

Capítulo 6: Capa Enlace de Datos y LANS

ELO322: Redes de Computadores
Agustín J. González

Este material está basado en:

- Material de apoyo al texto *Computer Networking: A Top Down Approach Featuring the Internet*. Jim Kurose, Keith Ross.

Capítulo 6: La Capa Enlace de Datos y LANs

Nuestros objetivos:

- ❑ Entender los principios detrás de los servicios de la capa enlace de datos:
 - Detección y corrección de errores
 - Compartición de canales broadcast: acceso múltiple
 - Direccionamiento de la capa enlace
 - Redes de área local: Ethernet
- ❑ Descripción e implementación de varias tecnologías de enlace

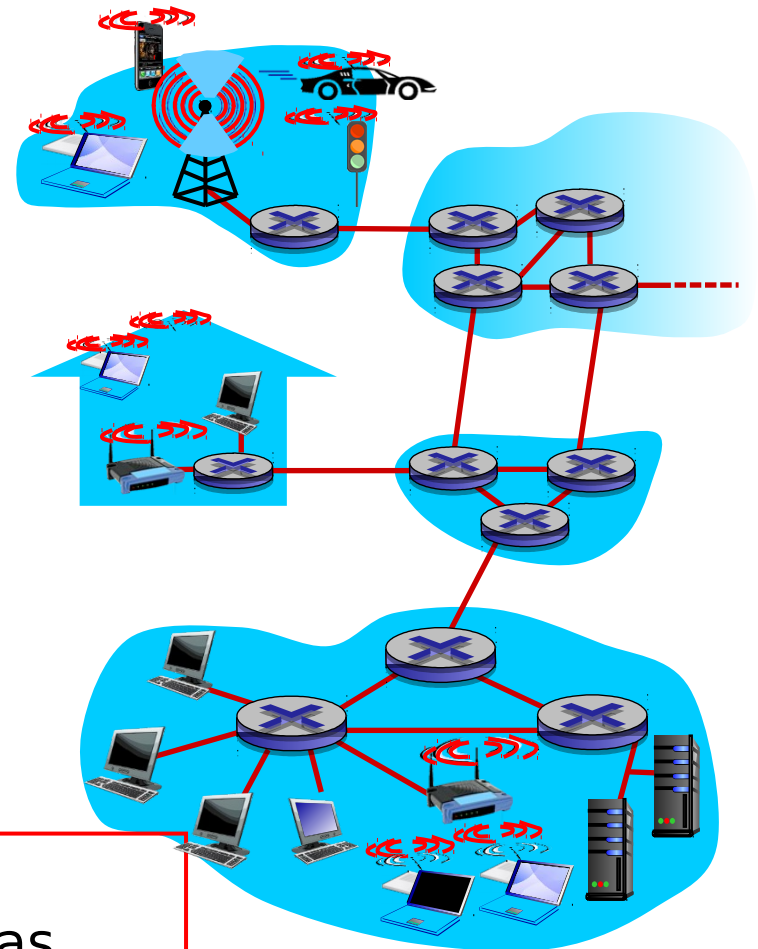
Capa Enlace de Datos, LANs

- ❑ 6.1 Introducción y servicios
- ❑ 6.2 Detección y corrección de errores
- ❑ 6.3 protocolos de acceso múltiple
- ❑ 6.4 LANs
- ❑ Secciones posteriores no son cubiertas en este curso → Redes de Computadores II

Capa Enlace: Introducción

Algo de terminología:

- Aquí llamaremos **nodo** a todo **a hosts y routers**
- Canales de comunicación que conectan nodos adyacentes a lo largo de un camino de comunicación son **enlaces (links)**. Ej.:
 - Enlaces cableados
 - Enlaces inalámbricos
 - LANs
- El paquete de capa 2 es la **trama (o frame)**, encapsula un datagrama



La capa de enlace de datos tiene la responsabilidad de transferir datagramas desde un nodo (computador o router) a otro nodo físicamente adyacente pasando por un enlace.

Capa Enlace: contexto

- ❑ Los datagramas son transferidos por diferentes protocolos de enlace y a través de enlaces físicos:
 - Por ejemplo: Ethernet en primer enlace entre host y switch y luego entre éste y el router, luego varios enlaces entre dos routers, entre un router y un access point y finalmente un enlace Wifi entre éste y un computador.



- ❑ Cada protocolo de enlace provee servicios diferentes
 - Por ejemplo: puede o no proveer transferencia confiable sobre el enlace

Servicios de Capa Enlace

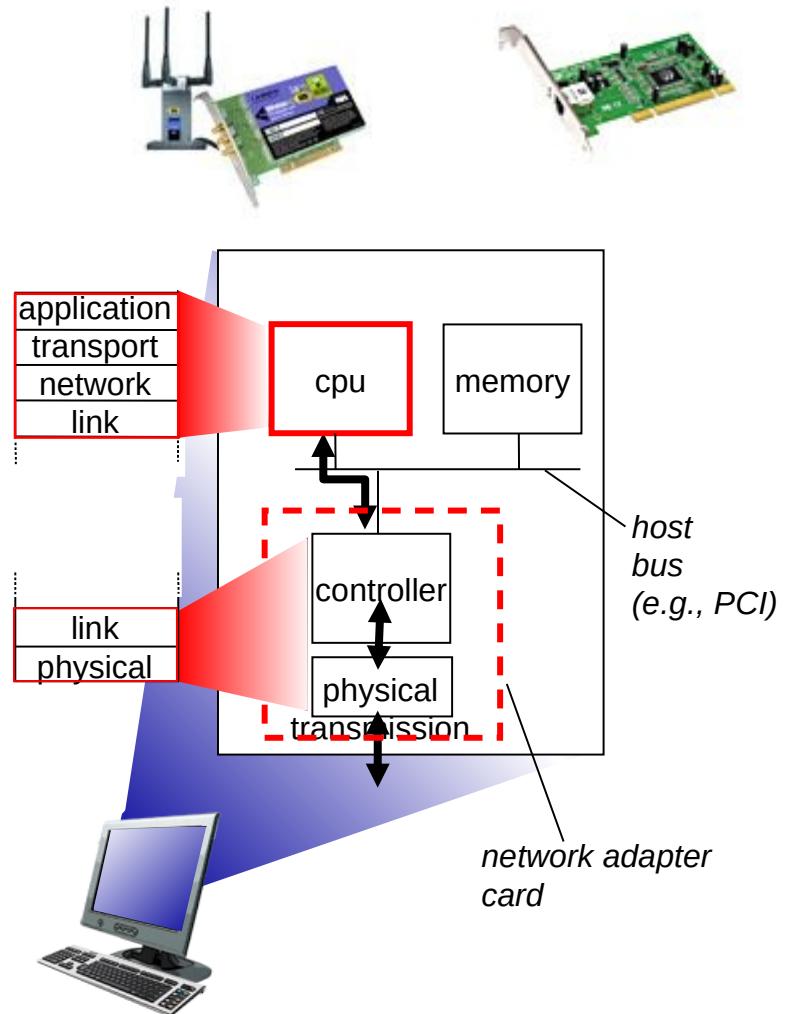
- ❑ **Construcción de tramas, acceso al enlace:**
 - Encapsula un datagrama en una trama, agregando encabezados (headers) y acoplados (trailers -al final de trama)
 - Acceso al medio vía un protocolo de Control de Acceso al Medio (MAC) si se trata de un acceso compartido
 - Dirección “MAC” usada en encabezados de tramas para identificar fuente y destino en capa de enlace.
 - Diferente de dirección IP!
- ❑ **Entrega confiable entre nodos adyacentes**
 - Ya vimos cómo hacer esto (al estudiar la capa de transporte)!
 - Raramente usado en enlaces de baja tasa de error de bits (como fibra, algunos pares de cobre trenzados)
 - Usado en enlaces propensos a errores. Ej: Enlaces inalámbricos.
 - Q: ¿por qué tener confiabilidad a nivel de enlace además de extremo a extremo?

Servicios de Capa Enlace (más)

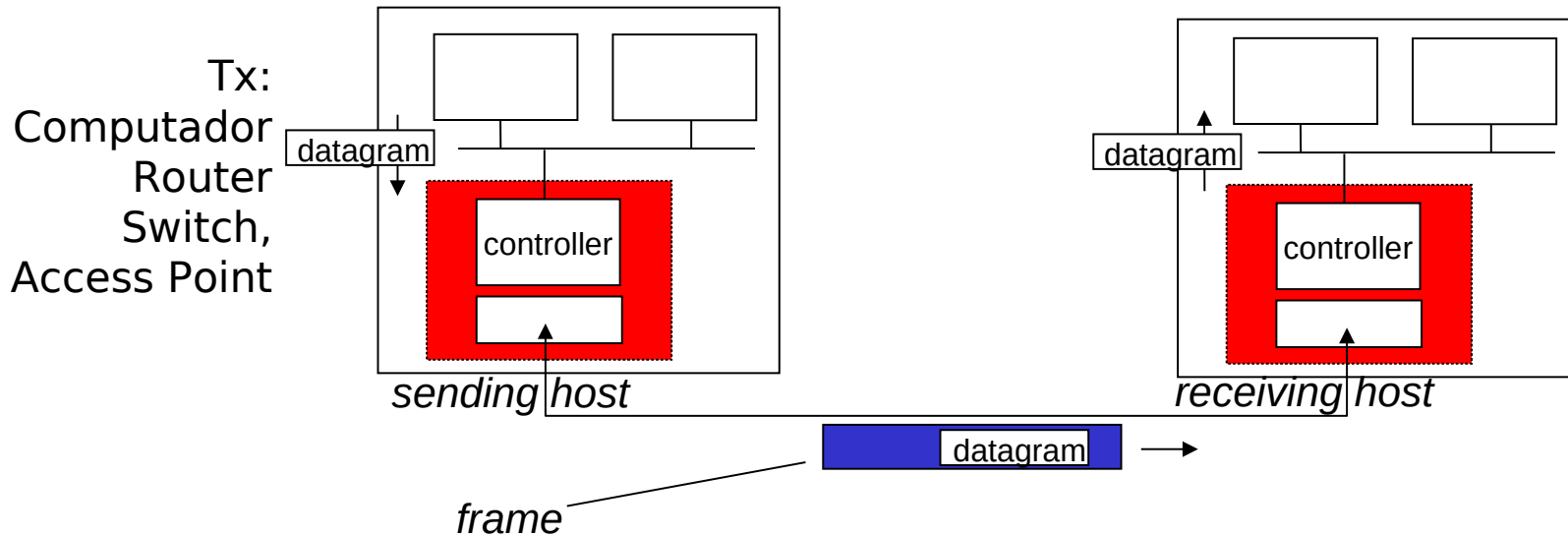
- ❑ **Control de flujo entre nodos adyacentes:** regular envío para no sobrecargar receptor
- ❑ **Detección de Errores:**
 - Introduce redundancia para detectar errores causados por atenuación de señal, ruido e interferencia.
 - Si el receptor detecta presencia de errores:
 - Pide retransmisión o descarta la trama.
 - Ésta está implementada por hardware.
- ❑ **Corrección de Errores (Forward error correction):**
 - Receptor identifica y corrige error(es) de bit(s) sin solicitar retransmisión (requiere el envío de más redundancia)
 - Su implementación es por hardware.

Adaptadores de comunicación

- Uno en cada equipo de la red
- La capa de enlace es implementada en un “adaptador” (NIC)
 - Ej. Tarjetas Ethernet ó 802.11 (WiFi)
 - Implementa la capa de enlace y la física
- Combina hardware, firmware y software.



Adaptadores de comunicación



❑ Lado transmisor:

- Encapsula el datagrama en una trama o frame
- Agrega bits de chequeo de errores, control de flujo, etc.

❑ Lado receptor

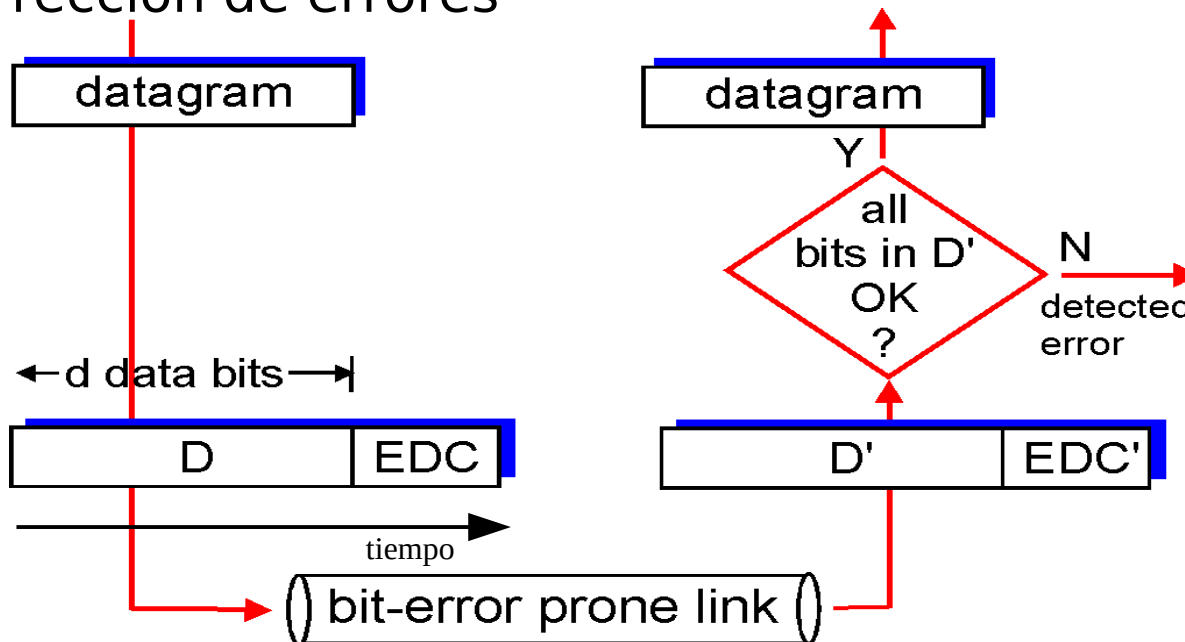
- Busca errores, procesa control de flujo, etc
- Extrae datagrama y lo pasa al nodo receptor

Capa Enlace de Datos, LANs

- ❑ 6.1 Introducción y servicios
- ❑ 6.2 Detección y corrección de errores
- ❑ 6.3 protocolos de acceso múltiple
- ❑ 6.4 LANs
- ❑ Secciones posteriores no son cubiertas en este curso → Redes de Computadores II

Detección de Errores

- ❑ EDC= Error Detection and Correction bits (redundancia)
- ❑ D = Dato protegido por chequeo de errores, podría incluir campos de encabezado
- ❑ La detección de errores no es 100% confiable!
 - el protocolo puede ignorar algunos errores, pero es raro.
 - Campos EDC grandes conducen a mejor detección y corrección de errores

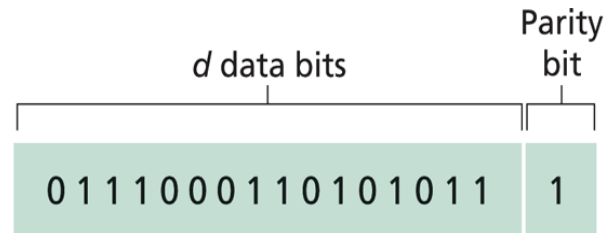


Chequeo de paridad

Bit de Paridad Simple:

Detecta errores simples

El bit de paridad es tal para completar un número par o impar de bits en uno. Decimos que usamos paridad par o impar respectivamente. Los ejemplos mostrados dan paridad par.

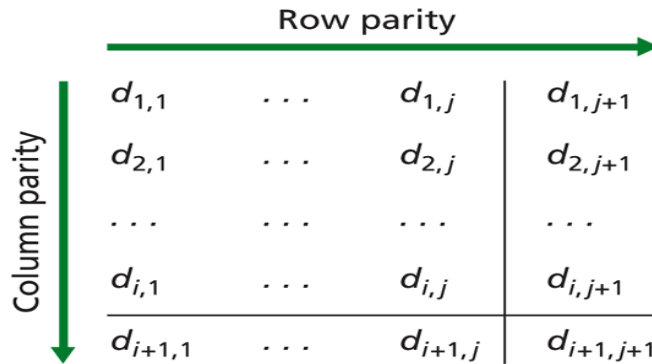


En hardware se usa Or-Ex

	0	1
0	0	1
1	1	0

Chequeo de paridad

Bit de paridad de dos dimensiones: Detecta y *corrige* errores simples



No errors

1	0	1	0	1		1
1	1	1	1	0		0
0	1	1	1	0		1
0	0	1	0	1		0

Correctable
single-bit error

1	0	1	0	1		1
1	0	1	1	0		0
0	1	1	1	0		1
0	0	1	0	1		0

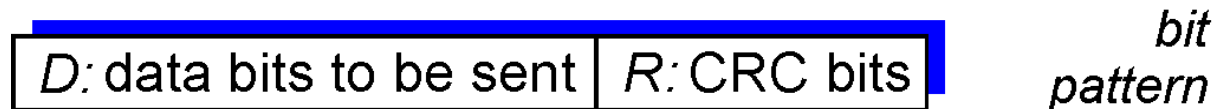
Parity error

Parity error

Sumas de chequeo: Chequeo de redundancia cíclica (CRC)

- ❑ Ampliamente usado en la práctica en capa enlace (e.g Ethernet, WiFi, ATM, HDCL)
- ❑ Bits de dato, **D**, son vistos como un número binario
- ❑ Se elige un patrón (generador) de $r+1$ bits, **G**.
- ❑ Objetivo: Obtener r bits de CRC, **R**, agregarlos luego de **D**, tal que:
 - $\langle D, R \rangle$ sea exactamente divisible por G (en “aritmética **módulo 2**”)
 - Rx conoce G , divide $\langle D, R \rangle$ por G . Si resto es no cero: hay error detectado!
 - Puede detectar secuencias de errores menores que $r+1$ bits

← d bits → ← r bits →



$D * 2^r \text{ XOR } R$ *mathematical formula*

CRC: Ejemplo

Queremos:

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

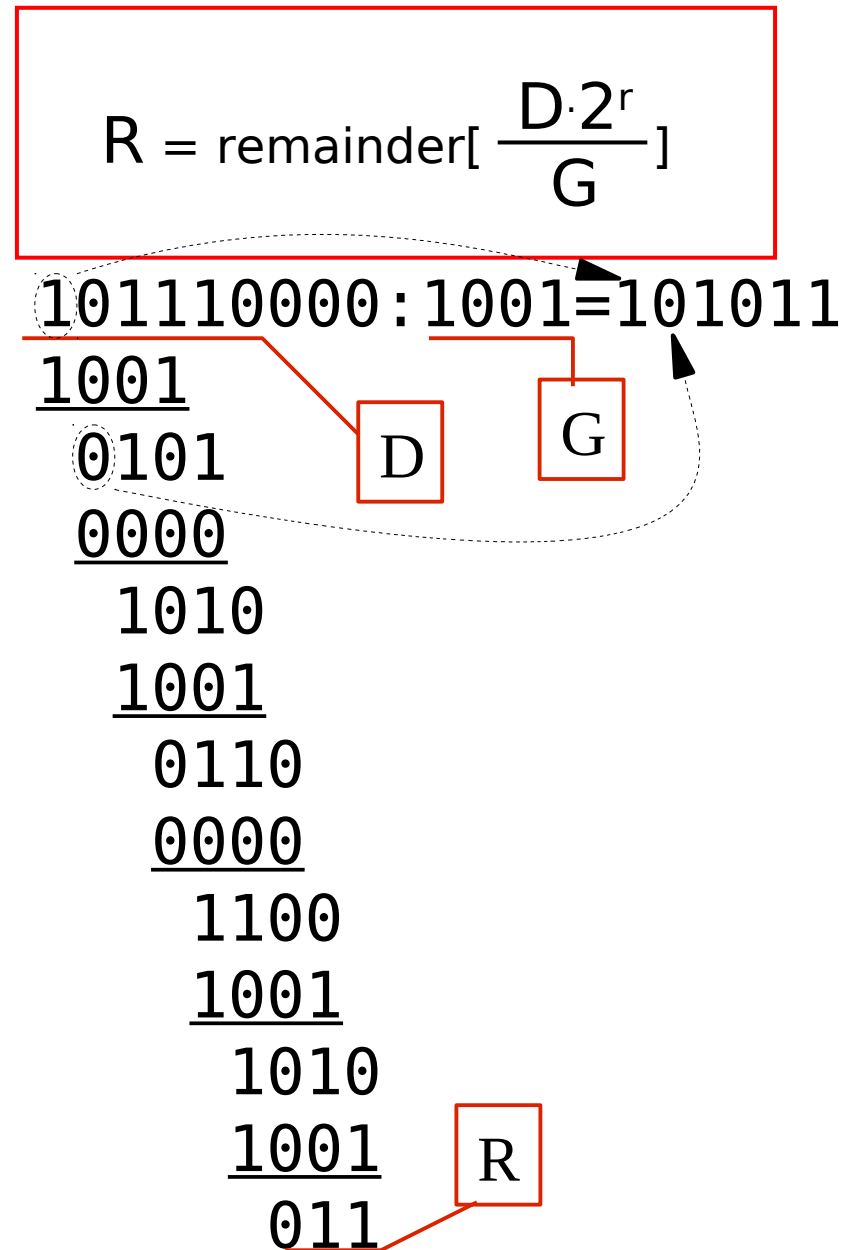
equivalentemente:

Si dividimos $D \cdot 2^r$ por G ,
obtendremos el resto R

Aquí todas las sumas y restas se
hacen dígito por dígito sin reservas.

Esto es:

$$A + B = A - B = A \text{ XOR } B$$



CRC: Ejemplo (cont)

1. Cálculo de la redundancia

$101110000 : 1001 = 101011$

<u>1001</u>		
0101	D	G
<u>0000</u>		
1010		
<u>1001</u>		
0110		
<u>0000</u>		
1100		
<u>1001</u>		
1010		
<u>1001</u>		
011	R	

2. Tx transmite data y redundancia

3. Rx verifica

$101110011 : 1001 = 101011$

<u>1001</u>	
0101	
<u>0000</u>	
1010	
<u>1001</u>	
0110	
<u>0000</u>	
1101	
<u>1001</u>	
1001	
<u>1001</u>	
000	=> Resto

Capa Enlace de Datos, LANs

- ❑ 6.1 Introducción y servicios
- ❑ 6.2 Detección y corrección de errores
- ❑ 6.3 protocolos de acceso múltiple
- ❑ 6.4 LANs
- ❑ Secciones posteriores no son cubiertas en este curso → Redes de Computadores II