Tecnología Digital IV: Redes de Computadoras

Clase 19: Nivel de Enlace - Parte 1

Lucio Santi & Emmanuel Iarussi

Licenciatura en Tecnología Digital
Universidad Torcuato Di Tella

29 de mayo de 2025

Agenda

- Introducción al Nivel de Enlace
  - Funcionalidades
  - Servicios
- Métodos de detección y corrección de errores
- Protocolos de acceso a medios compartidos
- LANs
  - Direccionamiento y ARP
  - Ethernet
  - Switches
  - Redes de datacenters

Objetivos

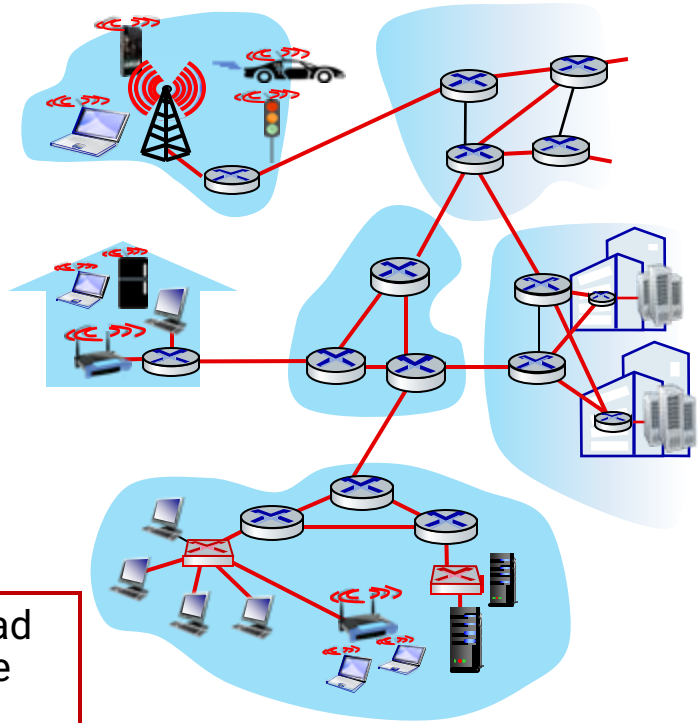
- Entender los principios detrás de los servicios del Nivel de Enlace:
  - Detección y corrección de errores
  - Canales compartidos: acceso múltiple
  - Direccionamiento
  - Redes de área local: Ethernet y VLANs
- Instanciación e implementación de distintas tecnologías

Introducción al Nivel de Enlace

Nivel de Enlace

Terminología:

- hosts y routers: **nodos**
- Canales de comunicación que conectan nodos adyacentes a lo largo de un camino: **enlaces**
  - Cableados
  - Inalámbricos
  - LANs
- Paquete de nivel 2: ***frame*** (encapsula datagramas)



El nivel de enlace tiene la responsabilidad de transmitir datagramas por un enlace desde un nodo hacia otro nodo físicamente adyacente

Contexto

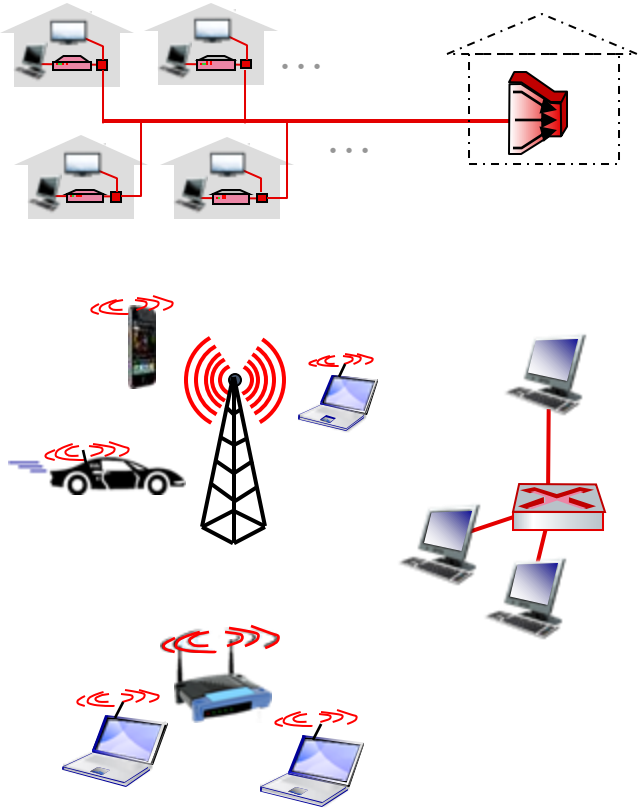
- Datagramas transmitidos por distintos protocolos de enlace sobre distintos enlaces:
  - ej., WiFi en el primer enlace y Ethernet en el siguiente
- Cada protocolo ofrece diferentes servicios
  - ej., puede o no proveer garantías de confiabilidad

Analogía:

- Viaje de UTDT a UNS (Bahía Blanca)
  - Cabify: de UTDT a Aeroparque (AEP)
  - Avión: de AEP a BHI
  - Colectivo: de BHI a UNS
- Alumno: datagrama
- Segmento del viaje: enlace
- Modo de transporte: protocolo de enlace
- Agente de viajes: algoritmo de ruteo

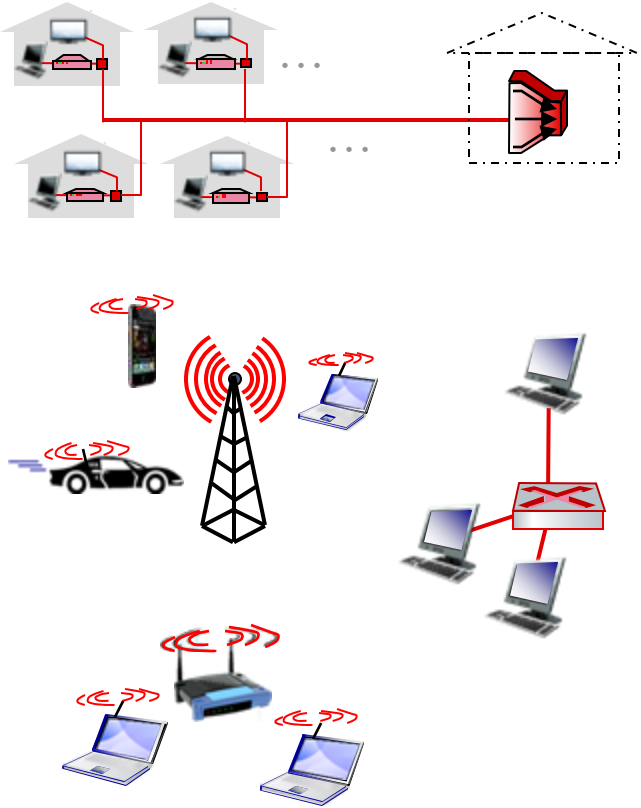
Servicios del Nivel de Enlace

- *Framing* y acceso al medio
  - Encapsular datagramas en *frames* (agregando *header* y *trailer*)
  - Acceder al canal (si el medio es compartido)
  - **Direcciones MAC** en los headers de los frames para identificar interlocutores (distintas de las direcciones IP)
- Entrega confiable entre nodos adyacentes
  - Ya sabemos cómo hacer esto
  - No suele emplearse en enlaces con tasas de error bajas
  - Enlaces inalámbricos: **tasas de error elevadas**



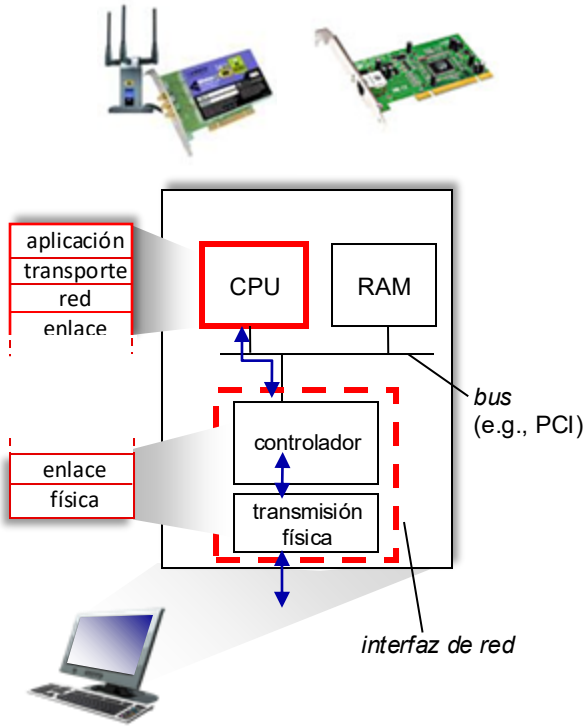
Servicios del Nivel de Enlace

- Control de flujo
  - "Ritmo de transmisión" entre nodos adyacentes
- Detección de errores
  - Errores causados ej. por ruido o atenuación de la señal
  - El receptor detecta errores y solicita retransmisión (o descarta el *frame*)
- Corrección de errores
  - El receptor identifica y **corrige** errores de bits sin retransmisiones
- *Half-duplex* y *full-duplex*
  - En *half-duplex*, los interlocutores pueden transmitir pero no al mismo tiempo

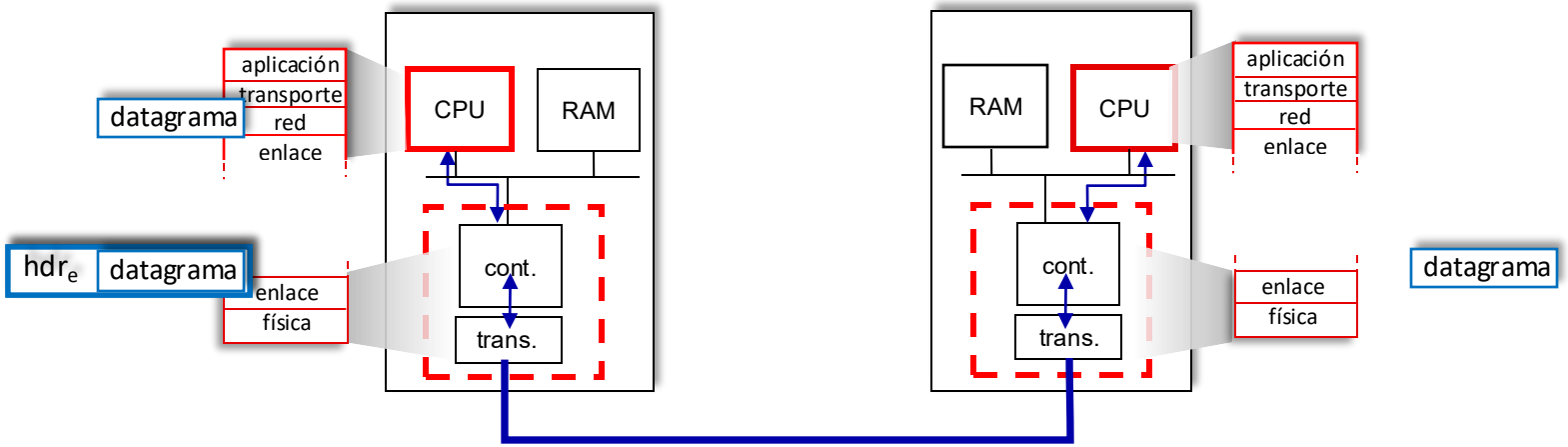


Implementación del Nivel de Enlace

- Presente en **todos** los hosts y dispositivos
- Se implementa en la placa de red (NIC, *Network Interface Card*) o en un chip
  - ej. placa Ethernet o WiFi
  - Implementa capa de enlace y capa física
- Se conecta a los *buses* del host
- Combinación de hardware, software y firmware



Comunicación entre nodos



Emisor

- Encapsula datagrama en un *frame*
- Agrega bits para control de errores, entrega confiable, control de flujo, etc.

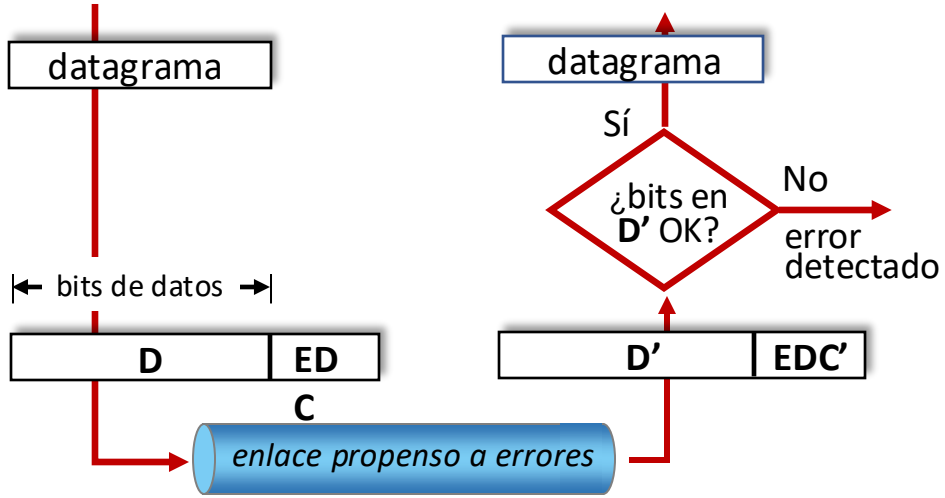
Receptor

- Verifica si hay errores
- Extrae datagrama y lo pasa a la capa superior

Mecanismos de detección y corrección de errores

Detección de errores

Bits de detección/corrección de errores: **EDC** (e.g., redundancia)
Datos (**D**) protegidos por el control de errores (puede incluir campos del *header*)



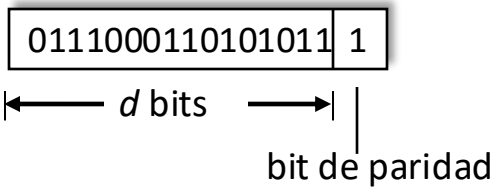
La detección **puede fallar**

- Algunos errores pueden quedar enmascarados
- Cuanto más bits tenga el campo **EDC**, (generalmente) mayor será la efectividad de la detección/corrección

Control de paridad

Paridad de un bit

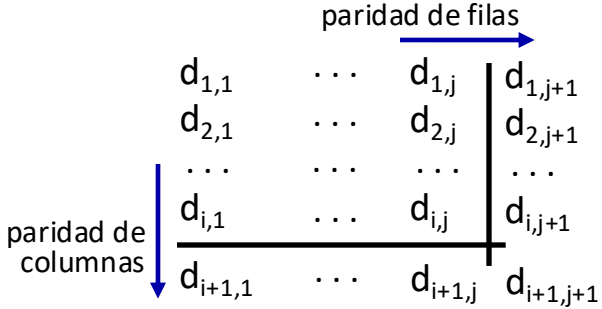
- Detecta errores de sólo un bit



Paridad par: poner el bit de paridad de forma tal que haya una cantidad par de unos

Paridad bidimensional

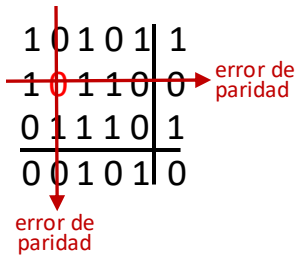
- Detecta y corrige errores de sólo un bit



sin errores:

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

error en un bit detectable y corregible:



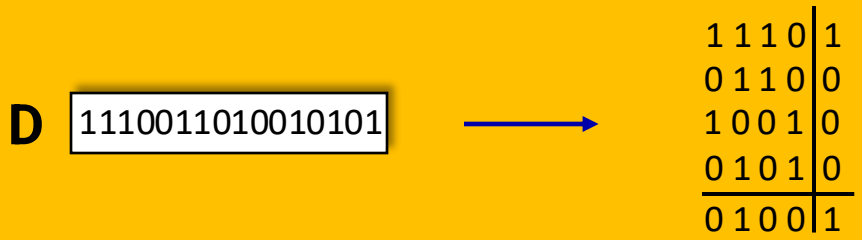
Ejercicio!

¿Cuáles serían los valores del campo que contiene los bits de paridad para el caso de un esquema de paridad bidimensional (par)? La respuesta debe minimizar la cantidad de bits de paridad utilizados.

D 1110011010010101

Ejercicio!

¿Cuáles serían los valores del campo que contiene los bits de paridad para el caso de un esquema de paridad bidimensional (par)? La respuesta debe minimizar la cantidad de bits de paridad utilizados.



Checksum de Internet (repaso)

Objetivo: detectar errores en los bits de los segmentos

Emisor

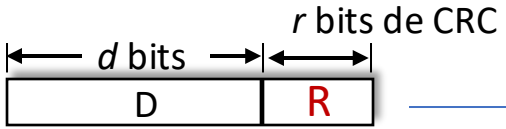
- Interpreta el contenido del segmento (incluyendo *header* + direcciones IP) como una tira de enteros de 16 bits
- *Checksum*: complemento a uno de la suma en complemento a uno del contenido del segmento
- Este valor se coloca en el campo *checksum* del *header* de transporte

Receptor

- Calcula el *checksum* del segmento recibido
- Comprueba si todos los bits son 0s:
  - Si no lo son: error!
  - Si lo son: no se detecta error… pero podría haberlos!

Cyclic Redundancy Check (CRC)

- Método de detección de errores más poderoso
- D: bits de datos (interpretados como un polinomio con coeficientes binarios)
- G: polinomio de grado *r* (*r* + 1 bits): *generador,* viene dado por el método



bits a transmitir

$<D,R> = D \cdot 2^r \text{ XOR } R$

$= x^r D(x) + R(x)$

fórmula para calcularlos

Objetivo: elegir *r* bits de CRC, R, tales que <D,R> sea divisible por G

- El receptor conoce G: divide <D,R> por G y detecta errores si obtiene un resto no nulo
- Puede detectar todos los errores en ráfagas de a lo sumo *r* bits
- Muy usado en la práctica (ej., en Ethernet y en WiFi)

CRC: ejemplo

- Supongamos CRC de 3 bits con generador G = $x^3 + x + 1$
- Supongamos que queremos transmitir el byte

D = 10010101 ($\Rightarrow$ D(x) = $x^7 + x^4 + x^2 + 1$)

1. Calculamos el resto R al dividir x^3D(x) por G:

$x^{10} + x^7 + x^5 + x^3 \mid x^3 + x + 1$

$x^{10} + x^8 + x^7$ $\qquad$ $x^7 + x^5 + x^3 + x$

$x^8 + x^5 + x^3$

$x^8 + x^6 + x^5$

$x^6 + x^3$

$x^6 + x^4 + x^3$

---

x^4

$x^4 + x^2 + x$

$\boldsymbol{x^2 + x}$ $\longrightarrow$ 3 bits: 110 = $1x^2 + 1x + 0$

2. Transmitimos <D,R> = ($x^{10} + x^7 + x^5 + x^3$) + ($x^2 + x$) = 10010101110

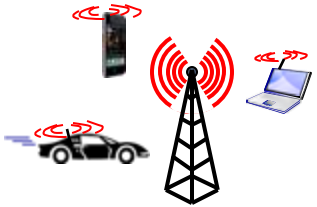
Acceso a medios compartidos

Enlaces (y protocolos) de acceso compartido

Dos tipos de “enlaces”

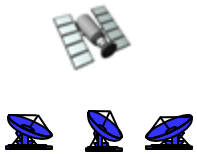
- *Punto a punto*
  - ej., enlace punto a punto entre host y switch en Ethernet
  - Protocolo PPP (e.g. acceso *dial-up*, vía línea telefónica)
- *Broadcast* (cable o medio compartido)
  - La Ethernet antigua
  - Upstream HFC en redes de acceso por cable
  - LANs WiFi (802.11); 4G/5G; acceso satelital

cable compartido Ethernet



acceso inalámbrico 4G/5G

acceso inalámbrico WiFi



acceso inalámbrico satélite

humanos en una fiesta (aire compartido)

Protocolos de acceso múltiple

- Canal *broadcast* compartido
- Si dos o más nodos transmiten simultáneamente, se produce **interferencia**
  - Colisión: los nodos reciben dos o más señales en el mismo instante

protocolo de acceso múltiple

- Algoritmo distribuido que determina cómo se comparte el canal entre los nodos (i.e., determina cuándo cierto nodo puede transmitir)
- La comunicación de control sobre la compartición del canal debe usar el mismo canal que se desea arbitrar
  - No hay otro canal “fuera de banda” para coordinar

Protocolos de acceso múltiple: características

Dado: canal de acceso múltiple (**MAC**) con tasa *R* bps

Características deseadas:

1. Cuando sólo un nodo quiere transmitir, puede hacerlo a tasa *R*
2. Cuando *M* nodos quieren transmitir, cada uno puede hacerlo a una tasa promedio *R*/*M*
3. Descentralización:
   - No hay un nodo coordinador (punto de falla)
   - No hay sincronización de relojes o *slots*
4. Simplicidad (“fácil” de implementar)

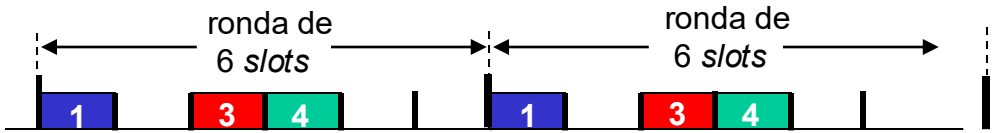
Taxonomía de protocolos MAC

- Canal particionado
  - El canal se divide en “piezas” más chicas (e.g. *slots* temporales o frecuencias)
  - Se reserva una tal pieza para uso exclusivo de un nodo
- Acceso aleatorio
  - El canal no se divide; se permiten colisiones
  - Ofrece mecanismos de recuperación ante colisiones
- Por turnos
  - Los nodos se turnan para transmitir
  - Los que tienen más datos pueden tomar turnos más largos

Canal particionado: TDMA

TDMA: *Time Division Multiple Access*

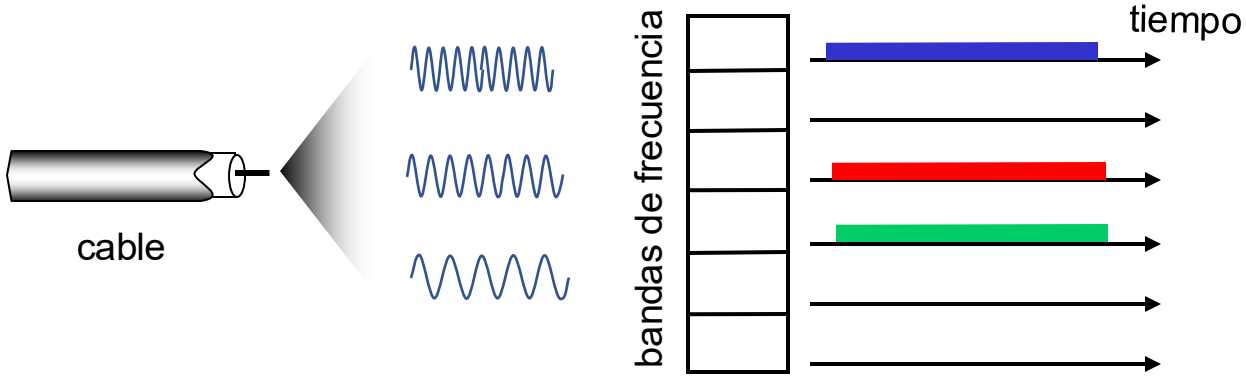
- Acceso al canal en “rondas”
- Cada nodo tiene un *slot* de longitud fija en cada ronda (*longitud*: tiempo de transmisión de un paquete)
- Los *slots* que no se usan quedan inactivos (*idle*)
- Ejemplo: LAN de 6 nodos donde 1,3,4 tienen paquetes para enviar; los *slots* 2,5,6 quedan inactivos



Canal particionado: FDMA

FDMA: *Frequency Division Multiple Access*

- El espectro del canal se divide en bandas de frecuencia
- A cada nodo se le asigna una banda fija
- Puede haber intervalos de tiempo inactivos en cada banda
- Ejemplo: LAN de 6 nodos donde 1,3,4 tienen paquetes para enviar; las bandas 2,5,6 permanecen inactivas



Protocolos de acceso aleatorio

- Cuando un nodo tiene un paquete para enviar,
  - Transmite a la capacidad total del canal, *R*
  - No hay coordinación previa entre los nodos
- Se produce una colisión si dos o más nodos transmiten
- Los protocolos de acceso aleatorio determinan:
  - Cómo detectar colisiones
  - Cómo recuperarse ante estas (e.g., vía retransmisiones con retrasos)
- Protocolos de ejemplo:
  - ALOHA, slotted ALOHA
  - CSMA, CSMA/CD (Ethernet), CSMA/CA (WiFi)

Slotted ALOHA

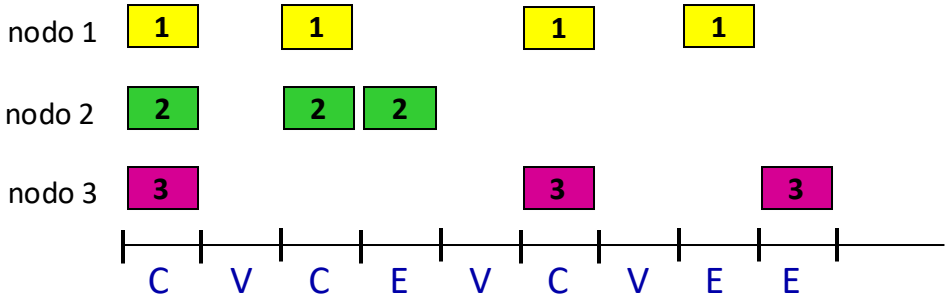
Suposiciones

- Los frames tienen igual tamaño
- El tiempo se divide en *slots* de igual tamaño (tiempo de transmisión de un frame)
- Los nodos empiezan a transmitir sólo al comienzo de un *slot*
- Los nodos están sincronizados
- Si dos o más transmiten en un *slot*, todos detectan la colisión

Operatoria

- Cuando un nodo recibe un frame, transmite en el siguiente *slot*
  - *Si no hay colisión:* el nodo puede enviar un nuevo frame en el siguiente *slot*
  - *Si hay colisión:* el nodo retransmite el frame en cada *slot* subsiguiente con probabilidad *p* (hasta tener éxito)

Slotted ALOHA



C: colisión
E: éxito
V: vacío

Ventajas

- Si hay un único nodo activo, puede transmitir continuamente a la tasa del canal
- Descentralizado: sólo los *slots* en los nodos deben estar sincronizados
- Simple

Desventajas

- Colisiones; *slots* desperdiciados
- *Slots* inactivos (vacíos)
- Sincronización de relojes

Eficiencia de Slotted ALOHA

Eficiencia: fracción de slots exitosos en el largo plazo (muchos nodos, cada uno con muchos frames a transmitir)

- Supongamos *N* nodos con muchos frames a enviar; cada uno transmite en un *slot* con probabilidad *p*
  - Probabilidad de que un nodo tenga éxito en un slot: $p\,(1-p)^{N-1}$
  - Probabilidad de que cualquier nodo tenga éxito: $N\,p\,(1-p)^{N-1}$
  - Eficiencia máxima: encontrar p^* que maximiza $N\,p\,(1-p)^{N-1}$
  - Para muchos nodos, tomar el límite de $N\,p^*\,(1-p^*)^{N-1}$ cuando $N \rightarrow \infty$

  Eficiencia máxima: $1/e = 0.37$
- En el mejor caso, el canal se utiliza el **~37% del tiempo**

CSMA (*Carrier Sense Multiple Access*)

CSMA simple: **escuchar** antes de transmitir

- Si el canal está **inactivo**: transmitir el frame completo
- Si está **ocupado**: diferir la transmisión

- Analogía en humanos: no interrumpir a los demás

CSMA/CD: CSMA con detección de colisiones

- Las colisiones se detectan en un período corto
- Las transmisiones involucradas se abortan, reduciendo así el desperdicio del canal
- Detectar colisiones es fácil en medios cableados pero **difícil** en entornos inalámbricos

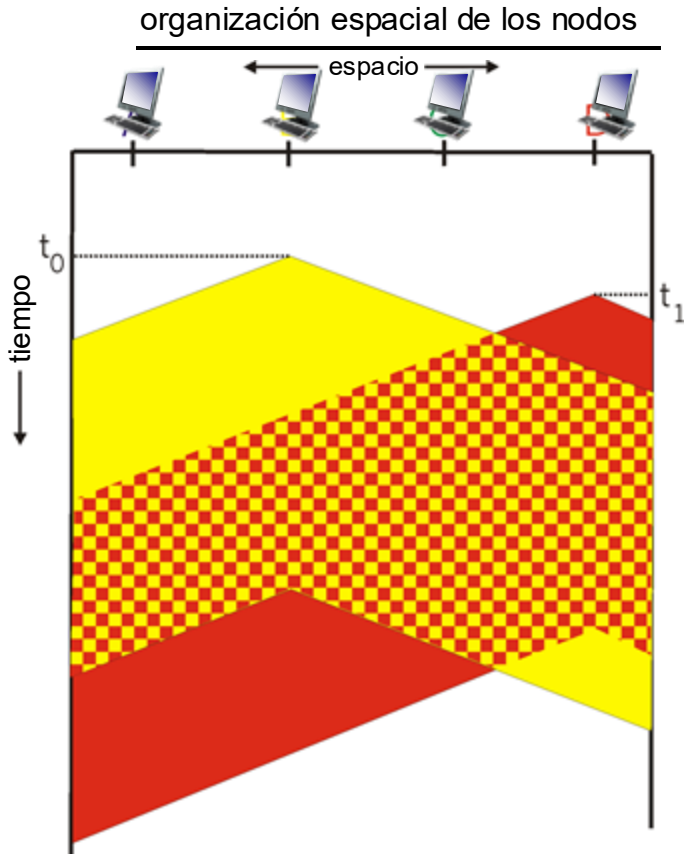
- Analogía en humanos: un interlocutor educado

Colisiones en CSMA

- Las colisiones pueden existir aún con *carrier sensing*
  - Dado el tiempo de propagación de la señal, dos nodos podrían no escuchar las transmisiones del otro cuando recién comienzan
- Colisión: desperdicia el tiempo completo de transmisión del paquete
  - La distancia y el tiempo de propagación son importantes para estimar la probabilidad de colisión

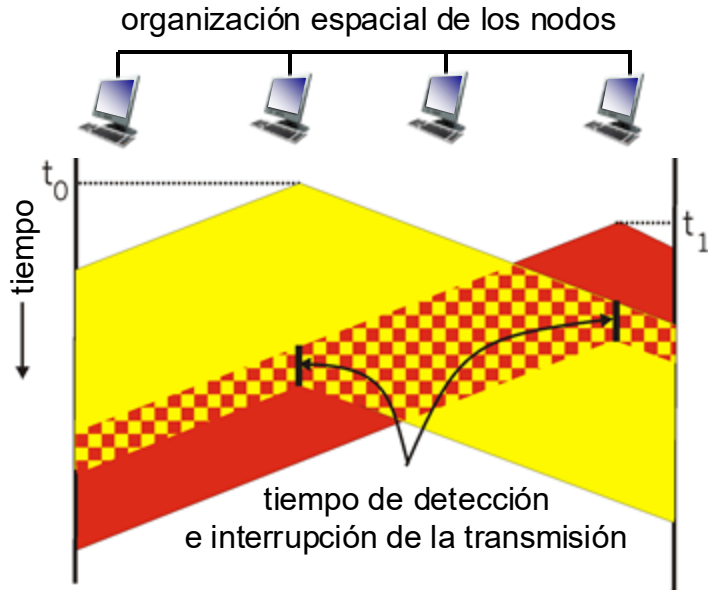
Colisiones en CSMA

- Las colisiones pueden existir aún con *carrier sensing*
  - Dado el tiempo de propagación de la señal, dos nodos podrían no escuchar las transmisiones del otro cuando recién comienzan
- Colisión: desperdicia el tiempo completo de transmisión del paquete
  - La distancia y el tiempo de propagación son importantes para estimar la probabilidad de colisión



Colisiones en CSMA/CD

- CSMA/CD reduce el tiempo desperdiciado en colisiones
  - La transmisión se aborta **inmediatamente** al detectar una colisión



El algoritmo CSMA/CD de Ethernet

1. La NIC recibe un datagrama de la capa de red y crea un frame
2. Luego, escucha el canal:
   Si está inactivo: comienza la transmisión del frame
   Si está ocupado: espera a que quede inactivo y luego transmite
3. Si la NIC transmite el frame completo sin colisiones, termina
4. Si la NIC detecta otra transmisión en curso, aborta y envía una señal de interferencia (*jamming signal*)
5. Luego de abortar, la NIC entra en *exponential backoff*:
   - Al cabo de la *m*-ésima colisión, la NIC elige un *K* al azar del intervalo [*0,1,2,…, 2^m-1*]. La NIC espera *K*·512 tiempos de bit y se vuelve al Paso 2 (escuchar el canal)
   - A más colisiones, más largo es el intervalo de *backoff*

Eficiencia de CSMA/CD

- t_{prop} = máximo delay de propagación entre dos nodos de la LAN
- t_{trans} = tiempo de transmitir un frame de tamaño máximo

$$efficiency = \frac{1}{1 + 5t_{prop}/t_{trans}}$$

- La eficiencia tiende a 1
  - Cuando t_{prop} tiende a 0 (“aborto instantáneo ante colisiones”), o bien
  - Cuando t_{trans} tiende a infinito (“transmisión muy prolongada”)
- Mejor rendimiento que ALOHA siendo simple, barato y descentralizado

El algoritmo CSMA/CA en WiFi

1 **Carrier Sense (Escucha el canal)**
Antes de transmitir, el dispositivo escucha si el canal está libre (sin otras transmisiones). Si está ocupado, espera.

2 **Collision Avoidance (Prevención de colisiones)**
Si el canal está libre, en vez de transmitir inmediatamente, espera un tiempo aleatorio (backoff) para reducir la probabilidad de colisiones con otros dispositivos que también estén esperando.

3 **RTS/CTS (Request to Send / Clear to Send)**
En algunos casos (opcional en Wi-Fi), el dispositivo envía una señal **RTS** al receptor para reservar el canal. Si el receptor responde con **CTS**, el canal queda reservado para la transmisión.

El algoritmo CSMA/CA en WiFi

4 **Transmisión y ACK (Acknowledgment)**

Una vez reservada la transmisión y enviada, el receptor responde con un **ACK** para confirmar la recepción exitosa. Si no recibe ACK, el transmisor asume que hubo colisión o pérdida y retransmite.

🏠 **Analogía cotidiana**

Imagina una reunión en la que varias personas quieren hablar, pero solo una puede hacerlo a la vez:

En **CSMA/CD** (Ethernet), todos intentan hablar a la vez y, si alguien se interrumpe, todos paran y esperan.

En **CSMA/CA** (Wi-Fi), cada persona espera su turno, levanta la mano (RTS) para hablar, recibe un asentimiento (CTS) del moderador, y luego habla. Así se evitan interrupciones

Protocolos MAC por turnos

Protocolos de canal particionado

- Comparten el canal eficientemente y de manera justa ante cargas altas
- Ineficientes ante cargas bajas: delay en acceso al canal; $1/N$ ancho de banda reservado incluso si hay sólo un nodo activo

Protocolos de acceso aleatorio

- Eficientes ante cargas bajas: un nodo único puede utilizar el canal por completo
- Sobrecarga por colisiones ante cargas altas

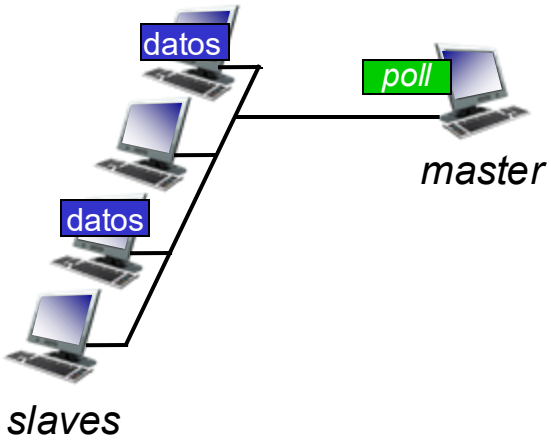
Protocolos por turnos

- Buscan obtener lo mejor de los dos paradigmas

Protocolos MAC por turnos

Polling

- Un nodo *master* sondea a los demás nodos para que transmitan de a uno por vez
- Se usa e.g. en Bluetooth
- Desventajas:
  - Delay y sobrecarga por *polling*
  - El nodo *master* constituye un punto único de falla



Protocolos MAC por turnos

Token passing

- Se pasa un ***token*** de control de un nodo al siguiente, de forma secuencial (no hay nodo *master*)
- El *token* es un frame especial (los nodos lo preservan sólo si tienen frames para transmitir)
- Desventajas:
  - Delay y sobrecarga por paso de *token*
  - El *token* introduce un punto único de falla