

# Capítulo 5:

## Capa de Enlace de Datos

# 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

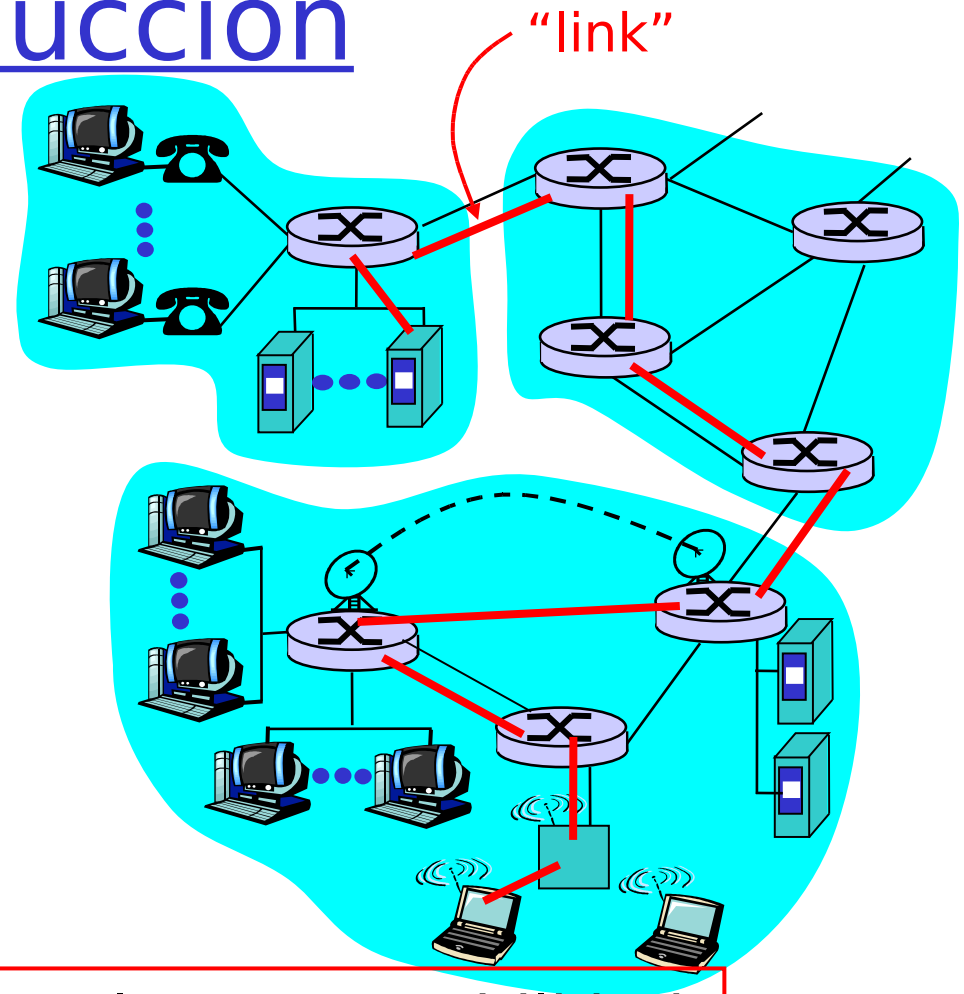
# Capa Enlace de Datos

- ❑ 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 un **frame (o trama)**, encapsula 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 el datagrama en 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

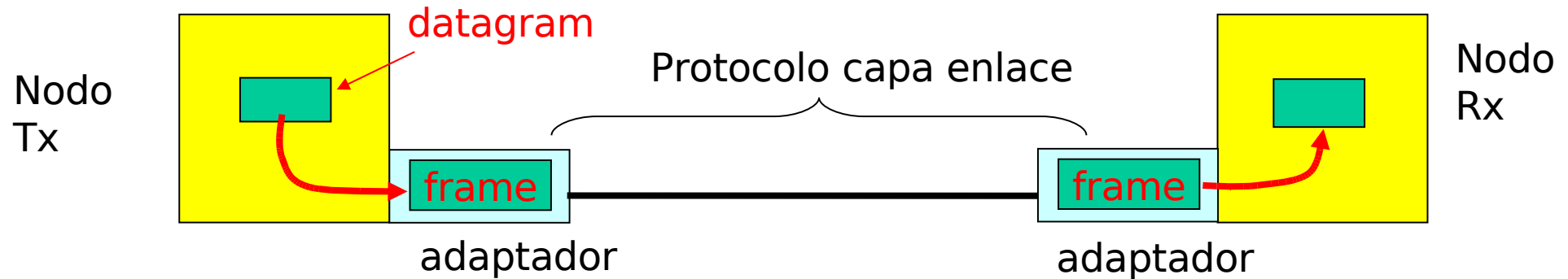
## ❑ *Corrección de Errores:*

- 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
  - 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

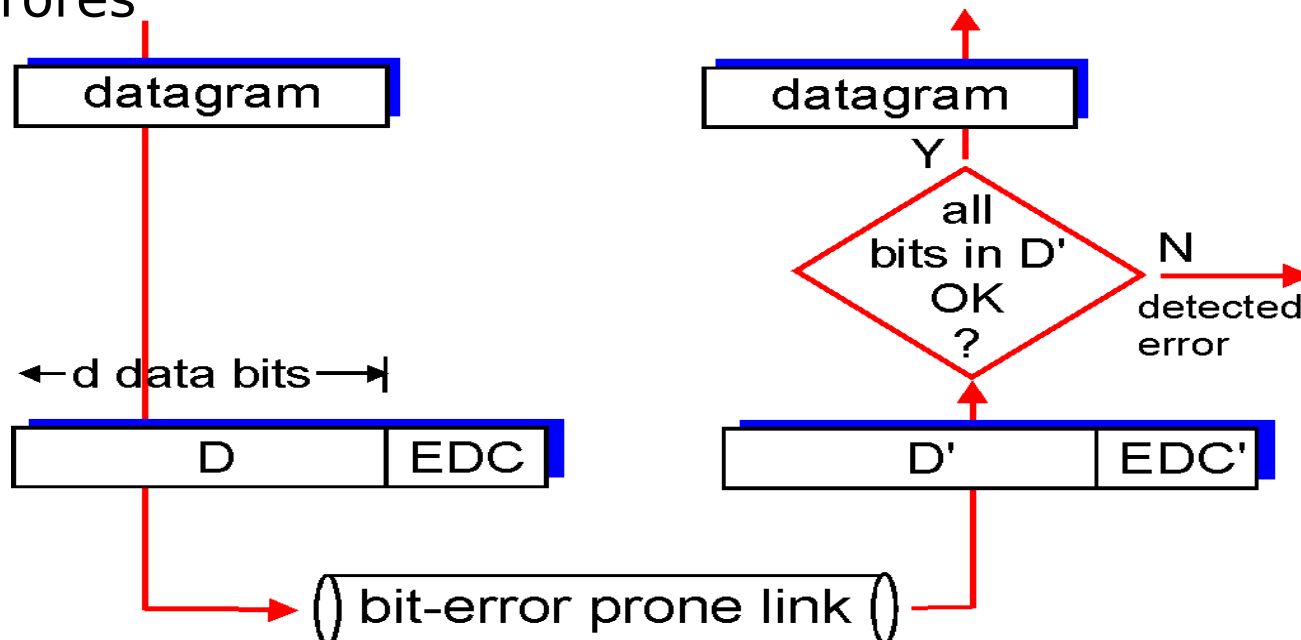
- ❑ 5.1 Introducción y servicios
- ❑ 5.2 Detección y corrección de errores
- ❑ 5.3 protocolos de acceso múltiple
- ❑ 5.4 Direccionamiento de capa enlace
- ❑ 5.5 Ethernet
- ❑ 5.6 Hubs y switches
- ❑ 5.7 PPP
- ❑ 5.8 Enlaces Virtuales: ATM y MPLS

# Detección de Errores

EDC= Error Detection and Correction bits (redundancia)

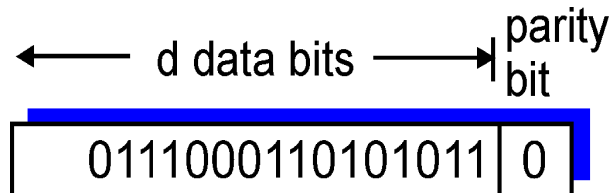
D = Datos protegidos por chequeo de errores podría incluir campos de encabezado

- 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

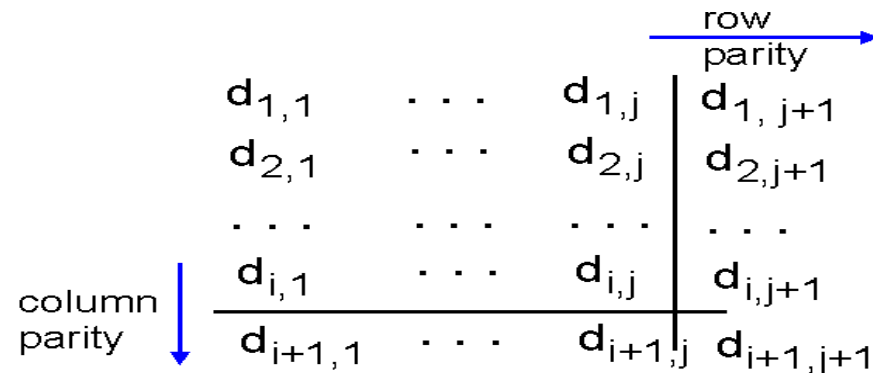


# Chequeo de paridad

## Bit de Paridad Simple: Detecta errores simples



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



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

*no errors*

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

*correctable  
single bit error*

# Cheksum de Internet

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

## Transmisor:

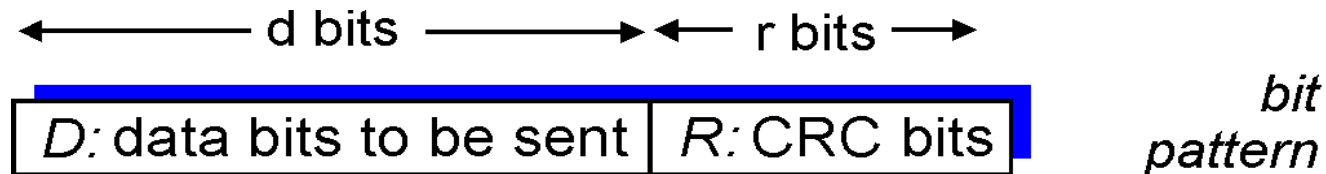
- ❑ Trata el contenido de los segmentos como una secuencia de enteros de 16 bits
- ❑ checksum: suma del contenido del segmento (complemento 1 de la suma)
- ❑ Tx pone el valor del checksum en el campo correspondiente de UDP o TCP

## 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  $G$  (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
- ❑ Ampliamente usado en la práctica (ATM, HDCL)



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

*mathematical formula*

# CRC: Ejemplo

Queremos:

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

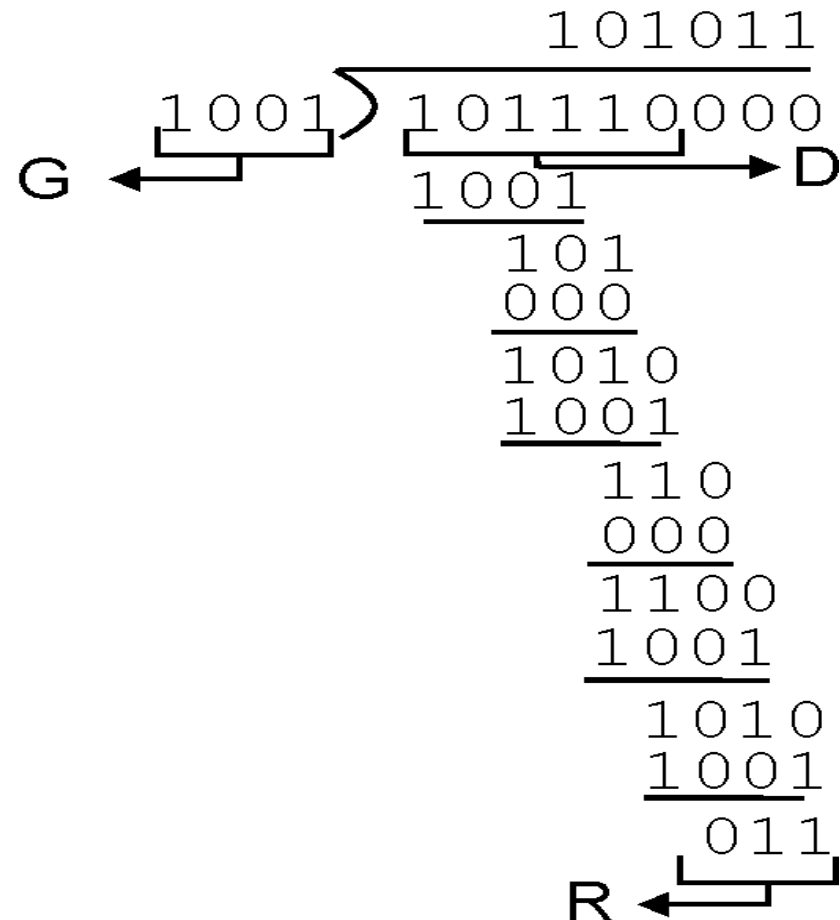
*equivalentemente:*

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

*equivalentemente:*

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

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



# 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

# Enlaces y Protocolos de Acceso Múltiple

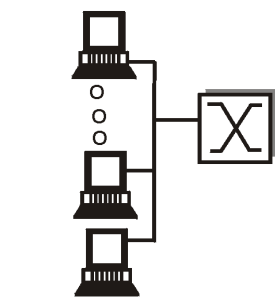
Dos tipos de “enlaces” :

❑ Punto-a-punto

- PPP para acceso discado
- Enlaces punto-a-punto entre switch Ethernet y host (computador)

❑ **broadcast** (cable o medio compartido)

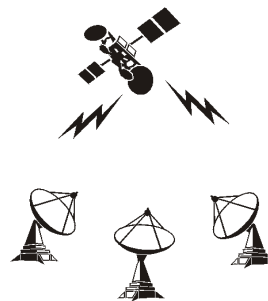
- Ethernet tradicional
- Flujo de subida en HFC (**H**ybrid **F**iber **C**oax)
- 802.11 LAN inalámbrica



shared wire  
(e.g. Ethernet)



shared wireless  
(e.g. Wavelan)



satellite



cocktail party



# 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
- ❑ La comunicación para ponerse de acuerdo sobre cómo compartir debe usar el mismo canal!
  - no hay canal “fuera de banda” para coordinación

# Protocolo de Acceso Múltiple Ideal

Supongamos un canal broadcast de tasa  $R$  bps

1. Cuando un nodo quiere transmitir, este puede enviar a tasa  $R$ .
2. Cuando  $M$  nodos quieren transmitir, cada uno puede enviar en promedio una tasa  $R/M$
3. Completamente descentralizado:
  - No hay nodo especial para coordinar transmisiones
  - No hay sincronización de reloj o ranuras
4. Es simple

# Taxonomía de protocolos MAC

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

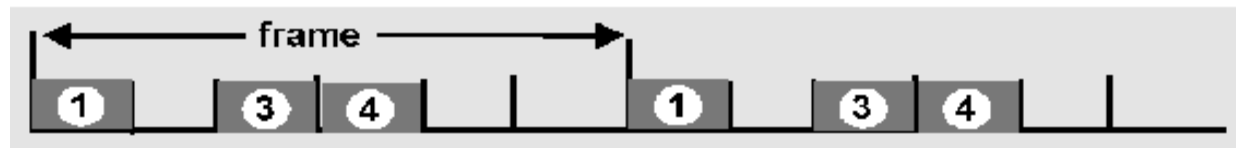
## ❑ “Tomando turnos”

- 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

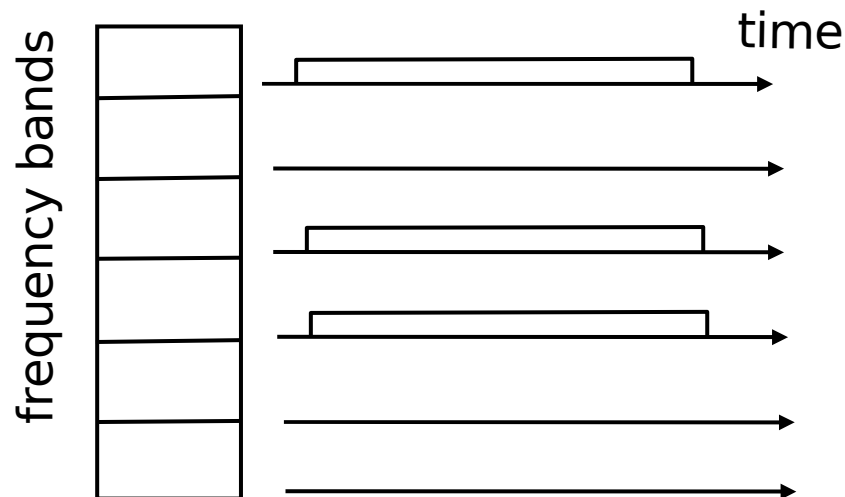
- ❑ Acceso a canales es en “rondas”
- ❑ Cada estación obtiene una ranura de largo fijo (largo= tiempo transmisión del paquete) en cada ronda
- ❑ Ranuras no usadas no se aprovechan
- ❑ ejemplo: LAN con 6 estaciones, 1,3,4 tienen paquetes, ranuras 2,5,6 no usadas



# Protocolos MAC en canal Particionado: 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



# 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

# ALOHA ranurado

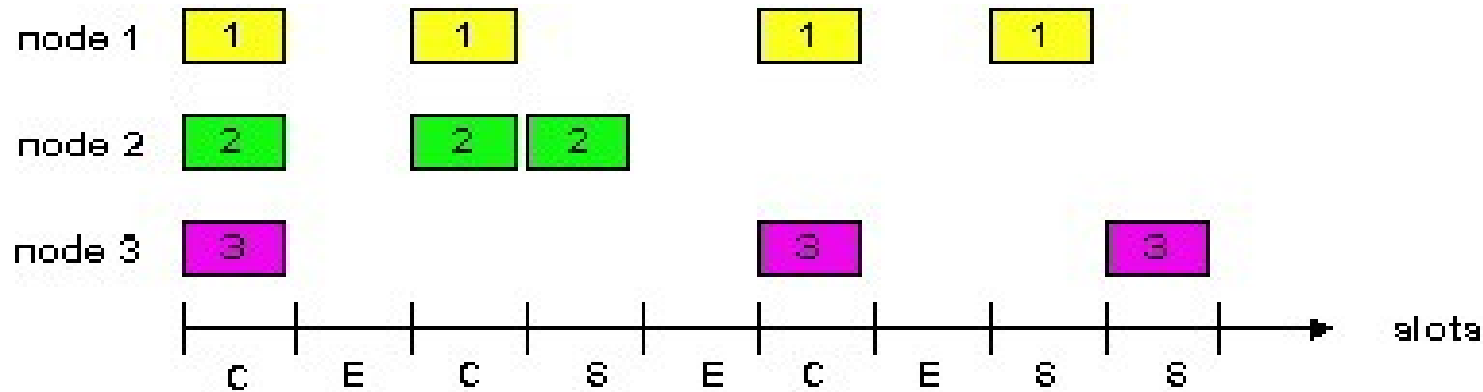
## Suposiciones

- ❑ Todos las tramas tienen igual tamaño
- ❑ Tiempo es dividido en igual tamaño de ranura, = tiempo para enviar una trama
- ❑ Nodos comienzan a transmitir sólo al inicio de cada ranura
- ❑ Nodos están sincronizados
- ❑ Si 2 o 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 subsiguiente con probabilidad  $p$  hasta transmisión exitosa

# ALOHA ranurado



## Ventajas

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

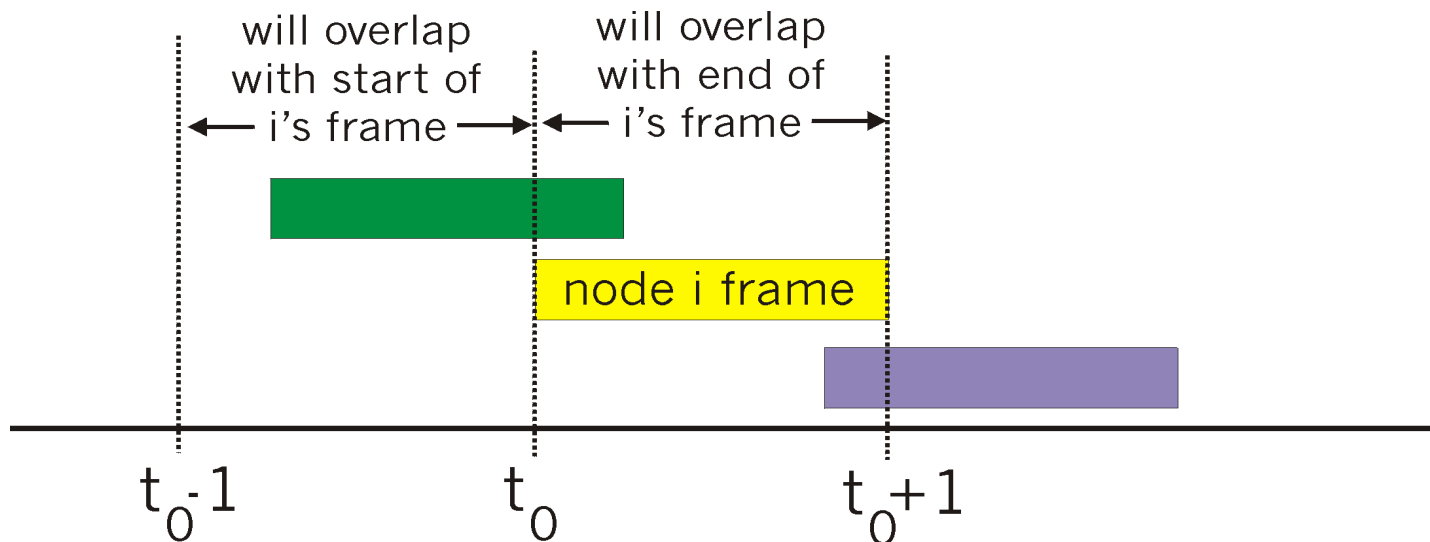
## Desventajas

- ❑ colisiones, ranuras desperdiciadas
- ❑ Ranuras no ocupadas
- ❑ Nodos podrían detectar la colisión en menor tiempo que el de transmitir un paquete
- ❑ Sincronización de relojes
- ❑ En mejor caso se logra 37% de utilización



# 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 a  $t_0$  colisiona con otras tramas enviadas en  $[t_0-1, t_0+1]$
- ❑ Probabilidad de éxito de transmisión de un nodo 18%



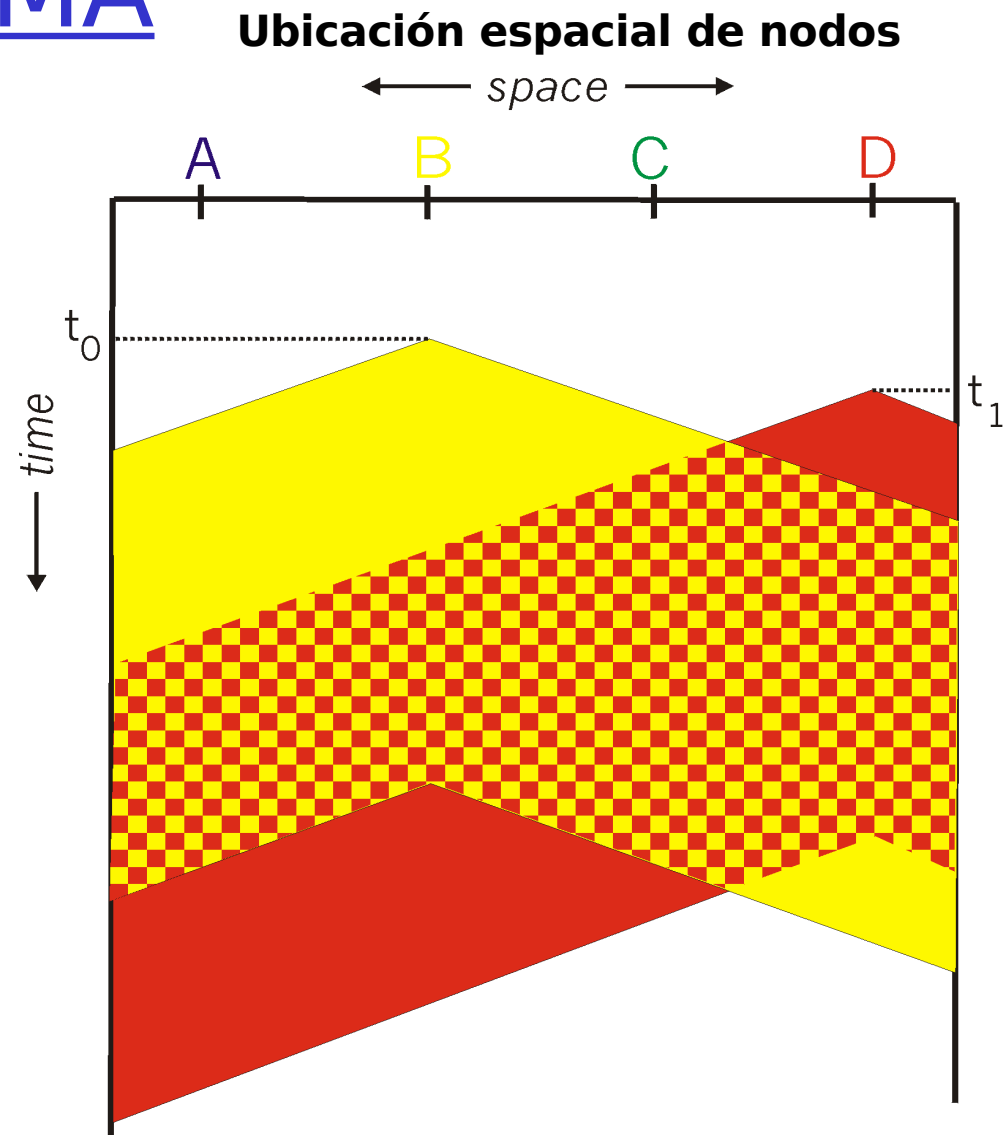
# CSMA (Carrier Sense Multiple Access)

**CSMA:** Sensa portadora antes de transmitir:

- ❑ Si el canal se sensa a libre, se transmite la trama entera
- ❑ Si el canal se sensa ocupado, postergar transmisión
  
- ❑ Analogía humana: no interrumpir a otros!

# 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

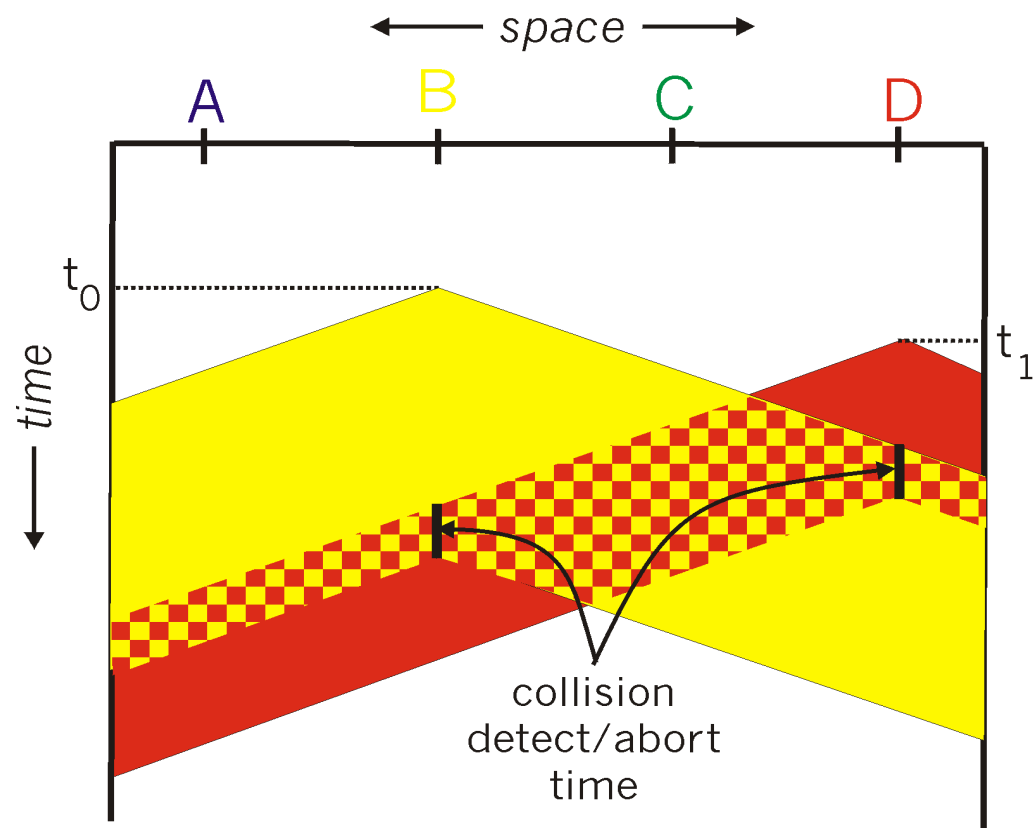
- colisiones son *detectadas* en corto tiempo
- Transmisiones en colisión son abortadas, reduciendo el mal uso del canal

## □ 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”

## 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!

## Protocolos de acceso aleatorio

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

## Protocolos de “toma de turnos”

- Buscan lo mejor de ambos mundos!

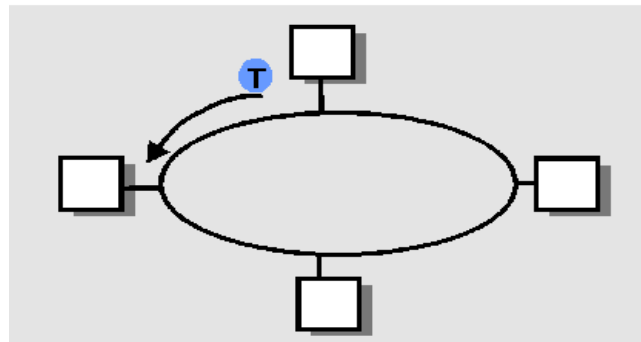
# Protocolos MAC de “Toma de turnos”

## Consulta:

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

## Paso de Testimonio:

- ❑ **Token** (testimonio) 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 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