

Capítulo 5: Capa Enlace de Datos - I

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 3rd* edition. Jim Kurose, Keith Ross Addison-Wesley, 2004.

Capítulo 5: La Capa Enlace de Datos

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
 - Transferencia de datos confiable y control de flujo: *ya lo hicimos!*
- Descripción e implementación de varias tecnologías de enlace

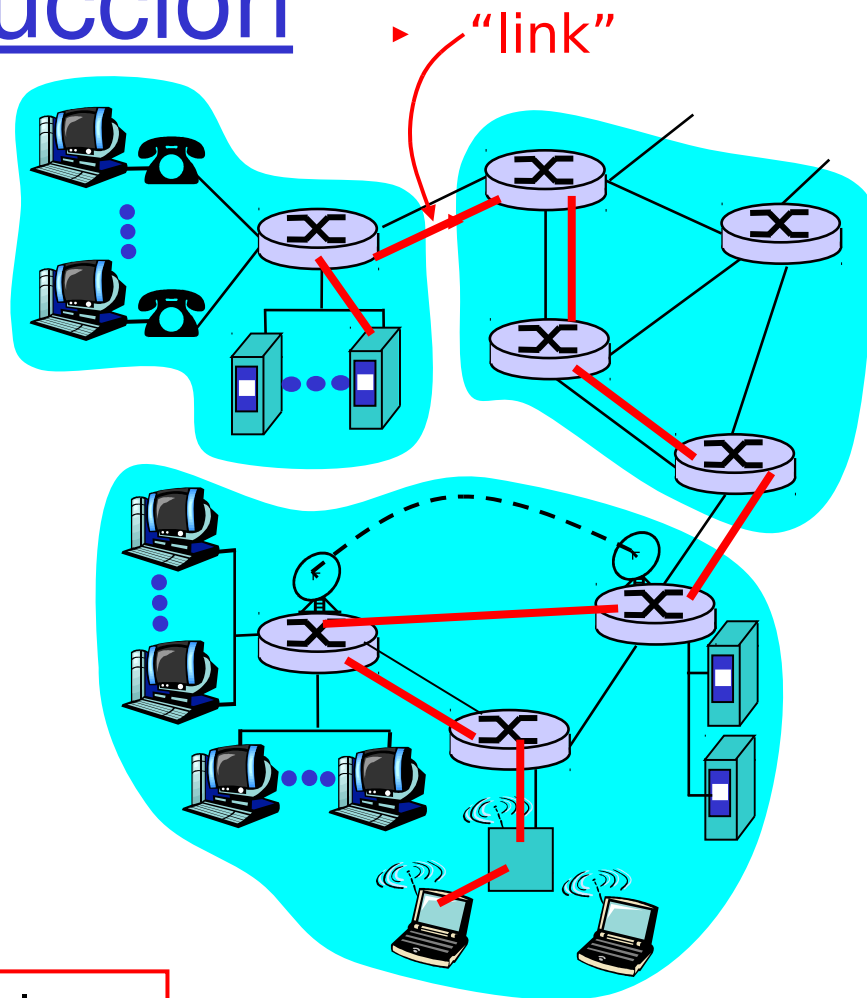
Capa Enlace de Datos

- 5.1 Introducción y servicios
- 5.2 Detección y corrección de errores
- 5.3 protocolos de acceso múltiple
- 5.4 Direccionamiento de capa enlace
- 5.5 Ethernet
- 5.6 Hubs y switches
- 5.7 PPP
- 5.8 Enlaces Virtuales: ATM y MPLS

Capa Enlace: Introducción

Algo de terminología:

- Hosts y routers son **nodos**
- Canales de comunicación que conectan nodos adyacentes a lo largo de un camino de comunicación son **enlaces**
 - Enlaces cableados
 - Enlaces inalámbricos
 - LANs
- El paquete de capa 2 es la **trama (o frame)**, encapsula a un datagrama



La capa de enlace de datos tiene la responsabilidad de transferir datagramas desde un nodo al nodo adyacente a través de un enlace

Capa Enlace: contexto

- Los datagramas son transferidos por diferentes protocolos de enlace en diferentes enlaces:
 - e.g., Ethernet en primer enlace, Frame Relay en enlaces intermedios, 802.11 en último enlace.
- Cada protocolo de enlace provee servicios diferentes
 - e.g., puede o no proveer transferencia confiable sobre el enlace

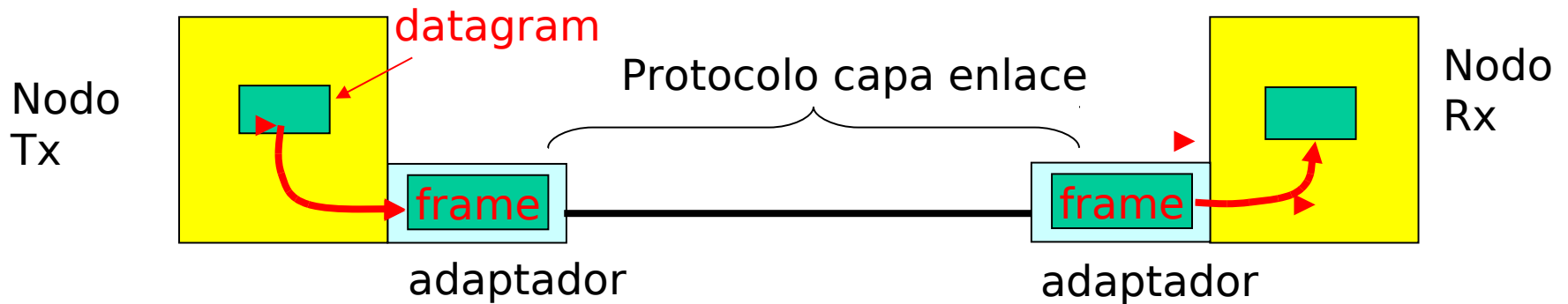
Servicios de Capa Enlace

- **Construcción de tramas, acceso al enlace:**
 - Encapsula un datagrama en una trama, agregando encabezados y acoplados (header & trailer)
 - Acceso al medio si se trata de un acceso compartido
 - Dirección “MAC” usada en encabezados de tramas para identificar fuente y destino
 - Diferente de dirección IP!
- **Entrega confiable entre nodos adyacentes**
 - Ya vimos cómo hacer esto (capa transporte)!
 - Raramente usado en enlaces de bajo error de bits (como fibra, algunos pares de cobre trenzados)
 - Enlaces inalámbricos: alta tasa de errores
 - Q: ¿por qué tener confiabilidad a nivel de enlace y extremo a extremo?

Servicios de Capa Enlace (más)

- **Control de Flujo:**
 - Paso entre nodos transmisor y receptor adyacentes
- **Detección de Errores:**
 - Errores causados por atenuación de señal y ruido.
 - Receptor detecta presencia de errores:
 - Pide al transmisor retransmisión o descartar la trama
- **Corrección de Errores (Forward error correction):**
 - Receptor identifica y corrige error(es) de bit(s) sin solicitar retransmisión
- **Half-duplex and full-duplex**
 - Con half duplex, los nodos de ambos extremos pueden transmitir pero no al mismo tiempo

Adaptadores de comunicación



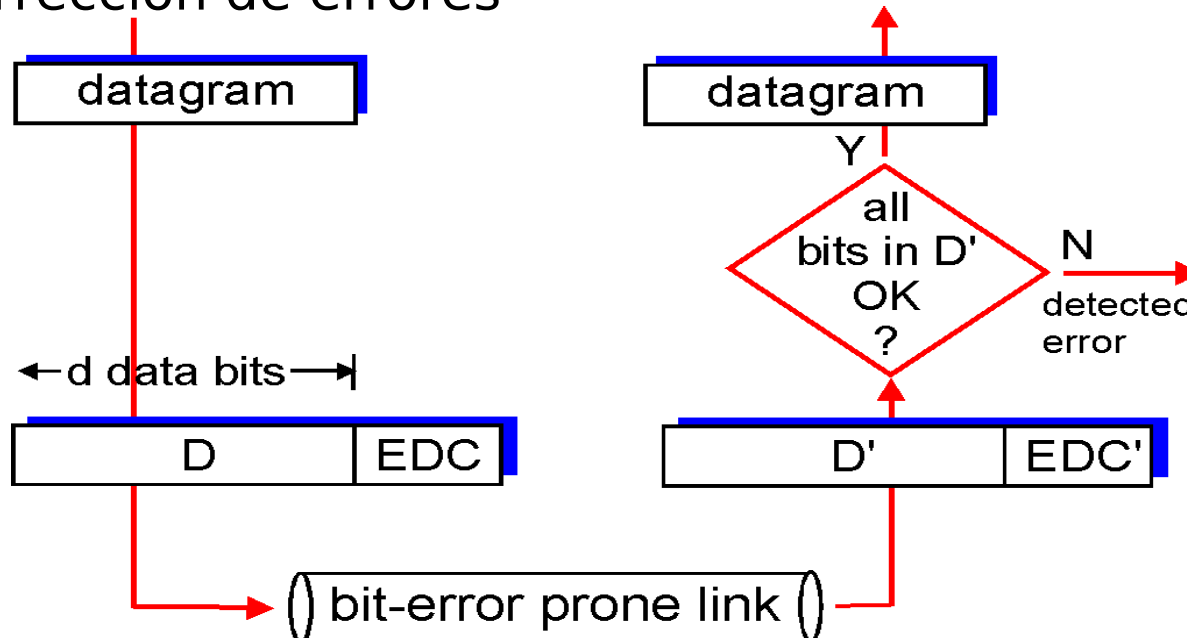
- La capa de enlace es implementada en un “adaptador” (NIC)
 - Tarjetas Ethernet, PCMCIA, ó 802.11
- Lado transmisor:
 - Encapsula el datagrama en una trama o frame
 - Agrega bits de chequeo de errores, control de flujo, etc.
- Lado receptor
 - Busca errores, control de flujo, etc
 - Extrae datagrama y lo pasa al nodo receptor
- El adaptador es semi-autónomo
- Capa enlace & capa física

Capa Enlace de Datos

- 5.1 Introducción y servicios
- 5.2 Detección y corrección de errores (saltado)
- 5.3 protocolos de acceso múltiple
- 5.4 Direccionamiento de capa enlace
- 5.5 Ethernet
- 5.6 Hubs y switches
- 5.7 PPP
- 5.8 Enlaces Virtuales: ATM y MPLS

Detección de Errores

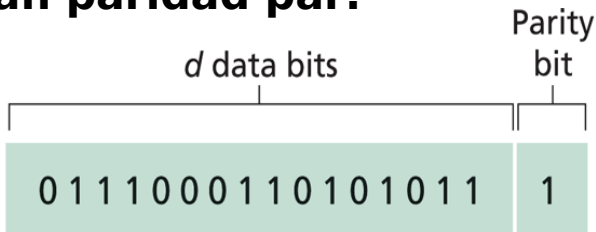
- EDC= Error Detection and Correction bits (redundancia)
- D = Datos protegidos por chequeo de errores podría incluir campos de encabezado
- La detección de errores no es 100% confiable!
 - el protocolo puede saltar algunos errores, pero es raro
 - Campos EDC grandes conducen a mejor detección y corrección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



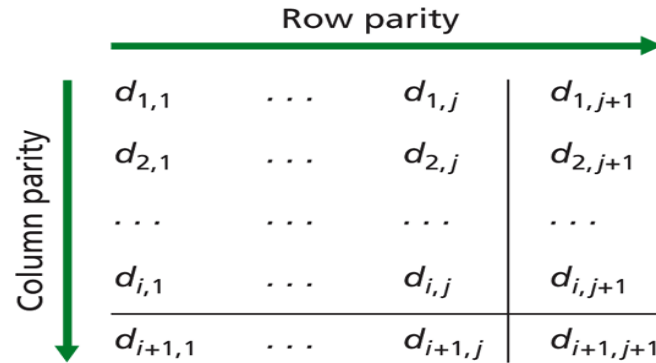
Chequeo de paridad

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.



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

Cheksum de Internet

Objetivo: detectar “errores” (e.g., bit invertidos) en segmentos transmitidos (nota: típicamente usado en capa transporte)

Transmisor:

- Trata el contenido de los segmentos como una secuencia de enteros de 16 bits
- checksum: suma del contenido del segmento (complemento 1 de la suma)
- Tx pone el valor del checksum en el campo correspondiente de UDP o TCP

Receptor:

- Calcula el checksum del segmento recibido
- Chequea si este checksum es igual al campo recibido:
 - NO - error detectado
 - SI - no hay error. *Pero podría haberlo?* Más luego

Sumas de chequeo: Chequeo de redundancia cíclica (CRC)

- Ve bits de datos, **D**, como números binarios
- Se elige un patrón (generador) de $r+1$ bits, **G**
- Objetivo: Elegir r bits de CRC, **R**, tal que:
 - $\langle D, R \rangle$ sea exactamente divisible por G (en aritmética **módulo 2**)
 - R_x 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
- Ampliamente usado en la práctica en capa enlace (e.g ATM, HDCL)

← 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:

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

equivalentemente:

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

Todas las sumas y restas se hacen
dígito por dígito sin carry en
aritmética **Módulo 2**:

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

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

$$\begin{array}{r}
 101110000 : 1001 = 101011 \\
 \underline{1001} \\
 101 \\
 \underline{000} \\
 1010 \\
 \underline{1001} \\
 110 \\
 \underline{000} \\
 1100 \\
 \underline{1001} \\
 1010 \\
 \underline{1001} \\
 011
 \end{array}$$

Diagram illustrating the long division of $D \cdot 2^r$ by G to find the remainder R . The dividend is 101110000 and the divisor is 1001 . The quotient is 101011 . The remainder R is 011 . The diagram uses boxes to label the dividend (D), divisor (G), and remainder (R).

CRC: Ejemplo (cont)

Queremos:

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

101110000 : 1001 = 101011

1001

101

000

1010

1001

110

000

1100

1001

1010

1001

011

D

G

R

Verificación:

101110011 : 1001 = 101011

1001

0101

0000

1010

1001

0110

0000

1101

1001

1001

1001

000 = Resto

Capa Enlace de Datos

- 5.1 Introducción y servicios
- 5.2 Detección y corrección de errores
- 5.3 protocolos de acceso múltiple
- 5.4 Direccionamiento de capa enlace
- 5.5 Ethernet
- 5.6 Hubs y switches
- 5.7 PPP
- 5.8 Enlaces Virtuales: ATM y MPLS

Protocolos de acceso múltiple

- Usan un canal simple de difusión compartida
- Puede haber dos o más transmisiones simultáneas por nodos: => Interferencia
 - **colisión** si un nodo recibe dos o más señales al mismo tiempo

Protocolos de acceso múltiple

- Algoritmo distribuido que determinan cómo los nodos comparten el canal, i.e., determina cuándo un nodo puede transmitir
- Son los mensajes para ponerse de acuerdo sobre cómo compartir el mismo canal!
 - no hay canal “fuera de banda” para coordinación

Protocolo de Acceso Múltiple Ideal

Supongamos un canal para broadcast de tasa R bps. Lo IDEAL sería:

1. Cuando un nodo quiere transmitir, éste puede enviar a tasa R .
2. Cuando M nodos quieren transmitir, cada uno puede enviar en promedio a una tasa R/M
3. Completamente descentralizado:
 - ▣ No hay nodo especial para coordinar transmisiones
 - ▣ No hay sincronización de reloj o ranuras
4. Es simple diseñarlo, este ideal no existe, pero define el máximo teórico.

Taxonomía de protocolos MAC (Media Access Control)

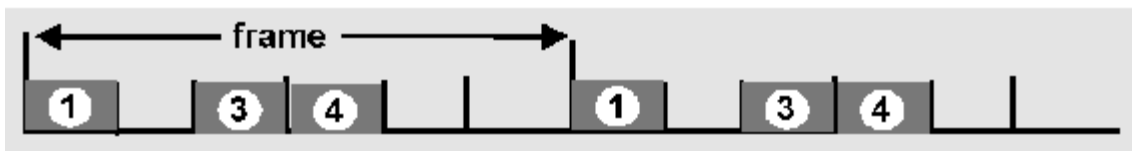
Tres clases amplias:

- **Canal Subdividido (“particionado”)**
 - Divide el canal en pequeños “pedazos” (ranuras de tiempo, frecuencia, código)
 - Asigna pedazos a un nodo para su uso exclusivo
- **Acceso Aleatorio**
 - Canal no es dividido, permite colisiones
 - Hay que “recuperarse” de las colisiones
- **“Tomando turnos”**
 - Los nodos toman turnos, pero nodos con más por enviar pueden tomar turnos más largos

Protocolo MAC en canal subdividido: TDMA

TDMA: time division multiple access

- Acceso a canales es en “rondas”
- Cada estación obtiene una ranura de largo fijo (largo= tiempo transmisión del paquete) en cada ronda
- Ranuras no usadas no se aprovechan
- ejemplo: LAN con 6 estaciones, 1,3,4 tienen paquetes, ranuras 2,5,6 no usadas

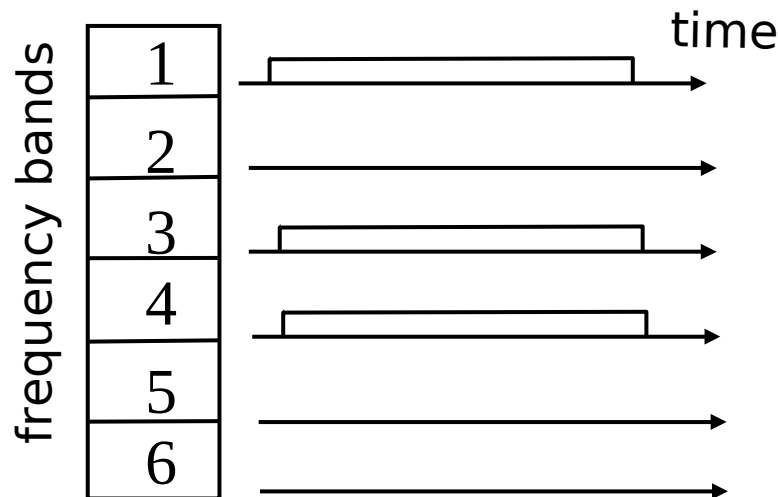


Eje. Reserva de esta sala para clases

Protocolos MAC en canal Subdividido: FDMA

FDMA: frequency division multiple access

- Espectro del canal es dividido en bandas de frecuencia
- Cada estación obtiene una banda de frecuencia fija
- Tiempo de transmisión no usado no es aprovechado
- Ejemplo: LAN de 6 estaciones, 1,3,4 tiene paquetes, bandas de frecuencias 2,5,6 no se aprovechan



Ej: Canales de televisión

Protocolos de Acceso Aleatorio

- Cuando un nodo tiene paquetes que enviar
 - Transmite a la tasa máxima del canal R.
 - No hay coordinación entre nodos
- Si dos o más nodos transmiten se produce “colisión”
- **Protocolos de acceso aleatorio** especifican:
 - Cómo detectar colisiones
 - Cómo recuperarse de una colisión (e.g., vía retransmisiones retardadas)
- Ejemplos de protocolos MAC de acceso aleatorio:
 - ALOHA ranurado
 - ALOHA
 - CSMA, CSMA/CD, CSMA/CA (CSMA: Carrier Sense Multiple Access)

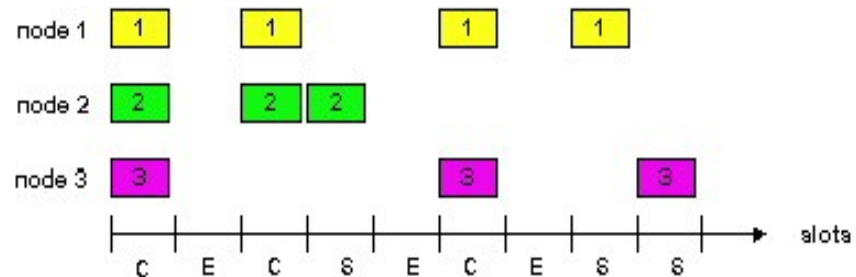
ALOHA ranurado

Suposiciones

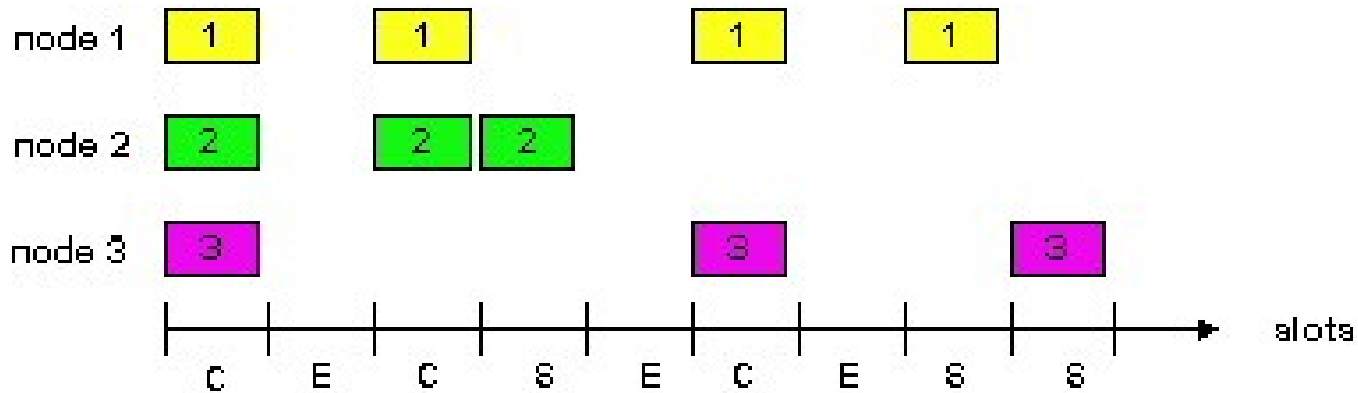
- Todos las tramas tienen igual tamaño
- Tiempo es dividido en ranuras de igual tamaño = tiempo para enviar una trama
- Nodos comienzan a transmitir sólo al inicio de cada ranura
- **Nodos están sincronizados**
- Si 2 ó más nodos transmiten en una ranura, todos los nodos detectan la colisión

Operación

- Cuando un nodo obtiene una trama nueva a enviar, éste transmite en próxima ranura
- Si no hay colisión, el nodo puede enviar una nueva trama en próxima ranura
- Si hay colisión, el nodo retransmite la trama en cada ranura siguiente con probabilidad p hasta transmisión exitosa



ALOHA ranurado



Ventajas

- Un único nodo activo puede transmitir continuamente a tasa máxima del canal
- Altamente descentralizado: pero cada nodo requiere sincronización en ranuras
- Simple

Desventajas

- Colisiones, las ranuras se desperdician
- Ranuras no ocupadas
- Nodos podrían detectar la colisión en menor tiempo que el de transmitir un paquete
- Sincronización de relojes

Eficiencia de Aloha ranurado (Slotted Aloha)

Eficiencia fracción a largo plazo de uso exitoso de ranuras cuando hay muchos nodos y cada uno tiene muchas tramas para enviar

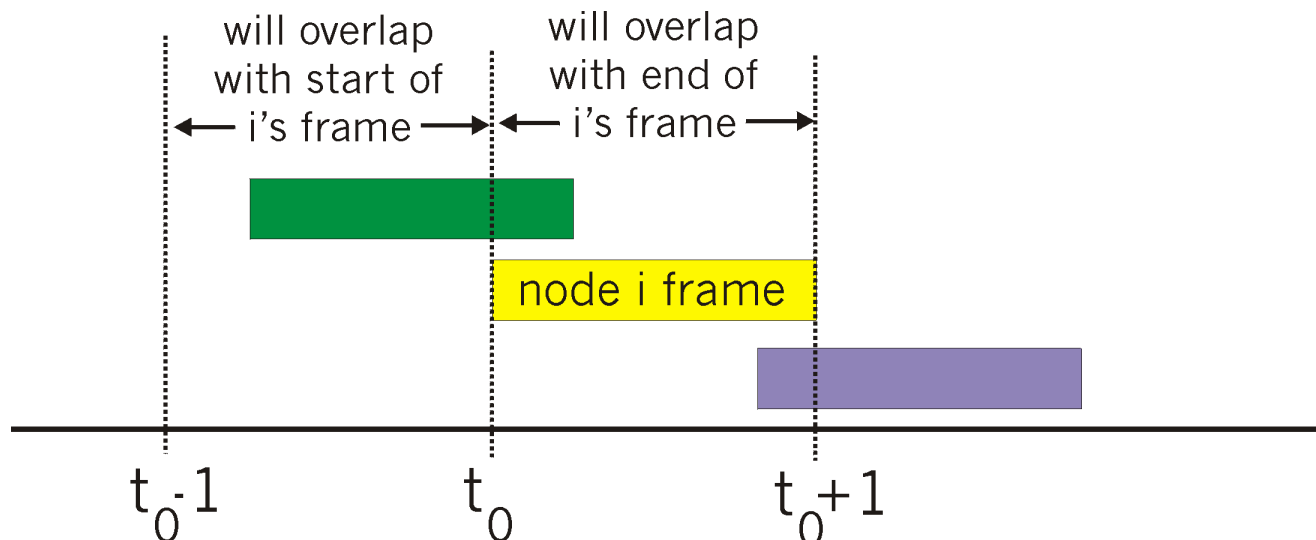
- ▣ Supongamos N nodos con muchas tramas a enviar, cada una transmite con probabilidad p
- ▣ Simplificación para el cálculo
- ▣ Prob que el nodo 1 tenga éxito en un slot = $p(1-p)^{N-1}$
- ▣ Prob que cualquier nodo tenga éxito = $Np(1-p)^{N-1}$

- ▣ Con N nodos activos la Eficiencia es: $E(p) = Np(1-p)^{N-1}$
- ▣ Para encontrar la máxima Eficiencia se debe encontrar p^* que maximiza $E(p)$. Ejercicio en guía.
- ▣ Para muchos nodos, tomar límite de $Np^*(1-p^*)^{N-1}$ cuando N va a infinito, da $1/e = .37$

Mejor caso: canal usado para transmisiones útiles 37% del tiempo!

ALOHA Puro (no ranurado)

- Aloha no ranurado: más simple, no hay sincronización
- Cuando una trama debe ser enviada
 - transmitir inmediatamente
- Probabilidad de colisión aumenta:
 - Trama enviada en t_0 colisiona con otras tramas enviadas en $[t_0-1, t_0+1]$



Eficiencia de Aloha puro

P(éxito transmisión de un frame en nodo) =

P(nodo transmita) *

P(ningún otro nodo transmita en $[t_0-1, t_0]$) *

P(ningún otro nodo transmita en $[t_0, t_0+1]$)

$$= p \cdot (1-p)^{N-1} \cdot (1-p)^{N-1}$$

$$= p \cdot (1-p)^{2(N-1)}$$

... elegir p óptimo y dejar que $N \rightarrow$ infinito ...

$$= 1/(2e) = .18$$

Incluso peor!

CSMA (Carrier Sense Multiple Access)

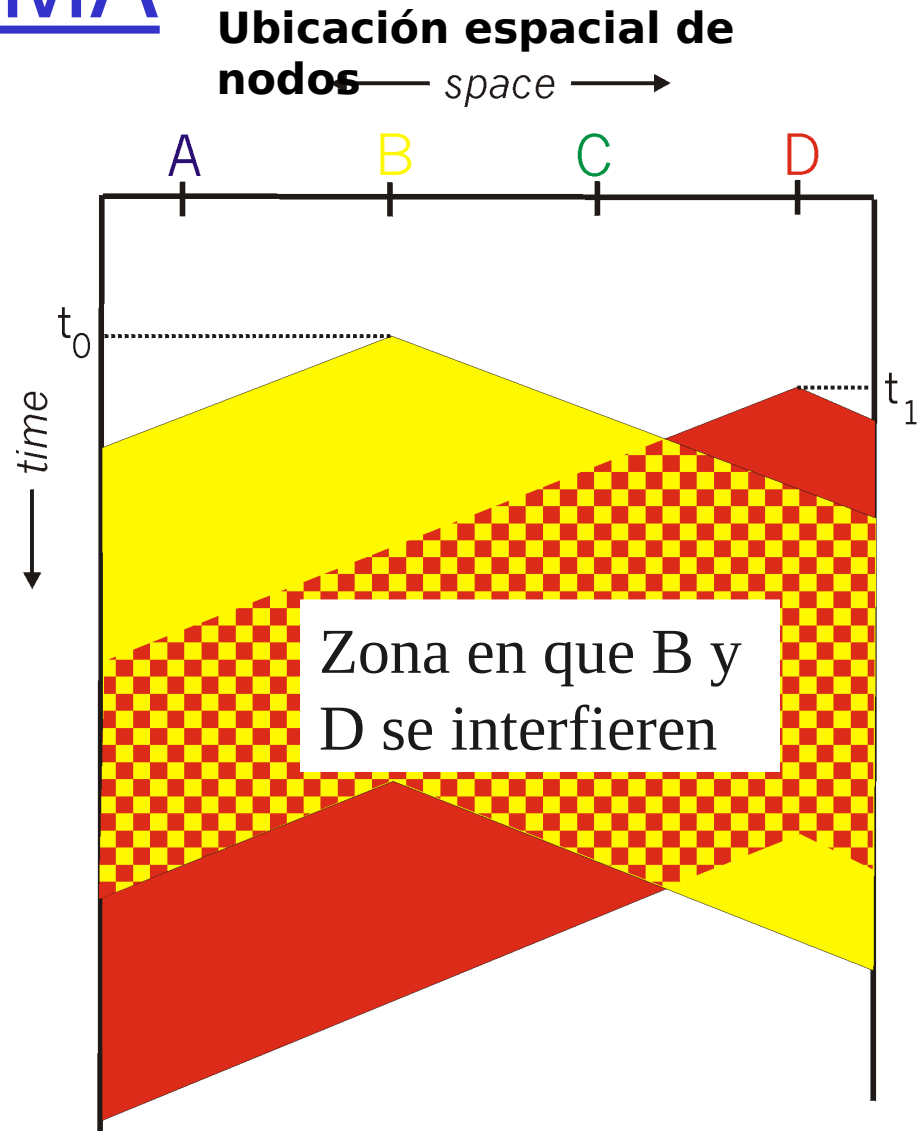
CSMA: Sensa señal portadora antes de transmitir:

- Si el canal se sensa libre, se transmite la trama entera
- Si el canal se detecta ocupado, postergar transmisión

- Analogía humana: no interrumpir mientras otros hablan!

Colisiones en CSMA

- Colisiones pueden ocurrir aún:
Retardo de propagación hace que dos nodos podrían no escuchar sus transmisiones
- Colisión:
El tiempo de transmisión del paquete entero es desaprovechado
- Notar:
El rol de la distancia y el retardo de propagación en la determinación de la probabilidad de colisión

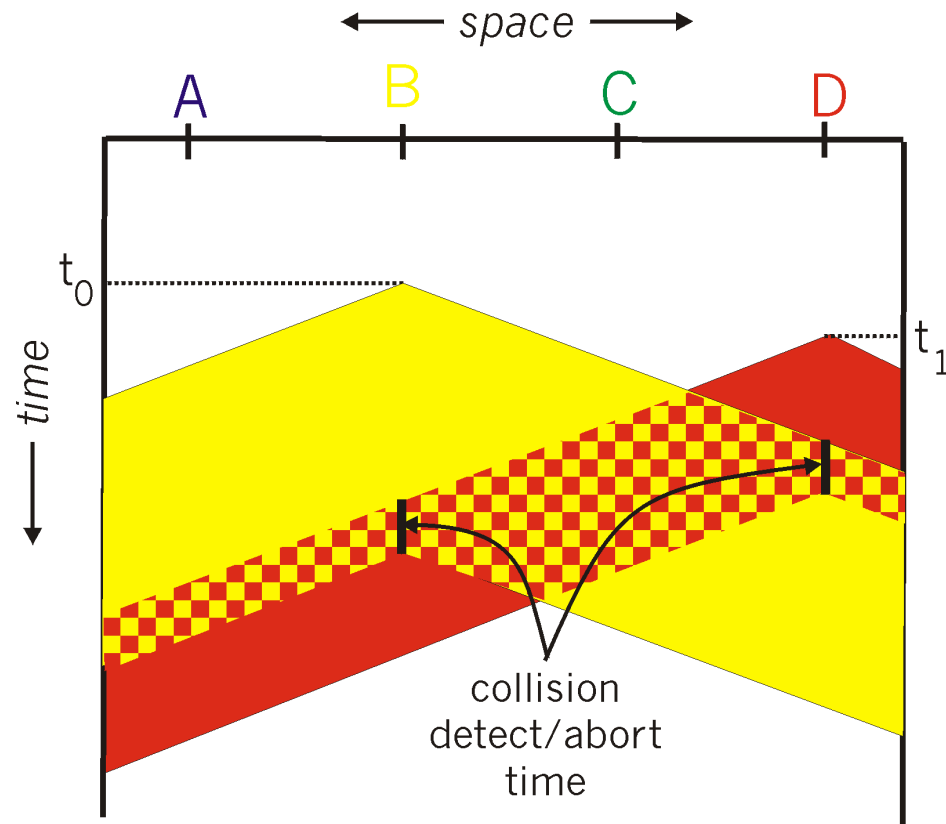


CSMA/CD (Detección de Colisiones)

CSMA/CD: carrier sensing, similar a CSMA

- colisiones son *detectadas* en corto tiempo
- Transmisiones en colisión son abortadas, reduciendo el mal uso del canal
- Detección de colisiones:
 - Fácil en LANs cableadas: se mide la potencia de la señal, se compara señales transmitidas con recibidas
 - Difícil LANs inalámbricas: receptor es apagado mientras se transmite
- Analogía humana: Conversadores respetuosos

CSMA/CD detección de colisiones



Protocolos MAC de “toma de turnos”

Vimos: Protocolos MAC que particionan el canal:

- Se comparte el canal eficientemente y equitativamente en alta carga
- Son ineficiente a baja carga: Hay retardo en acceso al canal, $1/N$ del ancho de banda es asignado aún si hay sólo un nodo activo!

Vimos: Protocolos de acceso aleatorio

- Son eficientes a baja carga: un único canal puede utilizar completamente el canal
- Alta carga: ineficiencias por colisiones

Idea: Protocolos de “toma de turnos”

- Buscan lo mejor de ambos mundos!

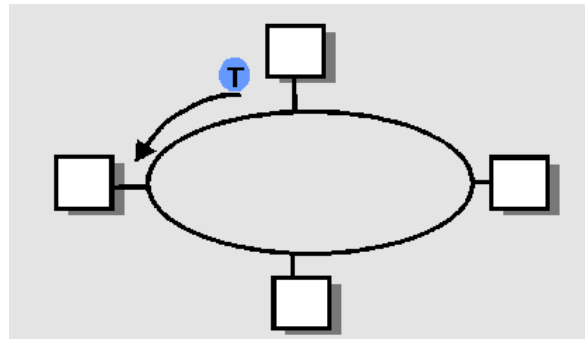
Protocolos MAC de “Toma de turnos”

Consulta:

- Nodo maestro “invita” a nodos esclavos a transmitir en turnos
- preocupaciones:
 - Overhead de la consulta
 - Latencia
 - Punto único de falla (maestro)

Paso de Token (Testimonio):

- **Token** (objeto) de control es pasado de nodo en nodo secuencialmente.
- Hay un mensaje con el token
- Preocupaciones:
 - Overhead del token
 - Latencia
 - Punto único de falla (el token)



Resumen de protocolos MAC

- ¿Qué hacemos en un medio compartido?
 - Subdivisión del canal: por tiempo, frecuencia, o código
 - Subdivisión aleatoria (dinámica),
 - ALOHA, ALOHA-R, CSMA, CSMA/CD
 - Sensado de portadora: fácil en algunas tecnologías (cable), difícil en otras (inalámbricas)
 - CSMA/CD (collision detection) es usado en Ethernet
 - CSMA/CA (collision avoidance) es usado en 802.11
 - Toma de turnos
 - Consultas desde un sitio central, o pasando un token

Capa Enlace de Datos

- 5.1 Introducción y servicios
- 5.2 Detección y corrección de errores
- 5.3 Protocolos de acceso múltiple
- 5.4 Direccionamiento de capa enlace
- 5.5 Ethernet
- 5.6 Hubs y switches
- 5.7 PPP
- 5.8 Enlaces Virtuales: ATM y MPLS