

**Redes de Computadores**  
*Nivel de Enlace: Introducción y  
control de acceso al medio*

Área de Ingeniería Telemática  
Dpto. Automática y Computación  
<http://www.tlm.unavarra.es/>

# En clases anteriores...

- ▶ Nivel de aplicación de Internet: sockets y protocolos de aplicación
- ▶ Nivel de transporte de Internet: TCP y UDP
- ▶ Nivel de red de Internet: IP, ICMP, direccionamiento...
- ▶ Falta saber como enviar mensajes a los vecinos conectados directamente o conectados a la misma red de área local

## En este tema:

- ▶ Nivel de enlace: redes de área local

# Objetivos

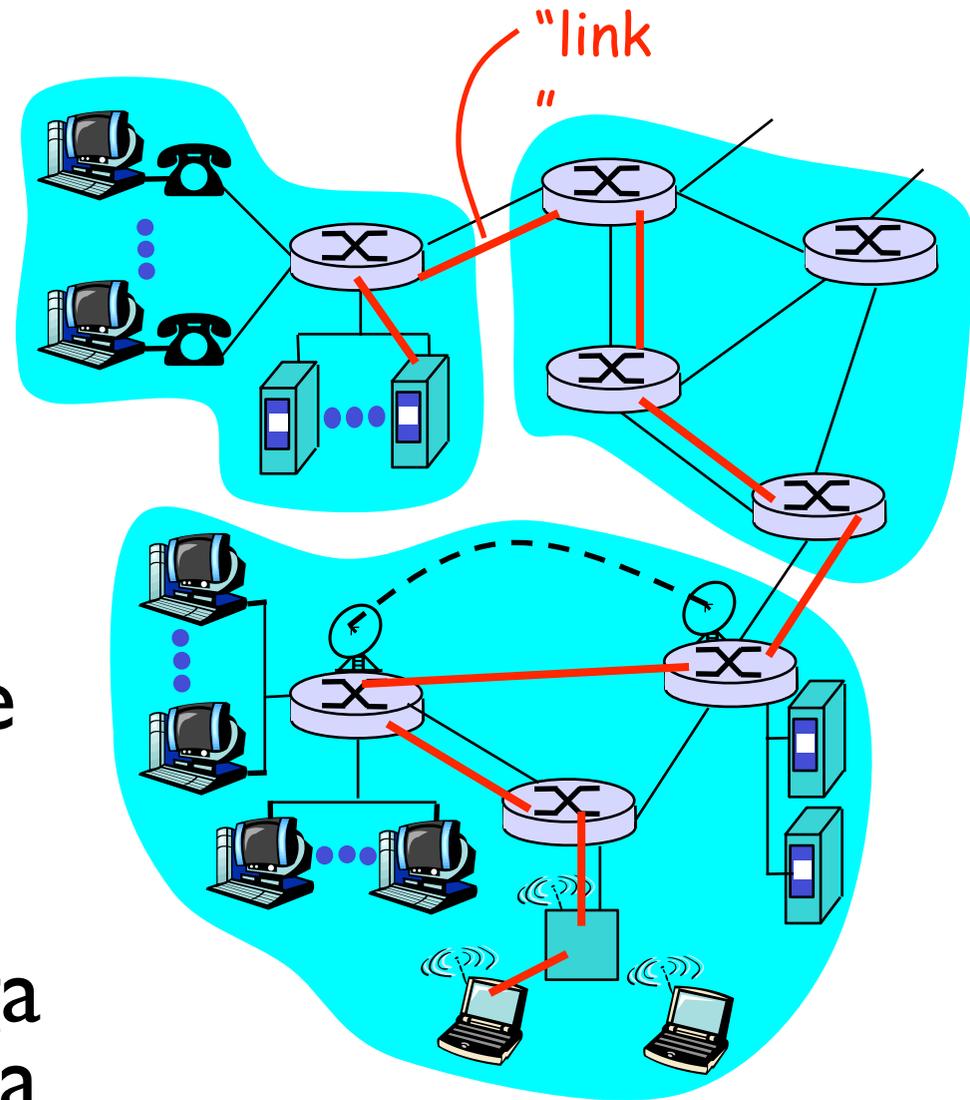
- ▶ **Comprender los conceptos y funciones fundamentales del nivel de enlace**
  - > Detección y corrección de errores
  - > Técnicas para compartir un canal de broadcast
  - > Direccionamiento de nivel de enlace
  - > Enlace de datos fiable y con control de flujo
- ▶ **Conocer el funcionamiento de algunas implementaciones de nivel de enlace**

# En la clase de hoy

- ▶ Nivel de enlace:funciones
- ▶ Detección de errores
- ▶ Acceso a medio compartido

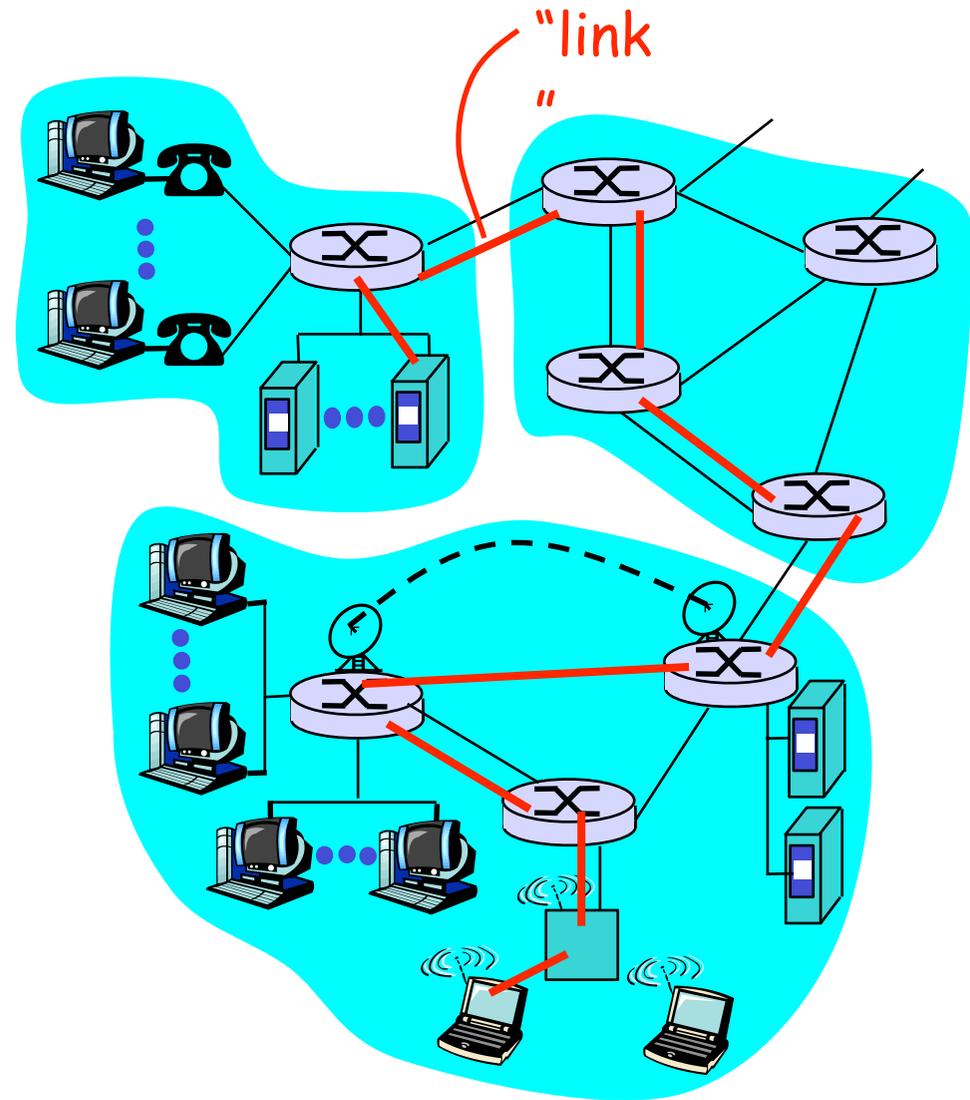
# Nivel de enlace

- ▶ routers y PCs : nodos
- ▶ conectados por: enlaces
  - > enlace de cable
  - > enlace inalámbricos
  - > red de area local
- ▶ Los paquetes de nivel de enlace se llaman *tramas* (*frames*)
- ▶ El nivel de enlace entrega tramas a nodos vecinos a través de un enlace



# Nivel de enlace

- ▶ Diferentes saltos con diferentes niveles de enlace
  - > Red de área local Ethernet
  - > Enlaces ATM entre routers
  - > Red de área local 802.11 inalámbrica
- ▶ Los diferentes niveles ofrecen diferentes funciones
  - > Entrega garantizada o no
  - > Probabilidad de errores
  - > ...



# Funciones del nivel de enlace

- ▶ **Encapsulado de nivel de enlace**

- > Construir tramas con los datos, añadiendo cabeceras

- ▶ **Acceso al canal si es un medio compartido**

## Medium Access Control (MAC)

- > Se usan direcciones de nivel de enlace (MAC) para identificar origen y destino

- ▶ **Entrega fiable entre vecinos**

- > técnicas igual que las del nivel de transporte
- > en medios con pocas pérdidas (fibra, par trenzado...) no se suelen usar  
Basta con detectar errores para no entregar
- > en redes inalámbricas de altas pérdidas si que es necesario
  - + ¿Porqué recuperar errores a nivel de enlace y extremo a extremo?

# Más funciones del nivel de enlace

## ▶ Control de flujo

- > Para no saturar el buffer de recepción del vecino

## ▶ Detección de errores

- > Causados por atenuación o ruido
- > el receptor señala al emisor o bien descarta la trama

## ▶ Corrección de errores

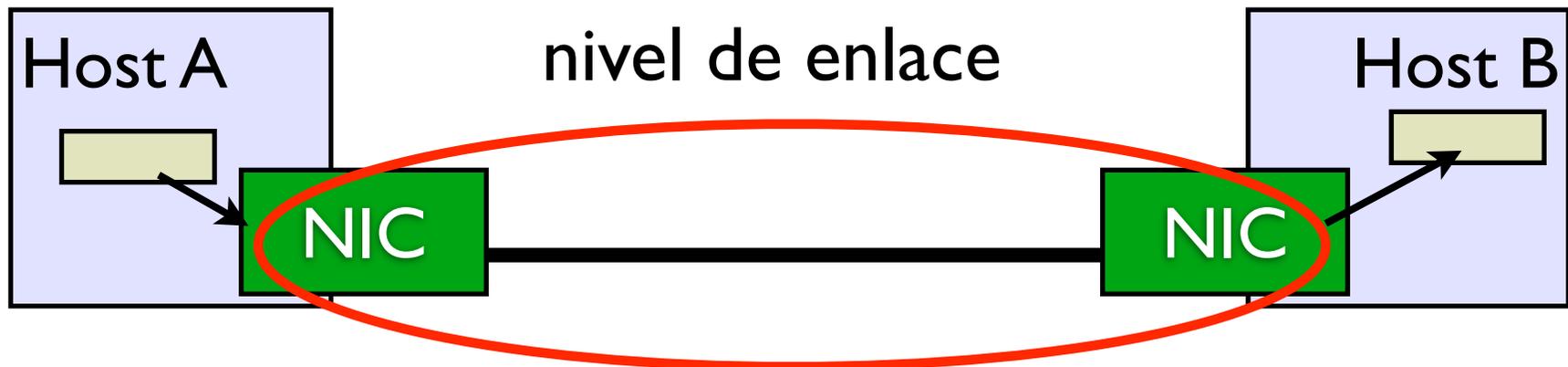
- > el receptor es capaz de detectar y corregir errores sin retransmisiones

## ▶ Half-duplex o full-duplex

- > Half duplex: se puede transmitir en los dos sentidos pero no al mismo tiempo
- > Full duplex: se puede transmitir en los dos sentidos al mismo tiempo

# Implementacion nivel de enlace

- ▶ Normalmente el nivel de enlace esta implementado en hardware en el adaptador (NIC, network interface card)



- > Emisor: encapsula la trama, añade bits de control de errores, flujo...
- > Receptor: detecta errores y extrae los datos, la CPU puede leer los datos si son correctos
- ▶ El adaptador es bastante autónomo
- ▶ Nivel físico y de enlace en una tarjeta

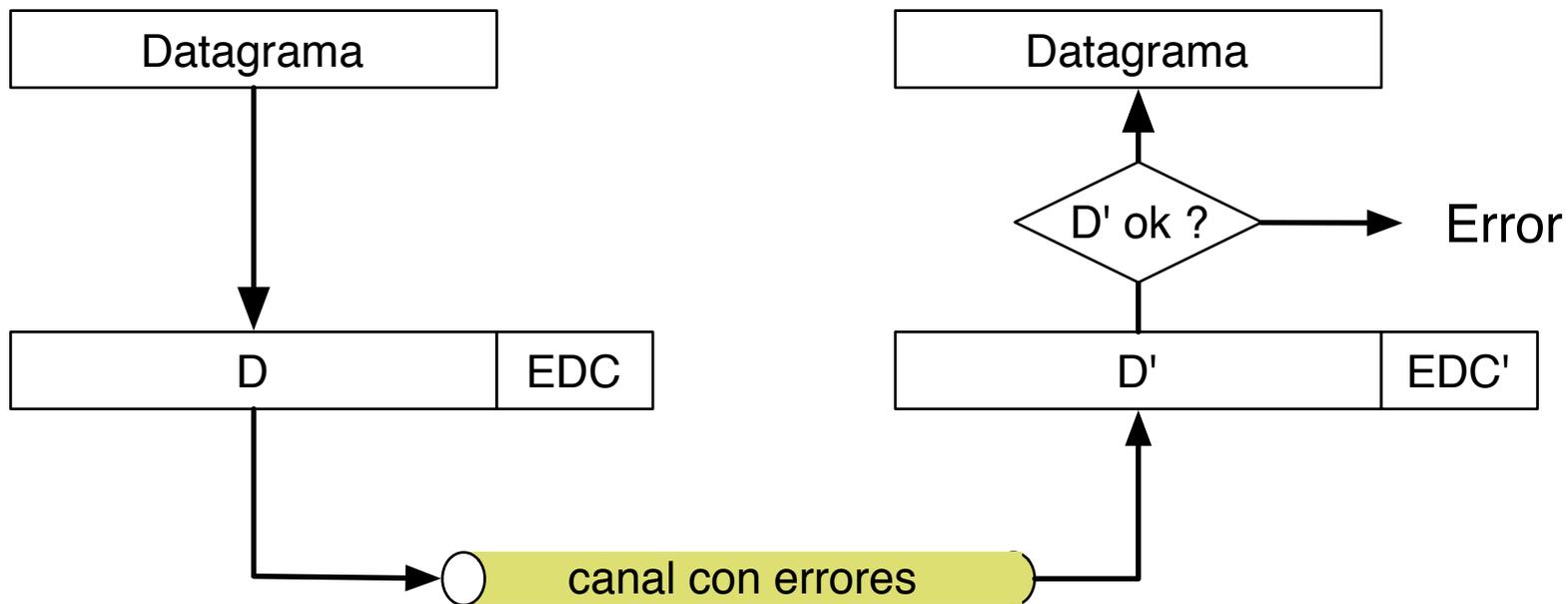
# Detección y corrección de errores

- ▶ D datos

EDC error detection and correction (redundantes)

no pueden detectarse el 100% de los errores pero se puede hacer que la probabilidad de no detectarlos sea pequeña

menos probabilidad requiere más bits redundantes



# Paridad

- ▶ Añadir bits para que el numero de unos sea par

- ▶ Ejemplo bit de paridad añadido a una palabra permite detectar errores de 1 bit

0111000110101011	1
------------------	---

- ▶ Ejemplo bits de paridad en una matriz permiten corregir errores de 1 bit

10101		1	10101		1
11110		0	<del>10110</del>		<del>0</del>
01110		1	01110		1
<hr/>			<hr/>		
00101		0	00101		0

- ▶ Detectar y corregir errores Forward Error Correction (FEC)

# Checksum

## ▶ Internet checksum

- > Tratar los datos como una serie de palabra (de 16 bits)
- > La suma en complemento a 1 se añade al paquete
- > Al recibir el paquete la suma de todo tiene que salir 0xFFFF
- > Ventajas: poco overhead, facil de hacer en software
- > Desventajas: no protege mucho

## ▶ Cyclic redundancy check (CRC)

# Cyclic redundancy check (CRC)

- ▶ D bits de datos como polinomio coefs 0 o 1  
R bits redundantes  
se elige un polinomio G generador de  $r+1$  bits
- ▶ Si elegimos R de forma que DR sea divisible por G (modulo 2)
  - > Si en recepcion al dividir DR por G no obtenemos 0 Error
  - > Este método detecta
    - + ráfagas de errores de menos de  $r+1$  bits
    - + ráfagas de errores de mas de  $r+1$  bits con probabilidad  $1-0.5^r$
    - + todos los errores en número impar
- ▶ Con la misma redundancia que el anterior CRC detecta muchos más errores
  - > CRC se suele usar mucho a nivel de enlace
  - > Entonces porqué IP,TCP y UDP usan el anterior???

# Acceso al medio

## Dos tipos de medio

### ▶ Punto a punto

- > ppp para acceso telefónico
- > conexión ethernet PC - switch
- > No tienen problemas de acceso compartido, lo que envía uno lo recibe el otro

### ▶ Broadcast

- > Ethernet
- > Redes de cable HFC (subida)
- > Redes inalámbricas 802.11
- > **Como arbitrar que varios dispositivos se comuniquen usando el mismo medio???**

# Control de Acceso al Medio (MAC)

- ▶ En un canal de broadcast de capacidad Rbps
- ▶ El control de acceso al medio ideal debería cumplir
  - > Si hay un solo dispositivo que quiere transmitir puede transmitir a Rbps
  - > Si hay N dispositivos que quieren transmitir cada uno de ellos puede transmitir a  $R/N$  bps
  - > Totalmente **descentralizado**:
    - + no es necesario un dispositivo especial para coordinar o repartir los recursos
    - + no es necesaria sincronización entre los dispositivos, no slots
  - > Lo más simple posible

# Acceso al medio clasificación

## Tres clases

### ▶ Partición de canal

- > dividir el canal en slots temporales TDM o de frecuencia FDM (o incluso código CDMA)
- > asignar los slots a diferentes dispositivos

### ▶ Acceso aleatorio

- > el canal no se reparte, los dispositivos pueden colisionar si transmiten a la vez
- > métodos de recuperación de colisiones

### ▶ Turnos

- > los dispositivos se pasan el turno

# Acceso aleatorio: protocolos

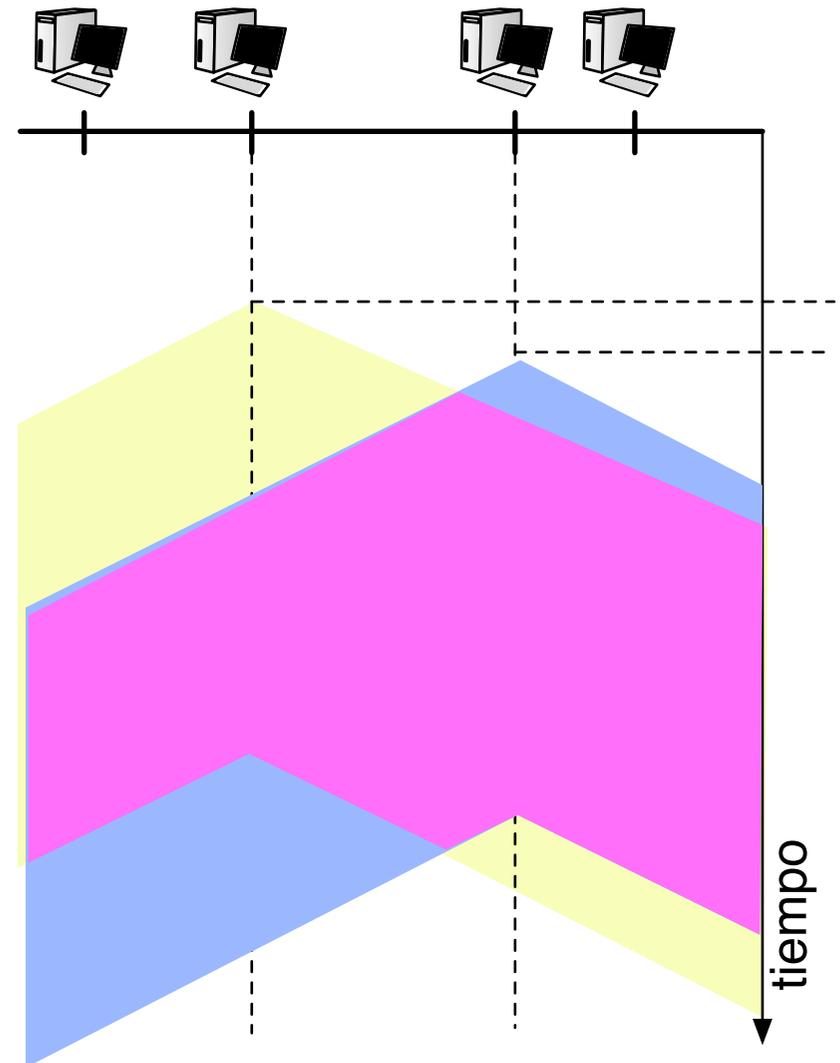
- ▶ Cuando un dispositivo tiene una trama a enviar
  - > envía la trama a velocidad  $R$  ocupando el canal
  - > sin coordinación con otros dispositivos
- ▶ Si más de un dispositivo transmite: **colisión**
- ▶ Los protocolos MAC (medium access control):
  - > cómo detectar colisiones
  - > cómo reaccionar a las colisiones
- ▶ Protocolos MAC típicos:
  - > ALOHA
  - > ALOHA ranurado
  - > CSMA, CSMA/CD, CSMA/CA

# Protocolos tipo CSMA

- ▶ CSMA: carrier sense multiple access
- ▶ Intuitivamente: no intrumpir al que habla
  
- ▶ Escucha antes de transmitir
- ▶ Si el canal esa libre: transmite
- ▶ Si el canal está ocupado: espera

# CSMA

- ▶ Aun así ocurren colisiones
  - > el tiempo de propagación hace que no vea inmediatamente si otro está transmitiendo
  - > al finalizar una transmisión puede haber más de un dispositivo esperando
- ▶ El tiempo de transmisión de la trama se desperdicia

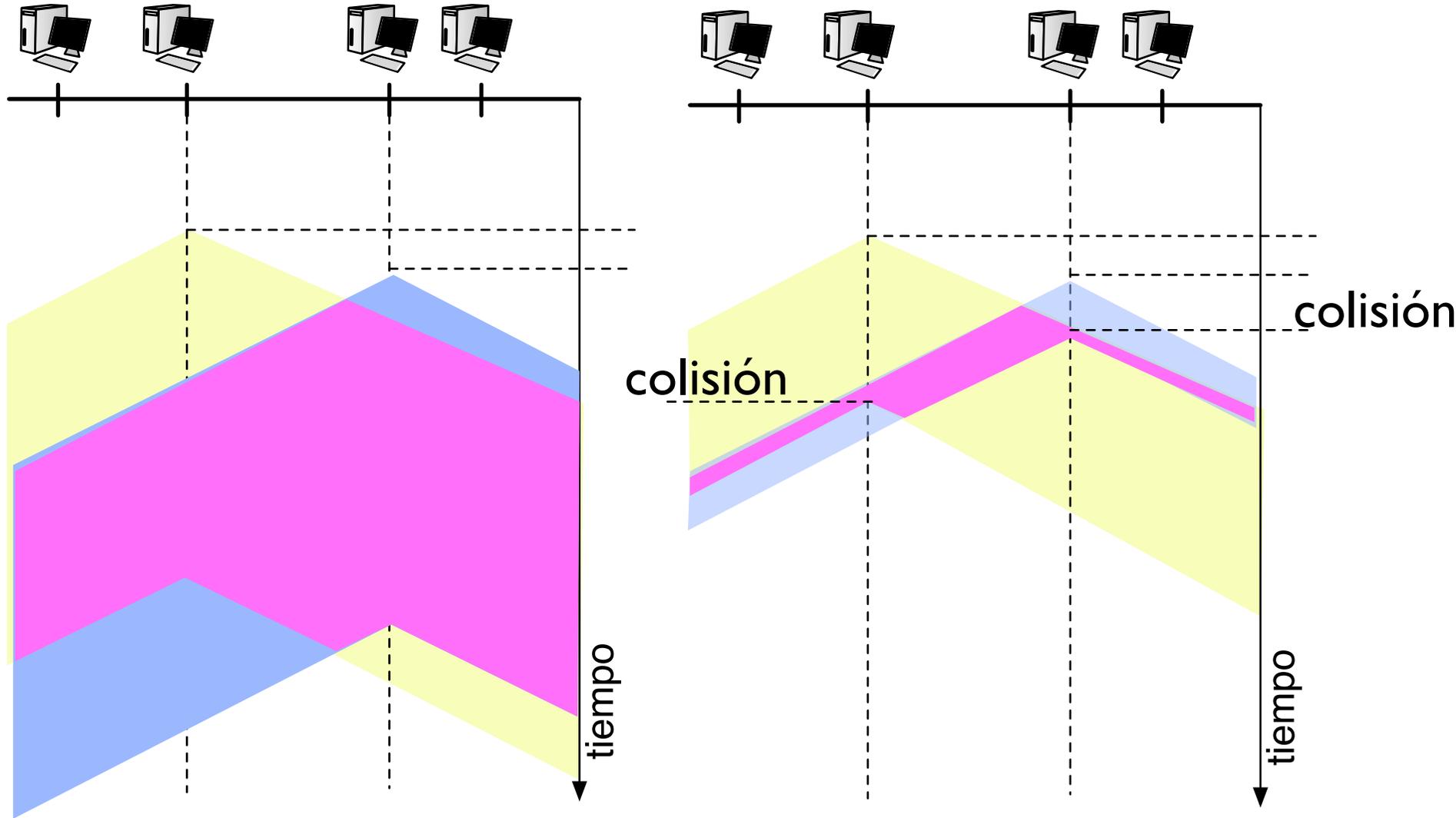


# CSMA/CD

- ▶ **CSMA+CD (collision detection)**
  - > la colisión llega al emisor poco tiempo después
  - > si se detecta colisión dejar de transmitir inmediatamente
  - > esperar un tiempo aleatorio antes de volver a transmitir (backoff)
- ▶ **Menos capacidad desperdiciada por colisión**
- ▶ **Detección de colisiones**
  - > Es fácil en redes de cable (comparar señal recibida y transmitida)
  - > Difícil en redes inalámbricas

# CSMA/CD

- ▶ Cada dispositivo para al detectar la colisión



# Conclusiones

- ▶ El nivel de enlace normalmente realiza las funciones de encapsulado, detección de errores y control de acceso al medio en medios compartidos
- ▶ Medios de interés
  - > Punto a punto
  - > Compartidos (redes de área local)
    - + Cable e Inalámbricos
    - + Hemos visto las técnicas de compartición pero...

## Próxima clase

- ▶ Como se dirigen las tramas? direccionamiento