

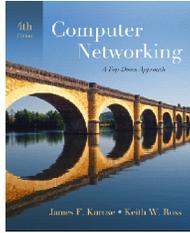
Introducción a las Redes de Computadores

Capítulo 5 Capa de Enlace y LANs

Nota acerca de las transparencias del curso:

Estas transparencias están basadas en el sitio web que acompaña el libro y han sido modificadas por los docentes del curso.

All material copyright 1996-2007
J.F. Kurose and K.W. Ross, All Rights Reserved



Computer Networking: A Top Down Approach
4th edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2007.

Capítulo 5: La Capa de Enlace de Datos

Objetivos:

- Entender los principios detrás de los servicios de la capa de enlace de datos:
 - detección de errores; corrección
 - compartir un canal de *broadcast*: acceso múltiple
 - direccionamiento de capa de enlace
 - transferencia de datos confiable, control de flujo

- Algunas tecnologías de Capa de Enlace

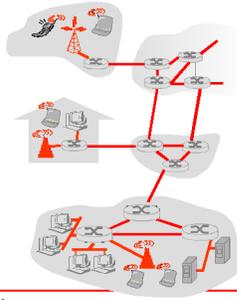
Capa de Enlace

- 5.1 Introducción y servicios
- 5.2 Detección y corrección de errores
- 5.3 Protocolos de acceso múltiple
- 5.4 Direccionamiento de Capa de Enlace
- 5.5 Ethernet
- 5.6 Switches de Capa de Enlace
- 5.7 PPP

Capa de Enlace: Introducción

Algo de terminología:

- hosts y routers son **nodes**
- los canales de comunicación que conectan nodos adyacentes a través de caminos de comunicación son **links**
 - enlaces cableados
 - enlaces inalámbricos
 - LANs
- la PDU de capa 2 es el **frame**, que encapsula un datagrama



la **capa de enlace de datos** tiene la responsabilidad de transferir datagramas desde un nodo a otro nodo adyacente, a través de un link

Capa de enlace: contexto

- los datagramas son transferidos por diferentes protocolos de enlace sobre diferentes enlaces:
 - p.e., Ethernet en el primer enlace, Frame Relay en los enlaces intermedios, 802.11 en el último enlace
 - cada protocolo de enlace brinda diferentes tipos de servicios
 - p.e., puede o no proveer **rdt** (*reliable data transfer*) sobre el enlace
- Analogía transporte**
- Viaje desde Montevideo a Mar del Plata
 - remise: Montevideo a Carrasco
 - avión: Carrasco a Aeroparque
 - ómnibus: Aeroparque a Mar del Plata
 - turista = **datagrama**
 - segmento de transporte = **enlace de comunicación**
 - modo de transporte = **protocolo de capa de enlace**
 - agencia de viaje = **algoritmo de enrutamiento**

Servicios de Capa de Enlace

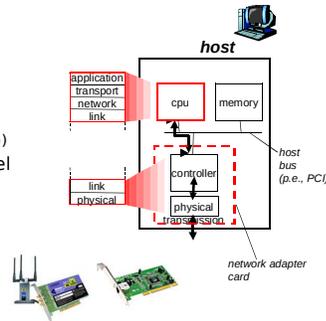
- **entramado (framing):**
 - encapsulado del datagrama en la trama, agregando encabezado (*header*) y cola (*trailer*)
- **acceso al enlace:**
 - acceso al canal si es un medio compartido (*Medium Access Control*)
 - direcciones "MAC" *addresses* utilizadas en los encabezados de las tramas para identificar el origen y el destino
 - distintas de las direcciones IP
- **entrega confiable:**
 - entre nodos adyacentes
 - ¡ya aprendimos cómo hacer esto (teo Capa de Transp.)!
 - rara vez utilizados en enlaces de pocos errores (fibra óptica, algunos pares trenzados)
 - enlaces inalámbricos: alta tasa de error
 - **P:** ¿Por qué confiabilidad a nivel de enlace y *end-end*?

Servicios de Capa de Enlace (más)

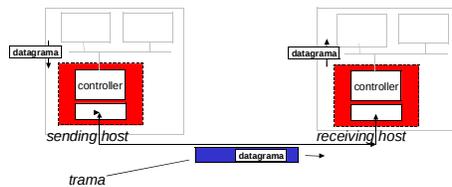
- **control de flujo:**
 - acuerdo entre los nodos emisor y receptor (aquí, adyacentes)
 - Recordar: *buffers* y capacidad de procesamiento
- **detección de errores:**
 - errores causados por atenuación de la señal, por ruido.
 - el receptor detecta presencia de errores:
 - señala al emisor para una retransmisión o descarta la trama
- **corrección de errores (FEC: Forward Error Correction):**
 - el receptor identifica *y corrige* el/los error/es en bit/s sin necesidad de retransmisión
- **half-duplex and full-duplex:**
 - con *half-duplex*, los nodos en los extremos del enlace pueden transmitir, pero no al mismo tiempo

¿Dónde está implementada la Capa de Enlace?

- En todos los *hosts*
- En el adaptador de red (**Network Interface Card: NIC**)
 - Tarjetas Ethernet, PCMCIA, 802.11
 - Implementa las capas de Enlace y Física (como mínimo)
- Incorporadas a los buses del sistema de los *hosts*
- combinación de *hardware, software, firmware*



Comunicación de adaptadores



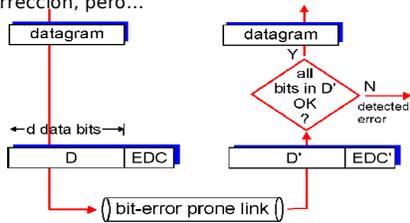
- **lado emisor:**
 - encapsula el datagrama en una trama
 - agrega bits de chequeo de error, rdt, control de flujo, etc.
- **lado receptor:**
 - busca por errores, rdt, control de flujo, etc
 - extrae el datagrama y lo pasa a las capas superiores

Capa de Enlace

- 5.1 Introducción y servicios
- 5.2 **Detección y corrección de errores**
- 5.3 Protocolos de acceso múltiple
- 5.4 Direccionamiento de Capa de Enlace
- 5.5 Ethernet
- 5.6 Switches de Capa de Enlace
- 5.7 PPP

Detección de errores

EDC= Error Detection and Correction bits (**redundancia**)
 D = Datos protegidos por chequeo de errores; puede incluir campos del encabezado
 ¡La detección de errores no es 100% confiable!
 • el protocolo puede perder algunos errores
 • un campo de EDC mayor proporciona mejor detección y corrección, pero...



Chequeo de paridad

Paridad de un bit:

Detecta errores en 1 bit
 ← d data bits → parity bit
 0111000110101011 0

La paridad puede ser par o impar, según la cantidad de 1's

Paridad par: ¿qué ocurre si hay una cantidad de errores par?

Paridad en dos dimensiones: Detecta y corrige errores en 1 bit

¿Detecta errores dobles?

		row parity
	$d_{1,1}$... $d_{1,j}$	$d_{1,j+1}$
	$d_{2,1}$... $d_{2,j}$	$d_{2,j+1}$

	$d_{i,1}$... $d_{i,j}$	$d_{i,j+1}$
column parity	$d_{i+1,1}$... $d_{i+1,j}$	$d_{i+1,j+1}$

101011	101011	
111100	101100	parity error
011101	011101	
001010	001010	
no errors	parity error	
	correctable	
	single bit error	

Internet checksum (suma de comprobación)

- **Objetivo:** detectar “errores” (bits cambiados) en el paquete transmitido (nota: generalmente utilizado en la capa de transporte)

- Recordar lo visto en Capa de Transporte

- En general es un método menos potente que el próximo que veremos

Internet checksum (suma de comprobación)

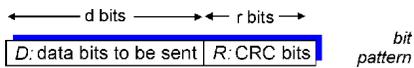
- Consiste en realizar la suma de los datos tomados de a k bits. El resultado es transmitido para la comprobación.
 - En Internet se considera la suma como enteros de 16 bits.
 - Generan poco *overhead*
 - Protección débil de los datos
- ¿Porque chequeos en capa 4 y capa2?
 - En capa 4 el chequeo puede tener poco éxito
 - En capa 2 se realiza chequeo por hardware con mejores resultados

Cyclic Redundancy Check

- códigos CRC o códigos polinómicos
- ampliamente utilizado en la práctica (Ethernet, 802.11 WiFi, ATM)
- ver a los bits de datos, D , como los coeficientes de un polinomio
 - por ejemplo: 110001 es x^5+x^4+1
- Toda la aritmética que se utiliza es módulo 2 sin *carry* en las operaciones (sumas y restas equivalentes a XOR)
- elegimos un patrón de **$r+1$ bits** (polinomio **generador**), G , de grado r , que conocen el transmisor y el receptor

Cyclic Redundancy Check

- objetivo: **determinar r CRC bits, R**, tal que
 - <D,R> (concatenado) es divisible exactamente por G
 - $D \times 2^r$ es desplazar hacia la izquierda r bits y agregando 0s
 - $D \times 2^r + R$ es concatenarlos
 - el receptor divide <D,R> entre G. Si el resto es distinto de cero: **error detectado!**



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

Ejemplo CRC

- El emisor busca R, tal que exista Q que cumpla:

$$D \cdot 2^r \text{ XOR } R = Q \cdot G$$

Que G divida a $D \cdot 2^r - R$ sin resto

$$D \cdot 2^r \text{ XOR } R = Q \cdot G$$

$$D \cdot 2^r \text{ XOR } R \text{ XOR } R = Q \cdot G \text{ XOR } R$$

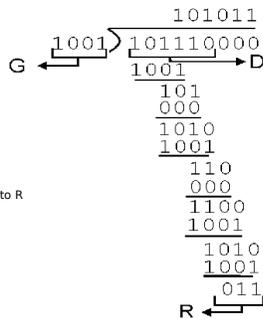
$$D \cdot 2^r = nG + R$$

D · 2^r: dividendo, G: divisor, Q: cociente,

R: resto

- si dividimos $D \cdot 2^r$ por G, buscamos el resto R

$$R = \text{resto} \left[\frac{D \cdot 2^r}{G} \right]$$



Estándares CRC

- Existen diferentes estándares de polinomios CRC
 - CRC-8
 - CRC-12
 - CRC-16
 - CRC-32
- Pueden detectar todas las ráfagas de errores menores a r+1 bits
- Si es divisible entre x+1, detecta todos los errores impares
- Para las mayores, el poder de detección es muy alto

Capa de Enlace

- 5.1 Introducción y servicios
- 5.2 Detección y corrección de errores
- 5.3 **Protocolos de acceso múltiple**
- 5.4 Direccionamiento de Capa de Enlace
- 5.5 Ethernet
- 5.6 Switches de Capa de Enlace
- 5.7 PPP

Protocolos y enlaces de acceso múltiple

Dos tipos de enlaces:

- punto a punto
 - PPP para acceso discado
 - Enlace punto a punto entre switch Ethernet y *host*
- **broadcast** (cable o medio compartido)
 - Ethernet "legacy"
 - HFC: *Hybrid Fiber Cable*
 - 802.11: LAN inalámbrica



cable compartido (p.e., cable Ethernet)



RF compartida (p.e., 802.11 WiFi)



RF compartida (satélite)



personas en una fiesta (aire compartido)

Protocolos de acceso múltiple

- Único canal *broadcast* compartido
 - Dos o más transmisiones simultáneas: interferencia
 - **Colisión**
 - si un nodo recibe dos o más señales al mismo tiempo
 - simultaneidad en el tiempo y en la frecuencia de dos o más tramas en el mismo medio físico
- Protocolo de Acceso Múltiple**
- Algoritmo distribuido que determina cómo los nodos comparten el canal, y determina cuándo el nodo puede transmitir
 - La comunicación acerca de compartir el canal debe utilizar el mismo canal
 - no canal *out-of-band* para coordinación

Protocolo de acceso múltiple ideal

Canal Broadcast con velocidad R bps

1. cuando un nodo quiere transmitir, lo hará a una velocidad R.
2. cuando M nodos quieren transmitir, cada uno enviará a una velocidad promedio de R/M
3. completamente descentralizado:
 - no hay un nodo especial para coordinar las transmisiones
 - no hay sincronización de relojes, *slots*
4. simple

Protocolos MAC: taxonomía

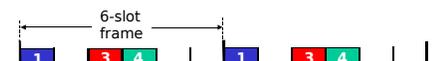
Tres grandes clases:

- **Particionado del canal**
 - Protocolos de arbitraje
 - divide el canal en pequeñas "piezas" (ranuras de tiempo, frecuencia, código)
 - asigna una pieza a un nodo para su uso exclusivo
 - estrategia estática
 - equitativo
- **Acceso Randómico**
 - el canal no se divide, permite colisiones
 - "recuperación" de colisiones
 - estrategia dinámica
- **"Toma de turnos"**
 - Los nodos toman turnos, pero los nodos con más tramas para enviar podrían tomar turnos más largos
 - estrategia dinámica
 - estrategias de reserva o centralizada

Protocolos MAC de particionado del canal: TDMA

TDMA: Time Division Multiple Access

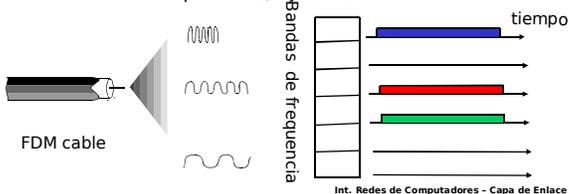
- acceso al canal rotativo
- cada estación tiene un *slot* de longitud fija (longitud = tiempo de transm. de la trama) en cada vuelta
- los *slots* sin usar quedan libres
- ejemplo: LAN con 6 estaciones, 1,3 y 4 tiene trama; los *slots* 2,5 y 6 quedan libres



Protocolos MAC de **particionado del canal: FDMA**

FDMA: *Frequency Division Multiple Access*

- el espectro del canal se divide en bandas de frecuencia
- a cada estación se le asigna una banda de frecuencia fija
- el tiempo de transmisión no utilizado en las bandas de frecuencia queda libre
- ejemplo: LAN con 6 estaciones, 1,3 y 4 tienen trama; las bandas de frecuencia 2,5 y 6 están libres



Protocolos de **acceso** **randómico**

- cuando un nodo tiene un paquete para enviar
 - transmite a la velocidad total del canal, R
 - no existe *a priori* coordinación entre nodos
- dos o más nodos transmitiendo → "colisión"
- **protocolos MAC de acceso randómico** especifican:
 - cómo detectar colisiones (directa o indirecta)
 - cómo recuperarse de las colisiones (p.e., a través de retransmisiones retrasadas)
- ejemplos de protocolos MAC de acceso randómico:
 - ALOHA ranurado, ALOHA
 - CSMA, CSMA/CD, CSMA/CA
 - También se les conoce como sistemas de contención o sistemas de contienda

ALOHA ranurado: (Roberts 1972)

Hipótesis:

- canal de R bps
- todas las tramas tienen el mismo tamaño (L bits)
- el tiempo está dividido en *slots* de igual tamaño (L/R segundos: el tiempo para transmitir 1 trama)
- los nodos comienzan a transmitir sólo al comienzo de cada *slot*
- los nodos están sincronizados (saben cuándo comienza cada *slot*)
- si 2 o más nodos transmiten en un *slot*, todos los nodos detectan la colisión antes que termine el *slot*

Operación:

- Cuando un nodo obtiene una trama nueva, la transmite en el *slot* siguiente
 - *si no hay colisión*: el nodo puede enviar una nueva trama en el siguiente *slot*
 - *si hay colisión*: el nodo retransmite la trama en cada *slot* siguiente con probabilidad **p** hasta el éxito. Esto lo hace cada nodo involucrado en la colisión

Eficiencia del Aloha puro

$$\begin{aligned} P(\text{éxito para un nodo } i) &= \\ &P(i \text{ transmite}) * \\ &P(\text{demás nodos no transmitan en } [t_i-1, t_i]) * \\ &P(\text{demás nodos no transmitan en } [t_i, t_i+1]) \\ &= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1} \\ &= p \cdot (1-p)^{2(N-1)} \end{aligned}$$

Calculando el p óptimo (p^*) y con $N \rightarrow$ infinito
Eficiencia máx. = $1/(2e) = 0,18$

Peor que el Aloha ranurado

CSMA (*Carrier Sense Multiple Access*)

CSMA: escuchar antes de transmitir

- Si el canal está libre: transmitir la trama entera
- Si el canal está ocupado: diferir la transmisión
 - volver a escuchar después de un tiempo
 - seguir escuchando hasta que quede libre y transmitir
 - seguir escuchando hasta que quede libre y transmitir con probabilidad p
- Analogía humana: no interrumpir a los otros!

Colisiones CSMA

las colisiones pueden

ocurrir:

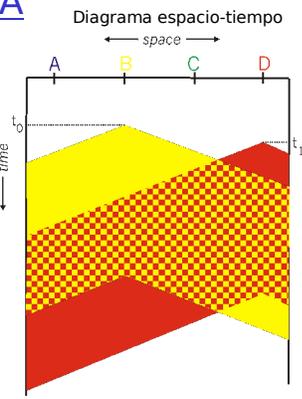
el retardo de propagación tiene como consecuencia que dos nodos puedan no oír la transmisión del otro

colisión:

El tiempo completo de transmisión de la trama se desperdicia

nota: factores relevantes

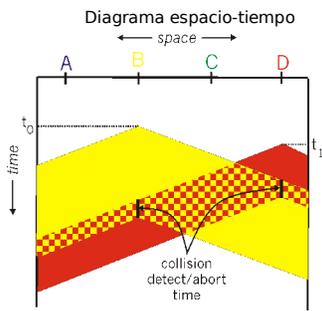
rol de la distancia y de la velocidad de propagación (ambos determinan el retardo de propagación) para inferir la probabilidad de colisión; además influyen la velocidad de transmisión y el largo del mensaje



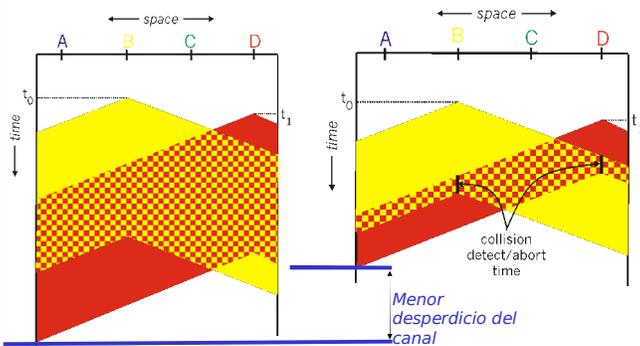
CSMA/CD (Collision Detection)

- CSMA/CD: si hay presencia de portadora, se difiere la transmisión, como en CSMA
 - las transmisiones que colisionan son abortadas, reduciendo el desperdicio de canal
 - colisión = desperdicio del canal
- detección de colisión:
 - relativamente fácil en LANs cableadas
 - difícil en LANs inalámbricas

CSMA/CD: Detección de Colisión



CSMA y CSMA/CD



Otro servicio de Capa de Enlace

□ En canales tipo broadcast

- el uso de los medios multiacceso involucra, entre otras cosas, gestionar las colisiones
- La gestión de colisiones involucra la detección de las mismas y la recuperación de la posibilidad de envío de tramas

Protocolos MAC “Toma de turnos”

protocolos MAC de particionado del canal:

- compartir el canal *justa* y *eficiente* a alta carga
- ineficiente a baja carga: retardo en el acceso al canal, ancho de banda $1/N$ asignado aún si hay un sólo nodo activo

protocolos MAC de acceso randómico

- eficiente a baja carga: un único nodo puede utilizar completamente el canal
- alta carga: *overhead* por colisión

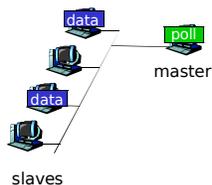
protocolos de “toma de turnos”

busca lo mejor de los dos mundos

Protocolos MAC “Tomando turnos”

Polling:

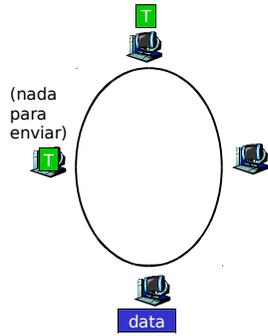
- el nodo *master* “invita” a los nodos *slaves* a transmitir en turnos
- típicamente utilizado con dispositivos *slaves* “tontos”
- sin colisiones
- determinístico
- involucra:
 - *overhead* por *polling*
 - latencia
 - único punto de falla (*master*)
- ejemplo
 - Bluetooth
 - IEEE 802.15
 - Un modo de operación de 802.11 (Wi Fi)



Protocolos MAC "Tomando turnos"

Paso de token:

- **token** (mensaje) de control pasado de un nodo a otro secuencialmente
- no existe un *master*
- involucra:
 - *overhead* por el *token*
 - latencia
 - único punto de falla (*token*)
- ejemplo:
 - Token Ring
 - IBM, IEEE 802.5



Resumen de protocolos MAC

- **particionado de canal**, en tiempo, frecuencia
 - división en el tiempo, división en la frecuencia
- **acceso randómico** (dinámico),
 - ALOHA, S-ALOHA, CSMA, CSMA/CD
 - Escucha de portadora: fácil en algunas tecnologías (cableadas), difícil en otras (inalámbricas)
 - CSMA/CD utilizado en Ethernet
 - CSMA/CA (*Collision Avoidance*) utilizado en 802.11
- **toma de turnos**
 - *polling* desde un sitio central, pasaje de *token*
 - Bluetooth, Token Ring
