**Capa 2: Capa de Enlace de Datos**

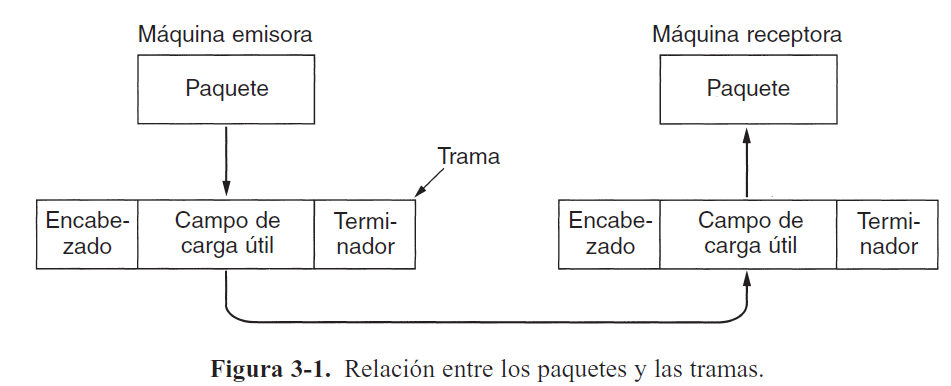
La propiedad esencial de un canal que lo hace asemejarse a un alambre es que los bits se entregan con exactitud en el mismo orden en que fueron enviados.

Los circuitos de comunicación cometen errores ocasionales. Tienen una tasa de datos finita y hay un retardo de propagación diferente de cero. Los protocolos usados deben considerar todos estos factores.

**Cuestiones de Diseño de la capa de Enlace de Datos**

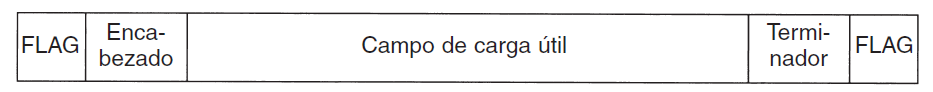
La capa de enlace de datos tiene cuatro funciones principales:

* Proporcionar una interfaz de servicio bien definida con la capa de red.
* Gestionar la forma en se generan las tramas o marcos (según sea emisor o receptor, debe ser capaz de armar y desarmar las tramas).
* Manejar los errores de transmisión
* Regular el flujo de datos para que receptores lentos no sean saturados por emisores rápidos.

Para cumplirlas, la capa de enlace de datos toma de la capa de red los paquetes y los encapsula en **tramas** (frame o marco) para transmitirlos. El manejo de las tramas es la tarea primordial de esta capa.

Cada trama contiene:

* Encabezado
* Campo de carga útil (*payload*) para almacenar el paquete
* Terminador o final (permite sabe dónde comienza y termina cada trama)



Servicios proporcionados a la Capa de Red

La función de la capa de enlace de datos es suministrar servicios a la capa de red. El servicio principal es transferir datos de la capa de red en la máquina de origen a la capa de red en la máquina destino. En la capa de red de la máquina de origen hay una entidad, llamada *proceso*, que entrega algunos bits a la capa de enlace de datos para transmitirlos a la máquina de destino. El trabajo de la capa de enlace de datos es transmitir los bits a la máquina de destino, para que puedan ser entregados a su capa de red.

Tres posibilidades razonables de servicios que normalmente se proporcionan son:

* *Servicio no orientado a la conexión sin confirmación de recepción*

Consiste en que la máquina de origen envíe tramas independientes a la máquina de destino sin pedir que ésta confirme la recepción. No se establece conexión de antemano ni se libera después. No se realiza ningún intento por detectar pérdidas ni recuperarse de ellas.

Este servicio es apropiado cuando la tasa de errores es muy baja o para el tráfico en tiempo real (voz, por ejemplo). La mayoría de las LANs utilizan este servicio en la capa de enlace de datos.

* *Servicio no orientado a la conexión con confirmación de recepción*

Tampoco utiliza conexiones lógicas, pero se confirma de manera individual la recepción de cada trama enviada. Así, el emisor sabe si la trama ha llegado bien o no. Si no ha llegado en un tiempo especificado, puede enviarse nuevamente. Este servicio es útil en canales inestables, como en los sistemas inalámbricos.

* Servicio orientado a la conexión con confirmación de recepción

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, que será recibida exactamente una vez y que todas se recibirán en el orden adecuado.

Las transferencias tienen tres fases. En la primera, la conexión se establece haciendo que ambos lados inicialicen las variables y los contadores necesarios para seguir la pista de las tramas que han sido recibidas y las que no. En la segunda, se transmiten una o más tramas. En la tercera, la conexión se cierra y libera las variables, los búferes y otros recursos utilizados para mantener la conexión.

Es responsabilidad del protocolo de enlace de datos hacer que las líneas de comunicación no estables parezcan perfectas o, cuando menos, bastante buenas.

Entramado

A fin de proporcionar servicios a la capa de red, la de enlace de datos debe utilizar los servicios que la capa física le proporciona. Lo que hace la capa física es aceptar un flujo de bits puros e intentar entregarlo al destino. Es responsabilidad de la capa de enlace de datos detectar y, de ser necesario, corregir los errores.

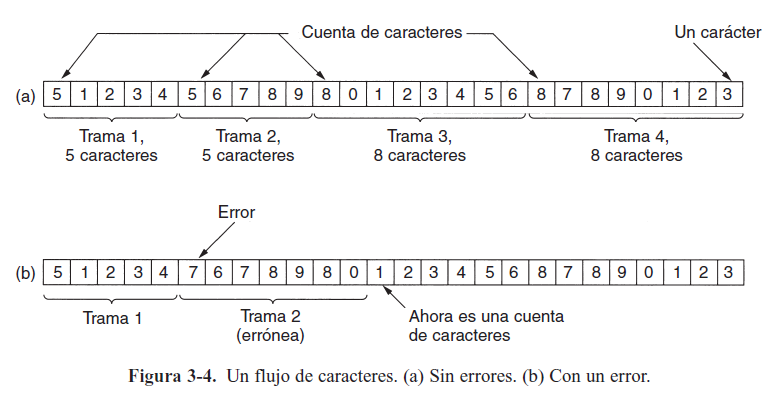
Puesto que es demasiado riesgoso depender de la temporización para marcar el inicio y el final de cada trama, se han diseñado otros métodos:

* *Conteo de caracteres*

Se vale de un campo en el encabezado para especificar el número de caracteres en la trama (no usa un finalizador). Cuando la capa de enlace de datos del destino ve la cuenta de caracteres, sabe cuántos caracteres siguen y, por lo tanto, dónde está el fin de la trama. Además, es muy susceptible a errores y depende de que el canal sea estable.

Problema: la cuenta puede alterarse por un error de transmisión. El destino perderá la sincronía y será incapaz de localizar el inicio de la siguiente trama. Incluso si el destino sabe que la trama está mal porque la suma de verificación es incorrecta, no tiene forma de saber dónde comienza la siguiente trama. Regresar una trama a la fuente solicitando una retransmisión tampoco ayuda, ya que el destino no sabe cuántos caracteres tiene que saltar para llegar al inicio de la retransmisión. Casi no se utiliza este método en la actualidad.

Esta técnica se muestra en la figura 3-4(a) para cuatro tramas de 5, 5, 8 y 8 caracteres de longitud, respectivamente.



Como se puede observar, en la Figura 3-4 (b) se alteró la cuenta de la segunda trama (debería ser 5 pero ahora es 7) y de esa manera se pierde sincronismo, entendiendo que la cuenta de la siguiente trama es 1 (en realidad es un carácter de datos y no de cuenta).

* *Banderas con relleno de caracteres*

Evita el problema de tener que sincronizar nuevamente después de un error, haciendo que cada trama inicie y termine con bytes especiales. Se denominan **banderas** (FLAG o indicadores) a los delimitadores de inicio y final. Si el receptor pierde la sincronía, simplemente puede buscar la bandera para encontrar el final e inicio de la trama actual. Dos banderas consecutivas señalan el final de una trama y el inicio de la siguiente.

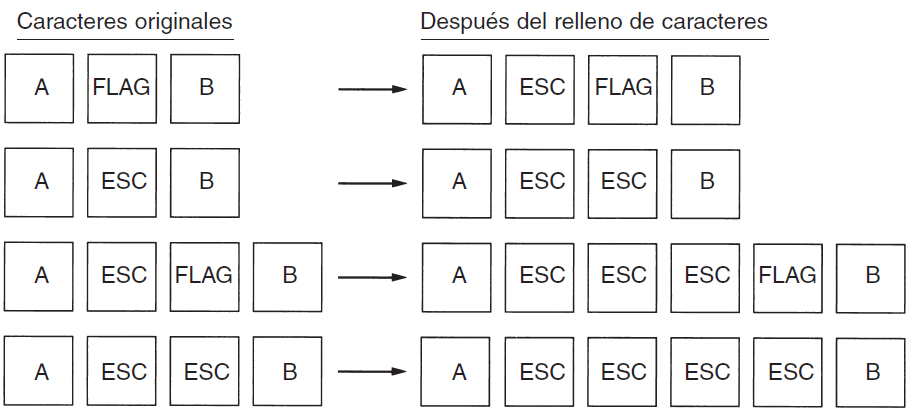
Situaciones posibles:

* + Problema: El patrón de bits de una bandera aparece en los datos.

Resolución: hacer que la capa de enlace de datos del emisor inserte un byte de escape especial (ESC) justo antes de cada bandera en los datos. La capa de enlace de datos del lado receptor quita el byte de escape antes de entregar los datos a la capa de red. Esta técnica se denomina **relleno de caracteres**. Una bandera de entramado se puede distinguir de una en los datos por la ausencia o presencia de un byte de escape que la antecede.

* + Problema: Un byte de escape aparece en los datos.

Resolución: También se rellena con un byte de escape. Cualquier byte de escape individual es parte de una secuencia de escape, mientras que uno doble indica que un escape sencillo apareció de manera natural en los datos.

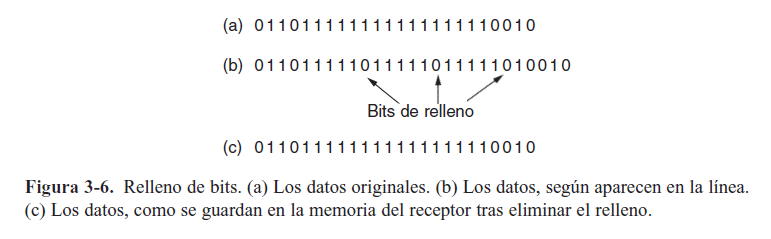
Una desventaja importante de esta técnica es que está fuertemente atada a los caracteres de 8 bits, siendo que no todos los códigos utilizan caracteres de 8 bits. Por ello, tuvo que desarrollarse una técnica nueva que permitiera caracteres de tamaño arbitrario.

* *Banderas de inicio y fin con relleno de bits*

Permite que las tramas de datos contenga un número arbitrario de bits y admite códigos de caracteres con un número arbitrario de bits por carácter. Cada trama comienza y termina con un patrón especial de bits, 01111110 (que es una bandera). Cada vez que la capa de enlace de datos del emisor encuentra cinco unos consecutivos en los datos, automáticamente inserta un bit 0 en el flujo de bits saliente. Este **relleno de bits** es análogo al relleno de caracteres.

Cuando el receptor ve cinco bits 1 de entrada consecutivos, seguidos de un bit 0, automáticamente borra el bit 0 de relleno.

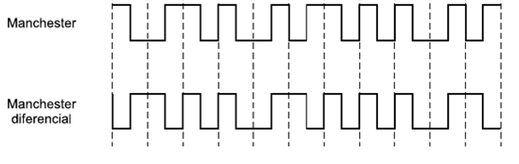
Si el receptor pierde la pista de dónde está, todo lo que tiene que hacer es explorar la entrada en busca de secuencias de banderas, pues sólo pueden ocurrir en los límites de las tramas y nunca en los datos.



* *Violaciones de codificación de la capa física*

Sólo se aplica a las redes en las que la codificación en el medio físico contiene cierta redundancia. Algunas LANs codifican un bit de datos usando dos bits físicos. Normalmente, un bit 1 es un par alto-bajo, y un bit 0 es un par bajo-alto. El esquema implica que cada bit de datos tiene una transición a medio camino, lo que hace fácil para el receptor localizar los límites de los bits. Las combinaciones alto-alto y bajo-bajo no se usan para datos, pero en algunos protocolos se utilizan para delimitar tramas, es decir, como inicio y fin de trama se envía cierto patrón no esperado (incoherente) por la técnica de codificación.

* **Manchester**: siempre hay una transición en mitad del intervalo de duración del bit. Esta transición en la mitad del bit sirve como procedimiento de sincronización, a la vez que sirve para transmitir los datos. Una transición de bajo a alto representa un 1, y una transición de alto a bajo representa un 0.
* **Manchester diferencial**: la transición a mitad del intervalo se utiliza tan sólo para proporcionar sincronización. La codificación de un 0 se representa por la presencia de una transición al principio del intervalo del bit, y un 1 se representa mediante la ausencia de una transición al principio del intervalo.



Muchos protocolos de enlace de datos usan, por seguridad, una combinación de cuenta de caracteres con uno de los otros métodos. Cuando llega una trama, se usa el campo de cuenta para localizar el final de la trama. Sólo si el delimitador apropiado está presente en esa posición y la suma de verificación es correcta, la trama se acepta como válida. De otra manera, se explora el flujo de entrada en busca del siguiente delimitador.

**La Subcapa de Control de Acceso al Medio**

Los canales de difusión a veces se denominan **canales multiacceso** o **canales de acceso aleatorio**.

Los protocolos usados para determinar quién sigue en un canal multiacceso pertenecen a una subcapa de la capa de enlace de datos llamada subcapa **MAC (Control de Acceso al Medio)**. La subcapa MAC tiene especial importancia en las LANs, casi todas las cuales usan un canal multiacceso como base para su comunicación. Las WANs usan enlaces punto a punto, excepto en las redes satelitales.

La subcapa MAC es la parte inferior de la capa de enlace de datos.

**El Problema de la Asignación del Canal**

Asignación estática de canal en LANs y MANs

La manera tradicional de asignar un solo canal entre varios usuarios competidores es la FDM (Multiplexación por División de Frecuencia). 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. Dado que cada usuario tiene una banda de frecuencia privada, no hay interferencia entre los usuarios. Cuando sólo hay una pequeña cantidad fija de usuarios, cada uno de los cuales tiene (en búfer) una carga de tráfico pesada, la FDM es un mecanismo de asignación sencillo y eficiente.

Sin embargo, cuando el número de emisores es grande y varía continuamente, o cuando el tráfico se hace en ráfagas, la FDM presenta algunos problemas. Si el espectro se divide en *N* regiones, y hay menos de *N* usuarios interesados en comunicarse actualmente, se desperdiciará una buena parte de espectro valioso. Si más de *N* usuarios quieren comunicarse, a algunos de ellos se les negará el permiso por falta de ancho de banda, aun cuando algunos de los usuarios que tengan asignada una banda de frecuencia apenas transmitan o reciban algo.

Sin embargo, aun suponiendo que el número de usuarios podría, de alguna manera, mantenerse constante en *N*, dividir el canal disponible en subcanales estáticos es inherentemente ineficiente. El problema básico es que, cuando algunos usuarios están inactivos, su ancho de banda simplemente se pierde. No lo están usando, y a nadie más se le permite usarlo. Es más, en casi todos los sistemas de cómputo el tráfico de datos se hace en ráfagas. En consecuencia, la mayoría de los canales estarán inactivos casi todo el tiempo.

Asignación dinámica de canales en LANs y MANs

Antes de entrar en el primero de muchos métodos de asignación de canal que se analizarán en este capítulo, vale la pena formular con cuidado el problema de la asignación. Todo el trabajo hecho en esta área se basa en cinco supuestos clave, que se describen a continuación:

1. **Modelo de estación.** El modelo consiste en *N* **estaciones** independientes (computadoras, teléfonos, servidores, etc.), cada una con un programa o usuario que genera tramas para transmisión. También se conocen como **terminales**. Una vez que se ha generado la trama, la estación se bloquea y no hace nada sino hasta que la trama se ha transmitido con éxito. En el modelo como mínimo, hay siempre 2 estaciones o terminales.
2. **Supuesto de canal único.** Hay un solo canal disponible para todas las comunicaciones. Todas las estaciones pueden transmitir en él y pueden recibir de él. En lo referente al hardware, todas las estaciones son equivalentes, aunque el software del protocolo puede asignarles prioridades.
3. **Supuesto de colisión.** Si dos tramas se transmiten en forma simultánea, se traslapan en el tiempo y la señal resultante se altera. Además se produce una diferencia de tensión que puede ser detectada por las estaciones. Este evento se llama **colisión**. Todas las estaciones pueden detectar colisiones, para ello, comienzan a transmitir sus tramas y verifican constantemente el canal en busca de una ráfaga de ruido. Una trama en colisión debe transmitirse nuevamente después. No hay otros errores excepto aquellos generados por las colisiones.
4. **Tiempo continuo.** La transmisión de una trama puede comenzar en cualquier momento. No hay reloj maestro que divida el tiempo en intervalos discretos.
5. **Tiempo ranurado.** El tiempo se divide en intervalos discretos (ranuras). La transmisión de las tramas siempre comienza al inicio de una ranura. Una ranura puede contener 0, 1 o más tramas, correspondientes a una ranura inactiva, una transmisión con éxito o una colisión, respectivamente.
6. **Detección de portadora.** Las estaciones pueden saber si el canal está en uso, antes de intentar usarlo. Si se detecta que el canal está en uso, ninguna estación intentará utilizarlo sino hasta que regrese a la inactividad. El hecho de poder detectar una señal portadora tiene la particularidad de que las estaciones escuchan el canal sin transmitir nada.
7. **Sin detección de portadora.** Las estaciones no pueden detectar el canal antes de intentar usarlo. Simplemente transmiten. Sólo después pueden determinar si la transmisión tuvo éxito.

*[Nota: Para los supuestos 4 y 5, sólo una de las alternativas dentro de cada uno es válida en un sistema dado.]*

**Protocolos de Acceso Múltiple (o de acceso al medio)**

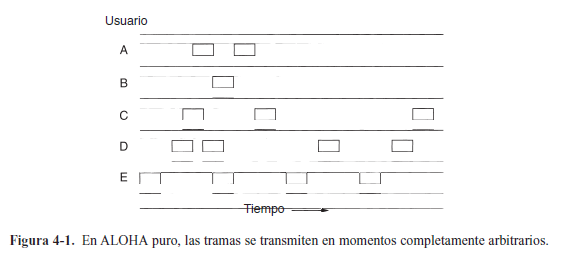
ALOHA Puro (década de 1970 – Norman Abramson – Universidad de Hawaii)

La idea básica es permitir que los usuarios transmitan cuando tengas datos para enviar. Por supuesto, habrá colisiones y las tramas involucradas se dañarán. Un emisor siempre puede saber si la trama fue destruida o no escuchando el canal. Si por alguna razón no es posible escuchar mientras se transmite, se necesitan confirmaciones de recepción. Si la trama fue destruida, el emisor simplemente espera un tiempo aleatorio y la envía de nuevo. El tiempo de espera debe ser aleatorio, o las mismas tramas chocarán una y otra vez, en sincronía. Los sistemas en los cuales varios usuarios comparten un canal común de modo tal que puede dar pie a conflictos se conocen como sistemas de **contención**.

La velocidad real de transporte de los sistemas ALOHA se maximiza al tener tramas con un tamaño uniforme en lugar de tramas de longitud variable.

Cada vez que dos tramas traten de ocupar el canal al mismo tiempo, habrá una colisión y ambas se dañarán. La suma de verificación no puede ni debe distinguir entre una pérdida total y un error ligero.

Por último, cabe aclarar que ALOHA puro no presenta detección de portadora y que lo máximo que podemos esperar es un uso del canal del **18%**.



ALOHA Ranurado

Para duplicar la capacidad de un sistema ALOHA, se propuso dividir el tiempo en intervalos discretos, cada uno de los cuales correspondía a una trama. Este enfoque requiere que los usuarios acuerden límites de ranura. Una manera de lograr la sincronización sería tener una estación especial que emitiera una señal al comienzo de cada intervalo, como un reloj.

En el método ALOHA Ranurado no se permite que una computadora envíe cada vez que se pulsa un retorno de carro. Se le obliga a esperar el comienzo de la siguiente ranura. Lo mejor que podemos esperar usando ALOHA ranurado es 37% de ranuras vacías, **37%** de éxitos y 26% de colisiones. Además, tampoco presenta la característica de poder detectar señales portadoras.

Protocolos de Acceso Múltiple con Detección de Portadora

CSMA Persistente y No Persistente

* CSMA (Acceso Múltiple con Detección de Portadora) persistente-1

Cuando una estación tiene datos por transmitir, primero escucha el canal para saber si otra está transmitiendo en ese momento. Si el canal está ocupado, la estación espera hasta que se desocupe. Cuando la estación detecta un canal inactivo, transmite una trama. Si ocurre una colisión, la estación espera una cantidad aleatoria de tiempo y comienza de nuevo. El protocolo se llama persistente-1 porque la estación transmite con una probabilidad de 1 cuando encuentra que el canal está inactivo.

El retardo de propagación tiene un efecto importante en el desempeño del protocolo. Hay una pequeña posibilidad de que, justo después de que una estación comience a transmitir, otra estación esté lista para enviar y detectar el canal. Si la señal de la primera estación no ha llegado aún a la segunda, esta última detectará un canal inactivo y comenzará a enviar también, lo que dará como resultado una colisión. Cuanto mayor sea el tiempo de propagación, más importante será este efecto, y peor el desempeño del protocolo.

Aún si el retardo de propagación es cero, habrá colisiones. Sin embargo, este protocolo es mucho mejor que el ALOHA.

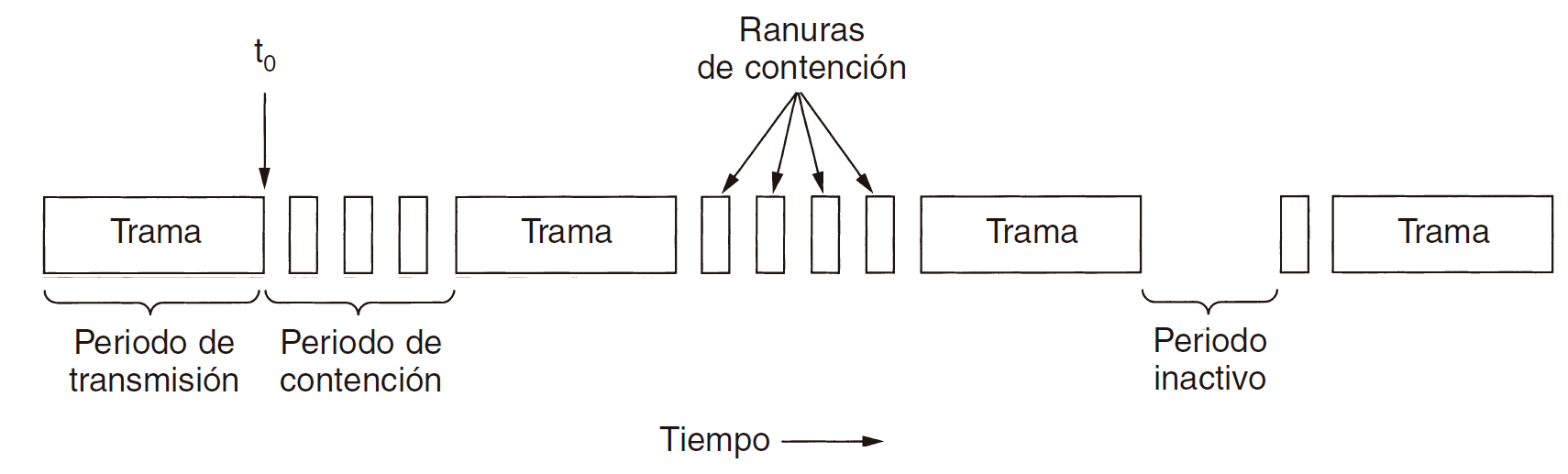
* CSMA No persistente

Antes de enviar, una estación escucha el canal. Si nadie más está transmitiendo, la estación comienza a hacerlo. Sin embargo, si el canal ya está en uso, la estación espera un período aleatorio y repite el algoritmo. En consecuencia, este algoritmo conduce a un mejor uso del canal pero produce mayores retardos que el CSMA persistente-1.

* CSMA Persistente-p

Se aplica a canales ranurados. Cuando una estación está lista para enviar, escucha el canal. Si éste se encuentra inactivo, la estación transmite con una probabilidad *p*. Con una probabilidad , se espera hasta la siguiente ranura. Si esta ranura también está inactiva, la estación transmite o espera nuevamente, con probabilidad *p* y *q*. Este proceso se repite hasta que la trama ha sido transmitida o hasta que otra estación ha comenzado a transmitir. Si la probabilidad p de esa estación determina que no es su “turno” para transmitir, la estación actúa como si hubiera habido una colisión (es decir, espera un tiempo aleatorio y comienza de nuevo). Si al inicio la estación detecta que el canal está ocupado, espera hasta la siguiente ranura y aplica el algoritmo anterior.

CSMA con Detección de Colisiones



Los protocolos CSMA persistentes y no persistentes son una mejora respecto a ALOHA porque aseguran que ninguna estación comienza a transmitir cuando detecta que el canal está ocupado.

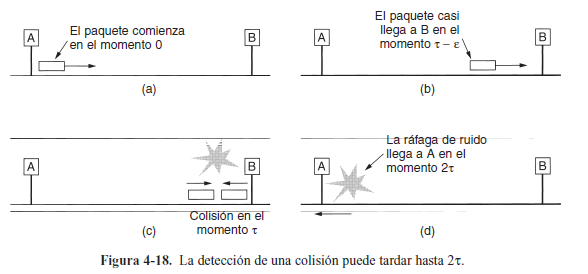
Otra mejora es que las estaciones aborten sus transmisiones tan pronto como detecten una colisión: en lugar de terminar de transmitir sus tramas, que de todos modos están irremediablemente alteradas, deben detener de manera abrupta la transmisión tan pronto como detectan la colisión. La terminación pronta de tramas dañadas ahorra tiempo y ancho de banda. Este protocolo, conocido como **CSMA/CD (Acceso Múltiple con Detección de Portadora y Detección de Colisiones)**, se usa ampliamente en las LANs en la subcapa MAC.

Una vez que una estación detecta una colisión, aborta la transmisión, espera un tiempo aleatorio e intenta de nuevo, suponiendo que ninguna otra estación ha comenzado a transmitir durante ese lapso. Por lo tanto, el modelo de CSMA/CD consistirá en periodos alternantes de contención y transmisión, ocurriendo periodos de inactividad cuando todas las estaciones están en reposo (por falta de trabajo, por ejemplo).

Supongamos que dos estaciones comienzan a transmitir de manera exacta en el mismo momento. ¿En cuánto tiempo se darán cuenta de que ha habido una colisión? El tiempo mínimo para detectar la colisión es sólo el tiempo que tarda la señal para propagarse de una estación a otra. Esta respuesta es vital para determinar la longitud del período de contención y, por lo tanto, el retardo y la velocidad real de transporte.

Se podría pensar que una estación que después de iniciar su transmisión no detecta una colisión durante un tiempo igual al tiempo completo de propagación del cable podrá estar segura de que ha tomado el cable (todas las demás estaciones sabían que estaba transmitiendo y no interfirieron). Esta conclusión es equivocada. En el peor caso una estación no puede estar segura de que ha tomado el canal hasta que ha transmitido durante sin detectar una colisión (tiempo que tarda una señal en propagarse entre las dos estaciones más lejanas).

En el *peor* de lo casos



La detección de colisiones es un proceso *analógico*. El hardware de la estación debe escuchar el cable mientras transmite. Si lo que lee es distinto de lo que puso en él, sabe que está ocurriendo una colisión. La implicación es que la codificación de la señal debe permitir que se detecten colisiones (por ejemplo, una colisión de dos señales de 0 voltios podría ser imposible de detectar).

Vale la pena mencionar que estación para debe monitorear de manera continua el canal en busca de ráfagas de ruido que indiquen que ha ocurrido una colisión. Es por este motivo que CSMA/CD con un solo canal es inherentemente un sistema semidúplex. Es imposible que una estación transmita y reciba tramas al mismo tiempo, debido a que la lógica de recepción está en uso en busca de colisiones durante cada transmisión.

Ningún protocolo de subcapa MAC garantiza la entrega confiable. Incluso en ausencia de colisiones, el receptor podría no haber copiado en forma correcta la trama por varias razones.

**Ethernet**

Tanto el estándar 802.3 (Ethernet) como el 802.11 (LAN Inalámbrica) tienen diferentes capas físicas y diferentes subcapas MAC, pero convergen en la misma subcapa de control lógico del enlace (que se define en el 802.2), por lo que tienen la misma interfaz a la capa de red.

Comúnmente se usan cuatro tipos de cableado:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nombre** | **Cable** | **Seg. Máx.** | **Nodos/seg** | **Ventajas** | **Topologías** |
| 10Base5 | Coaxial grueso | 500 m | 100 | Cable original; ahora obsoleto | Bus |
| 10Base2 | Coaxial delgado | 185 m | 30 | No se necesita concentrador | Bus |
| 10Base-T | Par Trenzado | 100 m | 1024 | Sistema más económico | Estrella |
| 10Base-F | Fibra Óptica | 2000 m | 1024 | Mejor entre edificios | Estrella |

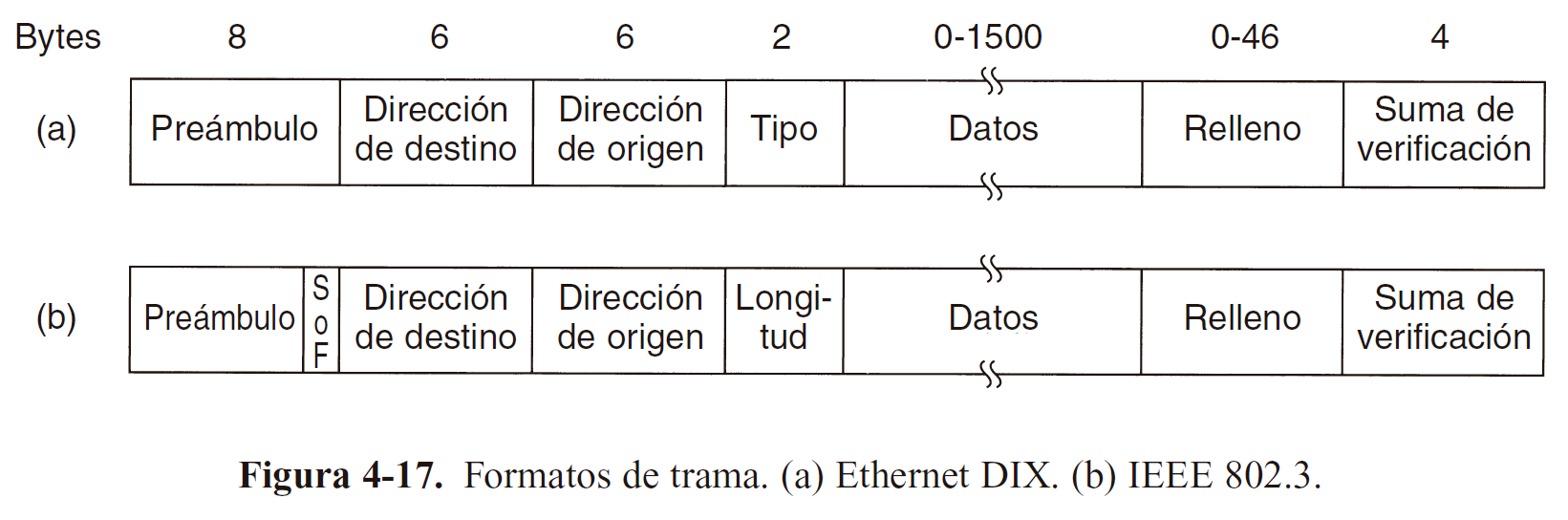
El primer número es la velocidad en Mbps (megabits por segundo), después viene la palabra “Base” para indicar la transmisión de banda base. Por último, si el medio es coaxial, su longitud se da redondeada a unidades de 100 m después de “Base”. Ejemplo: **10Base5** significa que opera a 10 Mbps, utiliza señalización de banda base y puede manejar segmentos de hasta 500 metros.

El segundo tipo de cable, **10Base2**, se dobla con facilidad y sus conectores son más fáciles de usar y más confiables. El Ethernet delgado es mucho más económico y fácil de instalar, pero sólo puede extenderse 185 metros por segmento, cada uno de los cuales puede manejar sólo 30 máquinas.

Los problemas asociados con la localización de rupturas de cable han empujado a los sistemas a un tipo de patrón de cableado diferente, en el que todas las estaciones tienen cables que conducen a un **concentrador** *(hub)* central, en el que se conectan de manera eléctrica. Por lo general, estos cables son pares trenzados telefónicos. Este esquema se llama **10Base-T**.

Una cuarta opción de cableado para Ethernet es **10Base-F**, que usa fibra óptica. Esta alternativa es cara debido al costo de los conectores y los terminadores, pero tiene excelente inmunidad contra el ruido y es el método a usar para conexiones entre edificios o entre concentradores muy separados. Se permiten separaciones de kilómetros entre conexiones. También ofrece buena seguridad debido a que es más difícil intervenir una conexión de fibra que una de cobre.

El protocolo de la subcapa MAC de Ethernet



Se muestra la estructura de trama original de DIX (DEC, Intel, Xerox). Cada trama inicia con un *Preámbulo* de 8 bytes, cada uno de los cuales contiene el patrón de bits 10101010. Se utiliza para la sincronización.

La trama contiene dos direcciones, una para el destino y una para el origen. El estándar permite direcciones de 2 y 6 bytes. 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 (multicast)**. La dirección que consiste únicamente en bits 1 está reservada para **difusión (broadcast) (FF:FF:FF:FF:FF:FF)**. Una trama de multidifusión se envía a un grupo seleccionado de estaciones de la Ethernet; una trama de difusión se envía a todas las estaciones de la Ethernet.

El campo de *Tipo* indica al receptor qué hacer con la trama y específica a qué proceso darle la trama. Después están los datos, de hasta 1500 bytes, límite establecido de manera algo arbitraria, principalmente con base en el hecho de que la RAM para los transceptores era muy cara en esa época.

Además de haber una longitud de trama máxima, también hay una longitud mínima. Cuando un transceptor detecta una colisión, trunca la trama actual, lo que significa que los bits perdidos y las piezas de las tramas aparecen todo el tiempo en el cable. Para que Ethernet pueda distinguir con facilidad las tramas válidas de la basura, necesita que dichas tramas tengan una longitud de por lo menos 64 bytes, de la dirección de destino a la suma de verificación, incluyéndolas. Si la porción de datos de una trama es menor que 46 bytes, el campo de *Relleno* se utiliza para rellenar 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 (explicado anteriormente en Figura 4-18). Si una estación intenta transmitir una trama muy corta, es concebible que ocurra una colisión, pero la transmisión se completa antes de que la ráfaga de ruido llegue de regreso. El emisor entonces supondrá incorrectamente que la trama se envió con éxito. Para evitar que esta situación ocurra, todas las tramas deberán tardar más de para enviarse (siendo el tiempo que tarda en llegar la trama al otro extremo), de manera que la transmisión aún esté llevándose a cabo cuando la ráfaga de ruido regrese al emisor.

El campo final de Ethernet es la *Suma de verificación*. De hecho, ésta es un código de *hash* de 32 bits de los datos. Si algunos bits de datos se reciben erróneamente (debido a ruido en el cable), es casi seguro que la suma de verificación esté mal y se detecte el error. El algoritmo de suma de verificación es *comprobación de redundancia cíclica (CRC)*.

Cuando IEEE estandarizó Ethernet, realizó dos cambios al formato DIX:

1. Redujo el preámbulo a 7 bytes (cada uno compuesto por el patrón 10101010) y utilizó el último byte para un *delimitador de Inicio de trama* *(SoF)* 101010**11**.
2. Cambió el campo de *Tipo* a un campo de *Longitud*. El problema de no saber ahora qué hacer con la trama entrante se resolvió con la adición de un pequeño encabezado a la porción de datos para proporcionar dicha información.

Además cabe aclarar como dato importante que el IEEE 802.3 utiliza como protocolo de acceso múltiple *CSMA P-1/CD.*

Direcciones MAC

Son direcciones asignadas a las interfaces físicas. Son de 6 bytes separados por “:”, de los cuales los 3 primeros corresponden al fabricante y los 3 últimos identifican al dispositivo con su respectivo modelo, teniendo cada dispositivo un número diferente.

Esos 6 bytes se representan de forma hexadecimal en 6 pares. Ejemplo: 01:23:45:67:89:AB

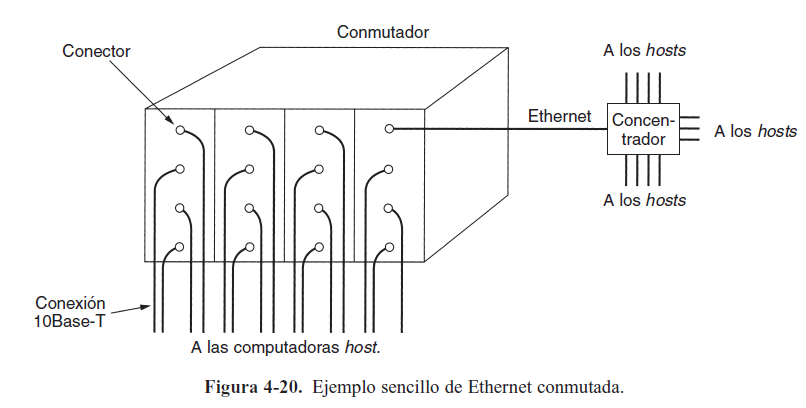
Tal como se mencionó anteriormente, dentro de las direcciones MAC, hay una que es especialmente dedicada para broadcast ( FF:FF:FF:FF:FF:FF). Esta dirección no está asignada a ningún dispositivo en particular sino que indica que lo que se está enviando, llegará a TODOS los dispositivos.

Comúnmente se suele decir que estas direcciones son planas ya que no me dan ayuda de cómo está conformada la red.

**Ethernet conmutada**

A medida que se agregan más y más estaciones a una Ethernet, aumenta el tráfico. En algún momento, la LAN se saturará. Una solución al problema es utilizar una velocidad mayor, digamos 100 Mbps en lugar de 10 Mbps. Pero con el crecimiento de la multimedia, incluso una Ethernet de 100 Mbps o de 1 Gbps puede saturarse.

Afortunadamente existe una solución diferente para tratar con el aumento de carga: una Ethernet conmutada, como la que se muestra en la figura 4-20. El corazón de este sistema es un conmutador (switch) que contiene una matriz de conmutación de alta velocidad y espacio (típicamente) para 4 a 32 tarjetas de línea, cada una de las cuales contiene de uno a ocho conectores. Lo más común es que cada conector tenga una conexión de cable de par trenzado 10Base-T a una sola computadora host.



Cuando una estación quiere transmitir una trama Ethernet, envía una trama estándar al conmutador. La tarjeta que recibe la trama la revisa para ver si está destinada a una de las otras estaciones conectadas a la misma tarjeta. De ser así, la trama se copia ahí. Si no, la trama se envía a través de la matriz de conmutación de alta velocidad a la tarjeta de la estación de destino. Por lo general, dicha matriz de conmutación funciona a más de 1 Gbps usando un protocolo patentado.

¿Qué ocurre si dos máquinas conectadas a la misma tarjeta de conexión transmiten tramas al mismo tiempo? Depende de la manera en que haya sido construida la tarjeta. Una posibilidad es que todos los puertos de la tarjeta estén alambrados entre sí para formar una LAN local dentro de la tarjeta. Las colisiones en esta LAN en tarjeta se detectan y manejan igual que cualquier otra colisión en una red CSMA/CD, en las que las retransmisiones utilizan el algoritmo de retroceso exponencial binario. Con este tipo de tarjeta sólo es posible una transmisión por tarjeta en un momento dado, pero todas las tarjetas pueden estar transmitiendo en paralelo. Con este diseño, *cada tarjeta* forma su propio dominio de colisión, independiente de los demás. Con sólo una estación por dominio de colisión, las colisiones son imposibles y el desempeño se mejora.

Con el otro tipo de tarjeta de conexión, cada puerto de entrada se almacena en un búfer, por lo que las tramas de entrada se almacenan en la RAM de la tarjeta conforme llegan. Este diseño permite que todos los puertos de entrada reciban (y transmitan) tramas al mismo tiempo, para una operación en paralelo completamente dúplex, algo que no es posible con CSMA/CD en un solo canal. Con este diseño, *cada puerto* es un dominio de colisión independiente, por lo que no ocurren colisiones.

Dado que el conmutador sólo espera tramas Ethernet estándar en cada puerto de entrada, es posible utilizar algunos de los puertos como concentradores. En la figura 4-20, el puerto de la esquina superior derecha no está conectado a una sola estación, sino a un concentrador de 12 puertos.

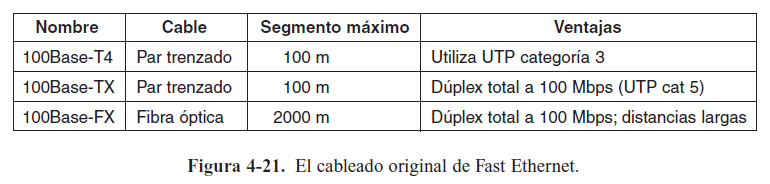
**Fast Ethernet (802.3u)**

El comité 802.3 decidió crear una Ethernet mejorada por tres razones principales:

1. La necesidad de compatibilidad hacia atrás con las LANs Ethernet existentes.
2. El miedo de que un nuevo protocolo tuviera problemas no previstos.
3. El deseo de terminar el trabajo antes de que la tecnología cambiara.

La idea básica detrás de Fast Ethernet era sencilla: mantener todos los formatos anteriores, interfaces y reglas de procedimientos, y sólo reducir el tiempo de bits de 100 nseg a 10 nseg. Técnicamente, habría sido posible copiar 10Base-5 o 10Base-2 y aún detectar colisiones a tiempo con sólo reducir la longitud máxima de cable por un factor de diez. Sin embargo, las ventajas del cableado 10Base-T eran tan abrumadoras que Fast Ethernet se basa por completo en este diseño. Por lo tanto, todos los sistemas Fast Ethernet utilizan concentradores y conmutadores; no se permiten cables con múltiples derivaciones vampiro ni conectores BNC.

Sin embargo, aún se tienen que tomar algunas decisiones, la más importante de las cuáles es qué tipos de cable soportar. Un contendiente era el cable de par trenzado categoría 3 que prácticamente todas las oficinas en el mundo occidental tienen por lo menos cuatro cables de par trenzado categoría tres (o mejor) que van de la oficina hacia un gabinete de cableado telefónico dentro de una distancia de 100 metros. La principal desventaja del cable de par trenzado categoría 3 es su incapacidad de llevar señales de 200 megabaudios (100 Mbps con codificación Manchester) a una distancia de hasta100 metros, que es la distancia máxima de computadora a concentrador especificada para 10Base-T (vea la figura 4-13). En contraste, el cable de par trenzado categoría 5 puede manejar 100 metros con facilidad, y la fibra puede ir mucho más rápido. El arreglo elegido fue permitir las tres posibilidades, como se muestra en la figura 4-21, pero fortalecer la solución categoría 3 para darle la capacidad de transmisión adicional necesaria.



El esquema UTP categoría 3, llamado **100Base-T4**, utiliza una velocidad de señalización de 25 MHz pero para alcanzar el ancho de banda necesario, 100Base-T4 requiere cuatro cables de par trenzado. Debido a que el cableado telefónico estándar durante décadas ha tenido cuatro cables de par trenzado por cable.

Para el cableado categoría 5, el diseño **100Base-TX** es más simple porque los cables pueden manejar velocidades de reloj de 125 MHz. Sólo se utilizan dos cables de par trenzado por estación, uno para enviar y otro para recibir.

La última opción, **100Base-FX**, utiliza dos filamentos de fibra multimodo, una para cada dirección, por lo que también es dúplex total con 100 Mbps en cada dirección. Además, la distancia entre una estación y el concentrador puede ser de hasta 2 km. Cabe aclarar que, los concentradores no están permitidos con 100Base-FX.

En respuesta a la demanda popular, en 1997 el comité 802 agregó un nuevo tipo de cableado, 100Base-T2, que permite que la Fast Ethernet se ejecute a través de dos pares de cables existentes de categoría 3. Sin embargo, se necesita un procesador de señales digital sofisticado para manejar el esquema de codificación requerido, lo que hace de esta opción algo muy costoso. Hasta ahora su uso es muy inusual debido a su complejidad y costo, así como al hecho de que muchos edificios de oficinas se han vuelto a cablear con UTP categoría 5.

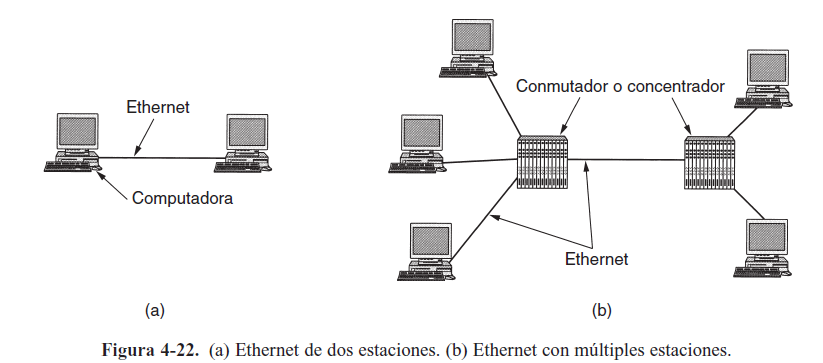
Como nota final, casi todos los conmutadores pueden manejar una mezcla de estaciones de 10 y 100 Mbps, para facilitar la actualización. El estándar mismo proporciona una forma para que dos estaciones negocien de manera automática la velocidad óptima (10 o 100 Mbps) y el tipo de transmisión dúplex (semi o total). La mayoría de los productos de Fast Ethernet utilizan esta característica para autoconfigurarse.

**Gigabit Ethernet**

La tinta apenas se estaba secando en el estándar de la Fast Ethernet cuando el comité 802 comenzó a trabajar en una Ethernet aún más rápida (1995). Se conoció como Gigabit Ethernet y fue aprobada por el IEEE en 1998 bajo el nombre **802.3z**.

Los objetivos del comité 802.3z eran esencialmente los mismos que los del comité 802.3u: hacer que Ethernet fuera 10 veces más rápida y que permaneciera compatible hacia atrás con todos los estándares Ethernet existentes.

Todas las configuraciones de Gigabit Ethernet son de punto a punto en lugar de múltiples derivaciones como en el estándar original de 10 Mbps, ahora conocido como *Ethernet clásica*.



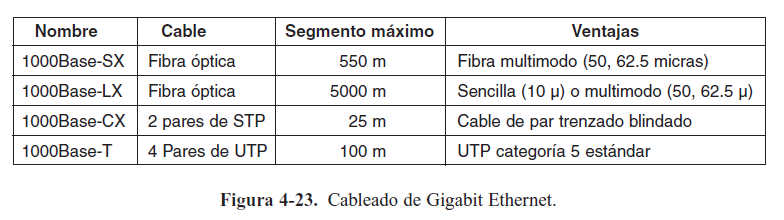
En la configuración más simple de Gigabit Ethernet, que se muestra en la figura 4-22(a), dos computadoras están conectadas de manera directa entre sí. Sin embargo, el caso más común es tener un conmutador o un concentrador conectado a múltiples computadoras y posiblemente a conmutadores o concentradores adicionales, como se muestra en la figura 4-22(b). En ambas configuraciones cada cable Ethernet individual tiene exactamente dos dispositivos en él, ni más ni menos.

Gigabit Ethernet soporta dos modos diferentes de funcionamiento: modo de dúplex total y modo de semidúplex.

* El modo “normal” es el de *dúplex total*, el cual permite tráfico en ambas direcciones al mismo tiempo. Este modo se utiliza cuando hay un conmutador central conectado a computadoras (o a otros conmutadores) en el periférico. En esta configuración, todas las líneas se almacenan en el búfer a fin de que cada computadora y conmutador pueda enviar tramas siempre que lo desee. El emisor no tiene que detectar el canal para ver si alguien más lo está utilizando debido a que la contención es imposible. En la línea entre una computadora y un conmutador, la computadora es el único emisor posible en esa línea al conmutador y la transmisión tiene éxito aun cuando el conmutador esté enviado actualmente una trama a la computadora (porque la línea es de dúplex total). Debido a que no hay contención, no se utiliza el protocolo CSMA/CD.
* El otro modo de operación, *semidúplex*, se utiliza cuando las computadoras están conectadas a un concentrador en lugar de a un conmutador. Un concentrador no almacena en el búfer las tramas entrantes. En su lugar, conecta en forma eléctrica todas las líneas internamente, simulando el cable con múltiples derivaciones que se utiliza en la Ethernet clásica. En este modo las colisiones son posibles, por lo que es necesario el protocolo CSMA/CD estándar. Debido a que una trama mínima (de 64 bytes) ahora puede transmitirse 100 veces más rápido que en la Ethernet clásica, la distancia máxima es 100 veces menor, o 25 metros, para mantener la propiedad esencial de que el emisor aún transmita cuando la ráfaga de ruido vuelva a él, incluso en el peor caso. Con un cable de 2500 metros de longitud, el emisor de una trama de 64 bytes a 1 Gbps podría terminar su transmisión antes de que la trama recorra una décima del camino, y muchísimo antes de que llegue al otro extremo y regrese. El comité 802.3z consideró un radio de 25 metros como inaceptable y agregó dos características al estándar para incrementar el radio. La primera, llamada *extensión de portadora*, esencialmente indica al hardware que agregue su propio relleno después de la trama normal para extenderla a 512 bytes. Puesto que este relleno es agregado por el hardware emisor y eliminado por el hardware receptor, el software no toma parte en esto, lo que significa que no es necesario realizar cambios al software existente. Por supuesto, utilizar 512 bytes de ancho de banda para transmitir 46 bytes de datos de usuario (la carga útil de una trama de 64 bytes) tiene una eficiencia de línea de 9%.

La segunda característica, llamada *ráfagas de trama*, permite que un emisor transmita una secuencia concatenada de múltiples tramas en una sola transmisión. Si la ráfaga total es menor que 512 bytes, el hardware la rellena nuevamente. Si suficientes tramas están esperando la transmisión, este esquema es muy eficiente y se prefiere antes que la extensión de portadora. Estas nuevas características amplían el radio de red de 200 metros, que probablemente es suficiente para la mayoría de las oficinas.

Cableado de Gigabit Ethernet:



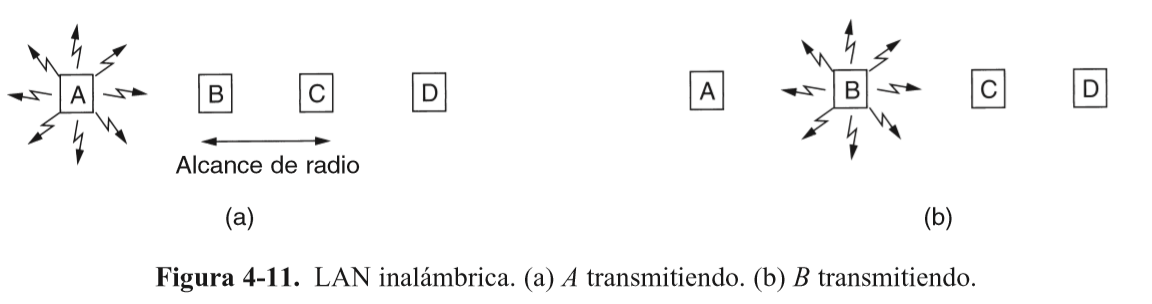
La opción **1000Base-CX** utiliza cables de cobre blindados cortos. Su problema es que compite con la fibra de alto desempeño por una parte, y con el UTP más económico por la otra. No es probable que se utilice mucho, si es que se utiliza.

La última opción son paquetes de cuatro cables UTP categoría 5 que trabajan juntos. Debido a que la mayor parte de este cableado ya está instalada, es probable que sea la Gigabit Ethernet de la mayoría de las personas/empresas que no hagan grandes inversiones.

**Protocolos de LANs inalámbricas**

Es importante darse cuenta de que en algunas LANs inalámbricas no todas las estaciones están dentro del alcance de todas las demás, lo que conduce a una variedad de complicaciones.

Un enfoque para usar una LAN inalámbrica podría ser intentar el CSMA. El problema radica en que este protocolo no es realmente adecuado porque lo que importa es la interferencia en el receptor, no en el emisor. Para ver la naturaleza de este problema, considere la figura 4-11, en la que se ilustran cuatro estaciones inalámbricas. Para nuestros fines, no importa cuáles son estaciones base ni cuáles son portátiles. El alcance de radio es tal que A y B están en el mismo alcance y potencialmente pueden interferir entre sí. C también podría interferir tanto con B como con D, pero no con A.



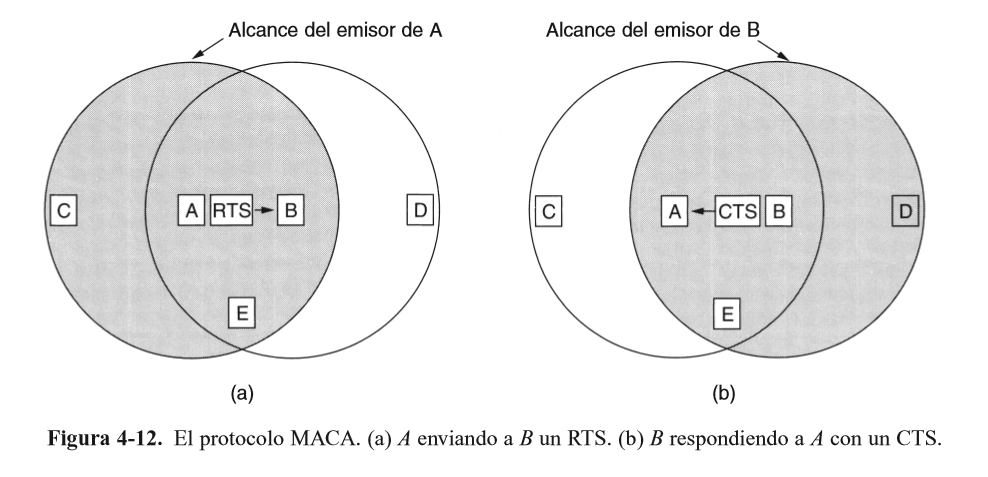
Primero considere lo que ocurre cuando A está transmitiendo hacia B, como se muestra en la figura 4-11(a). Si C detecta el medio, no podrá escuchar a A porque está fuera de su alcance y, por tanto, deducirá falsamente que puede transmitir a B. Si C comienza a transmitir, interferirá en B, eliminando la trama de A. El problema de que una estación no puede detectar a un competidor potencial por el medio, puesto que dicho competidor está demasiado lejos, se denomina **problema de estación oculta**.

Ahora consideremos la situación inversa: B transmitiendo a A, como se muestra en la figura 4-11(b). Si C detecta el medio, escuchará una transmisión y concluirá equivocadamente que no puede enviar a D, cuando de hecho tal transmisión causaría una mala recepción sólo en la zona entre B y C, en la que no está localizado ninguno de los receptores pretendidos. Esta situación se denomina **problema de estación expuesta**.

El problema es que antes de comenzar una transmisión, una estación realmente necesita saber si hay actividad o no alrededor del receptor. El CSMA simplemente le indica si hay o no actividad alrededor de la estación que está detectando la portadora.

**MACA y MACAW**

MACA (Acceso Múltiple con Prevención de Colisiones) es uno de los primeros protocolos diseñados para LANs inalámbricas. Se basa en que el emisor estimule al receptor a enviar una trama corta, de manera que las estaciones cercanas puedan detectar esta transmisión y eviten ellas mismas hacerlo durante la siguiente trama de datos (grande).



Consideremos ahora la manera en que A envía una trama a B. A comienza por enviar una trama RTS (Solicitud de Envío) a B, como se muestra en la figura 4-12(a). Esta trama corta (30 bytes) contiene la longitud de la trama de datos que seguirá posteriormente. Después B contesta con una trama CTS (Libre para Envío), como se muestra en la figura 4-12(b). La trama CTS contiene la longitud de los datos (copiada de la trama RTS). Una vez que sucede la recepción de la trama CTS, A comienza a transmitir.

Ahora veremos cómo reaccionan las estaciones que escuchan cualquiera de estas tramas. Cualquier estación que escuche el RTS evidentemente está bastante cerca de A y debe permanecer en silencio durante el tiempo suficiente para que el CTS se transmita de regreso a A sin conflicto. Cualquier estación que escuche el CTS está bastante cerca de B y debe permanecer en silencio durante la siguiente transmisión de datos, cuya longitud puede determinar examinando la trama CTS.

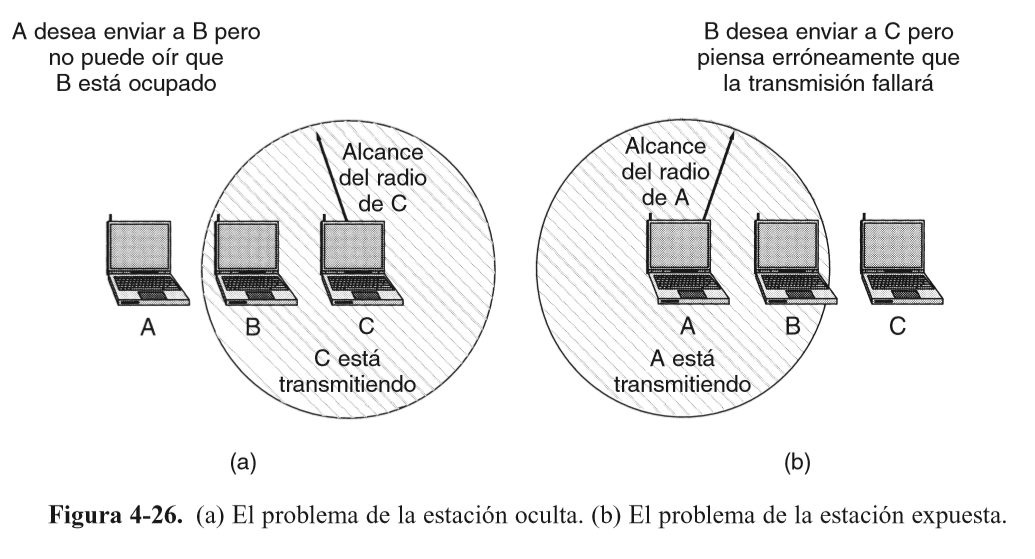
En la figura 4-12, C está en el alcance de A, pero no en el de B. Por lo tanto, escucha el RTS de A pero no el CTS de B. Mientras no interfiera con el CTS, está libre para transmitir mientras se está enviando la trama de datos. En contraste, D está en el alcance de B pero no de A. No escucha el RTS pero sí el CTS. Al escuchar el CTS se le indica que está cerca de una estación que está a punto de recibir una trama, por lo que difiere el envío de cualquier cosa hasta el momento en que se espera la terminación de esa trama. La estación E escucha ambos mensajes de control y, al igual que D, debe permanecer en silencio hasta que se haya completado la trama de datos.

Con base en estudios de simulación del MACA, se afinó para mejorar su desempeño y se llamó MACAW (MACA Inalámbrico) al nuevo protocolo. Se notó que, sin confirmación de recepción de la capa de enlace de datos, las tramas no eran retransmitidas sino hasta que la capa de transporte notaba su ausencia, mucho después. Se resolvió este problema introduciendo una trama ACK tras cada trama de datos exitosa. También se observó que CSMA puede servir para evitar que una estación transmita un RTS al mismo tiempo y destino que otra estación cercana, por lo que se agregó la detección de portadora.

**El protocolo de la subcapa MAC del 802.11**

Con Ethernet, una estación simplemente espera hasta que el medio queda en silencio y comienza a transmitir. Si no recibe una ráfaga de ruido dentro de los primeros 64 bytes, con seguridad la trama ha sido entregada correctamente. Esta situación no es válida para los sistemas inalámbricos.

Para empezar, existe el problema de la estación oculta mencionado con anterioridad, el cual se ilustra nuevamente en la figura 4-26(a). Puesto que no todas las estaciones están dentro del alcance de radio de cada una, las transmisiones que van en un lado de una celda podrían no recibirse en otro lado de la misma celda. En este ejemplo, la estación C transmite a la estación B. Si A detecta el canal, no escuchará nada y concluirá erróneamente que ahora puede comenzar a transmitir a B.

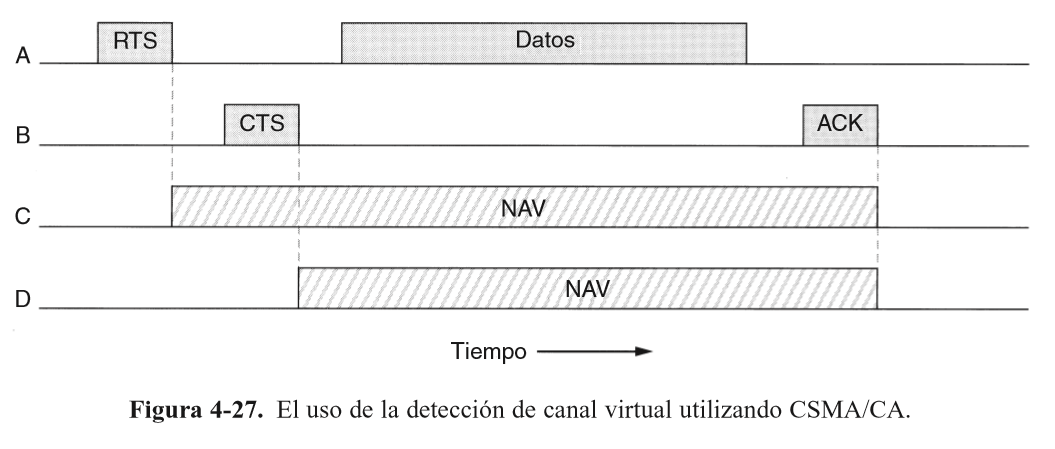


Además, existe el problema inverso, el de la estación expuesta, que se ilustra en la figura 4-26(b). Aquí B desea enviar a C por lo que escucha el canal. Cuando escucha una transmisión, concluye erróneamente que no debería transmitir a C, aunque A esté transmitiendo a D (lo cual no se muestra). Además, la mayoría de los radios son semidúplex, lo que significa que no pueden transmitir y escuchar ráfagas de ruido al mismo tiempo en una sola frecuencia. Como resultado de estos problemas, 802.11 no utiliza CSMA/CD, como lo hace Ethernet.

Para solucionar este problema, 802.11 suporta dos modos de funcionamiento. El primero, llamado **DCF (Función de Coordinación Distribuida)**, no utiliza ningún tipo de control central (en ese aspecto, es similar a Ethernet). El otro, llamado **PCF (Función de Coordinación Puntual)**, utiliza la estación base para controlar toda la actividad en su celda. Todas las implementaciones soportan DCF pero PCF es opcional. A continuación analizaremos estos dos modos a la vez.

Cuando se emplea DCF, 802.11 utiliza un protocolo llamado **CSMA/CA (CSMA con Evitación de Colisiones)**. En este protocolo, se utiliza tanto la detección del canal físico como la del canal virtual. Los dos métodos de funcionamiento son soportados por CSMA/CA. En el primer método, cuando una estación desea transmitir, detecta el canal. Si está inactivo, comienza a transmitir. No detecta el canal mientras transmite pero emite su trama completa, la cual podría ser destruida en el receptor debido a interferencia. Si el canal está ocupado, el emisor espera hasta que esté inactivo para comenzar a transmitir. Si ocurre una colisión, las estaciones involucradas en ella esperan un tiempo aleatorio, mediante el algoritmo de retroceso exponencial binario de Ethernet, y vuelve a intentarlo más tarde.

El otro modo de la operación CSMA/CA se basa en MACAW y utiliza la detección de canal virtual, como se ilustra en la figura 4-27. En este ejemplo, A desea enviar a B. C es una estación que está dentro del alcance de A (y posiblemente dentro del alcance de B, pero eso no importa). D es una estación dentro del alcance de B pero no dentro del de A.

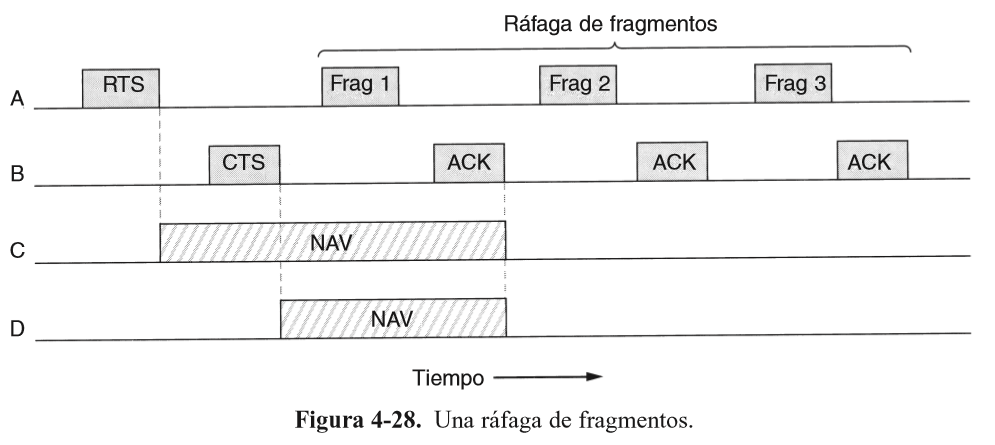


El protocolo inicia cuando A decide enviar datos a B. A inicia enviándole una trama RTS a B en la que le solicita permiso para enviarle una trama. Cuando B recibe esta solicitud, podría decidir otorgarle el permiso, en cuyo caso le regresa una trama CTS. Al recibir la CTS, A ahora envía su trama y comienza su temporizador de ACK. Al recibir correctamente la trama de datos, B responde con una trama de ACK, con lo que termina el intercambio. Si el temporizador de ACK de A termina antes de que el ACK regrese, todo el protocolo se ejecuta de nuevo.

Ahora consideremos este intercambio desde el punto de vista de C y D. C está dentro del alcance de A, por lo que podría recibir la trama RTS. Si pasa esto, se da cuenta de que alguien va a enviar datos pronto, así que por el bien de todos desiste de transmitir cualquier cosa hasta que el intercambio esté completo. A partir de la información proporcionada en la solicitud RTS, C puede estimar cuánto tardará la secuencia, incluyendo el ACK final, por lo que impone para sí misma un tipo de canal virtual ocupado, indicado por **NAV (Vector de Asignación de Red)** en la figura 4-27. D no escucha el RTS, pero sí el CTS, por lo que también impone la señal NAV para sí misma. Observe que las señales NAV no se transmiten; simplemente son recordatorios internos para mantenerse en silencio durante cierto periodo.

En contraste con las redes cableadas, las inalámbricas son ruidosas e inestables, en gran parte debido a los hornos de microondas, que también utilizan las bandas sin licencia ISM. Como consecuencia, la probabilidad de que una trama llegue a su destino se decrementa con la longitud de la trama.

Para solucionar el problema de los canales ruidosos, 802.11 permite dividir las tramas en fragmentos, cada uno con su propia suma de verificación. Cada fragmento se numera de manera individual y su recepción se confirma utilizando un protocolo de parada y espera (es decir, el emisor podría no transmitir fragmentos de k + 1 hasta que haya recibido la confirmación de recepción del fragmento k). Una vez que se ha adquirido el canal mediante RTS y CTS, pueden enviarse múltiples fragmentos en una fila, como se muestra en la figura 4-28. La secuencia de fragmentos se conoce como **ráfaga de fragmentos**.

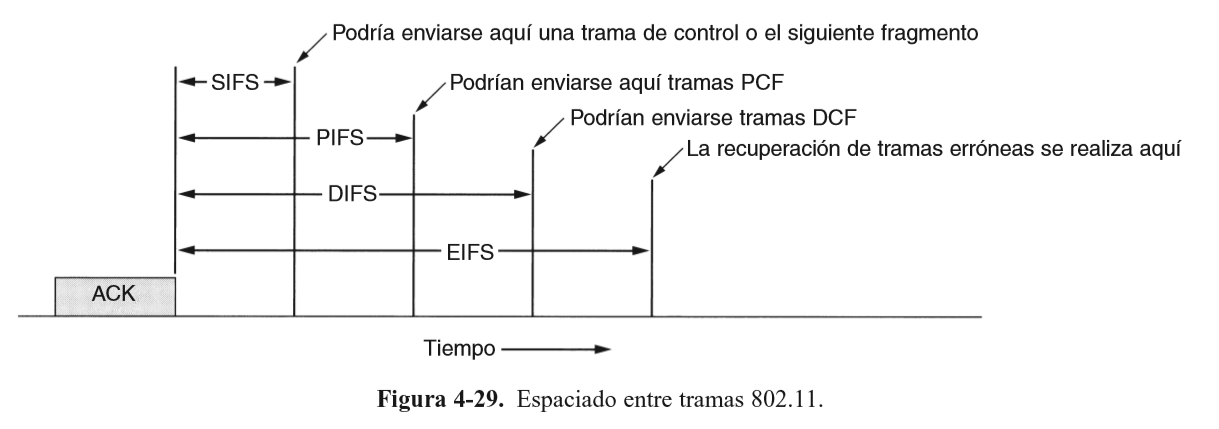


La fragmentación incrementa la velocidad real de transporte restringiendo las retransmisiones a los fragmentos erróneos en lugar de la trama completa. El tamaño del fragmento no lo fija el estándar pero es un parámetro de cada celda y la estación base puede ajustarlo. El mecanismo NAV mantiene otras estaciones en silencio sólo hasta la siguiente confirmación de recepción, pero se utiliza otro mecanismo (descrito a continuación) para permitir que otra ráfaga de fragmentos completa se envíe sin interferencia.

Todo el análisis anterior se aplica al modo DCF 802.11. En él, no hay control central y la estación compite por tiempo aire, como en Ethernet. El otro modo permitido es PCF, en el que la estación base sondea las demás estaciones, preguntándoles si tienen tramas que enviar. Puesto que el orden de transmisión se controla por completo por la estación base en el modo PCF, no ocurren colisiones.

El mecanismo básico consiste en que la estación base difunda una trama de beacon (trama guía o faro) de manera periódica (de 10 a 100 veces por segundo). Esta trama contiene parámetros de sistema. También invita a las nuevas estaciones a suscribirse al servicio de sondeo. Una vez que una estación se inscribe para el servicio de sondeo a cierta tasa, se le garantiza de manera efectiva cierta fracción de ancho de banda, y se hace posible proporcionar garantías de calidad de servicio.

PCF y DCF pueden coexistir dentro de una celda. Al principio podría parecer imposible tener control central y distribuido funcionando al mismo tiempo, pero 802.11 proporciona una forma de alcanzar este objetivo. Funciona definiendo cuidadosamente el intervalo de tiempo entre tramas. Después de que se ha enviado una trama, se necesita cierta cantidad de tiempo muerto antes de que cualquier estación pueda enviar una trama. Se definen cuatro intervalos diferentes, cada uno con un propósito específico. Estos intervalos se describen en la figura 4-29.



El intervalo más corto es **SIFS (Espaciado Corto Entre Tramas)**. Se utiliza para permitir que las distintas partes de un diálogo transmitan primero. Esto incluye dejar que el receptor envíe un CTS para responder a una RTS, dejar que el receptor envíe un ACK para un fragmento o una trama con todos los datos y dejar que el emisor de una ráfaga de fragmentos transmita el siguiente fragmento sin tener que enviar una RTS nuevamente.

Siempre hay una sola estación que debe responder después de un intervalo SIFS. Si falla al utilizar su oportunidad y transcurre un tiempo **PIFS (Espaciado Entre Tramas PCF)**, la estación base podría enviar una trama de beacon o una trama de sondeo. Este mecanismo permite que una estación base envíe una trama de datos o una secuencia de fragmentos para finalizar su trama sin que nadie interfiera, pero le da a la estación base la oportunidad de tomar el canal cuando el emisor anterior haya terminado, sin tener que competir con usuarios ansiosos.

Si la estación base no tiene nada que decir y transcurre un tiempo **DIFS (Espaciado Entre Tramas DCF)**, cualquier estación podría intentar adquirir el canal para enviar una nueva trama. Se aplican las reglas de contención normales, y si ocurre una colisión, podría necesitarse el retroceso exponencial binario.

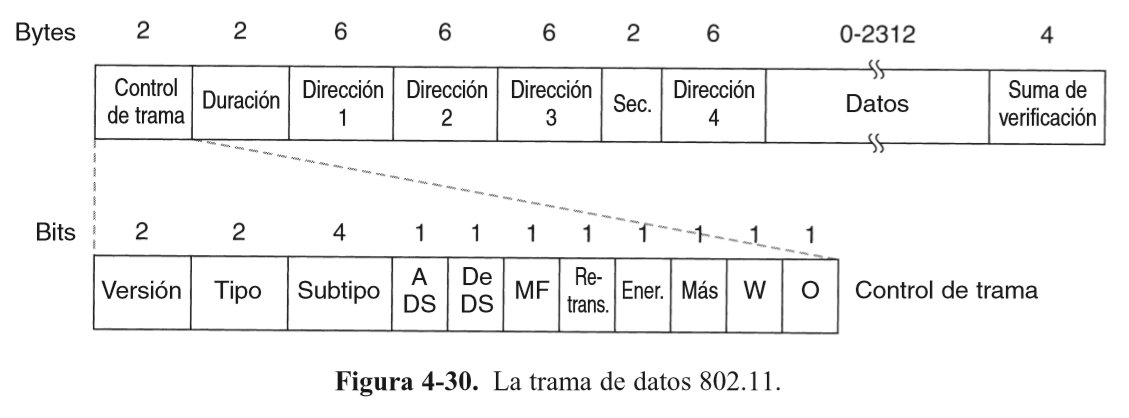
Sólo una estación que acaba de recibir una trama errónea o desconocida utiliza el último intervalo de tiempo, **EIFS (Espaciado Entre Tramas Extendido)**, para reportar la trama errónea. La idea de dar a este evento la menor prioridad es que debido a que el receptor tal vez no tenga idea de lo que está pasando, debe esperar un tiempo considerable para evitar interferir con un diálogo en curso entre las dos estaciones.

**La estructura de trama 802.11**

El estándar 802.11 define tres clases diferentes de tramas en el cable: de datos, de control y de administración. Cada una de ellas tiene un encabezado con una variedad de campos utilizados dentro de la subcapa MAC.

En la figura 4-30 se muestra el formato de la trama de datos. Primero está el campo de *Control de trama*. Éste tiene 11 subcampos:

* *Versión de protocolo*: que permite que dos versiones del protocolo funcionen al mismo tiempo en la misma celda.
* *Tipo* (de datos, de control o de administración)
* *Subtipo* (por ejemplo, RTS o CTS)
* Los bits *A DS* y *De DS* indican que la trama va hacia o viene del sistema de distribución entre celdas (por ejemplo, Ethernet).
* El bit *MF* indica que siguen más fragmentos.
* El bit *Retrans* marca una retransmisión de una trama que se envió anteriormente.
* El bit de *Administración de energía* es utilizado por la estación base para poner al receptor en estado de hibernación o sacarlo de tal estado.
* El bit *Más* indica que el emisor tiene tramas adicionales para el receptor.
* El bit *W* especifica que el cuerpo de la trama se ha codificado utilizando el algoritmo **WEP (Privacidad Inalámbrica Equivalente)**.
* El bit *O* indica al receptor que una secuencia de tramas que tenga este bit encendido debe procesarse en orden estricto.



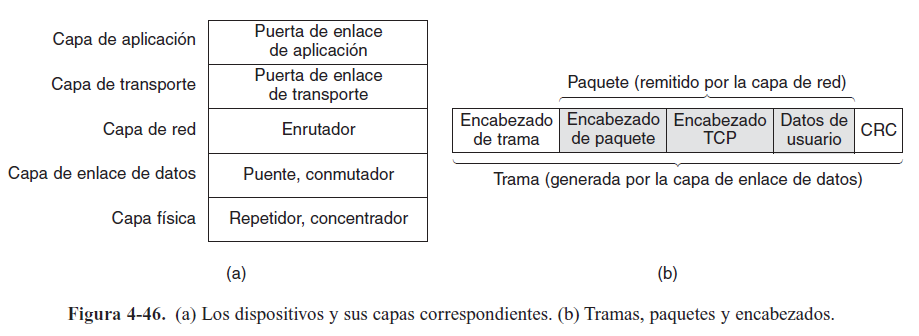
El segundo campo de la trama de datos, el de *Duración*, indica cuánto tiempo ocuparán el canal la trama y su confirmación de recepción. Este campo también está presente en las tramas de control y es la forma mediante la cual otras estaciones manejan el mecanismo NAV. El encabezado de trama contiene cuatro direcciones, todas en formato estándar IEEE 802. Obviamente se necesitan el origen y el destino. Las otras dos direcciones se utilizan para las estaciones base de origen y destino para el tráfico entre celdas.

El campo de *Secuencia* permite que se numeren los fragmentos. El campo de *Datos* contiene la carga útil, hasta 2312 bytes, y le sigue el campo común de *Suma de verificación*.

Las tramas de administración tienen un formato similar al de las tramas de datos, excepto que no tienen una de las direcciones de la estación base, debido a que las tramas de administración se restringen a una sola celda. Las tramas de control son más cortas; tienen una o dos direcciones, y no tienen ni campo de Datos ni de Secuencia. La información clave aquí se encuentra en el campo de Subtipo, que por lo general es RTS, CTS o ACK.

**DISPOSITIVOS - Repetidores, concentradores, puentes, conmutadores, enrutadores y puertas de enlace**

Para empezar, estos dispositivos operan en diferentes capas, como se muestra en la figura 4-46(a). La capa es importante porque los distintos dispositivos utilizan diferentes partes de información para decidir su modo de operación. En un escenario común, el usuario genera algunos datos que se enviarán a una máquina remota. Estos datos se pasan a la capa de transporte, que le agrega un encabezado, por ejemplo, un encabezado TCP, y pasa la unidad que resulta a la capa de red. Ésta incorpora su propio encabezado para obtener un paquete de capa de red, por ejemplo, un paquete IP. En la figura 4-46(b) podemos ver el paquete IP con un sombreado gris. A continuación, el paquete pasa a la capa de enlace de datos, que incorpora su propio encabezado y suma de verificación (CRC) y envía la trama resultante a la capa física para que desde ahí sea transmitida, por ejemplo, sobre una LAN.



En la capa física, se encuentran los ***repetidores***. Éstos son dispositivos análogos conectados a dos segmentos de cable. Una señal que aparece en uno de ellos es amplificada y enviada al otro. Los repetidores no distinguen entre tramas, paquetes o encabezados, simplemente manejan voltios.

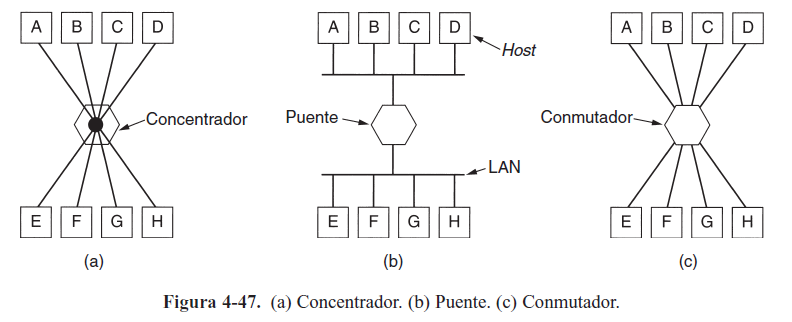
Pasemos ahora a los concentradores. Un ***concentrador*** tiene numerosos puertos de entrada que une de manera eléctrica. Las tramas que llegan a cualquiera de las líneas se envían a todas las demás. Si dos tramas llegan al mismo tiempo, chocarán, al igual que en un cable coaxial. En otras palabras, el concentrador constituye un solo dominio de colisión. Todas las líneas que convergen en un concentrador deben operar a la misma velocidad. Al igual que los repetidores, los concentradores no examinan las direcciones 802 ni las utilizan de ninguna manera.

Veamos a continuación la capa de enlace de datos donde operan los puentes y los conmutadores. Un ***puente*** conecta dos o más LANs, como se puede ver en la figura 4-47(b). Cuando llega una trama, el software del puente extrae la dirección de destino del encabezado y la busca en una tabla para averiguar a dónde debe enviar la trama. Un puente moderno podría tener tarjetas de línea para diferentes tipos de red y diferentes velocidades. En contraste con un concentrador, en un puente cada puerto constituye su propio dominio de colisión.

Los ***conmutadores*** son similares a los puentes en el aspecto de que ambos enrutan tomando como base las direcciones de las tramas. La principal diferencia consiste en que un conmutador se utiliza con mayor frecuencia para conectar computadoras individuales, como se puede ver en la figura 4-47(c). En consecuencia, cuando el host A de la figura 4-47(b) desea enviar una trama al host B, el puente toma la trama pero la descarta. Por el contrario, en la figura 4-47(c), el conmutador debe reenviar activamente la trama de A a B porque no existe otra forma para que ésta llegue ahí.

Ahora pasaremos a los ***enrutadores***, que son diferentes de todos los anteriores. Cuando un paquete llega a un enrutador, el encabezado y el terminador de la trama se eliminan y el paquete contenido en el campo de carga útil de la trama se pasa al software de enrutamiento. Este software se vale del encabezado del paquete para elegir un puerto de salida. En un paquete IP, el encabezado contendrá una dirección de 32 bits (IPv4) o 128 bits (IPv6), no una dirección 802 de 48 bits.

Una capa más arriba encontramos ***puertas de enlace*** de transporte. Estos dispositivos conectan dos computadoras que utilizan diferentes protocolos de transporte orientados a la conexión. Por ejemplo, imagine que una computadora que utiliza el protocolo TCP/IP orientado a la conexión necesita comunicarse con una computadora que emplea el protocolo de transporte ATM, también orientado a la conexión. La puerta de enlace de transporte puede copiar los paquetes de una conexión a la otra y darles el formato que necesiten.



**Capa 3: Capa de Red**

Esta capa se encarga de comunicar distintas redes. No se basa únicamente en host sino también en direcciones que indican dónde se encuentran esos nodos (hosts). Además determina si el destino está en la red y sino, en qué dirección está. Para saber a dónde debe ir existen algoritmos de enrutamiento.

Esta capa es independiente de la tecnología que se utiliza en las capas inferiores (es la primer capa que trabaja con software) pero, debe poder adaptarse a distintas tecnologías. Es la única capa que puede desplazarse en distintos escenarios.

Tiene 2 funciones:

1. *Fragmentación:* permite dividir el paquete en distintos tamaños y mandarlos por distintas tecnologías (varía el tamaño de la trama).
2. *Genera túneles:* sirve para conectar 2 puntos a través de un medio donde el que quiere ir de un lado a otro, no podría. En capa 3 sirve para enviar información encapsulada con la finalidad de ocultar la información original o pasar por una red que no soporta el protocolo original.

Servicios que proporciona la capa de red a la capa de transporte:

1. Los servicios deben ser independientes de la tecnología del enrutador.
2. La capa de transporte debe estar aislada de la cantidad, tipo y topología de los enrutadores presentes.
3. Las direcciones de red disponibles para la capa de transporte deben seguir un plan de numeración uniforme, aun a través de varias LANs y WANs.

Objetivos de la capa de red:

1. Llevar paquetes del origen al destino.
2. Conocer la topología de la subred.
3. Escoger las trayectorias adecuadas.
4. Evitar la sobrecarga de las líneas y los enrutadores.

Organización interna

Se pueden realizar dos formas de organización distintas, dependiendo del tipo de servicio que se ofrezca. Si se ofrece el servicio no orientado a la conexión, los paquetes se colocan individualmente en la subred y se enrutan de manera independiente. No se necesita una configuración avanzada. En este contexto, por lo general los paquetes se conocen como ***datagramas*** (en analogía con los telegramas) y la subred se conoce como *subred de datagramas*. En este modo cada paquete elige la ruta por donde va a comunicarse. Varios paquetes de una misma transmisión van a ir por caminos separados.

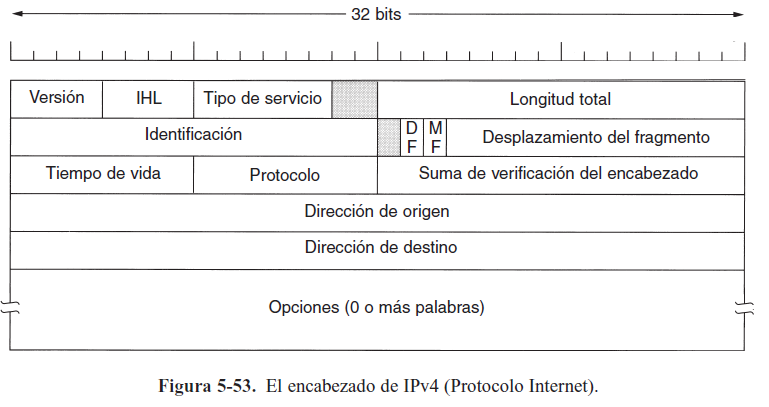
Si se utiliza el servicio orientado a la conexión, antes de poder enviar cualquier paquete de datos, es necesario establecer una ruta del enrutador de origen al de destino. Esta conexión se conoce como ***CV (circuito virtual)***, en analogía con los circuitos físicos establecidos por el sistema telefónico, y la subred se conoce como *subred de circuitos virtuales*. En este modo se elige la ruta a seguir con el primer paquete enviado, el resto de los paquetes van a utilizar el mismo camino que el inicial. Para eso, cada enrutador identifica el circuito virtual y reserva los recursos necesarios para el resto de la transmisión.

|  |  |  |
| --- | --- | --- |
|  | DATAGRAMAS | CV (circuito virtual) |
| Establecer conexión | No | Si |
| Direccionar | En cada datagrama o paquete debe tener dos direcciones: dirección origen y destino | No necesita porque cada paquete lleva el número de circuito a utilizar |
| Informar estado | No (no importa el orden) | Si (para saber el estado de circuito) |
| Enrutar | Si, se enrutan por separado | Si, se enrutan al generar el circuito |
| Efecto ante fallo | Se puede llegar a perder algún paquete pero se puede reenviar, prácticamente no afecta | Se corta el circuito, se debe volver a establecer conexión y reenviar todos los paquetes. |
| Dirección destino | Situada en cada paquete | En la inicialización del circuito únicamente. |
| Manejo de error | Los nodos terminales se encargan de verificar la conexión (origen y destino) | Cada nodo que está dentro del circuito virtual tiene la capacidad de control de errores. |
| Control de flujo | No | Sí, tienen que llegar en orden. |
| Secuenciamiento | No hace falta | Si es necesario |
| Iniciación | No | Si hay que determinar una inicialización |

**El Protocolo IP**

Es uno de los protocolos disponibles que utiliza TCP/IP. Se encarga del *encaminamiento* (dónde está) y del *direccionamiento* (cómo identificarlo).

Un lugar adecuado para comenzar nuestro estudio de la capa de red de Internet es el formato de los datagramas de IP mismos. Un datagrama IP consiste en una parte de encabezado y una parte de texto. El encabezado tiene una parte fija de 20 bytes y una parte opcional de longitud variable. El formato del encabezado se muestra en la figura 5-53.



* *Versión: son 4 bits que* llevan el registro de la versión del protocolo al que pertenece el datagrama (IPv4 – IPv6).
* *IHL:* dado que la longitud del encabezado no es constante (las opciones son de longitud variable), se incluye este campo en el encabezado.
* *Tipo de Servicio:* son 8 bits cuyo propósito es distinguir entre las diferentes clases de servicios. Son posibles varias combinaciones de confiabilidad y velocidad. Permite especificar, por ejemplo, retardo (delay), velocidad real de transporte, confiabilidad.
* *Longitud Total:* son 16 bits e incluye todo el datagrama, tanto el encabezado como los datos.
* *Identificación:* es necesario para que el host de destino determine a qué datagrama (o paquete) pertenece un fragmento recién llegado. Todos los fragmentos de un datagrama contienen el mismo valor de Identificación.
* *DF:* 1 bit. Significa no fragmentar (Don’t Fragment); es una orden para los enrutadores de que no fragmenten el datagrama, porque el destino es incapaz de juntar las piezas de nuevo.
* *MF:* 1 bit. Significa más fragmentos. Todos los fragmentos excepto el último tienen establecido este bit, que es necesario para saber cuándo han llegado todos los fragmentos de un datagrama.
* *Desplazamiento del Fragmento (Offset):* indica en qué parte del datagrama actual va este fragmento. Todos los fragmentos excepto el último del datagrama deben tener un múltiplo de 8 bytes, que es la unidad de fragmentos elemental.
* *Tiempo de Vida (TTL):* es un contador que sirve para limitar la vida de un paquete. En la práctica, simplemente cuenta los saltos. Cuando el contador llega a cero, el paquete se descarta y se envía de regreso un paquete de aviso al host de origen. Esta característica evita que los datagramas vaguen eternamente, algo que de otra manera podría ocurrir si se llegan a corromper las tablas de enrutamiento.
* *Protocolo:* son 8 bits que indican el protocolo de las capas superiores al que debe entregarse el paquete (TCP o UDP, por ejemplo).
* *Suma de Verificación:* verifica solamente el encabezado. Tal suma de verificación es útil para la detección de errores generados por palabras de memoria erróneas en un enrutador. Debe recalcularse en cada salto porque al menos uno de los campos cambia (tiempo de vida, por ejemplo).
* *Direcciones Origen y Destino:* indican el número de red y el número de host. Son direcciones IP de 32 bits.
* *Opciones:* se diseñó para proporcionar un recurso que permitiera que las versiones subsiguientes del protocolo -incluyeran información no presente en el diseño original, para permitir que los experimentadores prueben ideas nuevas y para evitar la asignación de bits de encabezado a información pocas veces necesaria. Las opciones son de longitud variable pero siempre rellenándolas de a 32 bits (múltiplo de 4 bytes). Pueden ser opciones de seguridad, enrutamiento, codificación, etc.

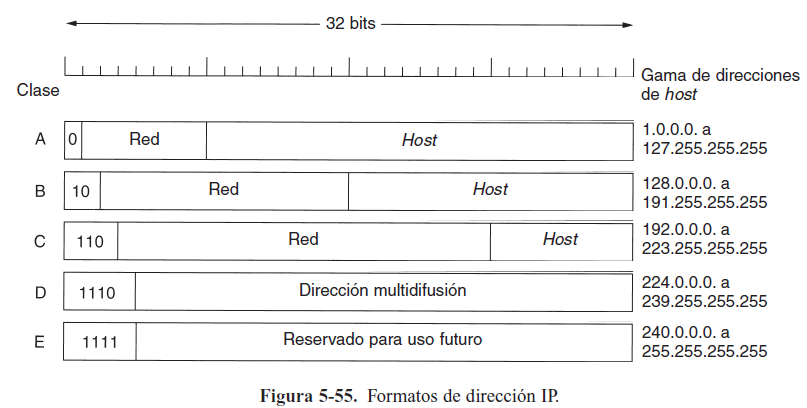
**Direcciones IP**

Cada host y enrutador de Internet tiene una dirección IP, que codifica su número de red y su número de host. La combinación es única: no hay dos máquinas que tengan la misma dirección IP (solo posee 1 y 0).

Cabe aclarar que una IP por sí sola no determina nada, siempre debe estar acompañada por la *máscara de red* (indica qué parte de la IP corresponde a red y qué parte a un host, además de permitir calcular direcciones de red y broadcast asociadas). Una máscara, para ser considerada válida debe estar formado por unos y luego ceros, no es válida aquella máscara que presente 0 y 1 alternados.

Todas las direcciones IP son de 32 bits de longitud y se usan en los campos de Dirección de origen y de Dirección de destino de los paquetes IP. En importante mencionar que una dirección IP realmente no se refiere a un host. En realidad se refiere a una interfaz de red, por lo que si un host está en dos redes, debe tener dos direcciones IP. Sin embargo, en la práctica, la mayoría de los hosts se encuentran en una red y, por lo tanto, tienen una dirección IP.

Las direcciones IP se dividieron en cinco categorías, las cuales se listan en la figura 5-55. Esta asignación se ha llamado **direccionamiento con clase** (classful addressing).



Redes privadas (fuera de internet) para cada clase

* A 10.0.0.0 a 10.255.255.255/8
* B 172.16.0.0 a 172.31.255.255/12
* C 192.168.0.0 a 192.168.255.255/16

En la actualidad no hay más direcciones IPv4 en la red. Solo quedan las direcciones privadas y por eso es que surge IPv6. Además, generalmente, se utiliza la clase A o C.

**Mascara:**

Ejemplo: 255.255.255.0

11111111.11111111.11111111.00000000 (parte binaria de esta mascara)

/24 (cantidad de 1 en la máscara)

Nos va a determinar que parte es de red y que parte de host. Obligatoriamente está formada por 1 y 0. Mediante la IP y la máscara se desprenden dos operaciones para saber que parte es red, que parte de host y que parte de brocadast.

Con el ejemplo: IP: 192.168.0.1 MASK: 255.255.255.0

***IP AND MASK*** = DIRECCION DE RED 🡪 11000000.10101000.00000000.00000000

11111111.11111111.11111111.00000000

-------------------------------------------------------

11000000.10101000.00000000.00000000

( 192 . 168 . 0 . 0 )

La operación recientemente realizada consiste en una operación *AND* entre la dirección IP y la máscara de red. Eso da como resultado la dirección de red.

***IP V¬ MASK*** = DIR BROADCAST 🡪 11000000.10101000.00000000.00000000

00000000.00000000.00000000.11111111

-------------------------------------------------------

11000000.10101000.00000000.11111111

( 192 . 168 . 0 . 255 )

La operación ahora consiste en la operación lógica *OR* entre la dirección IP y la máscara pero negada. El resultado es la dirección de broadcast.

**Puerta de Enlace:** es la salida hacia otras redes. Para saber si está en mi misma red se toma la IP and MASK si tiene mi misma dirección de red puedo enviarlo, si no lo envío al enrutador.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ip | Mask | Dir red | | Dir broadcast | |
| 10.0.10.15 | 255.255.255.0 | 10.0.10.0 | | 10.0.10.255 | |
| 192.168.10.220 | 255.255.0.0 | 192.168.0.0 | | 192.168.255.255 | |
| 172.145.15.178 | 255.255.255.0 | | 172.145.15.0 | 172.145.15.255 | |
| 10.0.0.45 | 255.0.0.0 | | 10.0.0.0 | 10.255.255.255 |

Notas: el rango de IP disponibles para hosts siempre va desde la dirección de red a la dirección de broadcast sin incluirlas. Ejemplo: dirección de red 10.0.10.0; dirección de broadcast 10.0.10.255; rango de IPs: 10.0.10.1 – 10.0.10.254

Para saber de qué clase es una dirección IP tendremos que fijarnos en lo siguiente: el primer octeto de esta. Mejor dicho en los 2 bits de mayor peso del primer octeto.

* Si el bit con valor 128 es 0 y el bit con valor 64 es 0, la dirección IP es de clase A.
* Si el bit con valor 128 es 1 y el bit con valor 64 es 0, la dirección IP es de clase B.
* Si el bit con valor 128 es 1 y el bit con valor 64 es 1, la dirección IP es de clase C.

Resumiendo:

* Clase A: 00 es decir ( (128\*0) + (64\*0) ) = 0
* Clase B: 10 es decir ( (128\*1) + (64\*0) ) = 128
* Clase C: 11 es decir ( (128\*1) + (64\*1) ) = 192

Por ejemplo, la dirección IP 192.168.0.1 en binario es 11000000.168.0.1. En base a esto, podremos deducir que la dirección IP 192.168.0.1 es de clase C. En cambio, la dirección IP 140.1.10.5 es de clase B.

La máscara de red por defecto de una dirección IP de clase C, por ejemplo, nos dice que solo el último octeto está destinado a hosts. Esos 8 números del último octeto, equivalen a 256 números posibles (). Pero solo 254 son posibles de utilizar para hosts. Esto se debe a que, de esos 256 números posibles, hay 2 que son reservados: El 192.168.0.0 y el 192.168.0.255 son las direcciones de red y broadcast, respectivamente. Entonces,

siendo la cantidad de bits destinados a host y la cantidad de números asignables a hosts.

**Subredes**

Permiten la división interna de una red privada. Dicho de otro modo, una subred es una dirección lógica dentro una red.

Ejemplo:

Tenemos las siguientes direcciones preestablecidas:

Dirección de Red 192.168.0.0 Máscara de Red 255.255.255.0

Para generar una nueva máscara para las subredes lo que se hace es agregar un 1 más a la máscara original de la red (obtengo 2 subredes). La cantidad de subredes que se pueden generar se determina mediante:

siendo la cantidad de 1 que agrego, por otro lado, es la cantidad de subredes que obtengo.

Tips generales:

* Cantidad de subredes útiles =
* Cantidad de hosts útiles para cada subred = (“bits de host” hace referencia a la cantidad de 0 que quedan en la máscara de subred, luego de agregar los 1 correspondientes para formar la misma).
* La dirección IP del primer host útil de cada subred =
* La dirección IP del último host útil de cada subred =
* Cuando tenemos dirección de red y broadcast y queremos calcular la máscara de red debemos analizar lo siguiente: cuando los octetos correspondientes de la dirección de red y broadcast son iguales, es porque en esos octetos la máscara forma 255. Ejemplo: red 192.168 y broadcast 192.168, entonces la máscara en esos octetos es 255.255.
* Para calcular rápidamente una máscara de red a partir de la dirección de red y broadcast podemos hacer lo siguiente: red 192.**88**.0.0 y broadcast 192.**91**.255.255; como dijimos anteriormente, el primer octeto será 255. Luego para determinar el siguiente octeto hacemos 91 – 88 = 4 (porque sería 3 + 1, el cual representa al 0) 256 – 4 = 252. Par finalizar tenemos la máscara, 255.**252**.0.0
* Otra forma de calcular lo anterior es, hacer 91 – 88 = 4 (porque sería 3 + 1, el cual representa al 0) para calcular el salto. Luego hacemos , posteriormente debemos calcular a qué número deberíamos elevar al 2 para obtener 64, para ello hacemos . Por último, ese 6 representa la cantidad de 1 que se agregan en ese octeto.
* La máscara 255.255.255.255 o /32 sirve para identificar un host específico dentro una misma red.

Otros protocolos de capa 3

* **Protocolo ICMP (internet Control Message Protocol)**

Son varios mensajes para realizar control en la subred. Cada uno de los mensajes tiene una respuesta esperada. El que usa el comando *ping* es el mensaje ECHO, que lo que hace es preguntarle a cierta dirección IP si esta activa, si esa dirección está activa responde con el ECHO reply (mensaje de respuesta). Cuando no encuentra destino devuelve DESTINATION UNRACHEABLE. TIME EXCEEDED (tiempo excedido) se devuelve cuando se termina el TTL.

El TRACE ROUTE devuelve todos los routers por los que pasa (funciona mandando el TIME STAMP con el TTL desde 1 y aumentando de a uno, así va armando la ruta de todos los routers por los que pasa).

Devuelve el mensaje TIME STAMP REQUEST y TIME STAMP REPLY, uno es el pedido y la otra es la respuesta que manda el router.

* **Protocolo ARP (Protocolo de resolución de direcciones)**

Devuelve las direcciones MAC asociadas a direcciones IP que se consultan.

* **Protocolo RARP (reversa de ARP)**

Se utiliza para conocer la propia IP desde donde se realiza la consulta.

* **Protocolo BOOTP**

Sirve para inicializar toda la interfaz de red completa mediante la información que envía un servidor. Permite, además, cargar imágenes del sistema operativo.

* **Protocolo DHCP**

Sirve para asignar direcciones IP en forma dinámica.

**Algoritmos de enrutamiento**

El algoritmo de enrutamiento es aquella parte del software de la capa de red encargada de decidir la línea de salida por la que se transmitirá un paquete de entrada. Si la subred usa datagramas de manera interna, esta decisión debe tomarse cada vez que llega un paquete de datos, dado que la mejor ruta podría haber cambiado desde la última vez. Si la subred usa circuitos virtuales internamente, las decisiones de enrutamiento se toman sólo al establecerse un circuito virtual nuevo. En lo sucesivo, los paquetes de datos simplemente siguen la ruta previamente establecida. Este último caso a veces se llama enrutamiento de sesión, dado que una ruta permanece vigente durante toda la sesión de usuario (por ejemplo, durante una sesión desde una terminal, o durante una transferencia de archivos).

Algunas veces es útil distinguir entre el enrutamiento, que es el proceso consistente en tomar la decisión de cuáles rutas utilizar, y el reenvío, que consiste en la acción que se toma cuando llega un paquete. Se puede considerar que un enrutador realiza dos procesos internos. Uno de ellos maneja cada paquete conforme llega, buscando en las tablas de enrutamiento la línea de salida por la cual se enviará. Este proceso se conoce como reenvío. El otro proceso es responsable de llenar y actualizar las tablas de enrutamiento. Es ahí donde entra en acción el algoritmo de enrutamiento.

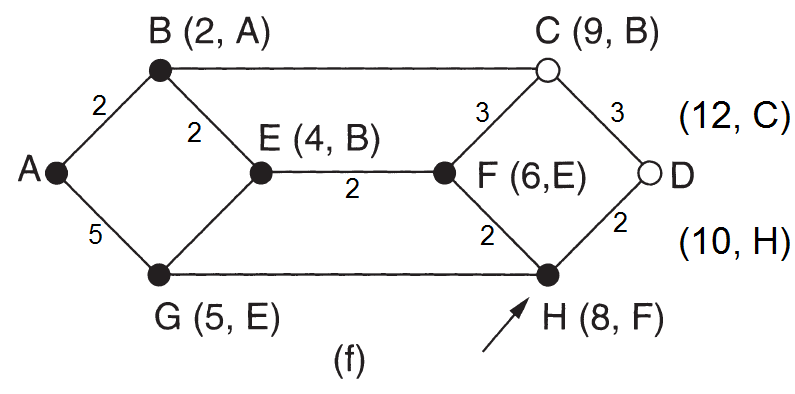
Los algoritmos de enrutamiento pueden agruparse en dos clases principales: no adaptativos (o estáticos) y adaptativos (o dinámicos). Los ***algoritmos no adaptativos*** no basan sus decisiones de enrutamiento en mediciones o estimaciones del tráfico y la topología actuales. En cambio, la decisión de qué ruta se usará para llegar de I a J (para todas las I y J) se toma por adelantado, fuera de línea, y se carga en los enrutadores al arrancar la red. Este procedimiento se conoce como enrutamiento estático.

En contraste, los ***algoritmos adaptativos*** cambian sus decisiones de enrutamiento para reflejar los cambios de topología y, por lo general también el tráfico. Los diferentes algoritmos adaptativos varían en el lugar de donde obtienen su información.

* ***Algoritmos Estáticos***

Enrutamiento por la trayectoria más corta

La idea es armar un grafo de la subred, en el que cada nodo representa un enrutador y cada arco del grafo una línea de comunicación (con frecuencia llamada enlace). Para elegir una ruta entre un par dado de enrutadores, el algoritmo simplemente encuentra en el grafo la ruta más corta entre ellos. Como desventaja del algoritmo podemos decir que si se cae algún nodo de la ruta calculada, no es posible llegar al destino, dado que es estático.



El camino más corto es

Inundación

Otro algoritmo estático es la inundación, en la que cada paquete de entrada se envía por cada una de las líneas de salida, excepto aquella por la que llegó. La inundación evidentemente genera grandes cantidades de paquetes duplicados; de hecho, una cantidad infinita a menos que se tomen algunas medidas para limitar el proceso. Una de estas medidas es integrar un contador de saltos (TTL) en el encabezado de cada paquete, que disminuya con cada salto, y el paquete se descarte cuando el contador llegue a cero. Lo ideal es inicializar el contador de saltos a la longitud de la ruta entre el origen y el destino. Si el emisor desconoce el tamaño de la ruta, puede inicializar el contador al peor caso, es decir, el diámetro total de la subred. Este algoritmo siempre llega al destino y siempre lo hace por la trayectoria más corta. No es adaptativo pero los paquetes terminan llegando por algún otro camino (si es que existe).

Una variación de la inundación, un poco más práctica, es la ***inundación selectiva***. En este algoritmo, los enrutadores no envían cada paquete de entrada por todas las líneas, sino sólo por aquellas que van aproximadamente en la dirección correcta.

Enrutamiento basado en flujo

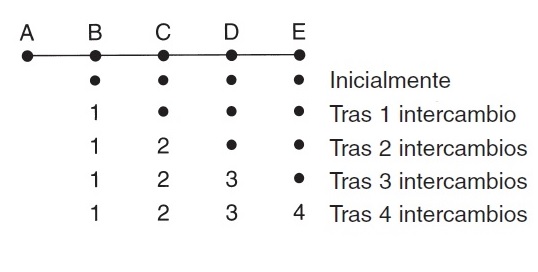
Es parecido al Dijkstra pero se basa en la carga que tienen las líneas de comunicación. Sabiendo la capacidad y la cantidad de nodos que tiene cada línea, genera las rutas, tratando de distribuir la carga lo más parejo posible.

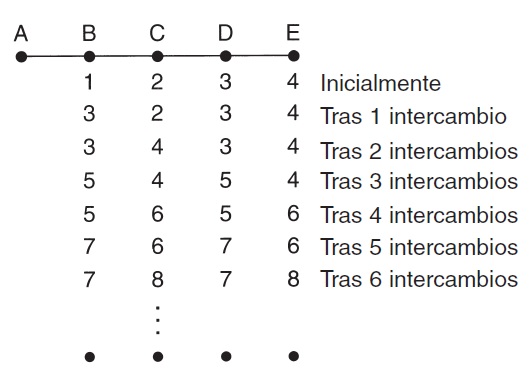
* ***Algoritmos dinámicos***

Enrutamiento por vector de distancia

Los algoritmos de enrutamiento por vector de distancia operan haciendo que cada enrutador mantenga una tabla (es decir, un vector) que da la mejor distancia conocida a todos y cada uno de los destinos y la línea que se puede usar para llegar ahí. Estas tablas se actualizan intercambiando información con los vecinos.

El enrutamiento por vector de distancia funciona en teoría, pero tiene un problema serio en la práctica: aunque llega a la respuesta correcta, podría hacerlo lentamente. En particular, reacciona con rapidez a las buenas noticias, pero con lentitud ante las malas.

A continuación, se observa cómo se enteran rápidamente los demás nodos de la aparición de .

Problema: problema de *“cuenta al infinito”*. Cuando desaparece un nodo (en este caso, ) nadie se entera de que desapareció puesto que siempre algún vecino tiene información de él. Esta información es consultada por todos los vecinos de este nodo que contiene la información, por lo que las tablas se irán actualizando mutuamente aumentando la cantidad de saltos para llegar a infinitamente.

Enrutamiento por estado de enlace

El concepto en que se basa el enrutamiento por estado del enlace es sencillo y puede enunciarse en cinco partes. Cada enrutador debe:

1. Descubrir a sus vecinos y conocer sus direcciones de red.
2. Medir el retardo o costo para cada uno de sus vecinos (ahora cada tabla solo tiene información de los vecinos y no de todos los nodos).
3. Construir un paquete que indique todo lo que acaba de aprender.
4. Enviar este paquete a *todos* los demás enrutadores.
5. Con el paquete que recibe, calcular la ruta más corta a todos los demás enrutadores utilizando Dijkstra.

La principal diferencia y ventaja con el algoritmo anterior es que un enrutador le pasa información de sus vecinos a todos y no de todos los enrutadores.

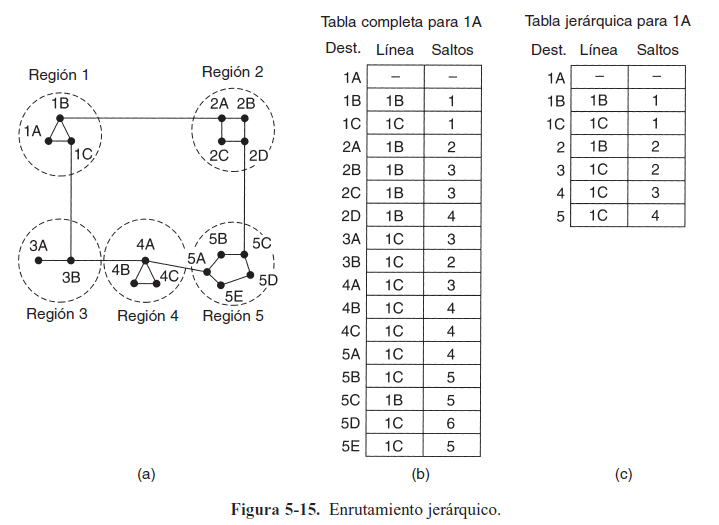
Enrutamiento jerárquico

A medida que crece el tamaño de las redes, también lo hacen, de manera proporcional, las tablas de enrutamiento del enrutador. Las tablas que siempre crecen no sólo consumen memoria del enrutador, sino que también se necesita más tiempo de CPU para examinarlas y más ancho de banda para enviar informes de estado entre enrutadores. En cierto momento, la red puede crecer hasta el punto en que ya no es factible que cada enrutador tenga una entrada para cada uno de los demás enrutadores, por lo que el enrutamiento tendrá que hacerse de manera jerárquica, como ocurre en la red telefónica.

Cuando se utiliza el enrutamiento jerárquico, los enrutadores se dividen en lo que llamaremos ***regiones*** o ***zonas***, donde cada enrutador conoce todos los detalles para enrutar paquetes a destinos dentro de su propia región, pero no sabe nada de la estructura interna de las otras regiones (solo sabe cómo llegar hasta éstas).

Cuando se interconectan diferentes redes, es natural considerar cada una como región independiente a fin de liberar a los enrutadores de una red de la necesidad de conocer la estructura topológica de las demás.

En la figura 5-15 se da un ejemplo cuantitativo de enrutamiento en una jerarquía de dos niveles con cinco regiones. La tabla de enrutamiento completa para el enrutador 1A tiene 17 entradas, como se muestra en la figura 5-15(b). Si el enrutamiento es jerárquico, como en la figura 5-15(c), hay entradas para todos los enrutadores locales, igual que antes, pero las demás regiones se han condensado en un solo enrutador, por lo que todo el tráfico para la región 2 va a través de la línea 1B-2A, pero el resto del tráfico remoto viaja por la línea 1C-3B. El enrutamiento jerárquico redujo la tabla de 17 entradas a 7. A medida que crece la razón entre la cantidad de regiones y el número de enrutadores por región, aumentan los ahorros de espacio de tabla.



Desgraciadamente, estas ganancias de espacio no son gratuitas. Se paga un precio, que es una longitud de ruta mayor. Por ejemplo, la mejor ruta de 1A a 5C es a través de la región 2 pero con el enrutamiento jerárquico, todo el tráfico a la región 5 pasa por la región 3, porque eso es mejor para *la mayoría* de los destinos de la región 5.

Hoy en día, este algoritmo es el que más se utiliza y se lo aplica a las direcciones IP dentro de una red. Análogamente, las letras pasan a ser direcciones IP y las zonas una red o subred.

**Capa 4: Capa de Transporte**

La tarea de esta capa es proporcionar un transporte de datos confiable y económico de la máquina de origen a la máquina de destino, independientemente de la red física en uso.

Internet tiene dos protocolos principales en la capa de transporte:

* UDP: protocolo no orientado a la conexión.
* TCP: protocolo orientado a la conexión.

**UDP** (Protocolo de datagrama de usuario).

Este protocolo proporciona una forma para que las aplicaciones envíen datagramas IP encapsulados sin tener que establecer una conexión.

No realiza control de flujo, control de errores o retransmisión cuando se recibe un segmento erróneo. No tiene número de secuencia. Provee velocidad de transmisión. Se usa en video y voz.

UDP es un protocolo simple y tiene algunos usos específicos, pero para la mayoría de las aplicaciones de Internet se necesita una entrega en secuencia confiable, por lo que se necesita el protocolo TCP.

**TCP** (**Protocolo de Control de Transmisión**)

Se diseñó específicamente para proporcionar un flujo de bytes confiable de extremo a extremo a través de una interred no confiable. Una interred difiere de una sola red debido a que diversas partes podrían tener diferentes topologías, anchos de banda, retardos, tamaños de paquete y otros parámetros. TCP tiene un diseño que se adapta de manera dinámica a las propiedades de la interred y que se sobrepone a muchos tipos de fallas.

TCP se definió en el estándar RFC 793.

Debe proporcionar la confiabilidad que la mayoría de los usuarios desean y que IP no proporciona. Muchas aplicaciones de la capa de aplicación confían a TCP la entrega de los paquetes al destino final. Maneja secuenciamiento de datos y recuperación de errores.

El modelo del servicio TCP

El servicio TCP se obtiene al hacer que tanto el servidor como el cliente creen puntos terminales, llamados *sockets*. Cada socket tiene un número (dirección), que consiste en la dirección IP del host y un *puerto*.

Todas las comunicaciones TCP se realizan mediante un puerto fuente y uno destino, los cuales pueden ser encontrados en la cabecera TCP. La condición para transmitir los datos es conocer el puerto destino. El puerto origen en la secuencia de transmisión no es muy importante y se selecciona de manera aleatoria. Dos o más conexiones pueden terminar en el mismo socket.

Hay 6535 puertos disponibles para una comunicación TCP, los cuales se dividen en dos grupos:

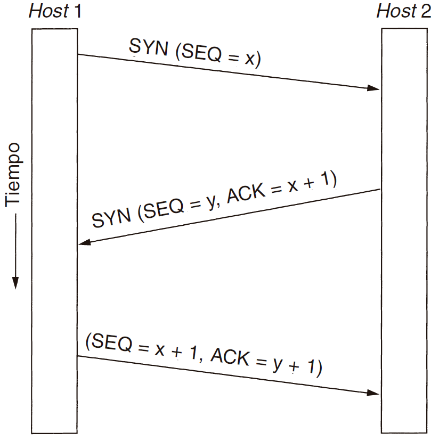
* Puertos estándar: 1 a 1023
* Puertos efímeros: 1024 a 6535.

Todas las conexiones TCP son de dúplex total y de punto a punto. No soporta la multidifusión ni la difusión.

Toda comunicación trabaja de la siguiente forma: Se elige un puerto origen aleatorio para comunicarse a un puerto destino desconocido. Una vez que el paquete es enviado, el dispositivo remoto se comunica con el dispositivo fuente usando los puertos establecidos.

Todas las comunicaciones comienzan con un saludo entre dos hosts. Tiene un saludo/acuerdo de 3 vías (*handshake*). Este proceso sirve a diferentes propósitos:

1. Permite al host transmisor asegurar que el host destino esté activo para comunicarse.
2. Dejar al transmisor chequear que el puesto destino esté escuchando.
3. Permitir al transmisor enviar su número de secuencia de comienzo al receptor para que ambos hosts puedan manejar la cadena de paquetes en la secuencia apropiada.

Pasos:

1. El host 1 envía un paquete SYN a su objetivo. SYN incluye un número de secuencia y el tamaño máximo de segmentos (MSS) que será usado en el proceso de comunicación.
2. El host 2 responde con SYN+ACK.
3. El host 1 envía un paquete ACK.

Después de esto, ambos dispositivos deberían tener toda la información para comunicarse adecuadamente.

TCP teardown: cada saludo tiene un teardown para finalizar la comunicación. Utiliza otra bandera llamada FIN. Consta de 4 paquetes (FIN – ACK – FIN –ACK).

TCP resets: cada conexión debería finalizar con un teardown, pero no siempre sucede. La bandera RST se utiliza para indicar que la conexión terminó abruptamente o para rehusar un intento de conexión.

Recuperación de errores:

TCP retransmisiones se basa en:

* Latencias: tiempo de comunicación entre emisor y receptor.
* Round-trip Time: tiempo de ida y vuelta estimado al destino.

Los paquetes TCP se pierden por aplicaciones que funcionan mal, routers congestionado o servidor fuera de servicio, etc.