CAPA DE ENLACE

Terminología (1/12)

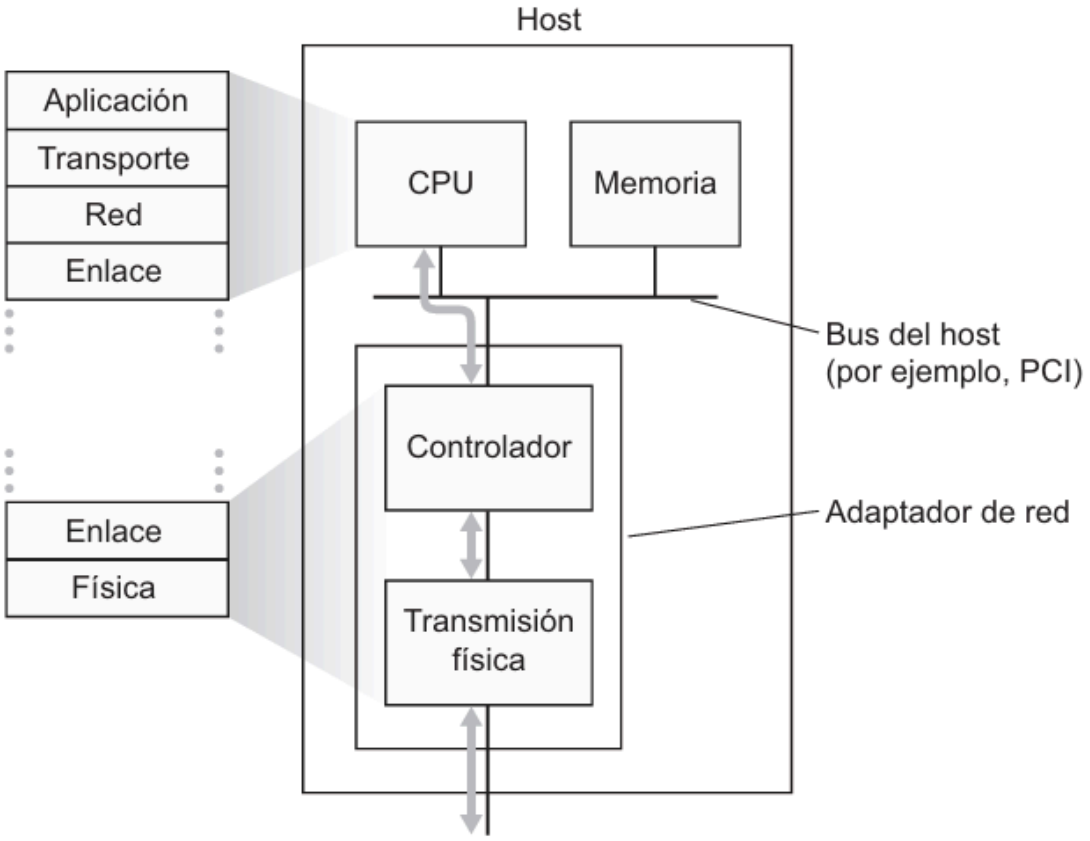
Hosts y routers: **Nodos**, es lo mismo en esta capa que es cada cosa.
Canales de comunicación entre nodos: **Enlaces**.
Unidad de la capa: **Datagrama.**

Servicios proporcionados (2/12)

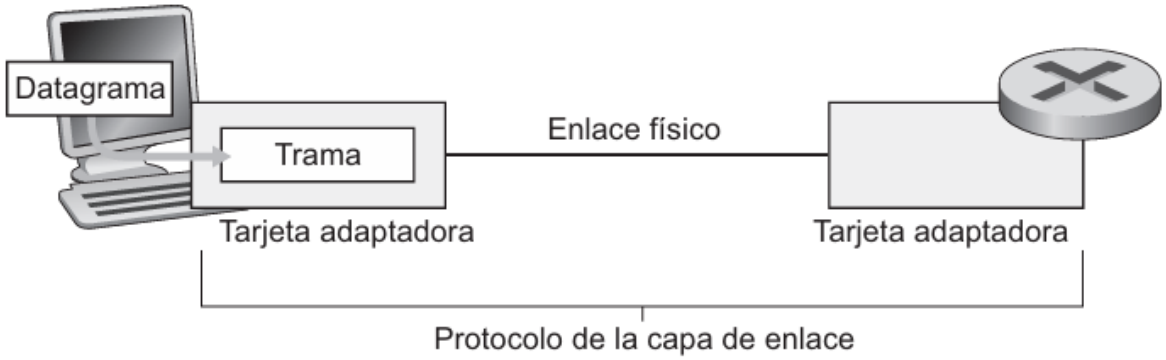
- Define formato de los paquetes que se intercambia entre los nodos en los extremos de los enlaces y las acciones a hacer cuando se envían o reciben.
- Transportar datagramas entre nodos sin importar el protocolo (Ethernet, PPP, WAN).
- Entramado: Cada trama de la capa de enlace (suele) encapsula un datagrama de red, cada trama consta de un campo de datos, y una cabecera.
- Acceso al enlace: Mediante el protocolo MAC (Medium access control) se especifican las reglas para transmitir una trama a través de un enlace.
- Entrega fiable: Los protocolos de enlace que tiene más porcentaje de fallos pueden implementar entrega fiable, los datagramas se entregan sin que se produzcan errores, suele implementarse mediante reconocimientos y retransmisiones. En protocolos con baja tasa de errores como enlaces de fibra no suelen tenerlo.
- Control de flujo: Los nodos tiene capacidad de almacenamiento limitado en su buffer de tramas, por tanto el protocolo de enlace puede proporcionar un mecanismo para controlar esto.
- Detección de errores: Se proporcionan bits de detección de errores. La detección en la capa de enlace suele ser más sofisticada y se implementa en hardware.
- Corrección de errores: El receptor además de detectar bits erróneos determina donde se han producido los errores y los corrige, algunos protocolos implementan esto.
- Semiduplex y full duplex: Los nodos pueden transmitir paquetes en ambas direcciones, sin embargo no necesariamente al mismo tiempo.

Donde se implementa la capa de enlace (3/12)

Se suele implementar en un adaptador de red, a veces llamada tarjeta de interfaz de red (NIC), el corazón de la tarjeta es el controlador de la capa de enlace que es un chip que solamente sirve para implementar servicios de la capa de enlace.



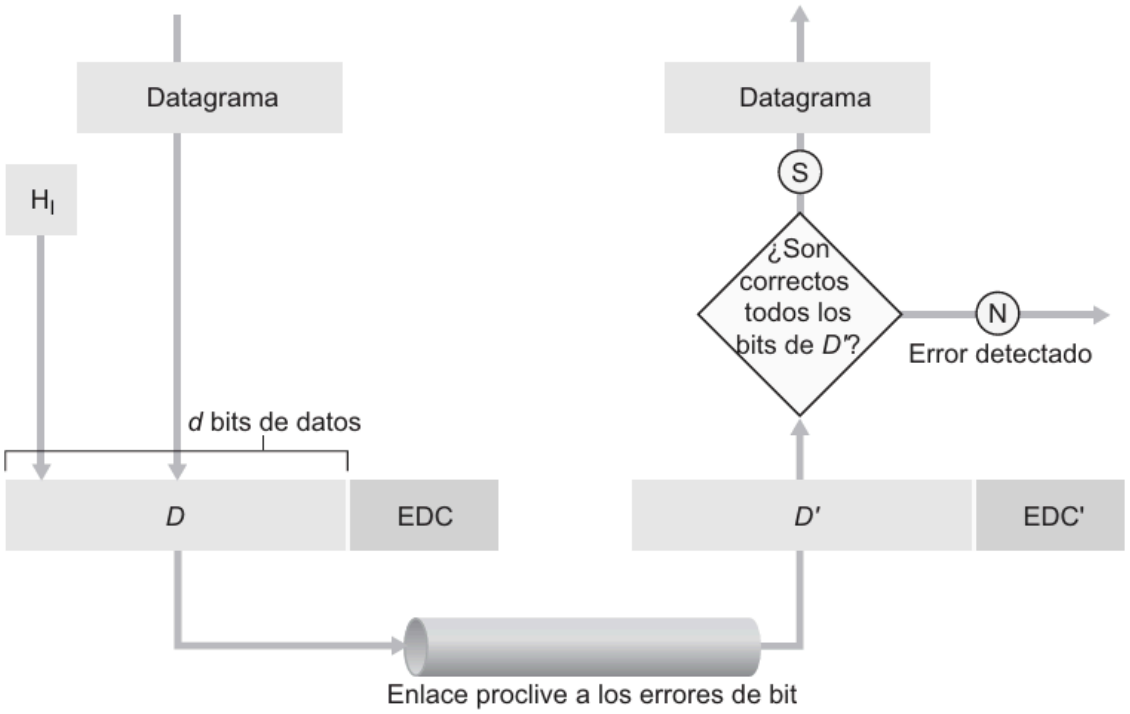
La imagen anterior muestra una adaptador de red conectado a un bus del host, de modo que el host lo trata como cualquier otro dispositivo de E/S. Aunque la mayoría de la capa está implementada en hardware, la parte de software implementa cosas como la recepción de datagramas, recibiendo interrupciones procedentes del controlador como recepción de tramas.



La imagen anterior muestra las tarjetas adaptadores de emisión y recepción, dado que la funcionalidad principal de la capa está en los controladores, estos adaptadores son unidades semi autónomas para transferencia de tramas. Hay algunas funcionalidades de capa de red y transporte en los controladores de enlace, lo que beneficia la velocidad de comprobación de checksum ya que se hace por hardware, pero rompe la división por capas, algo que es polémico.

Corrección de errores (4/12)

Para los enlaces donde suelen haber errores, lo que se hace es complementar los bits (D) con bits de detección y corrección (EDC).
Los datos a proteger suelen incluir el datagrama de red y la información de direccionamiento de enlace, números de secuencia y otros campos de la cabecera de enlace. D y EDC son enviados al receptor en un trama.
Esta técnica permite detectar errores en los bits, aunque no siempre.



Técnicas de comprobación

Cuanto más sofisticada la técnica suele tener más tasa de detección de errores.

Comprobaciones de paridad

Esta es la forma más simple de detección de errores:
Consiste en agregar un **bit de paridad** a un conjunto de bits de información.

- **Paridad par**: Se elige el bit de paridad de manera que el total de **1s** en el mensaje (incluyendo el bit de paridad) sea un número **par**.
- **Paridad impar**: Se elige el bit de paridad de manera que el total de **1s** sea un número **impar**.

El receptor verifica la paridad contando los **1s** en el mensaje recibido. Si la paridad no coincide con la esperada, significa que hubo un error en la transmisión. Ej: **0111000110101011** 1, el 1 final es el bit de paridad.

¿Qué pasa si los errores de bit hacen que haya una cantidad par de bits? No se detecta, esto es "aceptable" si hay pocos errores en el canal, pero los errores suelen existir como "rafagas" y este método no es muy bueno para comprobarlos.

Normalmente en no es un solo bit de paridad, sino que se utiliza **paridad bidimensional** donde hay un bit de paridad por cada columna y fila, si solo se produjo un error permite saber dónde fue y corregirlo (esto se debe que falla la paridad de la fila y la columna) y si fueron 2 puede detectarlos pero no corregirlos.

La capacidad del receptor para detectar y corregir errores a la vez se conoce con el nombre de Corrección de errores hacia adelante (FEC, Forward Error Correction). Se suelen usar en CD y reducen la cantidad de retransmisiones por el emisor y evita retardos de propagación de ida y vuelta para el NAK.

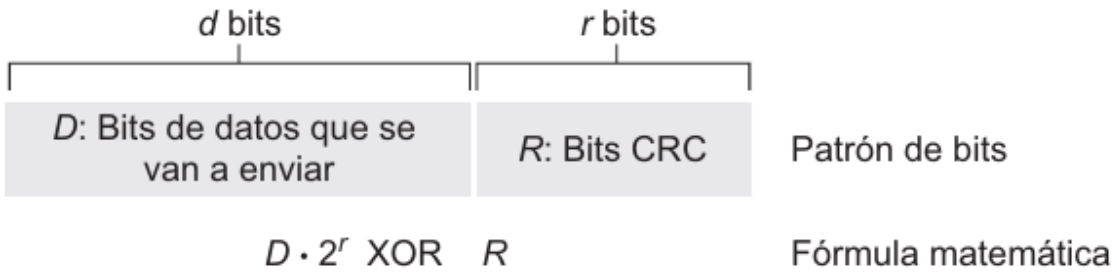
Checksum

Se trata a los bits de datos como una secuencia de enteros de K bits, una forma fácil de comprobarlos es sumar estos bits y la suma resultante serán los bits de errores.
Internet usa una comprobación de este estilo, los bytes de datos se tratan como enteros de 16 bits y se suman. Se utiliza el complemento a 1 de esta suma para formar la suma de comprobación de Internet que se incluye en la cabecera del segmento.
Estos métodos requieren poca sobrecarga de paquete. Toma poco espacio y es rápido de calcular en software, sin embargo la capa de enlace suele usar comprobación de redundancia cíclica ya que se implementa en hardware.

Comprobación de redundancia cíclica (CRC)

Una detección ampliamente usada es la basada en códigos de comprobación de redundancia cíclica (CRC, Cyclic Redundancy Check). Se los conoce también como códigos polinómicos, donde vemos la cadena de bits como un polinomio con coeficientes 0 o 1.

- Emisor y receptor deben acordar un patrón de r+1 bits llamada generador (G), Imponemos la condición de que el bit más significativo sea 1.
- Para una determinada secuencia de datos D, el emisor selecciona r bits adicionales, R, y los añadirá (concatenada) a D de modo que d+r sea divisible por G (sin resto) usando aritmética del módulo 2.
- El receptor entonces divide d+r bits entre G, si el resto es distinto de 0 se sabe que hay un error.



Todos los cálculos de CRC usan aritmética de módulo 2, sin acarreo, donde la suma y resta son idénticas o equivalentes a un XOR.

La multiplicación y la división son iguales que en aritmética en base 2, excepto porque las sumas y restas necesarias se llevan a cabo sin acarreos.

Para que el emisor calcule R debemos encontrar una secuencia R tal que exista que cumpla:

$$D \quad 2^r \text{ XOR } R = nG$$

Aplicando XOR a ambos lados tenemos:

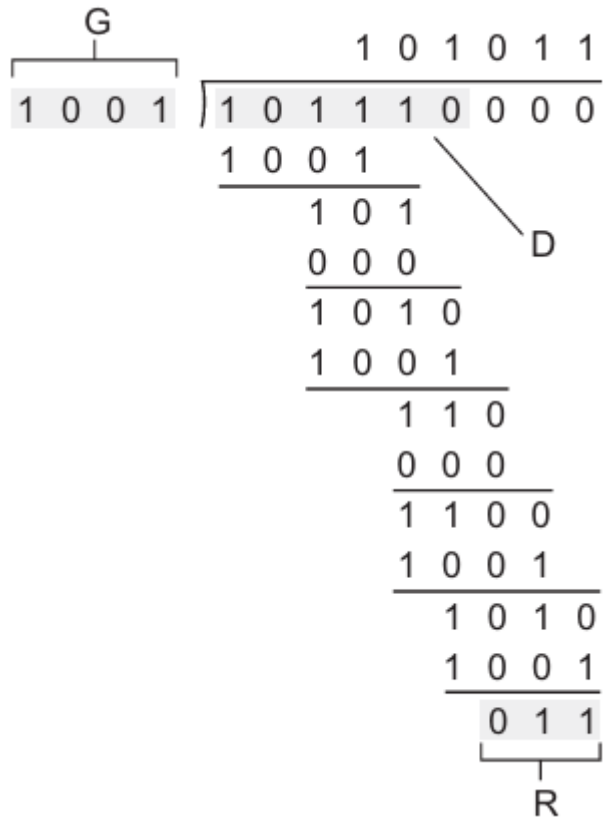
$$D \quad 2^r = nG \text{ XOR } R$$

Esta ecuación nos dice que si dividimos D 2^r entre G el valor del resto será R, por tanto podemos calcular R como:

$$R = \text{resto}\ \frac{D \cdot 2^r}{G}$$

Los generadores fueron preseleccionados para 8, 12, 16 y 32 bits:

GCRC-32= 10000010011000001000111011011011 1



La imagen anterior es un cálculo de CRC, donde sacamos el que codigo del generador usado para dividir los bits será 011, también podemos ver que la entrada de bits D es ampliada porque el generador es 4 bits.

Protocolos de acceso múltiple (protocolos MAC) (5/12)

Existen dos tipos de enlaces de red: **punto a punto** y **de difusión (broadcast)**. Los enlaces **punto a punto** conectan un único emisor con un único receptor y utilizan protocolos como **PPP** y **HDLC**. En cambio, los enlaces **de difusión**, empleados en **Ethernet** y redes **LAN inalámbricas**, permiten que múltiples nodos compartan un mismo canal, lo que genera el problema de **acceso**

múltiple, ya que las transmisiones simultáneas pueden colisionar y volverse ininteligibles.

Para coordinar el acceso al canal, existen **protocolos de acceso múltiple**, clasificados en **protocolos de particionamiento del canal**, **protocolos de acceso aleatorio** y **protocolos de toma de turnos**. Un protocolo ideal asignaría a cada nodo un **ancho de banda R/M** (siendo R la velocidad de transmisión y M la cantidad de nodos), además de ser **descentralizado y simple**.

Protocolos de particionamiento de canal

Hay 2 técnicas para dividir el canal, TDM multiplexación por división en el tiempo o multiplexación por división de frecuencia FDM.

TDM divide el tiempo en marcos y subdivide el marco en N particiones de tiempo o time frames, y asigna cada partición a uno de los nodos, cada vez que un nodo tenga que enviar paquete transmiten los bits del paquete durante su partición de tiempo asignada dentro del marco, normalmente el tamaño del particion debe permitir el envío de 1 solo paquete.

- Es equitativo.
- Alcanza R/M.
- Si solo hay uno nodo que quiere transmitir, su promedio es R/M siempre.
- Los nodos deben esperar turnos.

FDM por otro lado divide el canal en frecuencia de R/M de ancho de banda, por tanto crea M canales y asigna cada frecuencia a los M nodos. Comparte ventajas y desventajas de TDM.

Un tercer protocolo es el Acceso múltiple por división de código, (CDMA) donde cada nodo asigna un código diferente a cada nodo, y cada nodo entonces utiliza su código único para codificar los bits de datos a enviar, si estos códigos son correctamente elegidos permite que los nodos transmitan a la vez y que los receptores lso decodifiquein aunque haya colisiones.

Este protocolo anterior está muy relacionado a las transmisiones inalámbricas, por ejemplo telefonía.

Protocolos de acceso aleatorio

Otra forma de coordinar el acceso, todos los nodos transmiten a máxima velocidad y al colisionar esperan un toque (un tiempo aleatorio) y vuelven a transmitir. Hay un montón de protocolos pero los más utilizados son los ALOHA y los CSMA. Por ejemplo CSMA se utiliza en Ethernet.

ALOHA con particiones

NO APARECE EN EL RESUMEN DE LA CÁTEDRA, es uno de los más simples, suposiciones para el uso de este:

- Todas las tramas tienen exactamente L bits
- El tiempo está dividido en particiones de L/R segundos, es decir cada partición equivale al tiempo que se tarda en transmitir una trama.
- Los nodos comienzan su transmisión sólo al inicio de las particiones.
- Los Nodos están sincronizados, saben cuando comienzan las particiones
- Si 2 o más tramas colisionan entonces todos detectan la colisión.

Funciona así, asumiendo que p es un número entre 0 y 1:

- Para enviar una nueva trama espera hasta el comienzo de la siguiente partición y transmite toda la trama.
- Si no hay colisión anduvo bien y se preparar para la siguiente si tiene.
- Si hay colisión retransmitirá su trama en cada partición posterior con una probabilidad p hasta que se pueda hacer.

Aloha es descentralizado, con transmisión de velocidad máxima, requiere sincronización entre nodos. Sin embargo si colisionan no es eficiente suponiendo que tenemos un cantidad de nodos que tiende al infinito, la eficiencia es de 37%.
El protocolo original ALOHA puro no usaba particiones y sin sincronización.

Acceso múltiple con sondeo de portadora (CSMA)

El CSMA se basa en dos reglas fundamentales:

1. **Sondeo de portadora:** Antes de transmitir, un nodo escucha el canal. Si detecta actividad, espera un tiempo aleatorio y vuelve a verificar.
2. **Detección de colisiones (CSMA/CD):** Si un nodo detecta que su transmisión está interfiriendo con otra, detiene la transmisión y reintente más tarde.

Estas reglas se aplican para CSMA y CSMA/CD.

Aunque los nodos realizan sondeo de portadora, aún pueden ocurrir colisiones debido al **retardo de propagación**. Si un nodo inicia transmisión y su señal no ha llegado a otro nodo distante, este último puede asumir que el canal está libre y comenzar a transmitir, causando una colisión.

CSMA vs. CSMA/CD

- **CSMA:** No detecta colisiones; los nodos pueden seguir transmitiendo incluso si hay interferencias.
- **CSMA/CD (usado en Ethernet):** Detecta colisiones y detiene la transmisión inmediatamente para evitar desperdicio de ancho de banda.

El **rendimiento de CSMA/CD** mejora porque evita transmitir tramas dañadas en su totalidad, reduciendo el tiempo perdido en colisiones.

Protocolos de toma de turnos

Protocolo de sondeo

En este protocolo, un **nodo maestro** controla la transmisión mediante **sondeo en turno rotatorio (round robin)**. Envía un mensaje a cada nodo para indicarle cuándo y cuántas tramas puede transmitir. Una vez completada la transmisión, el maestro pasa al siguiente nodo, detectando el fin de cada turno mediante la monitorización del canal.

Este método **elimina colisiones y particiones vacías**, pero presenta desventajas:

- **Retardo de sondeo**, el tiempo necesario para que el maestro autorice a un nodo a transmitir.
- **Punto único de falla**, si el maestro falla, todo el sistema deja de funcionar.

Un ejemplo de este protocolo es **Bluetooth**.

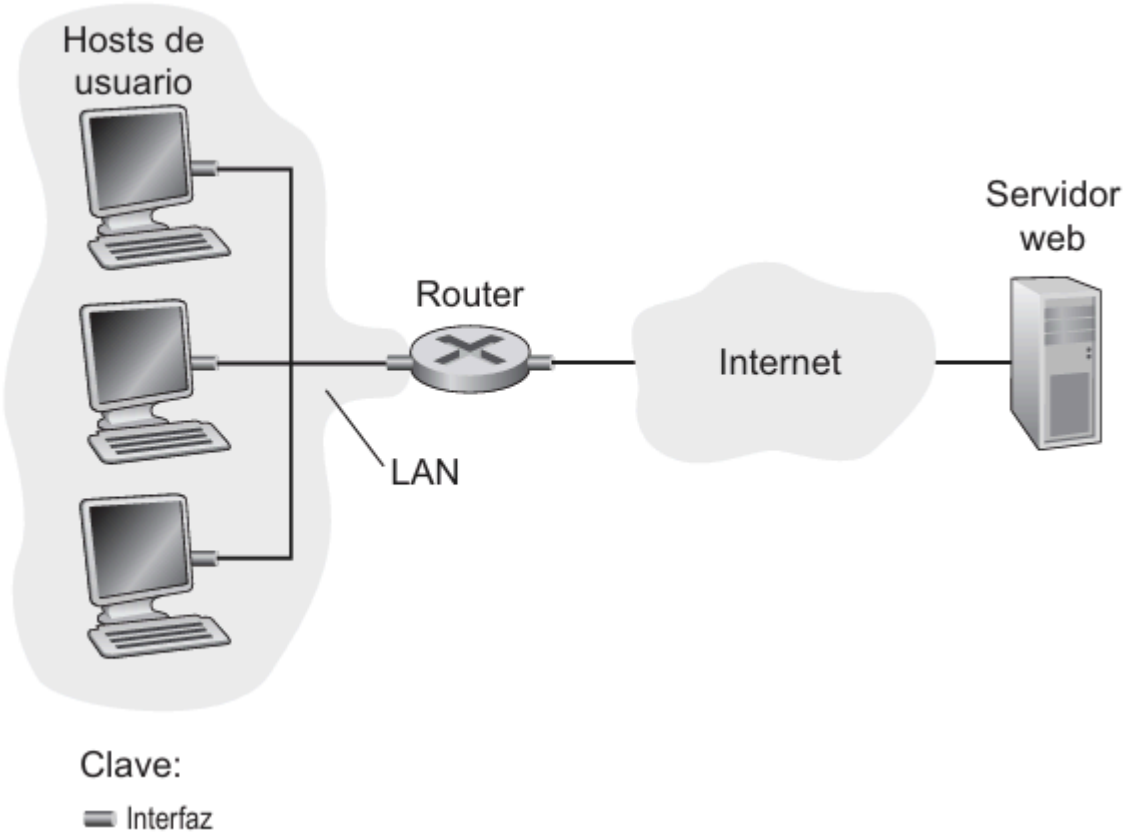
Protocolo de paso de testigo

En este protocolo, **no hay un nodo maestro**, sino una **trama especial llamada testigo o token**, que los nodos intercambian en un orden fijo. Cuando un nodo recibe el testigo, puede transmitir hasta un límite predefinido de tramas; si no tiene nada que enviar, simplemente lo pasa al siguiente nodo.

Este método es eficiente en la gestión del acceso al canal, pero tiene una desventaja: si un nodo falla o retiene el testigo indebidamente, **todo el canal puede quedar inutilizable**.

LAN (Redes de área local) (6/12)

Los **protocolos de acceso múltiple** se usan ampliamente en **LAN** (redes de área local), que conectan computadoras dentro de una zona geográfica limitada, como un edificio. Estas redes, comunes en entornos universitarios y corporativos, permiten conexiones rápidas y eficientes.



En los años 80, existían dos tecnologías principales para **LAN**:

1. **Ethernet**, basada en acceso aleatorio (vemos más tarde).
2. **Redes de paso de testigo**, como **Token Ring** y **FDDI**.

En una **LAN Token Ring**, los **N nodos** (hosts y routers) están conectados en un **anillo**, donde un **testigo** circula en orden predefinido. Un nodo sólo transmite cuando posee el testigo. La trama enviada recorre todo el anillo, permitiendo al nodo destino leerla a medida que pasa. Finalmente, el **nodo emisor** es responsable de eliminar la trama del anillo, evitando congestión en la red.

Por su parte FDDI se diseñó para LAN más grandes ya que es poco eficiente que una trama se propague por todo el circuito y devuelta al nodo emisor, FDDI hace que el nodo destino elimine la trama

Direccionamiento de la capa de enlace (7/12)

Las direcciones de la capa de enlace reciben distintos nombres, como **dirección LAN, dirección física y dirección MAC**, siendo esta última la más utilizada.

Las **direcciones MAC** tienen **6 bytes** y se representan en **hexadecimal**. Fueron diseñadas para ser **permanentes** y **únicas**, por lo que dos adaptadores no pueden compartir la misma. Para garantizar esto, los fabricantes deben comprar un rango de direcciones a la **IEEE**.

A diferencia de las direcciones IP, las direcciones MAC tienen una estructura **plana y no jerárquica**, y **nunca cambian**.

Cuando un adaptador de un emisor quiere enviar una trama a otro adaptador de destino, inserta la dirección MAC del de destino en la trama y luego la envía a través de la red LAN. Si la red LAN es una LAN de difusión la trama será recibida y procesada por todos los demás adaptadores de la LAN, estos se fijarán si la dirección MAC es la propia y en cuyo caso extrae el datagrama de la misma.
Si se quiere enviar una trama a todas las direcciones de la red podemos usar la dirección de broadcast FF-FF-FF-FF-FF-FF.

Protocolo de resolución de direcciones (ARP) (8/12)

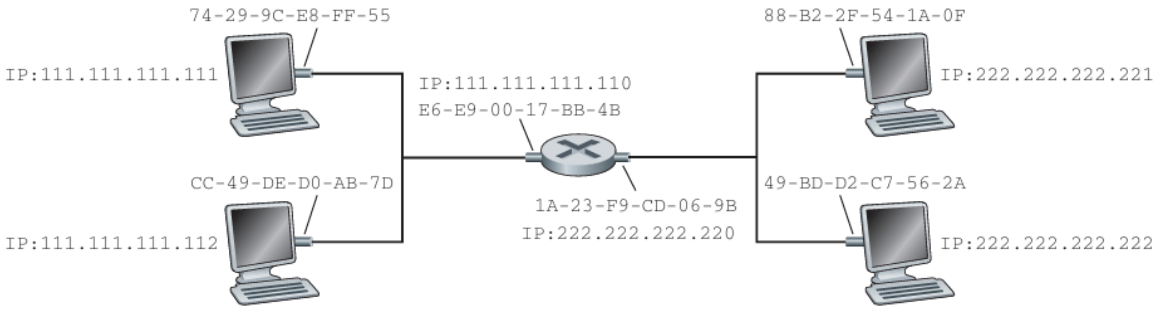
Cada nodo mantiene una **tabla ARP**, que asocia **direcciones IP con direcciones MAC** e incluye un **TTL** para eliminar entradas obsoletas.

Cuando un nodo necesita enviar un **datagrama IP** dentro de una subred, primero debe conocer la **dirección MAC** del destino. Si la correspondencia no está en la tabla ARP, se envía un **mensaje ARP de consulta** en una **trama de difusión**. Todos los nodos verifican si su IP coincide con la consultada; si es así, responden con un **mensaje ARP de respuesta** en una **trama estándar**, incluyendo su dirección MAC.

Teniendo en cuenta lo anterior, ARP es un protocolo que se construye automáticamente y se limpia por su cuenta.
ARP es un protocolo entre la capa de enlace y capa de red y no se ajusta a la pila de protocolos simple ya que incluye cosas de ambos.

Envío de datagramas fuera de la subred

Seguimos usando ARP para esto, en este caso tenemos un ejemplo:



Diferencia entre hosts y routers

- **Hosts**: Tienen **una única dirección IP** y un **único adaptador de red**.
- **Routers**: Poseen **múltiples interfaces**, cada una con su propia **dirección IP, adaptador y módulo ARP**.

Estructura de la red

- **Subred 1**: Usa direcciones **111.111.111.0**.
- **Subred 2**: Usa direcciones **222.222.222.0**.
- Un **router** conecta ambas subredes con **una dirección IP en cada una**.

Proceso de envío de un datagrama entre subredes

1. Un **host en la Subred 1** (**111.111.111.111**) quiere enviar un **datagrama** a un **host en la Subred 2** (**222.222.222.222**).
2. Como están en **subredes diferentes**, el datagrama se envía primero al **router**.
3. La **dirección MAC de destino** en la trama **no es la del host final**, sino la del **router** (interfaz en la Subred 1).
4. Si el host no conoce la MAC del router, usa **ARP** para obtenerla.

Reenvío del datagrama en el router

1. El **router** recibe el datagrama y consulta su **tabla de reenvío** para determinar la **interfaz de salida**.

2. **Encapsula** el datagrama en una nueva **trama** con la dirección MAC correspondiente a la **Subred 2**.
3. Antes de enviarlo, usa **ARP** para obtener la **dirección MAC del host destino** (**222.222.222.222**).

Conclusión

- **ARP se usa dos veces**:
    1. Para obtener la **MAC del router**.
    2. Para obtener la **MAC del destino final**.
- El **datagrama viaja primero al router**, que lo **reenvía a la subred de destino**.

Ethernet (9/12)

Ethernet es la **tecnología de área local (LAN)** más exitosa y utilizada debido a varios factores:

1. **Establecimiento temprano**: Ethernet se consolidó rápidamente, siendo conocida por administradores de red desde sus inicios.
2. **Costo y complejidad**: Es **más barata y menos compleja** en comparación con otras tecnologías como **Token Ring**, **FDDI** y **ATM**.
3. **Buena velocidad**: Ofrece velocidades **similares o superiores** a las de sus competidores.
4. **Hardware accesible**: El **hardware de Ethernet** se volvió **económico y fácil de usar**, lo que permitió su expansión incluso en redes domésticas.

Ethernet fue **creada en 1970** y a lo largo del tiempo se ha ido adaptando. En la década de 1990, **Ethernet reemplazó otras redes LAN**, adoptando una topología en **estrella** mediante el uso de **hubs** (dispositivos que replican y envían señales).

En **2000**, el **hub** fue reemplazado por un **conmutador (switch)**, que permite un rendimiento mejorado al eliminar las colisiones de datos.

Trama de ethernet

En una trama Ethernet, los campos principales son:

1. **Campo de datos** (46 a 1500 bytes): Aquí se coloca el **datagrama IP**. La **unidad máxima de transmisión (MTU)** de Ethernet es de **1500 bytes**, por lo que si el datagrama es más grande, se fragmenta para ajustarse a este límite.

2. **Dirección de destino** (6 bytes): La dirección **MAC del destino**.
3. **Dirección de origen** (6 bytes): La dirección **MAC del origen**.
4. **Campo de tipo** (2 bytes): Este campo se utiliza para **multiplexar los protocolos de capa de red**, indicando qué protocolo de la capa 3 se está utilizando (por ejemplo, **IPv4** o **IPv6**).
5. CRC (4 bytes)
6. El **preámbulo** en una trama Ethernet consta de **8 bytes** y cumple con funciones clave de sincronización y notificación:
    - **Primeros 7 bytes**: Cada byte tiene el valor **10101010**, lo que sirve para **sincronizar los relojes** de los adaptadores de recepción con el reloj del emisor y para **"despertar"** a los adaptadores para que estén listos para recibir datos.
    - **Octavo byte**: El valor es **10101011**. Los primeros dos **1s consecutivos** en este byte sirven para **alertar al adaptador B** de qué va a llegar información **importante** (es decir, el comienzo de la trama de datos).

Ethernet usa **transmisión en banda base**, lo que significa que los adaptadores envían señales digitales directamente al canal de difusión, sin desplazarlas a otra banda de frecuencias, como hacen tecnologías como **ADSL** o **módems por cable**.

Codificación Manchester:

- En Ethernet (como **10BASE-T**), se emplea la **codificación Manchester** para transmitir los datos.
    - Un **1** se representa por una transición de nivel **alto a bajo**.
    - Un **0** se representa por una transición de nivel **bajo a alto**.
- **¿Por qué Manchester?**
    - Se utiliza para compensar la falta de sincronización perfecta entre los relojes de los adaptadores de emisor y receptor.
    - La **transición en medio de cada bit** ayuda al receptor a **sincronizar su reloj** con el del emisor, lo que le permite identificar correctamente los valores de los bits (1 o 0).

La **codificación Manchester** es una característica de la **capa física** del modelo OSI, pero es crucial en Ethernet para asegurar que la transmisión de datos sea confiable, especialmente cuando los relojes de los dispositivos no están sincronizados.

Servicio sin conexión no fiable

Cuando un adaptador recibe una trama ejecuta comprobación CRC de la trama pero no envía reconocimiento ni reconocimiento negativos, por tanto el adaptador que emite no sabe qué ocurrió. Por esto es barato y rápido, pero a su vez para saber si hay huecos en la información, la comprobación se tendrá que hacer a nivel de TCP o QUIC.

CSMA/CD: protocolo de acceso múltiple de Ethernet

Si ethernet usa un hub (concentrador) y no un switch (conmutador) como modo de interconexión de nodos, este necesita un protocolo de acceso múltiple, ethernet usa CSMA/CD.

Repaso de su funcionamiento:

1. Cualquiera transmite cuando le pinta (sin particion de tiempo).
2. Si hay otro transmitiendo no transmite, esto se llama sondeo de portadora.
3. Si hay colisión se aborta la transmisión, es decir detecta colisiones.
4. Para retransmitir se espera un tiempo random antes de seguir la transmisión.

CSMA/CD es mejor que ALOHA con particiones en entornos LAN. Casi del 100% si el retardo máximo de propagación es muy pequeño.

Dentro de un adaptador específicamente el protocolo funciona así:
Preparación de la Trama

- El adaptador recibe un **datagrama** de la capa de red, lo encapsula en una **trama Ethernet** y lo almacena en su **buffer**.

Escucha del Canal

- Si el canal está **libre** durante **96 periodos de bit**, comienza la transmisión.
- Si el canal está **ocupado**, espera hasta que esté libre y luego transmite.

Monitoreo de Colisiones

- Mientras transmite, el adaptador **supervisa** si hay señales de otros dispositivos.
- Si **no detecta colisiones**, la transmisión se completa con éxito.

Detección de Colisión

- Si detecta una **colisión**, **detiene** la transmisión y envía una **señal de interferencia (jam)** de **48 bits** para informar a otros dispositivos.

Backoff Exponencial

- Tras una colisión, el adaptador **espera un tiempo aleatorio** antes de reintentar la transmisión.
- El tiempo de espera se calcula con:
    - K es un número aleatorio en {0,1,2,...,2m−1}, donde m=min(n,10).
    - Luego espera K×512 periodos de bit antes de reintentar en el paso 2.

Algoritmo de backoff exponencial

- Un **periodo de bit** en Ethernet a 10 Mbps es **0.1 µs**.
- Tras una colisión, se elige un número aleatorio **K** que determina el tiempo de espera antes de reintentar:
    - **1ª colisión:** K ∈ {0, 1} → espera hasta **51.2 µs**.
    - **2ª colisión:** K ∈ {0, 1, 2, 3}.
    - **3ª colisión:** K ∈ {0, 1, 2, ..., 7}.
    - **Hasta la 10ª colisión:** K ∈ {0, 1, 2, ..., 1023}.
- A medida que aumentan las colisiones, el rango de K crece **exponencialmente**, reduciendo la probabilidad de nuevas colisiones.
- Este mecanismo ayuda a evitar la sobrecarga en redes congestionadas.

Eficiencia de ethernet

Hay una formulita muy épica que explica porque cuando la propagación tiende a 0, da 1 la eficiencia:

$$\text{Eficiencia} = \frac{1}{1 + 5d_{\text{prop}}/\, d_{\text{trans}}}$$

Conmutadores (switches) (10/12)

En las **redes Ethernet modernas LAN**, se utiliza una **topología de estrella** con un **conmutador central (switch)**.

Funcionamiento del conmutador (switch)

- Es **transparente** para los nodos de la red: los dispositivos envían datos sin preocuparse por la ruta, ya que el **switch se encarga de redirigirlos** correctamente.

- Es **inteligente**: analiza la **dirección MAC** de cada trama y la envía solo por el **enlace correcto**, evitando colisiones y mejorando la eficiencia.
- Posee **buffers en las interfaces de salida**: si la **velocidad de entrada** de las tramas es mayor que la capacidad del enlace de salida, el switch almacena temporalmente los datos en **buffers** para evitar pérdida de información. Por esto mismo cada enlace es un dominio de colisión y permite transmisiones simultáneas.
- Es **plug & play, self learning**, es decir aprende sobre la marcha a medida que va recibiendo tramas, guardando en su tabla el par emisor y ubicación.

Esto hace que los conmutadores sean mucho más eficientes que los antiguos **hubs**, que simplemente replicaban las señales a todos los dispositivos de la red.

Reenvío y filtrado

Es una función del switch que se hace con su tabla interna, que contiene la MAC del emisor, la interfaz a donde va y cuando fue incluido la entrada en la tabla.
Imaginemos que llega una trama para dirección DD-DD-DD-DD-DD-DD desde la interfaz X:

- Si no tenemos ninguna entrada de DD-DD-DD-DD-DD-DD la trama se reenvía a todas las interfaces excepto la de llegada (X).
- Si existe la DD... en la tabla pero proviene de la interfaz (X) no hace nada.
- Si existe DD... de una interfaz no X se reenvía por la interfaz donde está.

Autoaprendizaje

1. Inicialmente, la tabla del conmutador está vacía.
2. Para cada trama entrante se guarda en la tabla la MAC, la interfaz de donde vino y la hora actual.
3. Borra la entrada pasado TTL.

Métodos de Conmutación

- Store and Forward (Almacena y Envía): Lee toda la trama y chequea CRC. Más seguro.
- Fragment Free (Libre de Fragmentos): Lee los primeros 64 bytes.
- Cut-through (de corte): Lee hasta la dirección destino. Más rápido.

Propiedades de la conmutación en la capa de enlace

1. **Eliminación de colisiones**
    - No hay desperdicio de ancho de banda por colisiones, ya que los conmutadores almacenan las tramas en búfer.
    - La **tasa de transferencia agregada** es la suma de las tasas de todas las interfaces del conmutador, mejorando el rendimiento.
2. **Soporte para enlaces heterogéneos**
    - Permite mezclar distintos tipos de enlaces con diferentes velocidades y medios físicos.
    - Ejemplo: Un nodo puede usar **10BASE-T (10 Mbps, cobre)**, otro **100BASE-FX (100 Mbps, fibra óptica)**, y otro **1000BASE-T (1 Gbps, cobre)** sin afectar la compatibilidad.
3. **Facilidad de administración**
    - Mejora la seguridad y permite gestionar problemas de red de forma remota.
    - Puede **desconectar automáticamente** un adaptador defectuoso que envía tramas continuamente.
    - Un corte en un cable solo afecta al nodo conectado, sin afectar al resto de la red.
    - Los conmutadores recopilan estadísticas de tráfico, tasas de colisión y uso del ancho de banda, facilitando la gestión y evolución de la red.

Switches y routers

- Ambos son store-and-forward
    - Los routers son de capa de red
    - Switches son de capa de enlace
- Los routers tienen tablas de ruteo e implementan algoritmos de ruteo.
- Los switches tienen tablas de switch e implementan filtrado, algoritmo de aprendizaje.

Pros y contras de los switches:

- Plug and play.
- Altas tasas de filtrado y reenvío.
- Si hay ciclos una trama podría conmutarse infinitamente (se soluciona con un árbol de recubrimiento).
- En redes conmutadas grandes las tablas ARP son grandes y habrá mucho tráfico del protocolo.
- No hay protección contra tormentas de difusión.

Pros y contras de los routers:

- Direccionamiento jerárquico y no plano como MAC, por eso no suele haber ciclos.
- Protección mediante cortafuegos a tormentas de difusión.
- No son plug and play ya que se deben configurar las IP's y otras cosas.

VLAN

Las redes de área local virtual (VLAN) permiten dividir una red en múltiples segmentos utilizando switches. Sin embargo, sin VLAN, surgen tres desventajas:

1. **Falta de aislamiento del tráfico**: Tramas ARP y DHCP se propagan por toda la red.
2. **Uso ineficiente del hardware**: Se requiere un switch por segmento, cuando un solo switch grande podría manejar toda la red.
3. **Gestión complicada**: Si una computadora cambia de grupo, es necesario modificar el cableado físico.

Los switches compatibles con VLAN solucionan estos problemas al permitir la creación de múltiples redes virtuales sobre una única infraestructura física. El administrador asigna los puertos a diferentes VLANs, manteniendo la correspondencia en una tabla interna del conmutador.

Para la comunicación entre VLANs, una opción es conectar un puerto del switch VLAN a un router externo que pertenezca a ambas redes, permitiendo la interconexión. Generalmente, un switch VLAN ya incluye funcionalidad de enrutamiento.

VLAN Trunking es un método más escalable que consiste interconectar conmutadores, donde un puerto especial de cada conmutador (1 del conmutador izquierda y 16 del derecho) se configura como puerto troncal de interconexión, el puerto troncal entonces pertenece a todas las VLANs. (Como se pone físicamente este trunking se puede ver en diagramas de la diapositiva).

Esto nos deja con el problema de saber de qué VLAN vino una trama al llegar al puerto central, la solución para esto es que la trama sea la trama de Ethernet normal, y una **etiqueta VLAN** de 4 bytes que identifica la VLAN. El conmutador del lado emisor añade esta etiqueta la cual es analizada y eliminada por el conmutador receptor.

La etiqueta se confirma con un Identificador de protocolo de etiquetado TPID de 2 bytes con valor hexadecimal fijo de 2 bytes y un campo de información de control de etiquetado de 2 bytes.

Las VLAN se pueden no solo basar en puertos, sino en direcciones.

Cosas relacionadas a los switches que no vi en el libro

Tipos de Hubs:

- Hubs pasivos: solo envían la señal por todos los puertos restantes.
- Hubs activos: regeneran la señal, mayor alcance.
- Hubs inteligentes: pueden poseen administración, permiten detectar problemas.

Los hubs pueden detectar colisiones y generar JAMs.

Bridge: Poder adaptar entre dos protocolos de nivel de enlace o físico, pueden ser diferentes. Dividir dominio de colisión. Un switch de varios puertos es un bridge multipuerto.

Administración: Hay una sección entera sobre la administración en la segunda diapo de redes, a un switch nos podemos conectar por consola o con interfaces web, ssh, telnet, etc.
Los administradores de red deben documentar y mantener la configuración de los dispositivos de red. Deben hacer copias de seguridad del disco y del firmware del servidor.

PPP: protocolo punto a punto (11/12)

Ya mencionamos anteriormente que para canales 1 a 1 usamos este protocolo que no es de difusión. Este protocolo está definido en un RFC y tiene ciertos requisitos:

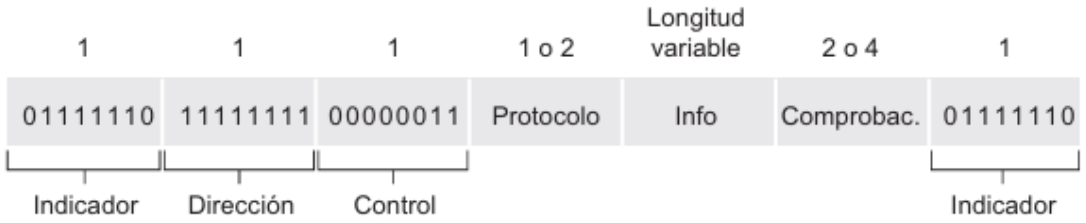
- **Entramado de paquetes:** El paquete de red debe ser encapsulado por el emisor y el receptor lo debe poder identificar el inicio y fin del paquete de la capa de red en el.
- **Transparencia:** No se puede restringir los datos de la capa de red.
- **Múltiples protocolos de capa de red:** Debe dar soporte a múltiples protocolos de red, IP, DECnet, etc para que se ejecuten sobre un mismo enlace físico de forma simultánea en una única conexión terminal a terminal y debe poder multiplexar diferentes protocolos de red sobre una unica conexion punto a punto.
- Debe **soportar muchos tipos de enlaces**.
- Debe **poder detectar errores de bit en las tramas**.
- **Pervivencia de la conexión**: De poder detectar errores en el enlace y señalar.

- **Negociaciones de capa de red**: Debe proporcionar mecanismo para que las capas de red que se estén comunicando puedan aprender o configurar las direcciones de la capa de red de cada una de ellas.
- **Simplicidad**: Debe ser simple, es lo más importante y el primero.

No debe implementar (especificado explícitamente):

- Corrección de errores
- Control de flujo
- Secuenciamiento
- Enlaces multipunto (único emisor único receptor).

Trama PPP



- Los tres primeros campos siempre tienen esos valores, ya que la RFC indica que podrán ser definidos más tarde pero nunca se hizo (hasta la fecha donde salió el libro).
- Protocolo indica el protocolo de la capa superior al cual los datos pertenecen. El receptor debe comprobar si la trama es correcta y pasar los datos al protocolo correcto.
- Información son los datos, dale amigo date cuenta. El valor predeterminado son 1.500 bytes.
- Checksum CRC standard HDLC de 2 o 4 bytes.

Problemas y rellenado de bytes

O sea digamos que aparece el indicador en medio del paquete, ¿el receptor podrá detectar el fin del paquete? Esto puede pasar por ejemplo si un protocolo de red usa ese identificador o viene en los datos.
Para resolver esto rellenamos bytes, específicamente is aparece 01111110 en cualquiera parte del paquete que no es el inicio o fin se le pone un byte de escape de control.
Si aparece el byte de escape de control random en el medio también hay que añadirle un byte de escape de control a ese de relleno para indicar que NO es un byte de escape de control. (Una verga).

MPLS (12/12)

No se si va esto, no lo vi en el resumen pero lo dejo por acá:

Conmutación de Etiquetas Multiprotocolo (MPLS)

MPLS es una tecnología que mejora la velocidad de reenvío de los routers IP al introducir etiquetas de longitud fija, complementando el enrutamiento IP en lugar de reemplazarlo. Estas etiquetas permiten que los routers reenvían tramas basándose en ellas, sin necesidad de analizar las direcciones IP de destino.

En las redes MPLS, las tramas incluyen una cabecera MPLS entre la capa de enlace (Ethernet, PPP) y la capa de red (IP). Los routers compatibles con MPLS, conocidos como "routers de conmutación de etiquetas", utilizan estas etiquetas para reenviar los paquetes rápidamente.

MPLS no solo mejora la velocidad, sino que también permite la gestión avanzada del tráfico, como la ingeniería de tráfico y la creación de rutas personalizadas. Además, se usa para la restauración rápida de rutas, implementación de redes privadas virtuales (VPN) y gestión de prioridades de tráfico.