

REDES DE COMPUTADORES Y LABORATORIO

Christian Camilo Urcuqui López, MSc



BIBLIOGRAFÍA



COMPETENCIAS

- Enuncie algunos elementos de la API de Java para la aplicación de multihilos
 - Aplique la API de Java para el desarrollo de hilos con prioridades.
 - Aplique la API de Java para la gestión de grupos de hilos.
 - Aplique la API de Java para la sincronización de hilos

PRIORIDADES DE LOS HILOS

- En Java cada hilo tiene una prioridad.
- Por defecto, un hilo hereda la prioridad de su padre.
- Los cambios de la prioridad se hacen a través del método *setPriority* el cual recibe un valor entre 0-10, entre más próximo sea a 10 el hilo tendrá más prioridad.
- El hilo con la más alta prioridad deja de ejecutarse hasta que:
 - Renuncia a su ejecución, invocando al método *yield()*.
 - Termina su ejecución o porque entra en un estado de bloqueo.
 - Un hilo de mayor prioridad entra en estado de ejecutable (porque se despierta, algún recurso que el espera ya se encuentra disponible o porque otro hilo lo despertó con *notify* o *notifyall*).
- Los hilos de alta prioridad se seleccionan de manera arbitraria por el sistema operativo. Ojo! Hay cambios en la asignación de prioridades dependiendo del sistema operativo.

GRUPOS DE LOS HILOS

- Grupos de hilos: la clase ThreadGroup
- Los grupos de hilos son utilizados para:
 - Agrupar los hilos de acuerdo con su función.
 - Trabajar simultáneamente con todos los hilos que pertenecen al grupo, es decir, poder realizar operaciones que los afecten a todos al tiempo.
- ThreadGroup(String nombre)
- ThreadGroup(ThreadGroup padre, String nombre)
String nombreGrupo= ...
ThreadGrupo g = new ThreadGrupo(nombreGrupo);

GRUPOS DE LOS HILOS

- Para adicionar o remover elementos del grupo de hilos existen un conjunto de constructores de la clase Thread, algunos de estos son:
- `Thread(ThreadGroup grupo, String nombre)`
- `Thread(ThreadGroup grupo, Runnable objetivo)`
- `Thread(ThreadGroup grupo, Runnable objetivo, String nombre)`
- **`void interrupt()`**: interrumpe todos los hilos en el grupo y en todos los hilos del grupo
- **`int activeCount()`**: retorna la cantidad de hilos activos en el grupo.
- **`ThreadGroup getParent()`**: Devuelve el padre del grupo de hilos.

SINCRONIZACIÓN DE HILOS

- Gestión de los hilos con el fin de mantener la integridad de los recursos.
- En Java, para indicar una operación (o un grupo de operaciones) sobre un objeto debe realizarse en forma sincronizada, se utiliza la etiqueta **synchronized**

```
public synchronized void setValorCuenta(int numCuenta, int valor){  
    Cuentas[numCuenta] +=valor;  
    valorTotal +=valor;  
}
```

HARDWARE DE RED

- **Enlaces punto a punto. Conectan pares individuales de máquinas.** Los paquetes pueden pasar por máquinas intermedias.
 - Si en la transmisión solo esta el emisor y receptor entonces se conoce como **unidifusión** (*unicasting*).
- **Redes de difusión. Todas las máquinas comparten el canal de comunicación;** los paquetes que envía una máquina son recibidos por todas las demás (Existe un campo en la dirección que especifica el receptor). Ejemplo, red inalámbrica.
 - Cuando el paquete es enviado a todos los nodos entonces se conoce como **difusión** (*broadcasting*).
 - Si el paquete es enviado a solo un subconjunto de máquinas, este tipo se conoce como multidifusión (*multicasting*).

LA CAPA DE ENLACE DE DATOS

“En cualquier red de difusión, el asunto clave es la manera de determinar quién puede utilizar el canal cuando tiene competencia por él.”

MAC - *MEDIUM ACCESS CONTROL*

- Los protocolos que se utilizan para determinar quién sigue en un canal multiacceso pertenecen a una subcapa de la capa de enlace de datos llamada subcapa **MAC** (**Control de Acceso al Medio**, del inglés *Medium Access Control*).
- No confundir con la dirección MAC
- Son los protocolos encargados de la asignación de canal y evitar colisiones entre los datos enviados y recibidos.

LA CAPA DE ENLACE DE DATOS

- En esta capa se aplican algoritmos para lograr una comunicación confiable y eficiente de unidades completas de información llamadas **tramas** entre dos máquinas adyacentes.
- La propiedad esencial de un canal es que los **bits deben entregarse exactamente en el mismo orden en que se enviaron.**
- Existen errores en los canales de comunicación, hay una tasa de transmisión finita y retardo de propagación.

LA CAPA DE ENLACE DE DATOS

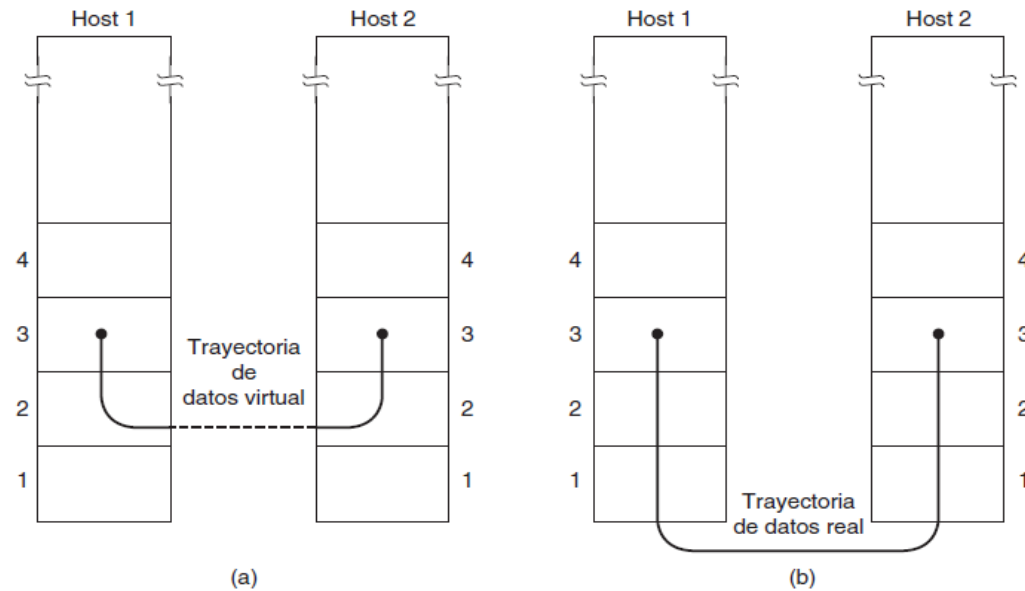
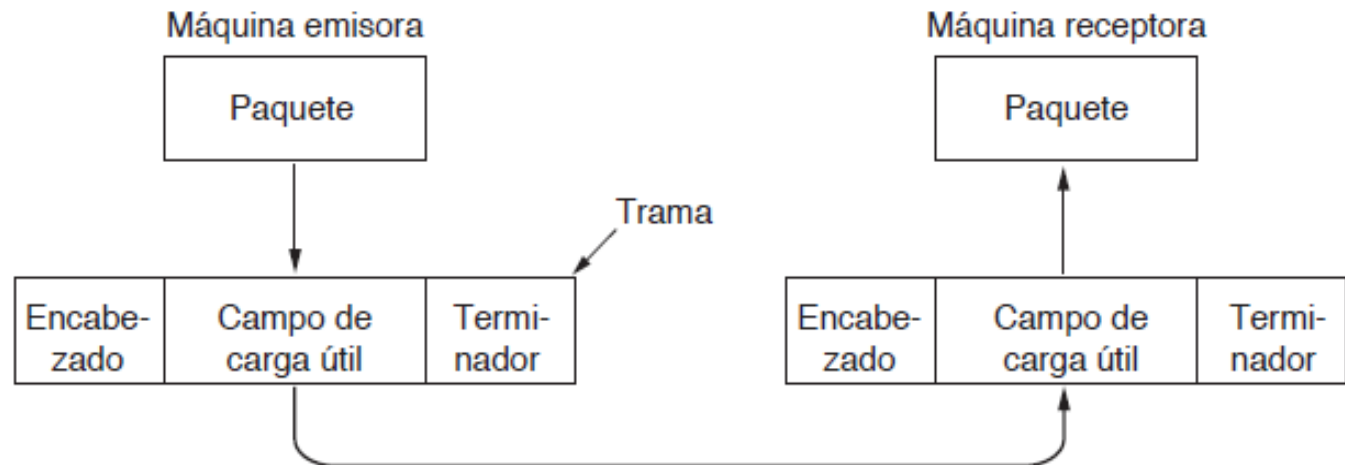


Figura 3-2. (a) Comunicación virtual. (b) Comunicación real.

LA CAPA DE ENLACE DE DATOS

- La capa de enlace de datos utiliza los servicios de la capa física para enviar y recibir bits a través de los canales de comunicación.
 - **Proporcionar a la capa de red una interfaz de servicio bien definida.**
 - **Manejar los errores de transmisión.**
 - **Regular el flujo de datos.**



LA CAPA DE ENLACE DE DATOS

- Servicios ofrecidos varían de un protocolo a otro, pero usualmente se proporcionan:
 - Servicio sin conexión ni confirmación de recepción.
 - Servicio sin conexión con confirmación de recepción.
 - Servicio orientado a conexión con confirmación de recepción.
- Actividades a tener en cuenta para proveer el servicio:
 - **Entramado**
 - Control de errores
 - Control de flujo

LA CAPA DE ENLACE DE DATOS

Entramado

Para proveer servicio a la capa de red, la capa de enlace de datos debe usar el servicio que la capa física le proporciona. Lo que hace la capa física es aceptar un flujo de bits puros y tratar de entregarlo al destino.

Si el canal es ruidoso, como en la mayoría de los enlaces inalámbricos y en algunos alámbricos, la capa física agregará cierta redundancia a sus señales para reducir la tasa de error de bits a un nivel tolerable

LA CAPA DE ENLACE DE DATOS

Entramado

1. **Conteo de bytes.**
2. **Bytes bandera con relleno de bytes.**
3. Bits bandera con relleno de bits.
4. Violaciones de codificación de la capa física.

LA CAPA DE ENLACE DE DATOS

Entramado - Conteo de bytes

- El conteo puede presentar problemas debido a un error de transmisión.
- Tampoco es útil enviar un trama de vuelta a la fuente ya que se desconoce cuántos bytes para regresar al inicio.

Suma de verificación

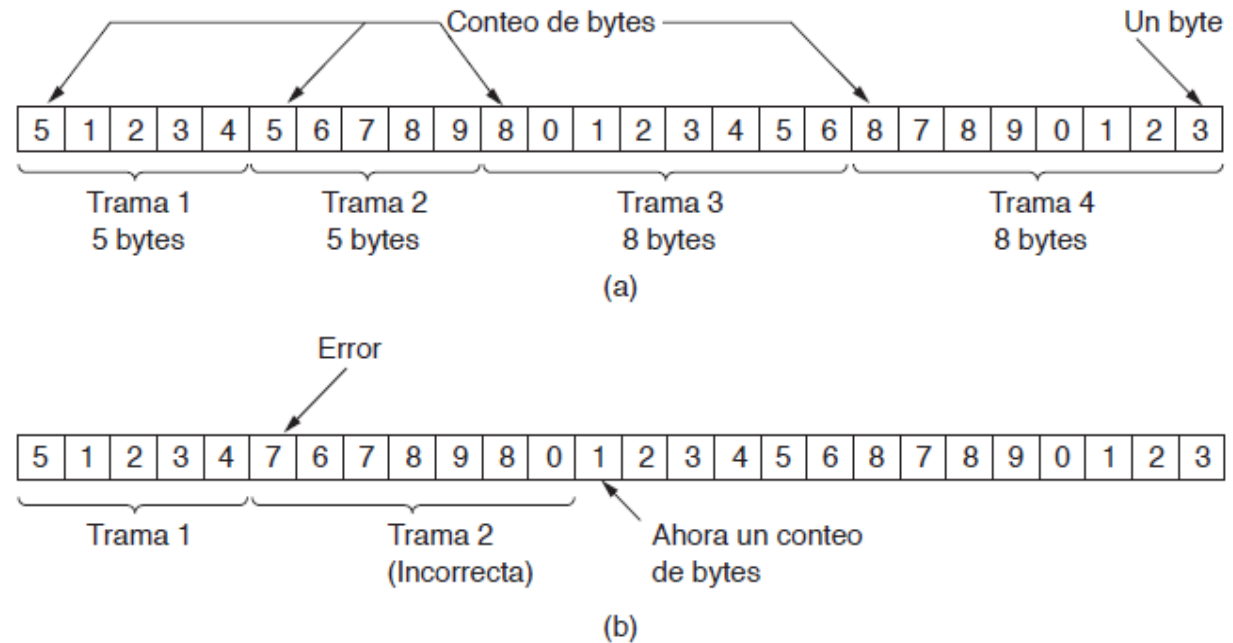


Figura 3-3. Un flujo de bytes. (a) Sin errores. (b) Con un error.

LA CAPA DE ENLACE DE DATOS

Entramado - Bytes bandera con relleno de bytes

- Hay un byte bandera (FLAG) como delimitador inicial y final. Dos bytes bandera consecutivos señalan el final de una trama y el inicio del siguiente.
- Si el receptor pierde la sincronización entonces busca el FLAG.
- Problema - puede suceder que el byte bandera aparezca en los datos.

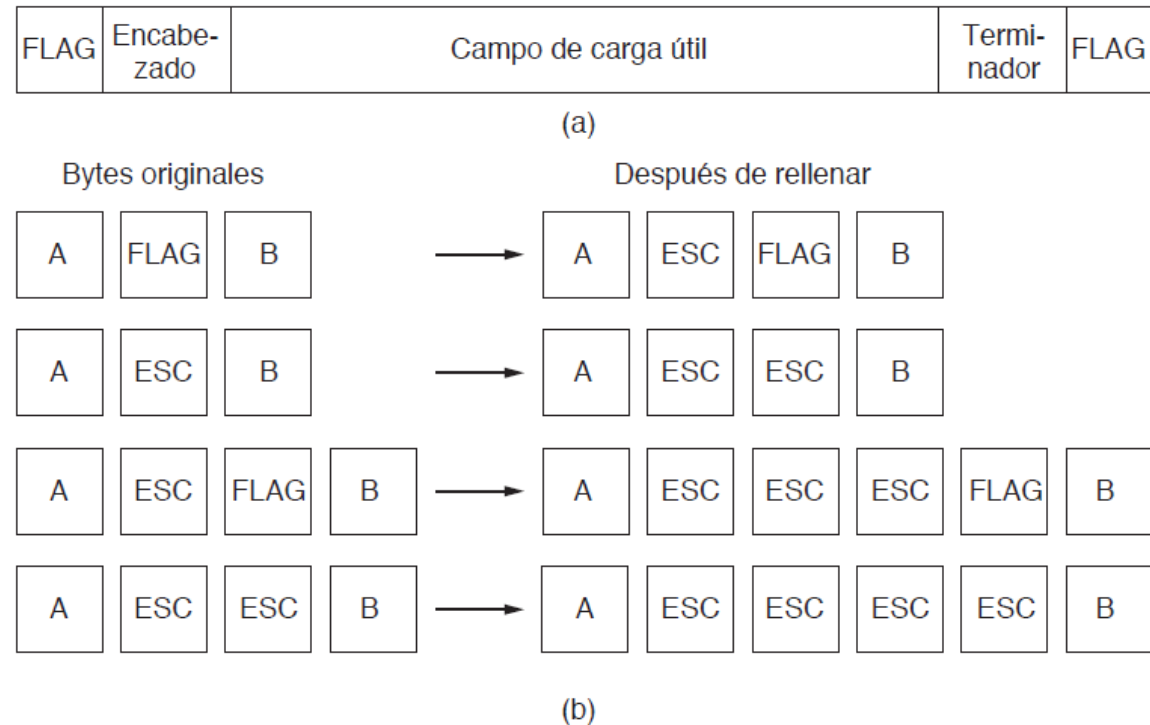


Figura 3-4. (a) Una trama delimitada por bytes bandera. (b) Cuatro ejemplos de secuencias de bytes antes y después del relleno de bytes.

LA CAPA DE ENLACE DE DATOS

Entramado - Bits bandera con relleno de bits.

- Para resolver el anterior problema se inserta un byte de escape especial (ESC) justo antes de cada byte bandera “accidental” en los datos.
- De esta forma se puede diferenciar un byte bandera del entramado de uno en los datos mediante la ausencia o presencia de un byte de escape.

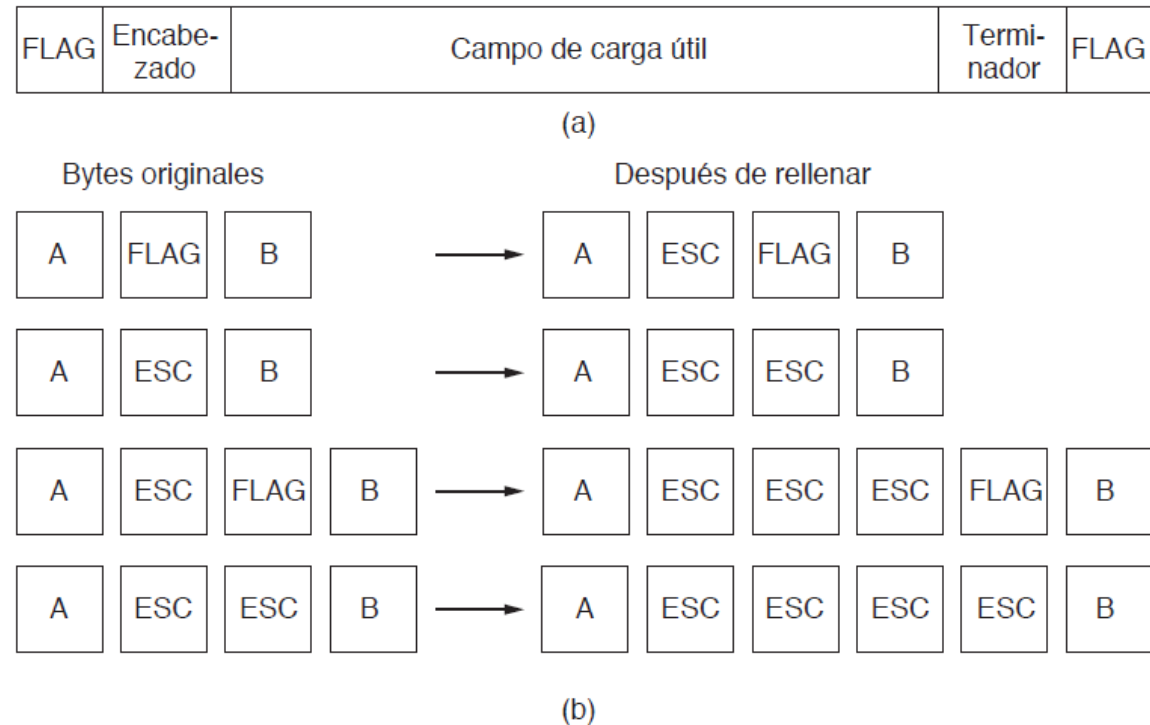


Figura 3-4. (a) Una trama delimitada por bytes bandera. (b) Cuatro ejemplos de secuencias de bytes antes y después del relleno de bytes.

LA CAPA DE ENLACE DE DATOS

- Muchos protocolos de enlace de datos usan una combinación de estos métodos por seguridad. Un patrón muy utilizado para **Ethernet y IEEE 802.11** es hacer que una trama inicie con un patrón bien definido, conocido como **preámbulo**.
- Ethernet e IEEE 802.11 son estándares de los dos niveles inferiores del modelo OSI.

LA CAPA DE ENLACE DE DATOS

Control de errores

- El proceso regular para asegurar la entrega confiable de datos es a través de la retroalimentación al emisor. Por lo general, el protocolo exige que el receptor devuelva tramas de control especiales con confirmaciones positivas o negativas.
- Se introducen temporizadores para el tiempo de espera de de los mensajes de confirmación.

Control de flujo

- Control de flujo basado en retroalimentación.
- Control de flujo basado en tasa.

LA CAPA DE ENLACE DE DATOS

Código de detección de errores

- Se utilizan más en los enlaces inalámbricos ya que son más ruidosos y propensos a errores a diferencia de los enlaces de fibra óptica.
- Existen tres códigos de detección de errores:
 - Paridad
 - Sumas de verificación
 - Pruebas de Redundancia Cíclica (CRC)

ETHERNET

- Es un estándar de redes LAN. Ethernet define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI.
 - **Ethernet clásica**, resuelve el problema de acceso múltiple. Operaba a tasas de transmisión de 3 a 10 Mbps.
 - **Ethernet conmutada**, los switches se utilizan para conectar distintas computadoras. Opera a 100, 1000 y 10000 Mbps, conocidas como Fast Ethernet, Gigabit Ethernet y 10 Gigabit Ethernet.

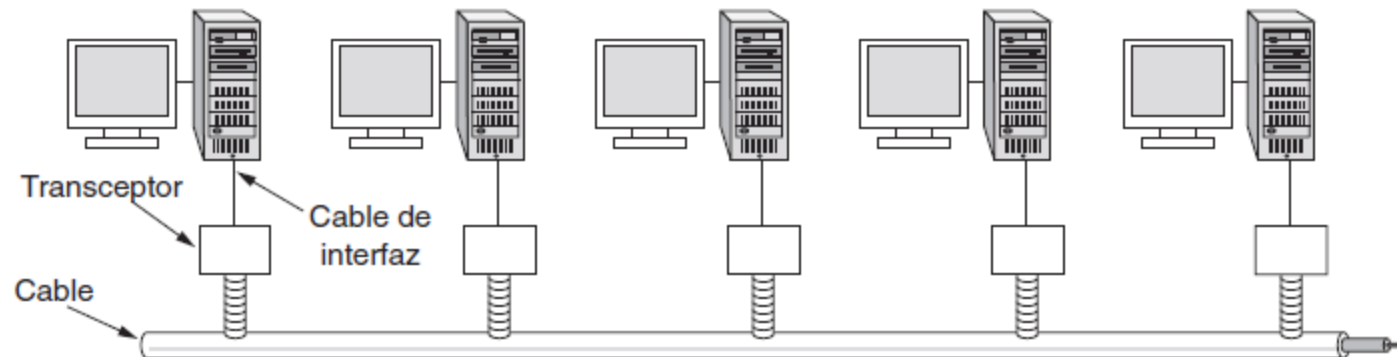


Figura 4-13. Arquitectura de Ethernet clásica.

ETHERNET

- El **preámbulo** es un patrón de bits 10101010 que le indica al receptor si es una trama Ethernet o IEEE 802.3. La trama Ethernet incluye un byte adicional que es equivalente al inicio de la trama (SOF) en IEEE 802.3 (es decir, los últimos dos bits se establecen en 11).
- El primer bit de la dirección de destino es un 0 para direcciones ordinarias (*multicasting*) y un 1 para direcciones de grupo (*broadcasting*).
- **Tipo**, especifica a qué proceso darle la trama. Por ejemplo, si es de IPv4.

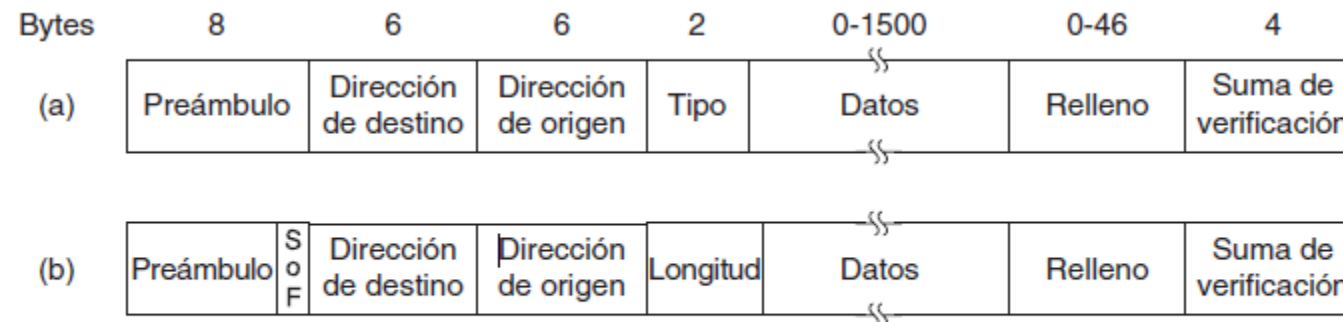


Figura 4-14. Formatos de trama. (a) Ethernet (DIX). (b) IEEE 802.3.

ETHERNET

- **Datos.** Con el fin de verificar tramas válidas de las inservibles, Ethernet especifica que las tramas deben tener como mínimo 64 bytes, de la dirección destino a la suma de verificación. Si la porción de una trama es menor que 46, el campo de **relleno** se utiliza para completarla.
- **Suma de verificación.** Determina si los bits de la trama se recibieron correctamente. Sólo realiza detección de errores y la trama se desecha si se detecta uno.

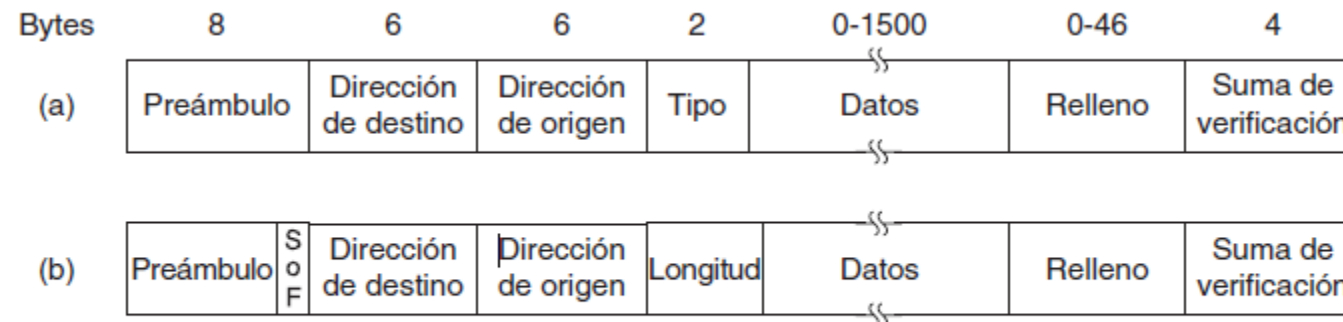


Figura 4-14. Formatos de trama. (a) Ethernet (DIX). (b) IEEE 802.3.

ETHERNET CONMUTADA

- Los problemas asociados con el hecho de encontrar interrupciones o conexiones flojas condujeron hacia un distinto tipo de patrón de cableado

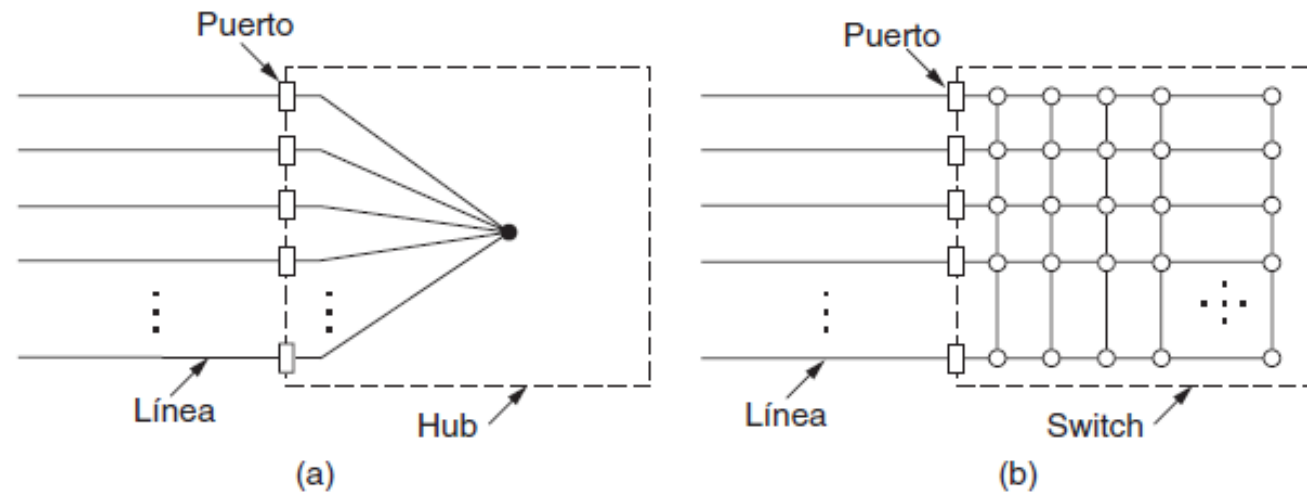


Figura 4-17. (a) Hub. (b) Switch.

ETHERNET CONMUTADA

- Pero un diseño con hubs (concentradores) no iba a incrementar la capacidad debido a que son lógicamente equivalentes al cable extenso individual de la Ethernet clásica.
- Existe una forma de aumentar la carga (una ethernet conmutada) donde el corazón de este sistema es un **conmutador (switch)** que contiene un plano posterior (backplane) de alta velocidad.



Small hub

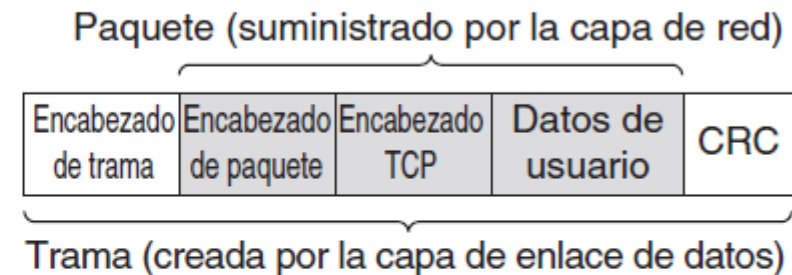


Workgroup switch

QUÉ DISPOSITIVO ESTÁ EN CADA CAPA.

Capa de aplicación	Puerta de enlace de aplicación.
Capa de transporte	Puerta de enlace de transporte.
Capa de red	Enrutador.
Capa de enlace de datos	Puente, switch.
Capa física	Repetidor, hub.

(a)

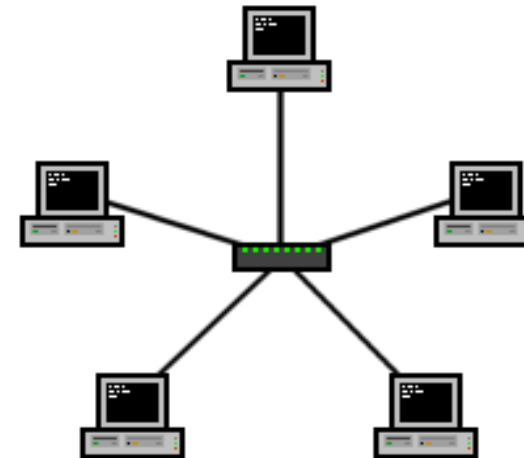
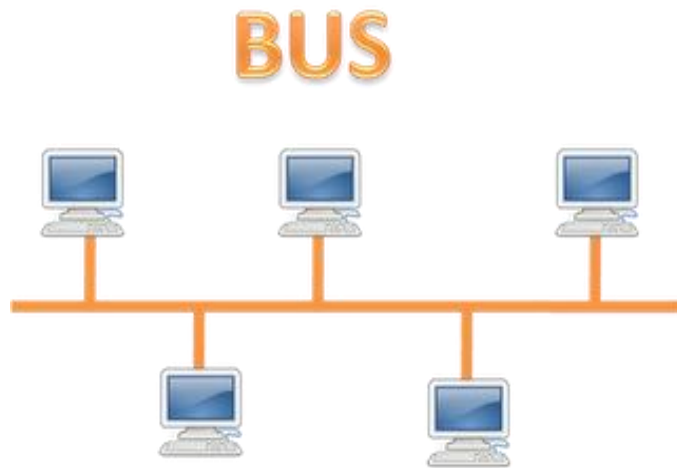


(b)

Figura 4-45. (a) Qué dispositivo está en cada capa. (b) Tramas, paquetes y encabezados.

RECORDEMOS...

- ¿Qué tipo de topologías hay? ¿cuales son sus ventajas y desventajas?



<https://www.conceptdraw.com/How-To-Guide/cisco-routers>

ETHERNET CONMUTADA

- El switch verifica la dirección de Ethernet para ver cuál es el puerto de destino de la trama.
- Utilizan algoritmos (por ejemplo el CSMA/CD) para evitar las colisiones entre las tramas, cada puerto es su propio dominio de colisión independiente.
- Recordemos... existen cables full-dúplex donde se envía información en ambas direcciones.

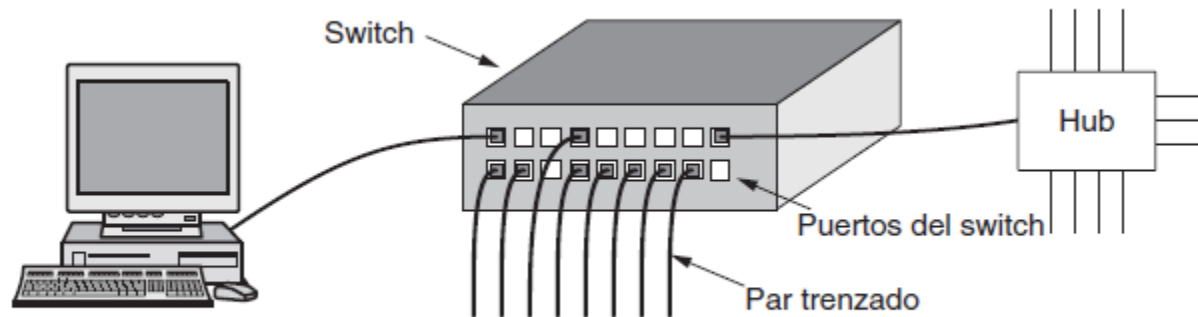
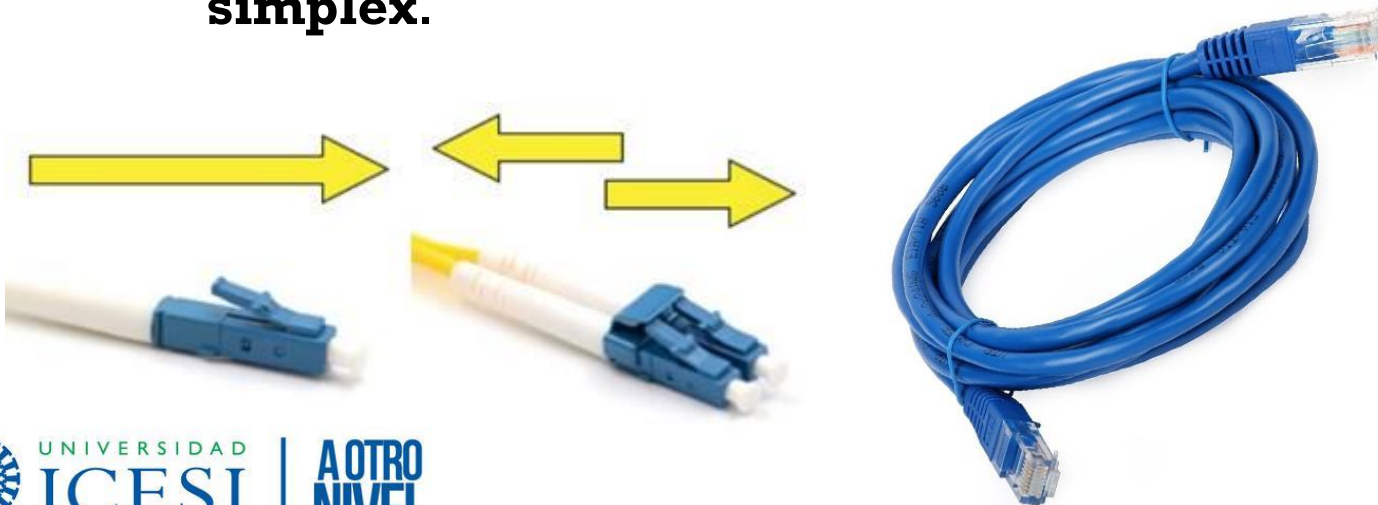


Figura 4-18. Un switch Ethernet.

TIPOS DE ENLACES — MEDIO ALAMBRADO

- Los enlaces que se pueden utilizar en ambas direcciones al mismo tiempo, como un camino de dos carriles, se llaman enlaces **full-dúplex**.
- Los enlaces que se pueden utilizar en cualquier dirección, pero sólo uno a la vez, como una vía de ferrocarril de un solo sentido, se llaman enlaces **half-dúplex**.
- Hay una tercera categoría que consiste en enlaces que permiten tráfico sólo en una dirección, como una calle de un solo sentido. A éstos se les conoce como enlaces **simplex**.



Los cables hasta la categoría 6 - UTP(**Par Trenzado sin Blindaje**, del inglés *Unshielded Twisted Pair*)

SEGURIDAD

- La mayoría de interfaces de LAN tienen un **modo promiscuo**, en el que **TODAS** las tramas se entregan a cada computadora y no sólo las que van dirigidas a ella.
- En un hub, cualquier computadora puede ver el tráfico de red transmitido entre los dispositivos.
- En un switch, el tráfico solo se reenvía sólo a los puertos a los que está destinado.

COMPETENCIAS, PRÓXIMA CLASE

- Explique la capa de enlace de datos

LECTURAS

Material utilizado	<p>1. Arboleda, L. (2012). Programación en Red con Java.</p> <p>2. Harold, E. (2004). Java network programming. " O'Reilly Media, Inc.".</p> <p>3. Tanenbaum, A. S. (2003). Redes de computadoras. Pearson educación.</p> <p>4. Reese, R. (2015). Learning Network Programming with Java</p>
Actividades DESPUÉS	<p>A2. Del libro 3 leer de la sección 4.8.1 hasta la 4.8.5</p> <p>A3. Para el proyecto, libro 4 – chapter 8</p>

REFERENCIAS

1. <https://arquitecturaecci.files.wordpress.com/2011/02/diagrama-bloques-de-cpu.jpg>
2. <https://www.gannett-cdn.com/-mm-/ee391ab73bef22fed8dca3c6af171e0a7102a51f/r=500x374/local-/media/2016/10/04/Rochester/wp-ROC-RocNext-10744-Security-is-like-an-onion1.jpg>
3. <https://infosegur.files.wordpress.com/2013/11/unidad-1.jpg>