencia. Las NAK estimulan la retransmisión antes de que el temporizador correspondiente expire y, por lo tanto, mejoran el rendimiento.

```
/* 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 protocolos anteriores, no existe el supuesto de 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
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; se usa para flujo de salida */
seq_nr ack_expected;                /* la trama más antigua no confirmada hasta el momento*/
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;                  /* número de búferes de salida actualmente en uso */
seq_nr i;                           /* se usa para indexar en el arreglo de búferes */
event_type event;

enable_network_layer();              /* permite eventos network_layer_ready */
ack_expected = 0;                   /* siguiente ack esperada en el flujo de entrada */
next_frame_to_send = 0;              /* siguiente trama de salida */
frame_expected = 0;                 /* número de trama de entrada esperada */
nbuffed = 0;                        /* al inicio no hay paquetes en el búfer */

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;
}
```

(Continúa)

(Continuación)

```

case frame_arrival:
    from_physical_layer(&r);           /* ha llegado una trama de datos o de control */
                                         /* 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);            /* contrae 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; /* aquí inicia la retransmisión */
    for (i = 1; i <= nbuffed; i++) {
        send_data(next_frame_to_send, frame_expected, buffer); /* reenvía trama */
        inc(next_frame_to_send);          /* se prepara para enviar la siguiente */
    }
}

if (nbuffed < MAX_SEQ)
    enable_network_layer();
else
    disable_network_layer();
}

```

Ilustración 42 - Un protocolo de ventana deslizante con retroceso n.

En la Ilustración 41(b), las tramas 0 y 1 se vuelven a recibir y confirmar correctamente, pero 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 en el búfer. Cuando llegan las tramas 4 y 5, la capa de enlace de datos también las almacena en el búfer, en lugar de pasárlas a la capa de red. En algún momento la NAK 2 llega al emisor, que inmediatamente reenvía la trama 2. Cuando ésta llega, la capa de enlace de datos tiene las tramas 2, 3, 4 y 5, y puede pasárlas todas a la capa de red en el orden correcto. También puede confirmar la recepción de todas las tramas 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) por su cuenta, pero eso podría tardar un poco más.

Estos dos métodos alternativos son concesiones entre el uso eficiente del 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 u otro. En la Ilustración 42 se muestra un protocolo de retroceso n en el que la capa de enlace de datos del receptor sólo acepta tramas en orden; las tramas que siguen después de un error se descartan. En este protocolo hemos omitido por primera vez el supuesto de que la capa de red siempre tiene que enviar un suministro infinito de paquetes. Cuando la capa de red tiene un paquete que desea enviar, puede hacer que ocurra un evento *network_layer_ready*. Sin embargo, para mantener el límite de control de flujo en la ventana del emisor o el número de tramas sin confirmación de recepción pendientes en cualquier momento, la capa de enlace de datos debe ser capaz de 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 tarea.

El número máximo de tramas que pueden estar pendientes en cualquier instante no es igual que el tamaño del espacio del número de secuencia. Para el retroceso n puede haber *MAX_SEQ* tramas pendientes en cualquier instante, aun cuando haya *MAX_SEQ+1* números de secuencia diferentes (que son 0, 1,...,

MAX_SEQ). En el siguiente protocolo, repetición selectiva, veremos una restricción aún mayor. Para ver por qué es necesaria esta restricción, considere el siguiente escenario en donde *MAX_SEQ*=7.

1. El emisor envía las tramas 0 a 7.
2. Llega al emisor una confirmación de recepción superpuesta para la trama 7.
3. El emisor envía otras ocho tramas, de nuevo 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 las tramas descartadas después de un error)? En ambos casos el receptor podría estar enviando la trama 7 como la confirmación de recepción. El emisor no tiene manera de saberlo. Por esta razón, el número máximo de tramas pendientes se debe restringir a *MAX_SEQ*.

Aunque el protocolo 5 no pone en el búfer las tramas que llegan después de un error, no escapa del problema de los búferes por completo. Dado que un emisor tal vez tenga que retransmitir en un futuro todas las tramas no confirmadas, debe retener todas las tramas transmitidas hasta saber con certeza que el receptor las aceptó. Cuando llega una confirmación de recepción para la trama n , las tramas $n-1, n-2$ y demás se confirman también de manera automática. Este tipo de confirmación de recepción se llama **confirmación de recepción acumulativa**. Esta propiedad es importante cuando algunas de las tramas previas portadoras de confirmaciones de recepción se perdieron o dañaron. Cuando llega una confirmación de recepción, la capa de enlace de datos verifica si se pueden liberar búferes. Si esto es posible (es decir, hay espacio disponible en la ventana), a una capa de red bloqueada se le puede permitir que 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. El protocolo 4 no necesita este supuesto debido a que envía una trama cada vez que recibe una, incluso si ya la ha enviado. En el siguiente protocolo resolveremos el problema del tráfico de un solo sentido de una forma elegante.

Como el protocolo 5 tiene múltiples tramas pendientes, por lógica necesita múltiples temporizadores, uno por cada trama pendiente. El temporizador de cada trama expira de manera independiente a los de las otras tramas. Sin embargo, todos estos temporizadores se pueden simular fácilmente en software, mediante el uso de un solo reloj de hardware que produzca interrupciones en forma periódica. Los temporizadores pendientes de expiration forman una lista enlazada, en la que cada nodo contiene la cantidad de pulsos de reloj que restan para que expire el temporizador, la trama temporizada y un apuntador al siguiente nodo.

Para ver cómo se pueden implementar los temporizadores, considere el ejemplo de la Ilustración 43(a). Suponga que el reloj pulsa una vez cada 1ms. Al principio, la hora real es 10:00:00.000; hay tres temporizadores por expiration pendientes, a las 10:00:00.005, 10:00:00.013 y 10:00:00.019. Cada vez que pulsa el reloj de hardware, se actualiza el tiempo real y se decrementa el contador de pulsos a la cabeza de la lista. Cuando el contador de pulsos llega a cero, un temporizador expira y se retira el nodo de la lista, como se muestra en la Ilustración 43(b). Aunque esta organización requiere que se analice la lista al llamar a *start_timer* o a *stop_timer*, no requiere mucho trabajo por pulso de reloj. En el protocolo 5 estas dos rutinas tienen un parámetro que indica la trama a temporizar.

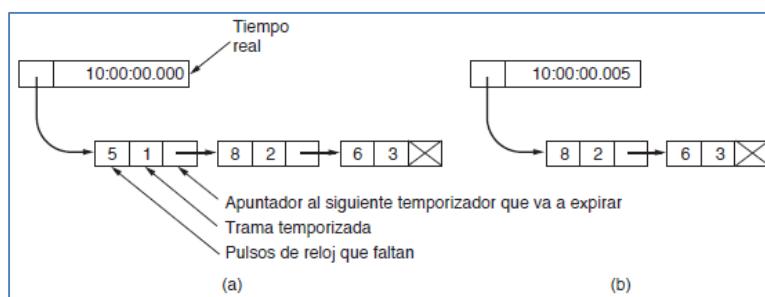


Ilustración 43 - Simulación de varios temporizadores en software. (a) Los temporizadores que van a expirar puestos en cola. (b) La situación después de que expira el primer temporizador.

Un protocolo que usa repetición selectiva

El protocolo de retroceso n funciona bien si los errores son poco frecuentes, pero si la línea es mala se desperdicia mucho ancho de banda en las tramas que se retransmiten. El protocolo de repetición selectiva es una estrategia alterna para que el receptor acepte y coloque en búferes las tramas que llegan después de una trama dañada o perdida.

```

/* 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 y 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 se necesita para tramas ack independientes */
}

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 */

nbuffered = 0;                             /* al principio no hay paquetes en el búfer */
for (i = 0; i < NR_BUFS; i++) arrived[i] = false;
while (true) {
    wait_for_event(&event);
    switch(event) {
        case network_layer_ready:
            /* cinco posibilidades: vea event_type al principio */
            /* acepta, guarda y transmite una trama nueva */
    }
}
}

```

(Continúa)

(Continuación)

```

nbuffed = nbuffed + 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 el búfer como lleno */
        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 independiente */
        }
    }
}
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)) {
    nbuffed = nbuffed - 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); /* expiró el temporizador */
break;

case ack_timeout:
send_frame(ack,0,frame_expected, out_buf); /* expiró el temporizador de ack; envía ack */
}
if (nbuffed < NR_BUFS) enable_network_layer(); else disable_network_layer();
}
}

```

Ilustración 44 - Protocolo de ventana deslizante con repetición selectiva.

En este protocolo, tanto el emisor como el receptor mantienen una ventana de números de secuencia pendientes y aceptables, respectivamente. El tamaño de la ventana del emisor comienza en 0 y crece hasta un máximo predefinido. Por el contrario, la ventana del receptor siempre es de tamaño fijo e igual al máximo predeterminado. El receptor tiene un búfer reservado para cada número de secuencia dentro de su ventana fija. Cada búfer tiene un bit asociado (*arrived*), el cual indica si el búfer está lleno o vacío. Cada vez que llega una trama, la función *between* verifica su número de secuencia para ver si cae dentro de la ventana. De ser así, y si todavía no se recibe, se acepta y almacena. Esta acción se lleva a cabo sin importar si la trama contiene o no el siguiente paquete que espera la capa de red. Claro que se debe mantener dentro de la capa de enlace de datos sin pasarlala a la capa de red hasta que se hayan entregado todas las tramas de menor número a la capa de red en el orden correcto. En la Ilustración 44 se presenta un protocolo que usa este algoritmo.

La recepción no secuencial introduce limitaciones adicionales en los números de secuencia de tramas, 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 3 bits, de modo que se permita al emisor transmitir hasta siete tramas antes de tener que esperar una confirmación de recepción. En un principio las ventanas del emisor y del receptor están como se muestra en la Ilustración 45(a). Ahora el emisor 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 la trama 7, 0, 1, 2, 3, 4 o 5, como se muestra en la Ilustración 45(b). Los siete búfer se marcan como vacíos.

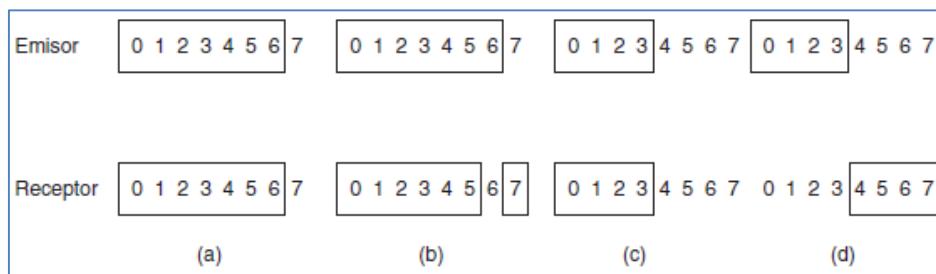


Ilustración 45 - (a) Situación inicial con una ventana de tamaño 7. (b) Despues de enviar y recibir 7 tramas sin regresar confirmaciones de recepción. (c) Situación inicial con un tamaño de ventana de 4. (d) Despues de enviar y recibir 4 tramas sin regresar confirmaciones de recepción.

En este momento, ocurre un desastre: un rayo cae en el poste telefónico y borra todas las confirmaciones de recepción. El protocolo debe ser capaz de operar en forma correcta a pesar de esto. En algún momento expira el temporizador del emisor y éste retransmite la trama 0. Cuando llega esta trama al receptor, se efectúa una verificación para saber si está dentro de la ventana del mismo. Por desgracia, en la Ilustración 45(b) la trama 0 está dentro de la nueva ventana, por lo que se acepta como una nueva trama. El receptor también envía una confirmación de recepción (superpuesta) para la trama 6, ya que se han recibido de la 0 a la 6.

El emisor está feliz de saber que todas las tramas que transmitió llegaron en forma correcta, por lo que avanza su ventana y de inmediato envía las tramas 7, 0, 1, 2, 3, 4 y 5. El receptor aceptará la trama 7 y su paquete se pasará directamente a la capa de red. Enseguida la capa de enlace de datos receptora verifica si ya tiene una trama 0 válida, descubre que sí y pasa el paquete anterior del búfer a la capa de red como si fuera el nuevo paquete. En consecuencia, la capa de red obtiene un paquete incorrecto y el protocolo fracasa.

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 contener tramas duplicadas (si se perdieron todas las confirmaciones de recepción) o nuevas (si se recibieron todas las confirmaciones de recepción). El 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. Esta situación se muestra en las Ilustraciones 45(c) y (d). Con 3 bits, los números de secuencia varían de 0 a 7. Sólo debe haber cuatro tramas sin confirmación de recepción pendientes en cualquier instante. De esa forma, si el receptor acaba de aceptar las tramas 0 a 3 y ha avanzado su ventana para permitir la aceptación de las tramas 4 a 7, puede distinguir sin ambigüedades si las tramas subsiguientes son retransmisiones (0 a 3) o si son nuevas (4 a 7). En general, el tamaño de la ventana para el protocolo 6 será $(MAX_SEQ+1)/2$.

Una pregunta interesante es: ¿cuántos búfer debe tener el receptor? En ninguna circunstancia podrá aceptar tramas cuyos números de secuencia estén por debajo del extremo inferior de la ventana o de las tramas cuyos números de secuencia estén por encima del extremo superior de la ventana. En consecuencia, el número de búfer 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 3 bits, se requieren cuatro búfer numerados del 0 al 3. Al llegar la trama i , se coloca en el búfer $i \bmod 4$. Tenga en cuenta que, aun cuando i e $(i+4) \bmod 4$ 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 5.

Por la misma razón, el número de temporizadores requeridos es igual al número de búfer no al tamaño del espacio de secuencia. En efecto, hay un temporizador asociado a cada búfer. Cuando expira el temporizador, el contenido del búfer se retransmite.

El protocolo 6 también suaviza la suposición implícita de que el canal está fuertemente cargado. Hicimos esta suposición en el protocolo 5, cuando requeríamos que se enviaran tramas en dirección inversa para superponer las confirmaciones de recepción. Si el tráfico de regreso es ligero, las confirmaciones de recepción se pueden retener por un largo periodo, lo cual puede provocar problemas. En un caso extremo, si hay mucho tráfico en una dirección y no hay tráfico en la otra dirección, el protocolo se bloqueará cuando la ventana del emisor llegue a su máximo.

Para suavizar esta suposición, se inicia un temporizador auxiliar mediante *start_ack_timer* después de que llega una trama de datos en la secuencia correcta. Si no se ha presentado tráfico de regreso antes de que expire este temporizador, se envía una trama de confirmación de recepción independiente. Una interrupción debida al temporizador auxiliar se denomina evento *ack_timeout*. Con este arreglo, ahora es posible el flujo de tráfico unidireccional, pues el que no haya tramas de datos de regreso a las que se puedan superponer las confirmaciones de recepción ya no es un obstáculo. Sólo existe un temporizador auxiliar; si se invoca a *start_ack_timer* mientras el temporizador está en funcionamiento, no tiene efecto. El temporizador no se restablece ni se extiende, ya que su propósito es proveer cierta tasa mínima de confirmaciones de recepción.

Es indispensable que el tiempo de expiración asociado al temporizador auxiliar sea notablemente más corto que el del temporizador usado para terminar las tramas de datos. Esta condición es necesaria para asegurarse de que la confirmación de recepción de una trama recibida correctamente llegue antes de que expire el temporizador de retransmisión de la trama y la retransmita.

El protocolo 6 utiliza una estrategia más eficiente que el protocolo 5 para manejar los errores. Cuando el receptor tiene razones para sospechar que ocurrió un error, envía al emisor una trama de confirmación de recepción negativa (NAK). Dicha 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 estar al tanto de si ya se ha enviado una NAK para una trama específica. 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 todas formas expirará el temporizador del emisor en un momento dado y retransmitirá la trama que se perdió. Si llega la trama equivocada después de haber enviado una NAK y que ésta se haya perdido, *no_nak* será *true* y se iniciará el temporizador auxiliar. Cuando expire, 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, se procese ahí y se devuelva la confirmación de recepción es (casi) constante. En estos casos, el emisor puede ajustar su temporizador para que apenas sea mayor que el intervalo esperado entre enviar una trama y la recepción de su confirmación. En este caso no se utilizan las NAK.

No obstante, en otros casos el tiempo puede ser muy variable. Por ejemplo, si el tráfico de regreso es esporádico, el tiempo antes de la confirmación de recepción será más corto cuando haya tráfico de regreso y más largo cuando no lo haya. El emisor se enfrenta a la elección entre establecer el intervalo a un valor pequeño (y arriesgarse a que haya retransmisiones innecesarias) o establecerlo a un valor grande (y estar inactivo por un largo periodo después de un error). Ambas opciones desperdician ancho de banda. En general, si la desviación estándar del intervalo de confirmación de recepción es grande en comparación con el intervalo mismo, el temporizador se establecerá a un valor “holgado” que sea conservador. Así, las NAK pueden acelerar de manera apreciable la retransmisión de tramas perdidas o dañadas.

Un aspecto muy relacionado con el asunto de la expiración de los temporizadores y las NAK es cómo determinar qué trama causó que expirara un temporizador. En el protocolo 5 siempre es *ack_expected*, puesto que es la más antigua. En el protocolo 6 no hay ninguna manera sencilla de determinar quién hizo que expirara 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 expira el temporizador de la trama 0, se transmite 5 (una trama nueva), expira el temporizador de 1, expira el temporizador de 2 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 (es decir, las tramas que llevan las confirmaciones de recepción) se pierde durante un momento, expirará el temporizador de las siete tramas pendientes en ese orden.

Para evitar que el ejemplo se complique aún más, no hemos mostrado la administración de los temporizadores. En cambio, sólo suponemos que la variable `oldest_frame` se establece al momento en que expira el temporizador, para indicar qué trama hizo que expirara.

Ejemplos de protocolos de enlace de datos

Existen múltiples protocolos que cubren las necesidades de la capa de enlace de datos, cada uno desarrollado para cubrir las necesidades de distintas tecnologías y distintos servicios asociados. A continuación describiremos los principales que podemos encontrar.

HDLC

HDLC (control de enlace de datos de alto nivel, del inglés *High-Level Data Link Control*) es un protocolo de comunicaciones de propósito general punto a punto, que opera a nivel de enlace de datos. Se basa en ISO3309 e ISO4335. Surge como una evolución del anterior **SDLC**. Proporciona recuperación de errores en caso de pérdida de paquetes de datos, fallos de secuencia y otros, por lo que ofrece una comunicación confiable entre el transmisor y el receptor. De este protocolo derivan otros como LAPB, LAPF, LLC y PPP.

Características básicas del HDLC

HDLC define tres tipos de estaciones, tres configuraciones del enlace y tres modos de operación para la transferencia de los datos.

Los tres tipos de estaciones son:

- **Estación primaria:** se caracteriza porque tiene la responsabilidad de controlar el funcionamiento del enlace. Las tramas generadas por la primaria se denominan órdenes.
- **Estación secundaria:** funciona bajo el control de la estación primaria. Las tramas generadas por la estación secundaria se denominan respuestas. La primaria establece un enlace lógico independiente para cada una de las secundarias presentes en la línea.
- **Estación combinada:** es una mezcla entre las características de las primarias y las secundarias. Una estación de este tipo puede generar tanto órdenes como respuestas.

Las tres posibles configuraciones del enlace son:

- **Configuración no balanceada:** está formada por una estación primaria y una o más secundarias. Permite transmisión semi-duplex.
- **Configuración balanceada:** consiste en dos estaciones combinadas. Permite igualmente transmisión full-duplex o semi-duplex.
- **Configuración simétrica:** dos estaciones físicas, cada una con una estación lógica, de forma que se conectan una primaria de una estación física con la secundaria de la otra estación física.

Los tres modos de transferencia de datos son:

- **Modo de respuesta normal (NRM, Normal Response Mode):** se utiliza en la configuración no balanceada. La estación primaria puede iniciar la transferencia de datos a la secundaria, pero la secundaria solo puede transmitir datos usando respuestas a las órdenes emitidas por la primaria.
- **Modo balanceado asíncrono (ABM, Asynchronous Balanced Mode):** se utiliza en la configuración balanceada. En este modo cualquier estación combinada podrá iniciar la transmisión sin necesidad de recibir permiso por parte de la otra estación combinada.
- **Modo de respuesta asíncrono (ARM, Asynchronous Response Mode):** se utiliza en la configuración no balanceada. La estación secundaria puede iniciar la transmisión sin tener permiso explícito por parte

de la primaria. La estación primaria sigue teniendo la responsabilidad del funcionamiento de la línea, incluyendo la iniciación, la recuperación de errores, y la desconexión lógica.

El NRM suele usarse en líneas con múltiples conexiones y en enlaces punto a punto, mientras que el ABM es el más utilizado de los tres modos; debido a que en ABM no se necesitan hacer sondeos, la utilización de los enlaces punto a punto con full-duplex es más eficiente con este modo. ARM solo se usa en casos muy particulares.

Estructura

HDLC usa transmisión síncrona. Todos los intercambios se realizan a través de tramas, HDLC utiliza un formato único de tramas que es válido para todos los posibles intercambios: datos e información de control.

En la Ilustración 46 se muestra la estructura de una trama HDLC. Al campo de delimitación, de dirección y de control, que preceden al campo de información, se denominan **cabecera**. La **FCS** (*Frame Check Sequence*) junto con el otro campo de delimitación final que está a continuación del campo de datos se denomina **cola**.

Flag	Dirección	Control	Información	FCS	Flag (comienzo de la trama siguiente)
8 bits	8 bits	8 o 16 bits	Longitud variable, 0 o más bits, múltiplos de 8	16 bits	8 bits

Ilustración 46 - Estructura de la trama HDLC.

Los campos de delimitación están localizados en los dos extremos de la trama, y ambos corresponden a la siguiente combinación de bits 01111110. Se puede usar un único delimitador como final y comienzo de la siguiente trama simultáneamente. A ambos lados de la interfaz entre el usuario y la red, los receptores estarán continuamente intentando detectar esta secuencia para sincronizarse con el comienzo de la trama. Cuando se recibe una trama, la estación seguirá intentando detectar esa misma secuencia para determinar así el final de la trama.

A su vez en campo control puede tomar las siguientes codificaciones dependiendo del tipo de trama (información, supervisión y no numeradas que se verán más adelante):

- I: Información:

1	2-4	5	6-8
0	N(S)	P/F	N(R)

N(S): Número de secuencia enviada.
N(R): Número de secuencia recibida.
P/F: Bit de Sondeo/Final ("Poll/Final")

- S: Supervisión:

1-2	3-4	5	6-8
10	S	P/F	N(R)

S: bits para las tramas de supervisión (se explicará más adelante).

- No numeradas:

1-2	3-4	5	6-8
11	M	P/F	M

M: Bits para las tramas no numeradas.

Campo de Dirección

El campo de dirección identifica a la estación secundaria que ha transmitido o que va a recibir la trama. Este campo no se usa en enlaces punto a punto. El mismo tiene normalmente 8 bits, puede usarse también un formato ampliado en el que la dirección tendrá un múltiplo de 7 bits. El bit menos significativo de cada octeto será respectivamente 1 o 0, si es o no el último octeto del campo de dirección. Los 7 bits restantes de cada octeto formarán la dirección propiamente dicha.

Campo de control

En HDLC se definen tres tipos de tramas, cada una con formato diferente para el campo de control. Las tramas de información (tramas-I) transportan los datos generados por el usuario. En estas tramas también se incluye información para el control ARQ de errores y de flujo. Las tramas de supervisión (tramas-S) proporcionan el mecanismo ARQ cuando la incorporación de las confirmaciones en las tramas-I no es factible. Las tramas no numeradas (Tramas-N) proporcionan funciones complementarias para controlar el enlace.

El primer o los dos primeros bits del campo de control se utilizan para identificar el tipo de trama. El resto de los bits se ubican en subcampos como se indica anteriormente.

Todos los formatos posibles del HDLC contienen el bit sondeo/fin (P/F "poll/final"). Su utilización es dependiente del contexto. Normalmente en las tramas de órdenes se denomina bit P, y se fija a 1 para solicitar (sondear) una respuesta a la entidad HDLC par.

En las tramas de respuesta, el bit se denomina F, y se fija a un valor 1 para identificar a la trama tipo respuesta devuelta tras la recepción de una orden.

Campo de información

El campo de información solo está presente en las tramas-I y en algunas tramas N. Este campo puede contener cualquier secuencia de bits, con la única restricción que el número de bits sea igual a un múltiplo entero de 8. La longitud de este campo es variable y siempre será menor que un valor máximo predefinido.

Campo para la secuencia de comprobación de la trama

La **secuencia de comprobación de la trama** (FCS, *Frame Check Sequence*) es un código para la detección de errores calculado a partir de los bits de la trama excluyendo los delimitadores.

Funcionamiento del HDLC

El funcionamiento del HDLC implica tres fases. Primero, uno de los dos extremos inicia el enlace de datos, de tal manera que las tramas se puedan intercambiar de una forma ordenada. Durante esta fase, se pactan las opciones que se usarán en el intercambio posterior. Después de la iniciación, los dos extremos intercambian los datos generados por los usuarios así como información de control para llevar a cabo los procedimientos de control del flujo y de errores. Finalmente, uno de los dos extremos comunicará la finalización de la transmisión.

Iniciación

La iniciación la puede solicitar cualquiera de los dos extremos transmitiendo una de entre las seis órdenes previstas para fijar el modo. Esta orden sirve para tres objetivos:

1. Se avisa al otro extremo sobre la solicitud de la iniciación.
2. Se especifica cuál de los tres modos (NRM, ABM, ARM) se está solicitando.
3. Se especifica si se van a utilizar números de secuencia de 3 o 7 bits.

Si el otro extremo acepta la solicitud, se informará al extremo sobre esta contingencia mediante la transmisión de una trama de **confirmación no numerada** (UA, *unnumbered acknowledged*). Si la solicitud se rechaza, se envía una trama de **modo desconectado** (DM, *disconnected mode*).

Transferencia de datos

Cuando la iniciación se haya solicitado y haya sido aceptada, entonces se habrá establecido la conexión lógica. A partir de entonces, ambos lados pueden comenzar a enviar datos mediante tramas-I, comenzando con el número de secuencia igual a 0. Los campos N(S) y N(R) de una trama-I contendrán los números de secuencia con los que se lleva a cabo el control del flujo y de errores. La secuencia de tramas-I se numerará secuencialmente módulo 8 o módulo 128, dependiendo de si se utilizan respectivamente 3 o 7 bits, utilizando el campo N(S). El campo N(R) se utiliza para la confirmación de las tramas-I recibidas; de esta forma se facilita que el módulo HDLC indique al otro extremo el número de trama-I que se espera recibir.

Las tramas-S también se usan para controlar el flujo y los errores. La trama **receptor preparado** (RR, *receive ready*) confirma una trama-I recibida, indicando a la vez la siguiente trama-I que se espera recibir. La RR se usa cuando no hay tráfico en el sentido contrario (tramas-I) en el que se puedan incluir las confirmaciones. La trama **receptor no preparado** (RNR, *receive not ready*) confirma una trama-I, como la hace la RR, pero a la vez solicita a la entidad situada al otro extremo del enlace que suspenda la transmisión de tramas-I. Cuando la entidad que envió la RNR este de nuevo preparada, enviará una RR. La trama **REJ** (*rechazo, reject*) sirve para iniciar el procedimiento ARQ (*automatic Repeat-reQuest*) con vuelta-atrás-N. Con ella se indica que la última trama-I recibida se ha rechazado y solicita la retransmisión de todas las tramas-I a partir de la N(R) indicada en la trama REJ. La trama de **rechazo selectivo** (SREJ, *selective reject*) se usa para solicitar la retransmisión de una única trama.

Desconexión

Cualquiera de las dos entidades situadas a ambos lados del enlace pueden iniciar la desconexión; tanto por iniciativa propia (si es que ha habido algún tipo de fallo) como tras la petición cursada por capas superiores. HDLC lleva a cabo la desconexión transmitiendo una trama de **desconexión** (DISC, *disconnect*). El otro extremo podrá aceptar dicha desconexión devolviendo una trama UA e informando al usuario de la capa 3 sobre el cierre de la conexión. Se puede perder cualquier trama-I pendiente de confirmarse, en ese caso su recuperación es responsabilidad de las capas superiores.

PPP

Protocolo punto a punto (PPP, del inglés *Point-to-Point Protocol*), es un protocolo del nivel de enlace de datos, utilizado para establecer una conexión directa entre dos nodos de una red. Conecta dos enrutadores directamente sin ningún equipo u otro dispositivo de red entre medias de ambos. Está estandarizado en el documento RFC1661. Puede proporcionar autenticación, cifrado de la transmisión y compresión.

PPP es usado en varios tipos de redes físicas, incluyendo: cable serial, línea telefónica, línea troncal, telefonía celular, especializado en enlace de radio y enlace de fibra óptica como SONET (*Synchronous Optical Network*). También es utilizado en las conexiones de acceso a Internet. Los proveedores de servicios de Internet (ISP) han usado PPP para que accedan a Internet los usuarios de una línea de conmutación, ya que los paquetes de IP no pueden ser transmitidos vía módem, sin tener un protocolo de enlace de datos.

Dos derivados del PPP son:

- Point-to-Point Protocol over Ethernet (PPPoE),
- Point-to-Point Protocol over ATM (PPPoA).

Son usados comúnmente por los ISP para establecer una **Línea de abonado digital** (*digital subscriber line*, DSL) de servicios de internet para clientes.

Por tanto, se trata de un protocolo asociado a la pila TCP/IP de uso en internet.

Descripción

PPP permite establecer una comunicación a nivel de la capa de enlace TCP/IP entre dos computadoras (redes punto a punto). Generalmente, se utiliza para establecer la conexión a Internet de un computador particular con su ISP a través de un módem telefónico. Ocasionalmente también es utilizado sobre conexiones de banda ancha (como PPPoE o PPPoA). Además del simple transporte de datos, PPP facilita dos funciones importantes:

1. **Autenticación**: generalmente mediante una clave de acceso.
2. **Asignación dinámica de IP**: los proveedores de acceso cuentan con un número limitado de direcciones IP y cuentan con más clientes que direcciones. Naturalmente, no todos los clientes se conectan al mismo tiempo. Así, es posible asignar una dirección IP a cada cliente en el momento en que se conectan al proveedor. La dirección IP se conserva hasta que termina la conexión por PPP. Posteriormente, puede ser asignada a otro cliente.

PPP también tiene otros usos, por ejemplo, se utiliza para establecer la comunicación entre un módem ADSL y la pasarela ATM del operador de telecomunicaciones.

Configuración automática

El **protocolo de control de enlace** (LCP del inglés *Link Control Protocol*) inicia y termina conexiones, permitiendo a los usuarios negociar las opciones de conexión. Es una parte integrada en el PPP, y está definido en el mismo estándar de especificación. LCP provee configuración automática de las interfaces de cada final y selecciona autenticación opcional. LCP corre encima del PPP y se utiliza el valor específico en su campo de protocolo de 0Xc021, y por lo mismo una conexión básica PPP debe estar establecida antes de que se configure LCP.

Múltiples protocolos de la capa de red

PPP permite, a múltiples protocolos de la capa de red, operar en el mismo enlace de comunicación. Para cada protocolo de capa de red usada, un **Protocolo de Control de Red** (NCP, *Network Control Protocol*) separado, ofrece opciones para negociar y encapsular múltiples protocolos de la capa de red. Negocia información de la capa de red como direcciones de red u opciones de compresión, después que la conexión fue establecida.

Opciones de configuración de PPP

PPP puede incluir las siguientes opciones de LCP:

- **Autenticación:** los routers de puerto intercambian mensajes de autenticación. Dos opciones de autenticación son:
 - Protocolo de Autenticación por Clave (PAP).
 - Protocolo de Autenticación de Desafío Mutuo (CHAP).
- **Compresión:** aumenta el rendimiento efectivo en las conexiones PPP, reduciendo la cantidad de data en la trama que debe viajar a través de los enlaces.
- **Detección de Error:** identifica condiciones de falla. La calidad y la opción de Números Mágicos ayudan a asegurar un confiable enlace de datos sin ciclos repetitivos.
- **Multienlace:** proporciona equilibrio de carga de varias interfaces usando el Multilink de PPP.

Funcionamiento

PPP consta de las siguientes fases:

1. **Establecimiento de conexión:** durante esta fase, una computadora contacta con la otra y negocian los parámetros relativos al enlace usando el protocolo LCP. Este protocolo es una parte fundamental de PPP y por ello está definido en el mismo RFC. Usando LCP se negocia el método de autenticación que se va a utilizar, el tamaño de los datagramas, números mágicos para usar durante la autenticación, etc.
2. **Autenticación:** no es obligatorio. Existen dos protocolos de autenticación. El más básico e inseguro es *Password Authentication Protocol* (PAP), aunque no se recomienda dado que envía el nombre de usuario y la contraseña en formato de texto plano. Un método más avanzado y preferido por muchos ISP es **CHAP**, en el cual la contraseña se manda cifrada.
3. **Configuración de red:** en esta fase se negocian parámetros dependientes del protocolo de red que se esté usando. PPP puede llevar muchos protocolos de red al mismo tiempo y es necesario configurar individualmente cada uno de estos protocolos.
4. **Transmisión:** durante esta fase se manda y recibe la información de red. LCP se encarga de comprobar que la línea está activa durante períodos de inactividad. Obsérvese que PPP no proporciona cifrado de datos.
5. **Terminación:** la conexión puede ser finalizada en cualquier momento y por cualquier motivo.

PPP tiene todas las propiedades de un protocolo de nivel de enlace:

- Garantía de recepción.
- Recepción ordenada.
- Usado en los **balanceadores de carga** (*Load Balancer*, LB) como protocolo de distribución.

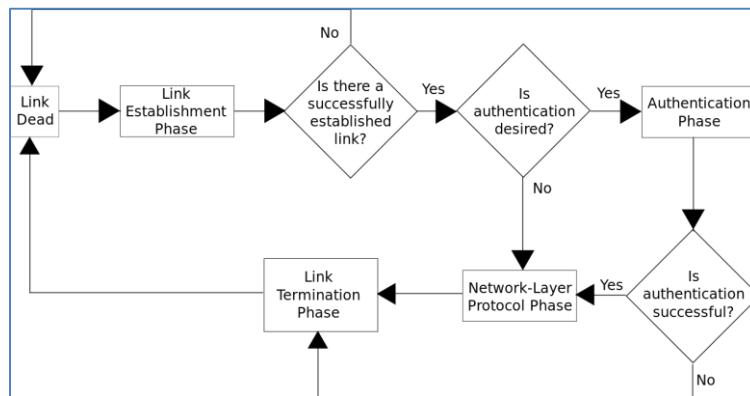


Ilustración 47 - Protocolo PPP

Fases del PPP y activación de la línea

Las fases del protocolo punto a punto, según el RFC 1661, son los siguientes:

1. **Enlace muerto:** esta fase se produce cuando falla la conexión, o en un lado se ha dicho que se desconecte (por ejemplo, un usuario ha terminado su conexión de acceso telefónico).
2. **Fase de establecimiento de enlace:** esta fase es donde se intenta negociar con el Protocolo de Control de Enlace. Si tiene éxito, ya sea de control va a la fase de autenticación o la fase de protocolo de red-capas, dependiendo de si se desea la autenticación.
3. **Fase de autentificación:** esta fase es opcional. Se permite que los lados se autentiquen entre sí antes de que se establezca una conexión. Si se tiene éxito, el control pasa a la fase de protocolo de capa de red.
4. **Fase de protocolo de la capa de enlace:** esta fase es donde se invoca a cada protocolo deseado de los protocolos de control de red. Transporte de datos para todos los protocolos que se iniciaron con éxito con sus protocolos de control de red también se produce en esta fase. Cierre de protocolos de red también se producen en esta fase.
5. **Fase de terminación de enlace:** en esta fase se cierra la conexión. Esto puede ocurrir si hay un error de autenticación, si hay tantos errores de suma de comprobación de que las dos partes deciden derribar el enlace de forma automática, si el enlace falla de repente, o si el usuario decide colgar su conexión.

La subcapa de acceso al medio

Los enlaces de red se pueden dividir en dos categorías: los que utilizan conexiones punto a punto y los que utilizan canales de difusión. Ahora hablaremos sobre 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 tiene 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 al mismo tiempo, lo que conducirá al caos. En una reunión cara a cara, el caos se evita por medios competentes. Por ejemplo, en una reunión la gente levanta la mano para solicitar permiso para hablar. Cuando sólo hay un canal disponible, es mucho más difícil determinar quién debería tener el turno. Se conocen muchos protocolos para resolver el problema. En la literatura, los canales de difusión a veces se denominan **canales multiacceso** o **canales de acceso aleatorio**.

Los protocolos que se utilizan 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**, del inglés *Medium Access Control*). La subcapa MAC tiene especial importancia en las LAN, en especial las inalámbricas puesto que el canal inalámbrico es de difusión por naturaleza. En contraste, las WAN usan generalmente enlaces punto a punto, excepto en las redes satelitales. Debido a que los canales multiacceso y las LAN están muy relacionados, en esta sección analizaremos las LAN en general, además de algunos aspectos que no son estrictamente parte de la subcapa MAC, pero el tema principal será el control del canal.

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. No obstante, para la mayoría 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 un poco de un orden de presentación estrictamente ascendente.

El problema de asignación del canal

El tema central es la forma de asignar un solo canal de difusión entre usuarios competidores. El canal podría ser una parte del espectro inalámbrico en una región geográfica, o un solo alambre o fibra óptica en donde se conectan varios nodos. En ambos casos, el canal conecta a cada usuario con todos los demás; cualquier usuario que utilice todo el canal interfiere con los demás que también desean usarlo.

Primero veremos las deficiencias de los esquemas de asignación estática para el tráfico en ráfagas. Después estableceremos las suposiciones clave que se utilizan para modelar los esquemas dinámicos que examinaremos en las siguientes secciones.

Asignación estática de canal

La manera tradicional de asignar un solo canal, como un troncal telefónico, entre múltiples usuarios competidores es dividir su capacidad mediante el uso de uno de los esquemas de multiplexación que ya vimos, como el **FDM** (Multiplexación por División de Frecuencia, del inglés *Frequency Division Multiplexing*). Si hay N usuarios, el ancho de banda se divide en N partes de igual tamaño, y a cada usuario se le asigna una parte. Debido a que cada usuario tiene una banda de frecuencia privada, ahora no hay interferencia entre ellos. Cuando sólo hay una pequeña cantidad fija y constante de usuarios, cada uno tiene un flujo estable o una carga de tráfico pesada, esta división es un mecanismo de asignación sencillo y eficiente. Las estaciones de radio de FM son un ejemplo inalámbrico. Cada estación recibe una parte de la banda de FM y la utiliza la mayor parte del tiempo para difundir su señal.

Sin embargo, cuando el número de emisores es grande y varía continuamente, o cuando el tráfico se hace en ráfagas, el FDM presenta algunos problemas. Si el espectro se divide en N regiones y actualmente hay menos de N usuarios interesados en comunicarse, se desperdiciará una buena parte del espectro. Y 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.

Aun suponiendo que el número de usuarios podría, de alguna manera, mantenerse constante en N , dividir el único canal disponible en varios subcanales estáticos es inefficiente por naturaleza. 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. Una asignación estática es un mal arreglo para la mayoría de los sistemas de cómputo, en donde el tráfico de datos se presenta en ráfagas muy intensas, a menudo con relaciones de tráfico pico a tráfico medio de 1000:1. En consecuencia, la mayoría de los canales estarán inactivos casi todo el tiempo.

Precisamente los mismos argumentos que se aplican a la FDM se adaptan a otras formas de dividir estáticamente el canal. Si se usara la multiplexación por división de tiempo (TDM) y a cada usuario se le asignara cada N -ésima ranura de tiempo, en caso de que un usuario no utilizara la ranura asignada, simplemente se desperdicia. Lo mismo se aplica si dividimos las redes físicamente.

Ya que ninguno de los métodos tradicionales de asignación estática de canal funciona bien con tráfico en ráfagas, ahora exploraremos los métodos dinámicos.

Supuestos para la asignación dinámica de canales

Antes de entrar en el primero de muchos métodos de asignación de canal que veremos en esta sección, 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. **Tráfico independiente.** El modelo consiste en N estaciones independientes (computadoras, teléfonos), cada una con un programa o usuario que genera tramas para transmisión. El número esperado de tramas que se generan 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 haya transmitido con éxito.
2. **Canal único.** Hay un solo canal disponible para todas las comunicaciones. Todas las estaciones pueden transmitir en él y pueden recibir de él. Se asume que las estaciones tienen una capacidad equivalente, aunque los protocolos pueden asignarles distintos roles (*prioridades*).
3. **Colisiones observables.** Si dos tramas se transmiten en forma simultánea, se traslanan en el tiempo y la señal resultante se altera. Este evento se llama *colisión*. Todas las estaciones pueden detectar una colisión que haya ocurrido. Una trama en colisión se debe volver a transmitir después. No hay otros errores, excepto aquéllos generados por las colisiones.
4. **Tiempo continuo o ranurado.** Se puede asumir que el tiempo es continuo, en cuyo caso la transmisión de una trama puede comenzar en cualquier momento. Por el contrario, el tiempo se puede ranurar o dividir en intervalos discretos (llamados *ranuras*). En este caso las transmisiones de las tramas deben empezar al inicio de una ranura. Una ranura puede contener 0, 1 o más tramas, correspondientes a una ranura inactiva, una transmisión exitosa o una colisión, respectivamente.
5. **Detección de portadora o sin detección de portadora.** Con el supuesto de detección de portadora, las estaciones pueden saber si el canal está en uso antes de intentar usarlo. Si se detecta que el canal está ocupado, ninguna estación intentará utilizarlo. Si no hay 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 llegadas de las tramas son independientes, tanto en todas las estaciones como en una estación específica, y que las tramas se generan en forma impredecible, pero a una tasa de transmisión constante. En realidad este supuesto no es en sí un buen modelo de tráfico de red, pues se sabe que los paquetes llegan en ráfagas durante un rango de escalas de tiempo. Sin embargo, los **modelos de Poisson**, son útiles debido a que son matemáticamente de fácil solución. Estos modelos nos ayudan a analizar protocolos para comprender a grandes rasgos cómo cambia el rendimiento durante un intervalo de operación y cómo se compara con otros diseños.

El supuesto del canal único es la esencia del modelo. No existen formas externas de comunicación. Las estaciones no pueden levantar la mano para solicitar que el maestro les ceda la palabra, por lo que tendremos que idear mejores soluciones.

Los tres supuestos restantes dependen de la ingeniería del sistema; le diremos cuáles supuestos son válidos cuando examinemos un protocolo en particular.

El supuesto de colisión es básico. Las estaciones necesitan una forma de detectar las colisiones si quieren retransmitir las tramas en vez de dejar que se pierdan. En los canales alámbricos, se puede diseñar el hardware de los nodos para detectar las colisiones cuando éstas ocurran. Así, las estaciones pueden terminar sus transmisiones en forma prematura para evitar desperdiciar capacidad. Esta detección es mucho más difícil para los canales inalámbricos, por lo que las colisiones casi siempre se deducen después de que ocurren, debido a que se esperaba una trama de confirmación de recepción y nunca llegó. También es posible que se reciban algunas tramas involucradas en una colisión, dependiendo de los detalles de las señales y del hardware receptor. Pero como esta situación no es el caso común, supongamos que se pierden todas las tramas involucradas en una colisión. También veremos protocolos diseñados sobre todo para evitar que ocurran colisiones.

La razón de las dos suposiciones alternativas sobre el tiempo es que el tiempo ranurado se puede usar para mejorar el rendimiento. Sin embargo, requiere que las estaciones sigan un reloj maestro o que sincronicen sus acciones entre sí para dividir el tiempo en intervalos discretos. Por ende, no siempre está disponible. En este texto estudiaremos y analizaremos sistemas con ambos tipos de supuestos sobre el tiempo. Para un sistema dado, sólo uno de ellos es válido.

De manera similar, una red puede tener detección de portadora o no. Por lo general, las redes alámbricas tienen esta detección de portadora. Las redes inalámbricas no siempre la pueden utilizar de manera efectiva porque tal vez no todas las estaciones estén dentro del rango radial de las demás. Asimismo, la detección de portadora no estará disponible en otros entornos en donde una estación no se pueda comunicar directamente con otra estación, por ejemplo un módem de cable en el cual las estaciones deben comunicarse a través del amplificador de cabecera.

Para evitar cualquier malentendido, debemos tener en cuenta que ningún protocolo multiacceso garantiza una entrega confiable. Aun cuando no haya colisiones, el receptor puede haber copiado alguna trama en forma incorrecta por diversas razones. Otras partes de la capa de enlace o las capas superiores se encargan de proveer confiabilidad.

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 cómo se usan comúnmente en la práctica.

ALOHA

La historia de nuestro primer protocolo MAC empieza en Hawái a principios de la década de 1970. En esta época Hawái no tenía un sistema telefónico funcional. Esto no hizo la vida más placentera para el investigador Norman Abramson y sus colegas de la Universidad de Hawái, quienes trataban de conectar a los usuarios en islas remotas a la computadora principal en Honolulu. Tender sus propios cables bajo el Océano Pacífico no era una opción viable, por lo que buscaron una solución diferente.

La que desarrollaron utilizaba radios de corto rango, en donde cada terminal de usuario compartía la misma frecuencia ascendente para enviar tramas a la computadora central. Incluía un método simple y elegante para resolver el problema de asignación de canal. Desde entonces, su trabajo ha sido extendido por muchos investigadores. 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 cuanto a si el tiempo es continuo, como en la versión pura, o si se divide en ranuras discretas en las que deben caber todas las tramas.

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. Los emisores necesitan alguna forma de saber si éste es el caso. En el sistema ALOHA, después de que cada estación envía su trama a la computadora central, ésta vuelve a difundir la trama a todas las estaciones. Así, una estación emisora puede escuchar la difusión de la estación terrena maestra (hub) para ver si pasó su trama o no. En otros sistemas, como las LAN alámbricas, el emisor podría ser capaz de escuchar si hay colisiones mientras transmite.

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 Ilustración 48 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 transmisión (*throughput*) de los sistemas ALOHA se maximiza al tener tramas con un tamaño uniforme en lugar de tramas de longitud variable.

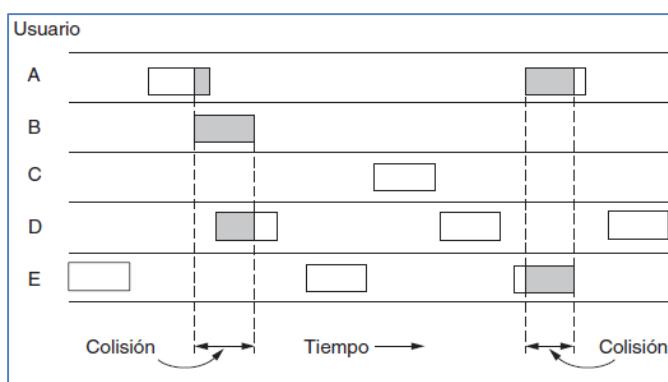


Ilustración 48 - En ALOHA puro, las tramas se transmiten en tiempos completamente arbitrarios.

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 traslape con el último bit de una trama casi terminada, ambas se destruirán por completo (es decir, tendrán sumas de verificación incorrectas) y ambas tendrán que volver a transmitirse más tarde. La suma de verificación no distingue (y no debe) entre una pérdida total y un error ligero.

Hay una pregunta interesante: ¿cuál es la eficiencia de un canal ALOHA? En otras palabras, ¿qué fracción de todas las tramas transmitidas escapa a las colisiones en estas caóticas circunstancias? Primero consideraremos un conjunto infinito de usuarios escribiendo en sus terminales (estaciones). Un usuario siempre está en uno de dos estados: escribiendo o esperando. Al principio todos los usuarios están en el estado de escritura. Al terminar una línea, el usuario deja de escribir, en espera de una respuesta. Después, la estación transmite una trama (que contiene la línea) a través del canal compartido hasta la computadora central 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 mientras la estación transmite la trama una y otra vez hasta que se envía con éxito.

Hagamos que el “tiempo de trama” denote el tiempo necesario para transmitir la trama estándar de longitud fija (es decir, la longitud de la trama dividida entre la tasa de bits). En este punto, suponemos que las tramas nuevas generadas por las estaciones están bien modeladas 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 tasa mayor que la que puede manejar el canal, y casi todas las tramas sufrirán una colisión. Para una velocidad de transmisión razonable esperaríamos que $0 < N < 1$.

Además de las nuevas tramas, las estaciones también generan retransmisiones de tramas que con anterioridad sufrieron colisiones. Supongamos también que las tramas nuevas y antiguas combinadas están bien modeladas según una distribución de Poisson, con una media de G tramas por tiempo de trama. Es evidente que $G \geq N$. 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 transmisión S es sólo la carga ofrecida, G , multiplicada 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 Ilustración 49. ¿Bajo 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 colisionará con el comienzo de la trama sombreada. De hecho, el destino de la trama sombreada está sentenciado 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. Asimismo, cualquier otra trama que se inicie entre t_0+t y t_0+2t chocará con el final de la trama sombreada.

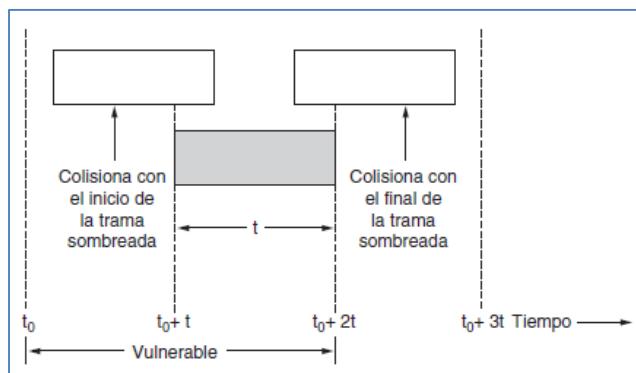


Ilustración 49 - Período vulnerable para la trama sombreada.

La probabilidad de que se generen k tramas durante un tiempo de trama determinado, en donde se esperan G tramas, está dada por la distribución de Poisson.

En la Ilustración 50 se muestra la relación entre el tráfico ofrecido y la velocidad real de transmisión. La máxima velocidad real de transmisión ocurre cuando $G=0.5$, con $S=1/2e$, que es alrededor de 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%.

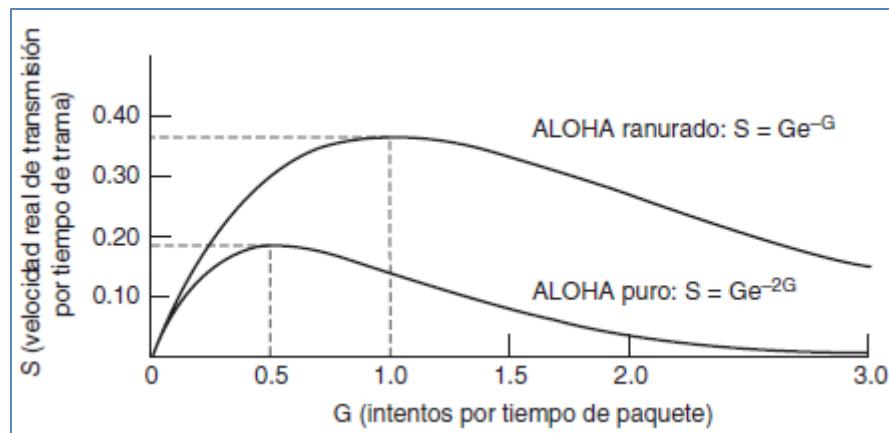


Ilustración 50 - Velocidad real de transmisión contra tráfico ofrecido en los sistemas ALOHA.

ALOHA ranurado

Poco después de que ALOHA apareció en escena, Roberts (1972) publicó un método para duplicar la capacidad de un sistema ALOHA. Su propuesta fue dividir el tiempo en intervalos discretos llamados ranuras, cada uno de los cuales correspondía a una trama. Este método 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 estación envíe cada vez que el usuario escribe una línea. En cambio, se le obliga a esperar el comienzo de la siguiente ranura. Por lo tanto, el ALOHA de tiempo continuo se convierte en uno de tiempo discreto. Esto reduce el periodo vulnerable a la mitad. Para ver esto, analice la Ilustración 49 e imagine las colisiones que puede haber ahora.

Como podemos ver en la Ilustración 50, el ALOHA ranurado alcanza su máximo valor en $G=1$, con una velocidad real de transmisión 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. Lo mejor que podemos esperar usando ALOHA ranurado es 37% de ranuras vacías, 37% de éxitos y 26% de colisiones. Si se opera con valores mayores de G se reduce el número de ranuras vacías pero aumenta de manera exponencial el número de colisiones.

Como resultado de la dependencia exponencial del número esperado de transmisiones 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 la década de 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. El ALOHA ranurado prácticamente se sacó del cesto de la basura para resolver el problema. Posteriormente, hacer que varias etiquetas RFID se comunicaran con el mismo lector RFID presentó otra variación del mismo problema. De nuevo salió al rescate el ALOHA ranurado, con unas cuantas ideas más mezcladas. 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) o debido a las tendencias siempre cambiantes de la tecnología. Después, años más tarde alguna persona astuta se dio cuenta de que un protocolo descartado por mucho tiempo es el que podía 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. Por supuesto, también estudiaremos varios protocolos que se utilizan en la actualidad.

Protocolos de acceso múltiple con detección de portadora

Con el ALOHA ranurado, el mejor aprovechamiento de canal que se puede lograr es aproximadamente del 37%. Este resultado tan bajo no es muy 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 LAN 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. En esta sección estudiaremos algunos protocolos para mejorar su desempeño.

Los protocolos en los que las estaciones escuchan una portadora (es decir, una transmisión) y actúan de manera acorde se llaman protocolos de **detección de portadora**. Se han propuesto varios de ellos y hace mucho tiempo se analizaron con detalle. 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, del inglés *Carrier Sense Multiple Access*) **persistente-1**. Es un nombre bastante largo para el esquema CSMA más simple. Cuando una estación tiene datos por enviar, primero escucha el canal para saber si alguien más está transmitiendo en ese momento. Si el canal está inactivo, la estación envía sus datos. Por el contrario, si el canal está ocupado, la estación espera hasta que se desocupa. A continuación, la estación 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.

Podría esperarse que este esquema evite las colisiones, excepto en el extraño caso de los envíos simultáneos, pero de hecho no lo hace. Si dos estaciones están listas a la mitad de la transmisión de una tercera estación, ambas esperarán amablemente hasta que termine la transmisión y después ambas empezarán a transmitir

exactamente al mismo tiempo, lo cual producirá una colisión. Si no fueran tan impacientes, habría menos colisiones.

Otro aspecto delicado es que el retardo de propagación tiene un efecto importante sobre las colisiones. Existe la posibilidad de que, justo después de que una estación comienza a transmitir, otra estación esté lista para enviar y detecte el canal. Si la señal de la primera estación no ha llegado aún a la segunda, esta última detectará un canal inactivo y comenzará también a enviar, lo que dará como resultado una colisión. Esta posibilidad depende del número de tramas que quepan en el canal, o **producto de ancho de banda-retardo** del canal. Si sólo cabe una pequeña fracción de una trama en el canal, lo cual es cierto en la mayoría de las redes LAN, ya que el retardo de propagación es pequeño, la posibilidad de que ocurra una colisión es pequeña. Cuanto mayor sea el producto de ancho de banda-retardo, más importante será este efecto y peor el desempeño del protocolo.

Aun así, este protocolo tiene un mejor desempeño que el ALOHA puro, ya que ambas estaciones tienen la decencia de dejar de interferir con la trama de la tercera estación. Lo mismo se aplica en el ALOHA ranurado.

Un segundo protocolo de detección de portadora es el **CSMA no persistente**. Como antes, una estación escucha el canal cuando desea enviar una trama y, si nadie más está transmitiendo, comienza a hacerlo. Pero si el canal ya está en uso, la estación no lo escuchará de manera continua con el fin de tomarlo de inmediato al detectar el final de la transmisión anterior, sino que esperará un periodo aleatorio y repetirá 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 se encuentra inactivo, la estación transmite con una probabilidad p . Con una probabilidad $q=1-p$, se posterga hasta la siguiente ranura. Si esa ranura también está inactiva, la estación transmite o posterga una vez más, con probabilidades p y q . Este proceso se repite hasta que se transmite la trama o hasta que otra estación comienza a transmitir. En el segundo caso, la desafortunada estación actúa como si hubiera ocurrido una colisión (es decir, espera un tiempo aleatorio y comienza de nuevo). Si al principio la estación detecta que el canal está ocupado, espera hasta la siguiente ranura y aplica el algoritmo anterior. El estándar IEEE 802.11 usa una versión refinada del CSMA persistente-p.

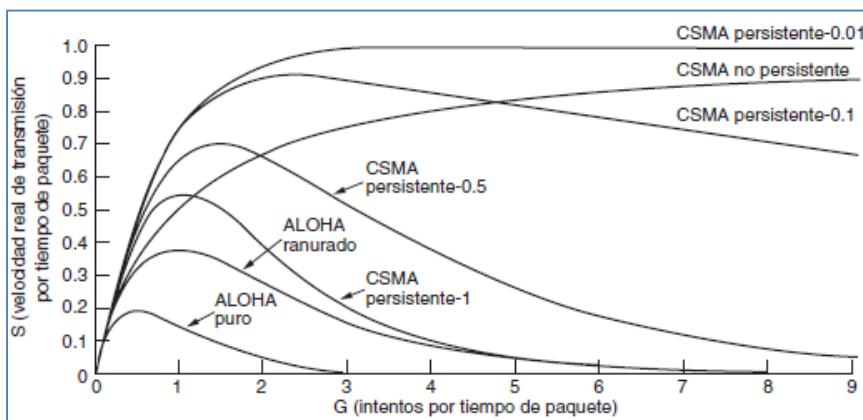


Ilustración 51 - Comparación de la utilización del canal contra la carga para varios protocolos de acceso aleatorio.

En la Ilustración 51 se muestra la velocidad real de transmisión calculada contra el tráfico ofrecido para los tres protocolos, así como para el ALOHA puro y el ranurado.

CSMA con detección de colisiones

En definitiva, los protocolos CSMA persistentes y no persistentes son una mejora respecto a ALOHA porque aseguran que ninguna estación empezará a transmitir mientras el canal esté ocupado. Pero si dos estaciones detectan que el canal está inactivo y empiezan a transmitir al mismo tiempo, sus señales de todas formas sufrirán una colisión. Otra mejora es que las estaciones detecten rápidamente la colisión y dejen de transmitir

de inmediato (en vez de terminadas las transmisiones), ya que de todas formas se alterarán y no se podrán recuperar. Esta estrategia ahorra tiempo y ancho de banda.

Este protocolo, conocido como **CSMA/CD** (CSMA con Detección de Colisiones, del inglés *CSMA with Collision Detection*), es la base de la clásica LAN Ethernet. Es importante tener en cuenta que la detección de colisiones es un proceso analógico. El hardware de la estación debe escuchar el canal mientras transmite. Si la señal que recibe es distinta de la señal que está enviando, sabe que está ocurriendo una colisión. Las implicaciones son que una señal recibida no debe ser pequeña en comparación con la señal transmitida (lo cual es difícil en las redes inalámbricas, ya que las señales recibidas pueden ser miles de veces más débiles que las señales transmitidas) y que la modulación se debe elegir de modo que permita detectar colisiones (por ejemplo, tal vez sea imposible detectar una colisión de dos señales de 0 volts).

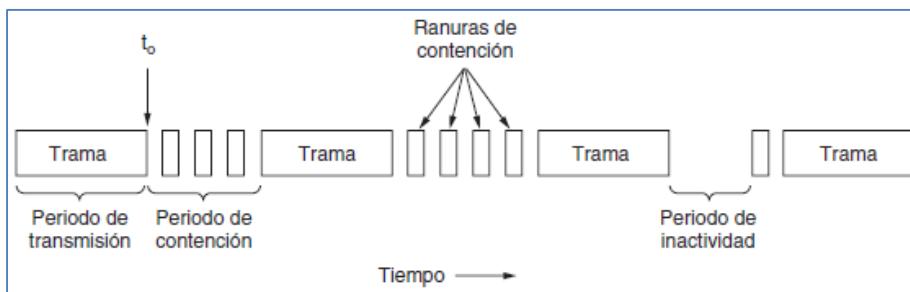


Ilustración 52 - CSMA/CD puede estar en estado de contención, de transmisión o inactivo.

Al igual que muchos otros protocolos de LAN, CSMA/CD utiliza el modelo conceptual de la Ilustración 52. En el punto marcado como t_0 , una estación ha terminado de transmitir su trama. Cualquier otra estación que tenga una trama por enviar puede intentar hacerlo ahora. Si dos o más estaciones deciden transmitir en forma simultánea, habrá una colisión. Si 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, con períodos de inactividad que ocurrirán cuando todas las estaciones estén 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 exactamente en el momento t_0 . ¿En cuánto tiempo se darán cuenta de que ha ocurrido una colisión? La respuesta a esta pregunta es vital para determinar la longitud del período de contención y, por lo tanto, el retardo y la velocidad real de transmisión.

El tiempo mínimo para detectar la colisión es tan sólo el tiempo que tarda la señal en propagarse de una estación a otra. Con base en esta información, podríamos pensar que una estación que no ha detectado una colisión durante un período igual al tiempo completo de propagación del cable después de iniciar su transmisión puede estar segura de que ha tomado el cable. Por “tomado” queremos decir que todas las demás estaciones saben que está transmitiendo y no interferirán. Esta conclusión es errónea.

Considere el siguiente escenario en el 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 $t_0 + \tau - \epsilon$, un instante antes de que la señal llegue a la estación más lejana, esa estación también comienza a transmitir. Por supuesto que 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, sino hasta el tiempo $2\tau - \epsilon$. 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.

Con este razonamiento, podemos pensar en la contención de CSMA/CD como un sistema ALOHA ranurado con un ancho de ranura de 2τ . En un cable coaxial de 1 km de longitud, $\tau \approx 5 \mu\text{seg}$. La diferencia para CSMA/CD en comparación con ALOHA ranurado es que las ranuras en las que sólo transmite una estación (por ejemplo, la que tomó el canal) van seguidas del resto de una trama. Esta diferencia mejorará en forma considerable el desempeño si el tiempo de la trama es mucho mayor que el tiempo de propagación.

Protocolos libres de colisiones

Aunque las colisiones no ocurren en CSMA/CD una vez que una estación ha capturado el canal sin ambigüedades, aún pueden ocurrir durante el periodo de contención. Estas colisiones afectan en forma adversa el desempeño del sistema, en especial cuando el producto ancho de banda-retardo es grande, como cuando el cable es largo (es decir, τ es grande) y las tramas son cortas. Las colisiones no sólo reducen el ancho de banda, sino que también hacen variable el tiempo de envío de una trama, lo cual no es bueno para el tráfico en tiempo real tal como la voz sobre IP. Además, CSMA/CD no se puede aplicar en forma universal.

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 estos protocolos no se utilizan en los sistemas grandes, pero en un campo en constante cambio, el hecho de tener algunos protocolos con buenas propiedades disponibles para sistemas futuros es con frecuencia algo bueno.

En los protocolos que describiremos supondremos que hay exactamente N estaciones, cada una programada con una dirección única de 0 a $N-1$. No importa el hecho de que algunas estaciones puedan estar inactivas una parte del tiempo. También damos por hecho que el retardo de propagación es insignificante. La pregunta básica persiste: ¿qué estación obtiene el canal después de una transmisión exitosa? Seguimos usando el modelo de la Ilustración 52 con sus ranuras de contención discretas.

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 exactamente de 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 bit 1 durante la ranura 1, pero sólo si tiene una trama puesta en la cola. En general, la estación j puede anunciar que tiene una trama por enviar, para lo cual inserta un bit 1 en la ranura j . Una vez que han pasado las N ranuras, cada estación tiene un completo conocimiento acerca de cuáles son las estaciones que quieren transmitir. En ese punto, las estaciones empiezan a transmitir en orden numérico (Ilustración 53).

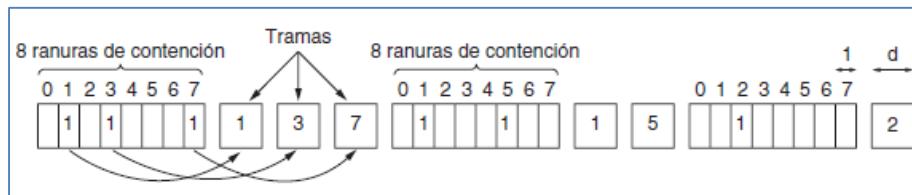


Ilustración 53 - El protocolo básico de mapa de bits.

Como todos están de acuerdo en quién sigue a continuación, nunca habrá colisiones. Una vez que la última estación lista haya transmitido su trama, un evento que pueden detectar fácilmente todas las estaciones, comienza otro periodo de contención de N bits. Si una estación está lista justo después de que ha pasado su ranura de bit, ha tenido mala suerte y deberá permanecer inactiva hasta que cada una de las demás estaciones haya tenido su oportunidad y el mapa de bits haya comenzado de nuevo.

Los protocolos como éste en los que el interés de transmitir se difunde antes de la transmisión se llaman **protocolos de reservación**, debido a que reservan la propiedad del canal por anticipado y evitan colisiones. Analicemos brevemente el desempeño de este protocolo. Por conveniencia, mediremos el tiempo en unidades de la ranura de bit 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 menor numeración, como 0 o 1. Por lo general, cuando la estación está 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 escaneo actual termine, además de otras N ranuras para que el siguiente escaneo se ejecute hasta su terminación, antes de que pueda empezar a transmitir.

Las posibilidades para las estaciones de mayor numeración son mejores. En general, éstas sólo tendrán que esperar la mitad de un escaneo ($N/2$ ranuras de bits) antes de comenzar a transmitir. Las estaciones de mayor numeración pocas veces tienen que esperar el siguiente escaneo. Dado que las estaciones de menor numeración deben esperar en promedio $1.5N$ ranuras y las estaciones de mayor numeración 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, lo cual nos da una eficiencia de $d/(d+N)$.

Si la carga es alta y todas las estaciones tienen algo que enviar todo el tiempo, el periodo de contención de N bits se prorrtea entre N tramas, lo cual produce una sobrecarga de sólo 1 bit por trama, o una eficiencia de $d/(d+1)$. El retardo promedio de una trama es igual a la suma del tiempo que está en cola en su estación, más un $(N+1)d+N$ adicional una vez que llega a la cabeza de su cola interna. Este intervalo indica cuánto tiempo hay que esperar a que las demás estaciones tomen su turno para enviar una trama y otro mapa de bits.

Paso de token

La esencia del protocolo de mapa de bits es que permite que cada estación transmita una trama por turno, en un orden predefinido. Otra forma de lograr lo mismo es pasar un pequeño mensaje conocido como **token** de una estación a otra, en el mismo orden predefinido. El token representa el permiso para enviar. Si una estación tiene una trama puesta en cola para transmitirla cuando recibe el token, puede enviar esa trama antes de pasar el token a la siguiente estación. Si no tiene una trama puesta en cola, simplemente pasa el token.

En un protocolo **token ring**, la topología de la red se utiliza para definir el orden en el que las estaciones envían información. Las estaciones están conectadas una con otra en un solo anillo. Así, el proceso de pasar el token a la siguiente estación consiste en recibir el token proveniente de una dirección y transmitirlo hacia la otra dirección, como podemos ver en la Ilustración 54. Las tramas también se transmiten en la dirección del token. De esta forma, circularán alrededor del anillo y llegarán a la estación de destino. Sin embargo, para evitar que la trama circule en forma indefinida (como el token), una estación necesita quitarla del anillo. Esta estación puede ser la que envió originalmente la trama, después de que haya pasado por un ciclo completo, o la estación destinada a recibir la trama.

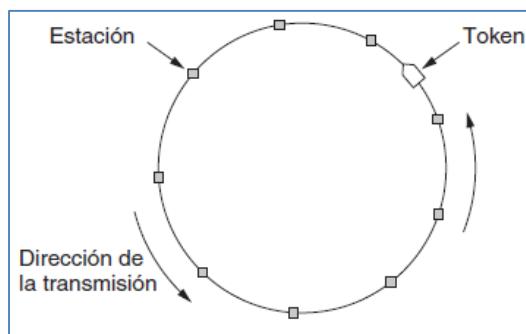


Ilustración 54 - El protocolo token ring.

Cabe mencionar que no necesitamos un anillo físico para implementar el paso del token. El canal que conecta a las estaciones podría ser también un solo bus extenso. Así, cada estación puede usar el bus para enviar el token a la siguiente estación en la secuencia predefinida. Al poseer el token, una estación puede usar el bus para enviar una trama, como antes. A este protocolo se le conoce como **token bus**.

El desempeño del protocolo de paso de token es similar al del protocolo de mapa de bits, aunque las ranuras de contención y las tramas de un ciclo están ahora entremezcladas. Después de enviar una trama, cada estación debe esperar a que las N estaciones (incluyéndose a sí misma) envíen el token a sus estaciones vecinas y que las otras $N-1$ estaciones envíen una trama, si es que la tienen. Una sutil diferencia es que, como todas las posiciones en el ciclo son equivalentes, no hay parcialidad por las estaciones de menor o de mayor numeración. Para token ring, cada estación también envía el token sólo hasta su estación vecina antes de que el protocolo lleve a cabo el siguiente paso. No es necesario propagar cada token a todas las estaciones antes de que el protocolo avance al siguiente paso.

Las redes de token ring han surgido como protocolos MAC con cierta consistencia. Uno de los primeros protocolos de este tipo (conocido como “Token Ring”, que se estandarizó como IEEE 802.5) fue popular en la década de 1980 como alternativa a la Ethernet clásica. En la década de 1990, una red token ring mucho más veloz conocida como **FDDI** (Interfaz de Datos Distribuidos por Fibra, del inglés *Fiber Distributed Data Interface*) fue vencida por la Ethernet comutada. En la década de 2000, una red token ring llamada **RPR** (Anillo de Paquetes con Recuperación, del inglés *Resilient Packet Ring*) se definió como el IEEE 802.17 para estandarizar la mezcla de anillos de área metropolitana que usaban los ISP.

Conteo descendente binario

Un problema con el protocolo básico de mapa de bits, y en consecuencia con el paso de token, 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 si usamos direcciones de estación binarias con un canal que combine las transmisiones. Una estación que quiere utilizar el canal en un momento dado difunde su dirección como una cadena binaria de bits, comenzando por el bit de mayor orden. Se supone que todas las direcciones tienen la misma longitud. A todos los bits en cada posición de dirección de las diferentes estaciones se les aplica un OR BOLEANO por el canal cuando se envían al mismo tiempo. A este protocolo lo llamaremos **conteo descendente binario**. Asume de manera implícita que los retardos de transmisión son insignificantes, de manera que todas las estaciones ven los bits instantáneamente.

Para evitar conflictos, es necesario aplicar una regla de arbitraje: tan pronto como una estación ve que una posición de bit de orden alto, cuya dirección es 0, ha sido sobreescrita 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 mayor numeración está compitiendo por el canal, por lo que se dan por vencidas durante esta ronda. Las estaciones 1001 y 1010 continúan.

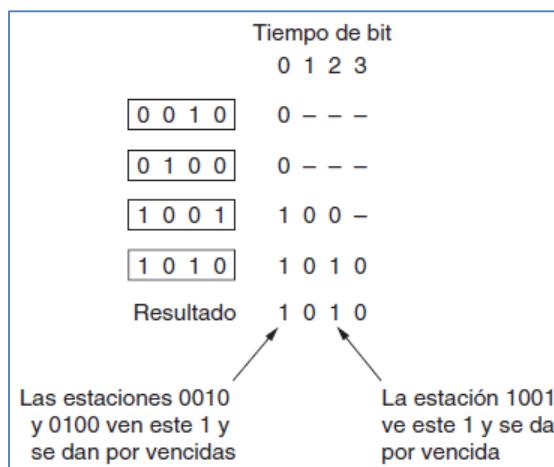


Ilustración 55 - El protocolo de conteo descendente binario. Un guión indica un silencio.

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 más alta. Después de ganar la contienda, ahora puede transmitir una trama, después de lo cual comienza otro ciclo de contienda. El protocolo se ilustra en la Ilustración 55. Tiene la propiedad de que estaciones con mayor numeración tienen una prioridad más alta que las estaciones con menor numeración, lo cual puede ser bueno o malo, dependiendo del contexto.

La eficiencia de canal de este método es de $d/(d+\log_2 N)$. Pero si el formato de trama se escoge ingeniosamente de modo que la dirección del emisor sea el primer campo en la trama, ni siquiera estos $\log_2 N$ bits se desperdician y la eficiencia es del 100%.

Protocolos de contención limitada

Hasta ahora hemos considerado dos estrategias básicas para adquirir el canal en una red de difusión: los protocolos de contención, como el CSMA, y los protocolos libres de colisión. Cada estrategia se puede recomendar según lo bien que funciona en relación con las dos medidas importantes de desempeño: el retardo con carga baja y la eficiencia del canal con carga alta. En condiciones de carga ligera, la contención (es decir, ALOHA puro o ranurado) es preferible debido a su bajo retardo (ya que las colisiones son raras). A medida que aumenta la carga, la contención se vuelve cada vez 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 (ya que las sobrecargas son fijas).

Sin duda, sería agradable si pudiéramos combinar las mejores propiedades de los protocolos de contención y los libres de colisiones, para idear un nuevo protocolo que usara contención cuando la carga fuera baja y con ello tener un retardo bajo, así como 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 son con los que concluiremos 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 cierta probabilidad, p , y todas las estaciones usan la misma p . Resulta interesante que el desempeño general del sistema se pueda mejorar a veces mediante el uso de un protocolo que asigne diferentes probabilidades a distintas estaciones.

Antes de analizar los protocolos asimétricos, demos un breve repaso al desempeño del caso simétrico. Suponga que hay k estaciones que compiten por el acceso al canal. Cada estación 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 la probabilidad de que cualquier otra estación transmita, con probabilidad p , y que las otras $k-1$ estaciones posterguen su transmisión, cada una con una probabilidad de $1-p$. Este valor es $kp(1-p)^{k-1}$. Para encontrar el valor óptimo de p , diferenciamos con respecto a p , igualamos el resultado a cero y despejamos p . Al hacer esto, encontramos que el mejor valor de p es $1/k$.

En la Ilustración 56 se grafica la probabilidad de éxito para p óptima. Para un número pequeño de estaciones, la posibilidad de éxito es buena, pero tan pronto como la cantidad de estaciones llega a cinco, la probabilidad disminuye hasta una cifra cercana a su valor asintótico de 37%.

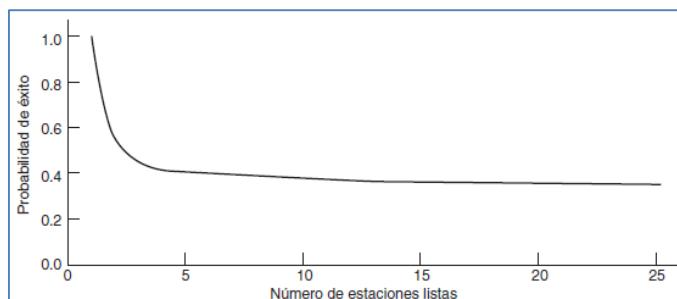


Ilustración 56 - Probabilidad de adquisición de un canal de contención simétrica.

En la Ilustración 56 es bastante evidente que la probabilidad de que una estación adquiera el canal sólo puede aumentar si disminuye 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 competirán por la ranura 1, etcétera. Al dividir en forma adecuada las estaciones en grupos, es posible reducir la cantidad de contenciones para cada ranura y, en consecuencia, se puede operar cada ranura cerca de la parte izquierda de la Ilustración 56.

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 a lo más sólo una estación competirá por una ranura dada. Ya hemos visto tales

protocolos (por ejemplo, el conteo descendente binario). El siguiente caso especial es asignar dos estaciones por grupo. La probabilidad de que ambas intenten transmitir durante una ranura es p^2 , que para una p pequeña es insignificante. A medida que se asignan cada vez más estaciones a la misma ranura, aumenta la probabilidad de colisión pero disminuye la longitud del escaneo del mapa de bits necesaria para dar a todos una oportunidad. El caso límite es un solo grupo que contenga todas las estaciones (es decir, ALOHA ranurado). Lo que necesitamos es una manera de asignar dinámicamente 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.

El protocolo de recorrido de árbol adaptable

Una manera muy 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. En esencia, el ejército tomaba una muestra de sangre de N soldados. Se vaciaba una parte de cada muestra en un solo tubo de ensayo. Luego se examinaba esta muestra mezclada en busca de anticuerpos. Si no se encontraban, todos los soldados del grupo se declaraban sanos. Si se encontraban anticuerpos, se preparaban dos nuevas muestras mezcladas, una de los soldados 1 a $N/2$ y otra de los demás. El proceso se repetía en forma recursiva hasta que se determinaban los soldados infectados.

Para la versión de computadora de este algoritmo es conveniente considerar a las estaciones como hojas de un árbol binario, como se muestra en la Ilustración 57. En la primera ranura de contención después de la transmisión exitosa 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 queden bajo el nodo 2 del árbol podrán competir. Si alguna de ellas adquiere el canal, la ranura que siga después de la trama se reservará 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 será el turno del nodo 4 durante la ranura 2.

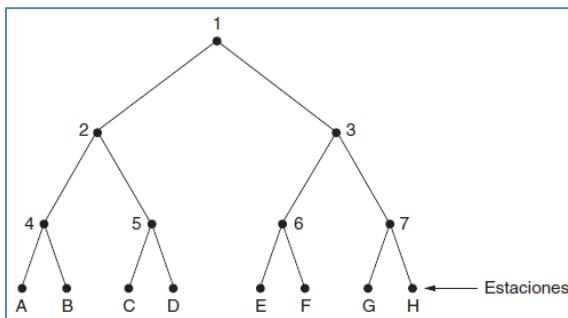


Ilustración 57 - El árbol para ocho estaciones.

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 específico del árbol. Si ocurre una colisión, continúa la búsqueda en forma recursiva con el hijo izquierdo y el derecho del nodo. Si una ranura de bits está inactiva o si sólo una estación que transmite en ella, se puede detener la búsqueda de su nodo, ya que 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 si vale la pena dedicarle la ranura 0 al nodo 1, porque eso sólo tiene sentido en el caso poco probable de que haya precisamente una estación que tenga una trama por enviar. Asimismo, podríamos argumentar que sería conveniente saltar los nodos 2 y 3 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, mediante una supervisión del tráfico reciente.

Para proceder, vamos a numerar los niveles del árbol desde arriba, con el nodo 1 de la Ilustración 57 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 de sólo $2^{-i}q$. Instintivamente esperaríamos que el nivel

óptimo para comenzar a examinar al árbol fuera aquel cuyo número promedio de estaciones contendientes por ranura sea 1; es decir, el nivel en el que $2^i q = 1$. Al resolver esta ecuación, encontramos que $i = \log_2 q$.

Se han descubierto muchas mejoras al algoritmo básico. 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). Podemos omitir la prueba de 3 para intentar con el nodo 6. Al no arrojar nada esta prueba, también podemos omitir el nodo 7 para intentar el nodo G después.

Protocolos de LAN inalámbrica

Un sistema de computadoras portátiles que se comunican por radio se puede considerar una LAN inalámbrica. Este tipo de LAN es un ejemplo de un canal de difusión. Además, tiene propiedades un tanto diferentes que la LAN alámbrica, por lo que requiere distintos protocolos MAC.

Una configuración común para una LAN inalámbrica es un edificio de oficinas con puntos de acceso (AP) ubicados de manera estratégica alrededor del edificio. Los AP están interconectados mediante cobre o fibra, y proveen conectividad a las estaciones que se comunican con ellos. Si la potencia de transmisión de los AP y las computadoras portátiles se ajusta de modo que tenga un alcance de decenas de metros, entonces los cuartos cercanos se convierten en una celda única y el edificio entero se convierte en algo así como el sistema de telefonía celular, excepto que cada celda sólo tiene un canal. Todas las estaciones en la celda comparten este canal, incluyendo el AP. Por lo general proporciona anchos de banda de varios megabits/s, hasta 600 Mbps.

Ya hemos recalcado de antemano que los sistemas inalámbricos no pueden por lo general detectar una colisión al momento en que ocurre. La señal recibida en una estación puede ser débil, tal vez un miles de veces más tenue que la señal transmitida. Encontrarla es como buscar una aguja en un pajar. En vez de ello se utilizan confirmaciones de recepción para descubrir las colisiones y otros errores después de que suceden.

Incluso hay una diferencia aún más importante entre las redes LAN inalámbricas y las LAN alámbricas (cableadas). Tal vez una estación en una LAN inalámbrica no pueda transmitir ni recibir tramas de todas las demás estaciones debido al rango de radio limitado de éstas. En las redes LAN alámbricas, cuando una estación envía una trama, todas las demás estaciones la reciben. La ausencia de esta propiedad en las redes LAN inalámbricas provoca una variedad de complicaciones.

En nuestros siguientes análisis haremos la suposición de simplificación de que cada transmisor de radio tiene cierto rango físico, el cual se representa mediante una región de cobertura circular dentro de la cual otra estación puede detectar y recibir la transmisión de la estación emisora. Es importante darse cuenta de que en la práctica, las regiones casi nunca son tan regulares debido a que la propagación de las señales de radio depende del entorno. Las paredes y otros obstáculos que atenúan y reflejan señales pueden hacer que el alcance difiera de manera considerable en distintas direcciones. Pero un simple modelo circular bastará para nuestros propósitos.

Un enfoque inicial para usar una LAN inalámbrica podría ser probar con CSMA: escuchar si hay otras transmisiones y sólo transmitir si nadie más lo está haciendo. El problema radica en que este protocolo no es en realidad una buena manera de pensar en lo inalámbrico, ya que lo que importa en la recepción es la interferencia en el receptor, no en el emisor. Para ver la naturaleza de este problema, considere la Ilustración 58, en la que se ilustran cuatro estaciones inalámbricas. Para nuestros fines, no importa cuáles son AP ni cuáles son computadoras portátiles. El alcance de radio es tal que A y B están en el mismo alcance y es probable que puedan interferir entre sí. C también podría interferir con B y con D , pero no con A .

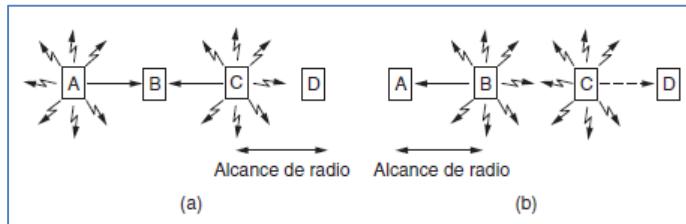


Ilustración 58 - Una LAN inalámbrica. (a) A y C son terminales ocultas cuando transmiten a B. (b) B y C son terminales expuestas cuando transmiten a A y D.

Primero considere lo que ocurre cuando A y C transmiten hacia B, como se muestra en la Ilustración 58(a). Si A envía y C detecta el medio de inmediato, no podrá escuchar a A porque está fuera de su alcance. Por lo tanto, C concluirá falsamente que puede transmitir a B. Si C comienza a transmitir, interferirá en B, eliminando la trama de A (en este caso suponemos que no se utiliza un esquema tipo CDMA para proveer múltiples canales, por lo que las colisiones alteran la señal y destruyen ambas tramas). Este problema se denomina **problema de terminal oculta**.

Ahora consideremos una situación distinta: B transmite a A al mismo tiempo que C desea transmitir a D, como se muestra en la Ilustración 58(b). Si C detecta el medio, escuchará una transmisión y concluirá equivocadamente que no puede enviar a D (lo cual se muestra con una línea punteada). De hecho, esa transmisión provocaría una mala recepción sólo en la zona entre B y C, en donde no hay ninguno de los receptores deseados. A esta situación se le denomina **problema de terminal expuesta**.

El problema es que antes de comenzar una transmisión, una estación realmente quiere saber si hay actividad o no alrededor del receptor. El CSMA simplemente le indica si hay o no actividad cerca del transmisor mediante la detección de la portadora. Con un cable, todas las señales se propagan a todas las estaciones, por lo que esta distinción no existe. Sin embargo, 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. Queremos que esta concurrencia ocurra a medida que aumenta el tamaño de la celda, de la misma forma que las personas en una fiesta no deben esperar a que todos en la habitación hagan silencio para poder hablar; puede haber múltiples conversaciones a la vez en un cuarto grande, siempre y cuando no estén dirigidas hacia la misma ubicación.

Uno de los primeros protocolos influyentes que aborda estos problemas para las redes LAN inalámbricas es **MACA** (Acceso Múltiple con Prevención de Colisiones, del inglés *Multiple Access with Collision Avoidance*) (Karn, 1990). El concepto en que se basa es que el emisor estimule al receptor para que envíe 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). Se utiliza esta técnica en vez de la detección de portadora.

El MACA se muestra en la Ilustración 59. Consideremos ahora la manera en que A envía una trama a B. A comienza enviando una trama **RTS** (Solicitud de Envío, del inglés *Request To Send*) a B, como se muestra en la Ilustración 59(a). Esta trama corta (30 bytes) contiene la longitud de la trama de datos que seguirá después. Después B contesta con una trama **CTS** (Libre para Envío, del inglés *Clear To Send*), como se muestra en la Ilustración 59(b). La trama CTS contiene la longitud de los datos (que copia de la trama RTS). Al recibir la trama CTS, A comienza a transmitir.

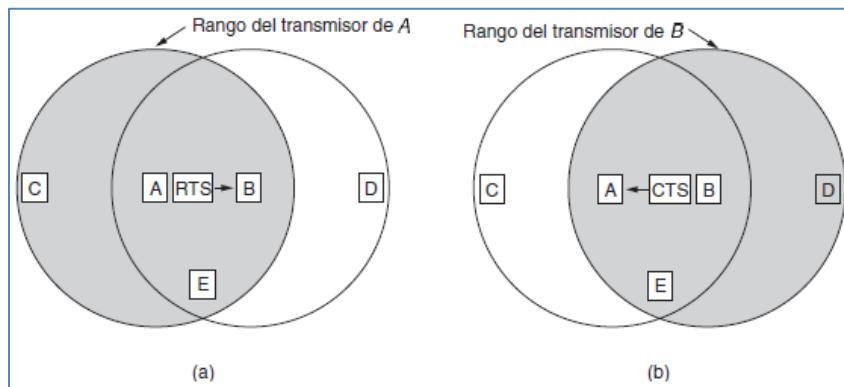


Ilustración 59 - El protocolo MACA. (a) A envía un RTS a B. (b) B responde con un CTS a A.

Ahora veamos cómo reaccionan las estaciones que escuchan cualquiera de estas tramas. Sin duda, cualquier estación que escuche el RTS 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. Es evidente que cualquier estación que escuche el CTS está bastante cerca de B y debe permanecer en silencio durante la siguiente transmisión de datos, cuya longitud puede determinar examinando la trama CTS.

En la Ilustración 59, C está en el alcance de A pero no en el alcance de B. Por lo tanto, escucha el RTS de A pero no el CTS de B. En tanto no interfiera con el CTS, está libre para transmitir mientras se envía 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 sabe 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 podrían enviar tramas RTS a A al mismo tiempo. Éstas chocarán y se perderán. En el caso de una colisión, un transmisor sin éxito (es decir, uno que no escucha un CTS en el intervalo esperado) espera un tiempo aleatorio y vuelve a intentar más tarde.

ETHERNET

Hemos finalizado 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. Muchos de los diseños para las redes personales, locales y de área metropolitana se han estandarizado bajo el nombre de IEEE 802. Algunos han sobrevivido pero muchos no. Los sobrevivientes más importantes son el 802.3 (Ethernet) y el 802.11 (LAN inalámbrica). Bluetooth (PAN inalámbrica) se utiliza mucho en la actualidad, pero se estandarizó fuera del 802.15.

Empezaremos nuestro estudio de los sistemas reales con Ethernet, que probablemente sea el tipo más ubicuo de red de computadoras en el mundo. Existen dos tipos de Ethernet: **Ethernet clásica**, que resuelve el problema de acceso múltiple mediante el uso de las técnicas que hemos analizado; el segundo tipo es la **Ethernet conmutada**, en donde los dispositivos llamados switchs se utilizan para conectar distintas computadoras. Es importante mencionar que, aunque se hace referencia a ambas como Ethernet, son muy diferentes. La Ethernet clásica es la forma original que operaba a tasas de transmisión de 3 a 10 Mbps. La Ethernet conmutada es en lo que se convirtió la Ethernet y opera a 100, 1000 y 10000 Mbps, en formas conocidas como FastEthernet, GigabitEthernet y 10GigabitEthernet. Actualmente, en la práctica sólo se utiliza Ethernet conmutada.

Analizaremos estas formas históricas de Ethernet en orden cronológico para mostrar cómo se desarrollaron. Puesto que Ethernet y el IEEE 802.3 son idénticos, excepto por una pequeña diferencia (que veremos en breve), muchas personas usan los términos "Ethernet" e "IEEE 802.3" sin distinción. Nosotros también lo haremos.

Capa física de Ethernet clásica

La historia de Ethernet empieza casi al mismo tiempo que ALOHA, cuando un estudiante llamado Bob Metcalfe obtuvo su licenciatura en el MIT y después obtuvo su doctorado en Harvard. Durante sus estudios oyó hablar del trabajo de Abramson. Se interesó tanto en él que, después de graduarse de Harvard, decidió pasar el verano en Hawái trabajando con Abramson antes de empezar a trabajar en Xerox PARC (*Palo Alto Research Center*). Cuando llegó a PARC, vio que los investigadores ahí habían diseñado y construido lo que después se conocería como computadora personal. Pero las máquinas estaban aisladas. Haciendo uso de su conocimiento sobre el trabajo de Abramson, junto con su colega David Boggs diseñó e implementó la primera red de área local. Esta red utilizaba un solo cable coaxial grueso y extenso; operaba a 3 Mbps.

Llamaron al sistema **Ethernet** en honor al éter luminífero, por medio del cual se pensaba antes que se propagaba la radiación electromagnética (cuando el físico inglés del siglo xix James Clerk Maxwell descubrió que la radiación electromagnética se podía describir mediante una ecuación de onda, los científicos asumieron que el espacio debía estar lleno de algún medio etéreo en el que se propagaba la radiación. No fue sino hasta después del famoso experimento de Michelson-Morley en 1887 que los físicos descubrieron que la radiación electromagnética se podía propagar en un vacío).

La Xerox Ethernet fue tan exitosa que DEC, Intel y Xerox idearon un estándar en 1978 para una Ethernet de 10 Mbps, conocido como **estándar DIX**. Con una modificación menor, el estándar DIX se convirtió en el estándar IEEE 802.3 en 1983. Cuando Xerox mostró poco interés en hacer algo con Ethernet aparte de ayudar a estandarizarla, Metcalfe formó su propia empresa llamada 3Com para vender adaptadores de Ethernet para PC. Vendió muchos millones de ellos.

La Ethernet clásica se tendía alrededor del edificio como un solo cable largo al que se conectaban todas las computadoras. Esta arquitectura se muestra en la Ilustración 60. La primera variedad, conocida popularmente como **Ethernet gruesa**, se asemejaba a una manguera de jardín amarilla, con marcas cada 2.5 metros para mostrar en dónde conectar las computadoras (el estándar 802.3 en realidad no requería que el cable fuera amarillo, pero sí lo sugería). Después le siguió la **Ethernet delgada**, que se doblaba con más facilidad y las conexiones se realizaban mediante conectores BNC. La Ethernet delgada era mucho más económica y fácil de instalar, pero sólo se podían tender 185 metros por segmento (en vez de los 500 m con la Ethernet gruesa), cada uno de los cuales sólo podía manejar 30 máquinas (en vez de 100).

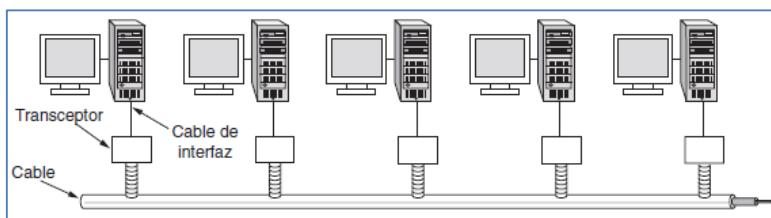


Ilustración 60 - Arquitectura de Ethernet clásica.

Cada versión de Ethernet tiene una longitud de cable máxima por segmento (es decir, longitud sin amplificar) a través de la cual se propagará la señal. Para permitir redes más grandes, se pueden conectar varios cables mediante **repetidores**. Un repetidor es un dispositivo de capa física que recibe, amplifica (es decir, regenera) y retransmite las señales en ambas direcciones. En cuanto a lo que al software concierne, una serie de segmentos de cable conectados por repetidores no presenta ninguna diferencia en comparación con un solo cable (excepto por una pequeña cantidad de retardo que introducen los repetidores).

La información se enviaba a través de cada uno de estos cables mediante la codificación Manchester. Una Ethernet podía contener varios segmentos de cable y múltiples repetidores, pero no podía haber dos transceptores separados por más de 2.5 km, y no podía haber una trayectoria entre dos transceptores en la que se colocaran más de cuatro repetidores. La razón de esta restricción era para que el protocolo MAC pudiera funcionar de manera correcta.

El protocolo de subcapa MAC de la Ethernet clásica

El formato utilizado para enviar tramas se muestra en la Ilustración 61. Primero viene un Preámbulo de 8 bytes, cada uno de los cuales contiene el patrón de bits 10101010 (con la excepción del último byte, en el que los últimos 2 bits se establecen a 11). Este último byte se llama delimitador de Inicio de trama en el 802.3. La codificación de Manchester de este patrón produce una onda cuadrada de 10 MHz durante 6.4 µs para permitir que el reloj del receptor se sincronice con el del emisor. Los últimos dos bits indican al receptor que está a punto de empezar el resto de la trama.

Bytes	8	6	6	2	0-1500	0-46	4
(a)	Preámbulo	Dirección de destino	Dirección de origen	Tipo	Datos ↓↓	Relleno	Suma de verificación
(b)	Preámbulo	S o F	Dirección de destino	Dirección de origen	Longitud ↓↓	Datos ↓↓	Relleno

Ilustración 61 - Formatos de trama. (a) Ethernet (DIX). (b) IEEE 802.3.

Después vienen dos direcciones, una para el destino y una para el origen. Cada una de ellas tiene una longitud de 6 bytes. El primer bit transmitido de la dirección de destino es un 0 para direcciones ordinarias y un 1 para direcciones de grupo. Las direcciones de grupo permiten que 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 (multicasting)**. La dirección especial que consiste únicamente en bits 1 está reservada para **difusión (broadcasting)**. Una trama que contiene sólo bits 1 en el campo de destino se acepta en todas las estaciones de la red. La multidifusión es más selectiva, pero involucra el manejo de grupos para definir qué estaciones están en un grupo. Por el contrario, la difusión no hace ninguna diferencia entre las estaciones, por lo que no requiere manejo de grupos.

Una característica interesante de las direcciones de origen de las estaciones es que son globalmente únicas; el IEEE las asigna de manera central para asegurar que no haya dos estaciones en el mundo con la misma dirección. La idea es que cualquier estación pueda direccionar de manera exclusiva cualquier otra estación con sólo dar el número correcto de 48 bits. Para hacer esto, se utilizan los primeros 3 bytes del campo de dirección para un **OUI** (Identificador Único Organizacional, del inglés *Organizationally Unique Identifier*). El IEEE asigna los valores para este campo, e indican un fabricante. A los fabricantes se les asignan bloques de 2^{24} direcciones. El fabricante asigna los últimos 3 bytes de la dirección y programa la dirección completa en la NIC antes de venderla.

A continuación está el campo *Tipo* o *Longitud*, dependiendo de si la trama es Ethernet o IEEE 802.3. Ethernet usa un campo Tipo para indicar 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 sistema operativo tiene que saber a cuál entregarle la trama. El campo Tipo especifica a qué proceso darle la trama. Por ejemplo, un código de tipo de 0x0800 significa que los datos contienen un paquete IPv4.

El IEEE 802.3 decidió que este campo transportaría la longitud de la trama, ya que para determinar la longitud de Ethernet había que ver dentro de los datos; una violación del uso de capas. Desde luego que esto significaba que no había forma de que el receptor averiguara qué hacer con una trama entrante. Para resolver ese problema se agregó otro encabezado para el protocolo **LLC** (Control de Enlace Lógico, del inglés *Logical Link Control*) dentro de los datos. Utiliza 8 bytes para transportar los 2 bytes de información del tipo del protocolo.

Por desgracia, para cuando se publicó el estándar 802.3, había ya tanto hardware y software para DIX Ethernet en uso que pocos fabricantes y usuarios se esforzaron en reempaquetar los campos Tipo y Longitud. En 1997, el IEEE desistió y dijo que estaba bien usar ambas formas. Por fortuna, todos los campos Tipo que se usaban antes de 1997 tenían valores mayores que 1500, que estaba bien establecido como el máximo tamaño de datos. Ahora la regla es que cualquier número ahí que sea menor o igual a 0x600 (1536) se puede interpretar como Longitud, y cualquier número mayor de 0x600 se puede interpretar como Tipo. Ahora el IEEE puede sostener que todos usan su estándar y que todos los demás pueden seguir haciendo lo que ya estaban haciendo (ignorar el LLC).

Después están los datos, de hasta 1500 bytes. Este límite fue elegido de manera algo arbitraria cuando se estableció el estándar Ethernet, sobre todo 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 mayor límite superior 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 lo inservible, 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 completar 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 Ilustración 62. En el tiempo 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 tiempo $\tau - \epsilon$) 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, bloquea el cable para asegurarse de que el emisor no ignore la colisión. Cerca del tiempo 2τ , el emisor ve la ráfaga de ruido y aborta también su transmisión. Luego espera un tiempo aleatorio antes de reintentarlo.

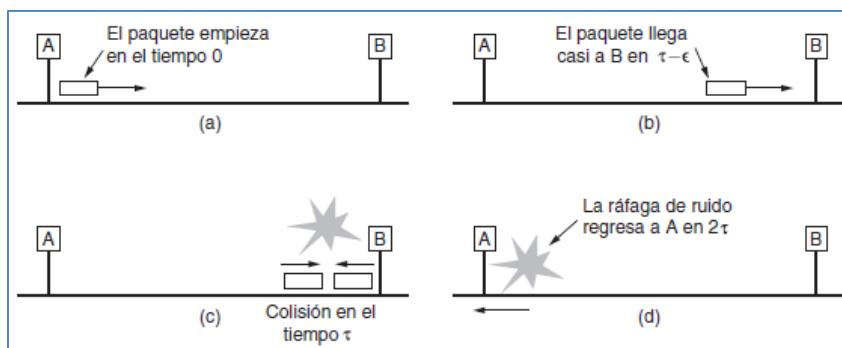


Ilustración 62 - La detección de una colisión puede tardar hasta 2τ .

Si una estación intenta transmitir una trama muy corta, es concebible que ocurra una colisión, pero la transmisión se completará antes de que la ráfaga de ruido llegue de regreso a la estación en 2τ . El emisor entonces supondrá de forma incorrecta que la trama se envió con éxito. Para evitar que ocurra esta situación, todas las tramas deberán tardar más de 2τ para enviarse, de manera que la transmisión aún se esté llevando 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 en cerca de 50 μ s en el peor de los casos. Por lo tanto, la trama más corta permitida se debe tardar por lo menos este tiempo en transmitir. A 10 Mbps, un bit tarda 100 ns, 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.

El campo final de es la Suma de verificación. Es un CRC de 32 bits que se define exactamente mediante el polinomio generador que vimos en la sección anterior, que funciona también para PPP, ADSL y otros enlaces. Esta CRC es un código de detección de errores que se utiliza para determinar si los bits de la trama se recibieron correctamente. Sólo realiza detección de errores y la trama se desecha si se detecta uno.

[CSMA/CD con retroceso exponencial binario](#)

La Ethernet clásica utiliza el algoritmo CSMA/CD persistente-1 que vimos en la sección 4.2. Este descriptor tan sólo significa que las estaciones detectan el medio cuando tienen una trama que desean enviar, y la envían

tan pronto como el medio está inactivo. Monitorean el canal por si hay colisiones al momento en que envían. Si hay una colisión, abortan la transmisión con una señal de bloqueo corta y vuelven a transmitir después de un intervalo aleatorio.

Ahora veamos cómo se determina el intervalo aleatorio cuando ocurre una colisión, ya que es un nuevo método. El modelo sigue siendo el de la Ilustración 52. Tras una colisión, el tiempo se divide en ranuras discretas cuya longitud es igual al tiempo de propagación de ida y vuelta para el peor de los casos 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 s.

Después de la primera colisión, cada estación espera 0 o 1 tiempos de ranura al azar antes de intentarlo de nuevo. Si dos estaciones entran en colisión y ambas escogen el mismo número aleatorio, habrá una nueva colisión. Después de la segunda colisión, cada una escoge 0, 1, 2 o 3 al azar y espera ese tiempo de ranura. Si ocurre una tercera colisión (la probabilidad de que esto suceda 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, después de i colisiones se elige un número aleatorio entre 0 y 2^i-1 , y se salta ese número de ranuras. Sin embargo, al llegar a 10 colisiones el intervalo de aleatorización se congela en un máximo de 1023 ranuras. Después de 16 colisiones, el controlador tira la toalla e informa a la computadora que fracasó. 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 sería insignificante, pero la espera promedio tras una colisión sería de cientos de tiempos de ranura, lo que introduce un retardo significativo. Por otra parte, si cada estación siempre se retardara 0 o 1 ranuras, 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. Al hacer que el intervalo de aleatorización crezca de manera exponencial a medida que ocurren cada vez 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 haya colisiones entre muchas estaciones. Al truncar el retroceso a 1023, evitamos que el límite crezca demasiado.

Si no hay colisión, el emisor supone que la trama probablemente se entregó con éxito. Es decir, ni CSMA/CD ni Ethernet proveen confirmaciones de recepción. Esta elección es apropiada para los canales de cable de cobre y de fibra óptica que tienen tasas de error bajas. Cualquier error que ocurra debe entonces detectarse mediante la CRC y recuperarse en las capas superiores. Para los canales inalámbricos que tienen más errores, veremos que se utilizan confirmaciones de recepción.

Desempeño de Ethernet

Ahora examinaremos brevemente el desempeño de la Ethernet clásica en condiciones de carga pesada y constante; es decir, con k estaciones siempre listas para transmitir. Es complicado un análisis riguroso del algoritmo de retroceso exponencial binario. El análisis matemático excede los propósitos de este apunte, con lo cual no limitaremos a exponer los resultados.

En la Ilustración 63 se presenta una gráfica de la eficiencia del canal contra el número de estaciones listas para $2\tau=51.2 \mu$ s y una tasa de transmisión de datos de 10 Mbps. 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 ranuras de 64 bytes por intervalo de contención, el periodo de contención tiene 174 bytes de longitud y la eficiencia es del 85%. Este resultado es mucho mejor que la eficiencia de 37% del ALOHA ranurado.

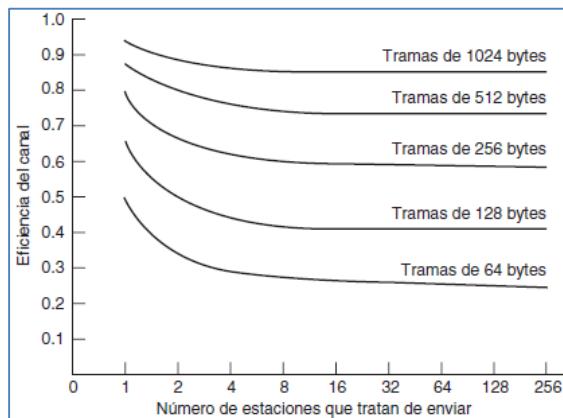


Ilustración 63 - Eficiencia de Ethernet a 10 Mbps con tiempos de ranuras de 512 bits.

Tal vez valga la pena mencionar que se ha realizado una gran cantidad de análisis teóricos del desempeño de Ethernet (y otras redes). Es conveniente tomar con cautela la mayoría de los resultados, por dos razones. Primero, casi todos estos trabajos han supuesto que el tráfico sigue una distribución de Poisson. A medida que los investigadores han comenzado a examinar datos reales, se ha hecho evidente que el tráfico en redes pocas veces se comporta así. Al contrario, es autosimilar o de ráfaga a través de un rango de escalas de tiempo. Lo que esto significa es que el promedio durante períodos extensos no hace al tráfico más uniforme. Además de usar modelos cuestionables, muchos de los análisis se enfocan en los casos “interesantes” de desempeño con una carga inusualmente alta.

Ethernet conmutada

Pronto Ethernet empezó a evolucionar y a alejarse de la arquitectura de un solo cable extenso de la Ethernet clásica. Los problemas asociados con el hecho de encontrar interrupciones o conexiones flojas condujeron hacia un distinto tipo de patrón de cableado, en donde cada estación cuenta con un cable dedicado que llega a un **hub** (concentrador) central. Un hub simplemente conecta de manera eléctrica todos los cables que llegan a él, como si estuvieran soldados en conjunto. Esta configuración se muestra en la Ilustración 64(a).

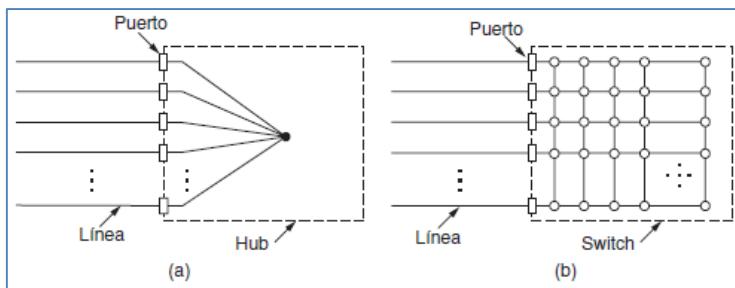


Ilustración 64 - (a) Hub. (b) Switch.

Los cables eran pares trenzados de la compañía telefónica, ya que la mayoría de los edificios de oficinas contaban con este tipo de cableado y por lo general había muchos de sobra. Esta reutilización fue una ventaja, pero a la vez se redujo la distancia máxima de cable del hub hasta 100 metros (200 metros si se utilizaban pares trenzados categoría 5 de alta calidad). En esta configuración es más simple agregar o quitar una estación, además de que los cables rotos se pueden detectar con facilidad. Con las ventajas de usar el cableado existente y la facilidad de mantenimiento, los hubs de par trenzado se convirtieron rápidamente en la forma dominante de Ethernet.

Sin embargo, los hubs no incrementan la capacidad debido a que son lógicamente equivalentes al cable extenso individual de la Ethernet clásica. A medida que se agregan más estaciones, cada estación recibe una parte cada vez menor de la capacidad fija. En un momento dado, la LAN se saturará. Una forma de solucionar esto es usar una velocidad más alta; por decir, de 10 Mbps a 100 Mbps, 1 Gbps o incluso mayores velocidades. Pero con el crecimiento de multimedia y los servidores, incluso una Ethernet de 1 Gbps se podría saturar.

Por fortuna existe otra forma de tratar con el aumento de carga: una Ethernet comutada. El corazón de este sistema es un **comutador** (*switch*) que contiene un **plano posterior** (*backplane*) de alta velocidad, el cual conecta a todos los puertos como se muestra en la Ilustración 64(b). Desde el exterior, un switch se ve igual que un hub. Ambos son cajas que por lo general contienen de 4 a 48 puertos, cada uno con un conector estándar RJ-45 r para un cable de par trenzado. Cada cable conecta al switch o hub con una sola computadora, como se muestra en la Ilustración 65. Un switch tiene también las mismas ventajas que un hub. Es fácil agregar o quitar una nueva estación con sólo conectar o desconectar un cable, y es fácil encontrar la mayoría de las fallas, ya que un cable o puerto defectuoso por lo general afectará a una sola estación. De todas formas hay un componente compartido que puede fallar (el mismo switch).

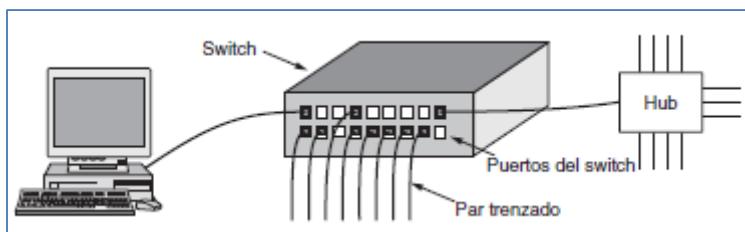


Ilustración 65 - Un switch Ethernet.

Sin embargo, dentro del switch ocurre algo muy distinto. Los switches sólo envían tramas a los puertos para los cuales están destinadas. Cuando el puerto de un switch recibe una trama Ethernet de una estación, el switch verifica las direcciones de Ethernet para ver cuál es el puerto de destino de la trama. Este paso requiere que el switch sea capaz de deducir qué puertos corresponden a qué direcciones. Por ahora supondremos que el switch conoce el puerto de destino de la trama. A continuación, el switch reenvía la trama a través de su plano posterior de alta velocidad hacia el puerto de destino. Por lo general, el plano posterior opera a muchos Gbps mediante el uso de un protocolo propietario que no necesita estandarización, ya que está completamente oculto dentro del switch. Después, el puerto de destino transmite la trama sobre el cable, de manera que pueda llegar a la estación de destino. Ninguno de los otros puertos sabe siquiera que existe la trama.

¿Qué ocurre si más de una estación o puerto desea enviar una trama al mismo tiempo? De nuevo, los switches difieren de los hubs. En un hub, todas las estaciones están en el mismo dominio de colisión. Deben usar el algoritmo CSMA/CD para programar sus transmisiones. En un switch, cada puerto es su propio **dominio de colisión** independiente. En el caso común en que el cable es full-dúplex, tanto la estación como el puerto pueden enviar una trama en el cable al mismo tiempo, sin preocuparse por los demás puertos y estaciones. Ahora las colisiones son imposibles y no se necesita CSMA/CD. Pero si el cable es half-dúplex, la estación y el puerto deben competir por la transmisión con CSMA/CD de la manera usual.

Un switch mejora el desempeño de la red en comparación con un hub de dos maneras. Primero, como no hay colisiones, la capacidad se utiliza con más eficiencia. Segundo y más importante, con un switch se pueden enviar varias tramas al mismo tiempo (por distintas estaciones). Estas tramas llegarán a los puertos del switch y viajarán hacia el plano posterior de éste para enviarlos por los puertos apropiados. No obstante, como se podrían enviar dos tramas al mismo puerto de salida y al mismo tiempo, el switch debe tener un búfer para que pueda poner temporalmente en cola una trama de entrada hasta que se pueda transmitir al puerto de salida. En general, estas mejoras producen una considerable ganancia en el desempeño que no es posible lograr con un hub. Con frecuencia, la velocidad real de transmisión total del sistema se puede incrementar en un orden de magnitud, dependiendo del número de puertos y patrones de tráfico.

El cambio en los puertos por donde se envían las tramas también incluye beneficios de seguridad. La mayoría de las interfaces de LAN tienen un **modo promiscuo**, en el que todas las tramas se entregan a cada computadora y no sólo las que van dirigidas a ella. En un hub, cualquier computadora conectada puede ver el tráfico transmitido entre todas las demás computadoras. Los espías y los intrusos aman esta característica. En un switch, el tráfico se reenvía sólo a los puertos a los que está destinado. Esta restricción provee un mejor aislamiento, de modo que el tráfico no escape fácilmente y caiga en las manos equivocadas. Sin embargo, es mejor cifrar el tráfico si de verdad se necesita seguridad.

Como el switch sólo espera tramas Ethernet estándar en cada puerto de entrada, es posible usar algunos de los puertos como concentradores. En la Ilustración 65, el puerto en la esquina superior derecha no está conectado a una sola estación, sino a un hub de 12 puertos. A medida que llegan tramas al hub, compiten por el cable de la manera usual, incluyendo las colisiones y el retroceso binario. Las tramas que tienen éxito pasan por el hub hasta el switch, en donde se tratan como cualquier otra trama entrante. El switch no sabe que tuvieron que competir para entrar. Una vez en el switch, se envían a la línea de salida correcta a través del plano posterior de alta velocidad. También es posible que el destino correcto estuviera en una de las líneas conectadas al hub, en cuyo caso la trama ya se entregó y el switch simplemente la descarta. Los hubs son más simples y económicos que los switchs, pero debido a que estos últimos han reducido su precio constantemente, los primeros han caído en desuso. Las redes modernas usan en su mayor parte Ethernet conmutada. Sin embargo, aún existen los hubs heredados.

FastEthernet

Al mismo tiempo que los switchs ganaban popularidad, la velocidad de 10 Mbps de Ethernet estaba bajo una presión cada vez mayor. Al principio, 10 Mbps parecían el cielo, al igual que los módems de cable parecieron el cielo a los usuarios de los módems telefónicos. Pero la novedad desapareció muy rápido. Como un tipo de corolario a la Ley de Parkinson (“El trabajo se expande hasta llenar el tiempo disponible para que se termine”), tal parecía que los datos se expandieron hasta llenar el ancho de banda disponible para su transmisión.

Muchas instalaciones necesitaban más ancho de banda y, por lo tanto, tenían numerosas redes LAN de 10 Mbps conectadas por una maraña de repetidores, hubs y switchs. Pero incluso con los switchs de Ethernet, el ancho de banda máximo de una sola computadora estaba limitado por el cable que lo conectaba con el puerto del switch.

Fue en este entorno que el IEEE convocó al comité 802.3 en 1992 con instrucciones de idear una LAN más rápida. Una propuesta fue mantener la red 802.3 igual a como estaba, sólo que 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. Esta estrategia cumpliría con el objetivo antes de que cambiara la tecnología, además de evitar los problemas imprevistos con un diseño totalmente nuevo. El nuevo diseño también sería compatible con las versiones previas de redes LAN Ethernet existentes. 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, formaron su propio comité y estandarizaron su LAN de todas maneras (que con el tiempo se llamó 802.12). Pero fracasó rotundamente.

El trabajo se terminó muy rápido (mediante las normas de los comités de estándares) y el resultado, 802.3u, fue aprobado de manera oficial por el IEEE en junio de 1995. Técnicamente, 802.3u no es un nuevo estándar sino un agregado al estándar, 802.3 existente (para enfatizar su compatibilidad con versiones anteriores). Esta estrategia es muy utilizada. Puesto que prácticamente todos lo llaman **FastEthernet** en vez de 802.3u, nosotros también lo haremos.

La idea básica detrás de FastEthernet era simple: mantener todos los formatos, interfaces y reglas de procedimientos anteriores, pero reducir el tiempo de bits de 100 ns a 10 ns. Técnicamente, habría sido posible copiar la Ethernet clásica de 10 Mbps y aún detectar colisiones a tiempo con sólo reducir la longitud máxima de cable por un factor de 10. Sin embargo, las ventajas del cableado de par trenzado eran tan abrumadoras que FastEthernet se basa por completo en este diseño. Por lo tanto, todos los sistemas FastEthernet utilizan hubs y switchs; no se permiten cables con múltiples derivaciones vampiro ni conectores BNC.

Sin embargo, aún había que tomar algunas decisiones, siendo la más importante de todas qué tipos de cable soportar. Una opción era el cable de par trenzado categoría 3. El argumento a su favor era que casi todas las oficinas en el mundo occidental tenían cables de par trenzado categoría 3 (o mejor) que iban desde ahí hasta un gabinete de cableado telefónico dentro de una distancia de 100 metros. Por lo tanto, al usar cable de par trenzado categoría 3 se podrían cablear las computadoras de escritorio mediante FastEthernet 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 transportar 100 Mbps a más de 100 metros, la máxima distancia de computadora a hub especificada para hubs de 10 Mbps. En contraste, el cable de par trenzado categoría 5 puede manejar 100 metros con facilidad, y la fibra puede recorrer mucha más distancia. El compromiso elegido fue permitir las tres posibilidades, como se muestra en la Ilustración 66, 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	Full-dúplex a 100 Mbps (UTP cat 5).
100Base-FX	Fibra óptica	2000 m	Full-dúplex a 100 Mbps; distancias largas.

Ilustración 66 - Cableado original de FastEthernet.

El esquema UTP categoría 3, llamado **100Base-T4**, utilizaba una velocidad de señalización de 25 MHz, tan sólo un 25% más rápida que los 20 MHz de la Ethernet estándar (recuerde que la codificación Manchester requiere dos periodos de reloj para cada uno de los 10 millones de bits que se envían cada segundo). Sin embargo, para alcanzar la tasa de bits necesaria, 100Base-T4 requiere cuatro cables de par trenzado. De los cuatro pares, uno siempre va al hub, uno siempre sale del hub y los otros dos se pueden conmutar a la dirección actual de la transmisión. Para obtener 100 Mbps de los tres pares trenzados en la dirección de la transmisión, se utiliza un esquema bastante complejo en cada par trenzado, que implica enviar dígitos ternarios con tres distintos niveles de voltaje. Omitiremos los detalles, dado que no es de importancia en este apunte. Sin embargo, y como el cableado telefónico estándar ha tenido durante décadas cuatro pares trenzados por cable, la mayoría de las oficinas pueden usar la planta de cableado existente. Claro que esto significa renunciar al teléfono de su oficina, pero sin duda es un pequeño precio a pagar para obtener correo electrónico, que es más rápido.

100Base-T4 quedó al borde del camino debido a que se actualizó el cableado de muchos edificios de oficinas por UTP categoría 5 para Ethernet **100Base-TX**, el cual llegó a dominar el mercado. Este diseño es más simple puesto que los cables pueden manejar velocidades de reloj de 125 MHz. Sólo se utilizan dos pares trenzados por estación, uno que va al hub y otro que viene de él. No se utiliza la codificación binaria directa (es decir, NRZ) ni la codificación Manchester. En cambio se utiliza la codificación 4B/5B. Se codifican 4 bits de datos como 5 bits de señal y se envían a 125 MHz para proveer 100 Mbps. Este esquema es simple pero tiene suficientes transiciones para la sincronización, además de que utiliza muy bien el ancho de banda del cable. El sistema 100Base-TX es full-dúplex; las estaciones pueden transmitir a 100 Mbps en un par trenzado y recibir a 100 Mbps en otro par trenzado al mismo tiempo.

La última opción, **100Base-FX**, utiliza dos filamentos de fibra multimodo, una para cada dirección, por lo que también es full-dúplex con 100 Mbps en cada dirección. En esta configuración, la distancia entre una estación y el switch puede ser de hasta 2 km.

FastEthernet permite la interconexión mediante hubs o switches. Para asegurar que el algoritmo CSMA/CD siga trabajando, es necesario mantener la relación entre el tamaño mínimo de trama y la longitud máxima del cable a medida que la velocidad de la red aumenta de 10 Mbps a 100 Mbps. Así, el tamaño mínimo de trama de 64 bytes debe aumentar o la longitud máxima de cable de 2500 debe disminuir, en forma proporcional. La elección fácil fue reducir la distancia máxima entre dos estaciones cualesquiera por un factor de 10, puesto que un hub con cables de 100 m ya se encuentra dentro de este nuevo valor máximo. Sin embargo, los cables 100Base-FX de 2 km son demasiado largos como para permitir un hub de 100 Mbps con el algoritmo de colisiones normal de Ethernet. Estos cables se deben conectar a un switch y operar en un modo full-dúplex para que no haya colisiones.

Los usuarios empezaron a implementar con rapidez el estándar FastEthernet, pero no deseaban tirar las tarjetas Ethernet de 10 Mbps en las computadoras antiguas. Como consecuencia, casi todos los switches FastEthernet pueden manejar una mezcla de estaciones de 10 Mbps y 100 Mbps. Para facilitar la actualización, el estándar provee por sí solo un mecanismo llamado autonegociación, el cual permite que dos estaciones negocien de manera automática la velocidad óptima (10 o 100 Mbps) y la duplicidad (half-dúplex o full-dúplex). Funciona bien la mayor parte del tiempo, pero se sabe que provoca problemas de desajuste de duplicidad

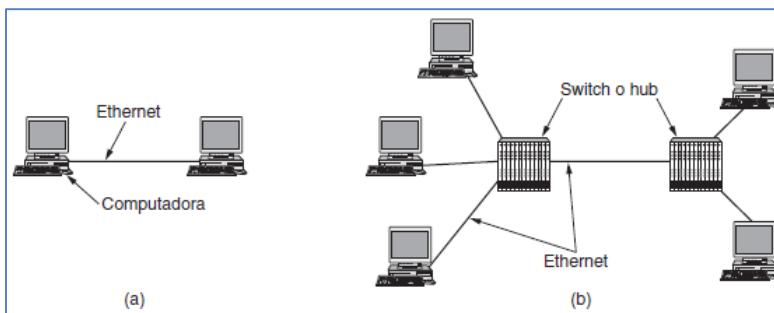
cuando un extremo del enlace realiza la autonegociación pero el otro extremo no, y se establece en modo full-dúplex. La mayoría de los productos Ethernet usan esta característica para configurarse a sí mismos.

GigabitEthernet

La tinta apenas se estaba secando en el estándar de la FastEthernet cuando el comité 802 comenzó a trabajar en una Ethernet aún más rápida, que de inmediato recibió el apodo de **GigabitEthernet**. El IEEE ratificó la forma más popular como 80.3ab en 1999. A continuación analizaremos algunas de las características clave de la GigabitEthernet.

Los objetivos del comité para la GigabitEthernet eran en esencia los mismos que los del comité para FastEthernet: que tuviera un desempeño 10 veces mayor y que mantuviera la compatibilidad con todos los estándares Ethernet existentes. En particular, GigabitEthernet tenía que ofrecer servicio de datagramas sin confirmación de recepción con unidifusión y multidifusión, utilizar el mismo esquema de direccionamiento de 48 bits que ya estaba en uso y mantener el mismo formato de trama, incluyendo los tamaños mínimo y máximo de trama. El estándar final cumple con todos estos objetivos.

Al igual que FastEthernet, todas las configuraciones de GigabitEthernet usan enlaces punto a punto. En la configuración más simple, que se muestra en la Ilustración 67(a), dos computadoras están conectadas directamente una con otra. Sin embargo, el caso más común es tener un switch o un hub conectado a varias computadoras y quizás a switches o hubs adicionales, como se muestra en la Ilustración 67(b). En ambas configuraciones, cada cable Ethernet individual tiene exactamente dos dispositivos en él, ni más ni menos.



Además, al igual que FastEthernet, GigabitEthernet soporta dos modos diferentes de funcionamiento: modo full-dúplex y modo half-dúplex. El modo “normal” es el modo full-dúplex, que permite tráfico en ambas direcciones al mismo tiempo. Este modo se utiliza cuando hay un switch central conectado a computadoras (o a otros switches) en la periferia. En esta configuración, todas las líneas se almacenan en el búfer con el fin de que cada computadora y switch 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 switch, la computadora es la única que puede enviar al switch y la transmisión tendrá éxito aun cuando el switch esté enviando ahora una trama a la computadora (porque la línea es full-dúplex). Debido a que no hay contención, no se utiliza el protocolo CSMA/CD y la longitud máxima del cable se determina con base en los aspectos relacionados con la intensidad de la señal, en vez de basarse en el tiempo que tarda una ráfaga de ruido en propagarse de vuelta al emisor en el peor de los casos. Los switches tienen la libertad de mezclar e igualar velocidades. La autonegociación se soporta al igual que en FastEthernet, sólo que ahora la opción está entre 10, 100 y 1000 Mbps.

El otro modo de operación es half-dúplex y se utiliza cuando las computadoras están conectadas a un hub en vez de un switch. Un hub no almacena 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 puede haber colisiones, por lo que se requiere el protocolo CSMA/CD estándar. Debido a que ahora se puede transmitir una trama de 64 bytes (la más corta permitida) 100 veces más rápido que en la Ethernet clásica, la longitud máxima del cable debe ser 100 veces menor, o de 25 metros, para mantener la propiedad esencial de que el emisor aún transmita cuando la ráfaga de ruido regrese a él, incluso en el peor de los casos. Con un cable de 2500 metros de longitud, el emisor de una trama de 64 bytes a 1 Gbps podría terminar su

transmisión mucho antes de que la trama recorriera siquiera una décima del camino, y por ende muchísimo antes de que llegara al otro extremo y regresara.

Esta restricción de longitud fue tan dolorosa que se agregaron dos características al estándar para incrementar la longitud máxima del cable a 200 metros, lo cual quizás es suficiente para la mayoría de las oficinas. La primera característica, llamada **extensión de portadora**, básicamente indica al hardware que agregue su propio relleno después de la trama normal para extenderla a 512 bytes. Como el hardware emisor agrega este relleno y el hardware receptor lo elimina, el software no toma parte en esto, lo que significa que no es necesario realizar cambios al software existente. La desventaja es que usar 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 sólo un 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 de nuevo. Si hay suficientes tramas esperando su transmisión, este esquema es muy eficiente y se prefiere en vez de la extensión de portadora.

Como se lista en la Ilustración 68, GigabitEthernet soporta tanto el cableado de cobre como el de fibra óptica. La señalización en, o cerca de, 1 Gbps requiere codificar y enviar un bit cada nanosegundo. En un principio este truco se lograba con cables cortos de cobre blindados (la versión 1000Base-CX) y con fibra óptica. En la fibra óptica se permiten dos longitudes de onda y resultan dos versiones distintas: 0.85 micras (corto, para 1000Base-SX) y 1.3 micras (largo, para 1000Base-LX).

Nombre	Cable	Segmento máximo	Ventajas
1000Base-SX	Fibra óptica	550 m	Fibra multimodo (50, 62.5 micras)
1000Base-LX	Fibra óptica	5000 m	Monomodo (10 μ) o multimodo (50, 62.5μ)
1000Base-CX	2 pares de STP	25 m	Par trenzado blindado
1000Base-T	4 pares de UTP	100 m	UTP estándar categoría 5

Ilustración 68 - Cableado de GigabitEthernet.

La señalización en la longitud de onda corta se puede realizar mediante LEDs económicos. Se utiliza con fibra multimodo y es útil para las conexiones dentro de un edificio, ya que puede funcionar hasta por 500 m para la fibra de 50 micras. La señalización en la longitud de onda larga requiere láser más costosos. Por otro lado, al combinarse con fibra monomodo (10 micras), la longitud de cable puede ser de hasta 5 km. Este límite permite conexiones de larga distancia entre edificios (por ejemplo, la red troncal de un campus) como un enlace punto a punto dedicado. Las versiones posteriores del estándar permitían enlaces aún más largos a través de fibra monomodo.

Para enviar bits por estas versiones de GigabitEthernet, se pidió utilizar la codificación 8B/10B, conocida como Canal de fibra. Este esquema codifica 8 bits de datos en palabras codificadas de 10 bits que se envían a través del cable o la fibra, de aquí que se llame 8B/10B. Las palabras codificadas se eligieron de modo que se pudieran balancear (es decir, que tengan el mismo número de 0s y 1s) con suficientes transiciones para la recuperación del reloj. Para enviar los bits codificados con NRZ se requiere un ancho de banda de señalización de un 25% más que el requerido para los bits sin codificación, una importante mejora en comparación con la expansión de 100% de la codificación Manchester.

Sin embargo, todas estas opciones requerían nuevos cables de cobre o de fibra para soportar la señalización más rápida. Ninguna de ellas hizo uso de la gran cantidad de cable UTP categoría 5 que se había instalado con la red FastEthernet. Antes de un año llegó 1000Base-T para llenar este vacío, y desde entonces ha sido la forma más popular de GigabitEthernet.

Para hacer que Ethernet opere a 1000 Mbps a través de cables categoría 5 se necesita una señalización más complicada. Para empezar se utilizan los cuatro pares trenzados en el cable, y cada par se utiliza en ambas direcciones al mismo tiempo mediante el uso de un procesamiento de señales digitales para separar las señales. En cada cable se utilizan cinco niveles de voltaje que transportan 2 bits para una señalización de 125

Msímbolos/s. La asignación para producir los símbolos a partir de los bits no es simple. Se requiere un mezclado (*scrambling*) para las transiciones, seguido de un código de corrección de errores en el que se incrustan cuatro valores en cinco niveles de señal.

Hay una extensión más que se introdujo junto con GigabitEthernet. Las **tramas Jumbo** que permiten que las tramas tengan una longitud mayor de 1500 bytes, por lo general de hasta 9 KB. Esta extensión es propietaria. No la reconoce el estándar debido a que si se utiliza, entonces Ethernet ya no es compatible con las versiones anteriores, pero la mayoría de los distribuidores la soporta de todas formas. La razón es que 1500 bytes es una unidad corta a velocidades de gigabits. Al manipular bloques más grandes de información, la tasa de transmisión de tramas se puede reducir, junto con el procesamiento asociado con ésta, tal como interrumpir al procesador para decir que llegó una trama, o dividir y recombinar mensajes que eran demasiado largos como para caber en una trama Ethernet.

10 GigabitEthernet

Tan pronto como el GigabitEthernet se estandarizó, el comité 802 volvió al trabajo. El IEEE les dijo que iniciaran una Ethernet de 10 gigabits. Este trabajo siguió casi el mismo patrón que los estándares Ethernet anteriores, en donde aparecieron estándares para fibra y cable de cobre blindado por primera vez en 2002 y 2004, seguidos de un estándar para par trenzado de cobre en 2006.

10 Gbps es una velocidad realmente prodigiosa, 1000 veces más rápida que la Ethernet original. ¿En dónde se podría necesitar? La respuesta es que dentro de los centros e intercambios de datos para conectar enruteadores, switchs y servidores de gama alta, así como en las troncales de larga distancia con alto ancho de banda entre las oficinas que permiten la operación de redes de área metropolitana completas, basadas en Ethernet y fibra. Las conexiones de larga distancia usan fibra óptica, mientras que las conexiones cortas pueden usar cobre o fibra.

Todas las versiones de Ethernet de 10 gigabits soportan sólo la operación full-dúplex. CSMA/CD ya no forma parte del diseño y los estándares se concentran en los detalles de las capas físicas que pueden operar a muy alta velocidad. Pero la compatibilidad aún sigue siendo importante, por lo que las interfaces Ethernet de 10 gigabits usan la autonegociación y cambian a la velocidad más alta soportada por ambos extremos de la línea.

En la Ilustración 69 se listan los principales tipos de Ethernet de 10 gigabits. Se utiliza fibra multimodo con la longitud de onda de 0.85 m (corta) para distancias medias, y la fibra monomodo a 1.3 m (larga) y 1.5 m (extendida) para distancias largas. 10GBase-ER puede operar en distancias de 40 km, lo cual la hace adecuada para aplicaciones de área amplia. Todas estas versiones envían un flujo serial de información que se produce mediante el mezclado de los bits de datos, para después codificarlos mediante un código 64B/66B. Esta codificación tiene menos sobrecarga que un código 8B/10B.

La primera versión de cobre que se definió (10GBase-CX4) utiliza un cable con cuatro pares de cableado twinaxial de cobre (similar al cable coaxial, solo que cuenta con dos núcleos de cobre). Cada par usa codificación 8B/10B y opera a 3.125 Gsímbolos/s para alcanzar 10 Gbps. Esta versión es más económica que la fibra y fue de las primeras en comercializarse.

Nombre	Cable	Segmento máximo	Ventajas
10GBase-SR	Fibra óptica	Hasta 300 m	Fibra multimodo (0.85 μ).
10GBase-LR	Fibra óptica	10 km	Fibra monomodo (1.3 μ).
10GBase-ER	Fibra óptica	40 km	Fibra monomodo (1.5 μ).
10GBase-CX4	4 pares de twinax	15 m	Cobre twinaxial.
10GBase-T	4 pares de UTP	100 m	UTP categoría 6a

Ilustración 69 - Cableado de Ethernet de 10 gigabits.

10GBase-T es la versión que usa cables UTP. Aunque requiere cableado categoría 6a, en distancias más cortas puede usar categorías más bajas (incluyendo la categoría 5) para reutilizar una parte del cableado ya instalado. No es sorpresa que la capa física esté bastante involucrada para llegar a 10 Gbps sobre par trenzado. Sólo

esbozaremos algunos de los detalles de alto nivel. Cada uno de los cuatro pares trenzados se utiliza para enviar 2500 Mbps en ambas direcciones. Para llegar a esta velocidad se utiliza una tasa de señalización de 800 Msímbolos/s, con símbolos que usan 16 niveles de voltaje. Para producir los símbolos se mezclan los datos, se protegen con un código **LDPC** (Verificación de Paridad de Baja Densidad, del inglés *Low Density Parity Check*) y se vuelven a codificar para corrección de errores.

A finales de 2007, el IEEE creó un grupo para estandarizar la Ethernet que opera a 40 Gbps y 100 Gbps. Esta actualización permitirá a Ethernet competir en ambientes de muy alto rendimiento, incluyendo las conexiones de larga distancia en redes troncales y las conexiones cortas a través de los planos posteriores de los equipos.

Redes LAN inalámbricas

Las redes LAN inalámbricas son cada vez más populares; los hogares, oficinas, cafeterías, bibliotecas, aeropuertos y demás sitios públicos se están equipando con este tipo de redes para conectar computadoras, dispositivos electrónicos y teléfonos inteligentes (*smartphones*) a Internet. Las redes LAN inalámbricas también se pueden usar para permitir que dos o más computadoras que estén cerca unas de otras se comuniquen sin necesidad de usar Internet.

El principal estándar de LAN inalámbrica es 802.11. En las siguientes secciones analizaremos la pila de protocolos, las técnicas de radiotransmisión de la capa física, el protocolo de subcapa MAC, la estructura de las tramas y los servicios ofrecidos.

La arquitectura de 802.11 y la pila de protocolos

Las redes 802.11 se pueden utilizar en dos modos. El modo más popular es conectar clientes, como laptops y teléfonos inteligentes, a otra red, como la intranet de una empresa o Internet. Este modo se muestra en la Ilustración 70(a). En el **modo de infraestructura**, cada cliente se asocia con un AP (Punto de Acceso, del inglés *Access Point*) que a su vez está conectado a la otra red. El cliente envía y recibe sus paquetes a través del AP. Se pueden conectar varios puntos de acceso juntos, por lo general mediante una red alámbrica llamada **sistema de distribución**, para formar una red 802.11 extendida. En este caso, los clientes pueden enviar tramas a otros clientes a través de sus APs.

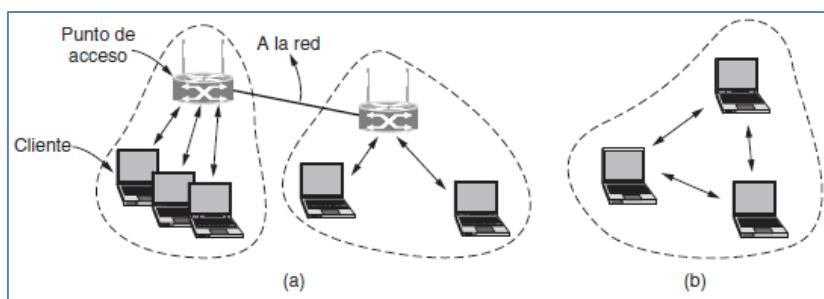


Ilustración 70 - Arquitectura 802.11. (a) Modo de infraestructura. (b) Modo ad hoc.

El otro modo, que se muestra en la Ilustración 70(b), es una **red ad hoc**. Este modo es una colección de computadoras que están asociadas de manera que puedan enviarse tramas directamente unas a otras. No hay punto de acceso. Como el acceso a Internet es la aplicación esencial para las redes inalámbricas, las redes ad hoc no son muy populares.

Todos los protocolos 802, incluyendo 802.11 y Ethernet, tienen ciertas similitudes en su estructura. En la Ilustración 71 se muestra una vista parcial de la pila de protocolos del estándar 802.11. La pila es la misma para los clientes y APs. La capa física corresponde muy bien con la capa física OSI, pero la capa de enlace de datos en todos los protocolos 802 se divide en dos o más subcapas. En el estándar 802.11, la subcapa MAC (Control de Acceso al Medio) 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 (Control de Enlace Lógico), cuya función es ocultar las diferencias entre las variantes 802 con el fin de que sean imperceptibles en lo que respecta a la capa de red. Ésta podría haber sido una responsabilidad considerable, pero en estos días la LLC es una capa de pegamento que identifica el protocolo (por ejemplo, IP) que se transporta dentro de una trama 802.11.

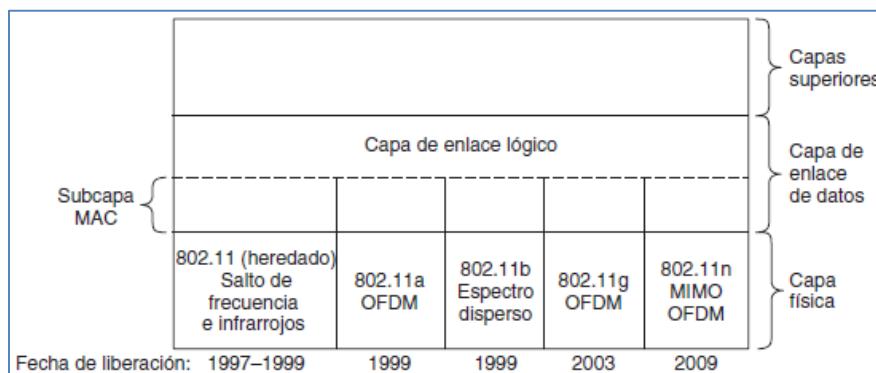


Ilustración 71 - Parte de la pila de protocolos 802.11.

Desde su aparición por primera vez en 1997, se han agregado varias técnicas de transmisión a la capa física a medida que el 802.11 ha ido evolucionando. Dos de las técnicas iniciales, infrarrojos en la forma de los controles remotos de televisión y el salto de frecuencia en la banda de 2.4 GHz, están ahora extintos. La tercera técnica inicial, el espectro disperso de secuencia directa a 1 o 2 Mbps en la banda de 2.4 GHz, se extendió para operar a tasas de hasta 11 Mbps y se convirtió rápidamente en un éxito. Ahora se conoce como 802.11b.

Se introdujeron en 1999 y 2003 nuevas técnicas de transmisión basadas en el esquema OFDM (Multiplexación por División de Frecuencia Ortogonal). La primera se llama 802.11a y utiliza una banda de frecuencia distinta, 5 GHz. La segunda se quedó con la banda de 2.4 GHz y la compatibilidad. Se llama 802.11g. Ambas ofrecen tasas de transmisión de hasta 54 Mbps.

En octubre de 2009 finalizó la creación de unas técnicas de transmisión que utilizan varias antenas en forma simultánea en el transmisor y el receptor para aumentar la velocidad, conocidas como 802.11n. Con cuatro antenas y canales más amplios, el estándar 802.11 definió tasas de transmisión asombrosas de hasta 600 Mbps. Ahora con 802.11ac y 802.11ax las tasas de transmisión han superado por mucho 1 Gbps.

Ahora examinaremos con brevedad cada una de estas técnicas de transmisión. Sin embargo, sólo cubriremos las que están en uso y omitiremos los métodos de transmisión 802.11 heredados.

La capa física del estándar 802.11

Cada una de las técnicas de transmisión hace que sea posible enviar una trama MAC por el aire, de una estación a otra. Sin embargo, difieren en la tecnología utilizada y en las velocidades alcanzables. Un estudio detallado de estas tecnologías está más allá del alcance de este curso, pero unas cuantas palabras sobre dichas tecnologías relacionarán las técnicas con la teoría vista y proveerán a los interesados algunos términos clave con los cuales podrán buscar más información en alguna otra parte.

Todas las técnicas 802.11 usan radios de corto alcance para transmitir señales en las bandas de frecuencias ISM de 2.4 GHz o de 5 GHz. Estas bandas poseen la ventaja de que no necesitan licencia y por ende, están libremente disponibles para cualquier transmisor que desee cumplir con ciertas restricciones, como una potencia radiada de al menos 1 W (aunque es más típico el valor de 50 mW para los radios LAN inalámbricos). Por desgracia, este hecho también lo conocen los fabricantes de abridores de puertas de cochera, teléfonos inalámbricos, hornos de microondas y numerosos dispositivos más, los cuales compiten con las computadoras portátiles por el mismo espectro. La banda de 2.4 GHz tiende a estar más saturada que la banda de 5 GHz, por lo que tal vez esta última sea mejor para ciertas aplicaciones, aun cuando tiene un alcance más corto debido a que la frecuencia es más alta.

Además, todos los métodos de transmisión definen múltiples tasas. La idea es que se puedan utilizar distintas tasas dependiendo de las condiciones actuales. Si la señal inalámbrica es débil, se puede usar una tasa baja. Si la señal es clara, se puede usar la tasa más alta. A este ajuste se le conoce como **adaptación de tasa**. Ya que las tasas varían por un factor de 10 o más, es importante una buena adaptación de tasa para un buen desempeño. Desde luego que, como no se necesita para la interoperabilidad, los estándares no dicen cómo se debe realizar la adaptación de tasa.

El primer método de transmisión que analizaremos es **802.11b**. Es un método de espectro disperso que soporta tasas de 1, 2, 5.5 y 11 Mbps, aunque en la práctica la tasa de operación es casi siempre de 11 Mbps. Es similar al sistema CDMA, excepto que sólo hay un código de dispersión que comparten todos los usuarios. La secuencia de dispersión que utiliza el 802.11b es una **secuencia de Barker**. Tiene la propiedad de que su autocorrelación es baja, excepto cuando las secuencias están alineadas. Esta propiedad permite a un receptor bloquear el inicio de una transmisión. Para enviar a una tasa de 1 Mbps, se utiliza la secuencia de Barker con modulación BPSK para enviar 1 bit por 11 chips. Los chips se transmiten a una tasa de 11 Mchips/s. Para enviar a 2 Mbps, se utiliza con modulación QPSK para enviar 2 bits por 11 chips. Las tasas más altas son distintas. Estas tasas usan una técnica llamada **CCK** (Modulación por Código Complementario, del inglés *Complementary Code Keying*) para construir códigos en vez de la secuencia de Barker. La tasa de 5.5 Mbps envía 4 bits en cada código de 8 chips, y la tasa de 11 Mbps envía 8 bits en cada código de 8 chips.

Ahora hablaremos sobre el 802.11a, que soporta tasas de hasta 54 Mbps en la banda ISM de 5 GHz. Tal vez piense que el 802.11a debe ir antes que el 802.11b, pero en este caso no fue así. Aunque el grupo 802.11a se estableció primero, el estándar 802.11b se aprobó antes y su producto llegó al mercado mucho antes de los productos 802.11a, en parte debido a la dificultad de operar en la banda más alta de 5 GHz.

El método 802.11a se basa en OFDM, ya que OFDM usa el espectro con eficiencia y resiste las degradaciones de las señales inalámbricas tales como multirayectoria. Los bits se envían a través de 52 subportadoras en paralelo, 48 de las cuales llevan datos y 4 se usan para sincronización. Cada símbolo dura 4 µs y envía 1, 2, 4 o 6 bits. Los bits están codificados para corrección de errores mediante un código convolucional primero, por lo que sólo 1/2, 2/3 o 3/4 partes de los bits no son redundantes. Con distintas combinaciones, el 802.11a puede operar a ocho tasas distintas, que varían desde 6 hasta 54 Mbps. Estas tasas son considerablemente más rápidas que las tasas del 802.11b, además de que hay menos interferencia en la banda de 5 GHz. Sin embargo, el 802.11b tiene un alcance aproximado de siete veces mayor que el del 802.11a, lo cual es más importante en muchas situaciones.

802.11g copia los métodos de modulación OFDM del 802.11a, pero opera en la banda ISM estrecha de 2.4 GHz junto con el 802.11b. Ofrece las mismas tasas de transmisión que el 802.11a (de 6 a 54 Mbps), además de compatibilidad con cualquier dispositivo 802.11b que se encuentre cerca. Todas estas distintas opciones pueden ser confusas para los clientes, por lo que es común que los productos soporten 802.11a/b/g en una sola NIC.

En 2009 se establece el estándar **802.11n**. Su objetivo era una tasa real de transferencia de por lo menos 100 Mbps después de eliminar todas las sobrecargas inalámbricas. Este objetivo exigía un aumento de por lo menos cuatro veces la velocidad en crudo. Para hacerlo realidad, el comité duplicó los canales de 20 MHz a 40 MHz y redujo las sobrecargas de entramado al permitir enviar todo un grupo de tramas a la vez. Pero es más notable el hecho de que el 802.11n usa hasta cuatro antenas para transmitir hasta cuatro flujos de información a la vez. Las señales de los flujos interfieren en el receptor, pero se pueden separar mediante técnicas de comunicaciones **MIMO** (Múltiples Entrada Múltiples Salida, del inglés *Multiple Input Multiple Output*). El uso de varias antenas ofrece un enorme aumento en la velocidad, o en su defecto un mejor alcance y confiabilidad.

El estándar **802.11ac** (también conocido como WiFi 5 o WiFi Gigabit) es una mejora al estándar 802.11n, se desarrolló entre el año 2011 y el 2013, y finalmente se aprobó en julio de 2014. Consiste en mejorar las tasas de transferencia hasta 433 Mbit/s por flujo de datos, consiguiendo teóricamente tasas de 1.3 Gbit/s empleando 3 antenas. Opera dentro de la banda de 5 GHz, amplía el ancho de banda hasta 160 MHz (40 MHz en las redes 802.11n), utiliza hasta 8 flujos MIMO e incluye modulación de alta densidad.

Por último **802.11ax** (también conocido como WiFi 6) está diseñado para operar en las bandas de 2.4 GHz y 5 GHz. Además de utilizar MIMO y MU-MIMO, la nueva modificación introduce OFDMA para mejorar la eficiencia espectral global, y soporte de modulación 1024-QAM de orden superior para un mayor rendimiento. Aunque la velocidad nominal de datos es solo un 37% más alta que 802.11ac, se espera que la nueva enmienda logre un aumento de 4x en el rendimiento del usuario debido a una utilización del espectro más eficiente con un menor consumo de energía. Los dispositivos que se presentaron en CES 2018 mostraron una velocidad máxima de 11 Gbps. A partir del 3 de octubre de 2018, la *Wi-Fi Alliance* decide renombrar el estándar a Wi-Fi

6 o 6th Generation, esto con el fin de simplificar al usuario final el reconocimiento de la tecnología en los dispositivos que se empezaran a fabricar a principios de 2019.

El protocolo de la subcapa MAC del 802.11

El protocolo de la subcapa MAC para el estándar 802.11 es muy diferente al de Ethernet, debido a dos factores fundamentales para la comunicación inalámbrica.

Primero, las radios casi siempre son half-dúplex, lo cual significa que no pueden transmitir y escuchar ráfagas de ruido al mismo tiempo en una sola frecuencia. La señal recibida puede ser miles de veces más débil que la señal transmitida, por lo que no se puede escuchar al mismo tiempo. Con Ethernet, una estación sólo espera hasta que el medio queda en silencio y después comienza a transmitir. Si no recibe una ráfaga de ruido mientras transmite los primeros 64 bytes, es muy probable que se haya entregado la trama correctamente. En los sistemas inalámbricos, este mecanismo de detección de colisiones no funciona.

En vez de ello, el 802.11 trata de evitar colisiones con un protocolo llamado **CSMA/CA** (CSMA con Evitación de Colisiones, del inglés CSMA with Collision Avoidance). En concepto, este protocolo es similar al CSMA/CD de Ethernet, con detección del canal antes de enviar y retroceso exponencial después de las colisiones. Sin embargo, una estación que desee enviar una trama empieza con un retroceso aleatorio (excepto en el caso en que no haya utilizado el canal recientemente y éste se encuentre inactivo). No espera una colisión. El número de ranuras para el retroceso se elige en el rango de 0 hasta, por ejemplo, 15 en el caso de la capa física OFDM. La estación espera hasta que el canal está inactivo, para lo cual detecta que no hay señal durante un periodo corto (llamado DIFS, como veremos más adelante) y realiza un conteo descendente de las ranuras inactivas, haciendo pausa cuando se envían tramas. Envía su trama cuando el contador llega a 0. Si la trama logra pasar, el destino envía de inmediato una confirmación de recepción corta. La falta de una confirmación de recepción se interpreta como si hubiera ocurrido un error, sea una colisión o cualquier otra cosa. En este caso, el emisor duplica el periodo de retroceso e intenta de nuevo, continuando con el retroceso exponencial como en Ethernet, hasta que la trama se transmite con éxito o se llegue al número máximo de retransmisiones.

En la Ilustración 72 se muestra una línea de tiempo de ejemplo. La estación A es la primera en enviar una trama. Mientras A envía, las estaciones B y C se preparan para enviar. Ven que el canal está ocupado y esperan a que esté inactivo. Poco después de que A recibe una confirmación de recepción, el canal queda inactivo. Sin embargo, en vez de enviar una trama de inmediato y colisionar, B y C realizan un retroceso. C elige un retroceso corto, por lo que envía primero. B detiene su conteo mientras detecta que C está usando el canal y lo reanuda después de que C recibe una confirmación de recepción. Poco después, B completa su retroceso y envía su trama.

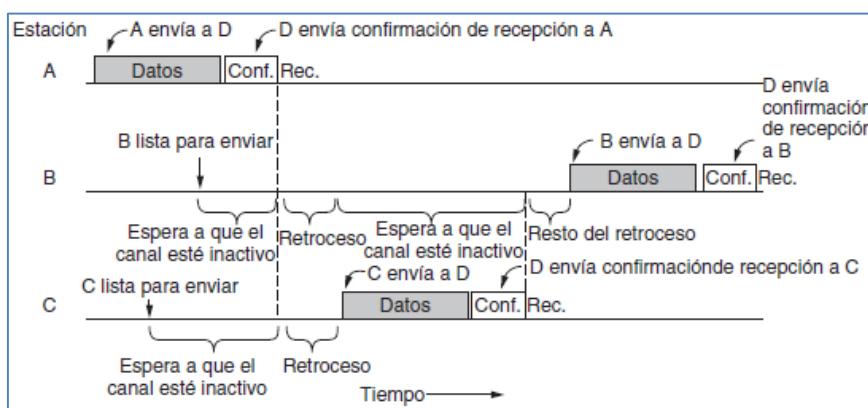


Ilustración 72 - Envío de una trama mediante CSMA/CA.

En comparación con Ethernet, hay dos diferencias principales. Primero, empezar los retrocesos lo más pronto posible ayuda a evitar las colisiones. Esto vale la pena debido a que las colisiones son costosas, puesto que se transmite toda la trama incluso aunque ocurra una colisión. Segundo, se utilizan confirmaciones de recepción para inferir colisiones, ya que estas últimas no se pueden detectar.

Este modo de operación se llama **DCF** (Función de Coordinación Distribuida, del inglés *Distributed Coordination Function*), ya que cada estación actúa en forma independiente, sin ningún tipo de control central. El estándar también incluye un modo opcional de operación llamado **PCF** (Función de Coordinación Puntual, del inglés *Point Coordination Function*), en donde el punto de acceso controla toda la actividad en su celda, justo igual que una estación base celular. Sin embargo, PCF no se utiliza en la práctica debido a que por lo general no hay forma de evitar que las estaciones en otra red cercana transmitan tráfico conflictivo.

El segundo problema es que los rangos de transmisión de las distintas estaciones pueden ser diferentes. Con un cable, el sistema se diseña de tal forma que todas las estaciones se puedan escuchar entre sí. Con las complejidades de la propagación de RF, esta situación no es válida para las estaciones inalámbricas. En consecuencia, pueden surgir situaciones como el problema de la terminal oculta que vimos antes. Como no todas las estaciones están dentro del alcance de radio de todas las demás, las transmisiones que se realizan en una parte de una celda tal vez no se reciban en las demás partes de la misma celda.

Para reducir las ambigüedades con respecto a qué estación va a transmitir, el 802.11 define la detección del canal como un proceso que consiste tanto de una detección física como de una detección virtual. En la detección física sólo se verifica el medio para ver si hay una señal válida. En la detección virtual, cada estación mantiene un registro lógico del momento en que se usa el canal rastreando el **NAV** (Vector de Asignación de Red, del inglés *Network Allocation Vector*). Cada trama lleva un campo NAV que indica cuánto tiempo tardará en completarse la secuencia a la que pertenece esta trama. Las estaciones que escuchen por casualidad esta trama saben que el canal estará ocupado durante el periodo indicado por el NAV, sin importar que puedan detectar o no una señal física. Por ejemplo, el NAV de una trama de datos indica el tiempo necesario para enviar una confirmación de recepción. Todas las estaciones que escuchen la trama de datos se retardarán durante el periodo de la confirmación de recepción, puedan o no escucharla.

Hay un mecanismo RTS/CTS opcional que usa el NAV para evitar que las terminales envíen tramas al mismo tiempo como terminales ocultas. Este mecanismo se muestra en la Ilustración 73. En este ejemplo, A desea enviar a B. C es una estación 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 alcance de A.

El protocolo empieza cuando A decide que desea enviar datos a B. A empieza por enviar una trama RTS a B para solicitar permiso de enviarle una trama. Si B recibe esta solicitud, responde con una trama CTS para indicar que el canal está libre para enviar. Al recibir la CTS, A envía su trama e inicia un temporizador ACK. Al recibir de forma correcta la trama de datos, B responde con una trama ACK para completar el intercambio. Si el temporizador ACK de A expira antes de que la trama ACK vuelva a ella, se considera como una colisión y se lleva a cabo todo el protocolo de nuevo, después de un retroceso.

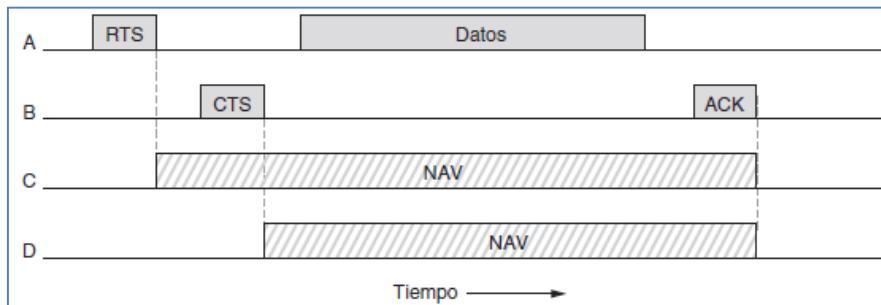


Ilustración 73 - Detección de canal virtual mediante CSMA/CA.

Ahora consideremos este intercambio desde los puntos de vista de C y D. C está dentro del alcance de A, por lo que puede recibir la trama RTS. Si pasa esto, se da cuenta de que alguien pronto va a enviar datos. A partir de la información proporcionada en la solicitud RTS, C puede estimar cuánto tardará la secuencia, incluyendo la trama ACK final. Entonces, por el bien de todos desiste de transmitir cualquier cosa hasta que el intercambio esté completo. A continuación actualiza su registro del NAV para indicar que el canal está ocupado, como se muestra en la Ilustración 73. D no escucha el RTS pero sí el CTS, por lo que también actualiza su NAV. Es necesario tener en cuenta que las señales NAV no se transmiten; sólo son recordatorios internos para mantenerse en silencio durante cierto periodo.

Pero aunque el método RTS/CTS suena bien en teoría, es uno de esos diseños que ha demostrado ser de poco valor en la práctica. No es útil para las tramas cortas (que se envían en vez de la trama RTS) ni para el AP (que todos pueden escuchar, por definición). Para otras situaciones, sólo reduce la operación. El método RTS/CTS en 802.11 es un poco distinto al del protocolo MACA que vimos, ya que todo el que escucha la trama RTS o CTS permanece en silencio para permitir que la trama ACK llegue a su destino sin que haya una colisión. Debido a esto, no es útil con las terminales expuestas como en el caso de MACA, sólo con las terminales ocultas. Lo más frecuente es que haya unas cuantas terminales ocultas y CSMA/CA les ayuda al reducir la velocidad de las estaciones que transmiten sin éxito, sin importar cuál sea la causa, para que sus transmisiones tengan más probabilidades de tener éxito.

CSMA/CA con detección física y virtual es el núcleo del protocolo 802.11. Sin embargo, existen otros mecanismos que se han desarrollado para trabajar con él. Cada uno de estos mecanismos fue impulsado por las necesidades de la operación real, por lo que los analizaremos de manera breve.

La primera necesidad que analizaremos es la **confiabilidad**. En contraste con las redes convencionales de cables, las redes inalámbricas son ruidosas y poco confiables, lo cual se debe en gran parte a la interferencia de otros tipos de dispositivos que también usan las bandas ISM sin licencia. El uso de confirmaciones de recepción y retransmisiones es de poca ayuda si es baja la probabilidad de lograr que una trama llegue a su destino es poca, en primer lugar.

La principal estrategia que se utiliza en este caso para incrementar las transmisiones exitosas es reducir la tasa de transmisión. Las tasas más bajas usan modulaciones más robustas que tienen mayor probabilidad de ser recibidas correctamente para una relación señal a ruido dada. Si se pierden demasiadas tramas, una estación puede reducir la tasa. Si se entregan tramas con pocas pérdidas, la estación puede algunas veces probar una tasa más alta para ver si es conveniente usarla.

Otra estrategia para mejorar la probabilidad de que la trama llegue sin daños es enviar tramas más cortas. Para implementar tramas más cortas es necesario reducir el tamaño máximo del mensaje que se aceptará de la capa de red. Como alternativa, el 802.11 permite dividir las tramas en piezas más pequeñas llamadas **fragmentos**, cada una con su propia suma de verificación. El tamaño del fragmento no es fijo según el estándar, sino un parámetro que el AP puede ajustar. Los fragmentos se enumeran en forma individual y su confirmación de recepción se realiza mediante un protocolo de parada y espera. Una vez que se adquiere el canal, se envían los múltiples fragmentos como una ráfaga. Van uno después del otro con una confirmación de recepción (y posiblemente retransmisiones) entre ellos, hasta que se haya enviado con éxito toda la trama o el tiempo de transmisión llegue al máximo permitido. El mecanismo NAV mantiene a las demás estaciones en silencio sólo hasta la siguiente confirmación de recepción, pero se utiliza otro mecanismo para permitir el envío de una ráfaga de fragmentos sin que otras estaciones envíen una trama a la mitad de la transmisión.

La segunda necesidad que estudiaremos es el **ahorro de energía**. La vida de las baterías siempre es un asunto importante para los dispositivos inalámbricos móviles. El estándar 802.11 pone atención a la cuestión de la administración de energía, de modo que los clientes no tengan que desperdiciarla cuando no tengan información qué enviar o recibir. El mecanismo básico para ahorrar energía se basa en las **tramas baliza** (*beacon frames*). Las balizas son difusiones periódicas que realiza el AP (por ejemplo, cada 100 ms). Las tramas anuncian la presencia del AP a los clientes y llevan los parámetros del sistema, como el identificador del AP, el tiempo, cuánto falta para la siguiente baliza y la configuración de seguridad. Los clientes pueden establecer un bit de administración de energía en las tramas que envían al AP para indicarle que entrarán en el **modo de ahorro de energía**. En este modo, el cliente puede dormitar y el AP pondrá en el búfer el tráfico destinado a este cliente. Para verificar el tráfico entrante, el cliente se despierta durante cada baliza y verifica un mapa de tráfico que se envía como parte de ella. Este mapa indica al cliente si hay tráfico en el búfer. De ser así, el cliente envía un mensaje de sondeo al AP, quien a su vez le envía el tráfico que está en el búfer. Después el cliente puede regresar al modo suspendido hasta que se envíe la siguiente baliza.

En 2005 se agregó otro mecanismo de ahorro de energía, conocido como **APSD** (Entrega Automática con Ahorro de Energía, del inglés *Automatic Power Save Delivery*). Con este nuevo mecanismo, el AP coloca las tramas en el búfer y las envía a un cliente justo después de que éste envía tramas al AP. Así, el cliente puede

regresar al modo suspendido hasta que tenga más tráfico para enviar (y recibir). Este mecanismo funciona bien para las aplicaciones como VoIP que tienen tráfico frecuente en ambas direcciones.

La tercera y última necesidad que examinaremos es la **calidad del servicio**. Cuando el tráfico VoIP en el ejemplo anterior compite con el tráfico de igual a igual, el primero es el que sufre, ya que se retrasará debido a la contención con el tráfico de igual a igual que requiere un ancho de banda alto, aun cuando el ancho de banda de VoIP es bajo. Es probable que los retardos degraden las llamadas de voz. Para evitar esta degradación, sería conveniente dejar que el tráfico VoIP vaya adelante del tráfico de igual a igual, puesto que es de mayor prioridad.

El estándar IEEE 802.11 tiene un ingenioso mecanismo para proveer este tipo de calidad de servicio, el cual se introdujo como un conjunto de extensiones bajo el nombre 802.11e en 2005. Su función consiste en extender el CSMA/CA con intervalos cuidadosamente definidos entre las tramas. Después de enviar una trama, se requiere cierta cantidad de tiempo inactivo antes de que una estación pueda enviar otra para verificar si el canal ya no se está usando. El truco es definir distintos intervalos para los distintos tipos de tramas.

En la Ilustración 74 se ilustran cinco intervalos. El intervalo entre las tramas de datos regulares se conoce como **DIFS** (Espaciado Entre Tramas DCF, del inglés *DCF InterFrame Spacing*). Cualquier estación puede intentar adquirir el canal para enviar una nueva trama después de que el medio haya estado inactivo durante un tiempo DIFS. Se aplican las reglas de contención usuales y tal vez se requiera el retroceso exponencial binario si ocurre una colisión. El intervalo más corto es **SIFS** (Espaciado Corto Entre Tramas, del inglés *Short InterFrame Spacing*) y se utiliza para permitir que las partes en un diálogo sencillo tengan la oportunidad de tomar el primer turno. Algunos ejemplos son: permitir que el receptor envíe una trama ACK, otras secuencias de tramas de control como RTS y CTS, o permitir que un emisor transmita una ráfaga de fragmentos. Enviar el siguiente fragmento después de esperar sólo un tiempo SIFS es lo que evita que otra estación irrumpa con una trama a mitad del intercambio.

Los dos intervalos **AIFS** (Espaciado Entre Tramas de Arbitraje, del inglés *Arbitration InterFrame Space*) muestran ejemplos de dos niveles de prioridad distintos. El intervalo corto, AIFS₁, es más pequeño que el intervalo DIFS pero más largo que SIFS. El AP lo puede usar para transportar voz u otro tipo de tráfico de alta prioridad al inicio de la línea. El AP esperará un intervalo más corto antes de enviar el tráfico de voz, y por ende lo enviará antes del tráfico regular. El intervalo largo, AIFS₄, es más largo que DIFS. Se utiliza para el tráfico de fondo que se puede aplazar hasta después del tráfico regular. El AP esperará un intervalo más largo antes de enviar este tráfico, para dar al tráfico regular la oportunidad de transmitir primero. El mecanismo completo de calidad del servicio define cuatro distintos niveles de prioridad que tienen diferentes parámetros de retroceso, así como diferentes parámetros de inactividad.

El último intervalo, **EIFS** (Espaciado Entre Tramas Extendido, del inglés *Extended InterFrame Spacing*), lo utiliza sólo una estación que acaba de recibir una trama defectuosa o desconocida, para reportar el problema. La idea es que, como el receptor tal vez no tenga idea de lo que está sucediendo, debería esperar un poco para evitar interferir con un diálogo existente entre dos estaciones.

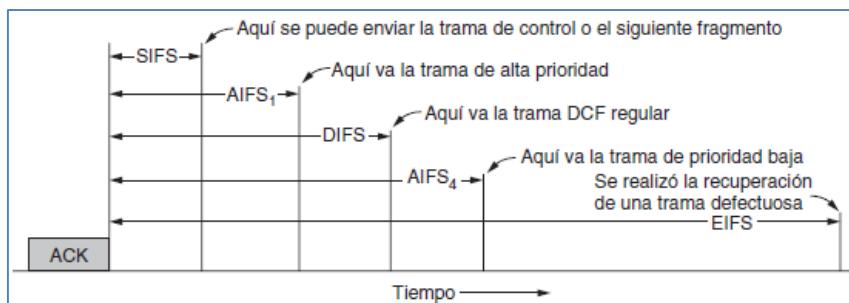


Ilustración 74 - Espaciado entre tramas en el 802.11.

Una parte adicional de las extensiones de calidad del servicio es la noción de una **TXOP**, u **oportunidad de transmisión**. El mecanismo original CSMA/CA permitía que las estaciones enviaran una trama a la vez. Este diseño estaba bien hasta que aumentó el rango de las tasas de transmisión. En el 802.11a/g, una estación

podría enviar a 6 Mbps y otra estación podría enviar a 54 Mbps. Las dos tienen oportunidad de enviar una trama, pero la estación de 6 Mbps se tarda nueve veces más (ignorando las sobrecargas fijas) que la estación de 54 Mbps en enviar su trama. Esta disparidad tiene el desafortunado efecto secundario de reducir la velocidad de un emisor rápido que compite con un emisor lento, a una tasa aproximada a la del emisor lento. Este problema se conoce como la **anomalía de tasa**.

Con las oportunidades de transmisión, cada estación recibe una cantidad equivalente de tiempo aire, no un número equivalente de tramas. Las estaciones que envían a una tasa más alta durante su tiempo aire obtendrán una velocidad de transmisión real más alta.

La estructura de trama 802.11

El estándar 802.11 define tres clases diferentes de tramas en el aire: de datos, de control y de administración. Cada una tiene un encabezado con una variedad de campos que se utilizan dentro de la subcapa MAC. Además, hay algunos encabezados utilizados por la capa física, pero como éstos tienen que ver en su mayor parte con las técnicas de modulación utilizadas, no los trataremos aquí.

Analizaremos como ejemplo el formato de la trama de datos, que se muestra en la Ilustración 75. Primero está el campo de **Control de trama**, que consta de 11 subcampos. El primero es la *Versión de protocolo*, que se establece como 00. Está ahí para que las futuras versiones del protocolo 802.11 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). Para una trama de datos regular (sin calidad de servicio), se establecen en 10 y 0000 en binario. Los bits *Para DS* y *De DS* se establecen para indicar que la trama va hacia o viene de la red conectada a los APs, a la cual se le conoce como sistema de distribución. El bit *Más fragmentos* indica que siguen más fragmentos. El bit *Retransmitir* marca una retransmisión de una trama que se envió antes. El bit de *Administración de energía* indica que el emisor va a entrar al modo de ahorro de energía. El bit *Más datos* indica que el emisor tiene tramas adicionales para el receptor. El bit *Trama protegida* indica que el cuerpo de la trama se cifró por seguridad. Por último, el bit de *Orden* indica al receptor que la capa superior espera que la secuencia de tramas llegue de modo riguroso en orden.

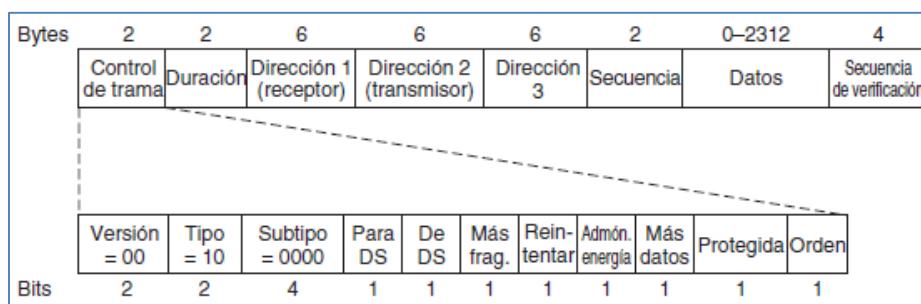


Ilustración 75 - Formato de la trama de datos 802.11.

El segundo campo de la trama de datos, el campo **Duración**, indica cuánto tiempo ocuparán el canal la longitud de la trama y su confirmación de recepción, lo cual se mide en microsegundos. Está presente en todos los tipos de tramas, incluyendo las tramas de control, y es lo que utilizan las estaciones para administrar el mecanismo NAV.

Después siguen las direcciones. Las tramas de datos que se envían hacia o se reciben de un AP tienen tres direcciones, todas en formato estándar de IEEE 802. La primera dirección es el receptor, y la segunda dirección es el transmisor. Recuerde que el AP sólo es un punto de relevo para las tramas, a medida que viajan entre un cliente y otro punto en la red, tal vez un cliente distante o un portal de Internet. La tercera dirección provee este punto final distante.

El campo **Secuencia** numera las tramas de manera que se puedan detectar tramas duplicadas. De los 16 bits disponibles, 4 identifican el fragmento y 12 transportan un número que avanza con cada nueva transmisión. El campo **Datos** contiene la carga útil, hasta 2312 bytes. Los primeros bytes de esta carga útil están en un formato conocido como **LLC** (Control de Enlace Lógico, del inglés *Logical Link Control*). Esta capa es la unión

que identifica al protocolo de capa superior (por ejemplo, IP) al que se deben pasar las cargas útiles. Por último tenemos la **Secuencia de verificación de tramas**, que viene siendo la misma CRC de 32 bits.

Las tramas de administración tienen el mismo formato que las tramas de datos, además de un formato para la parte de los datos que varía con el subtipo (por ejemplo, los parámetros en las tramas de baliza).

Las tramas de control son cortas. Al igual que todas las tramas, tienen los campos Control de trama, Duración y Secuencia de verificación de trama. Sin embargo, ellas pueden tener sólo una dirección y ninguna porción de datos. La mayoría de la información clave se transmite mediante el campo Subtipo (por ejemplo, ACK, RTS y CTS).

Servicios

El estándar 802.11 define los servicios que los clientes, los puntos de acceso y la red que los conecta deben proveer para poder ser una LAN inalámbrica que se apegue a dicho estándar. Estos servicios se dividen en varios grupos.

El servicio de **asociación** lo utilizan las estaciones móviles para conectarse ellas mismas a los AP. Por lo general, se utiliza después de que una estación se mueve dentro del alcance de radio del AP. Al llegar, la estación conoce la identidad y las capacidades del AP, ya sea mediante tramas baliza o preguntando directamente al AP. Entre las capacidades se incluyen las *tasas de datos soportadas*, los *arreglos de seguridad*, las *capacidades de ahorro de energía*, el soporte de la *calidad del servicio*, etcétera. La estación envía una solicitud para asociarse con el AP. Éste puede aceptar o rechazar dicha solicitud.

La **reasociación** permite que una estación cambie su AP preferido. Esta herramienta es útil para las estaciones móviles que cambian de un AP a otro en la misma LAN 802.11 extendida. Si se utiliza en forma correcta, no se perderán datos como consecuencia del traspaso (pero al igual que Ethernet, el 802.11 es sólo un servicio de mejor esfuerzo). También es posible que la estación o el AP se **desasocien**, con lo que se rompería su relación. Una estación debe usar este servicio antes de desconectarse o salir de la red. El AP lo puede usar antes de desconectarse por cuestión de mantenimiento.

Las estaciones también se deben **autenticar** antes de poder enviar tramas por medio del AP, pero la autenticación se maneja en formas distintas dependiendo del esquema de seguridad elegido. Si la red 802.11 está “*abierta*”, cualquiera puede usarla. En caso contrario, se requieren credenciales para autenticarse. El esquema recomendado, conocido como **WPA2** (Acceso Protegido WiFi 2, del inglés *WiFi Protected Access 2*), implementa la seguridad según lo definido en el estándar 802.11i (el WPA simple es un esquema interno que implementa un subconjunto del 802.11i). En el WPA2, el AP se puede comunicar con un servidor de autenticación que tenga una base de datos con nombres de usuario y contraseñas para determinar si la estación puede acceder a la red. Como alternativa se puede configurar una *clave precompartida* o contraseña de red. Se intercambian varias tramas entre la estación y el AP con un reto y respuesta que permite a la estación demostrar que tiene las credenciales apropiadas. Este intercambio ocurre después de la asociación.

El esquema que se utilizaba antes de WPA se llama **WEP** (Privacidad Equivalente a cableado, del inglés *Wired Equivalent Privacy*). En este esquema, la autenticación con una clave precompartida ocurre antes de la asociación. Sin embargo, no se recomienda su uso debido a fallas de diseño que comprometen fácilmente su seguridad.

Una vez que las tramas llegan al AP, el servicio de **distribución** determina cómo encaminarlas. Si el destino es local para el AP, las tramas se pueden enviar en forma directa por el aire. En caso contrario, habrá que reenviarlas por la red alámbrica. El servicio de **integración** maneja cualquier traducción necesaria para enviar una trama fuera de la LAN 802.11, o para que llegue desde el exterior de la LAN 802.11. Aquí el caso común es conectar la LAN inalámbrica a Internet.

Y como lo esencial aquí es la transmisión de datos, es lógico que la red 802.11 provea el servicio de **entrega de datos**. Este servicio permite a las estaciones transmitir y recibir datos mediante el uso de los protocolos que describimos antes. Como el estándar 802.11 está modelado en base a Ethernet y no se garantiza que la transmisión por Ethernet sea 100% confiable, tampoco se garantiza que la transmisión por 802.11 sea confiable. Las capas superiores deben lidiar con la detección y corrección de errores.

La señal inalámbrica es de difusión. Para mantener confidencial la información que se envía por una LAN inalámbrica, hay que cifrarla. Para lograr esto se utiliza un servicio de **privacidad** que administra los detalles del cifrado y el descifrado. El algoritmo de cifrado para WPA2 se basa en el estándar **AES** (Estándar de Cifrado Avanzado, del inglés *Advanced Encryption Standard*) del gobierno de Estados Unidos. Las claves que se utilizan para el cifrado se determinan durante el procedimiento de autenticación.

Para manejar tráfico con distintas prioridades, existe un servicio llamado **programación de tráfico QoS**, el cual utiliza los protocolos que describimos antes para dar al tráfico de voz y de video un tratamiento preferencial, en comparación con el tráfico del mejor esfuerzo y de fondo. Hay un servicio complementario que también provee la sincronización de los temporizadores de las capas superiores. Esto permite a las estaciones coordinar sus acciones, que pueden ser útiles para procesar los medios.

Por último hay dos servicios que ayudan a las estaciones a administrar la forma en que utilizan el espectro. El servicio de **control de potencia de transmisión** brinda a las estaciones la información que necesitan para cumplir con los límites regulatorios sobre la potencia de transmisión, que varían de una región a otra. El servicio de **selección de frecuencia dinámica** ofrece a las estaciones la información que necesitan para evitar transmitir en frecuencias de la banda de 5 GHz que se utilicen para radar en las proximidades.

Comutación en la capa de enlace de datos

Muchas organizaciones tienen varias redes LAN y desean interconectarlas. ¿No sería conveniente si tan sólo pudiéramos unir las redes LAN para formar una LAN más grande? De hecho, este tipo de redes se puede conectar mediante dispositivos llamados puentes. Los switchs de Ethernet que describimos antes son un nombre moderno para los **puentes**; proveen una funcionalidad que va más allá de los hubs de Ethernet clásica y Ethernet para facilitar la unión de varias redes LAN en una red más grande y veloz. Utilizaremos los términos “puente” y “switch” para indicar lo mismo.

Los puentes operan en la capa de enlace de datos, por lo que examinan las direcciones de la capa de enlace de datos para reenviar tramas. Como no tienen que examinar el campo de carga útil de las tramas que reenvían, pueden manejar paquetes IP al igual que otros tipos de paquetes, como Apple-Talk. En contraste, los enruteadores examinan las direcciones de los paquetes y realizan su trabajo de enruteamiento con base en ellas, por lo que sólo funcionan con los protocolos para los cuales se diseñaron.

En esta sección analizaremos la forma en que funcionan los puentes y cómo se utilizan para unir varias redes LAN físicas en una sola LAN lógica. También veremos cómo hacer lo inverso y tratar una LAN física como varias redes LAN lógicas, llamadas redes **VLAN** (LAN virtuales, del inglés *Virtual LANs*). Ambas tecnologías proveen una flexibilidad conveniente para administrar redes.

Usos de los puentes

Antes de entrar de lleno a la tecnología de los puentes, veamos algunas situaciones comunes en las cuales se utilizan los puentes. Mencionaremos tres razones por las cuales una sola organización podría terminar trabajando con varias LAN.

En primer lugar, muchas universidades y departamentos corporativos tienen sus propias redes LAN para conectar sus propias computadoras personales, servidores y dispositivos como impresoras. Dado que los objetivos de los distintos departamentos difieren, los distintos departamentos pueden establecer diferentes redes LAN, sin importarles lo que hagan los demás departamentos. Pero tarde o temprano surge la necesidad de interacción, y aquí es donde entran los puentes. En este ejemplo surgieron múltiples redes LAN debido a la autonomía de sus propietarios.

En segundo lugar, la organización puede estar distribuida geográficamente en varios edificios, separados por distancias considerables. Puede ser más económico tener redes LAN independientes en cada edificio y conectarlas mediante puentes y unos cuantos enlaces de fibra óptica de larga distancia que tender todos los cables hacia un solo switch central. Incluso si es fácil tender los cables, existen límites en cuanto a sus longitudes (por ejemplo, 200 m para GigabitEthernet de par trenzado). La red no funcionaría con cables más largos debido a la excesiva atenuación de la señal, o al retardo de viaje redondo. La única solución es dividir la LAN e instalar puentes para unir las piezas y poder incrementar la distancia física total que se puede cubrir.

En tercer lugar, tal vez sea necesario dividir lo que por lógica es una sola LAN en varias redes LAN individuales (conectadas mediante puentes) para manejar la carga. Por ejemplo, en muchas universidades grandes, hay miles de estaciones de trabajo disponibles para los estudiantes y el cuerpo docente. Las empresas también pueden tener miles de empleados. La escala de este sistema hace imposible poner todas las estaciones de trabajo en una sola LAN; hay muchas más computadoras que puertos en cualquier hub Ethernet y más estaciones de lo que se permite en una sola Ethernet clásica.

Incluso si fuera posible cablear todas las estaciones de trabajo juntas, al colocar más estaciones en un hub Ethernet o en una red Ethernet clásica no se agrega capacidad. Todas las estaciones comparten la misma cantidad fija de ancho de banda. Entre más estaciones haya, menor será el ancho de banda promedio por estación.

Sin embargo, dos redes LAN separadas tienen el doble de la capacidad de una sola LAN. Los puentes permiten unir redes LAN y mantener al mismo tiempo esta capacidad. La clave es no enviar tráfico a los puertos en los que no se necesita, de modo que cada LAN pueda operar a toda velocidad. Este comportamiento también aumenta la confiabilidad, ya que en una sola LAN, un nodo defectuoso que siga transmitiendo un flujo continuo de basura puede llegar a obstruir toda la LAN completa. Al decidir qué reenviar o no, los puentes actúan como puertas contra incendios en un edificio, pues evitan que un solo nodo errático haga fallar todo el sistema.

Para que estos beneficios pudieran estar fácilmente disponibles, los puentes ideales tendrían que ser totalmente transparentes. Debería ser posible comprar los puentes, conectar los cables de LAN en los puentes y que todo funcionara a la perfección en un instante. No debería existir la necesidad de cambios de hardware o de software, ni de configurar switchs de direcciones o descargar tablas de enrutamiento o parámetros, nada de eso. Simplemente conectar los cables y seguir con nuestras actividades cotidianas. Lo que es más, la operación de las redes LAN existentes no se debería ver afectada por los puentes para nada. En cuanto a las estaciones, no debería haber ninguna diferencia observable en cuanto a si son parte o no de una LAN con puente. Debería ser igual de fácil mover estaciones alrededor de una LAN con puente que moverlas en una sola LAN.

Aunque resulta sorprendente, en realidad es posible crear puentes que sean transparentes. Se utilizan dos algoritmos: un algoritmo de aprendizaje hacia atrás para detener el tráfico que se envía a donde no es necesario, y un algoritmo de árbol de expansión para romper los ciclos que se pueden formar cuando los switchs se conectan entre sí de manera no intencional. Ahora analizaremos cada uno de estos algoritmos para ver cómo se logra esta magia.

Puentes de aprendizaje

La topología de dos redes LAN conectadas por un puente se muestra en la Ilustración 76 para dos casos. En el lado izquierdo, dos redes LAN *multiderivación* (por ejemplo, redes Ethernet clásicas) se unen mediante una estación especial (el puente) que se sitúa entre ambas redes LAN. Del lado derecho, se unen redes LAN con cables punto a punto, incluyendo un hub. Los puentes son los dispositivos a los que se conectan las estaciones y el hub. Si la tecnología de LAN es Ethernet, los puentes son mejor conocidos como switchs Ethernet.

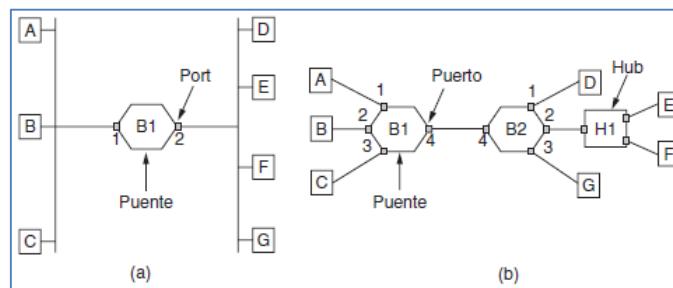


Ilustración 76 - (a) Puente que conecta dos redes LAN multiderivación. (b) Puentes (y un hub) que conectan siete estaciones punto a punto.

Los puentes se desarrollaron cuando se usaban redes Ethernet clásicas, por lo que a menudo se muestran en topologías con cables multiderivación, como en la Ilustración 76(a). Sin embargo, todas las topologías en la

actualidad están compuestas de cables punto a punto y switchs. Los puentes funcionan de la misma forma en ambas configuraciones. Todas las estaciones conectadas al mismo puerto en un puente pertenecen al mismo dominio de colisión, y éste es distinto al dominio de colisión para otros puertos. Si hay más de una estación, como en una red Ethernet clásica, un hub o un enlace half-dúplex, se utiliza el protocolo CSMA/CD para enviar tramas.

Sin embargo, hay una diferencia en cuanto a la forma en que se construyen las redes LAN con puentes. Para conectar redes LAN multiderivación con puentes, se agrega un puente como una nueva estación en cada LAN multiderivación, como en la Ilustración 76(a). Para conectar redes LAN punto a punto mediante puentes, los hubs se conectan a un puente o, lo que es preferible, se reemplazan con un puente para incrementar el desempeño. En la Ilustración 76(b) los puentes reemplazaron a todos los hubs excepto uno.

También se pueden conectar distintos tipos de cables a un puente. Por ejemplo, el cable que conecta el puente *B1* con el puente *B2* en la Ilustración 76(b) podría ser un enlace de fibra óptica de larga distancia, mientras que el cable que conecta los puentes con las estaciones podría ser una línea de par trenzado de corta distancia. Esta disposición es útil para conectar redes LAN mediante puentes en distintos edificios.

Ahora consideremos lo que ocurre dentro de los puentes. Cada puente opera en modo promiscuo; es decir, acepta cada una de las tramas que transmiten las estaciones conectadas a cada uno de sus puertos. El puente debe decidir si va a reenviar o desechar cada trama y, en caso de que sea la primera opción, también debe decidir por qué puerto enviar la trama. Esta decisión se basa en la dirección de destino. Como ejemplo, considere la topología de la Ilustración 76(a). Si la estación *A* envía una trama a la estación *B*, el puente *B1* recibirá la trama en el puerto 1. Esta trama se puede desechar de inmediato sin más preámbulos, debido a que ya se encuentra en el puerto correcto. Sin embargo, suponga que en la topología de la Ilustración 76(b) la estación *A* envía una trama a *D*. El puente *B1* recibirá la trama en el puerto 1 y la enviará por el puerto 4. Después el puente *B2* recibirá la trama en su puerto 4 y la enviará por el puerto 1.

Una forma simple de implementar este esquema es mediante una gran tabla (*hash*) dentro del puente. La tabla puede listar cada posible destino y a qué puerto de salida pertenece. Por ejemplo, en la Ilustración 76(b), la tabla en *B1* listaría a *D* como perteneciente al puerto 4, ya que todo lo que *B1* tiene que saber es por qué puerto enviar las tramas para llegar a *D*. El que, de hecho, se lleven a cabo más reenvíos posteriormente cuando la trama llegue a *B2* no es de interés para *B1*.

Cuando se conectan por primera vez los puentes, 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 por todos los puertos a los que está conectado el puente, excepto por el que llegaron. Con el paso del tiempo, los puentes aprenden dónde están los destinos. Una vez conocido un destino, las tramas destinadas para él se colocan sólo en el puerto apropiado; no se inundan.

El algoritmo que usan los puentes es el de **aprendizaje hacia atrás**. Como ya mencionamos, los puentes funcionan en modo promiscuo y de esta manera pueden ver todas las tramas que se envían por cualquiera de sus puertos. Al analizar las direcciones de origen, pueden saber cuáles máquinas están disponibles en cuáles puertos. Por ejemplo, si el puente *B1* de la Ilustración 76(b) ve una trama en el puerto 3 que proviene de *C*, sabe que es posible acceder a *C* por medio del puerto 3, así que registra una entrada en su tabla de hash. Cualquier trama subsecuente dirigida a *C* que llegue desde el puente *B1* por cualquier puerto se reenviará al puerto 3.

La topología puede cambiar conforme las máquinas y los puentes se enciendan y apaguen, o cuando se trasladan 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. Cada vez que llega una trama cuyo origen ya está en la tabla, su entrada se actualiza con la hora actual. Así, la hora asociada a cada entrada indica la última vez que se registró una trama proveniente de esa máquina.

Un proceso en el puente analiza de manera periódica la tabla de hash y purga todas las entradas que tengan más de algunos minutos de antigüedad. 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 se inundará hasta que la máquina misma envíe una trama.

El procedimiento de enrutamiento para una trama entrante depende del puerto por el que llegue (el puerto de origen) y de la dirección a la cual está destinada (la dirección de destino). El procedimiento se muestra a continuación.

1. Si el puerto para la dirección de destino es el mismo que el puerto de origen, se desecha la trama.
2. Si el puerto para la dirección y el puerto de origen son diferentes, se reenvía la trama por el puerto de destino.
3. Si se desconoce el puerto de destino, se recurre a la inundación y envía la trama por todos los puertos excepto el de origen.

Tal vez se pregunte si el primer caso puede ocurrir con enlaces punto a punto. La respuesta es que puede ocurrir si se usan hubs para conectar un grupo de computadoras a un puente. En la Ilustración 76(b) se muestra un ejemplo, en donde las estaciones *E* y *F* se conectan al hub *H1*, el cual a su vez está conectado al puente *B2*. Si *E* envía una trama a *F*, el hub la retransmitirá a *B2* y también a *F*. La trama llegará a *B2* en el puerto 4, que ya es el puerto de salida correcto para llegar al destino. El puente *B2* sólo tiene que desechar la trama.

Como los puentes sólo analizan las direcciones MAC para decidir cómo reenviar las tramas, es posible empezar a reenviar tan pronto como llega el campo del encabezado de destino, antes de que haya llegado el resto de la trama (siempre y cuando la línea de salida esté disponible, claro está). Este diseño reduce la latencia de pasar a través del puente, así como el número de tramas que el puente debe ser capaz de colocar en el búfer. Se denomina **conmutación al vuelo** (*cut-through*) o **enrutamiento de agujero de gusano** (*wormhole*) y por lo general se maneja en el hardware.

Podemos ver la operación de un puente en términos de las pilas de protocolos, para comprender lo que significa ser un dispositivo de capa de enlace. Considere una trama que se envió de la estación *A* a la estación *D* en la configuración de la Ilustración 76(a), en donde las redes LAN son Ethernet. La trama pasará a través de un puente. La vista de procesamiento de la pila de protocolos se muestra en la Ilustración 77.

El paquete llega de una capa superior y desciende a la capa MAC Ethernet. Adquiere un encabezado de Ethernet (y también un terminador, que no se muestra en la figura). Esta unidad se pasa a la capa física, sale por el cable y el puente la recoge.

En el puente, la trama se pasa de la capa física a la capa MAC Ethernet. Esta capa tiene un procesamiento extendido, en comparación con la capa MAC en una estación. Pasa la trama a un retransmisor, todavía dentro de la capa MAC. La función de retransmisión del puente sólo usa el encabezado MAC Ethernet para determinar cómo manejar la trama. En este caso, pasa la trama a la capa MAC Ethernet del puerto utilizado para llegar a la estación *D*, y la trama continúa su camino.

En el caso general, la retransmisión en una capa dada puede escribir los encabezados para esa capa. En breve veremos un ejemplo con redes VLAN. En ningún caso el puente deberá ver dentro de la trama y descubrir que transmite un paquete IP; esto es irrelevante para el procesamiento del puente y violaría el uso de capas de protocolos. Cabe mencionar además que un puente con *k* puertos tendrá *k* instancias de capas MAC y físicas. El valor de *k* es 2 para nuestro ejemplo.

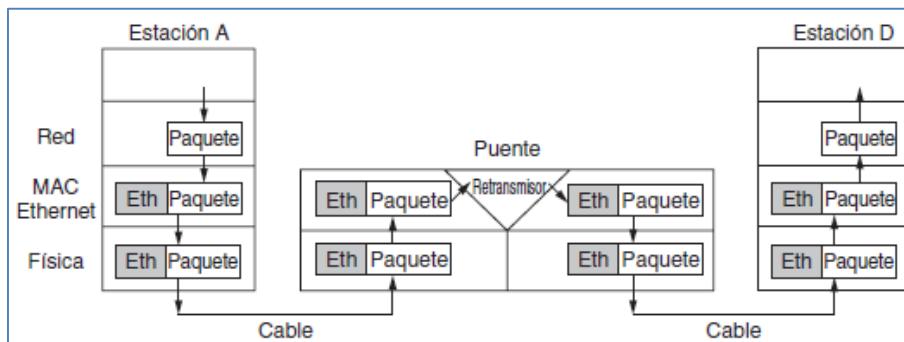


Ilustración 77 - Procesamiento de protocolos en un puente.

Puentes con árbol de expansión

Para incrementar la confiabilidad, se pueden usar enlaces redundantes entre puentes. En el ejemplo de la Ilustración 78, hay dos enlaces en paralelo entre un par de puentes. Este diseño asegura que si se corta un enlace, la red no se dividirá en dos conjuntos de computadoras que no se pueden comunicar entre sí.

Sin embargo, esta redundancia introduce algunos problemas adicionales, porque crea ciclos en la topología. En la Ilustración 78 podemos ver un ejemplo simple de estos problemas, al observar cómo se maneja la forma en que la estación A envía una trama a un destino que no se había observado antes. Cada puente sigue la regla normal para el manejo de destinos desconocidos, que es inundar la trama. Sea F_0 la trama de A que llega al puente B1. El puente envía copias de esta trama a todos sus otros puertos. Sólo consideraremos los puertos del puente que conectan B1 con B2 (aunque la trama se enviará también a los demás puertos). Como hay dos enlaces de B1 a B2, dos copias de la trama llegarán a B2. En la Ilustración 78 se muestran como F_1 y F_2 .

Poco después, el puente B2 recibe estas tramas. Sin embargo, no sabe (ni puede saber) que son copias de la misma trama, en vez de ser dos tramas distintas que se envían una después de la otra. Por lo tanto, el puente B2 toma la trama F_1 y envía copias de ella a todos los demás puertos; además toma a F_2 y envía copias de ella a todos los otros puertos. Esto produce las tramas F_3 y F_4 que se envían a través de los dos enlaces, de vuelta a B1. A continuación, el puente B1 ve dos nuevas tramas con destinos desconocidos y las copia de nuevo. Este ciclo continúa en forma indefinida.

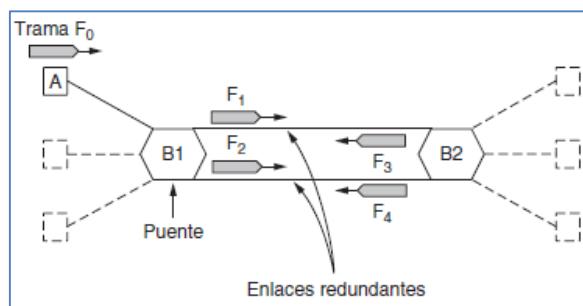


Ilustración 78 - Puentes con dos enlaces paralelos.

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 todos los puentes. En efecto, algunas conexiones potenciales entre los puentes se ignoran en el afán de construir una topología ficticia libre de ciclos, que sea un subconjunto de la topología actual.

Por ejemplo, en la figura Ilustración 79 vemos cinco puentes interconectados y que también tienen estaciones conectadas. Cada estación se conecta sólo a un puente. Hay algunas conexiones redundantes entre los puentes, de modo que las tramas se reenviarán en ciclos si se utilizan todos los enlaces. Podemos considerar esta topología como un grafo en el que los puentes son los nodos y los enlaces punto a punto son los bordes. El grafo se puede reducir a un árbol de expansión, el cual no tiene ciclos por definición, si eliminamos los enlaces que se muestran como líneas punteadas en la Ilustración 79. Si usamos este árbol de expansión, hay exactamente una ruta de cada estación a cada una de las demás estaciones. Una vez que los puentes se hayan puesto de acuerdo en cuanto al árbol de expansión, todos los reenvíos entre las estaciones se hacen a través del árbol de expansión. Puesto que existe una única ruta de cada origen a cada destino, es imposible que se produzcan ciclos.

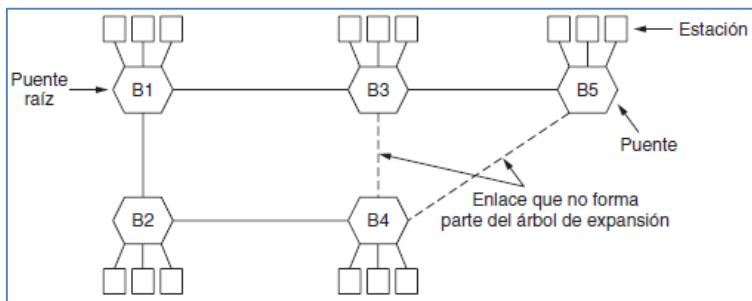


Ilustración 79 - Un árbol de expansión que conecta cinco puentes. Las líneas punteadas son enlaces que no forman parte del árbol de expansión.

Para construir el árbol de expansión, los puentes ejecutan un algoritmo distribuido. Cada puente difunde en forma periódica un mensaje de configuración a través de todos sus puertos hacia sus vecinos y procesa los mensajes que recibe de otros puentes, como veremos a continuación. Estos mensajes no se reenvían, ya que su propósito es construir el árbol y usarlo para los reenvíos.

Los puentes primero tienen que escoger una conexión que sea la raíz del árbol de expansión. Para hacer esta elección, cada uno incluye un identificador basado en su dirección MAC en el mensaje de configuración, junto con el identificador del puente que creen que es la raíz. El fabricante instala las direcciones MAC, que tienen la garantía de ser únicas en todo el mundo, por lo cual son identificadores convenientes y únicos. Los puentes seleccionan la conexión con el menor identificador para que sea la raíz. Después de haber intercambiado suficientes mensajes para esparcir las noticias, todos los puentes se pondrán de acuerdo en cuál será la raíz. En la Ilustración 79, el puente *B1* tiene el menor identificador y se convierte en la raíz.

A continuación se construye un árbol con las rutas más cortas de la raíz a cada uno de los puentes. En la Ilustración 79 se puede llegar a los puentes *B2* y *B3* directamente desde el puente *B1*, en un salto que sea una de las rutas más cortas. Se puede llegar al puente *B4* en dos saltos, a través de *B2* o de *B3*. Para romper este empate se selecciona la ruta por el puente con el menor identificador, así que se llega a *B4* a través de *B2*. Se puede llegar al puente *B5* en dos saltos a través de *B3*.

Para encontrar estas rutas más cortas, los puentes incluyen la distancia desde la raíz en sus mensajes de configuración. Cada puente recuerda la ruta más corta que encuentra hacia la raíz. Después, los puentes desactivan los puertos que no formen parte de la ruta más corta.

Aunque el árbol abarca todos los puentes, no todos los enlaces (e incluso los puentes) están necesariamente presentes en el árbol. Esto ocurre debido a que al desactivar los puertos se cortan algunos enlaces de la red para evitar los ciclos. Incluso después de que se ha establecido el árbol de expansión, el algoritmo continúa en ejecución durante la operación normal con el fin de detectar de manera automática los cambios en la topología y actualizar el árbol.

El algoritmo del árbol de expansión se estandarizó como el IEEE 802.1D y se usó durante muchos años. En 2001 se revisó para encontrar con más rapidez un nuevo árbol de expansión después de un cambio de topología.

Repetidores, hubs, puentes, switchs, enrutadores y puertas de enlace (gateways)

Hasta ahora hemos visto una variedad de formas para desplazar tramas y paquetes de una computadora a otra. Hemos mencionado repetidores, hubs, puentes, switchs, 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.

La clave para entender estos dispositivos es tener en cuenta que operan en distintas capas, como se ilustra en la Ilustración 80(a). La capa es importante porque los distintos dispositivos utilizan diferentes piezas de información para decidir cómo van a comutar. En un escenario común, el usuario genera algunos datos para enviarlos 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 le agrega su propio encabezado para formar un paquete de capa de red (por ejemplo, un paquete IP). En la Ilustración 80(b)

podemos ver el paquete IP sombreado en color gris. Después, el paquete pasa a la capa de enlace de datos, la cual agrega su propio encabezado y una suma de verificación (CRC), y entrega la trama resultante a la capa física para su transmisión; por ejemplo, sobre una LAN.

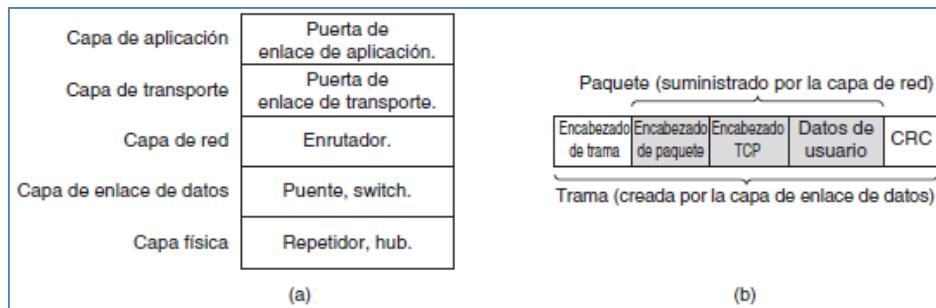


Ilustración 80 - (a) Qué dispositivo está en cada capa. (b) Tramas, paquetes y encabezados.

Ahora veamos los dispositivos de conmutación y cómo se relacionan con los paquetes y las tramas. En la parte inferior (en la capa física) se encuentran los **repetidores**. Éstos son dispositivos analógicos que funcionan con señales de los cables a los que están conectados. Una señal que aparece en un cable se limpia, amplifica y pone en otro cable. Los repetidores no distinguen entre tramas, paquetes o encabezados. Ellos comprenden los símbolos que codifican bits como voltios.

Un **hub** tiene varias líneas de entrada que unen de manera eléctrica. Las tramas que llegan a cualquiera de las líneas se envían por todas las demás. Si dos tramas llegan al mismo tiempo colisionarán, al igual que en un cable coaxial. Todas las líneas que convergen en un hub deben operar a la misma velocidad. A diferencia de los repetidores, los hubs (por lo general) no amplifican las señales entrantes y están diseñados para múltiples líneas de entrada, aunque las diferencias son ligeras. Al igual que los repetidores, los hubs son dispositivos de capa física que no examinan las direcciones de la capa de enlace ni las utilizan de ninguna manera.

Veamos a continuación la capa de enlace de datos, en donde se encuentran los puentes y los switches. Ya hemos visto algo de los puentes. Un **puente** conecta dos o más redes LAN. Al igual que un hub, un puente moderno cuenta con múltiples puertos, por lo general suficientes para tener de 4 a 48 líneas de entrada de cierto tipo. A diferencia de un hub, cada puerto está aislado para ser su propio dominio de colisión; si el puerto tiene una línea punto a punto full-dúplex, no se necesita el algoritmo CSMA/CD. Cuando llega una trama, el puente extrae la dirección de destino del encabezado y la busca en una tabla para averiguar a dónde debe enviar la trama. El puente sólo envía la trama por el puerto en el que se necesita y puede reenviar varias tramas al mismo tiempo.

Los puentes ofrecen un desempeño muy superior al de los hubs, además el aislamiento entre los puertos del puente también significa que las líneas de entrada pueden operar a distintas velocidades, e incluso tal vez con distintos tipos de redes. Un ejemplo común es un puente con puertos que se pueden conectar a redes Ethernet de 10, 100 y 1000 Mbps. Se requiere un búfer dentro del puente para aceptar una trama en un puerto y transmitirla por un puerto distinto. Si las tramas llegan con más rapidez de lo que se pueden retransmitir, el puente se puede quedar sin espacio de búfer y tal vez tenga que empezar a desechar tramas. Este problema existe aunque todos los puertos operen a la misma velocidad, ya que tal vez varios puertos envíen tramas a un puerto de destino dado.

Los puentes se diseñaron originalmente para poder unir distintos tipos de redes LAN; por ejemplo, una LAN Ethernet y una LAN Token Ring. Sin embargo, esto nunca funcionó bien debido a las diferencias entre las redes LAN. En los distintos formatos de trama se requieren procesos de copia y reformateo, para lo cual se necesita tiempo de CPU, es necesario calcular una nueva suma de verificación además de que se introduce la posibilidad de que haya errores sin detectar debido a bits defectuosos en la memoria del puente. Las distintas longitudes máximas de trama también son un problema grave sin una buena solución. En esencia, se deben desechar las tramas que son demasiado largas como para reenviarlas. Adiós a la transparencia.

Otras dos áreas en las que las redes LAN pueden diferir son la seguridad y la calidad del servicio. En consecuencia, cuando una trama debe viajar entre estos tipos de LAN, tal vez no se pueda proveer la seguridad

o calidad del servicio que espera el emisor. Por todas estas razones, los puentes modernos funcionan por lo general con un tipo de red, y los enrutadores son los que se usan para unir redes de distintos tipos.

Los **switchs** son otro nombre para los puentes modernos. Las diferencias se relacionan más con la comercialización que con las cuestiones técnicas, aunque hay algunos puntos que vale la pena conocer. Los puentes se desarrollaron cuando se usaba la Ethernet clásica, ellos tendían a unir pocas redes LAN y, por ende, tienen pocos puertos. Hoy en día es más popular el término “switch”. Además, todas las instalaciones modernas usan enlaces punto a punto por lo que cada computadora se conecta directamente a un switch y es lógico que tenga muchos puertos. Por último, “switch” también se utiliza como un término general. Con un puente, la funcionalidad es clara. Por otro lado, un switch se puede referir a un switch Ethernet o a un tipo de dispositivo por completo diferente que toma decisiones de reenvío, como un comutador telefónico (switch telefónico).

Hasta ahora hemos visto repetidores y hubs, que en realidad son bastante similares, así como puentes y switchs, que son aún más similares. Ahora pasaremos a los **enrutadores**, que son distintos de todos los anteriores componentes. Cuando un paquete llega a un enrutador, se quita el encabezado y el terminador de la trama, y se pasa el campo de carga útil de la trama al software de enrutamiento. Este software usa el encabezado del paquete para elegir una línea de salida. En un paquete IP, el encabezado contiene una dirección de 32 bits (IPv4) o 128 bits (IPv6). El software de enrutamiento no ve las direcciones de las tramas y ni siquiera sabe si el paquete llegó por una LAN o por una línea punto a punto. En la próxima unidad estudiaremos los enrutadores y el enrutamiento.

Una capa más arriba tenemos las **puertas de enlace de transporte**. Estos dispositivos conectan dos computadoras que utilizan diferentes protocolos de transporte orientados a conexión. Por ejemplo, imagine que una computadora que utiliza el protocolo TCP/IP orientado a conexión necesita comunicarse con una computadora que emplea un protocolo distinto de transporte orientado a conexión, el cual se conoce como SCTP. 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** entienden el formato y contenido de los datos; y pueden traducir los mensajes de un formato a otro. Por ejemplo, una puerta de enlace de correo electrónico puede traducir los mensajes de Internet en mensajes SMS para teléfonos móviles. Al igual que “switch”, “puerta de enlace” es algo así como un término general. Se refiere a un proceso de reenvío que opera en una capa alta.

Redes LAN virtuales

En los primeros días de las redes de área local, cables amarillos gruesos viajaban por los ductos de muchos edificios de oficinas. Conectaban a todas las computadoras por las que pasaban. 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. La geografía triunfaba sobre los gráficos organizacionales corporativos.

Todo cambió con el surgimiento de los cables de par trenzado y los hubs en la década de 1990. El cableado de los edificios se renovó (a un costo considerable) para instalar cables de par trenzado desde cada oficina hasta gabinetes centrales al final de cada pasillo o en una sala central de máquinas, como se observa en la Ilustración 81. Si el vicepresidente a cargo 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 FastEthernet).

En la actualidad, los cables han cambiado y los hubs se han convertido en switchs, pero el patrón de cableado sigue siendo el mismo. Este patrón hace posible la configuración de redes LAN lógicas en vez de físicas. Por ejemplo, si una empresa desea k redes LAN, podría comprar k switchs. Al elegir con cuidado qué conectores enchufar en qué switchs, los ocupantes de una LAN se pueden seleccionar de tal forma que tenga sentido organizacional, sin tener mucho en cuenta la geografía.

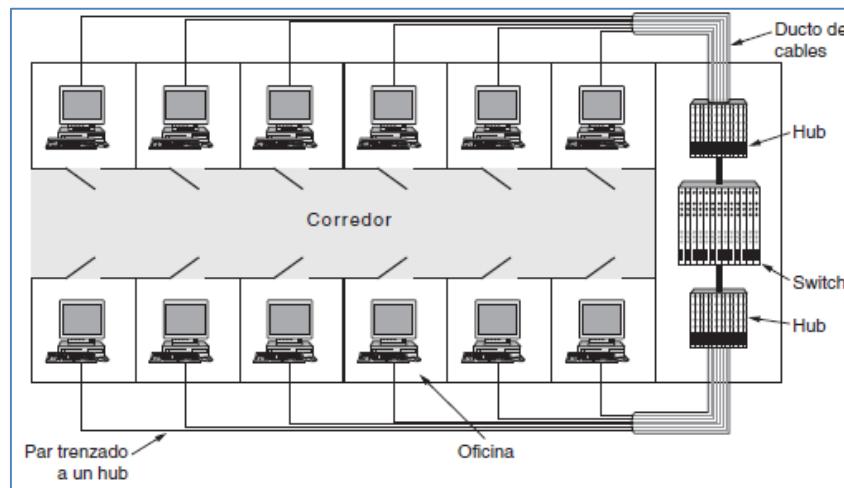


Ilustración 81 - Un edificio con cableado centralizado en el que se usan hubs y un switch.

¿Es importante quién está en qué LAN? Después de todo, en casi todas las organizaciones las redes LAN están interconectadas. En resumen, por lo general sí es importante. Por diversas razones, a los administradores de red les gusta agrupar a los usuarios en redes LAN para reflejar la estructura de la organización más que el diseño físico del edificio. Un aspecto es la seguridad. Una LAN podría hospedar a los servidores web y otras computadoras destinadas para uso público. Otra LAN podría hospedar a las computadoras que contengan los registros del departamento de recursos humanos que no deben salir de ese departamento. En este caso, se justifica que todas las computadoras se asignen a una sola LAN y que no se permita acceder a los servidores fuera de esa LAN.

Un segundo aspecto es la carga. Algunas redes LAN se utilizan mucho más que otras, y en ocasiones podría ser conveniente 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 que usaban en las videoconferencias para ayudarles.

Un tercer aspecto es el tráfico de difusión. Los puentes difunden el tráfico cuando no conocen la ubicación de destino, y los protocolos de capas superiores también usan la difusión. Por ejemplo, cuando un usuario desea enviar un paquete a una dirección IP x, ¿cómo sabe qué dirección MAC poner en la trama? En la próxima unidad 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. A medida que aumenta el número de computadoras en una LAN, también aumenta el número de difusiones. Cada difusión consume más capacidad de la LAN que una trama regular, ya que se entrega a todas las computadoras en la LAN. Al evitar que las redes LAN crezcan más de lo necesario, se reduce el impacto del tráfico de difusión.

Las difusiones tienen el problema asociado de cuando una interfaz de red se avería o desconfigura y empieza a generar flujos interminables de tramas de difusión. Si la red es realmente mala, algunas de estas tramas provocarán respuestas que a su vez generarán más tráfico. El resultado de esta **tormenta de difusión** es que (1) las tramas de difusión ocupan toda la capacidad de la LAN, y (2) las máquinas de todas las redes LAN interconectadas se atascan con sólo procesar y desechar todas las tramas difundidas.

A primera vista parecería que podemos limitar la magnitud de las tormentas de difusión si sepáramos las redes LAN mediante puentes o switches, pero si el objetivo es conseguir transparencia (es decir, que una máquina se pueda cambiar a una LAN distinta al otro lado del puente sin que nadie lo note), entonces los puentes tienen que reenviar las tramas difundidas.

Ya que analizamos por qué las empresas podrían requerir varias redes LAN con alcances limitados, regresemos al problema de desacoplar la topología lógica de la física. Para construir una topología física que refleje la estructura organizacional tal vez se requiera más trabajo y aumente el costo, incluso con un cableado centralizado y switches. Por ejemplo, si dos personas en el mismo departamento trabajan en distintos edificios, puede ser más fácil conectarlos a distintos switches que pertenezcan a redes LAN diferentes. Aun si éste no es el caso, un usuario podría transferirse de un departamento a otro de la misma empresa sin cambiar de oficina,

o podría cambiar de oficina pero no de departamento. Así, el usuario podría estar en la LAN incorrecta hasta que un administrador cambiara el conector del usuario de un switch a otro. Además, tal vez el número de computadoras que pertenecen a distintos departamentos no sea adecuado para el número de puertos en los switchs; algunos departamentos podrían ser demasiado pequeños y otros tan grandes que requieran varios switchs. Como resultado, se desperdician los puertos que no se utilizan de los switchs.

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 switch correcto, o los puertos disponibles del switch están en la LAN incorrecta.

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). El comité IEEE 802 lo estandarizó y ahora se ha implementado ampliamente en muchas organizaciones. Ahora vamos a analizarlo de forma breve.

Las redes VLAN se basan en switchs especialmente diseñados para este propósito. Para configurar una red VLAN, el administrador de la red decide cuántas VLAN habrá, qué computadoras habrá en cuál VLAN y cómo se llamarán las VLAN. A menudo se les asignan nombres mediante colores (de manera informal), ya que de esta manera es posible imprimir diagramas a 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 la LAN con puente de la Ilustración 82, en la cual nueve de las máquinas pertenecen a la VLAN G (gris) y cinco forman parte de la VLAN W (blanca). Las máquinas de la VLAN gris están distribuidas a través de dos switchs, incluyendo dos máquinas que se conectan a un switch mediante un hub.

Para que las VLAN funcionen correctamente, es necesario establecer tablas de configuración en los puentes. Estas tablas indican cuáles VLAN se pueden acceder a través de qué puertos. Cuando una trama llega procedente de, digamos, la VLAN gris, se debe reenviar a todos los puertos identificados con una G. Esto es válido para el tráfico ordinario (es decir, de unidifusión) en el que los puentes no conocen la ubicación del destino, así como para el tráfico de multidifusión y de difusión. Cabe mencionar que podemos etiquetar un puerto con varios colores de VLAN.

Como ejemplo, suponga que una de las estaciones grises conectadas al puente B1 envía una trama a un destino que no se conoce de antemano. El puente B1 recibirá la trama y verá que proviene de una máquina en la VLAN gris, por lo que inundará esa trama en todos los puertos etiquetados como G (excepto el puerto entrante). La trama se enviará a las otras cinco estaciones grises conectadas a B1, así como a través del enlace de B1 al puente B2. En el puente B2, la trama se reenvía de manera similar a todos los puertos etiquetados como G. Esto envía la trama a una estación más y al hub (que transmitirá la trama a todas sus estaciones). El hub tiene ambas etiquetas debido a que se conecta a las máquinas de ambas redes VLAN. La trama no se envía en otros puertos que no tengan G en la etiqueta, puesto que el puente sabe que no hay máquinas en la VLAN gris a las que se pueda llegar por medio de estos puertos.

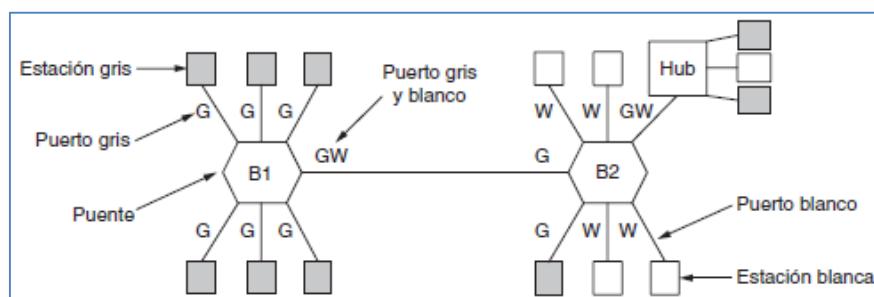


Ilustración 82 - Dos redes VLAN, gris y blanca, en una LAN con puente.

En nuestro ejemplo, la trama sólo se envía del puente *B1* al puente *B2* ya que hay máquinas en la VLAN gris que están conectadas a *B2*. Si analizamos la VLAN blanca, podemos ver que el puerto del puente *B2* que se conecta al puente *B1* no está etiquetado como W. Esto significa que una trama en la VLAN blanca no se reenviará del puente *B2* al puente *B1*. Este comportamiento es correcto, ya que no hay estaciones en la VLAN blanca que estén conectadas a *B1*.

El estándar IEEE 802.1Q

Para implementar este esquema, los puentes necesitan saber a qué VLAN pertenece una trama entrante. Sin esta información, por ejemplo, cuando el puente *B2* recibe una trama del puente *B1*, no puede saber si reenviar la trama a la VLAN gris o blanca. Si estuviéramos diseñando un nuevo tipo de LAN, sería muy fácil sólo agregar un campo VLAN en el encabezado. Pero, ¿qué podemos 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 IEEE 802.1Q, emitido en 1998. El nuevo formato contiene una etiqueta VLAN, que examinaremos en breve. No es de sorprender que cambiar algo tan bien establecido como el encabezado de Ethernet no sea nada sencillo. Algunas de las preguntas que nos vienen a la mente son:

1. ¿Tenemos que tirar a la basura los 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?

La clave para la solución consiste en comprender que los campos VLAN sólo los utilizan los puentes y los conmutadores, no las máquinas de los usuarios. Así, en la Ilustración 82 no es 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. Además, para utilizar VLAN, los puentes deben tener soporte para VLAN. Este hecho ayuda a que el diseño sea viable.

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, las nuevas tarjetas Ethernet son compatibles con el 802.1Q y pueden llenar bien los campos VLAN.

Puesto que puede haber computadoras (y switchs) que no tengan soporte para VLAN, el primer puente con soporte para VLAN en tocar una trama agrega campos VLAN y el último en el camino los elimina. En la Ilustración 83 se muestra un ejemplo de una topología mixta. En esta figura, las computadoras con soporte para VLAN generan tramas etiquetadas (802.1Q) directamente, y los switchs posteriores utilizan estas etiquetas. Los símbolos sombreados tienen soporte para VLAN y los vacíos no.

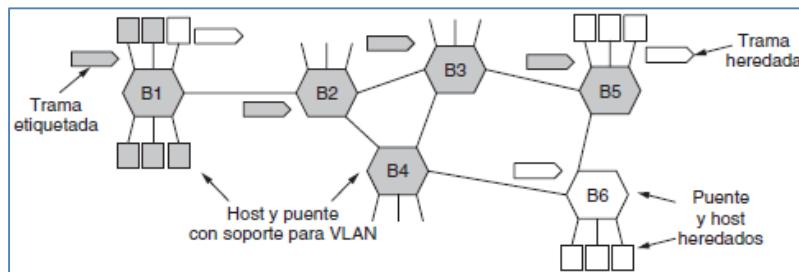


Ilustración 83 - LAN con puentes que sólo cuenta con soporte parcial para VLAN. Los símbolos sombreados tienen soporte para VLAN. Los vacíos no.

En el 802.1Q, se asignan colores a las tramas dependiendo del puerto por el que se reciban. Para que este método funcione, todas las máquinas en un puerto deben pertenecer a la misma VLAN, lo cual reduce la flexibilidad. Por ejemplo, en la Ilustración 82 esta propiedad es válida para todos los puertos en donde se conecte una computadora individual a un puente, pero no para el puerto en donde el hub se conecta al puente *B2*.

Además, el puente puede usar el protocolo de la capa superior para seleccionar el color. De esta forma, las tramas que llegan a un puerto se podrían colocar en distintas redes VLAN, dependiendo de si transmiten paquetes IP o tramas PPP.

Hay otros métodos posibles, pero no están soportados por el estándar 802.1Q. Como ejemplo, se puede usar la dirección MAC para seleccionar el color de VLAN. Esto podría ser útil para tramas que provienen de una LAN 802.11 cercana, en donde las computadoras portátiles envían tramas a través de distintos puertos a medida que se desplazan. En este caso, una dirección MAC se asignaría a una VLAN fija, sin importar por qué puerto entró a la LAN.

En cuanto al problema de las tramas mayores a 1518 bytes, el 802.1Q tan sólo incrementó el límite a 1522 bytes. Por suerte, sólo las computadoras y switchs con soporte para VLAN deben soportar estas tramas más largas.

Ahora veamos el formato de trama del 802.1Q, que se muestra en la Ilustración 84. El único cambio es la adición de un par de campos de 2 bytes. El primero es *ID del protocolo de VLAN*. Siempre tiene el valor 0x8100. Como este número es mayor de 1500, todas las tarjetas Ethernet lo interpretan como un tipo y no como una longitud. Lo que una tarjeta heredada hace con una trama como ésta es discutible, ya que dichas tramas no deberían enviarse a tarjetas heredadas.

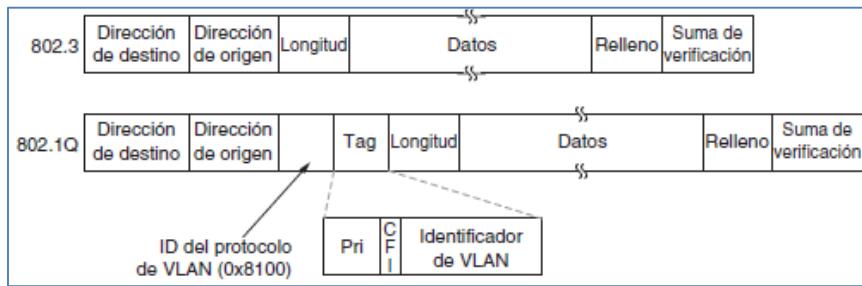


Ilustración 84 - Los formatos de trama Ethernet 802.3 (heredada) y 802.1Q.

El segundo campo de 2 bytes contiene tres subcampos. El principal es *Identificador de VLAN*, que ocupa los 12 bits de menor orden. Éste es el punto central de la cuestión: el color de la VLAN a la que pertenece la trama. El campo *Prioridad* de 3 bits no tiene absolutamente nada que ver con las VLAN, pero como cambiar el 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 insensible al tiempo, con el propósito de ofrecer una mejor calidad de servicio sobre Ethernet. Esto es necesario para el transporte de voz sobre Ethernet.

El último campo, *CFI (Indicador del Formato Canónico)*. Su propósito original era indicar el orden de los bits en las direcciones MAC (*Little endian* en comparación con *big endian*), pero su uso se perdió en otras controversias. En la actualidad, su presencia indica que la carga útil contiene una trama 802.5 congelada-seca que espera encontrar otra LAN 802.5 en el destino, cuando se transmite a través de Ethernet. Por supuesto, este arreglo no tiene absolutamente nada que ver con las VLAN. 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 switch con soporte para VLAN, éste utiliza el identificador de la VLAN como índice en una tabla para averiguar a cuáles puertos enviar la trama. Pero, ¿de dónde proviene la tabla? Si se construye en forma manual, tenemos que empezar desde cero: la configuración manual de los puentes. La ventaja de los puentes transparentes es que son plugandplay y no requieren configuración manual. Sería una gran pena perder esa propiedad. Por fortuna, los puentes con soporte para VLAN también se pueden autoconfigurar con sólo observar las etiquetas entrantes. Si una trama etiquetada como VLAN 4 llega por el 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 haciendo referencia a porciones apropiadas del estándar 802.1D.

Para utilizar las VLAN de manera apropiada, cada trama lleva un identificador especial nuevo que se utiliza como índice en una tabla dentro del switch para averiguar el destino al que se debe enviar la trama. Esto es lo que se hace en las redes orientadas a conexión. En las redes sin conexión, la dirección de destino es la que se utiliza para el enrutamiento, no un tipo de identificador de conexión. En el capítulo 5 veremos más sobre este conexionismo gradual.

Referencias sección uno

- *Comunicaciones y Redes de Computadores 7ma Edición*, William Stallings, Pearson Educación, 2004.
- *Redes de Computadoras 5ta Edición*, Andrew S. Tanenbaum, Pearson Educación, 2012

Análisis de equipos

En la siguiente sección haremos un análisis de las características y funciones de los equipos de red que operan con protocolos de capa física y capa de enlace de datos, como así también veremos algunas configuraciones básicas de los mismos.

Capa física

Concentrador

Un concentrador de Ethernet, concentrador activo, concentrador de red, hub repetidor, repetidor multipuerto, o simplemente **hub** es un hardware de red dispositivo para conectar múltiples dispositivos Ethernet juntos y haciendo que actúen como un único segmento de red. Tiene múltiples puertos de entrada/salida (E/S), en los que una señal introducida en la entrada de cualquier puerto aparece en la salida de cada puerto excepto el entrante original.

Un concentrador funciona en la capa física (capa 1) del modelo OSI. Un concentrador repetidor también participa en la detección de colisiones, reenviando una señal de atasco a todos los puertos si detecta una colisión. Además de los puertos estándar 8P8C ("**RJ45**"), algunos concentradores también pueden venir con un BNC o un conector de interfaz de unidad de conexión (**AUI**) para permitir la conexión a segmentos de red 10BASE2 o 10BASE5 heredados.

Los concentradores ahora son en gran parte obsoletos, ya que han sido reemplazados por commutadores de red, excepto en instalaciones muy antiguas o aplicaciones especializadas. A partir de 2011, la conexión de segmentos de red por repetidores o concentradores está en desuso por IEEE 802.3.

Información técnica

Función de capa física

Un concentrador de red es un dispositivo poco sofisticado en comparación con un commutador. Como repetidor multipuerto, funciona repitiendo las transmisiones recibidas de uno de sus puertos a todos los demás puertos. Es consciente de los paquetes de capa física, es decir, puede detectar su inicio (preámbulo), una línea inactiva (espacio entre paquetes) y detectar una colisión que también se propaga enviando una señal de atasco. Un concentrador no puede seguir examinando o gestionando el tráfico que lo atraviesa. Un concentrador no tiene memoria para almacenar datos y solo puede manejar una transmisión a la vez. Por lo tanto, los concentradores solo pueden ejecutarse en modo half dúplex. Debido a que los concentradores forman parte de un **dominio de colisión**, las colisiones de paquetes son más probables en redes conectadas usando concentradores que en redes conectadas usando dispositivos más sofisticados (los cuales limitan y dividen los dominios de colisión).

Conectando múltiples concentradores

La necesidad de que los hosts puedan detectar colisiones limita la cantidad de concentradores y el tamaño total de una red construida utilizando concentradores (una red construida mediante commutadores no tiene estas limitaciones). Para redes de 10Mbit/s construidas utilizando concentradores repetidores, se debe seguir la regla 5-4-3: se permiten hasta cinco segmentos (cuatro concentradores) entre dos estaciones finales. Para redes 10BASE-T, se permiten hasta cinco segmentos y cuatro repetidores entre dos hosts. Para redes de 100Mbit/s, el límite se reduce a 3 segmentos (2 concentradores) entre dos estaciones finales, e incluso eso solo se permite si los concentradores son de Clase II. Algunos concentradores tienen puertos de pila específicos del fabricante que les permiten combinarse de una manera que permite más concentradores que el simple encadenamiento a través de cables Ethernet, pero, aun así, es probable que una gran red FastEthernet requiera commutadores para evitar los límites de encadenamiento de los concentradores.

Funciones adicionales

La mayoría de los concentradores detectan problemas típicos, como colisiones excesivas y *jabbering* en puertos individuales, y separan el puerto, desconectándolo del medio compartido. Por lo tanto, el Ethernet de par trenzado basado en concentrador es generalmente más robusto que el Ethernet basado en cable coaxial

(por ejemplo, 10BASE2), donde un dispositivo que se comporta mal puede afectar negativamente a todo el dominio de colisión. Incluso si no se separa automáticamente, un concentrador simplifica la resolución de problemas porque eliminan la necesidad de solucionar fallas en un cable largo con múltiples derivaciones; las luces de estado en el concentrador pueden indicar la posible fuente del problema o, como último recurso, los dispositivos se pueden desconectar de un concentrador uno a la vez con mucha más facilidad que desde un cable coaxial.

Para pasar los datos a través del repetidor de manera utilizable de un segmento al siguiente, el marco y la velocidad de datos deben ser los mismos en cada segmento. Esto significa que un repetidor no puede conectar un segmento 802.3 (Ethernet) y un segmento 802.5 (Token Ring) o un segmento de 10Mbit/s a 100Mbit/s Ethernet.

Clases FastEthernet

Los concentradores y repetidores de 100Mbit/s vienen en dos grados de velocidad diferentes: la *clase I* retrasa la señal por un máximo de 140 bits (permitiendo la traducción/grabación entre 100BASE-TX, 100BASE-FX y 100BASE-T4) y los concentradores de *clase II* retrasan la señal para un máximo de 92 bits (permitiendo la instalación de dos concentradores en un solo dominio de colisión).

Hub de doble velocidad

En los primeros días de FastEthernet, los conmutadores Ethernet eran dispositivos relativamente caros. Los concentradores sufrieron el problema de que si había algún dispositivo 10BASE-T conectado, toda la red debía funcionar a 10Mbit/s. Por lo tanto, se desarrolló un compromiso entre un concentrador y un conmutador, conocido como hub de doble velocidad. Estos dispositivos hacen uso de un conmutador interno de dos puertos, puenteando los segmentos de 10Mbit/s y 100Mbit/s. Cuando un dispositivo de red se activa en cualquiera de los puertos físicos, el dispositivo lo conecta al segmento de 10Mbit/s o al segmento de 100Mbit/s, según corresponda. Esto evitó la necesidad de una migración de “*todo o nada*” a las redes FastEthernet. Estos dispositivos se consideran concentradores porque el tráfico entre dispositivos conectados a la misma velocidad no se conmuta.

Concentrador Gigabit Ethernet

Se han definido concentradores repetidores para GigabitEthernet, pero los productos comerciales no han aparecido debido a la transición de la industria a la conmutación.

Usos

Históricamente, la razón principal para comprar concentradores en lugar de conmutadores era su precio. Este motivador ha sido eliminado en gran medida por las reducciones en el precio de los conmutadores, pero los concentradores aún pueden ser útiles en circunstancias especiales:

- Para insertar un analizador de protocolos en una conexión de red, un concentrador es una alternativa a una conexión de red o duplicación de puertos.
- Se puede usar un concentrador con puertos 10BASE-T y un puerto 10BASE2 para conectar un segmento 10BASE2 a una red moderna Ethernet sobre par trenzado.
- Se puede usar un concentrador con puertos 10BASE-T y un puerto AUI para conectar un segmento 10BASE5 a una red moderna.
- Como los concentradores tienen menor latencia y *jitter* en comparación con los conmutadores, son más adecuados para redes en tiempo real.

Configuración de un concentrador

Los concentradores no poseen parámetros configurables, ni por software ni hardware, su funcionalidad es automática. Basta conectar los cables a distintos puertos y encender el dispositivo y este funcionará de forma adecuada.

Repetidor

Es un dispositivo electrónico que recibe una señal débil o de bajo nivel y la retransmite a una potencia o nivel más alto, de tal modo que se puedan cubrir distancias más largas sin degradación o con una degradación tolerable.

Conceptos

1. Un dispositivo analógico que amplifica una señal de entrada, independientemente de su naturaleza analógica o digital.
2. Un dispositivo digital que amplifica, conforma, retemporiza o lleva a cabo una combinación de cualquiera de estas funciones sobre una señal digital de entrada para su retransmisión.

Utilización

Los repetidores se utilizan a menudo en los cables transcontinentales y transoceánicos ya que la atenuación perdida de señal en tales distancias sería completamente inaceptable sin ellos.

Los repetidores se utilizan tanto en cables de cobre como en cables de fibra óptica. Los repetidores se utilizan también en los servicios de radiocomunicación. Un subgrupo de estos son los repetidores usados por los radioaficionados.

Asimismo, se utilizan repetidores en los enlaces de telecomunicación punto a punto mediante radioenlaces que funcionan en el rango de las microondas, como los utilizados para distribuir las señales de televisión entre los centros de producción y los distintos emisores o los utilizados en redes de telecomunicación para la transmisión de telefonía.

En comunicaciones ópticas el término repetidor se utiliza para describir un elemento del equipo que recibe una señal óptica, la convierte en eléctrica, la regenera y la retransmite de nuevo como señal óptica. Dado que estos dispositivos convierten la señal óptica en eléctrica y nuevamente en óptica, estos dispositivos se conocen a menudo como repetidores electroópticos.

Características

Los repetidores son equipos que trabajan a nivel 1 de la pila OSI, es decir, repiten todas las señales de un segmento a otro a nivel de señal física.

Estos equipos sólo aislan entre los segmentos los problemas asociados a características físicas que pudieran existir en algunos de ellos.

El número máximo de repetidores en cascada es de cuatro, pero con la condición de que los segmentos 2 y 4 sean **IRL**, es decir, que no tengan ningún equipo conectado que no sean los repetidores. En caso contrario, el número máximo es de 2, interconectando 3 segmentos de red.

El repetidor tiene dos puertas que conectan dos segmentos Ethernet por medio de *transceivers* (instalando diferentes transceivers es posible interconectar dos segmentos de diferentes medios físicos) y cables *drop*.



Ilustración 85 - Ejemplo de repetidores

Par trenzado – Tipos de cableado

Existen varios tipos de estándares de cables de red, como por ejemplo, el cable coaxial, el cable de par trenzado, el cable USB, el cable cruzado, el cable directo, el cable de fibra óptica, etc. Entre estos, cable directo y cable cruzado son los tipos de cable más desconocidos. Curiosamente, ambos son dos tipos de cable Ethernet con las mismas características físicas. ¿Cuál es la diferencia entre estos dos, entonces?

T-568A vs. T-568B

Existen dos formas de conectividad diferentes, dependiendo de estos dos tipos de disposición de cableado de red. La disposición de T-568B es sin duda la más común, aunque muchos dispositivos también son compatibles con la distribución T-568A. Si los dos extremos del cable directo están cableados conforme a un estándar, entonces estamos hablando de una **conexión directa**, siendo posible aplicar cualquiera de las disposiciones. Por el contrario, hablaríamos entonces de una **conexión cruzada**. Algunas aplicaciones de red requieren un cable cruzado Ethernet, con un conector T-568A en un extremo y uno T-568B en el otro. Este tipo de cable se usa generalmente para conexiones directas de ordenador a ordenador. En la siguiente sección hablaremos más detalladamente sobre las características del cable directo y el cable cruzado.

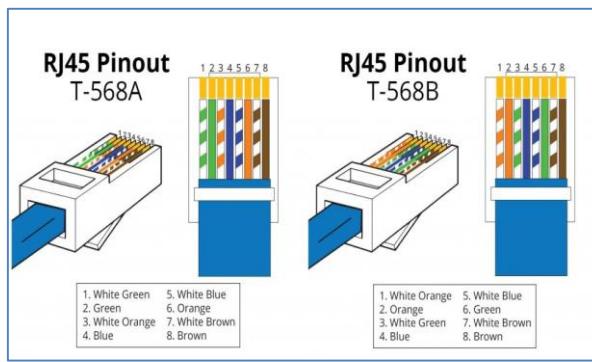


Ilustración 86 - Estándares T-568 A y B

¿Qué es el cable de red directo?

El cable de red directo no cambia su dirección. Ambos extremos utilizan el mismo estándar de cableado: T-568A o T-568B. Por lo tanto, ambos extremos (conector A y conector B) del cable directo tienen una disposición de cables del mismo color (como se muestra en la siguiente imagen). Así, el Pin 1 en el conector A se dirige al Pin 1 en el conector B, el Pin 2 al Pin 2, etc. Estos cables son ampliamente utilizados para conectar ordenadores a switchs, concentradores o enrutadores.

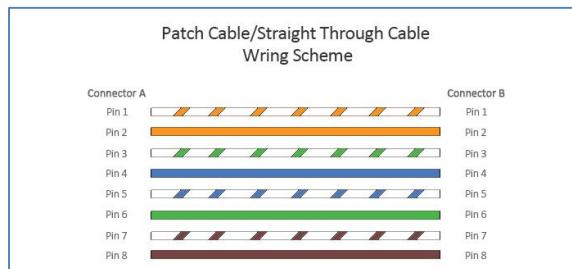


Ilustración 87 - Cable directo

¿Qué es el cable cruzado?

El cable cruzado, como su nombre indica, se cruza o cambia de dirección de un extremo a otro. A diferencia del cable directo, el cable cruzado utiliza diferentes estándares de cableado en cada uno de sus extremos: uno el estándar T568A y el otro el estándar T568B. Ambos lados (conector A y conector B) del cable cruzado tendrán una disposición de cables de diferente color; los cables que salen del conector A deben coincidir con sus pins correspondientes en el conector B, tal y como se muestra en el siguiente ejemplo. Los cables cruzados se usan principalmente para conectar dos enrutadores, ordenadores o concentradores(hub).

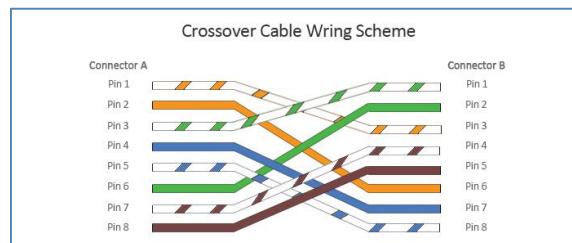


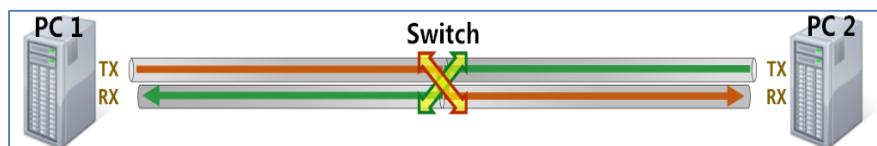
Ilustración 88 - Cable cruzado

Cable directo vs. cable cruzado: ¿cuándo usarlos?

En general, un cable cruzado se utiliza para conectar dos dispositivos del mismo tipo, como por ejemplo un PC a una PC o un switch a otro switch. Por otro lado, el cable directo conecta dos dispositivos diferentes entre sí, como por ejemplo un PC y un switch. A continuación, explicaremos sus diferentes aplicaciones a través de diferentes escenarios.

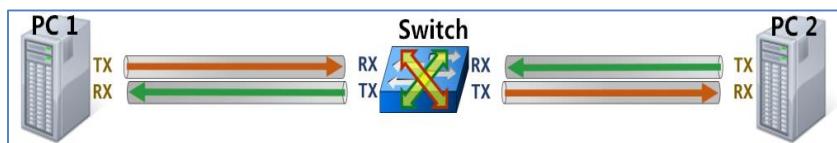
- **Escenario 1: PC a PC**

Si tuviésemos dos computadoras conectadas directamente entre sí intentando transmitir datos a través del TX, sus señales chocarían y no se conseguiría ninguna transmisión en el RX. Por lo tanto, ninguna de ellas recibiría una señal. Es por este motivo que necesitaríamos el cable cruzado para este tipo de conexiones entre dos PCs. Dado que este tipo de cable se cruza, la señal enviada en el cable TX desde la PC 1 puede ser recibida en el cable RX de la PC 2; de ahí que los cables cruzados sean la mejor opción para conectar dos dispositivos iguales.



- Escenario 2: De PC a PC a través de switch

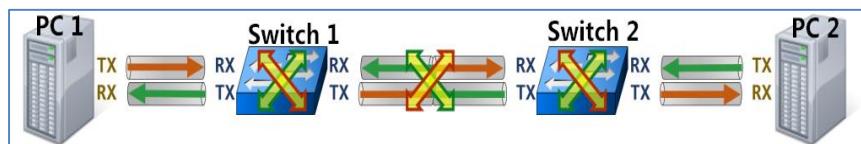
¿Qué pasa cuando tenemos un switch en el medio de la conexión entre ordenadores? Los switches están diseñados para asistir en la comunicación entre dos computadoras a través del cruce de transmisiones interno. La PC 1 envía sus datos a través del cable de TX y el switch los recibe en su cable RX; este los transmite entonces a través de su cable de TX, y la PC 2 finalmente los recibe en su cable RX. El mismo proceso sucedería de igual forma en la dirección opuesta. Por lo tanto, en este caso, podríamos entonces utilizar un cable recto para nuestra conexión ordenador a ordenador.



- Escenario 3: de PC a switch y de switch a PC

¿Qué pasa entonces si tenemos dos switches? Los dos switches cruzarían el cable por separado, originando así la transmisión cruzada entre los switches. Como se mencionó anteriormente, dos dispositivos iguales necesitarían un cable cruzado para realizar la conexión. En el diagrama de arriba podemos ver que:

- (1) Cuando la PC 1 se conecta al Switch 1, necesitamos un cable directo.
- (2) Cuando el Switch 1 se conecta al Switch 2, necesitamos un cable cruzado.
- (3) Cuando el Switch 2 se conecta a la PC 2, necesitamos un cable directo.



Capa de enlace de datos

NIC

La tarjeta de red, también conocida como placa de red, adaptador de red, adaptador LAN, Interfaz de red física o sus términos en inglés *Network Interface Card* o *Network interface controller* (NIC), cuya traducción literal del inglés es «tarjeta de interfaz de red» (TIR), es un componente de hardware que conecta una computadora a una red informática y que posibilita compartir recursos entre dos o más host, es decir, en una red de computadoras.

Implementación

Las primeras tarjetas de interfaz de red se implementaban comúnmente en tarjetas de expansión que se conectaban en un bus de la computadora. El bajo costo y la ubicuidad del estándar Ethernet hizo posible que la mayoría de las computadoras modernas tengan una interfaz de red integrada en la placa base. Las placas base de servidor más nuevas pueden incluso tener interfaces de red duales incorporadas.

Las capacidades de Ethernet están ahora integradas en el chipset de la placa base o implementadas a través de un chip Ethernet dedicado de bajo costo, conectado a través del bus PCI (o PCI Express), así que no se requiere una tarjeta de red por separada a menos que se necesiten interfaces adicionales o se utilice otro tipo de red.

La NIC pueden utilizar una o más de las siguientes técnicas para indicar la disponibilidad de paquetes a transferir:

- **Polling**, donde la CPU examina el estado del periférico bajo el control del programa.
- **IRQ-E/S controlada**, donde el periférico alerta a la CPU de que está listo para transferir datos.

Además, los NIC pueden utilizar una o más de las siguientes técnicas para transferir datos de paquetes:

- **Entrada/salida programada**, donde la CPU mueve los datos hacia o desde la NIC a la memoria.
- **DMA**, donde algún otro dispositivo que no sea la CPU asume el control del bus de sistema para mover datos hacia o desde la NIC a la memoria. Esto elimina la carga de la CPU, pero requiere más lógica en la tarjeta.

Además, un búfer de paquetes en la NIC puede no ser necesario y puede reducir la latencia. Existen dos tipos de DMA:

- **DMA de terceros**, en el que un controlador DMA distinto del NIC realiza transferencias y,
- **Bus mastering**, donde el propio NIC realiza transferencias.

Una tarjeta de red Ethernet normalmente tiene un socket 8P8C donde está conectado el cable de red. Las NICs más antiguas también proporcionaban conexiones BNC, o AUI. Algunos LEDs informan al usuario si la red está activa y si se produce o no transmisión de datos. Las tarjetas de red Ethernet suelen soportar Ethernet de 10 Mbit/s, 100 Mbits/s y 1000 Mbits/s. Tales tarjetas son designadas como "10/100/1000", lo que significa que pueden soportar una tasa de transferencia máxima nocial de 10, 100 o 1000 Mbit/s. También están disponibles NIC de 10 Gbits/s.

Propósito

La NIC implementa los circuitos electrónicos necesarios para comunicarse sobre una red de computadoras, ya sea utilizando de cables como Token Ring, Ethernet, fibra, o sin cables como Wi-Fi, es por tanto un dispositivo de capa física y uno de capa de enlace de datos ya que proporciona acceso físico a un medio de red y, para IEEE 802 y redes similares, proporciona un sistema de direccionamiento de bajo nivel mediante el uso de la dirección **MAC** que se asignan exclusivamente a las tarjetas de red.

Esto proporciona una base para una pila de protocolos de red completa, permitiendo la comunicación entre pequeños grupos de computadoras en la misma red de área local (LAN) y comunicaciones de red a gran escala a través de protocolos enrutables, como *Internet Protocol* (IP).

Aunque existen otras tecnologías de red, las redes IEEE 802, incluidas las variantes Ethernet, han alcanzado casi la ubicuidad desde mediados de los años noventa.

Dirección MAC

Cada tarjeta de red tiene un número de identificación único de 48 bits en hexadecimal que asignan los fabricantes legales de Hardware llamado dirección MAC (*Media Access Control*; control de acceso al medio) también conocido como dirección física que es independiente al protocolo de red que se utilice. Estas direcciones únicas de hardware son administradas por el “Instituto de Ingeniería Eléctrica y Electrónica” (**IEEE**, *Institute of Electronic and Electrical Engineers*). Los tres primeros octetos (24 bits) del número MAC, identifican al proveedor específico y es conocido como número **OUI** (*Organizationally unique identifier*, identificador único de organización), designado por IEEE, que combinado con otro número de 24 bits forman la dirección MAC completa.

La numeración de una dirección MAC tiene el siguiente formato: «B7-T0-B6-D3-E9-99».

Tipos de tarjetas de red

Existen diversos tipos de tarjetas, placas o adaptadores de red, en función del tipo de cableado o arquitectura de red:

- Token Ring
- ARCNET
- Ethernet
- Wi-Fi

Token Ring

Las tarjetas para red Token Ring están prácticamente en desuso, debido a la baja velocidad y elevado costo respecto de Ethernet. Tenían conector DB-9. También se utilizó el conector RJ-45 para las NIC y las **MAU** (*Multiple Access Unit*, unidad de múltiple acceso), que era el núcleo de una red Token Ring.

Arcnet

Las tarjetas para red ARCNET utilizaban principalmente conector BNC y/o puertos RJ-45.

Ethernet

Las tarjetas de red para Ethernet utilizan conectores:

- **RJ-45** (*Registered jack*): 10/100/1000,
- **BNC** (*Bayonet Neill-Concelman*): 10,
- **AUI** (*Attachment Unit Interface*): 10,
- **MII** (*Media Independent Interface*): 100,
- **GMII** (*Gigabit Media Independent Interface*): 1000

El caso más habitual es el de la tarjeta con el conector RJ-45, aunque durante la transición del uso mayoritario de cable coaxial (10 Mbit/s) al cable de par trenzado (100 Mbit/s) abundaron las tarjetas con conectores BNC y RJ-45, e incluso BNC / AUI / RJ-45 (en muchas de ellas se pueden ver serigrafiados los conectores no usados).

Con la entrada de las redes Gigabit y el que en las casas sea frecuente la presencia de varias computadoras comienzan a verse tarjetas y placas base (con NIC integradas) con 2 y hasta 4 puertos RJ-45, que antes estaba reservado a los servidores.

Pueden variar en función de la velocidad de transmisión, normalmente 10 Mbit/s ó 10/100 Mbit/s. también se utilizan las de 100 Mbit/s, conocida como Gigabit Ethernet y en algunos casos 10 Gigabit Ethernet, utilizando también cable de par trenzado, de categorías: 6, 6a y Cat 7, que funcionan a frecuencias más altas.

Las velocidades especificadas por los fabricantes son teóricas, por ejemplo, las de 100 Mbit/s realmente pueden llegar como máximo a 78,4 Mbit/s.

Wi-Fi

También son NIC las tarjetas inalámbricas (*wireless*), que vienen en diferentes variedades dependiendo de la norma a la cual se ajusten, usualmente son 802.11b, 802.11g y 802.11n. Las más populares son la 802.11b que transmite a 11 Mbit/s con una distancia teórica de 100 metros y la 802.11g que transmite a 54 Mbit/s.

Puente

Puente de red (*bridge*) es el dispositivo de interconexión de redes de computadoras que opera en la capa 2. Interconecta segmentos de red (o divide una red en segmentos) haciendo la transferencia de datos de una red hacia otra con base en la dirección física de destino de cada paquete.

El término bridge, formalmente, responde a un dispositivo que se comporta de acuerdo al estándar IEEE 802.1D.

Un bridge conecta segmentos de red formando una sola subred (permite conexión entre equipos sin necesidad de *routers*). Funciona a través de una tabla de direcciones MAC detectadas en cada segmento al que está conectado. Cuando detecta que un nodo de uno de los segmentos está intentando transmitir datos a un nodo del otro, el bridge copia la trama para el otro segmento de red, teniendo la capacidad de desechar la trama (filtrado) en caso de no tener dicho segmento de red como destino. Para conocer por dónde enviar cada trama que le llega (encaminamiento) incluye un mecanismo de aprendizaje automático (auto aprendizaje) por lo que no necesitan configuración manual.

Clasificación de Puentes de red

Se pueden clasificar los puentes de red, atendiendo dos aspectos: según el tipo de interfaz y según la localización geográfica de las redes de área local (LAN) que se van a interconectar.

Según interfaz

- **Puentes homogéneos.**
Interconecta LAN con el mismo protocolo MAC (el nivel físico puede diferir), es decir, no hay conversión de protocolos a nivel 2, simplemente almacenamiento y reenvío de tramas. Un ejemplo de dispositivo homogéneo es un Switch Ethernet.
- **Puentes heterogéneos.**
El puente dispone de una entidad superior encargada de la transformación de cabeceras entre distintos tipos de interfaces. Recibe tramas por una interfaz (por ejemplo: Wi-Fi) para enviarlas por otra de otro tipo (por ejemplo: Ethernet). Un ejemplo de dispositivo, con las interfaces de ejemplo anteriores, es un punto de acceso en una red wi-fi.

Según localización geográfica

- **Puentes locales.**
Sirven para enlazar directamente dos redes físicamente cercanas.
- **Puentes remotos o de área extensa.**
Se conectan en parejas enlazando dos o más redes locales y formando una red de área extensa (WAN) a través de líneas telefónicas.

Autoaprendizaje

Los puentes de red usan una tabla de reenvío para enviar tramas a lo largo de los segmentos de la red. Si una dirección de destino no se encuentra en la tabla, la trama es enviada por medio de *flooding* por todos los puertos (de todas las interfaces de red) del bridge excepto por el puerto por el que llegó. Por medio de este envío “masivo” de tramas el dispositivo de destino (desde el segmento de red en el que se encuentre) recibirá el paquete y responderá con otra trama (una trama de respuesta en este caso), quedando así registrada la dirección física MAC destino en una entrada de la tabla, a la que se acompañará el número de la interface por la que se llega a ese dispositivo (que podrá ser un host, un servidor, etc.). Dicha tabla incluye tres campos:

dirección MAC del dispositivo, interfaz del puente por la que se llega al dispositivo y la hora a la que llegó la trama (a partir de este campo y la hora actual se puede saber si la entrada está vigente en el tiempo). El bridge utilizará esta tabla para determinar qué hacer con las tramas que le llegan.

En el caso de un bridge de dos puertos, la tabla de reenvío puede considerarse como un filtro: el bridge lee en la trama la dirección del destinatario y decide si reenviarlo por el puerto por el que no ha llegado o filtrarlo (desechando dicha trama). Es decir, si el bridge determina que el nodo de destino está ubicado en el otro segmento de la red (por el que no ha llegado), lo retransmite. En caso de detectar que la trama lleva como destino un nodo del mismo segmento de red por el que ha llegado, la trama se descarta.

El término de autoaprendizaje se utiliza también para dispositivos con más de dos puertos. Como ejemplo, considerando tres equipos (A, B y C) conectados a los puertos (1, 2 y 3, respectivamente) de un bridge; inicialmente la tabla está vacía y ocurre lo siguiente: el equipo "A" envía una trama al "B", esta trama llega al bridge por el interface 1, a continuación el bridge examina la dirección de origen y al no existir ninguna entrada, la crea para "A", consignando la dirección física MAC de "A", el número de interface 1 y el tiempo. A continuación, comprueba la dirección de destino y la busca en la tabla. Como no existe se envía dicha trama por los puertos 2 y 3. Una vez la trama sea recibida por "B", este responde a dicha trama y esta respuesta llega al puente a través de la interface 2; entonces se crea una nueva entrada para "B" en la tabla, consignando la dirección física MAC de "B", el número de interface 2 y el tiempo. "C" también recibe el envío, pero al no ser el destinatario, simplemente desecha el paquete. A partir de este momento es posible enviar paquetes entre "A" y "B" utilizando sólo el ancho de banda necesario, (sin inundar los segmentos de red a través de todos los puertos de las interfaces del bridge, menos el puerto por el que llega la trama). En el caso de "C" se repetirá el mismo proceso anterior cuando sea conveniente, quedando guardada la información en la tabla de idéntica manera a lo expuesto.

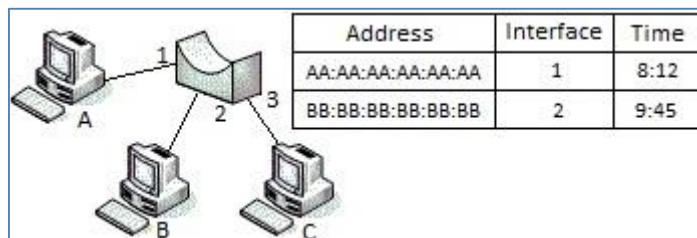


Ilustración 89 - Comunicación mediante bridge

Bridges frente a switch

La diferencia más importante entre un bridge y un switch es que los bridges normalmente tienen un número pequeño de interfaces (de dos a cuatro), mientras que los switches pueden llegar a tener docenas; por tanto, este último necesita un diseño de prestaciones elevadas.

Bridges frente a hubs

La principal diferencia entre un bridge y un hub es que el segundo repite todas las tramas con cualquier destino para el resto de los nodos conectados; en cambio el primero sólo reenvía las tramas pertenecientes a cada segmento. De esta forma se aislan dominios de colisión mejorando el rendimiento de las redes interconectadas: se disminuye el tráfico inútil, permite un mayor caudal de transmisión, proporciona mayor cobertura geográfica y permite dar servicio a más dispositivos.

Bridges frente a routers

Tanto un bridge como un router son dispositivos que se utilizan para encaminar datos, pero lo hacen de diferente manera. Los bridges operan en la capa 2 (nivel de enlace de datos), mientras que los routers lo hacen en la capa 3 (nivel de red). Es decir, el bridge toma sus decisiones sobre la base de la dirección MAC y el router lo hará a partir de una dirección IP. Esto se traduce en que los bridges no son capaces de discernir entre subredes, mientras que los routers sí lo son. Cuando se diseña una red se puede optar por múltiples opciones, como juntar varios segmentos mediante un bridge o dividirla en subredes e interconectarla mediante routers. Para este último caso, si un equipo conectado a una subred se mueve físicamente a otra subred, ha de

cambiarse la IP para tener conexión. Sin embargo, si un equipo se mueve dentro de una red conectada mediante bridges no haría falta reconfigurar nada.



Ilustración 90 – Ejemplos de puentes o bridges

Comutador

Comutador (*switch*) es el dispositivo digital lógico de interconexión de equipos que opera en la capa de enlace de datos. Su función es interconectar dos o más host de manera similar a los puentes de red, pasando datos de un segmento a otro de acuerdo con la dirección MAC de destino de las tramas en la red y eliminando la conexión una vez finalizada ésta.

Los comutadores se utilizan cuando se desea conectar múltiples tramos de una red, fusionándolos en una sola red. Al igual que los puentes, dado que funcionan como un filtro en la red y solo retransmiten la información hacia los tramos en los que hay el destinatario de la trama de red, mejoran el rendimiento y la seguridad de las redes de área local (LAN).

Conexiones en un comutador Ethernet

Los comutadores poseen la capacidad de aprender y almacenar las direcciones de red de la capa 2 (direcciones MAC) de los dispositivos alcanzables a través de cada uno de sus puertos. Por ejemplo, un equipo conectado directamente a un puerto de un comutador provoca que el comutador almacene su dirección MAC. Esto permite que, a diferencia de los concentradores, la información dirigida a un dispositivo vaya desde el puerto origen al puerto de destino.

En el caso de conectar dos comutadores o un comutador y un concentrador, cada comutador aprenderá las direcciones MAC de los dispositivos accesibles por sus puertos, por lo tanto, en el puerto de interconexión se almacenan las MAC de los dispositivos del otro comutador.

Bucles de red e inundaciones de tráfico

Uno de los puntos críticos de estos equipos son los bucles, que consisten en habilitar dos caminos diferentes para llegar de un equipo a otro a través de un conjunto de comutadores. Los bucles se producen porque los comutadores que detectan que un dispositivo es accesible a través de dos puertos emiten la trama por ambos. Al llegar esta trama al comutador siguiente, este vuelve a enviar la trama por los puertos que permiten alcanzar el equipo. Este proceso provoca que cada trama se multiplique de forma exponencial, llegando a producir las denominadas inundaciones de la red, provocando en consecuencia el fallo o caída de las comunicaciones.

Clasificación

Atendiendo al método de direccionamiento de las tramas utilizadas

- *Store-and-Forward*

Los comutadores Store-and-Forward guardan cada trama en un búfer antes del intercambio de información hacia el puerto de salida. Mientras la trama está en el búfer, el switch calcula el CRC y mide el tamaño de la misma. Si el CRC falla, o el tamaño es muy pequeño o muy grande (una trama Ethernet tiene entre 64 bytes y 1518 bytes) la trama es descartada. Si todo se encuentra en orden es encaminada hacia el puerto de salida.

Este método asegura operaciones sin error y aumenta la confianza de la red. Pero el tiempo utilizado para guardar y chequear cada trama añade un tiempo de demora importante al procesamiento de las mismas. La demora o *delay* total es proporcional al tamaño de las tramas: cuanto mayor es la trama, más tiempo toma este proceso.

- **Cut-Through**

Los conmutadores cut-through fueron diseñados para reducir esta latencia. Esos switches minimizan el retardo leyendo sólo los 6 primeros bytes de datos de la trama, que contiene la dirección de destino MAC, e inmediatamente la encaminan.

El problema de este tipo de switch es que no detecta tramas corruptas causadas por colisiones, ni errores de CRC. Cuanto mayor sea el número de colisiones en la red, mayor será el ancho de banda que consume al encaminar tramas corruptas.

Existe un segundo tipo de switch cut-through, los denominados *fragment free*, fue proyectado para eliminar este problema. El switch siempre lee los primeros 64 bytes de cada trama, asegurando que tenga por lo menos el tamaño mínimo, y evitando el encaminamiento de colisiones por la red.

- **Adaptive Cut-Through**

Son los conmutadores que procesan tramas en el modo adaptativo y son compatibles tanto con store-and-forward como con cut-through. Cualquiera de los modos puede ser activado por el administrador de la red, o el switch puede ser lo bastante inteligente como para escoger entre los dos métodos, basado en el número de tramas con error que pasan por los puertos.

Cuando el número de tramas corruptas alcanza un cierto nivel, el conmutador puede cambiar del modo cut-through a store-and-forward, volviendo al modo anterior cuando la red se normalice.

Los conmutadores cut-through son más utilizados en pequeños grupos de trabajo y pequeños departamentos. En esas aplicaciones es necesario un buen volumen de trabajo, ya que los errores potenciales de red quedan en el nivel del segmento, sin impactar la red corporativa.

Los conmutadores store-and-forward son utilizados en redes corporativas, donde es necesario un control de errores.

Atendiendo a la forma de segmentación de las subredes

- **Comutadores de capa 2**

Son los conmutadores tradicionales, que funcionan como puentes multi-puertos. Su principal finalidad es dividir una LAN en múltiples dominios de colisión, o en los casos de las redes en anillo, segmentar la LAN en diversos anillos. Basan su decisión de envío en la dirección MAC destino que contiene cada trama.

Los conmutadores de la capa 2 posibilitan múltiples transmisiones simultáneas sin interferir en otras sub-redes. Los switches de capa 2 no consiguen, sin embargo, filtrar difusiones o broadcasts, multicasts (en el caso en que más de una sub-red contenga las estaciones pertenecientes al grupo multicast de destino), ni tramas cuyo destino aún no haya sido incluido en la tabla de direccionamiento.

- **Comutadores de capa 3**

Son los conmutadores que, además de las funciones tradicionales de la capa 2, incorporan algunas funciones de enrutamiento, como por ejemplo la determinación del camino basado en informaciones de capa de red, validación de la integridad del cableado de la capa 3 por *checksum* y soporte a los protocolos de ruteo tradicionales (RIP, OSPF, etc.).

Los conmutadores de capa 3 soportan también la definición de redes virtuales (VLAN), y según modelos posibilitan la comunicación entre las diversas VLAN sin la necesidad de utilizar un router externo.

Por permitir la unión de segmentos de diferentes dominios de difusión o broadcast, los switches de capa 3 son particularmente recomendados para la segmentación de redes LAN muy grandes, donde la simple utilización de switches de capa 2 provocaría una pérdida de rendimiento y eficiencia de la ADSL, debido a la cantidad excesiva de broadcasts.

Dentro de los comutadores de la capa 3 tenemos:

- Paquete por paquete

Un comutador paquete por paquete (*packet by packet*) es un caso especial de un comutador Store-and-Forward pues, al igual que este, almacena y examina el paquete, calculando el CRC y decodificando la cabecera de la capa de red para definir su ruta a través del protocolo de enrutamiento adoptado.

- Cut-through

Un comutador de la capa 3 Cut-Through (no confundir con un comutador Cut-Through), examina los primeros campos, determina la dirección de destino (a través de la información de los *headers* o cabeceras de capa 2 y 3) y, a partir de ese instante, establece una conexión punto a punto (a nivel 2) para conseguir una alta tasa de transferencia de paquetes.



Ilustración 91 – Ejemplos de comutadores o switchs

Capas superiores

Router

Un rúter, enrutador (del inglés router) o encaminador, es un dispositivo que permite interconectar computadoras que funcionan en el marco de una red. Su función: se encarga de establecer la ruta que destinará a cada paquete de datos dentro de una red informática.

Funcionamiento

El funcionamiento básico de un enrutador o encaminador, como se deduce de su nombre, consiste en enviar los paquetes de red por el camino o ruta más adecuada en cada momento. Para ello almacena los paquetes recibidos y procesa la información de origen y destino que poseen. Con arreglo a esta información reenvía los paquetes a otro encaminador o bien al receptor, en una actividad que se denomina “*encaminamiento*” o “*enrutamiento*”. Cada encaminador se encarga de decidir el siguiente salto en función de su tabla de reenvío o tabla de encaminamiento, la cual se genera mediante protocolos que deciden cuál es el camino más adecuado o corto.

Por ser los elementos que forman la capa de red, tienen que encargarse de cumplir las dos tareas principales asignadas a la misma:

- **Reenvío de paquetes:** cuando un paquete llega al enlace de entrada de un encaminador, este tiene que pasar el paquete al enlace de salida apropiado. Una característica importante de los encaminadores es que no difunden tráfico difusivo.
- **Encaminamiento de paquetes:** mediante el uso de algoritmos de encaminamiento tiene que ser capaz de determinar la ruta que deben seguir los paquetes a medida que fluyen de un emisor a un receptor.

Por tanto, debemos distinguir entre reenvío y encaminamiento. Reenvío consiste en coger un paquete en la entrada y enviarlo por la salida que indica la tabla, mientras que por encaminamiento se entiende el proceso de hacer esa tabla.

Arquitectura física

En un enrutador se pueden identificar cuatro componentes:

- **Puertos de entrada:** realiza las funciones de la capa física consistentes en la terminación de un enlace físico de entrada a un encaminador; realiza las funciones de la capa de enlace de datos necesarias para interoperar con las funciones de la capa de enlace de datos en el lado remoto del enlace de entrada; realiza también una función de búsqueda y reenvío de modo que un paquete reenviado dentro del entramado de conmutación del encaminador emerge en el puerto de salida apropiado.
- **Entrada de conmutación:** conecta los puertos de entrada del enrutador a sus puertos de salida.
- **Puertos de salida:** almacena los paquetes que le han sido reenviados a través del puerto de conmutación y los transmite al enlace de salida. Realiza entonces la función inversa de la capa física y de la capa de enlace que el puerto de entrada.
- **Procesador de encaminamiento:** ejecuta los protocolos de encaminamiento, mantiene la información de encaminamiento y las tablas de reenvío y realiza funciones de gestión de red dentro del enrutador.

Tipos de enrutamiento

Tanto los enrutadores como los anfitriones guardan una tabla de enrutamiento. El núcleo del sistema lee la tabla de enrutamiento antes de reenviar paquetes a la red local. La tabla de enrutamiento enumera las direcciones IP de las redes que conoce el sistema, incluida la red local predeterminada del sistema. La tabla también enumera la dirección IP de un sistema de puerta de enlace para cada red conocida. La puerta de enlace es un sistema que puede recibir paquetes de salida y reenviarlos un salto más allá de la red local.

Enrutamiento estático

Hosts y redes de tamaño reducido que obtienen las rutas de un enrutador predeterminado, y enrutadores predeterminados que sólo necesitan conocer uno o dos enrutadores.

Determinación de enrutamiento

La información de enrutamiento que el encaminador aprende desde sus fuentes de enrutamiento se coloca en su propia tabla de enrutamiento. El encaminador se vale de esta tabla para determinar los puertos de salida que debe utilizar para retransmitir un paquete hasta su destino. La tabla de enrutamiento es la fuente principal de información del enrutador acerca de las redes. Si la red de destino está conectada directamente, el enrutador ya sabrá el puerto que debe usar para reenviar los paquetes. Si las redes de destino no están conectadas directamente, el encaminador debe aprender y calcular la ruta más óptima a usar para reenviar paquetes a dichas redes. La tabla de enrutamiento se constituye mediante uno de estos dos métodos o ambos:

- **Manualmente**, por el administrador de la red.
- A través de procesos dinámicos que se ejecutan en la red (ejemplo: DHCP).

Rutas estáticas

Las rutas estáticas se definen administrativamente y establecen rutas específicas que han de seguir los paquetes para pasar de un puerto de origen hasta un puerto de destino. Se establece un control preciso de enrutamiento según los parámetros del administrador.

Las rutas estáticas por defecto especifican una puerta de enlace de último recurso, a la que el enrutador debe enviar un paquete destinado a una red que no aparece en su tabla de enrutamiento, es decir, se desconoce.

Las rutas estáticas se utilizan habitualmente en enrutamientos desde una red hasta una red de conexión única, ya que no existe más que una ruta de entrada y salida en una red de conexión única, evitando de este modo la sobrecarga de tráfico que genera un protocolo de enrutamiento. La ruta estática se configura para conseguir conectividad con un enlace de datos que no esté directamente conectado al enrutador. Para conectividad de extremo a extremo, es necesario configurar la ruta en ambas direcciones. Las rutas estáticas permiten la construcción manual de la tabla de enrutamiento.

Enrutamiento dinámico

El enrutamiento dinámico le permite a los encaminadores ajustar, en tiempo real, los caminos utilizados para transmitir paquetes IP. Cada protocolo posee sus propios métodos para definir rutas (camino más corto, utilizar rutas publicadas por pares, etc.).

Gateway

La pasarela (en inglés *gateway*) o puerta de enlace es el dispositivo que actúa de interfaz de conexión entre aparatos o dispositivos, y también posibilita compartir recursos entre dos o más ordenadores.

Su propósito es traducir la información del protocolo utilizado en una red inicial, al protocolo usado en la red de destino.

La pasarela es normalmente un equipo informático configurado para dotar a las máquinas de una red de área local (Local Area Network, LAN) conectadas a él de un acceso hacia una red exterior, generalmente realizando para ello operaciones de traducción de direcciones de red (*Network Address Translation, NAT*). Esta capacidad de traducción de direcciones permite aplicar una técnica llamada enmascaramiento de IP , usada muy a menudo para dar acceso a Internet a los equipos de una LAN compartiendo una única conexión a Internet, y por tanto, una única dirección IP externa.

Un equipo que haga de puerta de enlace en una red debe tener necesariamente dos tarjetas de red (Network Interface Card, NIC).

La puerta de enlace predeterminada (default gateway) es la ruta predeterminada o ruta por defecto que se le asigna a un equipo y tiene como función enviar cualquier paquete del que no conozca por cuál interfaz enviarlo y no esté definido en las rutas del equipo, enviando el paquete por la ruta predeterminada.

En entornos domésticos, se usan los routers como puertas de enlace para conectar la red local doméstica con Internet; aunque esta puerta de enlace no conecta dos redes con protocolos diferentes, sí que hace posible conectar dos redes independientes haciendo uso de NAT.

Qué son el dominio de Colisión y el Dominio de Difusión

Si un dispositivo envía una señal en una red Ethernet, los dispositivos conectados al mismo medio físico recibirán el mensaje. Si la información no va para ellos, la descartarán y no habrá problemas. Sin embargo, el hecho de que el mensaje se envíe a todos los dispositivos nos hace plantear lo siguiente: cuantos más dispositivos haya, más señales se enviarán por la red y la eficiencia de esta se verá reducida. De este problema surgen los conceptos siguientes.

Dominio de colisión

Se conoce como dominio de colisión al espacio físico con un ancho de banda compartido por un conjunto de dispositivos. En el caso que dos de esos dispositivos quieran transmitir al mismo tiempo, existe la posibilidad de que sus mensajes colisionen el espacio compartido y, o bien acaben convertidos en una amalgama de bits o bien no se pueda asegurar que al receptor le ha llegado el mensaje, de ahí el término colisión. Para resolverlo, Ethernet implementa CSMA/CD (*Carrier sense multiple access with collision detection*).

La buena práctica en redes Ethernet consiste en mantener los dominios de colisión pequeños, es decir, con pocos dispositivos. Cuantos menos nodos comparten un espacio físico menor será la probabilidad de que transmitan a la vez y surja una colisión.

Cuanta más colisiones, peor rendimiento de la red. Este se expresa tal que así:

$$\text{Rendimiento} = \left(1 - \left(\frac{\text{Colisiones Totales}}{\text{Paquetes}} \right) \right) * 100$$

Dominio de difusión (Broadcast)

El dominio de difusión es la división lógica de la red dentro de la cual los dispositivos envían mensajes de difusión, también llamados broadcast. Dos dispositivos dentro del dominio de difusión comparten puerta de enlace (*gateway*), dirección de subred y pueden transmitir a otro dispositivo dentro del dominio sin precisar encaminamiento; es decir, se encuentran en la misma LAN.

Los dominios broadcast o de difusión están delimitados por routers.

Cómo actúan los dispositivos de red dividiendo los dominios de colisión y difusión

Como veremos a continuación en los ejemplos, los dispositivos de red actúan de la siguiente forma:

- Los hubs o concentradores extienden el dominio de colisión ya que reenvían todos los mensajes que reciben de un dispositivo a los otros dispositivos conectados. Todos los elementos conectados al mismo hub (aunque estén en diferentes puertos) tienen el mismo dominio de colisión y dominio de difusión.
- Los switches o comutadores segmentan los dominios de colisión, pero expanden el dominio de difusión. Es decir, el alcance de un mensaje broadcast no es limitado por un switch (No obstante, sí limitan su alcance en el caso de las VLAN).
- Los routers o encaminadores también segmentan los dominios de colisión, pero además también lo hacen con los de difusión (un mensaje broadcast es limitado por el router y no sale de la LAN en la que se encuentra).

Ejemplo

En esta topología de red podemos observar que solamente hay un router (en la derecha, señalizado con R1). La computadora 5 se encuentra en un solo dominio de difusión, al igual que la computadora 6; el resto (1,2 y 3) se encuentran en otro, por lo que en esta imagen hay 3 dominios de difusión.

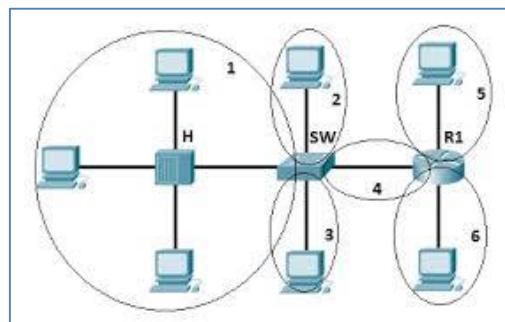


Ilustración 92 - Dominios de colisión y difusión.

Los dominios de colisión están señalizados con un círculo en la imagen. Tal y como se mencionó anteriormente, el hub extiende el dominio de colisión, por eso el Dominio de Colisión 1 tiene 3 dispositivos conectados.

Control de acceso al medio

En los estándares IEEE 802 LAN/MAN, la subcapa de **control de acceso al medio** (MAC, del inglés *Media Access Control*) es la capa que controla el hardware responsable de la interacción con el medio de transmisión por cable, óptico o inalámbrico. La subcapa MAC y la subcapa de **control de enlace lógico** (LLC) juntas forman la capa de enlace de datos. Dentro de la capa de enlace de datos, la LLC proporciona control de flujo y multiplexación para el enlace lógico, mientras que el MAC proporciona control de flujo y multiplexación para el medio de transmisión.

Estas dos subcapas juntas corresponden a la capa 2. Por razones de compatibilidad, LLC es opcional para implementaciones de IEEE 802.3 (las tramas son entonces "en bruto"), pero obligatorio para implementaciones de otros estándares de capa física IEEE 802. Dentro de la jerarquía de capas y los estándares IEEE 802, la subcapa MAC proporciona una abstracción de control de la capa física de modo que las complejidades del control del enlace físico son invisibles para la LLC y las capas superiores de la pila de red. Por lo tanto, cualquier subcapa LLC (y capas superiores) se puede usar con cualquier MAC. A su vez, el bloque de control de acceso medio está conectado formalmente al PHY a través de una interfaz independiente del medio. Aunque el bloque MAC está integrado en la actualidad con el PHY dentro del mismo paquete de dispositivos, históricamente cualquier MAC podría usarse con cualquier PHY, independientemente del medio de transmisión.

Al enviar datos a otro dispositivo en la red, la subcapa MAC encapsula tramas de nivel superior en tramas apropiadas para el medio de transmisión (es decir, el MAC agrega un preámbulo de palabra de sincronización y también relleno si es necesario), agrega una secuencia de verificación de trama para identificar errores de transmisión, y luego reenvía los datos a la capa física tan pronto como lo permita el método de acceso al canal apropiado. Para las topologías con un dominio de colisión (bus, anillo, malla, topologías de punto a multipunto), es necesario controlar cuándo se envían los datos y cuándo esperar para evitar colisiones. Además, la MAC también es responsable de compensar las colisiones al iniciar la retransmisión si una señal de atasco es detectada. Al recibir datos de la capa física, el bloque MAC garantiza la integridad de los datos verificando las secuencias de verificación de trama del remitente y elimina el preámbulo y el relleno del remitente antes de pasar los datos a las capas superiores.

Funciones realizadas en la subcapa MAC

De acuerdo con el estándar IEEE 802-2001, sección 6.2.3 "Subcapa MAC", las funciones principales realizadas por la capa MAC son:

- Delimitación y reconocimiento de tramas.
- Direccionamiento de estaciones de destino (como estaciones individuales y como grupos de estaciones).
- Transporte de información de direccionamiento de la estación fuente.
- Transferencia de datos transparente de PDU LLC, o de información equivalente en la subcapa Ethernet.
- Protección contra errores, generalmente mediante la generación y verificación de secuencias de verificación de tramas.
- Control de acceso al medio de transmisión física.

En el caso de Ethernet, de acuerdo con 802.3-2002 sección 4.1.4, las funciones requeridas de un MAC son:

- Recibir/transmitir tramas normales.
- Retransmisión semidúplex y funciones de retroceso.
- Agregar/verificar FCS (secuencia de verificación de trama).
- Descartar tramas malformadas.

Mecanismo de direccionamiento

Las direcciones de red locales utilizadas en redes IEEE 802 se denominan **direcciones de control de acceso a medios** (direcciones MAC o *MAC Address*); se basan en el esquema de direccionamiento que se utilizó en las

primeras implementaciones de Ethernet. Una dirección MAC está pensada como un número de serie único. Las direcciones MAC generalmente se asignan al hardware de la interfaz de red en el momento de la fabricación. La parte más importante de la dirección identifica al fabricante, quien asigna el resto de la dirección, por lo tanto, proporciona una dirección potencialmente única. Esto hace posible que las tramas se entreguen en un enlace de red que interconecta hosts mediante alguna combinación de repetidores, concentradores, puentes y conmutadores, pero no por enrutadores de capa de red. Así, por ejemplo, cuando un paquete IP llega a su (sub)red de destino, la dirección IP de destino (un concepto de capa 3 o capa de red) se resuelve con el Protocolo de resolución de direcciones para IPv4, o mediante el Protocolo de descubrimiento de vecinos (IPv6) en la dirección MAC (un concepto de capa 2) del host de destino.

Ejemplos de redes físicas son las redes Ethernet y las redes Wi-Fi, las cuales son redes IEEE 802 y utilizan direcciones MAC IEEE 802 de 48 bits.

No se requiere una capa MAC en la comunicación punto a punto full dúplex, pero los campos de dirección se incluyen en algunos protocolos punto a punto por razones de compatibilidad.

Mecanismo de control de acceso al canal

Los mecanismos de control de acceso al canal proporcionados por la capa MAC también se conocen como protocolo de acceso múltiple. Esto hace posible que varias estaciones conectadas al mismo medio físico lo compartan. Ejemplos de medios físicos compartidos son redes de bus, redes de anillo, redes de concentrador, redes inalámbricas y enlaces punto a punto half dúplex. El protocolo de acceso múltiple puede detectar o evitar colisiones de paquetes de datos si se utiliza un método de acceso de canal basado en contención de modo de paquete, o reservar recursos para establecer un canal lógico si un circuito conmutado o se utiliza el método de acceso al canal basado en la canalización. El mecanismo de control de acceso al canal se basa en un esquema de multiplexado de capa física.

El protocolo de acceso múltiple más extendido es el protocolo CSMA/CD basado en contención utilizado en las redes Ethernet. Este mecanismo solo se utiliza dentro de un dominio de colisión de red, por ejemplo, una red de bus Ethernet o una red de topología en estrella basada en concentrador. Una red Ethernet puede dividirse en varios dominios de colisión, interconectados por puentes y conmutadores.

No se requiere un protocolo de acceso múltiple en una red full-dúplex conmutada, como las redes Ethernet conmutadas de la actualidad, pero a menudo está disponible en el equipo por razones de compatibilidad.

Análisis de protocolos

Capa de enlace de datos

Spanning Tree Protocol

En una red LAN la redundancia se logra teniendo varios enlaces físicos entre los hosts, de forma que queden varios caminos para llegar a un mismo destino. El resultado de esto es que la red LAN queda con ciclos o bucles.

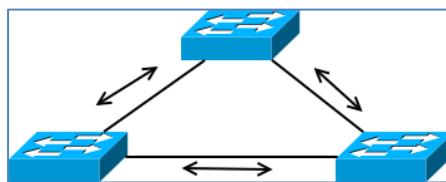


Ilustración 93 - Red LAN con redundancia.

Si bien la red anterior es redundante los ciclos son altamente perjudiciales para la misma dado que producen una serie de problemas que acabarán por dejarla inutilizada. Dentro de dichos problemas podemos encontrarnos con:

- **Tormentas de broadcast:** los broadcasts en la red son reenviados una y otra vez y permanecen circulando en la misma sin fin, dado que en Ethernet no existe como en IP un campo de temporización (TTL). Lógicamente, al no eliminarse la situación se agrava con cada nuevo broadcast.
- **Múltiples copias de una trama:** con la redundancia es muy probable que un host reciba una trama repetida, dado que la misma podría llegar por dos enlaces diferentes.
- **Tabla CAM inconsistente:** una trama que proviene de una MAC en particular podría llegar desde enlaces diferentes.
- **Bucles recursivos:** un bucle puede generar un nuevo bucle y estos crecer de forma exponencial. En una situación así la red quedará inusuable en pocos segundos.

Ante la necesidad de tener una red LAN redundante y dinámica libre de los problemas asociados a la redundancia resulta evidente que es necesario un protocolo que sea capaz de resolver estas cuestiones. Es aquí donde entra en acción el **Protocolo de Spanning Tree (STP)**.

Terminología básica

Para comprender el funcionamiento del STP es necesario conocer alguna terminología indispensable asociada al mismo.

- **Bridge ID:** es el identificador de cada bridge. Es el resultado de combinar la prioridad del bridge con su dirección MAC base.
- **Root bridge** (puente raíz): es el punto focal de la red y el que se toma como referencia para las decisiones del STP. El RB será aquel switch que tenga el menor bridge ID.
- **BPDU (Bridge Protocol Data Unit):** son pequeñas unidades de datos que transportan información de control del STP. Se las utiliza en primera instancia para escoger el RB y luego para detectar posibles fallos en la red.
- **Bridges no raíz:** son todos los demás bridges de la topología. Participan en el intercambio de BPDUs y actualizan a su vez su base de datos del STP.
- **Costo de un puerto:** se determina en base al ancho de banda del enlace y será el valor que se utilice para decidir el camino más corto al RB.
- **Costo del camino al RB:** el costo de un camino al RB es la suma de los costos de cada enlace por el que pasa. El camino elegido por el STP al RB será aquel cuyo costo sea más bajo.
- **Puerto raíz (designado):** es el puerto de cada bridge que se encuentra en el camino mínimo al RB. Sólo hay uno por bridge que siempre estará en estado de *forwarding*.
- **Puerto no designado:** todo puerto en un bridge con mayor costo que el puerto designado. Será puesto en estado de bloqueo.

Estado de los puertos

Cada puerto que participa del STP puede estar en uno de cinco estados. Estos son:

- **Bloqueado (BLK)**: no reenvía tramas de datos, aunque sí recibe y envía BPDUs. Es el estado por defecto de los puertos cuando un switch se enciende y su función es la de prevenir ciclos.
- **Escuchando (LST)**: recibe, analiza y envía BPDUs para asegurarse que no existen bucles.
- **Aprendiendo (LRN)**: al igual que el estado LST, recibe, analiza y envía BPDUs, aunque aquí también comienza a armar la tabla CAM. En este estado aún no se reenvían tramas de datos.
- **Reenviando (FWD)**: envía y recibe todas las tramas de datos. Los puertos designados al final del estado de LRN serán marcados como FWD.
- **Deshabilitado**: es un puerto deshabilitado administrativamente y que no participará en el STP. Para el STP un puerto en este estado es como si no existiera.

Operación del STP

El protocolo de STP cumple con una serie de pasos antes de alcanzar el estado estable y comenzar a enviar tramas de datos. Los mismos son los que se listan a continuación.

1. Escoger el RB:
 - a. Se elige el bridge con prioridad más baja.
 - b. Si uno o más switches tienen la prioridad más baja se elige entre ellos el que posea la MAC base más baja.
2. Se eligen los puertos raíz: cada bridge encuentra el menor camino hasta el RB y, con él, su puerto designado.
3. Cada uno de los bridges escucha BPDUs en todos sus puertos y, si detecta algún bucle en un puerto, lo bloquea. De lo contrario lo pone en estado FWD. El criterio para decidir qué puerto bloquear en un switch es el siguiente:
 - a. Si debe escogerse un puerto entre dos switches diferentes se elige para bloquear el de aquel switch con el mayor bridge ID.
 - b. Si debe escogerse un puerto dentro del mismo switch entonces se escoge aquel que tenga el mayor costo. En caso de coincidir el costo, el puerto que se bloquea es aquel que tenga el identificador más alto.

A continuación, se lista el costo de cada tipo de enlace para el STP:

Velocidad	Costo
4 Mbps	250
10 Mbps	100
16 Mbps	62
100 Mbps	19
1 Gbps	4
2 Gbps	3
10 Gbps	2

Ilustración 94 - Costo de enlace según su velocidad.

Topología de ejemplo

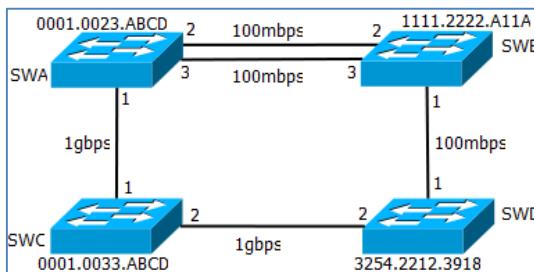


Ilustración 95 - Topología de ejemplo

Se asume que todos los bridges tienen en la topología anterior la misma prioridad. Entonces, sabemos que el primer paso es escoger el RB. Para ello, al tener toda la misma prioridad vamos a escoger el que tenga menor dirección MAC, que resulta ser el SWA.

El próximo paso es encontrar el camino mínimo desde cada switch hasta el RB. Empezando por el SWB vemos que tiene tres puertos que nos permiten alcanzar el RB. Por el puerto 1 el costo total es de 27, resultado de sumar 19 del enlace de 100Mbps y 4 de cada enlace de 1Gb. Luego, tanto por el puerto 2 como por el puerto 3 el costo total es de 19. En este caso, se escoge el puerto con identificador más pequeño, que es el 2. De esta manera, el puerto 2 será el puerto raíz y pasará a estado de FWD y el 3 quedará bloqueado. Dejaremos el análisis del puerto 1 para un poco más adelante.

Para el caso del SWC, el puerto raíz es el 1, ya que el costo por allí es de 4. El análisis del puerto 2 para más adelante.

Finalmente, el SWD tiene dos posibles caminos al RB. Uno de ellos con costo 8 a través del puerto 2 y el otro con costo 38 por el puerto 1. Lógicamente será puerto raíz el puerto 2.

El último paso es decidir, de los puertos que quedaron sin determinar, cuál pasará a estado BLK para romper el bucle. Veamos las posibilidades:

- Puerto 1 de SWB.
- Puerto 2 de SWC.
- Puerto 1 de SWD.

De lo anterior, el puerto 2 de SWC no podrá ser dado que está en el camino mínimo del SWD al RB. Así que ese pasará a estado FWD. La decisión se reduce entonces entre el puerto 1 del SWB y el 1 del SWD. Ante esta situación se resolvía bloqueando el puerto del bridge con mayor MAC, lo que implica que el puerto 1 del SWD quedará en BLK y el puerto 1 del SWB en FWD.

Rapid Spanning Tree

El protocolo **RSTP** es un estándar que incorpora muchas características que aceleran el proceso de convergencia inicial y ante un fallo. Es totalmente compatible con STP y de hecho un bridge ejecutando STP y otro RSTP pueden convivir perfectamente en la misma red, aunque utilizando el protocolo STP. Por ello, para que RSTP funcione, todos los switches deben soportarlo.

Etherchannel

Cuando existen dos enlaces entre dos switches una alternativa es utilizar *Etherchannel*. Esta tecnología lo que hace es combinar dos o más puertos como si fueran uno solo, agregando el ancho de banda a ese único canal lógico.

Un etherchannel provee también redundancia, dado que si uno de los links se cae sigue funcionando perfectamente, con un ancho de banda reducido. La ventaja entonces en comparación con STP es que etherchannel hace uso activo de todos los enlaces, en contraste con STP que de un conjunto utilizaría sólo uno.

CSMA/CS

CSMA/CD (del inglés *Carrier Sense Multiple Access with Collision Detection*) o, en español, acceso múltiple por detección de portado y detección de colisiones, es un algoritmo de acceso al medio compartido. Su uso está especialmente extendido en redes Ethernet donde es empleado para mejorar sus prestaciones. En CSMA/CD, los dispositivos de red escuchan el medio antes de transmitir, es decir, es necesario determinar si el canal y sus recursos se encuentran disponibles para realizar una transmisión. Además, mejora el rendimiento de CSMA finalizando el envío cuando se ha detectado una colisión.

Procedimiento

El siguiente procedimiento se usa para iniciar una transmisión. El procedimiento se completa cuando la trama se transmite con éxito o se detecta una colisión durante la transmisión.

1. ¿Hay una trama lista para transmitir? Si no, esperar por una trama.
2. ¿Está el medio inactivo? Si no, esperar hasta que esté listo.
3. Comenzar a transmitir y monitorear colisiones durante la transmisión.
4. ¿Ocurrió una colisión? De ser así, ir al procedimiento de colisión detectada.
5. Restablecer los contadores de retransmisión y completar la transmisión de la trama.

El siguiente procedimiento se usa para resolver una colisión detectada. El procedimiento se completa cuando se inicia la retransmisión o se cancela la retransmisión debido a numerosas colisiones.

1. Continuar la transmisión (con una señal de atasco en lugar de un encabezado de trama/datos/CRC) hasta que se alcance el tiempo mínimo para garantizar que todos los receptores detecten la colisión.
2. Incrementar el contador de retransmisión
3. ¿Se alcanzó el número máximo de intentos de transmisión? Si es así, abortar la transmisión.
4. Calcular y esperar el período de espera aleatorio según el número de colisiones.
5. Volver a ingresar al procedimiento principal en el paso 1.

Los métodos para la detección de colisiones dependen de los medios. En un bus eléctrico compartido, como 10BASE5 o 10BASE2, las colisiones se pueden detectar comparando los datos transmitidos con los datos recibidos o reconociendo una amplitud de señal superior a la normal en el bus. En todos los demás medios, una portadora detectada en el canal de recepción mientras se transmite desencadena un evento de colisión. Los repetidores o concentradores detectan colisiones por sí mismos y propagan señales de atasco.

El procedimiento de recuperación de colisión se puede comparar con lo que sucede en una cena, donde todos los invitados hablan entre sí a través de un medio común (el aire). Antes de hablar, cada invitado educadamente espera que termine el orador actual. Si dos invitados comienzan a hablar al mismo tiempo, ambos se detienen y esperan períodos de tiempo cortos y aleatorios (en Ethernet, este tiempo se mide en microsegundos). La esperanza es que al elegir un período de tiempo aleatorio, ambos invitados no elijan el mismo momento para tratar de hablar nuevamente, evitando así otra colisión.

Trama de CSMA/CD

La trama empleada en CSMA/CD está formada por ocho campos:

- El **preámbulo**, formado por 7 octetos, es el encargado de que el receptor pueda sincronizarse con el emisor, de forma que pueda localizarse el principio de la trama.
- **Delimitador de inicio**: es un byte empleado para indicar al receptor el inicio de la trama.
- **Dirección de destino**: contiene la dirección física (MAC) del equipo destinatario de la trama.
- **Dirección de origen**: contiene la dirección MAC de la estación emisora de la trama y tiene un formato similar al de la dirección de destino.
- **Longitud**: indica la longitud del campo de datos que se encuentra a continuación. Es necesaria para determinar la longitud del campo de datos en los casos que se utiliza un campo de relleno.
- **Información**: contiene los datos transmitidos. Es de longitud variable, por lo que puede tener cualquier longitud entre 42 y 1500 bytes.

- **Relleno:** es usado para que la trama alcance la longitud mínima requerida. Una trama debe contener un mínimo número de bytes para que las estaciones puedan detectar las colisiones con precisión.
- **Chequeo:** contiene un código de redundancia cíclica de 32 bits. Es utilizada como mecanismo de control de errores en la transmisión.

Tipos de CSMA/CD

El algoritmo CSMA/CD puede estar basado en cualquiera de los siguientes procedimientos:

- **CSMA 1-persistente:** cuando una estación quiere transmitir, primero escucha el canal. Si éste está libre entonces transmite inmediatamente. En el caso contrario permanece a la escucha hasta que esté libre. En el momento en el que la estación considere que el canal está disponible, se transmite inmediatamente. El problema radica en que varias estaciones pueden estar esperando a que el canal esté libre para transmitir, dando lugar a una colisión de sus tramas.
- **CSMA no persistente:** funciona de forma análoga al anterior excepto en el hecho de que cuando detecta que el canal está ocupado, en vez de permanecer a la espera escuchándolo, espera un tiempo aleatorio y vuelve a escuchar el canal. Con este método se reducen las colisiones si el tráfico es elevado, mejorándose la utilización del canal. Sin embargo aumentan los retardos para cargas de tráfico bajas .
- **CSMA p-persistente:** al igual que en los casos anteriores se escucha el canal, sin embargo si éste está libre, en vez de transmitir inmediatamente, se transmite con una probabilidad p , o bien se retrasa la emisión una ranura temporal con una probabilidad $q=1-p$. Esta ranura temporal suele ser igual al máximo retardo de propagación de la señal.

Funcionamiento

Funcionamiento general

En CSMA/CD, cada estación que desea transmitir debe realizar una escucha del medio –detección de portadora– para comprobar si éste se encuentra libre, es decir, para comprobar que ninguna otra estación está en ese instante transmitiendo un mensaje. Si el medio se encuentra libre entonces tiene lugar dicha transmisión. Aun así, puede ocurrir que varias estaciones tengan mensajes para enviar y que comiencen a transmitir una trama en el mismo instante. Cuando esto se sucede, se dice que ha ocurrido una colisión en la red. La estación que ha detectado la colisión procederá a enviar un mensaje de *jam* de 32 bits al resto de estaciones para notificar dicho evento. Una vez que todas las estaciones han sido notificadas, automáticamente se paran todas las transmisiones y se ejecuta un algoritmo de *backoff* (o de postergación) que consiste en esperar un tiempo aleatorio antes de volver a intentar la transmisión. Durante los 10 primeros intentos el valor medio del tiempo de espera se duplica mientras que durante los 6 siguientes intentos adicionales, se mantiene. Tras 16 intentos fallidos, el algoritmo notificará un error a las capas superiores.

Ventajas

- La detección de colisiones en redes LAN cableadas es fácil.
- El tiempo medio necesario para detectar una colisión es relativamente bajo.
- Puede ser empleado en sistemas de control de procesos continuos si la carga de tráfico de la red es baja (inferior al 20 %).
- Ofrece un rendimiento mayor en especial cuando existen pocas colisiones.

Desventajas

- Una de las desventajas más importantes radica en que no es posible garantizar un tiempo máximo finito para el acceso de las tramas al canal de comunicación, por lo cual no resulta adecuado para aplicaciones de tiempo real.
- Normalmente las redes CSMA/CD son de tipo half dúplex, lo cual significa que mientras una estación envía información es incapaz de escuchar el tráfico existente.
- Problemática en redes inalámbricas.

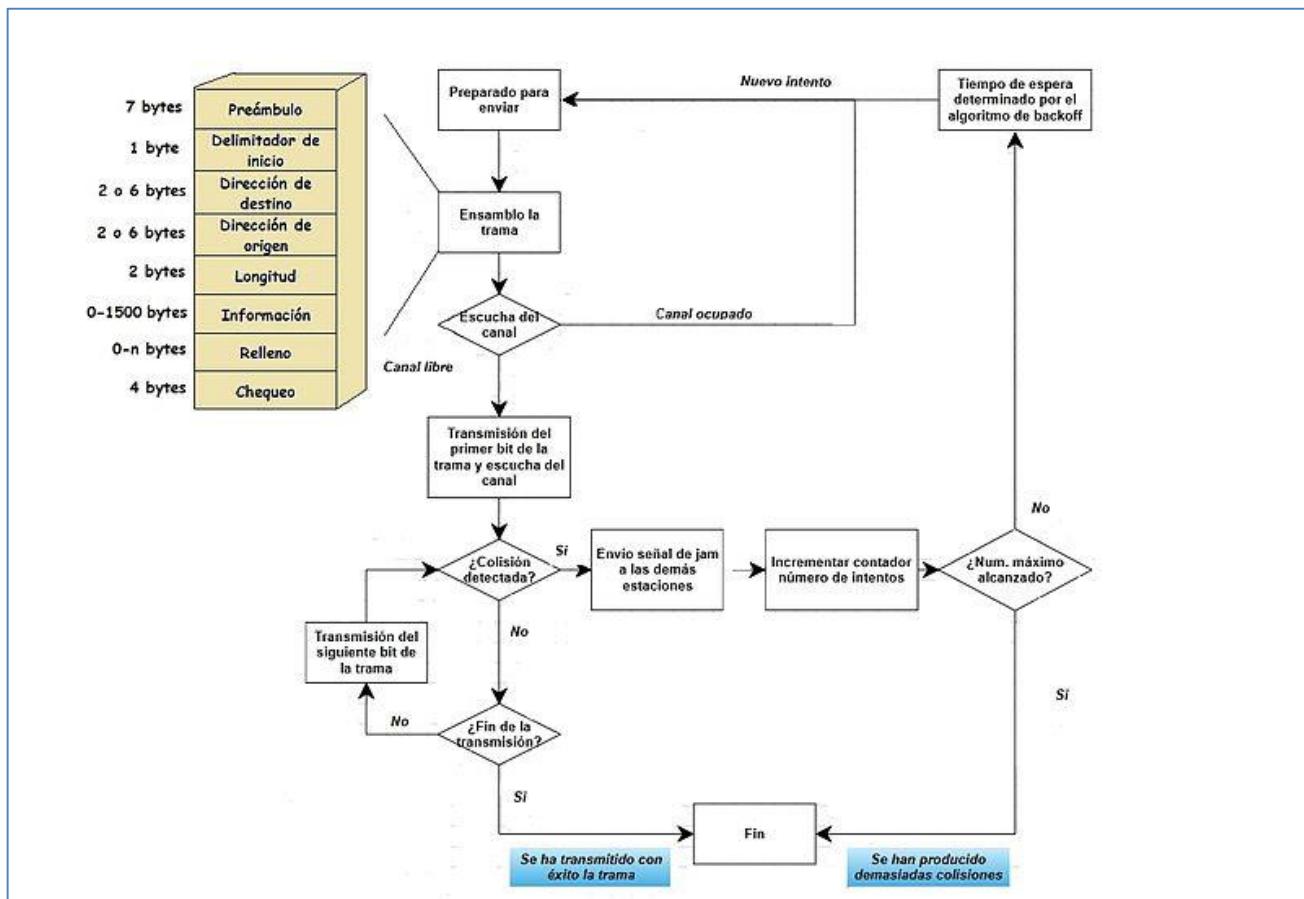


Ilustración 96 - Algoritmo CSMA/CD

Problemática en redes inalámbricas

En las redes inalámbricas proceder a la escucha del medio y por lo tanto detectar las colisiones producidas, puede resultar complicado. Esto se manifiesta en dos problemáticas:

- **Problema del nodo oculto:** una estación puede creer que el canal (medio) está libre cuando en realidad está ocupado por otra estación a la que no oye. En la siguiente imagen se muestra como A y C transmiten hacia B ya que ambos detectaron que el canal estaba libre. Sin embargo B escucha a ambos nodos, dando lugar a una colisión.

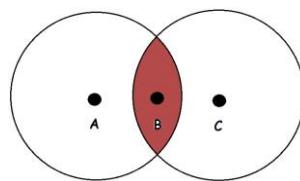


Ilustración 97 - Nodo oculto.

- **Problema del nodo expuesto:** una estación puede creer que el canal está ocupado cuando en realidad lo está ocupando otra estación que no interferiría en su transmisión a otro destino. En la figura se muestra como C está comunicándose con B. Como D detecta que el canal está ocupado, no puede transmitir hacia E, cuando lo idóneo sería que sí pudiese.

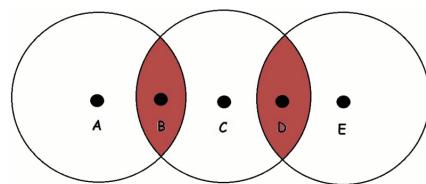


Ilustración 98 - Nodo expuesto.

Estos problemas fueron resueltos con la implementación del algoritmo CSMA/CA (*MultiAccess Collision Avoidance*)

Bibliografía y referencias

- https://es.qwe.wiki/wiki/Ethernet_hub (Consultada por última vez: 01/10/2020)
- <https://es.wikipedia.org/wiki/Repetidor> (Consultada por última vez: 01/10/2020)
- <https://community.fs.com/es/blog/patch-cable-vs-crossover-cable-what-is-the-difference.html> (Consultada por última vez: 01/10/2020)
- https://es.wikipedia.org/wiki/Tarjeta_de_red (Consultada por última vez: 01/10/2020)
- https://es.wikipedia.org/wiki/Puente_de_red (Consultada por última vez: 01/10/2020)
- [https://es.wikipedia.org/wiki/Conmutador_\(dispositivo_de_red\)](https://es.wikipedia.org/wiki/Conmutador_(dispositivo_de_red)) (Consultada por última vez: 01/10/2020)
- <https://es.wikipedia.org/wiki/Router> (Consultada por última vez: 01/10/2020)
- https://es.wikipedia.org/wiki/Puerta_de_enlace (Consultada por última vez: 01/10/2020)
- https://tecnologia.facilisimo.com/que-son-el-dominio-de-collision-y-el-dominio-de-difusion_2402064.html (Consultada por última vez: 01/10/2020)
- https://es.qwe.wiki/wiki/Medium_access_control (Consultada por última vez: 01/10/2020)
- <https://www.mikroways.net/2009/11/02/protocolo-spanning-tree/> (Consultada por última vez: 01/10/2020)
- https://es.wikipedia.org/wiki/Carrier_sense_multiple_access_with_collision_detection (Consultada por última vez: 01/10/2020)