**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.

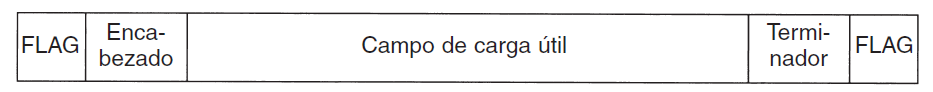
Cuestiones de Diseño de la capa de Enlace de Datos

Tiene dos subcapas: LLC (Control del enlace lógico) y MAC (Control de acceso al medio)

**LLC (Control del enlace lógico)**

Cuatro funciones principales:

1. Gestionar las tramas o marcos (debe ser capaz de armar y desarmar).
2. Proporcionar a la capa de red una interfaz de servicio bien definida.
3. Manejar los errores de transmisión.
4. Regular el flujo de datos para evitar saturación.

Para cumplirlas, la capa de enlace toma de la capa de red los paquetes y los encapsula en **tramas** (frame o marco) para transmitirlos. Cada trama contiene:

* Encabezado.
* Campo de carga útil para almacenar el paquete.
* Terminador o final (permite saber 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 origen a la capa de red en la máquina destino. En la capa de red de la máquina origen hay una entidad, llamada *proceso*, que entrega algunos bits a la capa de enlace para transmitirlos a la máquina destino. El trabajo de la capa de enlace es transmitir los bits a la máquina destino, para que puedan ser entregados a su capa de red.

Tres servicios que normalmente se proporcionan:

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

Consiste en que la máquina origen envíe tramas independientes a la máquina destino sin pedir que ésta confirme la recepción. No se establece conexión de antemano. 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). La mayoría de las LANs utilizan este servicio.

* *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. Este servicio es útil en canales inestables (inalámbricos).

* *Servicio orientado a la conexión con confirmación de recepción*

Las máquinas de origen y destino establecen una conexión antes de transferir datos. Cada trama enviada a través de la conexión está numerada, y la capa de enlace garantiza que cada trama llegará a su destino, que será recibida sólo una vez y en el orden adecuado.

Las transferencias tienen tres fases: se establece la conexión haciendo que ambos lados inicialicen las variables y los contadores, se transmiten una o más tramas y se cierra la conexión liberando los recursos utilizados.

Entramado

La capa de enlace 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. La capa de enlace debe detectar y corregir los errores.

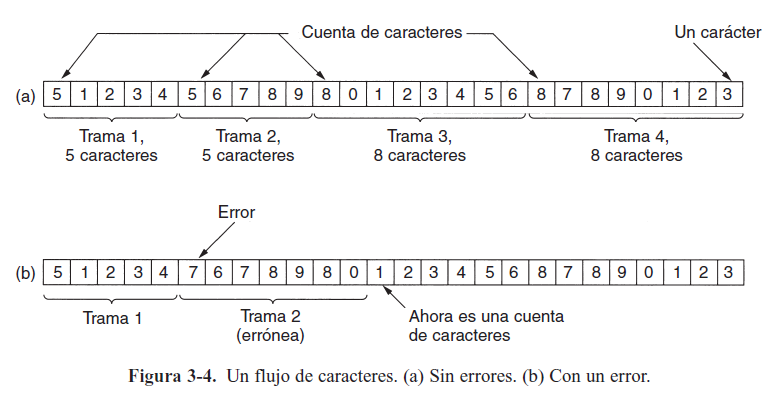
Métodos para marcar el inicio y el final de cada trama:

* *Conteo de caracteres*

Usa un campo en el encabezado para especificar el número de caracteres en la trama, con el cual sabe cuántos caracteres siguen, por lo que no usa finalizador.

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.



Se alteró la cuenta de la segunda trama

* *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 llamados **banderas** (FLAG o indicadores). 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.

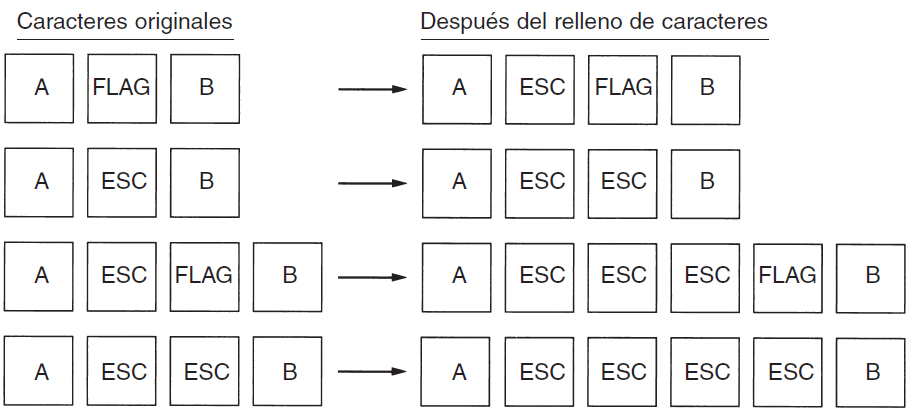
Situaciones posibles:

* + 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**.

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

Resolución: También se rellena con un byte de escape antes del dato ‘escape’.

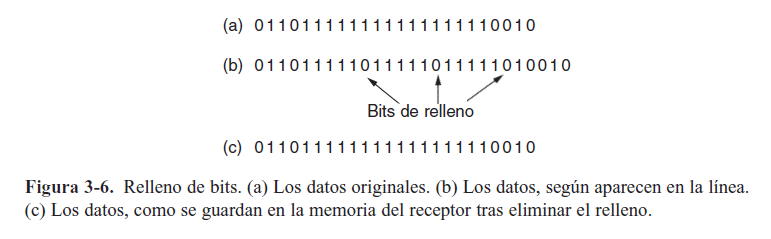
Una desventaja importante de esta técnica es que está fuertemente atada a los caracteres de 8 bits, siendo que no todos los códigos utilizan caracteres de 8 bits.

* *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’. Cada vez que la capa de enlace del emisor encuentra cinco unos consecutivos en los datos, automáticamente inserta un bit 0 en el flujo de bits saliente. Esta técnica se llama **relleno de bits**.

Cuando el receptor ve cinco bits 1 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. Consiste en utilizar pulsos eléctricos incoherentes como indicadores de inico y fin de trama. Las combinaciones alto-alto y bajo-bajo no se usan para datos, pero en algunos protocolos se utilizan como indicadores.

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 y, si el delimitador apropiado está en esa posición y la suma de verificación es correcta, la trama se acepta como válida.

**MAC (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 esta subcapa. 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.

**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.

Problemas: Cuando algunos usuarios están inactivos, su ancho de banda simplemente se pierde. No lo están usando, y a nadie más se le permite usarlo. 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.

Asignación dinámica de canales en LANs y MANs

Todo el trabajo hecho en esta área se basa en cinco supuestos claves:

1. **Modelo de estación:** Consiste en *N* estaciones independientes, cada una con un programa o usuario que genera tramas para transmisión. Una vez que se ha generado la trama, la estación se bloquea y no hace nada hasta que la trama se haya transmitido con éxito.
2. **Supuesto de canal único:** Hay un solo canal disponible para todas las comunicaciones. Todas las estaciones pueden transmitir en él y pueden recibir de él.
3. **Supuesto de colisión:** Si dos tramas se transmiten en forma simultánea, se traslapan en el tiempo y la señal resultante se altera produciendo una colisión. Todas las estaciones pueden detectar colisiones. Una trama en colisión debe transmitirse nuevamente después.
4. **Tiempo continuo:** La transmisión de una trama puede comenzar en cualquier momento.
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. 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

La idea básica es permitir que los usuarios transmitan cuando tengan datos para enviar. Cada vez que dos tramas traten de ocupar el canal al mismo tiempo, habrá una colisión y ambas se dañarán. Un emisor siempre puede saber si la trama fue destruida o no escuchando el canal, pero este protocolo no presenta detección de portadora por lo que se utiliza la suma de verificación para detectar cualquier tipo de daño en la transmisión.

Las tramas se transmiten en momentos completamente arbitrarios. El uso del canal es del **18%**. 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.

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.

ALOHA Ranurado

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

Una estación debe esperar el comienzo de una ranura libre para transmitir. Lo mejor que podemos esperar usando ALOHA ranurado es 37% de ranuras vacías, **37%** de éxitos y 26% de colisiones. Además, tampoco puede 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 el 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.

Desventaja: Retardo de propagación. 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 producirá 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*

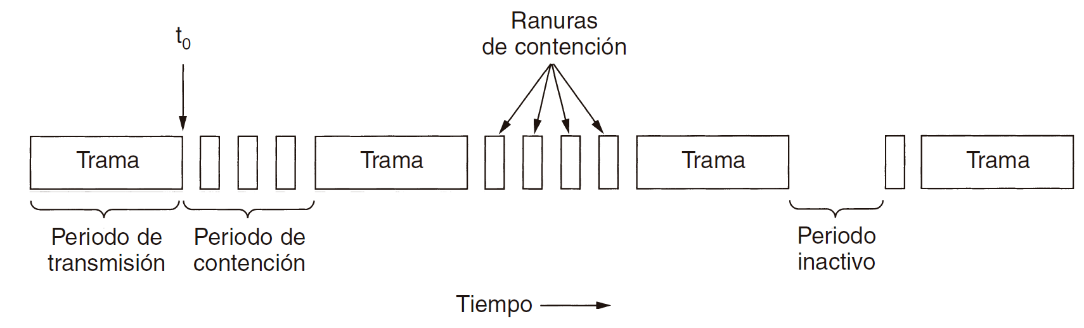
Se aplica a canales ranurados. Cuando una estación está lista para enviar, escucha el canal. Si éste se encuentra inactivo, la estación transmite con una probabilidad *p*. Con una probabilidad , se espera hasta la siguiente ranura. Si esta ranura también está inactiva, la estación transmite o espera 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 el protocolo **CSMA/CD (Acceso Múltiple con Detección de Portadora y Detección de Colisiones)**, en el que las estaciones abortan sus transmisiones tan pronto como detectan una colisión. La terminación pronta de tramas dañadas ahorra tiempo y ancho de banda. Este protocolo se usa ampliamente en las LANs.

Una vez que se aborta la transmisión, la estación 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).



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.

Vale la pena mencionar que una estación debe monitorear de manera continua el canal en busca de ráfagas de ruido que indiquen 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 (802.3)**

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:

. 1° nro: velocidad en Mbps.   
. Base-[ ]: indica la transmisión de banda base y el medio.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |

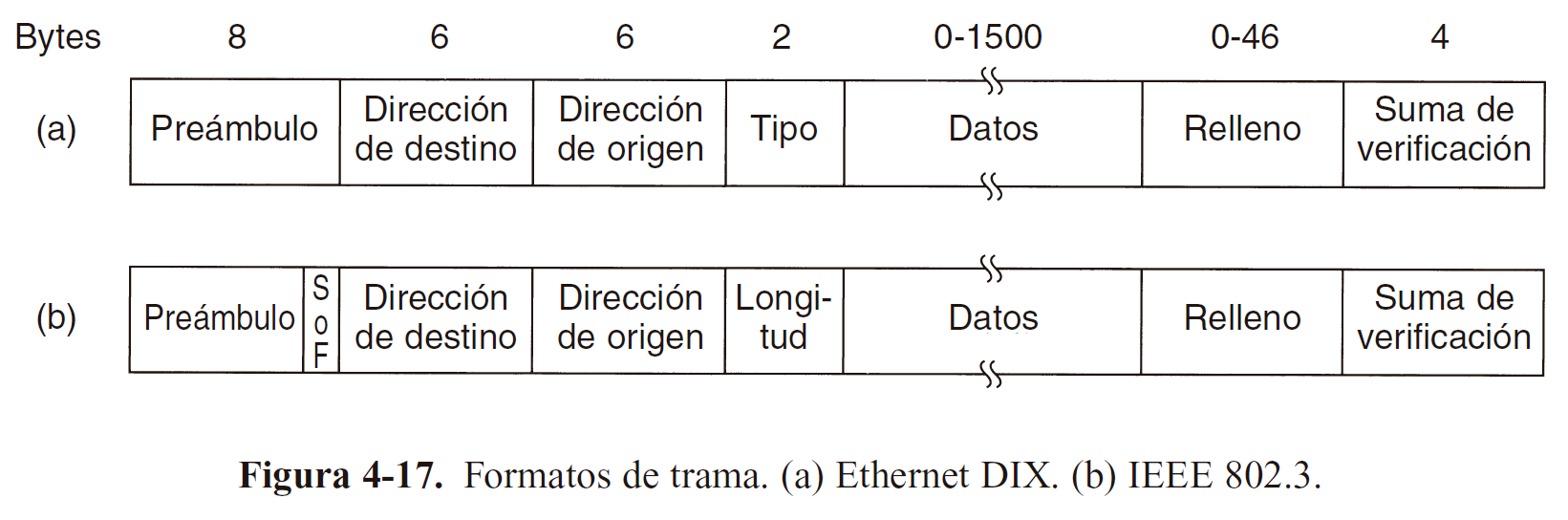
**10Base5**: Opera a 10Mbps, utiliza señalización de banda base y puede manejar segmentos de hasta 500 metros.

**10Base2**: El Ethernet delgado es mucho más económico, se dobla con facilidad y sus conectores son más fáciles de usar y más confiables. Sólo puede extenderse 185 metros por segmento.

**10Base-T**: Todas las estaciones tienen cables que conducen a un concentrador central o hub, al que se conectan de manera eléctrica. Soluciona los problemas asociados con la localización de rupturas de cable.

**10Base-F**: Usar fibra óptica es caro 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 y ofrece buena seguridad.

El protocolo de la subcapa MAC de Ethernet



El tamaño mínimo de trama es de 64 bytes (de la dirección de destino a la suma de verificación) y es necesaria para que Ethernet pueda distinguir con facilidad las tramas válidas de la basura. Además sirve en el caso que se quiera enviar una trama muy corta, donde puede ocurrir una colisión, pero la transmisión se completa antes de que la ráfaga de ruido llegue de regreso.

Estructura de una trama original de DIX:

* Inicia con un *Preámbulo* de 8 bytes, cada contiene el patrón de bits 10101010. Se utiliza para la sincronización.
* Contiene dos *Direcciones* (origen y destino) de 6 bytes cada una. Se denominan direcciones MAC.

Multicast (multidifución): Es una dirección de grupo que permite 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.

Broadcast (difución): Es una dirección reservada que consiste únicamente en bits 1 (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.

* *Tipo*: Indica al receptor qué hacer con la trama y a qué proceso darle la trama.
* *Datos*: Máximo de 1500 bytes.
* *Relleno*: Si la longitud de una trama es menor a 46 bytes, se usa para rellenar hasta el tamaño mínimo.
* *Suma de verificación*: Es un código de *hash* de los datos de 32 bits. Si algunos bits de datos se reciben erróneamente, 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, IEEE 802.3 utiliza como protocolo de acceso múltiple *CSMA P-1/CD.*

Direcciones MAC

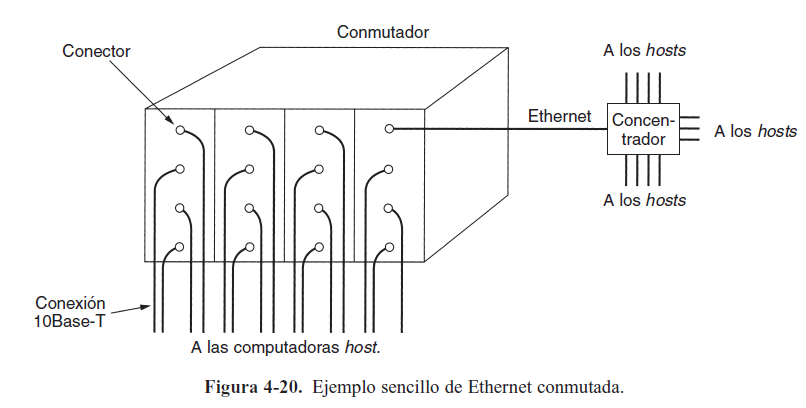
Direcciones asignadas a las interfaces físicas. Son de 6 bytes separados por “:”. Los 3 primeros corresponden al fabricante y los 3 últimos identifican al dispositivo con su respectivo modelo, teniendo cada dispositivo un número diferente. Esos 6 bytes se representan de forma hexadecimal en 6 pares.

Dentro de las direcciones MAC, hay una que es especialmente dedicada para broadcast (FF:FF:FF:FF:FF:FF).

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 sería utilizar una velocidad mayor pero incluso una Ethernet de 1 Gbps puede saturarse. Una solución diferente sería una Ethernet conmutada. El corazón de este sistema es un conmutador (switch) que contiene una matriz de conmutación de alta velocidad y espacio para 4 a 32 tarjetas de línea, cada una 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. Por lo general, dicha matriz de conmutación funciona a más de 1 Gbps usando un protocolo patentado.



Cuando una estación quiere transmitir envía una trama estándar al conmutador.

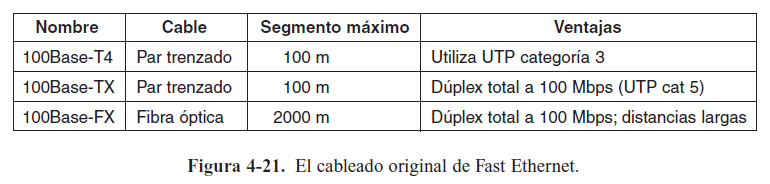
La tarjeta que recibe la trama la revisa para ver si está destinada a una de las otras estaciones conectadas a la misma tarjeta. De ser así, la trama se copia ahí.

Si no, la trama se envía a través de la matriz de conmutación de alta velocidad a la tarjeta de la estación de destino.

**Fast Ethernet (802.3u)**

El comité 802.3 decidió crear una Ethernet mejorada por la necesidad de compatibilidad hacia atrás con las LANs Ethernet existentes. La idea básica fue mantener todos los formatos anteriores, interfaces y reglas de procedimientos, y sólo reducir el tiempo de bits de 100 nseg a 10 nseg (para hacer que sea 10 veces más rápida se baja el pulso de corriente). Se basa por competo en el cableado 10Base-T por sus ventajas. Por lo tanto, todos los sistemas Fast Ethernet utilizan concentradores y conmutadores.

Soporta los cables UTP de categoría 3 y 5 y fibra óptica. La principal desventaja del UTP 3 es su incapacidad de llevar señales de 100 Mbps a una distancia de hasta100 metros por lo que hay que tener ciertas consideraciones.



**100Base-**T4: Usa 4 cables de par trenzado del UTP 3 con una velocidad de señalización de 25 MHz.

**100Base-TX**: UTP 5 puede manejar velocidades de reloj de 125 MHz. Sólo se utilizan dos cables de par trenzado por estación, uno para enviar y otro para recibir, permitiendo dúplex total.

**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. Los concentradores no están permitidos con este cableado.

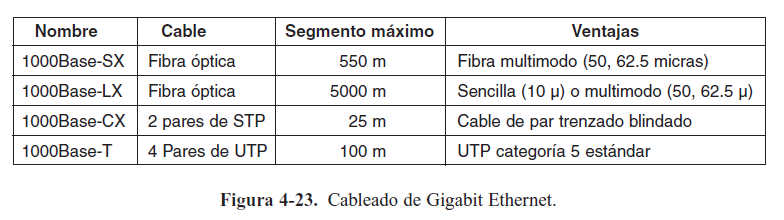
Existe 100Base-T2 que permite que la Fast Ethernet se ejecute a través de dos pares de cables existentes de categoría 3 pero es muy costoso y casi no se usa.

*Nota*: Casi todos los conmutadores pueden manejar una mezcla de estaciones de 10 y 100 Mbps. El estándar mismo proporciona una forma para que dos estaciones negocien de manera automática la velocidad óptima y el tipo de transmisión dúplex. La mayoría de los productos de Fast Ethernet utilizan esta característica para autoconfigurarse.

**Gigabit Ethernet (802.3z)**

Los objetivos eran esencialmente los mismos: 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.

Cableado de Gigabit Ethernet:



**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. Casi no se usa.

**1000Base-T**: Usa paquetes de cuatro cables UTP 5 que trabajan juntos. Por el tipo de cableado, es lo más usado.

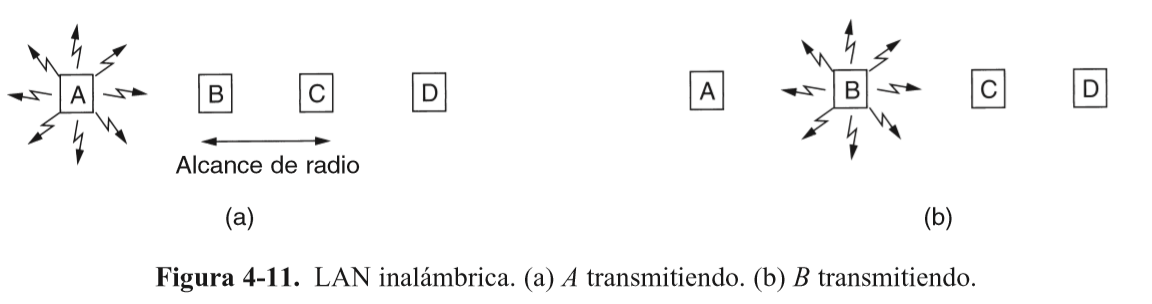
Soporta dos modos diferentes de funcionamiento: dúplex total y semidúplex.

* 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 (switch). El conmutador almacena todas las tramas entrantes en un buffer. El emisor no tiene que escuchar el canal ya que es imposible que haya colisiones por ser dúplex total. Cada computadora y conmutador puede enviar tramas siempre que lo desee. 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 (hub). 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. 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 (tiene una eficiencia 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.

**Protocolos de LANs inalámbricas (802.11)**

En las LANs inalámbricas no todas las estaciones están dentro del alcance de todas las demás ya que las ondas electromagnéticas tienen poca fuerza, lo que conduce a una variedad de complicaciones. 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. No se pude usar el protocolo CDMA ya que detecta la interferencia en el emisor, no en el receptor.

Surgirían los siguientes problemas:



**Problema de estación oculta**: Una estación no puede detectar a un competidor potencial por el medio, porque está demasiado lejos. Ej: Cuando A está transmitiendo hacia B y C quiere transmitirle a B, 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.

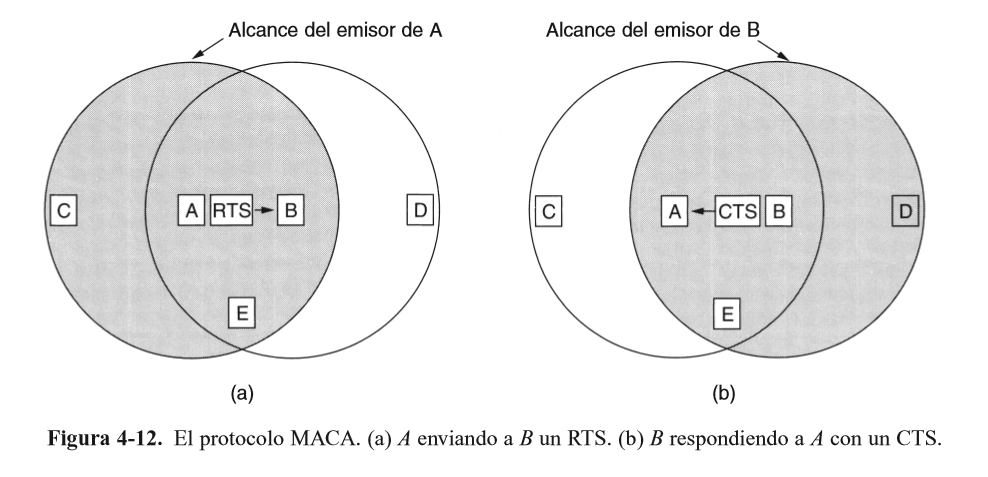
**Problema de estación expuesta**: No se realiza una transmisión porque el emisor, al detectar el canal ocupado, supone que incorrectamente que va a generar conflictos. Ej: Cuando B está transmitiendo a A y C quiere comunicarse con D, si C detecta el medio, escuchará una transmisión y concluirá equivocadamente que no puede enviar a D, cuando de hecho tal transmisión causaría una mala recepción sólo en la zona entre B y C.

**MACA y MACAW**

MACA (Acceso Múltiple con Prevención de Colisiones) es uno de los primeros protocolos diseñados para solucionar estos problemas. 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.

El emisor enviará un RTS para solicitar en envío de tramas y el receptor responderá con un CTS si está disponible. Estas tramas serán escuchadas por las estaciones cercanas al emisor y receptor y deberán permanecer calladas durante el tiempo suficiente para que se reciban los datos correctamente.

La trama RTS contiene la longitud de la trama de datos a enviar y la CTS va a copiar esta longitud. Ambas tramas tienen las direcciones MAC de origen y destino.



Consideremos ahora la manera en que A envía una trama a B. A comienza por enviar una trama RTS a B. Después B contesta con una trama CTS (Libre para Envío). Una vez que sucede la recepción de la trama CTS, A comienza a transmitir. 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.

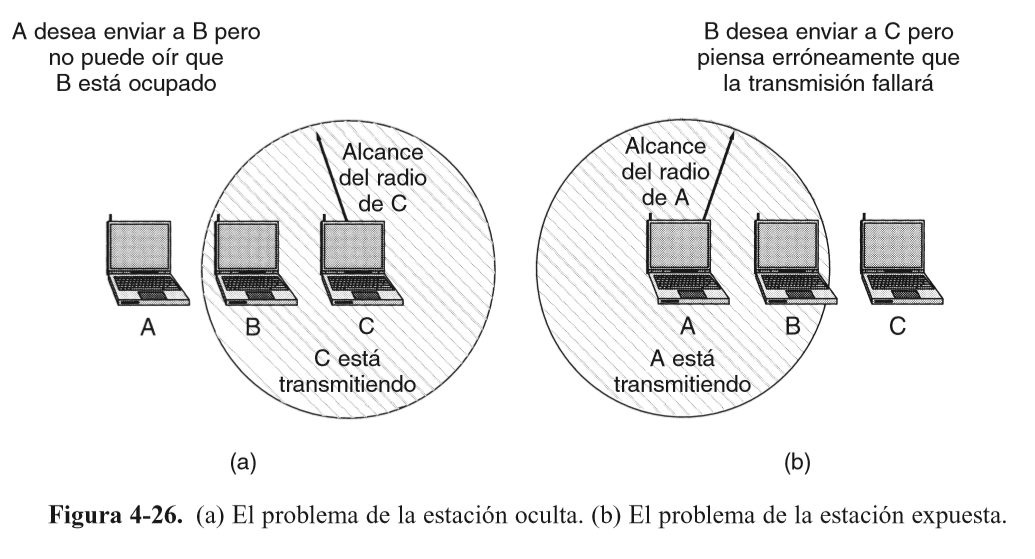
C escucha sólo el RTS de A porque está en su alcance. Mientras no interfiera con el CTS, está libre para transmitir mientras se está enviando la trama de datos. En cambio, D está en el alcance de B y al escuchar el CTS se le indica que está cerca de una estación que está a punto de recibir una trama, por lo que difiere el envío de cualquier cosa hasta la terminación de esa trama. La estación E también debe permanecer en silencio.

MACAW (MACA Inalámbrico) mejora el desempeño de MACA. 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 y estación expuesta como los de la imagen:



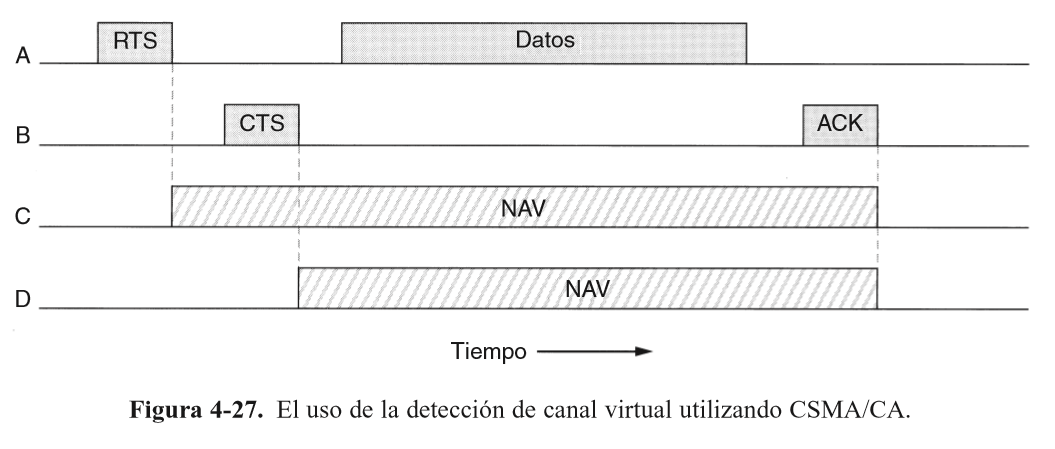
Para solucionar este problema, 802.11 utiliza un protocolo llamado **CSMA/CA (CSMA con Evitación de Colisiones)** que utiliza tanto la detección del canal físico como la del canal virtual.

Soporta dos modos de funcionamiento. El primero, llamado **DCF (Función de Coordinación Distribuida)**, no utiliza ningún tipo de control central. El otro, llamado **PCF (Función de Coordinación Puntual)**, utiliza un controlador central 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.

Con DCF, primero se 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 y vuelve a intentarlo más tarde.

El modo PCF se basa en MACAW y utiliza la detección de canal virtual. 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, no ocurren colisiones. Envía tramas de sondeo.

En este ejemplo se muestra PCF, donde A desea enviar a B. C es una estación que está dentro del alcance de A. 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.

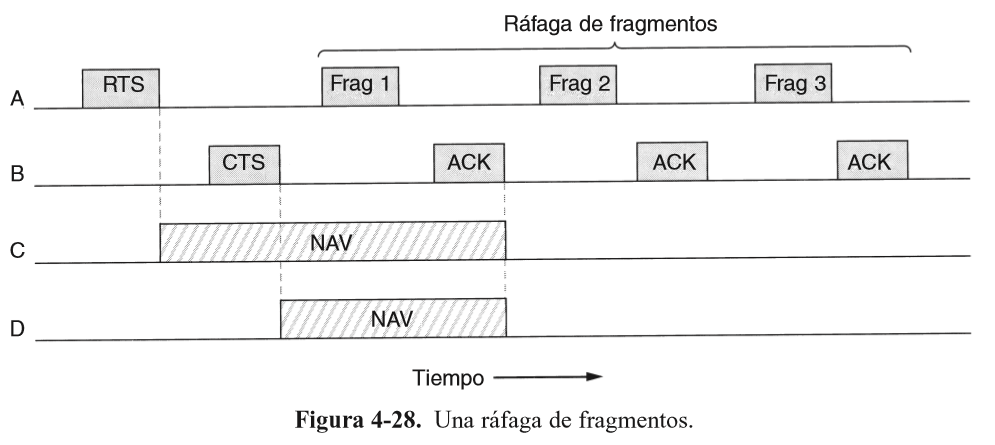
1. A inicia enviándole un RTS a B.
2. B recibe y responde con un CTS.
3. Al recibir la CTS, A envía su trama y comienza su temporizador de ACK.
4. Al recibir correctamente la trama de datos, B responde con una trama de ACK, con lo que termina el intercambio.

Si el temporizador de ACK de A termina antes de que el ACK regrese, todo el protocolo se ejecuta de nuevo.

C está dentro del alcance de A, por lo que recibe la trama RTS y se da cuenta de que alguien va a enviar datos pronto. A partir de la información proporcionada en la solicitud RTS, C puede estimar cuánto tardará la secuencia, por lo que impone para sí misma un tipo de canal virtual ocupado, indicado por **NAV (Vector de Asignación de Red)**. D escucha el CTS, por lo que también impone la señal NAV para sí misma.

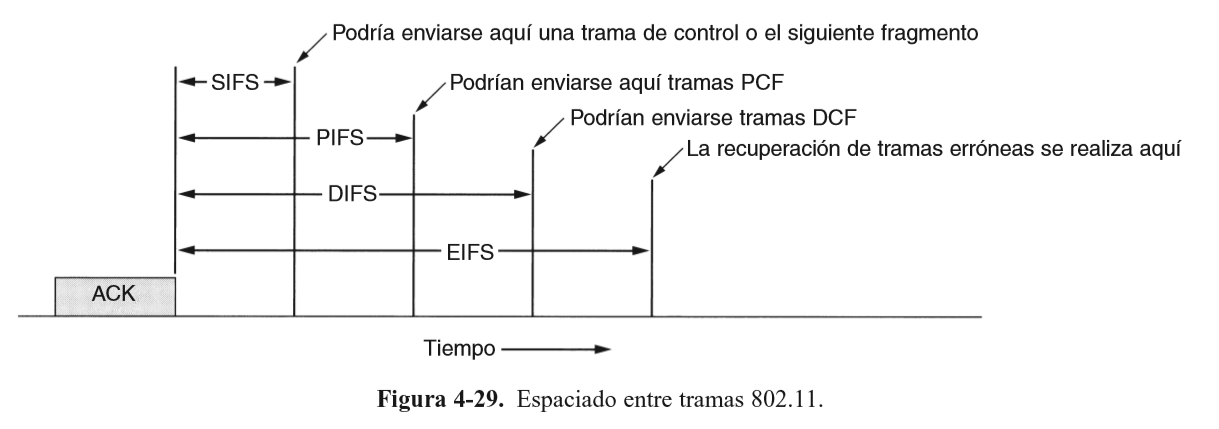
Las señales NAV no se transmiten; simplemente son recordatorios internos para mantenerse en silencio.

Las redes inalámbricas son ruidosas e inestables, por lo que la probabilidad de que una trama llegue a su destino disminuye 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. Su recepción se confirma utilizando un protocolo de parada y espera (ACK). Una vez que se ha adquirido el canal mediante RTS y CTS, pueden enviarse múltiples fragmentos en una fila (**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 mecanismo NAV mantiene otras estaciones en silencio sólo hasta la siguiente confirmación de recepción, pero se utiliza otro mecanismo para permitir que otra ráfaga de fragmentos completa se envíe sin interferencia.

PCF y DCF pueden coexistir dentro de una celda. 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.



**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.

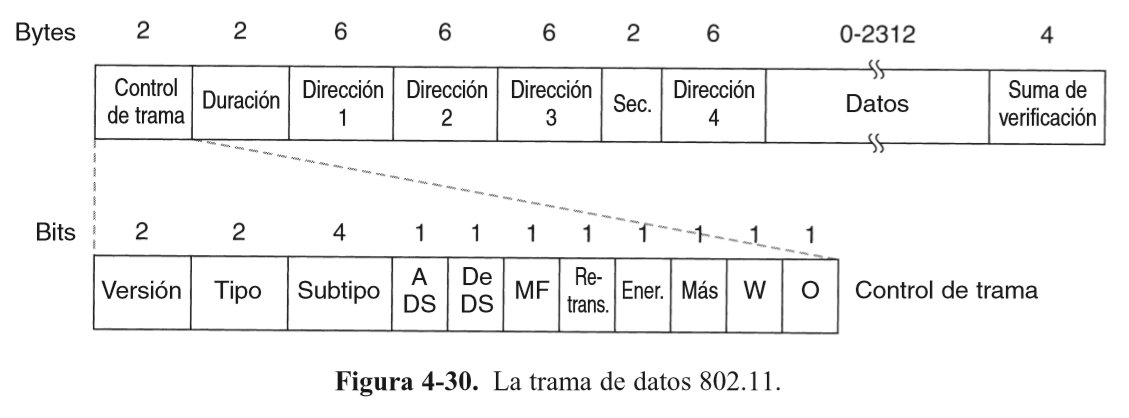
**PIFS (Espaciado Entre Tramas PCF)**: La estación base podría enviar una trama de sondeo. Este mecanismo permite que una estación base envíe una trama de datos o una secuencia de fragmentos para finalizar su trama sin que nadie interfiera, pero le da a la estación base la oportunidad de tomar el canal cuando el emisor anterior haya terminado, sin tener que competir con usuarios ansiosos.

**DIFS (Espaciado Entre Tramas DCF)**: Mediante el envío de tramas DCF, cualquier estación podría intentar adquirir el canal para enviar una nueva trama. Se aplican las reglas de contención normales.

**EIFS (Espaciado Entre Tramas Extendido)**: Sólo una estación que acaba de recibir una trama errónea o desconocida utiliza este tiempo para reportar la trama errónea.

**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.



*Control de trama* 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.

*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.

*Cuatro direcciones*: todas en formato estándar IEEE 802. El origen y el destino. Las otras dos se utilizan para las estaciones base de origen y destino para el tráfico entre celdas.

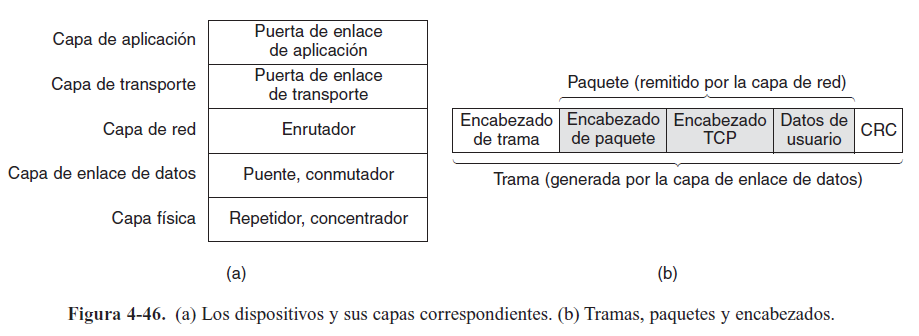
*Secuencia*: permite que se numeren los fragmentos.

*Datos*: contiene la carga útil, hasta 2312 bytes, y le sigue el campo común de *Suma de verificación*.

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

Para empezar, estos dispositivos operan en diferentes capas. La capa es importante porque los distintos dispositivos utilizan diferentes partes de información para decidir su modo de operación.

Cuando el usuario genera datos que se enviarán a una máquina remota, estos se pasan a la capa de transporte, que le agrega 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, un paquete IP. Este 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.



R**epetidor**: Dispositivo de dos bocas que amplifica la señal para lograr mayor alcance. Los repetidores no distinguen entre tramas, paquetes o encabezados, sólo repite señales eléctricas.

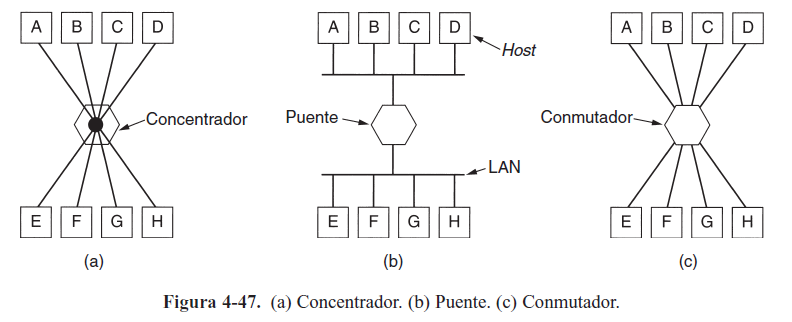
C**oncentrador (hub)**: Tiene numerosos puertos de entrada que une de manera eléctrica y permite conectar varios equipos. Las tramas que llegan a cualquiera de las líneas se envían a todas las demás. Si dos tramas llegan al mismo tiempo, chocarán. Todas las líneas que convergen en un concentrador deben operar a la misma velocidad. Al igual que los repetidores, los concentradores no examinan las direcciones 802 ni las utilizan de ninguna manera. Permite un solo domino de broadcast.

**Puente**: Conecta dos o más LANs por lo que conoce los protocolos y puede entender el formato de las tramas. Cuando llega una trama, se extrae la dirección de destino del encabezado y la busca en una tabla para averiguar a dónde debe enviar la trama. Descarta el resto de 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 broadcast.

**Conmutadores (switch)**: Enruta tomando como base las direcciones de las tramas. Envía datos únicamente por la boca del destino. Se utiliza para conectar computadoras individuales y por cada boca sabe qué dispositivo está conectado mediante una tabla MAC. Usa buffers para almacenar los datos lo que reduce la posibilidad de colisiones. A diferencia del puente, permite simular conexiones simultáneas. Tienen un puerto trunk para conectar 2 switchs y simular que son uno.

**Enrutadores**: 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.

**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. Usa paquetes con un solo encabezado.

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

Tiene 2 funciones:

1. *Fragmentación:* permite dividir el paquete en distintos tamaños y mandarlos por distintas tecnologías.
2. *Genera túneles:* sirve para conectar 2 puntos a través de un medio donde el que quiere ir de un lado a otro, no puede. Para enviar información encapsulada o pasar por una red que no soporta el protocolo original.

Servicios que ofrece:

1. Independizar la capa de la tecnología de capa 2.
2. Aislar la capa de transporte de la topología de las subredes.
3. Seguir un plan de numeración uniforme para las redes disponibles.

Objetivos:

1. Llevar paquetes.
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:

**Servicio no orientado a la conexión**: los paquetes se colocan individualmente en la subred y se enrutan de manera independiente. Los paquetes se conocen como *datagramas*. Varios paquetes de una misma transmisión van a ir por caminos separados.

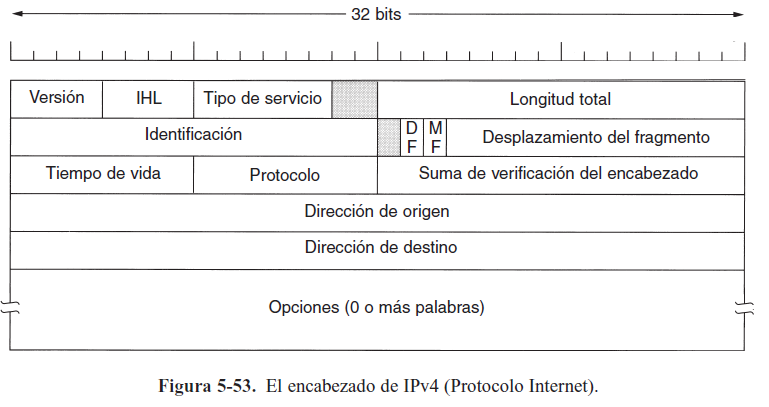
**Servicio orientado a la conexión**: antes de poder enviar cualquier paquete de datos, es necesario establecer una ruta del enrutador del origen al de destino y reservarse los recursos necesarios para la transmisión. Todos los paquetes van a utilizar el mismo camino. Esta conexión se conoce como *CV (circuito virtual)*.

|  |  |  |
| --- | --- | --- |
|  | **DATAGRAMAS** | **CV (circuito virtual)** |
| **Establecer conexión** | No | Si |
| **Direccionar** | Cada datagrama debe tener dos direcciones: origen y destino | No necesita porque cada paquete lleva el número de circuito a utilizar |
| **Enrutar** | Si, se enrutan por separado | Si, se enrutan al generar el circuito |
| **Efecto ante fallo** | Reenviar un paquete | 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 |

**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.



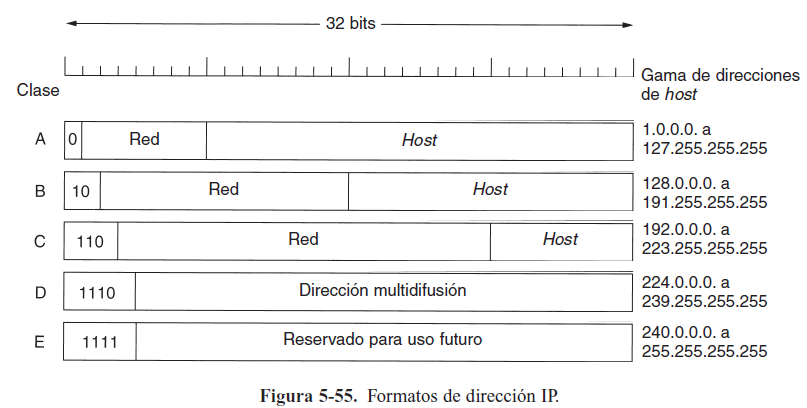
* *Versión: son 4 bits que* llevan el registro de la versión del protocolo al que pertenece el datagrama (IPv4 – IPv6).
* *IHL:* longitud del encabezado.
* *Tipo de Servicio:* son 8 bits para distinguir el servicio de ese paquete. Son posibles varias combinaciones de confiabilidad y velocidad. Permite especificar, por ejemplo, retardo (delay), velocidad real de transporte, confiabilidad.
* *Longitud Total:* son 16 bits e incluye todo el datagrama.
* *Identificación:* es necesario para que el host de destino determine a qué datagrama pertenece un fragmento recién llegado. Todos los fragmentos de un datagrama contienen el mismo valor de Identificación.
* *DF:* 1 bit. Significa no fragmentar.
* *MF:* 1 bit. Significa que hay más fragmentos por enviar.
* *Desplazamiento del Fragmento (Offset):* indica en qué parte del datagrama actual va este fragmento.
* *Tiempo de Vida (TTL):* es un contador que sirve para limitar la vida de un paquete. 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. Debe recalcularse en cada salto porque al menos uno de los campos cambia (tiempo de vida, por ejemplo).
* *Direcciones Origen y Destino:* indican la dirección de red y de host.
* *Opciones:* de longitud variable y sirve para especificar 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. Cada dirección es única. Cada IP debe estar acompañada por la *máscara de red*.

Todas las direcciones IP son de 32 bits de longitud. Una dirección IP realmente no se refiere a un host, sino a una interfaz de red, por lo que si un host está en dos redes, debe tener dos direcciones IP.

Las direcciones IP se dividieron en cinco categorías. Esta asignación se ha llamado **direccionamiento con clase**.



Redes privadas (fuera de internet) para cada clase

* A 10.0.0.0

10.255.255.255/8

* B 172.16.0.0

172.31.255.255/12

* C 192.168.0.0

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. Generalmente, se utiliza la clase A o C.

**Mascara:**

Nos va a determinar qué parte es de red (unos) y qué parte de host (ceros). Formada primero por 1 y después por 0. Se pueden mezclar los tipos de máscara con diferentes tipos de IP. Los tipos por defecto para cada IP son:

|  |  |
| --- | --- |
| Tipo | Mask |
| A | 255.0.0.0 |
| B | 255.255.0.0 |
| C | 255.255.255.0 |

Mediante la IP y la máscara se desprenden dos operaciones para averiguar las direcciones restantes:

* Para averiguar la dirección de red se realiza una operación *AND* entre la dirección IP y la máscara. (ip and mask)
* La de broadcast se encuentra mediante la operación *OR* entre la dirección IP y la máscara negada. (*ip or ¬mask*)

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

Notas: el rango de IP disponibles para hosts siempre va desde la dirección de red a la dirección de broadcast sin incluirlas.

; 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. Una subred es una división lógica dentro una red.

Para generar una nueva máscara para las subredes lo que se hace es agregar un una cantidad de 1 determinada a la máscara original de red.

Las posibles subredes son:

1. 1 → .128 – 2 subredes
2. 11 → .192 – 4 subredes
3. 111 → .224 – 8 subredes
4. 1111 → .240 – 16 subredes
5. 11111 → .248 – 32 subredes
6. 111111 → .252 – 64 subredes
7. 1111111 → .254 – 128 subredes

La cantidad de subredes que se pueden generar se determina mediante:

; siendo la cantidad de 1 que agrego, e es la cantidad de subredes que obtengo.

Cuestiones a tener en cuenta:

* La dirección 0.0.0.0/32 siempre se pone al final de la tabla
* Entre router e internet se usa una red pública.
* Si un router tiene salida a internet, no puede tener restricciones.
* Las subredes pueden ser de diferentes tamaños. Sólo hay que tener en cuenta el rango de ip adecuado para cada subred. (La subred más chica va a empezar su rango desde la última dirección de la subred más grande)
* En la tabla se pone la red con su máscara y la boca del router por la que sale.
* Si hay dos switchs conectados a un router, poner en el gráfico las direcciones de las puertas de enlace entre switch y router.
* Se usa como puerta de enlace la última dirección disponible en la red.
* Si tenemos dirección de red y broadcast y queremos calcular la máscara de red, tener en cuenta que los octetos de ambas direcciones son iguales, es porque en esos octetos la máscara es 255.
* Para calcular una máscara a partir de la dirección de red y broadcast hacer:

red 192.**88**.0.0 y broadcast 192.**91**.255.255.

91 – 88 = 4 256 – 4 = 252.

Máscara: 255.**252**.0.0

Cálculos:

;

; ;

; ;

Otros protocolos de capa 3

* **Protocolo ICMP (Protocolo para mensajes de control)**

Son varios mensajes para realizar control en la subred. Cada uno de los mensajes tiene una respuesta esperada. Se usan para preguntarle a cierta dirección IP si esta activa, y mostrar todos los routers por los que pasa.

* **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 SO.

* **Protocolo DHCP**

Sirve para asignar direcciones IP en forma dinámica. (Inalámbrico)

**Algoritmos de enrutamiento**

Es la 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.

Hay que distinguir entre 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. El algoritmo de enrutamiento se encarga de actualizar las tablas de enrutamiento para que la acción de reenvío puede buscar en dichas tablas la ruta para enviar los paquetes.

Los algoritmos de enrutamiento pueden agruparse en dos clases principales:

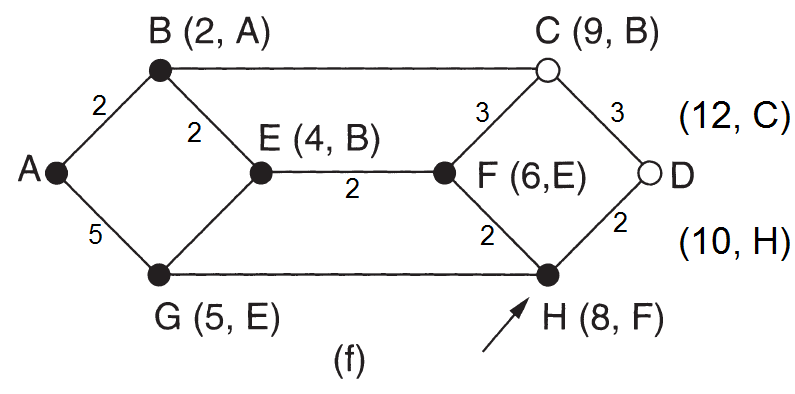
**Algoritmos estáticos:** Las decisiones de enrutamiento se toma por adelantado y se carga en los enrutadores al arrancar la red. La desventaja es que si hay algún cambio en la topología, no se podrá adaptar.

**Algoritmos dinámicos:** Calculan las rutas a medida que la topología va cambiando.

* ***Algoritmos Estáticos***

Enrutamiento por la trayectoria más corta

Consiste en armar un grafo de la subred, en el que cada nodo representa un enrutador y cada línea de comunicación o enlace tiene un peso. Para elegir una ruta entre un par dado de enrutadores, el algoritmo busca la ruta más corta entre ellos.

 El camino más corto es

Inundación

Cada paquete de entrada se envía por cada una de las líneas de salida, excepto por la que llegó. Genera grandes cantidades de paquetes duplicados por lo que se le integrar un contador de saltos (TTL) en el encabezado de cada paquete para que disminuya con cada salto y 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 o la longitud total de la subred. Este algoritmo siempre llega al destino y por la trayectoria más corta.

Una variación es la *inundación selectiva*. Los enrutadores envían cada paquete sólo por las líneas que van aproximadamente en la dirección correcta.

Enrutamiento basado en flujo

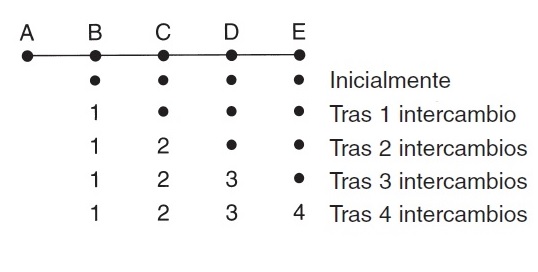
Es parecido a Inundación pero se basa en la carga que tienen las líneas de comunicación. Sabiendo la capacidad y la cantidad de nodos que tiene cada línea, genera las rutas, tratando de distribuir la carga lo más parejo posible.

* ***Algoritmos dinámicos***

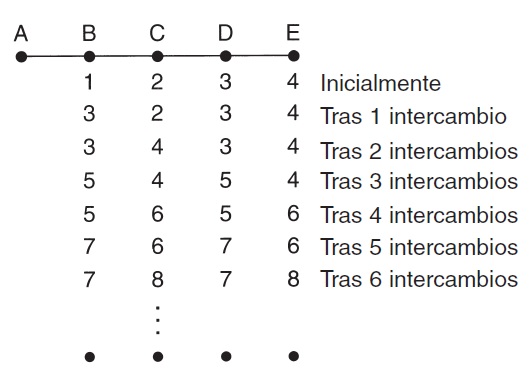
Enrutamiento por vector de distancia

Operan haciendo que cada enrutador mantenga una tabla o vector que da la mejor distancia conocida a todos los destinos y la línea que se puede usar para llegar ahí. Estas tablas se actualizan intercambiando información con los vecinos. Es un algoritmo lento.

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



Problema: *“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

Es la solución al problema anterior, en el que cada tabla solo tendrá información de los vecinos y no de todos los nodos. La ventaja con el algoritmo anterior es que un enrutador le pasa información sólo de sus vecinos a todos los enrutadores.

Cada enrutador debe:

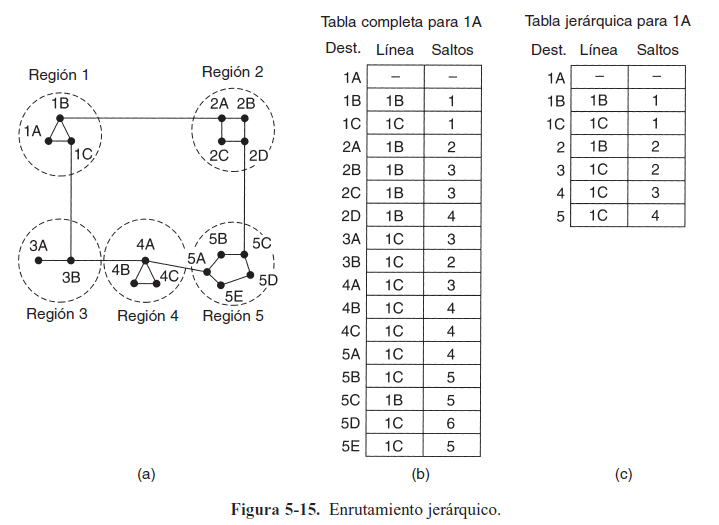
1. Descubrir a sus vecinos y conocer sus direcciones de red.
2. Medir el retardo o costo para cada uno de sus vecinos.
3. Construir un paquete que indique todo lo que acaba de aprender.
4. Enviar este paquete a *todos* los demás enrutadores.
5. Con el paquete que recibe, calcular la ruta más corta a todos los demás enrutadores utilizando Dijkstra.

Enrutamiento jerárquico

A medida que crece el tamaño de las redes, también crecen las tablas de enrutamiento, las que consumen memoria del enrutador y necesitan más tiempo de CPU y más ancho de banda para analizarlas.

Con el enrutamiento jerárquico, los enrutadores se dividen en ***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.

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. Si el enrutamiento es jerárquico, hay entradas para todos los enrutadores locales, 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 y el resto del tráfico remoto viaja por la línea 1C-3B.



Las ganancias de espacio en las tabla traen una desventaja 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.

**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.

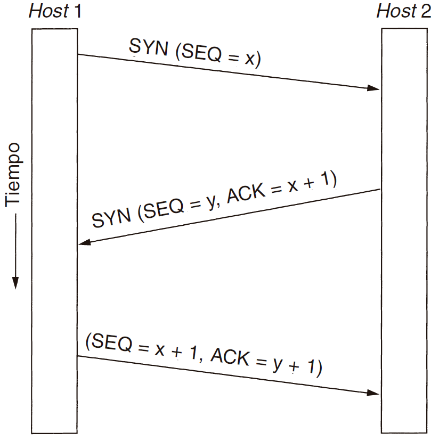
Todas las conexiones TCP son de dúplex total y de punto a punto. No soporta la multidifusión ni la difusión.

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.