

# **Redes de Computadores**

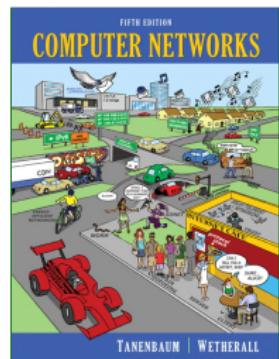
## **Tema 1 – Introducción y Arquitecturas de Red**

**Natalia Ayuso, Juan Segarra y Jesús Alastruey**



Departamento de  
Informática e Ingeniería  
de Sistemas  
**Universidad Zaragoza**

1. Introducción
  - 1.1. Evolución histórica
2. Terminología
  - 2.1. Topología de red
  - 2.2. Tipos de envío, por destino
  - 2.3. Unidades y prefijos
3. Estándares
4. Arquitectura de red
  - 4.1. Modelo OSI
  - 4.2. Modelo TCP/IP
  - 4.3. Comparativa OSI-TCP/IP
  - 4.4. Encapsulado de protocolos
  - 4.5. Retransmisores



Capítulo 1

# 1 Introducción



Paloma mensajera

Líneas telégrafo

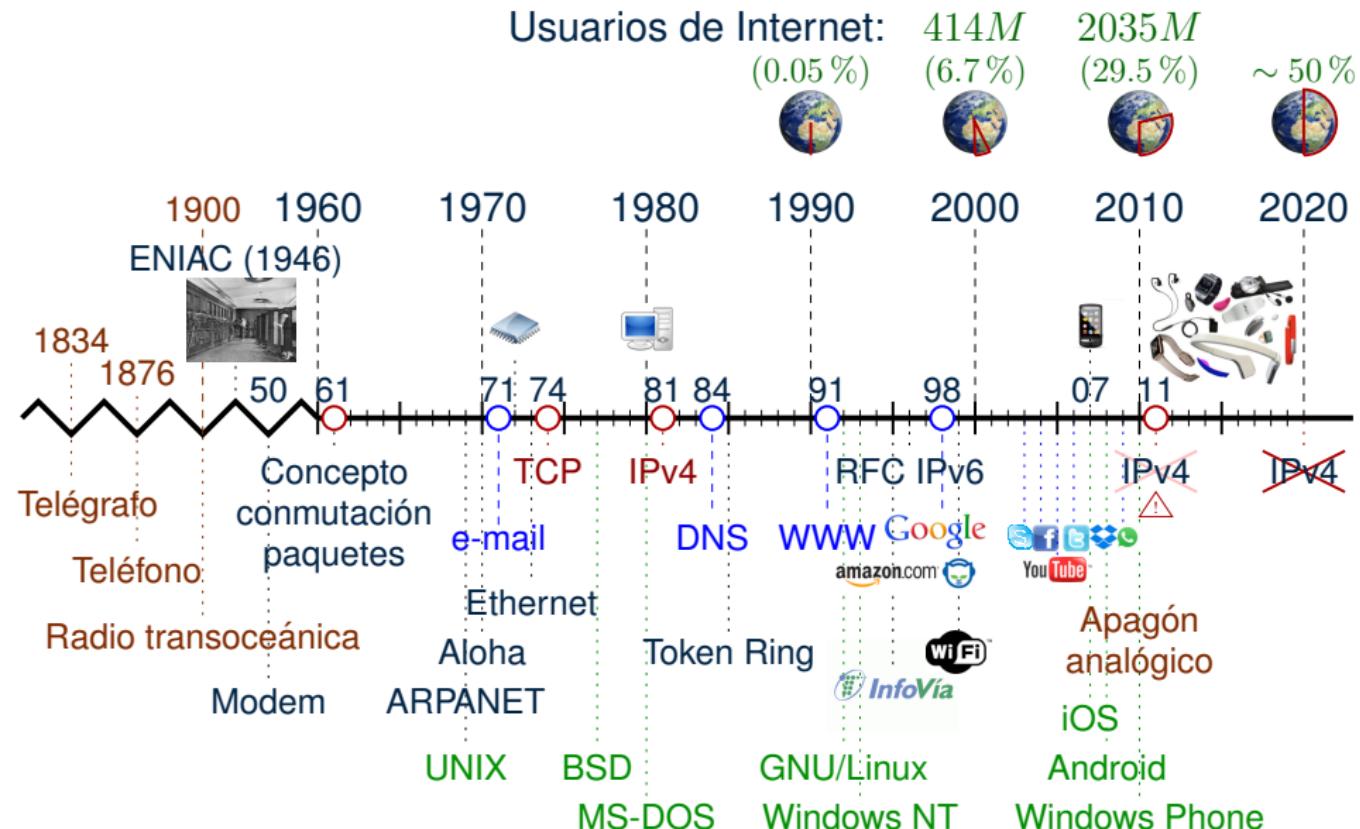
Ordenador conectado

- Las **redes de comunicaciones** permiten superar limitaciones geográficas «rápido»
  - Servicios: reservas, venta, banca, vídeo bajo demanda, etc.
- Las **redes de computadores**, al prescindir de emisor/receptor humanos, además permiten:
  - Información voluminosa → más servicios
  - Supercomputación (cluster/grid)
  - Independencia del equipo de trabajo (nube/cloud)
  - Trabajo en grupo / redes sociales
  - etc.

## 1.1 Evolución histórica

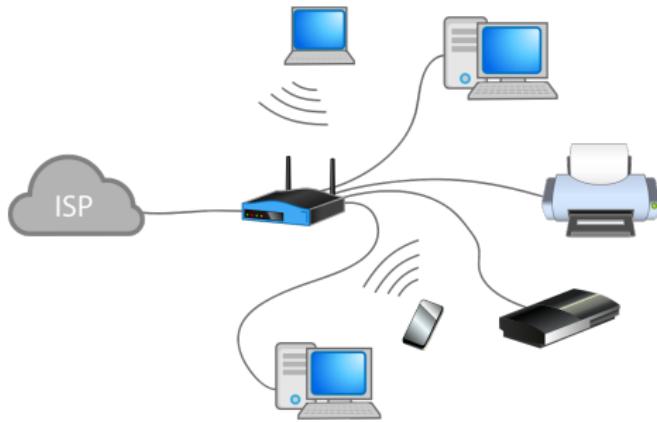


474



## 2 Terminología

**Red de computadores:** sistema de comunicación que permite intercambiar información entre dos o más equipos informáticos



**Nodo:** dispositivo físico o virtual que pueden enviar, recibir o reexpedir información sobre una red

## 2 Terminología (II)

---



1474

**Dispositivo de interconexión:** nodo que retransmite la información recibida, eg. *router*

**Estación/host/end-point:** nodo que hospeda aplicaciones/servicios

**Cliente:** nodo que solicita servicios

**Servidor:** nodo que proporciona servicios

**Peer (par):** cliente + servidor

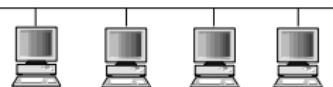
**Enlace:** conexión física o lógica entre dispositivos

**Punto-a-punto:** conexión física directa entre dos o más dispositivos, e.g. cable, aire

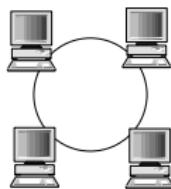
**Extremo-a-extremo:** conexión lógica entre dos dispositivos, normalmente a través de dispositivos de interconexión

## 2.1 Topología de red

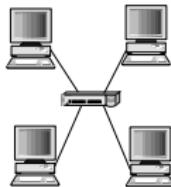
Topología de red: disposición en que se encuentran los nodos de la red. Ejemplos:



Bus



Anillo



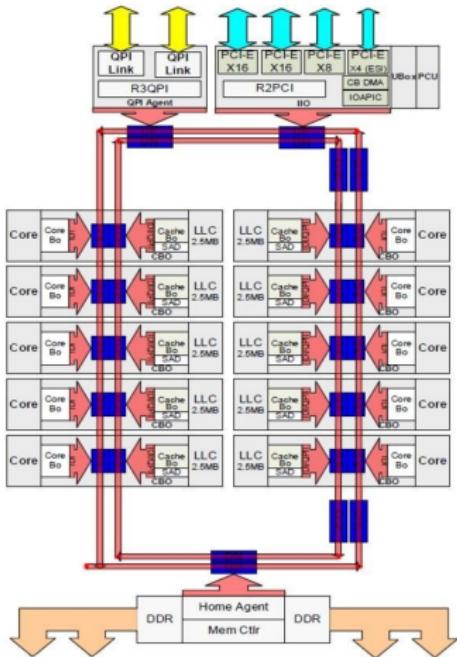
Estrella/Árbol



Topología de RedIRIS

## 2.1 Topología de red: anillo

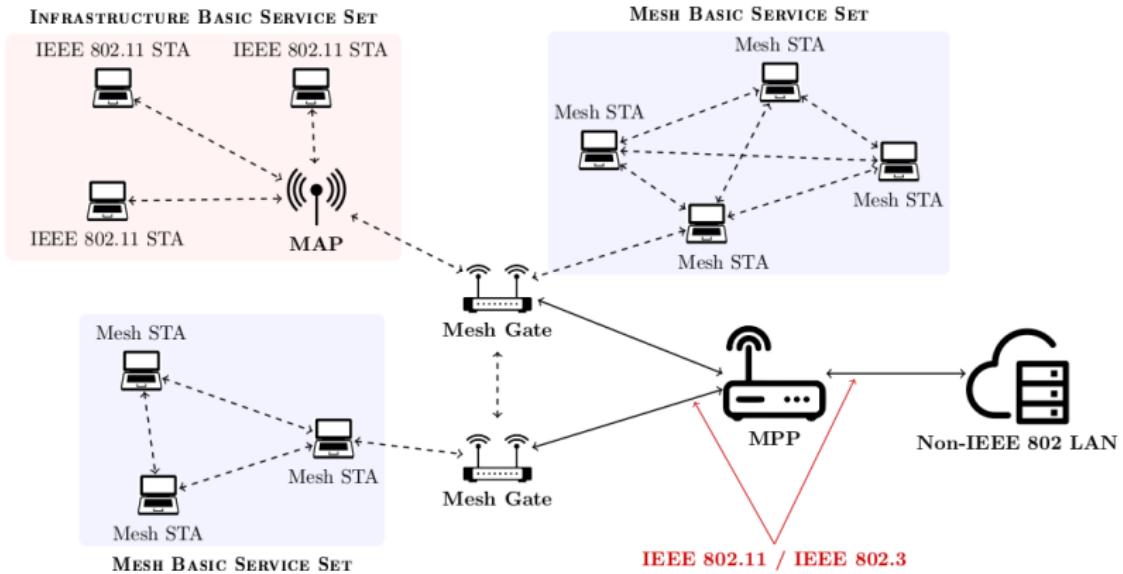
► Ejemplo de red en anillo: Intel Broadwell



Fuente: <https://www.tomshardware.co.uk/intel-mesh-architecture-skylake-x-hedt,news-56015.html>

## 2.1 Topología de red: malla

### ► Ejemplo de red en malla Wi-Fi



Fuente: <https://www.mdpi.com/1999-5903/11/4/99>

## 2.2 Tipos de envío, por destino

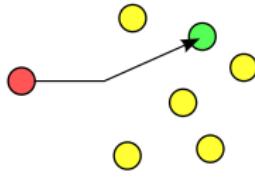
Dependiendo del destino, un mensaje puede ser:

**Unicast:** un único destino

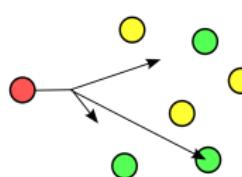
**Anycast:** un destino cualquiera (el más cercano) de un conjunto

**Multicast/Multidestino:** un conjunto de destinos

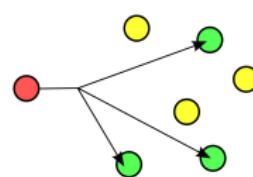
**Broadcast/Difusión:** todos los destinos



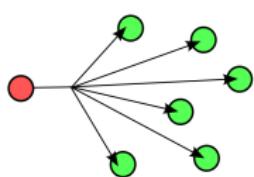
Unicast



Anycast



Multicast



Broadcast

No todos los tipos de envío son siempre posibles

## 2.3 Unidades y prefijos



Bit (bit o b): binary digit (1/0)

Byte (B): vector de 8 bits

Prefijo decimal (SI)				P. binario (ISO/IEC)	
Valor	Prefijo	Valor	Prefijo	Valor	Prefijo
$10^{-3}$	mili (m)	$10^3$	kilo (k)	$2^{10}$	kibi (Ki)
$10^{-6}$	micro ( $\mu$ )	$10^6$	mega (M)	$2^{20}$	mebi (Mi)
$10^{-9}$	nano (n)	$10^9$	giga (G)	$2^{30}$	gibi (Gi)
$10^{-12}$	pico (p)	$10^{12}$	tera (T)	$2^{40}$	tebi (Ti)
$10^{-15}$	femto (f)	$10^{15}$	peta (P)	$2^{50}$	pebi (Pi)
$10^{-18}$	atto (a)	$10^{18}$	exa (E)	$2^{60}$	exbi (Ei)
$10^{-21}$	zepto (z)	$10^{21}$	zetta (Z)	$2^{70}$	zebi (Zi)
$10^{-24}$	yocto (y)	$10^{24}$	yotta (Y)	$2^{80}$	yobi (Yi)

### 3 Estándares

---



1474

- Al principio cada fabricante tenía especificaciones propias
  - E.g. SNA (IBM), IPX/SPX (Novell), Appletalk (Apple)
  - Problema: interoperatividad limitada entre fabricantes
- Solución: establecer especificaciones públicas entre todos, aprobadas por organismos internacionales, que todos puedan seguir
- ¿Por qué siguen existiendo especificaciones privadas?
  - Forzar la compra de productos del mismo fabricante (*SMB*)
  - Retener a los usuarios para vender la información que generan (*Whatsapp*)
  - Coste de salida elevado: cambiar todos los equipos a la vez, aislarlos de los contactos

### 3 Estándares (II)

---



Principales organizaciones de estándares:

**ISOC**: *Internet Society*

**IAB**: *Internet Architecture Board*

**IETF**: *Internet Engineering Task Force*

► Request For Comments (RFCs)

**IRTF**: *Internet Research Task Force*

**IESG**: *Internet Engineering Steering Group*

**ANSI**: *American National Standards Institute*

**IEEE**: *Institute of Electrical and Electronics Engineers*

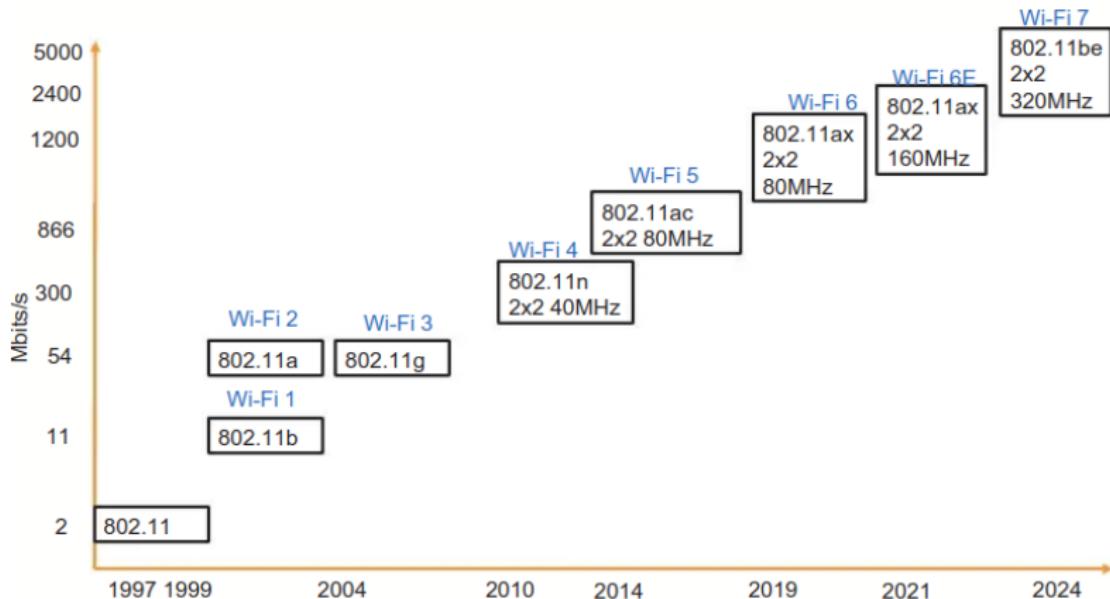
**ISO**: *International Organization for Standardization*

**ITU-T**: *International Telecommunication Union -  
Telecommunications Sector*

**W3C**: *World Wide Web Consortium*

### 3 Estándares (III)

#### Estándares Wi-Fi a lo largo de los años



Fuente: Anil Kumar, Jafer Hussain y Anthony Chun. Connecting the Internet of Things: IoT Connectivity Standards and Solutions. Apress, 2023.

# 4 Arquitectura de red

---

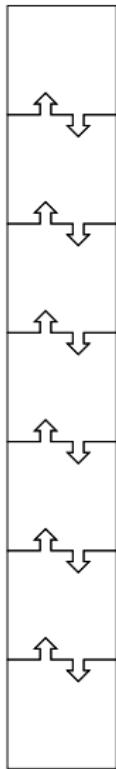


1474

La **arquitectura de red** es el patrón común al que han de ceñirse los elementos de red para mantener compatibilidad entre sí

- En la interconexión de computadores intervienen muchos elementos hardware/software desarrollados por distintos fabricantes
- Especificar detalladamente todo el problema de forma conjunta no es factible
  - Un «firefox» específico para cada tipo de tarjeta de red
- Mejor dividir el problema mediante un **modelo de capas**
  - Permite describir el funcionamiento de las redes de forma modular y hacer cambios de manera sencilla
  - Modelo de referencia: *Open System Interconnection (OSI)* de ISO

# 4 Arquitectura de red (II)



El modelo de capas se basa en:

- Cada capa resuelve un problema concreto
- Un mismo problema puede resolverse de distintas formas, detalladas en protocolos
  - Una misma capa puede albergar varios protocolos
  - Dos capas en sistemas distintos se pueden comunicar si usan el mismo protocolo
- Cada capa/protocolo tiene una interfaz de comunicación con sus capas superior e inferior (e.g. *API socket*)
- El conjunto de protocolos (uno por capa) usados en una comunicación concreta se conoce como **pila de protocolos**

## 4.1 Modelo OSI



Capa	Descripción
7. Aplicación	Protocolos específicos para aplicaciones
6. Presentación	Conversión de datos al formato requerido por la aplicación
5. Sesión	Control/coordinación de comunicaciones
4. Transporte	Comunicación extremo-a-extremo ( <i>end-to-end</i> )
3. Red	Búsqueda de caminos + llevar mensajes a destino
2. Enlace de datos	Control de acceso al medio de transmisión (MAC) + comunicación punto-a-punto
1. Física	Especifica parámetros mecánicos/eléctricos/funcionales de uso del medio de transmisión

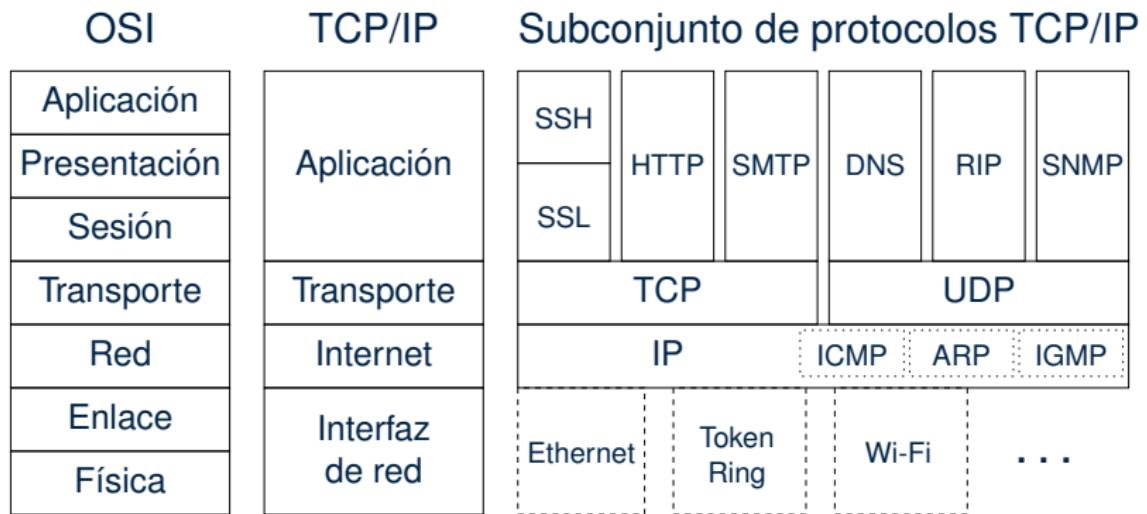
## 4.2 Modelo TCP/IP

---



- Anterior al modelo OSI
- Diseñado por el departamento de defensa de los EE.UU. (ARPANET)
- Objetivo: proporcionar comunicaciones con tolerancia a fallos (comutación de paquetes)
- Diseñado sin las perspectivas de su uso actual (Internet)
- Comunicación entre procesos
- Capa de interfaz de red aglutina las capas física y de enlace de datos
- Capa de aplicación aglutina las capas por encima de la capa de transporte

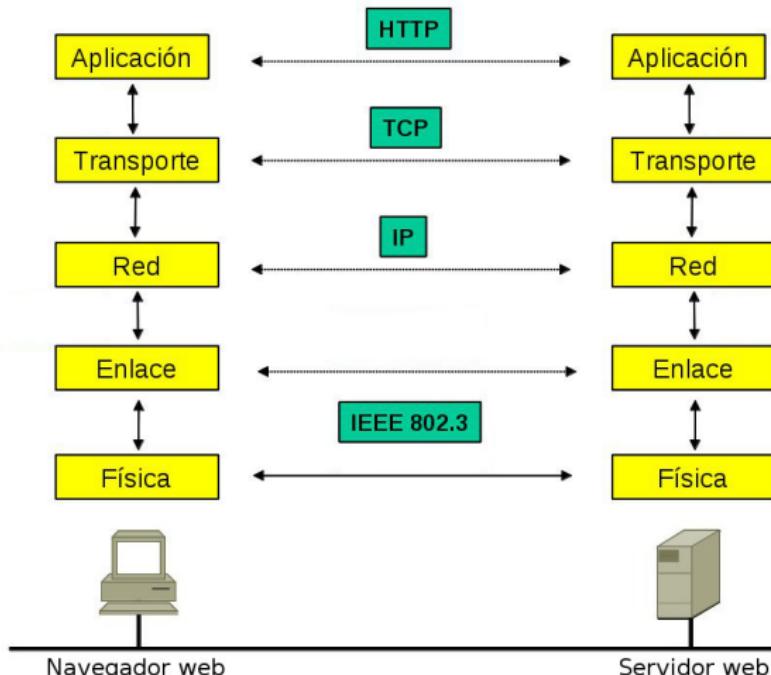
## 4.3 Comparativa OSI-TCP/IP



- A menudo se sigue un modelo híbrido entre ambos

## 4.3 Comparativa OSI-TCP/IP (II)

- E.g. acceso a un servidor web en la misma LAN

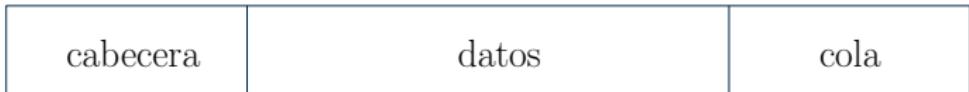


- ¿Cuál es la pila de protocolos?

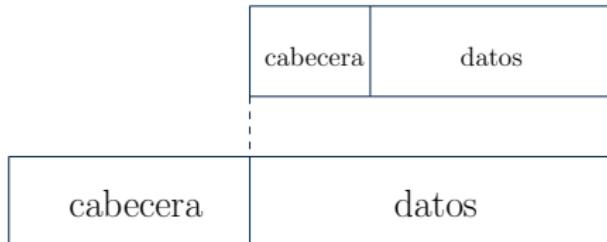
## 4.4 Encapsulado de protocolos



- **Mensaje:** unidad de información de un protocolo. Consta de información de control (**cabecera/header**) y datos (**cuerpo/payload**). A veces incluye **cola/footer/trailer**)



- **Encapsulado:** el mensaje de una capa es el *payload* de la capa inferior



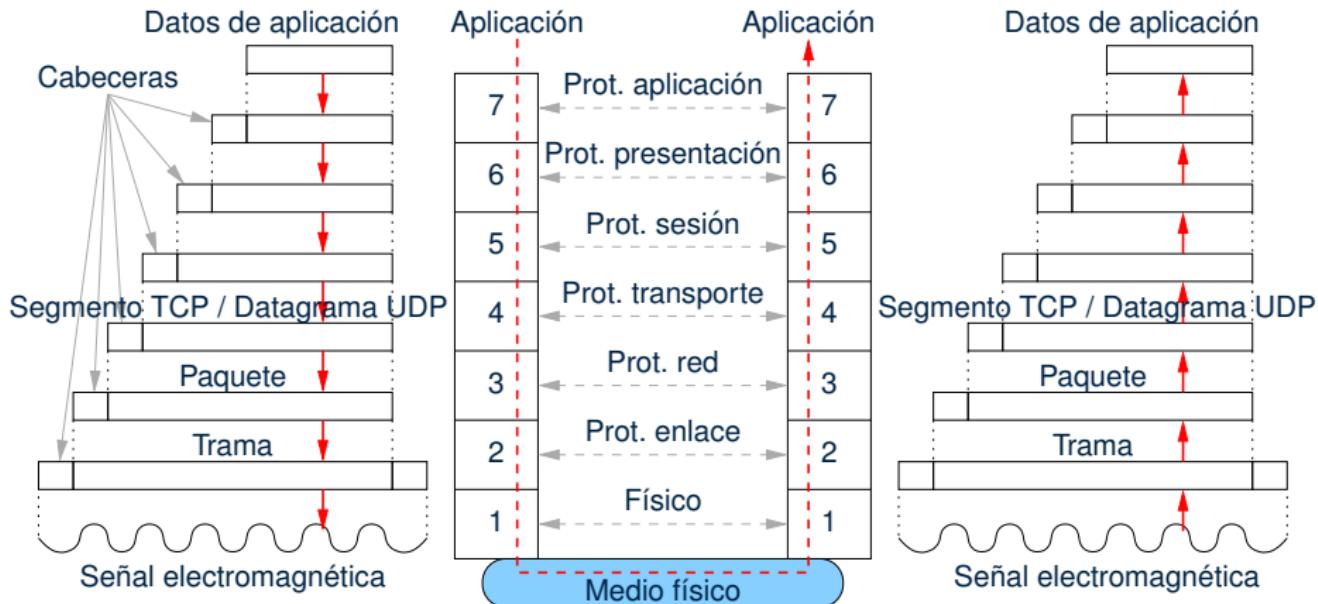
## 4.4 Encapsulado de protocolos (II)



- Encapsulado en pila de protocolos

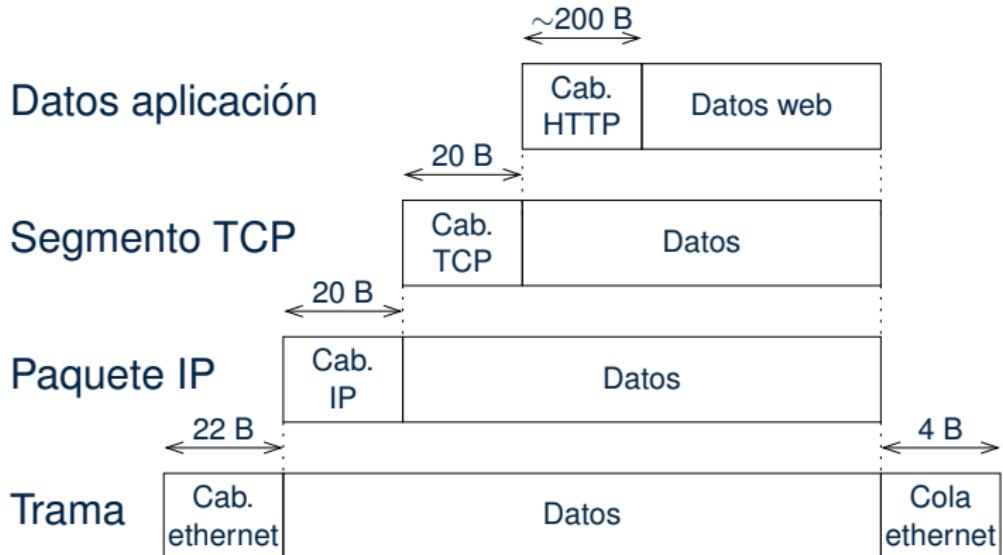


## 4.4 Encapsulado de protocolos (III)



## 4.4 Encapsulado de protocolos (IV)

- La información de control añadida reduce la eficiencia.



Eficiencia: datos/datos totales

Sobrecarga (overhead): datos de control/datos totales  
(=1-Eficiencia)

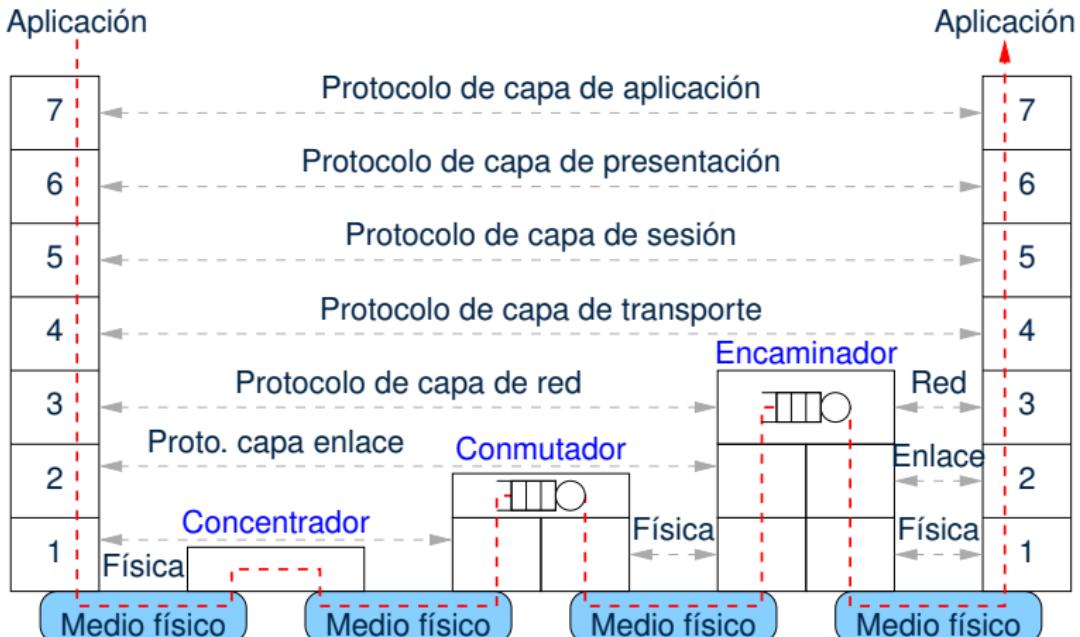
## 4.4 Encapsulado de protocolos (V)



1. Calcula la sobrecarga en el caso de que se envíe 1 byte de datos en el *payload* de la capa de aplicación.
2. Si la velocidad de transmisión es  $v_t = 100 \text{ Mbps}$ , calcula la velocidad efectiva de transmisión de datos de aplicación.
3. Repetir los dos apartados anteriores para el caso de que se envíen 1000 bytes de datos en el *payload* de la capa de aplicación.

## 4.5 Retransmisores

- Física: amplificador, repetidor, concentrador (*hub*) ¡transp.!
- Enlace: conmutador (*switch*), puente (*bridge*) ¡transparente!
- Red: encaminador (*router*)



## 4.5 Retransmisores (II)

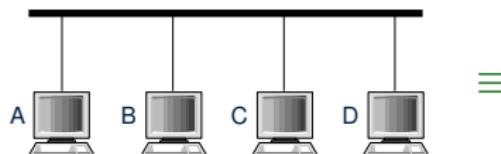
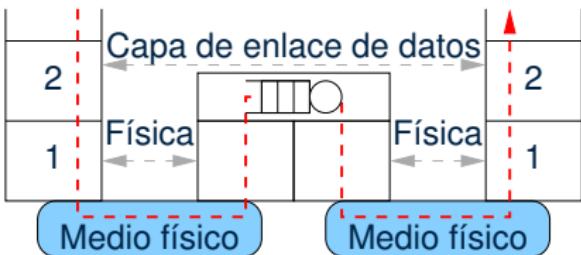
---

- Capa física: repetidor
- Función principal: extender rango
- ¡Transparente!: los equipos no saben si hay repetidor o no

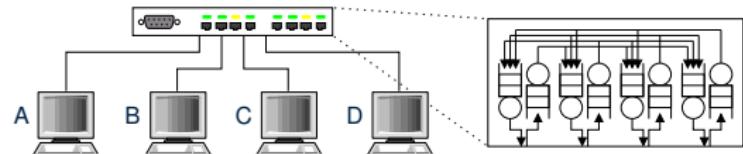


## 4.5 Retransmisores (III)

- Capa de enlace: conmutador (*switch*), puente (*bridge*)
- Función principal: reducir colisiones en una red
- ¡Transparente!: los equipos no saben si hay conmutador o no

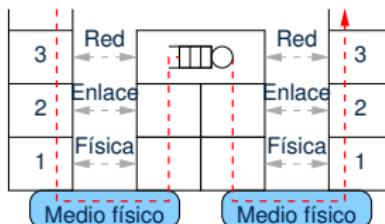


≡



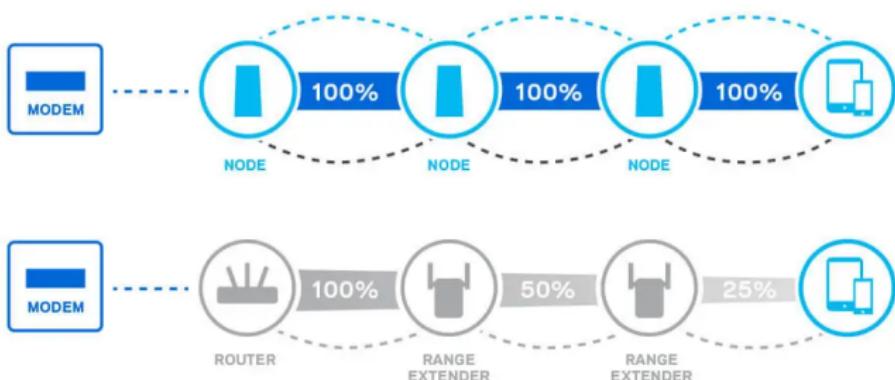
## 4.5 Retransmisores (IV)

- Capa de red: encaminador (*router*)
- Función principal: interconectar redes
- Capa de enlace: control de acceso al medio
- Capa física: MODEM: MO-dulator DEM-modulator
- ¡No transparente!: los equipos conocen a su encaminador e interactúan explícitamente con él



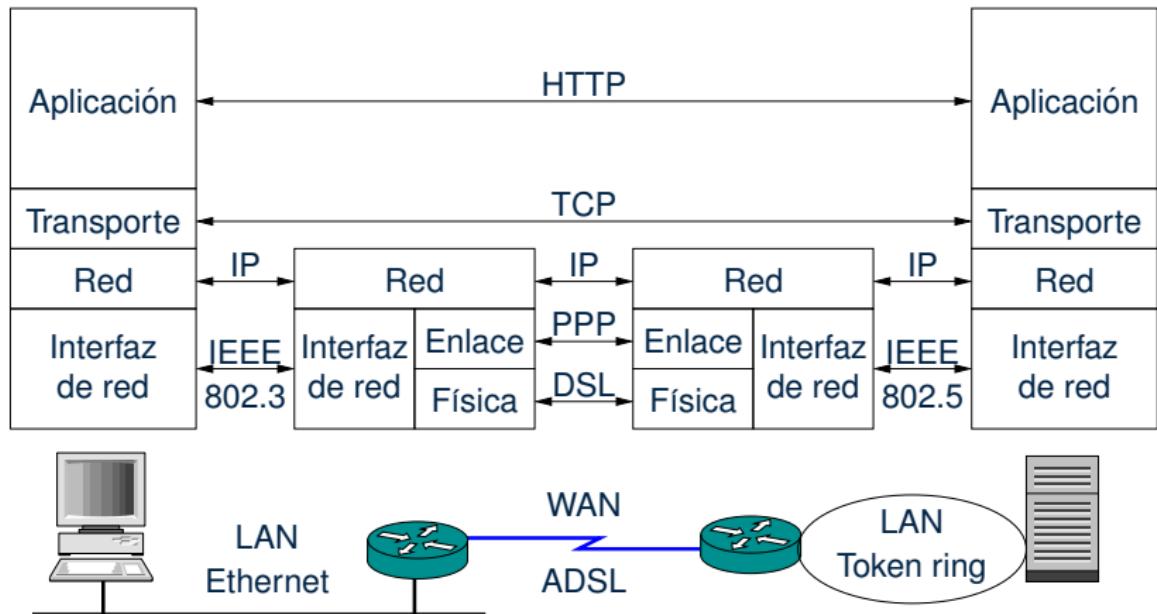
## 4.5 Retransmisores (V)

### Mesh vs. Range Extender



## 4.5 Retransmisores (VI)

- E.g. acceso web atravesando tres redes físicas



- ¿Cuál es la pila de protocolos?
- ¿Y con un conmutador entre el cliente y su encaminador?

# **Redes de Computadores**

## **Tema 2 – Capa Física**

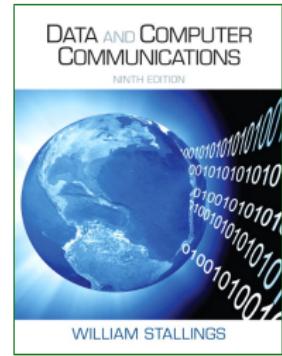
**Natalia Ayuso, Juan Segarra y Jesús Alastruey**



Departamento de  
Informática e Ingeniería  
de Sistemas

**Universidad** Zaragoza

1. Introducción
2. Conceptos y terminología
3. Medios de transmisión
4. Transmisión digital
5. Capacidad de un canal con ruido
6. Sincronismo
7. Modos de transmisión
8. Conclusiones



Cap. 3, 4, 5 y 6

# 1 Introducción

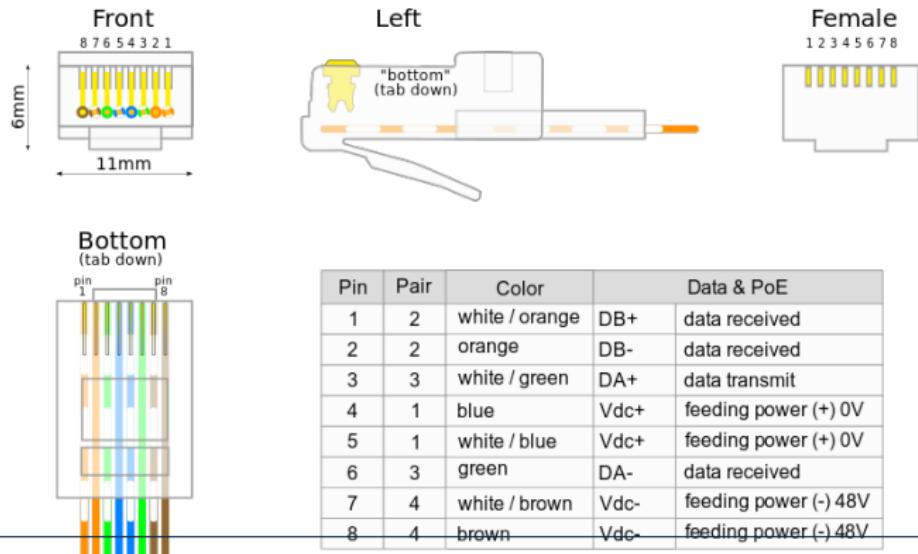
---

*According to the European Telecommunications Standards Institute (ETSI), a standard is a “document, established by consensus and approved by a recognized body, that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at achievement of the optimum degree of order in a given context”*

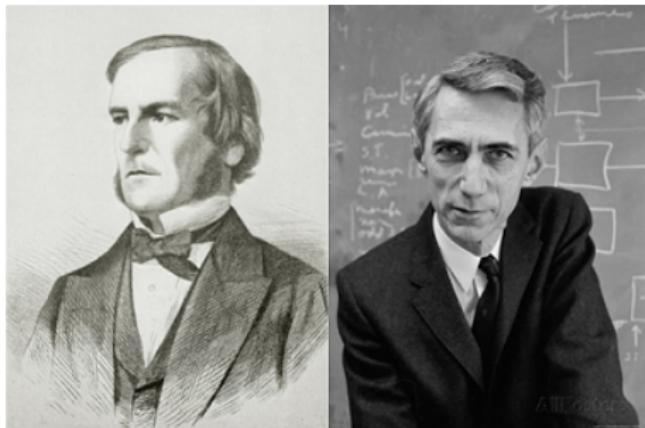
# 1 Introducción (II)

La capa física se encarga de la *interfaz física* entre las tecnologías de transmisión de la red:

- Especificaciones *mecánicas* de conectores y cables
- Especificaciones *electromagnéticas* de la señal
- Especifica cómo *emitir* los bits e *interpretar* la señal



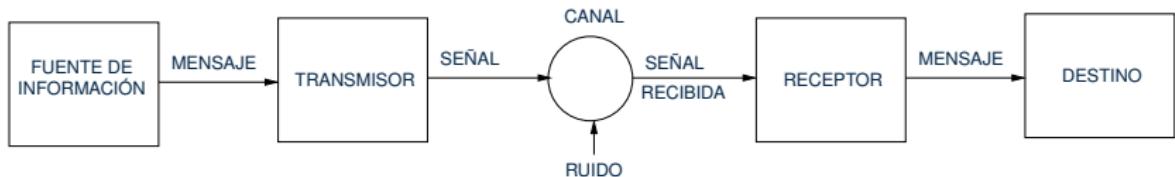
## 2 Conceptos y terminología



George Boole & Claude Shannon

## 2 Conceptos y terminología (II)

- Diagrama general de un sistema de comunicaciones:



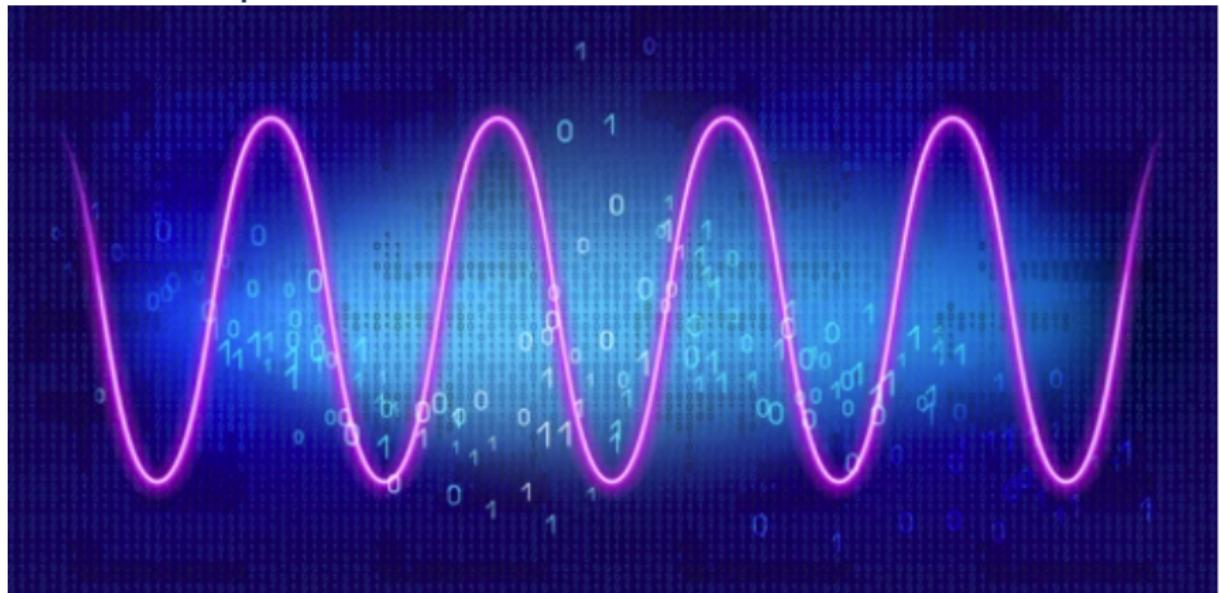
 C. E. Shannon (1948), 'A mathematical theory of communication', *The Bell System Technical Journal* (27), pp. 379–423, 623–656.

- Elementos básicos de un sistema de comunicación:

- Fuente de información: produce el mensaje
- Transmisor: transforma el mensaje para generar una señal que pueda ser enviada a través del canal
- Canal: medio por el que se envía la señal
- Receptor: transforma la señal de nuevo en el mensaje
- Destino: receptor del mensaje

## 2 Conceptos y terminología (III)

¿Cómo se transmiten los datos tanto si utilizas una conexión cableada o por el aire?



## 2 Conceptos y terminología (IV)

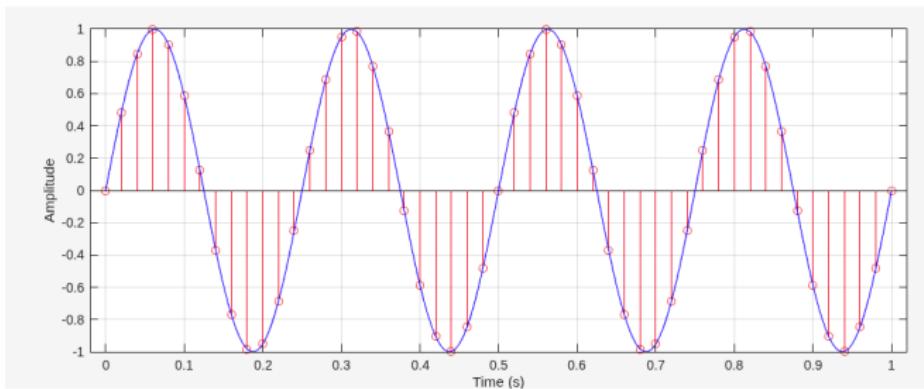


Los datos se transmiten como una *onda electromagnética* que se comporta de acuerdo a las ecuaciones de Maxwell. Las ondas electromagnéticas se componen de campos eléctricos y magnéticos que se caracterizan por:

- Frecuencia de portadora ( $f_c$ )
- Longitud de onda ( $\lambda$ )
- Velocidad de la luz.  $c = 3 \times 10^8$  m/s en el vacío
- Fase  $\theta$  en radianes
- Amplitud en voltios
- Potencia en watos (J/s)
- Energía (J) que es la potencia acumulada a lo largo del tiempo
- Ancho de banda (Hz) que es la porción del espectro que utiliza la información o datos que modula la onda

## 2 Conceptos y terminología (V)

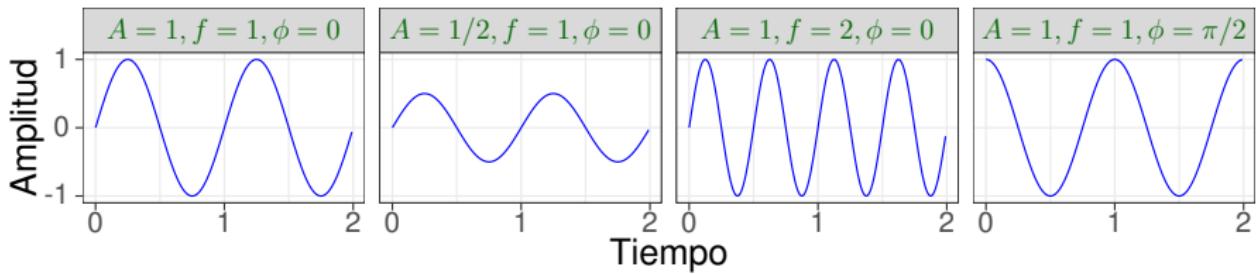
- Señal: función que transmite información
  - Continua:  $x(t)$  vs. discreta:  $x(kT)$
  - Periódica: se repite un patrón a lo largo del tiempo



## 2.1 Señales periódicas

$$s(t) = A \cdot \sin(2\pi ft + \phi)$$

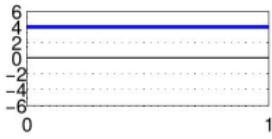
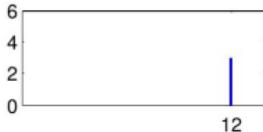
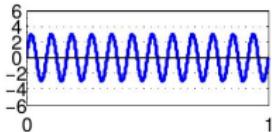
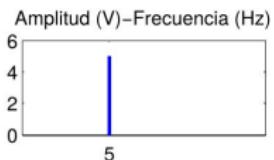
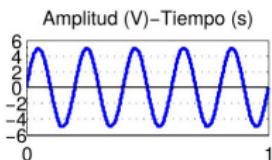
- *Amplitud de pico (A)*: valor máximo de la señal
- *Frecuencia (f)*: razón a la que se repite la señal, en Hercios (Hz) o ciclos por segundo (cps)
- *Periodo (1/f)*: tiempo transcurrido entre dos repeticiones consecutivas de la señal, en segundos (s)
- *Fase ( $\phi$ )*: posición relativa de la señal dentro de un periodo



## 2.2 Análisis de Fourier

- Toda señal periódica puede representarse como la suma de múltiples señales sinusoidales

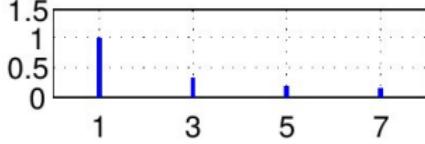
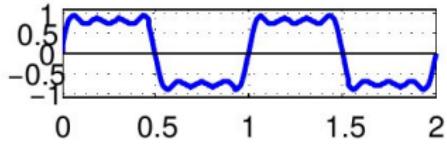
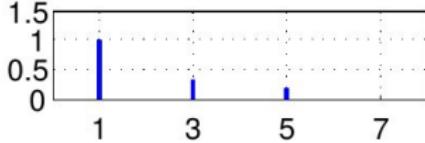
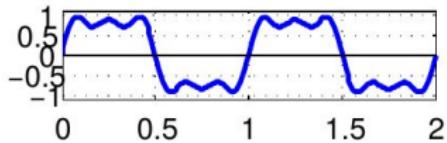
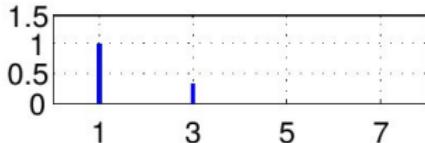
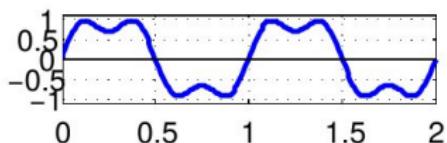
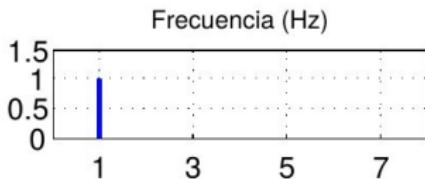
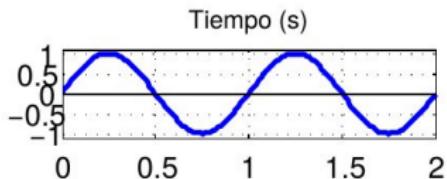
$$s(t) = \frac{1}{2}A_0 + \sum_{n=1}^{\infty} \left[ A_n \cos(2\pi n f_0 t) + B_n \sin(2\pi n f_0 t) \right]$$



- Componente continua (DC): aquella con frecuencia cero

## 2.2 Análisis de Fourier (II)

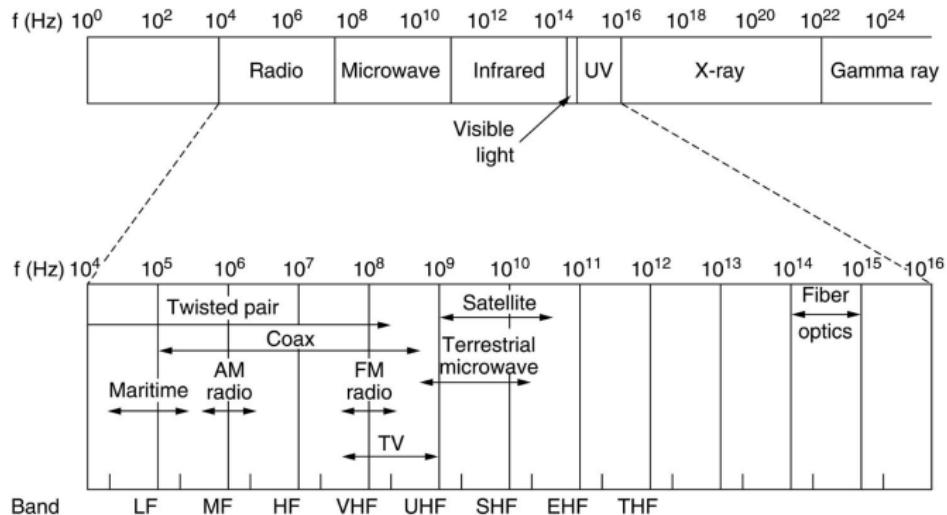
► Ejemplo: onda cuadrada



## 2.3 Espectro electromagnético



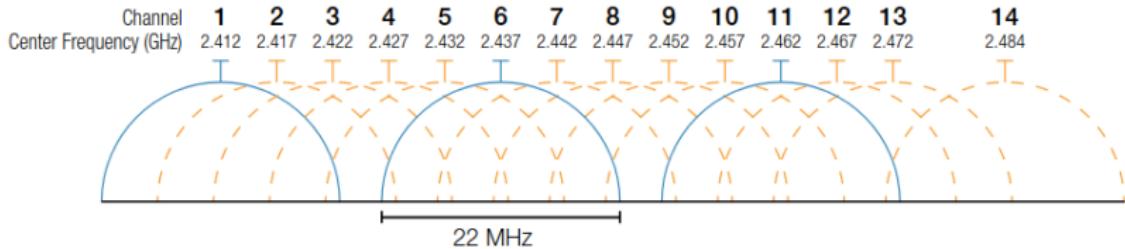
- El *espectro electromagnético* es la distribución energética del conjunto de ondas electromagnéticas



©Tanenbaum, Prentice Hall International

## 2.3 Espectro electromagnético (II)

- *Ancho de banda de frecuencias ( $B$ ): rango de frecuencias que forman la señal*
  - Voz: 100–7000 Hz →  $B = 6900 \text{ Hz}$
  - Teléfono tradicional: 300–3400 Hz →  $B = 3100 \text{ Hz}$
  - Wi-Fi (802.11g): canales de  $B = 22 \text{ MHz}$  en banda 2.4 GHz



Fuente: Tektronix. Wi-Fi: Overview of the 802.11 Physical Layer and Transmitter Measurements.

- Cualquier sistema de transmisión sólo puede transmitir por una banda limitada de frecuencias
  - por limitaciones físicas/tecnológicas
  - por legislación vigente

## 2.4 Decibelio (dB)

---



- Los *decibelios* (dB) se usan para expresar la *relación* (logarítmica) entre dos valores  $P_1$  y  $P_2$ :

$$\frac{P_2}{P_1} \text{dB} = 10 \times \log_{10} \left( \frac{P_2}{P_1} \right)$$

- Usos:
  - Ganancia/atenuación de un sistema
  - Relación entre potencia de señal y ruido
  - Expresión de potencias relativas
  - Cálculo de enlaces

## 2.4.1 Ganancia/atenuación

- Ganancia ( $G_{dB}$ ) / Atenuación ( $L_{dB}$ ) de un sistema:

$$G_{dB} = 10 \cdot \log_{10} \frac{P_s}{P_e}$$

$$L_{dB} = -10 \cdot \log_{10} \frac{P_s}{P_e} = 10 \cdot \log_{10} \frac{P_e}{P_s}$$

con  $P_s$  y  $P_e$  potencias a la salida y entrada del sistema.

- Se cumple que:

$$G_{dB} = -L_{dB}$$

## 2.4.1 Ganancia/atenuación (II)



- Ejemplo: una señal con un nivel de potencia de 10 mW se envía a través de una línea de transmisión. La potencia a la salida es 5 mW, la atenuación es:

$$L_{dB} = 10 \cdot \log_{10} \frac{P_e}{P_s} = 10 \cdot \log_{10} \frac{10 \text{ mW}}{5 \text{ mW}} = 10 \cdot 0.3 = 3 \text{ dB}$$

- El decibelio es una medida relativa, no absoluta.  
Si  $P_e = 1000 \text{ mW}$  y  $P_s = 500 \text{ mW}$ ,  $L_{dB} = 3 \text{ dB}$
- Utilidad: los cálculos de potencias cuando hay ganancias o atenuaciones se reducen a sumas y restas

## 2.4.2 Relación señal/ruido

- Relación entre potencia de señal y ruido (SNR)

$$SNR_{dB} = 10 \cdot \log_{10} \left( \frac{P_{signal}}{P_{noise}} \right) = 10 \cdot \log_{10} \left( \frac{S}{N} \right)$$

- Algunos valores:
  - Punto de acceso WiFi: SNR recomendado = 20 dB
  - Receptor GPS: SNR mínimo = 4 dB
- Ejemplo: calcula la SNR de un sistema con potencia de señal de 1000 mW y potencia de ruido de 1 mW:

$$SNR_{dB} = 10 \cdot \log_{10} \left( \frac{1000 \text{ mW}}{1 \text{ mW}} \right) = 30 \text{ dB}$$

## 2.4.3 Valores potencia relativos

- Medida de valores de potencia relativos a un valor dado:

$$P_{dBm} = 10 \cdot \log_{10} \frac{P_{mW}}{1 \text{ mW}}$$

- Ejemplo: la potencia típica de transmisión de WiFi en portátiles es 32 mW.

$$P_{dBm} = 10 \cdot \log_{10} \frac{32 \text{ mW}}{1 \text{ mW}} = 15 \text{ dBm}$$

- Ejemplo: la potencia de la señal recibida en un teléfono móvil es -100 dBm (puede verse la intensidad de señal recibida en los ajustes del teléfono).

$$P_{mW} = 10^{\frac{P_{dBm}}{10}} = 10^{\frac{-100 \text{ dBm}}{10}} = 0.1 \cdot 10^{-9} \text{ mW} = 0.1 \text{ pW}$$

## 2.4.3 Valores potencia relativos (II)



- Comparación de valores de potencia en vatios y dBm

Potencia (vatios)	Potencia (dBm)
1 W	+30 dBm
100 mW	+20 dBm
10 mW	+10 dBm
5 mW	+7 dBm
1 mW	0 dBm
500 μW	-3 dBm
100 μW	-10 dBm
10 μW	-20 dBm
1 μW	-30 dBm
100 nW	-40 dBm

- Ejemplo: redes 3G y 4G/LTE:

	EXCELENTE	BUENA	ACEPTABLE	MALA	SIN COBERTURA
3G	-70 dBm o más	De -71 a -85 dBm	De -86 a -100 dBm	De -101 a -109 dBm	-110 dBm o menos
4G / LTE	-90 dBm o más	De -91 a -105 dBm	De -106 a -110 dBm	De -111 a -119 dBm	-120 dBm o menos

## 2.4.3 Valores potencia relativos (III)



Considera un sistema con  $P_e = 4 \text{ mW}$ , una línea de transmisión con  $L_{dB} = 12 \text{ dB}$ , un amplificador con  $G_{dB} = 35 \text{ dB}$  y otra línea de transmisión con  $L_{dB} = 10 \text{ dB}$ . ¿Cuál es la potencia de salida  $P_s$ ?

## 2.4.3 Valores potencia relativos (IV)



La antena DSS-65 situada en el Complejo de Comunicaciones con el Espacio Profundo de Madrid (Madrid Deep Space Communications Complex, MDSCC<sup>i</sup>) recibe de la sonda espacial OSIRIS-REx una señal de  $7.0 \cdot 10^{-19}$  kW. Calcula la potencia de la señal recibida en dBm.

La antena DSS-63 situada en el Complejo de Comunicaciones con el Espacio Profundo de Madrid (Madrid Deep Space Communications Complex, MDSCC<sup>i</sup>) recibe de la sonda espacial Stereo A una señal de -115.8 dBm. Calcula la potencia de la señal recibida en vatios.

## 2.4.3 Valores potencia relativos (V)



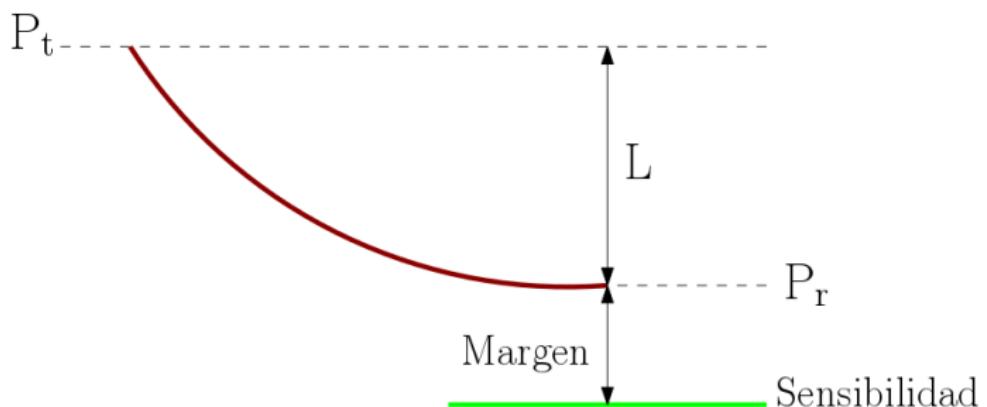
1474

✍ Una señal de 15 dBm llega a un amplificador con una ganancia de 10 dB, ¿qué potencia en dBm se tiene a su salida?

✍ Dada una señal con una atenuación de 1 dB/m, si el emisor transmite 10 dBm y el receptor requiere una señal de -20 dBm, ¿cuál es la distancia máxima entre ambos?

## 2.5 Cálculo de un enlace

- Esquema simplificado:



$$P_t - L = P_r \geq S_r + M$$

con  $P_t, P_r, S_r$  en dBm, dBW ...;  $L, M$  en dB

## 2.5 Cálculo de un enlace (II)

Un enlace LoRa tiene los siguientes parámetros:

- $P_t = 17 \text{ dBm}$
  - $S_r = -137 \text{ dBm}$
  - $f = 868.1 \text{ MHz}$  (frecuencia portadora)
- Calcula la atenuación máxima del enlace ( $L_{max}$ )
  - La atenuación  $L$  en el espacio libre viene dada por:

$$L(dB) = 32.4 + 20 \cdot \log_{10} f(MHz) + 20 \cdot \log_{10} d(km)$$

siendo  $f$  la frecuencia de la portadora en MHz y  $d$  la distancia del enlace en kilómetros. Calcula el alcance máximo del enlace ( $d_{max}$ ).

## 2.5 Cálculo de un enlace (III)

---



- *Link Budget*: cálculo de la potencia recibida en base a la potencia transmitida y todas las ganancias y pérdidas estimadas
- Atenuación en transmisión inalámbrica
  - Atenuación en el espacio libre
  - Atenuación debida a la aborcion de la atmósfera. Eg. a 3 GHz, la niebla atenúa 0.06 dB/km y la lluvia intensa 28 dB
  - Atenuación debida a obstáculos. Eg. pared de hormigón 22.792 dB a 2.4 Ghz y 44.769 dB a 5 GHz.
- Atenuación en el cable. Eg. LMR-400 21.7 - 22.2 dB/100m
  - Frecuencia
  - Longitud del cable
  - Diámetro
  - Calidad del material

## 2.5 Cálculo de un enlace (IV)

La hoja de características del módulo de Bluetooth Low Energy (BLE) de Texas Instruments CC2650MODA  indica los siguientes datos:

- $P_{TX} = 5\text{dBm}$
- $S_{RX} = -97\text{dBm}$  a 2 Mbps para una tasa de error en bit (BER de sus siglas en inglés: Bit Error Rate)  $1 \times 10^{-3}$
- $G_{TX} = G_{RX} = 1.26\text{dBi}$  ganancia de la antena integrada 1.26 dBi (i: isotropic)
- $L_{TXcable} = L_{RXcable} = 0\text{dB}$

Calcula el margen del enlace

### 3 Medios de transmisión

---



1474

Soporte a través del cual se propaga la señal transmitida

- Guiados o líneas de transmisión:

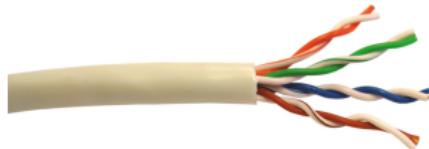
- Par trenzado
- Cable coaxial
- Fibra óptica

- No guiados (inalámbrico):

- Radiofrecuencia
- Infrarrojos (IrDA)
- Láser in Free-Space Optical

### 3.1 Par trenzado

- Dos conductores: señal + referencia
- Cada cable está compuesto por una serie de pares trenzados (4, 25, 50, 100, 200 y 300)
- Se trenzan para reducir las interferencias externas y entre pares adyacentes
- Transmiten señales moduladas y con codificación de pulsos en banda base
- El ancho de banda depende de la sección y longitud
- Bajo coste y facilidad de instalación



### 3.1 Par trenzado: tipos

---



- Tipos de aislamiento:
  - U (*unshielded*): no apantallado
  - F (*Foiled*): lámina de aluminio
  - S (*Shielded*: cable completo apantallado con malla
- Designación ISO/IEC 11801 según el aislamiento global (X) y de cada par (Y): X/YTP
  - U/UTP: no apantallado
  - U/FTP: pares apantallados con lámina de aluminio
  - SF/UTP, S/FTP ...
- Clasificación según prestaciones (ancho de banda, interferencias, pérdidas de propagación):
  - Cat. 5, cat. 6, cat. 6a, cat. 7, cat. 8.1, cat. 8.2

# 3.1 Par trenzado: tipos

The following are the types of cable recognised in the ISO/IEC 11801 standard.

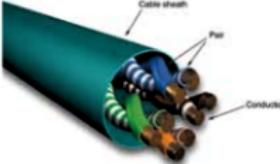
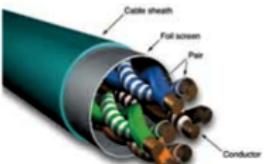
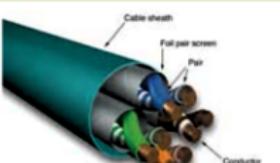
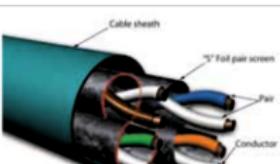
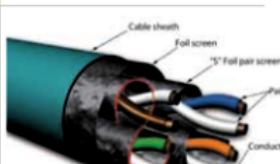
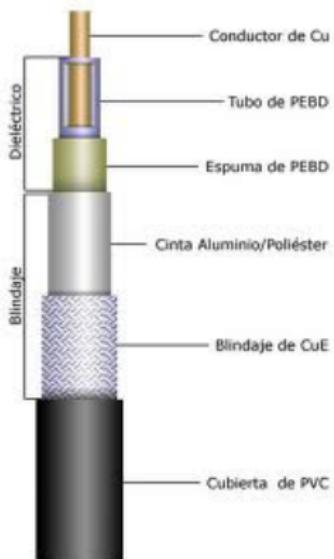
<b>U/UTP</b> Unscreened outer with unscreened twisted pairs		<b>SF/UTP</b> Screened braid and foil outer with unscreened twisted pairs	
<b>F/UTP</b> Screened foil outer with unscreened twisted pairs		<b>S/FTP</b> Screened braid outer with individual screened foil twisted pairs	
<b>U/FTP</b> Unscreened outer with individual screened foil twisted pairs		<b>F/FTP</b> Screened foil outer with individual screened foil twisted pairs	
<b>U/FTP</b> Unscreened outer with two sets of two pairs foil screened in "S" configuration		<b>F/FTP</b> Screened foil outer with two sets of two pairs foil screened in "S" configuration	

Figura: <https://electronics.stackexchange.com/questions/120737/>

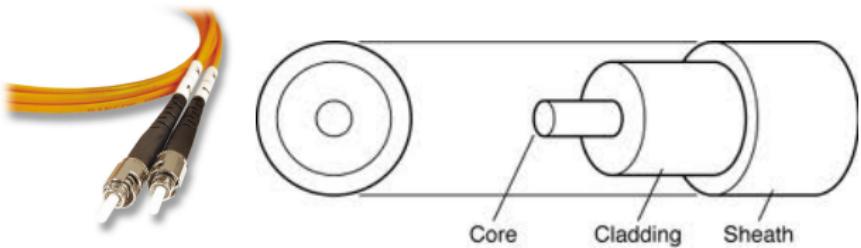
## 3.2 Cable coaxial

- Hilo conductor central de cobre rodeado por malla de hilos de cobre
- Conductores separados por plástico
- Puede estar apantallado
- Buen ancho de banda (1 GHz) y excelente inmunidad al ruido
- Coste elevado
- Está siendo sustituido por la fibra óptica
- Uso más común: TV y cableado final en accesos domésticos de fibra óptica



### 3.3 Fibra óptica

- Hilo muy fino de vidrio o plástico por el que se envían pulsos de luz
- El haz de luz queda confinado y se propaga por el interior de la fibra con cierto ángulo de reflexión
- La fuente de luz puede ser IDL (Injection Laser Diode) o LED (Light Emission Diode)
- En recepción se utilizan fotodiodos o fototransistores



### 3.3 Fibra óptica (II)

---



#### ► Ventajas

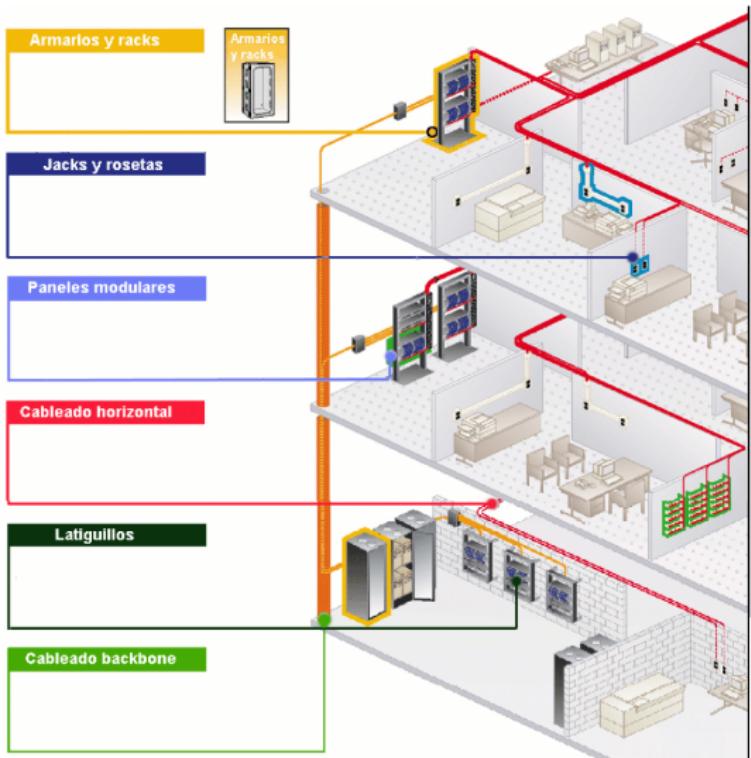
- Ancho de banda elevado (Gb/s)
- Baja atenuación → largas distancias
- Pequeñas dimensiones, flexibilidad, ligereza
- Inmunidad total a perturbaciones electromagnéticas
- Seguridad: no emite radiaciones y es difícil de "pinchar"
- Resistencia mecánica, calor, corrosión

#### ► Desventajas

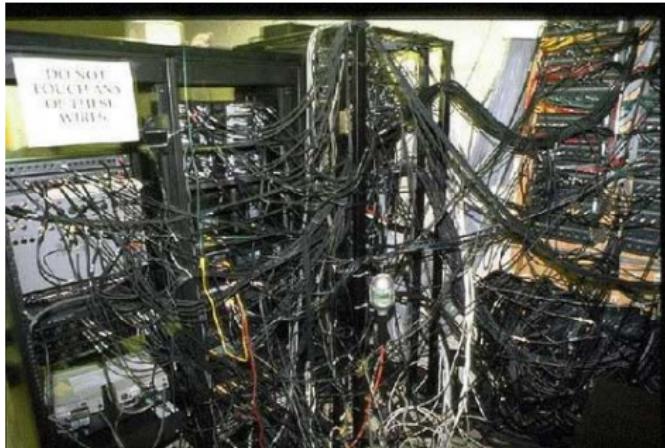
- Fragilidad de las fibras
- Emisores y receptores caros
- Velocidad condicionada por la electrónica de emisión y recepción (10 Gb/s)
- No pueden transmitir electricidad para alimentar receptores o repetidores
- Empalmes difíciles de realizar en campo

# 3.4 Cableado estructurado

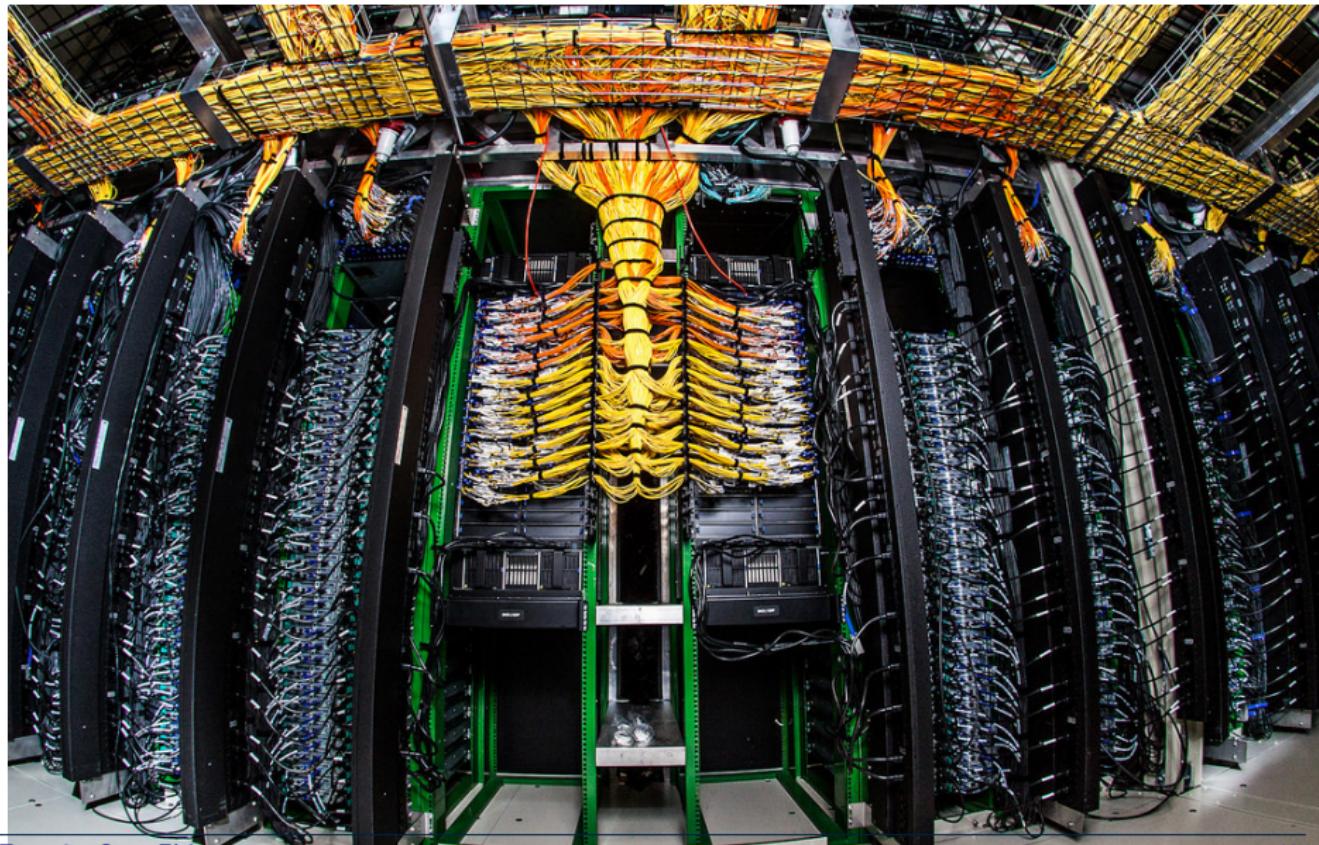
- Objetivo: optimizar gestión y mantenimiento
- División en tramos estructurados: cabl. horizontal + vertical



## 3.4 Cableado estructurado (II)



## 3.4 Cableado estructurado (III)



### 3.5 Comunicaciones inalámbricas

---

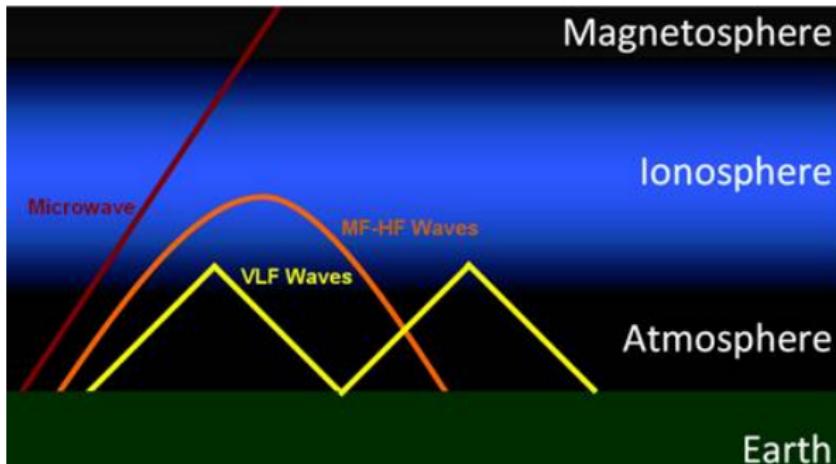


- International Telecommunication Union (ITU) y organismos de regulación locales establecen:
  - Banda de frecuencias
  - La división en canales de la banda
  - Las frecuencias dentro de cada canal
  - $P_{dB_{TX}}$
  - La potencia fuera del canal
  - El procedimiento para obtener la regulación
- ISM (Industrial, Scientific and Medical)-Unlisenced Band .  
E.g. (2.4-2.5 GHz)

### 3.5 Comunicaciones inalámbricas (II)



- Rango de frecuencias usado en radiocomunicaciones (20 kHz - 300 GHz):
  - Ondas de radio (30 MHz - 1 GHz)
    - Low-Power Wide-Area Networks (LPWANs)
  - Microondas terrestre (2 - 40 GHz)
    - Wi-Fi y LTE/3G/4G/5G
  - Microondas satélite (2 - 40 GHz)



### 3.5 Comunicaciones inalámbricas (III)

- FSO (Free Space Optics) para conectividad en la “Last mile” en la banda del infrarrojo (300 GHz, 400 THz)

#### CENTAURI - REPLACING FIBER UNDERGROUND



10Gbps

Full Duplex Consistent  
Data Rate



Impossible to Hack  
or Jam

3 km

Reliably under Equatorial  
Region P rain conditions



Zero Spectrum  
costs (plug-n-play)



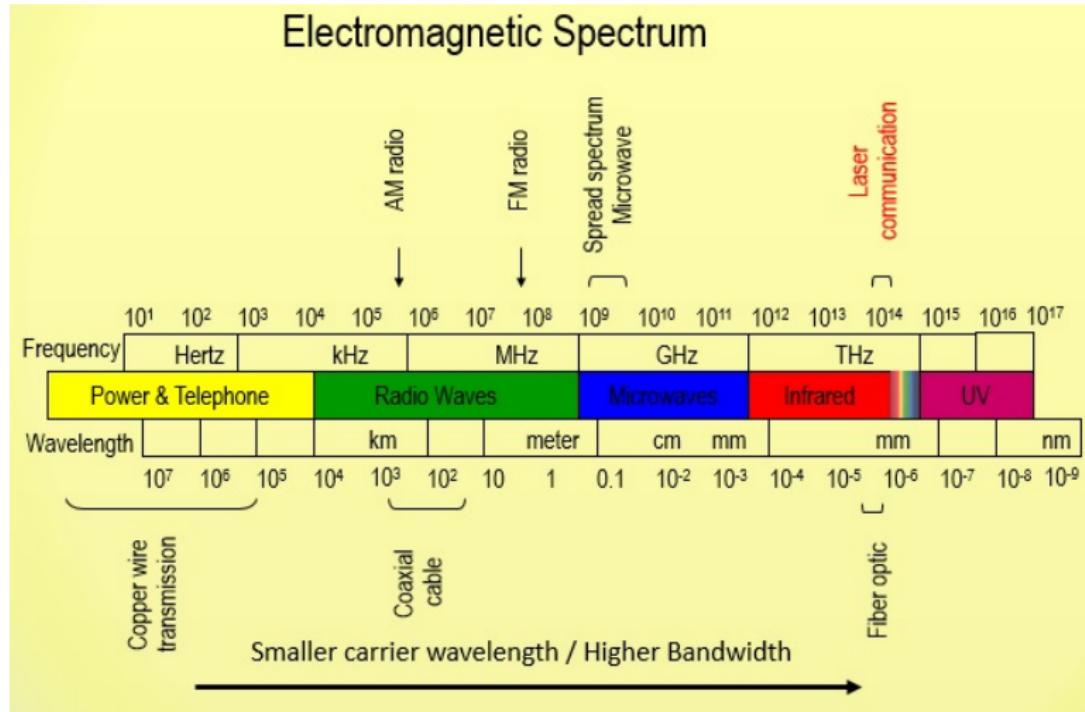
Fast installation  
in 1-4 hours



1-Person  
installation



# 3.6 Resumen



### 3.7 Velocidad de propagación



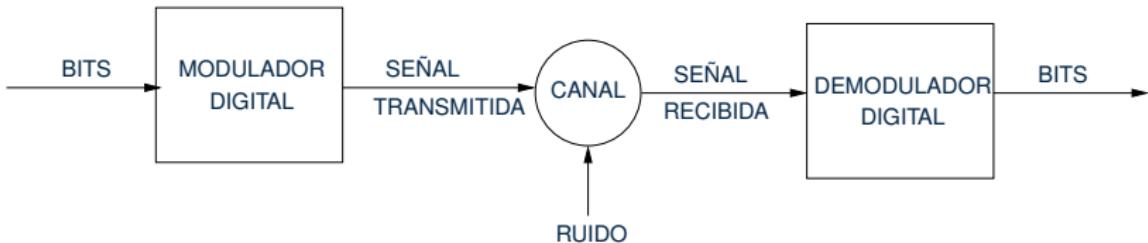
- $V_p$ : velocidad a la que una onda electromagnética viaja a través de un medio
- Factor de velocidad,  $VF$ : relación entre las velocidades de una onda EM en un medio y en el vacío

Medio	$V_p$ (m/s)	VF (%)
Espacio libre	$c = 3 \cdot 10^8$	100
Cable de cobre (cat. 7a)	$2.4 \cdot 10^8$	80
Fibra óptica	$\approx 2 \cdot 10^8$	67

☞ La distancia de la Tierra a Marte (cuando están lo más cerca posible) es aproximadamente de  $55 \times 10^9$  m y los datos viajan en el enlace a la velocidad de la luz. ¿Cuánto tarda la señal en viajar de la Tierra a Marte?

# 4 Transmisión digital

- Bloques básicos de un sistema de comunicaciones digitales:



- Señal transmitida/recibida:
  - *Señal analógica*: continua que varía suavemente en el tiempo para representar la información
  - *Señal digital*: representa los datos utilizando una secuencia discreta de valores

## 4 Transmisión digital (II)

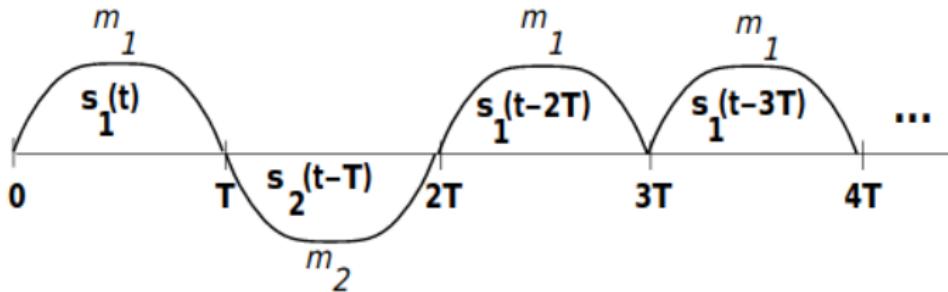
---



- Dada una secuencia binaria a transmitir, hay que convertir los bits en ondas compatibles con el canal:
  - Transmisión en banda base: *modulación por pulsos* o *modulación en banda base*. Por ej.: tarjeta red Ethernet sobre pares trenzados de cobre
  - Transmisión en canal paso banda: *modulación por portadora* (señal sinusoidal). Por ej.: tarjeta red Wi-Fi por el aire.

## 4.1 Símbolos y bits

- *Símbolo*: forma de onda de la señal modulada, se transmite durante un tiempo  $T_s$



Fuente: A. Goldsmith. Wireless communications. Stanford University, 2004.

- *Sistema M-ario*: usa un conjunto de  $M = 2^k$  símbolos. Cada símbolo codifica  $k = \log_2(M)$  bits
  - $k = 1$  bit/símb →  $M = 2$  símbolos: sistema 2-ario (binario)
  - $k = 2$  bits/símbolo →  $M = 4$  símbolos: sistema 4-ario

## 4.2 Tasa de símbolos



- *Tiempo de símbolo ( $T_s$ )*: tiempo entre transiciones de símbolos
- *Tasa de símbolos ( $R_s$ )*: número de símbolos en la señal modulada por unidad de tiempo.  
Se mide en *baudios* (Bd, símbolos/s)

$$R_s = \frac{\text{símbolos}}{\text{tiempo}} = \frac{1}{T_s} \quad (\text{Bd})$$

## 4.3 Tasa de bits

- *Tasa de bits* ( $R_b$ ): número de bits en la señal modulada por unidad de tiempo. Se mide en *bits/segundo* (b/s, bps)

$$R_b = \frac{\text{bits}}{\text{tiempo}}$$
 (b/s)

- Relación entre  $R_b$  y  $R_s$

$$R_b = \frac{\text{bits}}{\text{tiempo}} = \frac{k \cdot \text{símbolos}}{\text{tiempo}} = R_s \cdot k = R_s \cdot \log_2(M)$$

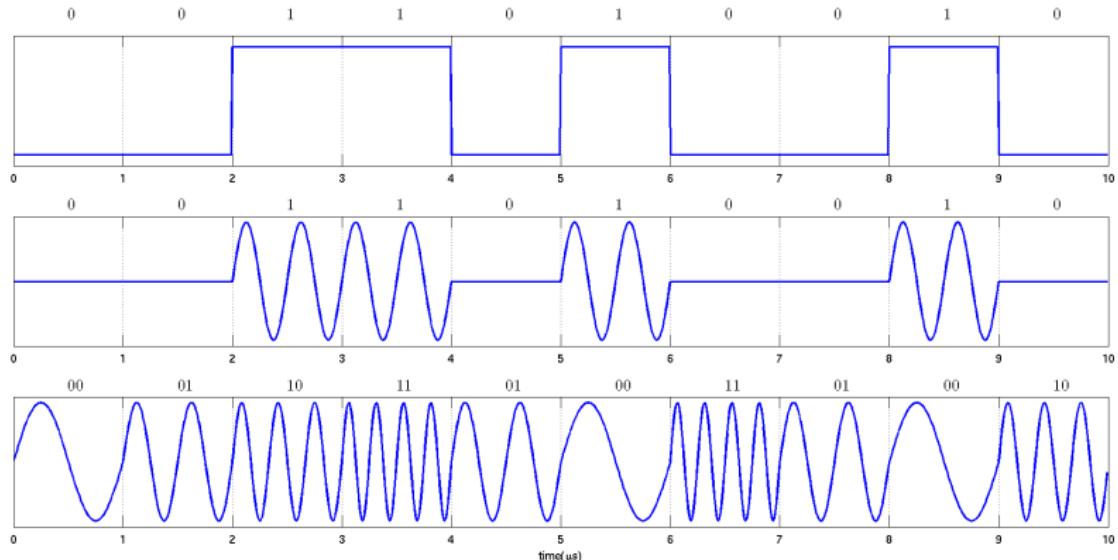
- $k = 1$  bit/símbolo  $\rightarrow R_b = R_s$
- $k = 2$  bits/símbolo  $\rightarrow R_b = 2 \cdot R_s$

- Algunos valores

- Red L1.02: 1 Gbps
- FFTH (fiber to the home): 100 Mbps - 1 Gbps
- Wi-Fi 802.11b/g/n: hasta 300 Mbps

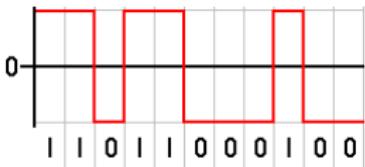
## 4.3 Tasa de bits (II)

✍ Asumiendo que el tiempo de símbolo es  $1 \mu\text{s}$ , calcula la tasa de símbolos y la tasa de bits de las siguientes señales moduladas (PCM-ASK-FSK). Indica asimismo el número de símbolos  $M$ .

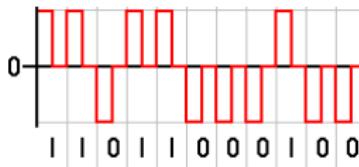


## 4.4 Modulación por pulsos

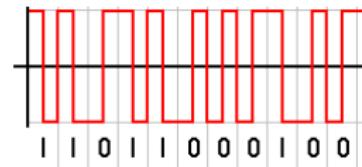
- Pulse Code Modulation (PCM)
- Los bits se transforman en formas de onda pulsadas que se transmiten por un canal banda base
- Características de las formas de onda o códigos de línea:
  - Nonreturn-to-zero (NRZ) / Return-to-zero (RZ)
  - Unipolar / Bipolar
  - Phase encoded
  - Multilevel binary



NRZ



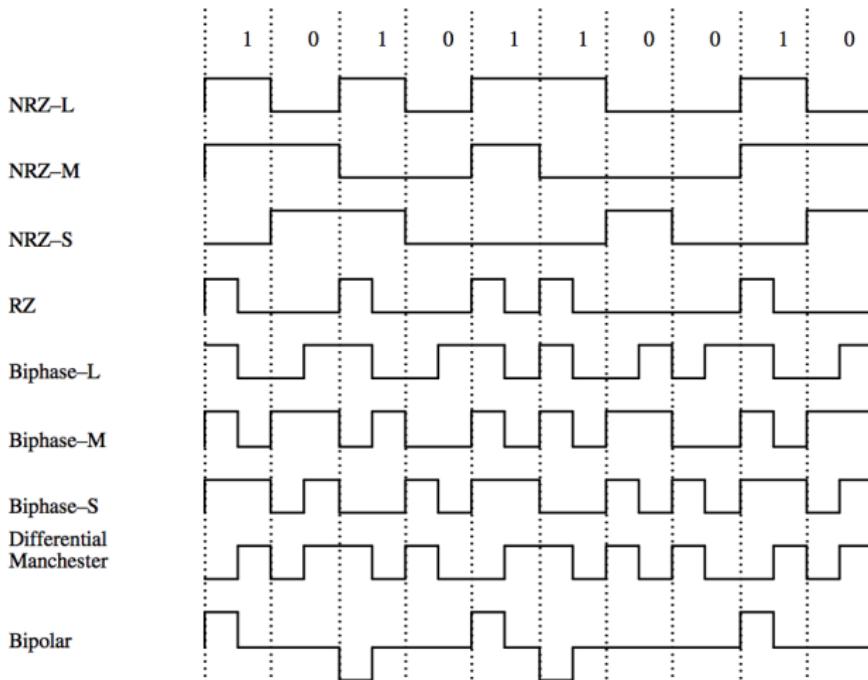
RZ



Phase encoded

## 4.4 Modulación por pulsos (II)

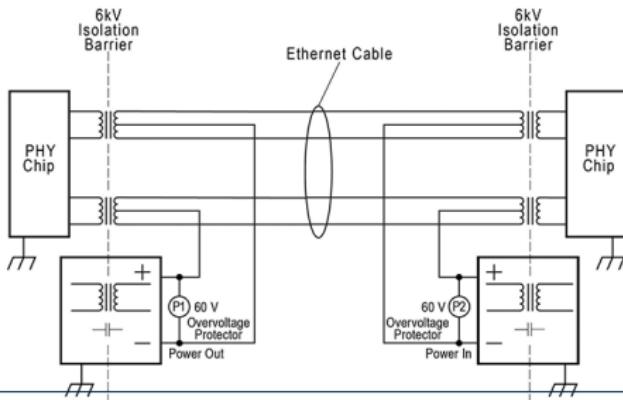
### ► Codificaciones más usadas:



Fuente: <https://commons.wikimedia.org/w/index.php?curid=47007673>

## 4.4 Modulación por pulsos (III)

- NRZ-L, por nivel:  $0 \equiv V^-$ ,  $1 \equiv V^+$
- NRZ-M, diferencial:  $0$  mantiene  $V$ ,  $1$  cambia  $V$
- RZ: transición a  $0V$  durante el pulso
- NRZ-AMI, inversión alterna:  $0 \equiv 0V$ ,  $1$  alterna  $V^{+/-}$
- Manchester:  $0 \equiv V^- \rightarrow V^+$ ,  $1 \equiv V^+ \rightarrow V^-$   
(señal de reloj integrada en la codificación)
- Duobinario-NRZ: mismo valor en bit  $\equiv 0V$ ,  
distinto valor  $\equiv$  alterna  $V^{+/-}$



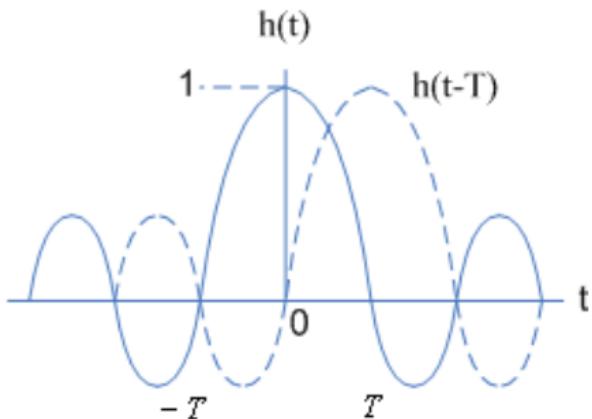
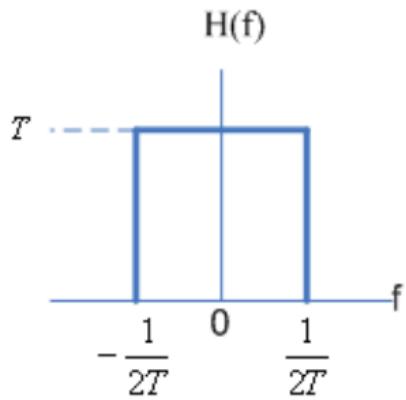
## 4.4 Modulación por pulsos (IV)



- ¿Por qué hay tantos códigos de línea PCM?
- Por la variedad de requisitos que tienen las distintas aplicaciones
  - Velocidad de transmisión
  - Coste
  - Prestaciones
  - Tecnología de implementación
- La elección de un código de línea determina
  - Uso del ancho de banda
  - Componente continua (DC)
  - Sincronización
  - Capacidad de detectar errores
  - Inmunidad al ruido
  - Inversión de polaridad del cable

## 4.4.1 Interferencia intersímbolos (ISI)

- Los pulsos recibidos se solapan por la distorsión del canal
- Nyquist<sup>1</sup>: el ancho de banda mínimo teórico para detectar  $R_s$  Bd sin ISI es  $Rs/2$  Hz  $\rightarrow B \geq Rs/2$



- Un sistema con ancho de banda  $B$  soporta un máximo de  $R_s = 2 \cdot B$  Bd, es decir,  $Rs/B \leq 2$  Bd/Hz

## 4.4.2 Eficiencia espectral

- Tasa de bits por ancho de banda,  $E = R_b/B$  b/s/Hz

$$E = R_b/B = R_s \cdot k/B = R_s \cdot \log_2(M)/B$$

Como  $R_s/B \leq 2$ , entonces  $E \leq 2 \cdot \log_2(M)$  b/s/Hz

- Ejemplo: cable Marea, 2019.

### Real-time 16QAM Transatlantic Record Spectral Efficiency of 6.21 b/s/Hz Enabling 26.2 Tbps Capacity

Stephen Grubb<sup>1</sup>, Pierre Mertz<sup>2</sup>, Ales Kumpera<sup>3</sup>, Lee Dardis<sup>4</sup>, Jeffrey Rahn<sup>4</sup>, James O'Connor<sup>4</sup>, Matthew Mitchell<sup>1</sup>

<sup>1</sup>Facebook, 1 Hacker Way, Menlo Park, CA 94025

<sup>2</sup>Infinera Maryland, 9005 Junction Dr., Savage, MD 20763

<sup>3</sup>Infinera Canada, 555 Legget Dr, Ottawa, ON K2K 2X3, Canada

<sup>4</sup>Infinera Corporation, 140 Caspian Ct., Sunnyvale, CA 94089

E-mail address: pmertz@infinera.com

**Abstract:** Real-time, error-free 16QAM transmission at a record spectral efficiency of 6.21 b/s/Hz enables transatlantic (6,644 km) fiber capacity of 26.2 Tbps, using precision, multi-carrier common wavelocking; digitally synthesized subcarriers; near-Nyquist pulse shaping; and large-area, positive dispersion fiber.

☞ ¿Cuál es el valor máximo de  $E$  para una modulación 16QAM? Calcula el cociente  $R_s/B$  del cable Marea.

## 4.5 Modulación por portadora



1474

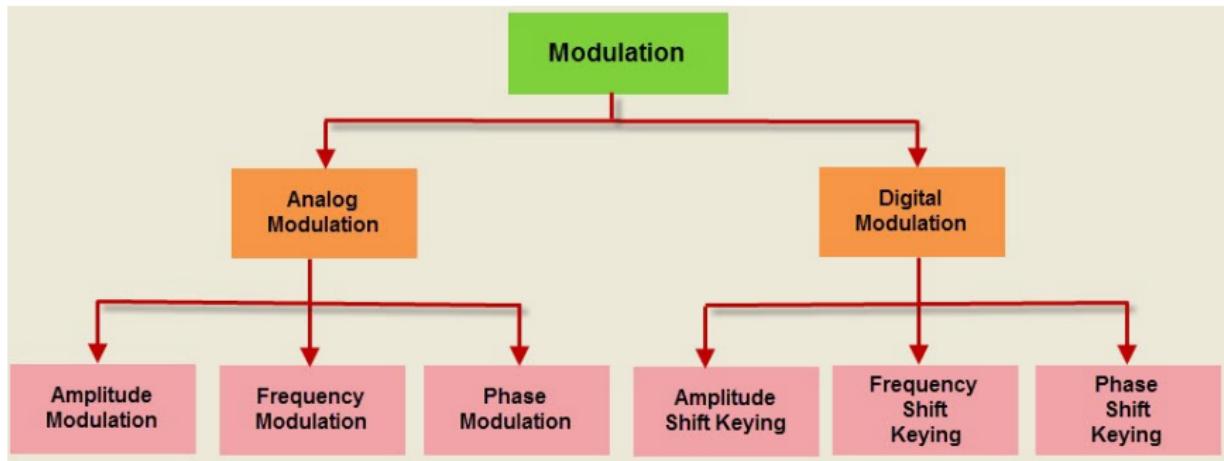
- Proceso por el que la amplitud, frecuencia o fase de una portadora varía según la información a transmitir

$$s(t) = A \cdot \cos(2\pi f_c t + \phi)$$

- Uso más eficiente del canal: las ondas cuadradas son espectralmente ineficientes
- Diseño hardware más sencillo: antenas, filtros, amplificadores ...
- Módem:
  - Modulador: modifica alguna característica de la portadora
  - Demodulador: elimina la portadora



## 4.5.1 Modulación analógica vs. digital



## 4.5.2 ASK (amplitude shift keying)

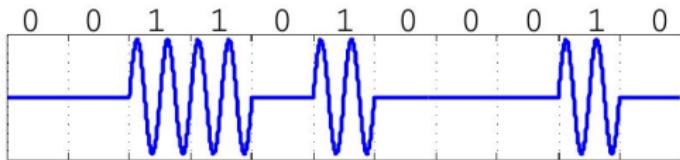


- Los símbolos se representan mediante diferentes amplitudes ( $A_i$ ) de la portadora:

$$s(t) = A_i \cdot \cos(2\pi f t + \phi), i = 1, \dots, M$$

- Por ejemplo, ASK binario (*on-off keying, OOK*):

$$s(t) = \begin{cases} A \cdot \cos(2\pi f_c t) & 1 \\ 0 & 0 \end{cases}$$



- Sensible a cambios repentinos de la ganancia (ruido impulsivo)
- Se usa en fibras ópticas

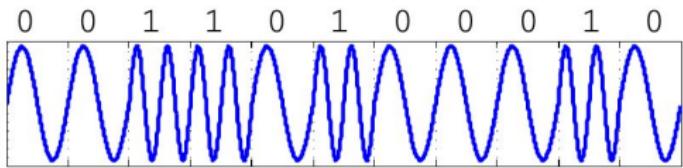
## 4.5.3 FSK (frequency shift keying)

- ▶ Los símbolos se representan mediante diferentes frecuencias ( $f_i$ ) de la portadora

$$s(t) = A \cdot \cos(2\pi f_i t + \phi), i = 1, \dots, M$$

- ▶ Por ejemplo, FSK binario (BFSK):

$$s(t) = \begin{cases} A \cdot \cos(2\pi f_1 t) & 1 \\ A \cdot \cos(2\pi f_2 t) & 0 \end{cases}$$



- ▶ Se utiliza en telefonía digital para identificación de llamada

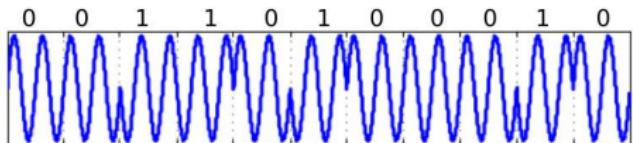
## 4.5.4 PSK (phase shift keying)

- ▶ Los valores de los símbolos se representan mediante diferentes fases ( $\phi_i$ ) de la portadora:

$$s(t) = A \cdot \cos(2\pi ft + \phi_i), i = 1, \dots, M$$

- ▶ Por ejemplo, PSK binario (BPSK):

$$s(t) = \begin{cases} A \cdot \cos(2\pi ft + \pi) & 1 \\ A \cdot \cos(2\pi ft + 0) & 0 \end{cases}$$



- ▶ PSK diferencial (DPSK): la fase depende del anterior símbolo transmitido
- ▶ Amplio uso: 802.11b, Bluetooth, Zigbee, sonda espacial New Horizons ...

## 4.5.5 QAM (quadrature amplitude mod.)



- Combinación de ASK y PSK
- Usa distintas amplitudes ( $A$ ) y fases ( $\phi$ ) para codificar varios bits por símbolo
- Ejemplo: 8-QAM circular: dos amplitudes ( $A$  y  $B$ ) y cuatro fases por amplitud

$$000 \rightarrow s(t) = A \cdot \sin(2\pi ft + 0^\circ)$$

$$001 \rightarrow s(t) = B \cdot \sin(2\pi ft + 45^\circ)$$

$$010 \rightarrow s(t) = A \cdot \sin(2\pi ft + 90^\circ)$$

$$011 \rightarrow s(t) = B \cdot \sin(2\pi ft + 135^\circ)$$

$$100 \rightarrow s(t) = A \cdot \sin(2\pi ft + 180^\circ)$$

$$101 \rightarrow s(t) = B \cdot \sin(2\pi ft + 225^\circ)$$

$$110 \rightarrow s(t) = A \cdot \sin(2\pi ft + 270^\circ)$$

$$111 \rightarrow s(t) = B \cdot \sin(2\pi ft + 315^\circ)$$

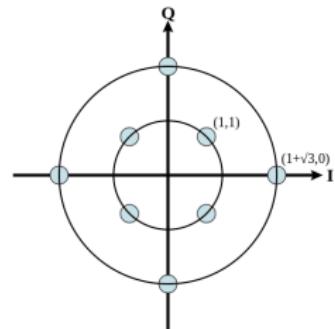
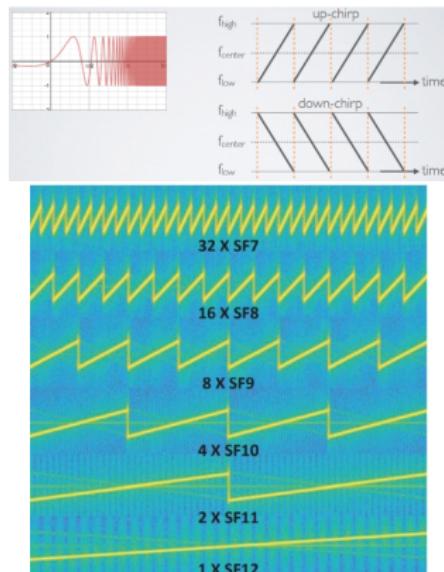


Diagrama de constelación  
Fuente: Life of Riley

- Uso: Wi-Fi (802.11ax: 16-,64-,256- y 1024-QAM)

## 4.5.6 CSS (chirp spread spectrum)

- Propietaria de LoRa (IoT)
- La frecuencia de la portadora crece (1) o decrece (0) en un determinado tiempo
- Inmunidad frente al ruido, largo alcance



## 4.5.6 Ejercicio



Un módem transmite a 1200 Bd y 2400 bps utilizando una modulación M-QAM. Si el valor máximo de  $R_s$  para una banda  $B$  en la modulación QAM es aproximadamente igual a la banda de paso, calcula:

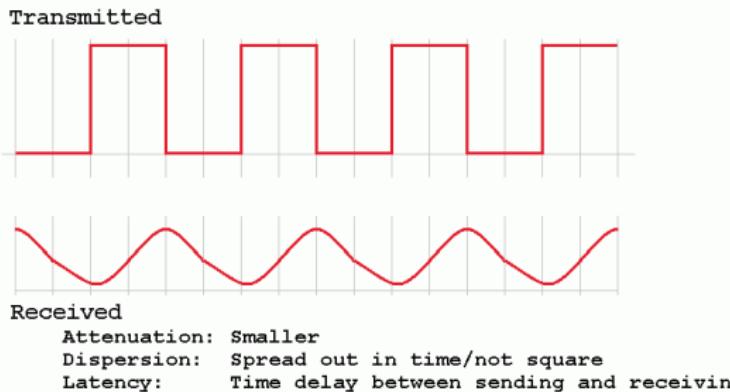
- ¿Cuántos bits de información transporta cada símbolo?
- ¿Cuántos símbolos distintos ( $M$ ) genera el módem?
- ¿Qué ancho banda necesita?
- Si la portadora es  $f_p = 1800$  Hz, ¿cuál es la frecuencia inferior de corte? ¿Y la superior?

# 5.1 Perturbaciones en la transmisión



Cualquier canal tiene perturbaciones:

- *Atenuación*: pérdida de energía de la señal al propagarse
  - Espacio libre
  - Absorción en la atmósfera: agua, niebla ...
  - Absorción debida a obstáculos: puertas, paredes ...
- *Distorsión de retardo*: distintas frecuencias viajan a distintas velocidades (medios dispersivos)



## 5.1 Perturbaciones en la transmisión (II)



1474

- *Ruido*: señales insertadas entre emisor y receptor
  - Ruido térmico: agitación de los electrones
  - Intermodulación: varias frecuencias en el mismo medio
  - Diafonía (cable): acoplamiento entre líneas
  - Ruido impulsivo: pulsos cortos e irregulares
  - Multipath (inalámbrico): reflexiones retardadas de la señal

Las perturbaciones pueden hacer que el receptor confunda los símbolos de la señal

- El tipo de modulación, el número de símbolos  $M$  y su tasa  $R_s$  vendrán limitados por la capacidad para distinguirlos

## 5.2 Teorema de Shannon-Hartley

- *Shannon*: la capacidad máxima  $C$  de un canal perturbado por ruido es función de la potencia media de la señal recibida  $S$ , la potencia media del ruido  $N$ , y el ancho de banda  $B$ .

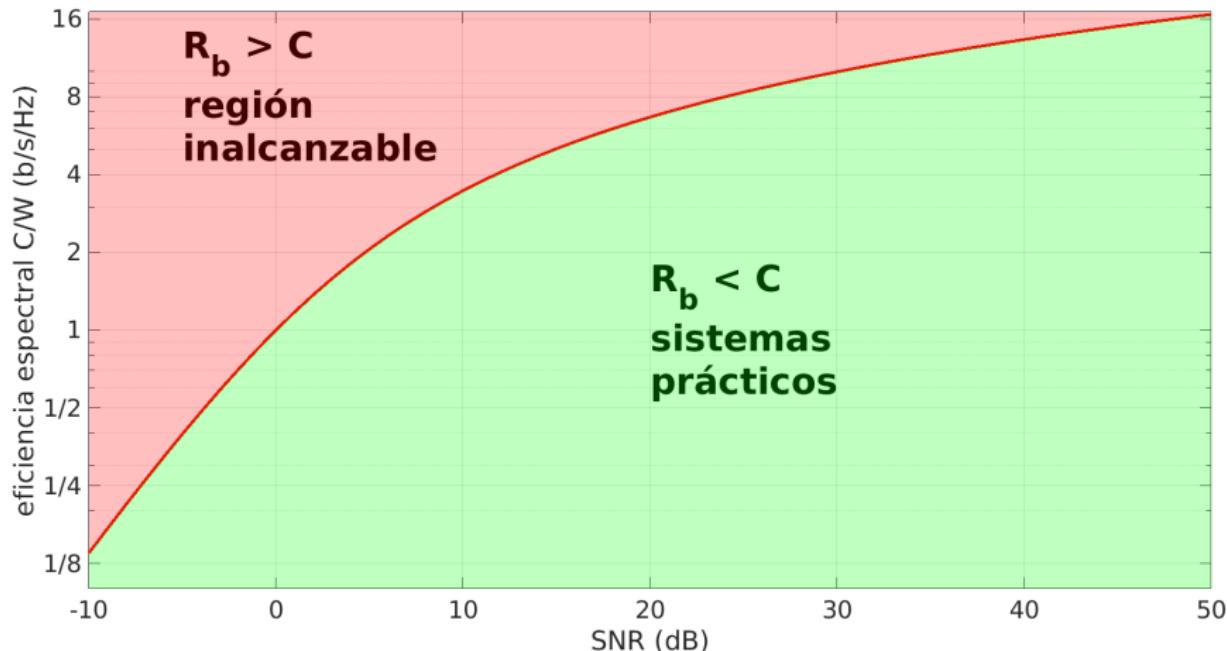
$$C = B \cdot \log_2 \left( 1 + \frac{S}{N} \right)$$

con  $C$  en bits por segundo (bps) y  $B$  en Hercios (Hz).

- Es posible transmitir información a una tasa  $R_b \leq C$  con una probabilidad de error arbitrariamente baja
- Recordar:  $\log_2(x) = \frac{\ln(x)}{\ln(2)} = \frac{\log_{10}(x)}{\log_{10}(2)}$

## 5.2 Teorema de Shannon-Hartley (II)

### ► Límite de prestaciones



Fuente: elaboración propia.

## 5.2 Teorema de Shannon-Hartley (III)

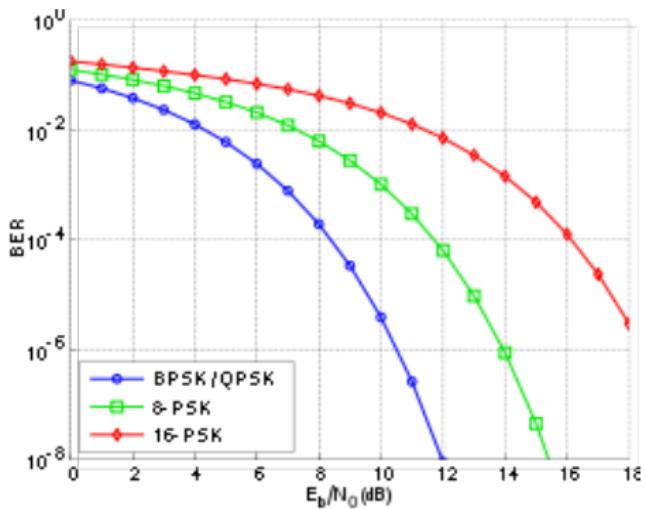


1474

- ✍ Considera un canal telefónico con ancho de banda de 3 kHz.
- ¿Cuál es la capacidad del canal si SNR es 30 dB?
  - ¿Cuál es el mínimo valor de SNR requerido para transmitir a 4800 bps?
  - Repetir el apartado anterior para 19200 bps.

## 5.3 BER: probabilidad de error de bit

*Bit Error Ratio (BER)*: fracción de bits erróneos (adimensional)



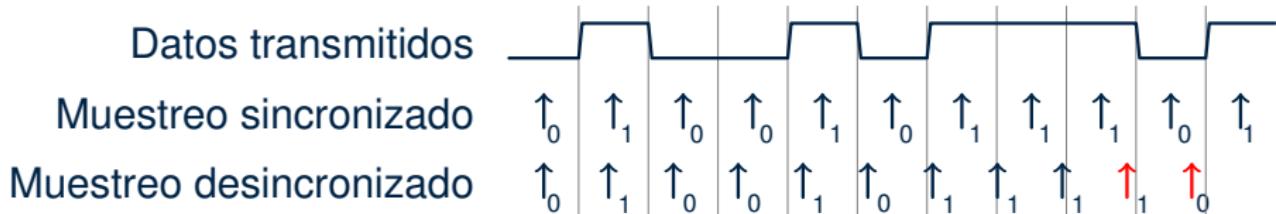
💡 Considera un canal con  $B = 3$  kHz,  $\text{SNR} = 12$  dB por el que se transmite una señal BPSK a 3000 bps. Calcula la probabilidad de error de bit (BER).

Nota:  $E_b/N_0 = (S/N) \cdot (B/R_b)$ .

# 6 Sincronismo

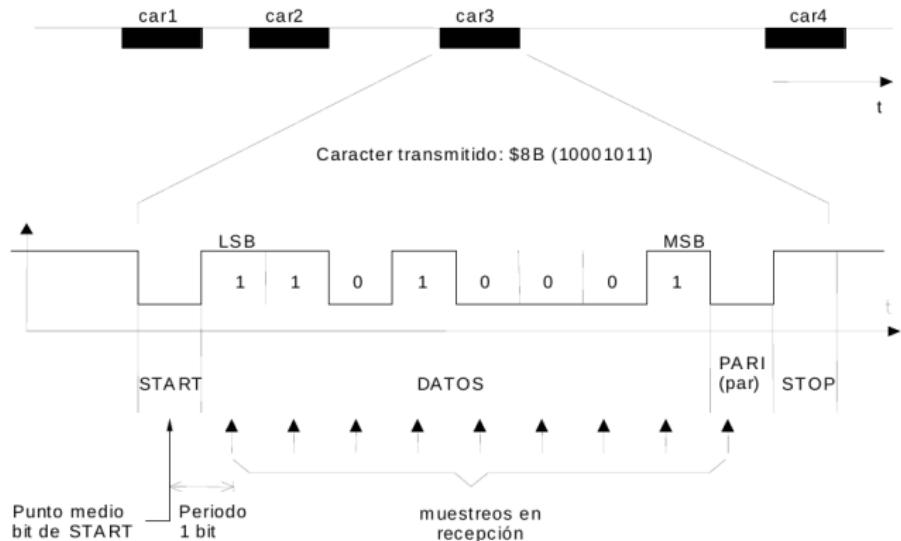


- El receptor debe saber en qué momento:
  - empieza/acaba cada bloque de datos (trama)
  - muestrear cada bit (símbolo)
- Alta velocidad → relojes con precisión de  $\mu\text{s}/\text{ns}$
- Todos los relojes se atrasan/adelantan → la precisión requerida no se puede garantizar durante mucho tiempo
- Dos estrategias para sincronizar emisor y receptor:  
transmisión asíncrona y síncrona



# 6.1 Transmisión asíncrona

- ▶ Envío de caracteres entre 5 y 8 bits + pausa
- ▶ Sincronización de relojes al inicio de cada carácter
- ▶ Sencilla y barata
- ▶ Velocidad de transmisión baja, sobrecarga alta



## 6.1 Transmisión asíncrona (II)

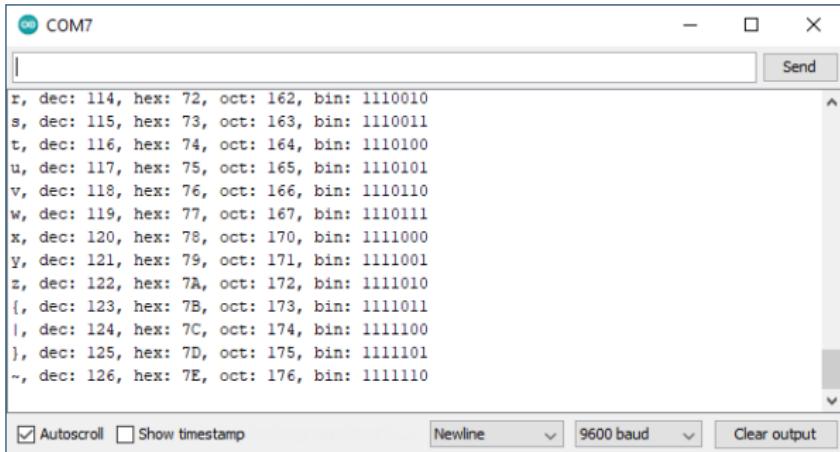
Se especifica:

- No transmisión (*idle*): tensión correspondiente a 1 binario
- Velocidad de transmisión, es decir, el tiempo de bit
- Número de bits de cada carácter
- Bit de paridad
- Tiempo de parada

Velocidad	Bits de datos	Paridad	Tiempo de parada
4800	5	N: <i>none</i> (sin)	1
9600	6	E: <i>even</i> (par)	1.5
19200	7	O: <i>odd</i> (impar)	2
...	8		

# 6.1 Transmisión asíncrona (III)

- Eg. Arduino: UART+USB-to-serial chip+Virtual COM port+serial monitor. 9600/8N1



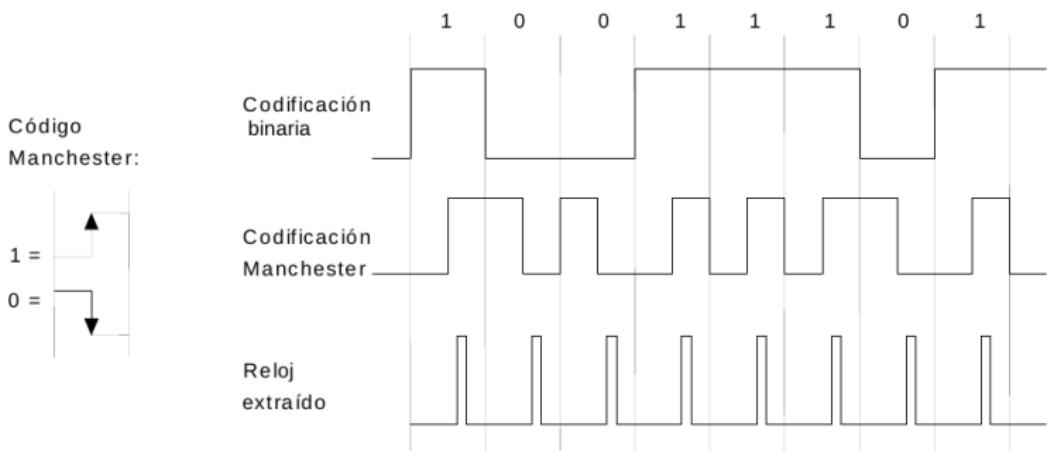
The screenshot shows a serial monitor window titled "COM7". The main area displays a list of characters with their corresponding numerical values in decimal, hex, octal, and binary formats. The characters listed are r, s, t, u, v, w, x, y, z, {, }, |, ], and ~. The window includes standard controls like minimize, maximize, and close buttons, and a "Send" button. At the bottom, there are checkboxes for "Autoscroll" and "Show timestamp", a "Newline" dropdown set to "Newline", a "9600 baud" dropdown, and a "Clear output" button.

Character	dec:	hex:	oct:	bin:
r	114	72	162	1110010
s	115	73	163	1110011
t	116	74	164	1110100
u	117	75	165	1110101
v	118	76	166	1110110
w	119	77	167	1110111
x	120	78	170	1111000
y	121	79	171	1111001
z	122	7A	172	1111010
{	123	7B	173	1111011
	124	7C	174	1111100
}	125	7D	175	1111101
~	126	7E	176	1111110

## 6.2 Transmisión síncrona



- Contexto: flujo constante de grandes bloques de bits o altas velocidades de transmisión
- Solución 1: línea adicional que transmite la señal de reloj
  - Funciona bien en distancias cortas
- Solución 2: la información del reloj se empotra en la señal de datos
  - Por ej., Manchester codifica una transición a mitad de bit



## 6.2 Transmisión síncrona (II)

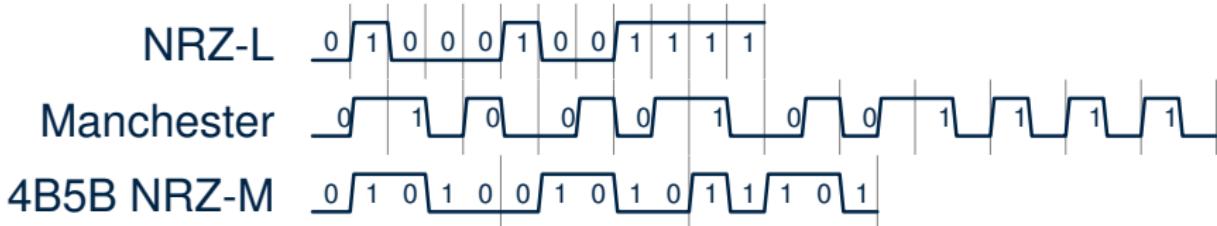


- Codificaciones sin información de reloj: códigos de sustitución garantizan transiciones: 4B5B, B8ZS, HDB3

Datos	4B5B	Datos	4B5B	Datos	4B5B	Datos	4B5B
0000	11110	0100	01010	1000	10010	1100	11010
0001	01001	0101	01011	1001	10011	1101	11011
0010	10100	0110	01110	1010	10110	1110	11100
0011	10101	0111	01111	1011	10111	1111	11101

4B5B: nunca más de tres 0s consecutivos

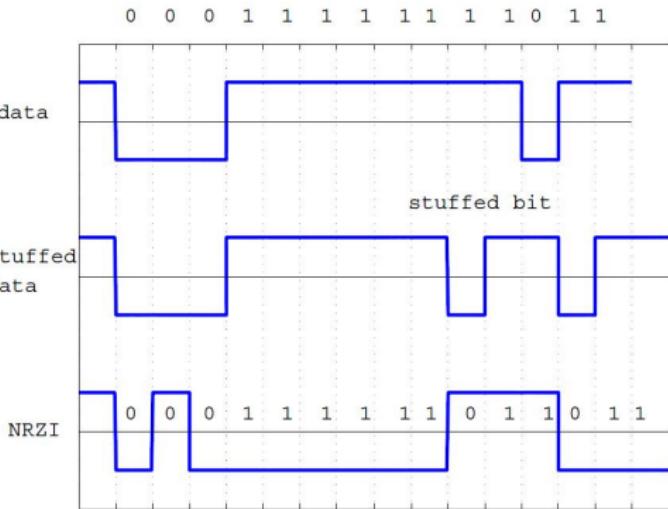
- Ejemplo NRZ-L (no sincronizable), Manchester, 4B5B sobre NRZ-M (Ethernet 100 Mb/s), todos con misma tasa de símbolos



## 6.2 Transmisión síncrona (III)

- *Inserción de bits (bit stuffing):* dos aplicaciones
  - El número consecutivo de bits del mismo valor se limita insertando un bit del valor opuesto
  - Para que los datos de una trama no contengan la secuencia delimitadora de la misma

Se utiliza en USB (Universal Serial Bus) o HDLC (High-Level Data Link Control):  
Emisor  $111111 \rightarrow 1111110$   
Receptor  $1111110 \rightarrow 1111111$   
(NRZI: NRZ-M invertido)



# 7 Modos de transmisión



Simplex: comunicación en un único sentido



Half-duplex: en ambos sentidos pero no a la vez



Full-duplex: en ambos sentidos simultáneamente



# 7 Modos de transmisión (II)

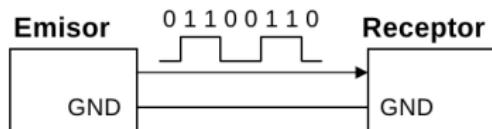


**Serie:** las señales de múltiples bits se transmiten en el mismo canal *secuencialmente*

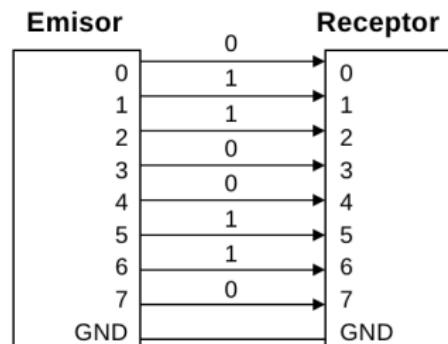
- Mínimo dos cables: señal y referencia

**Paralelo:** las señales de múltiples bits se transmiten en diferentes canales *simultáneamente*

- Más caro ya que requiere más cables
- Menor distancia (diafonía, sincronización entre canales)



Transmisión serie



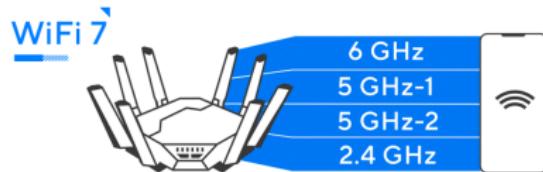
Transmisión paralelo

# 7 Modos de transmisión (III)



**Transmisión:** tanto la transmisión *serie* como la *paralela* pueden realizarse de forma *síncrona* o *asíncrona*

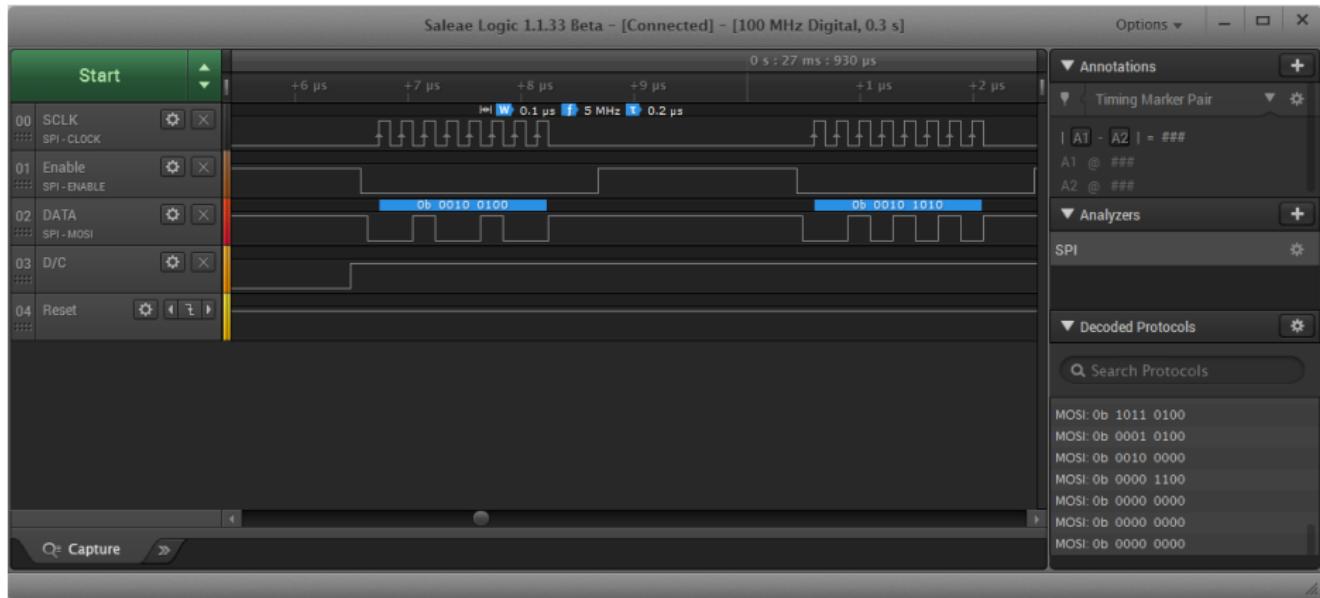
- Serie:
  - Síncrono: Ethernet, CAN, I2C, JTAG, USB, SPI
  - Asíncrono: UART, RS-232, SCI, Virtual COM port (USB-to-serial adapter)
- Paralelo síncrono: PCI, PCIe, PATA (Parallel ATA)



# 7 Modos de transmisión (IV)



- Logic Analyzer: hardware + virtual COM port+serial monitor + logic analyzer software



# 8 Conclusiones



# **Redes de Computadores**

## **Tema 3 – Capa de enlace de datos**

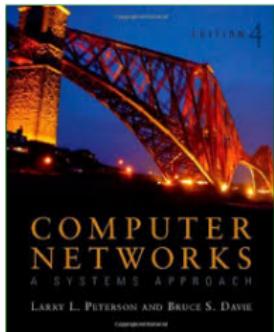
**Natalia Ayuso, Juan Segarra y Jesús Alastruey**



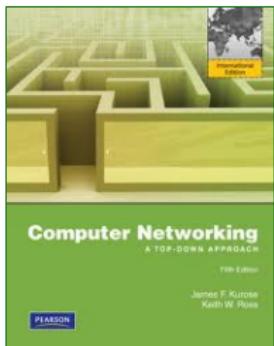
Departamento de  
Informática e Ingeniería  
de Sistemas

**Universidad** Zaragoza

1. Introducción
2. Definiciones y métricas
3. Control de acceso al medio (MAC)
4. Protocolos de particionado de canal
5. Protocolos de acceso aleatorio
6. Protocolos “por turnos”
7. Comutación en Ethernet
8. Control del enlace de datos
9. Control de errores
10. Secuenciación de datos



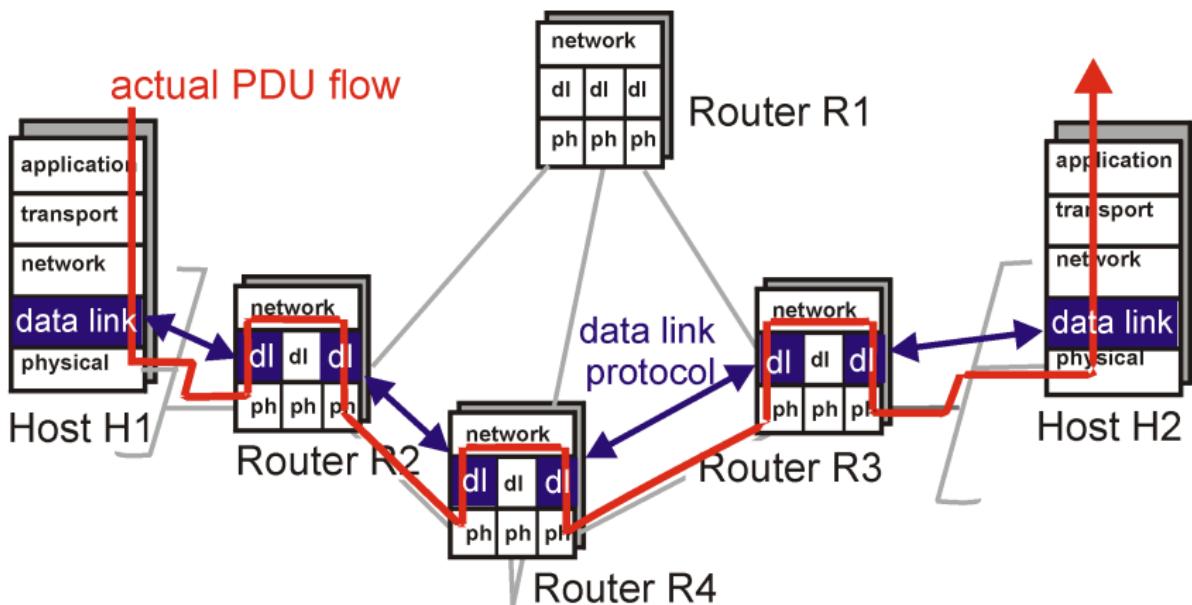
Capítulo 2



Capítulo 5

# 1 Introducción

## ► Contexto:



Fuente: The Data Link Layer: Introduction, Services

# 1 Introducción (II)



Funciones típicas de la capa de enlace:

- Reglas para que varias entidades compartan un mismo canal de transmisión: *control de acceso al medio (MAC)*
- *Comunicación fiable* entre dos entidades:
  - Control de errores
  - Secuenciación de datos



- Unidad básica de nivel de enlace: *trama (frame)*

## 2 Definiciones y métricas

---



Longitud trama ( $L$ ): longitud de la trama (bits, b)

Velocidad de transmisión ( $V_t$ ): “User data rate” velocidad nominal de la capa MAC (bits/segundo,bps)

Tiempo de transmisión ( $T_t$ ): tiempo necesario para injectar una trama en el medio de transmisión (segundos, s)  $T_t = \frac{L}{V_t}$

Velocidad de capa física ( $V_{PHY}$ ): tasa de bits que se inyecta al medio de transmisión (bits/segundo, bps). Incluye los datos de usuario y el overhead requerido en la capa física. Eg.

4B/5B

- Ethernet 100BASE-TX,  $V_t = 100$  Mbps (capa MAC)  
 $V_{PHY} = \frac{5}{4} \times V_t = 125$  Mbps (capa Física)

## 2 Definiciones y métricas (II)



- Ethernet anuncia la capacidad del usuario
- Wi-Fi anuncia la máxima posible. E.g. 802.11a/g de 54 Mbps → 20-25 Mbps; 802.11n de 600 Mbps → 150 -200 Mbps

Ethernet Standard	Common Name	Data Rate (Bit Rate)
10BASE-T	Standard Ethernet	10 Mbps
100BASE-TX	Fast Ethernet	100 Mbps (125 Mbps 4B/5B)
1000BASE-T	Gigabit Ethernet	1 Gbps (1.25 Gbps 8B/10B)
10GBASE-T	10 Gigabit Ethernet	10 Gbps
25G, 40G, 100G, 400G	Data Center Standards	25 Gbps up to 400 Gbps

Standard	Wi-Fi Generation	Frequency Band(s)	Data Rate (Bit Rate)
802.11b	Wi-Fi 2	2.4 GHz	11 Mbps
802.11a/g	Wi-Fi 3	5 GHz (a), 2.4 GHz (g)	54 Mbps
802.11n	Wi-Fi 4 (HT)	2.4 GHz and 5 GHz	600 Mbps
802.11ac	Wi-Fi 5 (VHT)	5 GHz	6.9 Gbps
802.11ax	Wi-Fi 6 (HE)	2.4 GHz, 5 GHz, and 6 GHz	9.6 Gbps
802.11be	Wi-Fi 7 (EHT)	2.4 GHz, 5 GHz, and 6 GHz	46 Gbps

### ➤ Velocidad en la LAN:

- Real: iperf3, transferencia de fichero (1 GB)...
- La que se negocia en la red (máxima): ~\$ iw wlp3s0 link  
~\$ cat /sys/class/net/eth0/speed

## 2 Definiciones y métricas (III)

---



Distancia ( $D$ ): longitud del enlace (metros, m)

Velocidad de propagación ( $V_p$ ): velocidad a la que la onda EM viaja por el enlace (metros/segundo, m/s) [Tema 2]

Tiempo de propagación ( $T_p$ ): tiempo necesario para que la onda EM viaje de emisor a receptor (segundos, s)

E.g. en ethernet, un segmento de red está limitado a 100 m.

Del dispositivo al router pasando por 3 switches, 300 m. Los retardos son del orden de los  $\mu$  s

Round Trip Time (RTT): es el tiempo de ida y vuelta de la señal ( $RTT = 2 \cdot T_p$ ).

## 2 Definiciones y métricas (IV)

Tiempo de acceso al canal ( $T_a$ ): desde que un nodo quiere transmitir hasta que empieza a transmitir la trama «definitiva» (esperas por colisiones, por turnos, etc.)

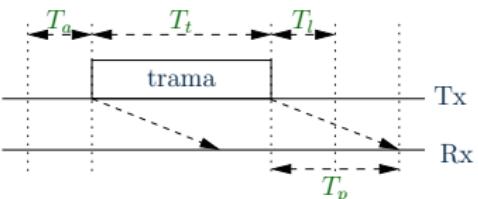
Tiempo liberación del canal ( $T_l$ ): desde finalización de transmisión hasta intento de una nueva transmisión

Tiempo de procesamiento: espera / toma de decisiones

Velocidad efectiva ( $V_e$ ): tasa media de bits de datos enviados

Utilización efectiva del canal ( $U_e$ ): porcentaje de tiempo en el que se transmiten datos por el canal

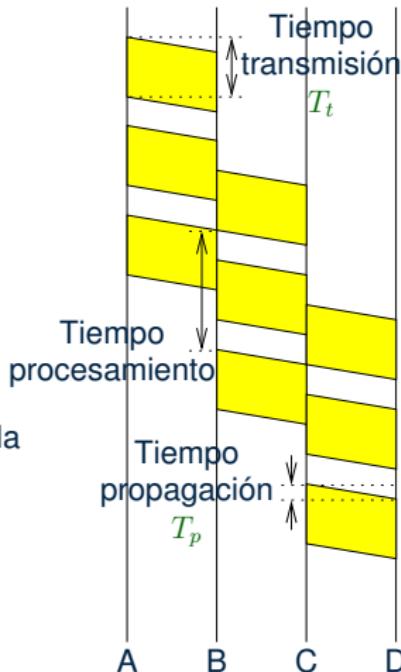
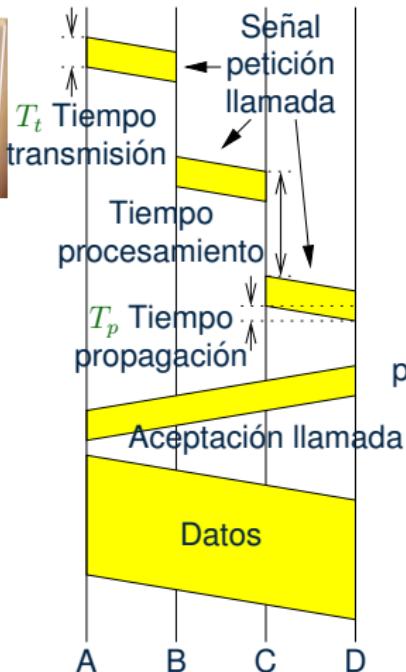
$$V_e = \frac{\text{datos}}{T_a + T_t + T_l} \quad U_e = \frac{V_e}{V_t} \cdot 100$$



 Reproduce los tiempos en un diagrama vertical

## 2 Definiciones y métricas (V)

### ► Comutación de circuitos vs. paquetes



## 2 Definiciones y métricas (VI)

---



### ► Comutación de circuitos:

- Se establece *un circuito* antes de iniciar la transmisión
- El establecimiento va asociado con una reserva de recursos en los nodos
- Una vez establecido el circuito, todos los datos viajan por él
- Si se corta el circuito, la conexión se cierra y hay que reiniciarla
- *Calidad de servicio* (QoS) implícita en el circuito establecido
- *Control de admisión*: cuando se agotan los recursos, no se permiten más circuitos y no se admiten más conexiones
- *Comutación por circuito virtual*: uso de circuitos lógicos sobre comutación de paquetes [Tema 4]
- Ejemplos: **Spectrum monitoring. Standards**

## 2 Definiciones y métricas (VII)

### ➤ Comutación de paquetes:

- Comm. paquetes es más *tolerante a fallos* que comm. circuitos: ante fallos, los paquetes pueden ir por rutas alternativas [Tema 4]
- Suele implicar *más sobrecarga*: información de control por paquete y no por canal como en commutación de circuitos
- Ocupación del canal bajo demanda
  - Si no hay datos a transmitir, no se ocupa el canal
  - Funciona muy bien para tráfico en ráfagas (usual en Internet)

### 3 Control de acceso al medio (MAC)

- Un medio de transmisión suele ser compartido
- Si varios emisores transmiten a la vez, sus señales pueden colisionar (superponerse) y no ser recibidas correctamente
- El control de acceso al medio establece reglas de uso del canal para evitar/minimizar colisiones
- Tipos básicos de MAC:
  - *Protocolos de particionado del canal:* TDMA, FDMA y CDMA
  - *Protocolos de acceso aleatorio:* Aloha y CSMA
  - *Protocolos “por turnos”:* consulta y paso de testigo

## 4 Protocolos de particionado de canal

---

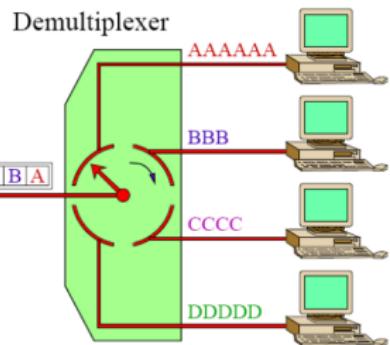
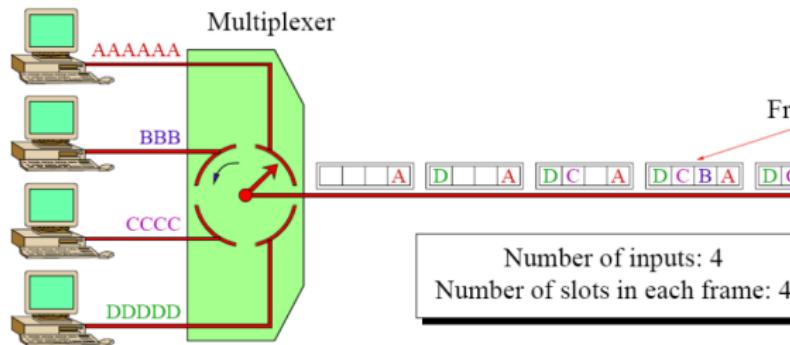
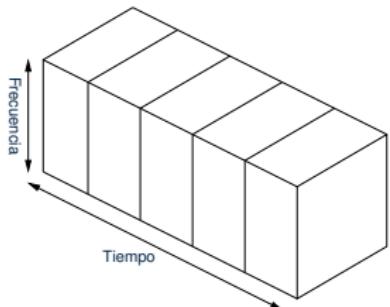


- Dividen el medio compartido en “trozos” más pequeños sin solapamientos
- Cada “trozo” se reserva para un usuario específico o un par de usuarios, eliminando las colisiones por completo

# 4.1 TDMA

*Time Division Multiple Access: acceso múltiple por división en tiempo (TD)*

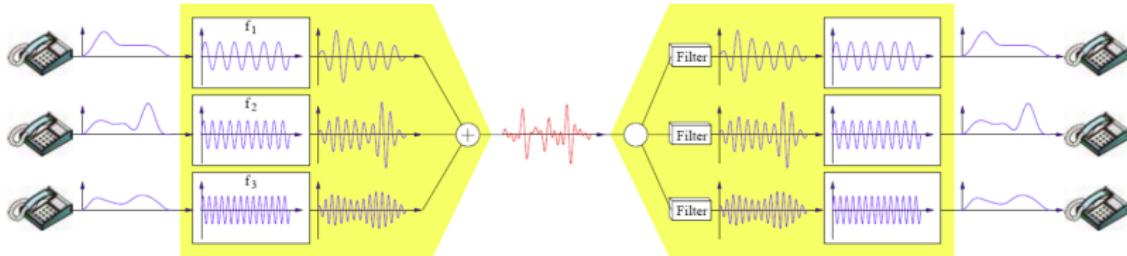
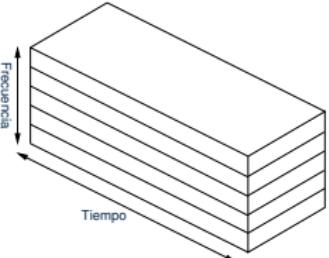
- Necesita *buffers*
- Necesita sincronización
- Fácil jerarquización de canales
- E.g. GSM (Global System for Mobile communications) y Bluetooth



## 4.2 FDMA

*Frequency Division Multiple Access: acceso múltiple por división en frecuencia (FD)*

- Señales moduladas con portadoras a distinta frecuencia (sin solapar)
- E.g. Bluetooth: 79 canales 1 MHz, 2.402-2.480 GHz, ADSL (Asymmetric Digital Subscriber Line): 1 canal voz,  $n$  canales uplink,  $m(> n)$  downlink



## 4.2.1 WDMA

*Wavelength Division Multiple Access:* acceso múltiple por división en longitudes de onda (WD)

- División en frecuencia en comunicaciones ópticas

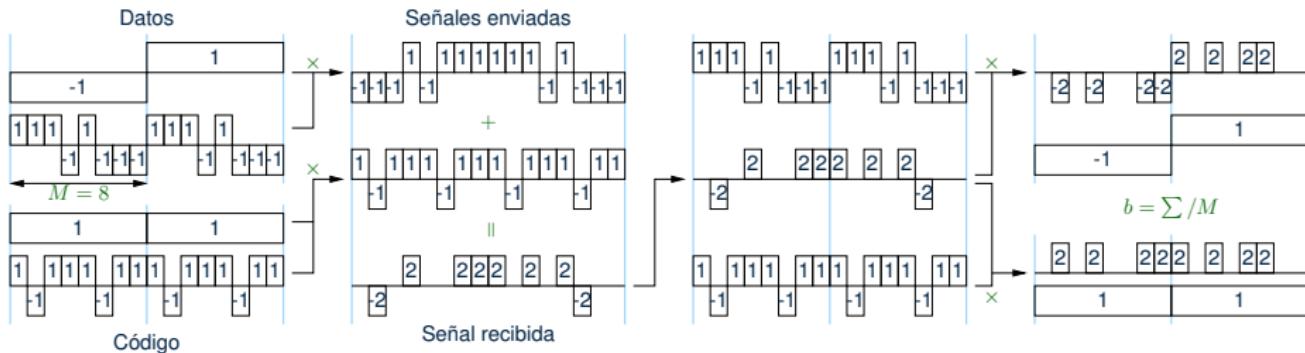
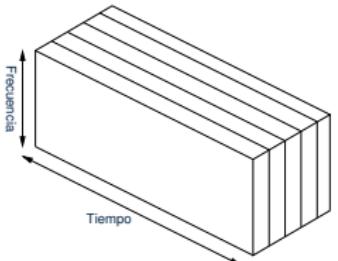


Fuente: Commscope. Data Center Best Practices

# 4.3 CDMA

*Code Division Multiple Access: acceso múltiple por división de código (CD).*

- ▶ Código de espectro ensanchado
- ▶ Usado en comunicaciones inalámbricas
- ▶ Mayor tolerancia ante interferencias
- ▶ E.g. GPS (Global Positioning System),  
UMTS (Universal Mobile Telecommunication System)



## 5 Protocolos de acceso aleatorio

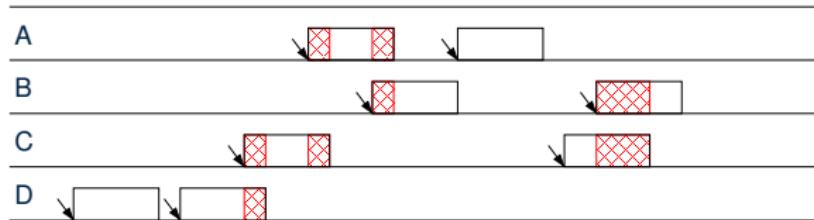
---

- Permiten que una estación transmita cuando quiera
- Las colisiones son posibles!
- Se basa en protocolos que detectan y se recuperan de las colisiones

# 5.1 Aloha

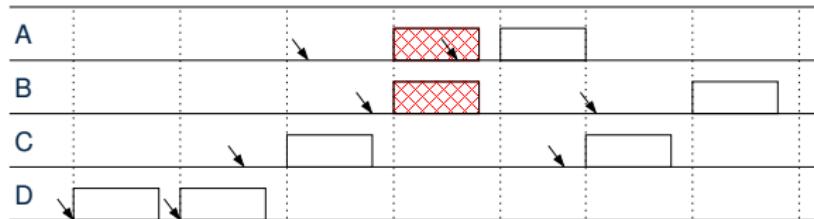
Pensado para comunicaciones de radio entre islas

- Aloha puro: pueden iniciarse transmisiones en cualquier momento. Si no hay colisión:  $T_a = 0$



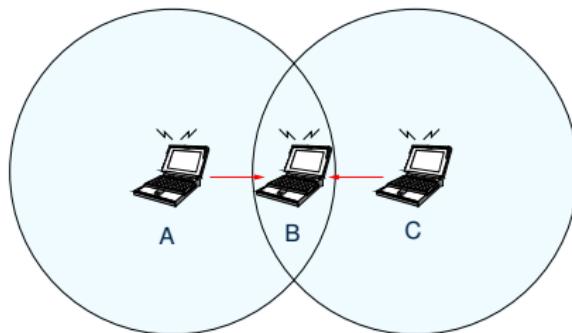
- Aloha ranurado: sólo pueden iniciarse transmisiones al inicio de ranuras/slots periódicas. Si no hay colisión:

$$T_a = [0, T_{slot}]$$



## 5.2 CSMA/CA

*Carrier Sense Multiple Access / Collision Avoidance*



Procedimiento emisor

1. Escucha el canal hasta que esté libre +
2. Espera un tiempo breve (*interframe gap*) y transmite
3. Espera confirmación de recepción (trama ACK)
4. Si no recibe ACK (colisión en el receptor por nodo oculto), espera un tiempo aleatorio y vuelve al paso 1

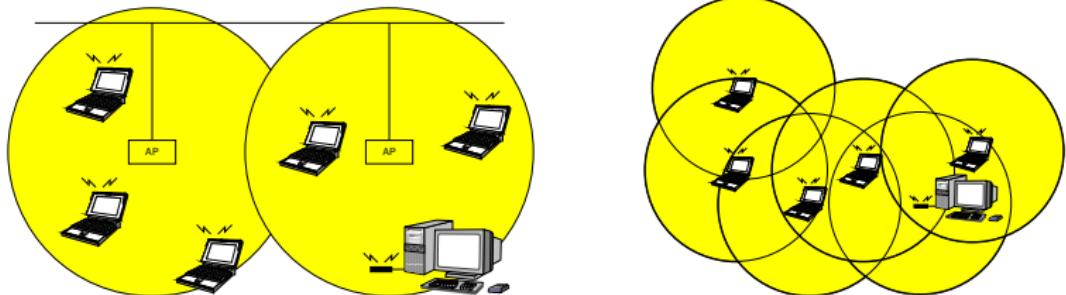
## 5.2 CSMA/CA: *backoff exponencial*

---

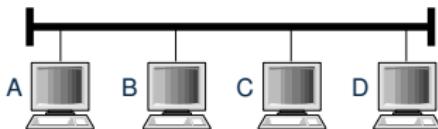
- Si el *tiempo de espera* fuera determinista → nueva colisión
- Siendo aleatorio se reduce la probabilidad de colisión
- Se adapta aleatoriedad a probabilidad de colisión
  - Empezar asumiendo poca probabilidad
  - Ante colisiones consecutivas, aumentar exponencialmente el intervalo de tiempo máximo
- $TiempoEspera = backoff \cdot T_{slot}$ . E.g.  $T_{slot} = 9 \mu s$  en 802.11n
- Ejemplo:
  - 1 col:  $backoff \in \{0, 1\}$
  - 2 col:  $backoff \in \{0, 1, 2, 3\}$
  - 3 col:  $backoff \in \{0, 1, 2, 3, 4, 5, 6, 7\}$
  - 4 col:  $backoff \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$
  - ...
- Inconveniente: efecto LIFO (*last-in-first-out*)

## 5.2 CSMA/CA: uso en 802.11

- Puntos de acceso (access point, AP) 
  - Los APs conforman una red ya desplegada
  - Los nodos se conectan a uno de los APs
  - Los APs proporcionan configuración dinámica de red a los nodos (DHCP)
  - Los nodos envían/reciben siempre a través del AP
- Redes ad-hoc
  - No hay una red desplegada a la que conectarse
  - Los nodos se autoorganizan funcionando como retransmisores



## 5.3 CSMA/CD



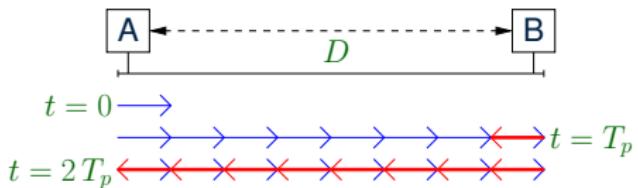
*Carrier Sense Multiple Access with Collision Detection*

Procedimiento emisor

1. Escucha el canal hasta que esté libre
  2. Espera un tiempo breve (*interframe gap*) y transmite
  3. Si detecta colisión, sustituye la transmisión por una señal corta (32 bits) de alerta (*jam*) para enfatizar la colisión
  4. Después de enviar la señal de alerta, espera un tiempo aleatorio y vuelve al paso 1
- No hay ACK → *el emisor debe detectar colisiones*
  - *Backoff* igual que en CSMA/CA, con  $T_{slot} \approx T_t$  (*trama min*)
  - Usado en IEEE 802.3 Ethernet

## 5.3.1 Detección de colisiones

El emisor debe detectar *siempre* toda colisión en sus envíos



- Cuando el emisor transmite, comprueba que la señal en el canal coincide con la que envía
- Hay que transmitir al menos  $T_t = 2 \cdot T_p$  para detectar colisión en el punto más lejano, con  $T_p = D/V_p$  y  $T_t = L/V_t$
- Por tanto:  $T_t = L/V_t \geq 2 \cdot T_p = 2 \cdot D/V_p$ 
  - Si se fija  $T_{p_{max}}$ , entonces  $L_{min} = 2 \cdot T_{p_{max}} \cdot V_t$
  - Por ejemplo, en Ethernet sobre par trenzado:  
 $T_{p_{max}} = 25.6 \mu s, V_t = 10 Mbps, V_p = 177000 km/s$   
 $\rightarrow L_{min} = 512 bits = 64 bytes$  y  $D_{max} = 4531 m$

## 5.3.1 Detección de colisiones: ejercicio



✍ Se desea aumentar la velocidad de transmisión Ethernet a  $V_t = 100 \text{ Mbps}$ .

1. Si se mantiene el tiempo máximo de propagación,  $T_{p_{max}} = 25.6 \mu\text{s}$ , ¿qué ocurre con el tamaño mínimo de trama Ethernet  $L_{min}$ ?
2. Si se mantiene el tamaño mínimo de trama,  $L_{min} = 64 \text{ bytes}$ , ¿qué ocurre con el tiempo máximo de propagación  $T_{p_{max}}$ ? ¿Y con la distancia máxima  $D_{max}$ ?

## 5.3.2 Ejemplo trama

Ejemplo: Trama Ethernet II (64–1518 bytes + preámbulo)

7	1	6	6	2	46–1500	4
Preámbulo	SFD	Dir. destino	Dir. origen	Tipo	Datos	FCS

Preámbulo: 7 x 10101010

SFD: Start-of-Frame-Delimiter: 10101011

Direcciones: identificadores MAC de las tarjetas de red origen y destino. Por ejemplo, para lab000: 50:65:f3:42:43:37<sup>i</sup>

Tipo: identificador<sup>i</sup> del protocolo encapsulado en el campo «Datos» (demultiplexor)

FCS: Frame Check Sequence, CRC de 32 bits [p. 54]



# 6 Procolos “por turnos”

---

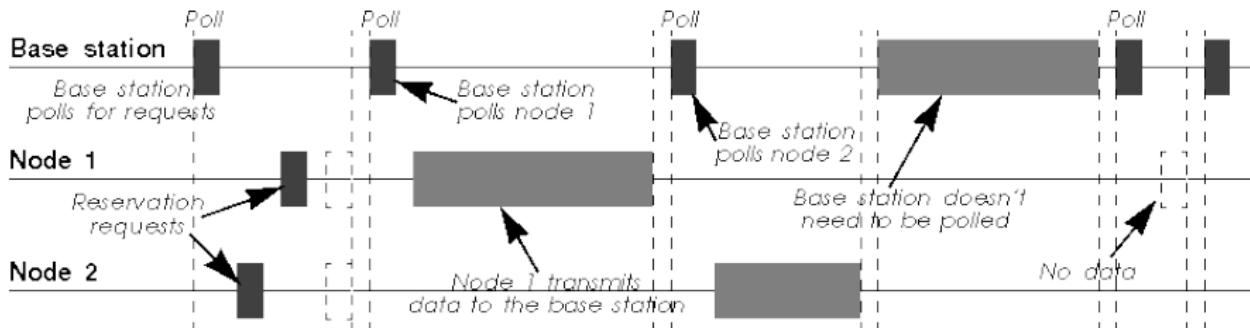


- Pensados con las ventajas de los dos anteriores: evitar la colisión sin la ineficiencia de las reservas cuando los usuarios están *idle*

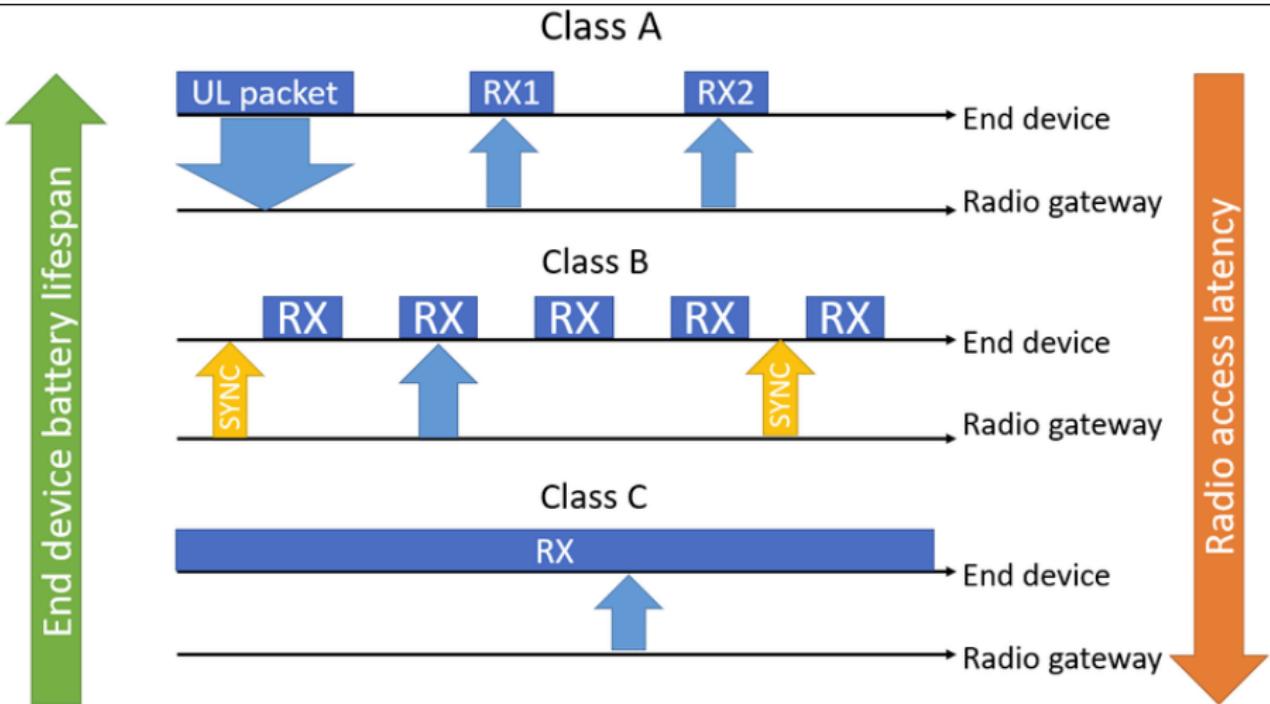
# 6.1 Polling



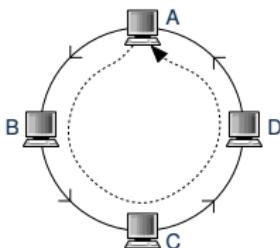
- Se basa en una estación que actúa como controladora y el resto responden
- Funcionamiento: la controladora consulta/invita a cada estación. Si tiene datos los transmite y si no, lo indica
- Es muy eficiente cuando hay mucha carga
- Es vulnerable si falla la controladora



## 6.1.1 Ejemplo LoRa



## 6.2 Token Ring (IEEE 802.5)



- Estaciones conectadas en anillo unidireccional
- Cada estación propaga las tramas que recibe (con 1 bit de retardo para procesar bits particulares)
- Una trama especial (*testigo/token*) circula por el anillo
- Cuando se recibe el token, se puede capturar o propagar
- Para transmitir hay que estar en posesión del *token*:
  - Esperar a que llegue el *token* y capturarlo
  - Transmitir la trama de datos en un sentido del anillo
  - Quitar la trama de datos cuando llegue por el otro lado
  - Transmitir el *token*

## 6.2 Token Ring (IEEE 802.5) (II)



- Transmisiones por turnos → no hay colisiones
- No hay tamaño máximo de trama,  
sino tiempo máximo de posesión del *token*
- El *token* no se puede transmitir antes de que el primer bit  
de la trama de datos haya dado la vuelta
- Permite usar prioridades
  - Mensajes más prioritarios se transmiten antes
- Control distribuido del anillo. Todas las estaciones deben  
acordar qué estación monitoriza el funcionamiento:
  - que el turno vaya pasando
  - que ninguna trama de datos se quede dando vueltas  
indefinidamente
  - que no se «atasque» la prioridad
  - que quien monitoriza el funcionamiento no se bloquee

## 6.2.1 Ejemplo tramas

Tramas de token y datos en Token Ring (IEEE 802.5)

1	1	1								
SD	AC	ED								
1	1	1	2-6	2-6	n	4	1	1		
SD	AC	FC	Dir. destino	Dir. origen	Datos	FCS	ED	FS		

SD: Starting Delimiter

AC: Access Control, PPPTMRRR

(P: prioridad, T: token, M: monitor, R: reserva)

FC: Frame Control, tipo trama (demultiplexor) y bits de control

Direcciones: identificadores MAC origen y destino

FCS: Frame Check Sequence, CRC de 32 bits [p. 54]

ED: Ending Delimiter

FS: Frame Status, ACrrACrr

(A: dir. reconocida, C: trama copiada, r: reservado)

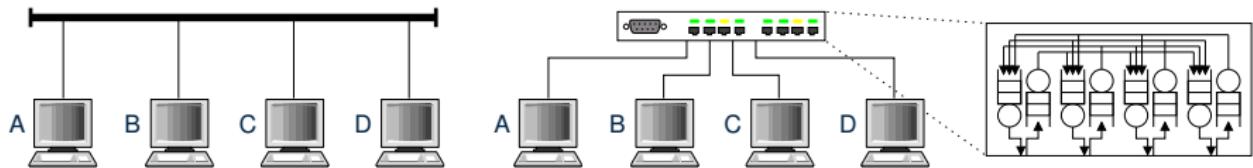
## 7. Comutación en Ethernet

- 7.1. Conmutadores aprendices
- 7.2. LANs virtuales (VLANs)
- 7.3. Protocolo Spanning Tree
- 7.4. Ejercicio resumen commutación
- 7.5. Estándares Ethernet

# 7 Conmutación en Ethernet



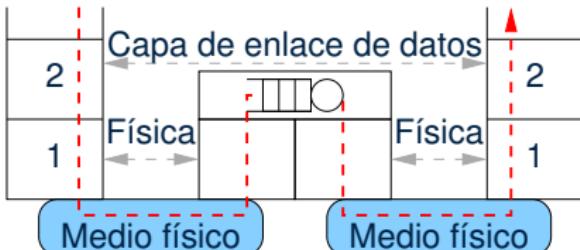
- LANs Ethernet (CSMA/CD) fueron ganando terreno
  - Fácil mantenimiento, buenas prestaciones con baja carga
- Más dispositivos en medio compartido → rendimiento *no escala* → menos prestaciones
- Solución: conmutador (*switch*)  que implementa conjunto de enlaces no compartidos



- Cada nodo está conectado al conmutador con un *cable dedicado* → todos pueden transmitir a la vez → *n* veces más capacidad pico

# 7 Comutación en Ethernet (II)

- Cambio de comunicación directa por medio compartido a indirecta a través de conmutador



- El conmutador es *transparente* para los dispositivos
- El conmutador almacena y reexpide las tramas (*store & forward*) o las reenvía al vuelo (*cut-through*)
- ✖ *Congestión*: pérdida de tramas por falta de memoria en conmutador [Tema 6]
- ✖ *Contención*: latencias adicionales y variables en colas de conmutador
- ✖ El conmutador sólo une *redes del mismo tipo o muy similares*, e.g. AP conecta ethernet y WiFi

# 7 Ejemplo: switch L1.02

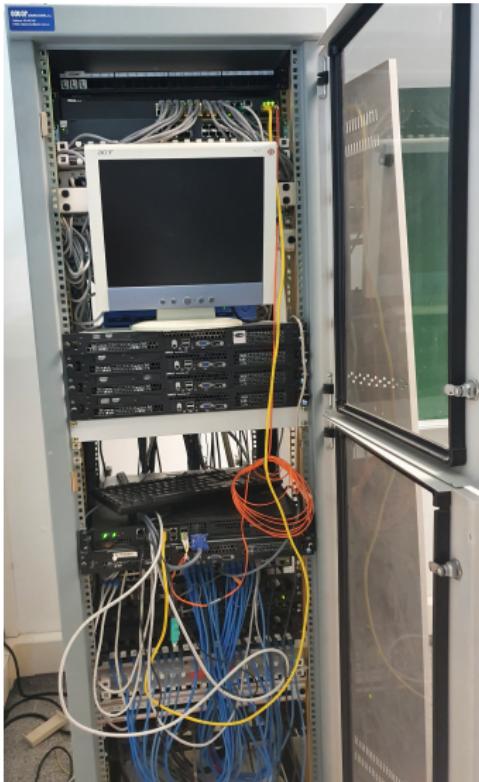


Imagen: rack que aloja equipos de comunicaciones y servidores del L1.02.



## Dell Networking N1524

- 24 puertos RJ-45 10/100/1000Mb con detección automática
- 4 puertos 10GbE SFP+ integrados
- 1 fuente de alimentación integrada (40 W de CA)

Figura: vista frontal y características básicas del switch Dell N1524



Imagen: detalle del cableado del switch Dell N1524

# 7 Ejemplo: switch L1.02 (II)

- ▶ Puerto RJ-45: conector para pares trenzados
- ▶ Puerto SFP+: conector para fibra óptica
- ▶ RJ45 (8P8C)

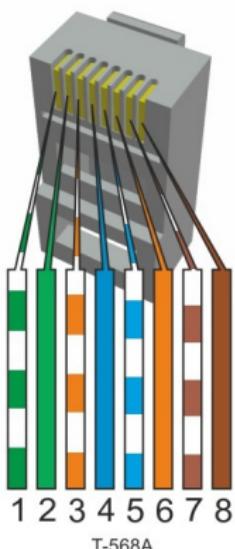


Figura: Conector TIA T568A.

Pin	Description	10base-T	100Base-T	1000Base-T
1	Transmit Data+ or BiDirectional	TX+	TX+	BI_DA+
2	Transmit Data- or BiDirectional	TX-	TX-	BI_DA-
3	Receive Data+ or BiDirectional	RX+	RX+	BI_DB+
4	Not connected or BiDirectional	n/c	n/c	BI_DC+
5	Not connected or BiDirectional	n/c	n/c	BI_DC-
6	Receive Data- or BiDirectional	RX-	RX-	BI_DB-
7	Not connected or BiDirectional	n/c	n/c	BI_DD+
8	Not connected or BiDirectional	n/c	n/c	BI_DD-

 PinoutsGuide.com

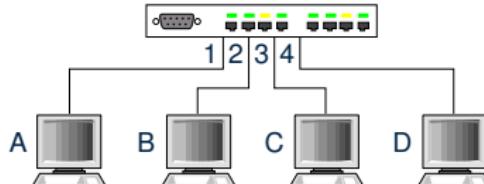
- ▶ SFP+: *small form-factor pluggable*
  - ▶ transceptor: emisor + receptor



Imagen: Transceptor, SFP+, 10GbE, LR, 1310 nm, 10 km. Fuente: [www.dell.com](http://www.dell.com).

## 7.1 Conmutadores aprendices

- Sin conmutadores aprendices, aunque no se comparta el canal, el tráfico es el mismo → *no escala*
- Los conmutadores aprendices ven tramas con identificador MAC de origen *x* que llegan por puerto *y*
- Mantienen tabla con parejas  $\langle x, y \rangle$  que expiran con el tiempo
- Al recibir tramas dirigidas a *x*, se reexpedirán sólo por *y*
  - Se evita tráfico innecesario
  - Se dificulta la monitorización del tráfico de otros
- Tramas con destinos no conocidos o de difusión total (*broadcast*) se reexpiden a todos



Estación	Puerto HW
A	1
B	2
C	3
D	4

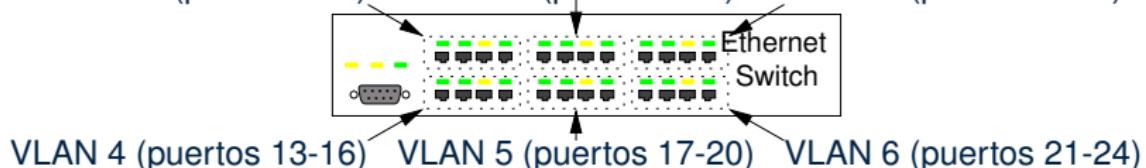
## 7.2 LANs virtuales (VLANs)



- Las tramas de difusión (*broadcast*) han de llegar a todos → no escala para redes muy grandes
- División de LAN en dominios de difusión (VLANs)
  - Tramas de difusión no se reexpiden a otras VLANs
  - Id. VLAN asociado a puertos o id. MAC
  - IEEE 802.1Q: modifica la cabecera ethernet (0x8100) para incluir campo id. VLAN (VLAN trunking)

...	6	6	4	2	...
	Dir. destino	Dir. origen	802.1Q	Tipo	

VLAN 1 (puertos 1-4)      VLAN 2 (puertos 5-8)      VLAN 3 (puertos 9-12)



## 7.3 Protocolo *spanning tree*



- Tolerancia a fallos → conmutadores y enlaces redundantes
- Bucles dan problemas (ejemplo: 1 2 3 2 3 2 3 ...)
  - Tramas duplicadas y dando vueltas indefinidamente:  
Tormenta de difusión (*broadcast storm*)
  - Conmutadores no pueden asociar dir. origen con puerto



- El objetivo del protocolo *spanning tree* (STP) es generar un conjunto de rutas libre de bucles
- Para ello, este protocolo distribuido crea dinámicamente una topología de árbol mediante el bloqueo de ciertos puertos
- Inconveniente: *algunas tramas no siguen el camino óptimo*

## 7.3 Protocolo *spanning tree* (II)



1474

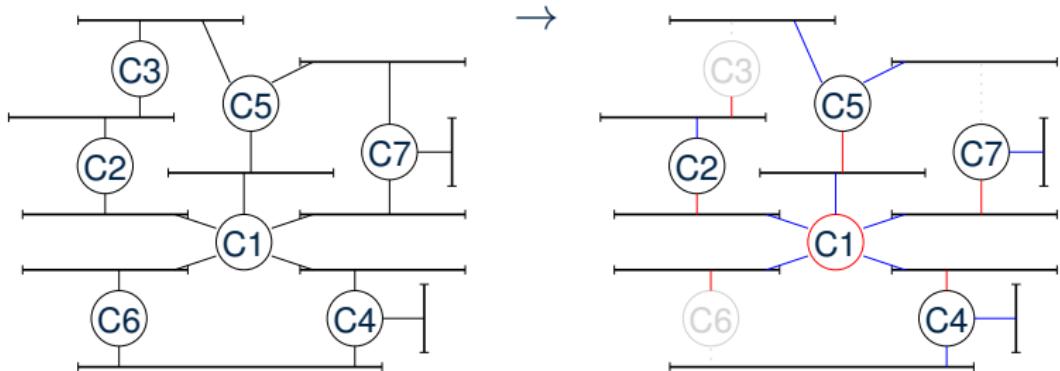
- Basado en algoritmo inventado por *Radia Perlman*<sup>i</sup>
  - 1985 - An Algorithm for Distributed Computation of a Spanning Tree in an Extended LAN<sup>i</sup>
  - 1990 - Protocolo estandarizado como 802.1D
  - Actualizaciones en 1998 y 2004
- Fases:
  1. Selección conmutador raíz
  2. Selección camino a conmutador raíz
- Resultado: todos los conmutadores aprenden
  - Cuál es su puerto más cercano al raíz
  - Si para cierto segmento están en el camino más corto hacia raíz

## 7.3 Protocolo *spanning tree* (III)



- Cada commutador tiene un id. obtenido concatenando un valor de prioridad (configurable) y su dir. MAC (única)
- Inicialmente, todos los commutadores creen ser raíz
- Quien cree ser raíz genera mensajes de configuración  $\langle id\_propio, id\_raiz, distancia\_a\_raiz \rangle$
- Quien aprende que no es raíz (recibido  $id\_raiz < id\_propio$ ) no crea más mensajes, pero sí reexpide ( $distancia + 1$ ) los que llegan desde raíz
- Sólo el raíz genera mensajes de configuración. Si no se reciben durante cierto tiempo, se reinicia el proceso

## 7.3 Protocolo *spanning tree* (IV)



- Raíz: conmutador con el menor id. (único)
- Puerto raíz de cada conmut. (RP): el más cercano al conmut. raíz (empates → menor id.)
- Puerto designado de cada segmento de red (DP): el más cercano al conmut. raíz (empates → menor id.)
- ... Comunicadores bloquean sus puertos no-RP y no-DP

## 7.4 Ejercicio resumen conmutación



1474

Asocia cada concepto con su principal aportación:

Concepto	Aportación
Comutador	a) Reduce el tráfico unicast
Conmut. aprendiz	b) Reduce el tráfico broadcast
VLANs	c) Reduce colisiones
Spanning tree	d) Permite añadir redundancia

## 7.5 Estándares Ethernet

---



- Ethernet está estandarizado por el *Institute for Electrical and Electronics Engineers* (IEEE)
  - IEEE 802.3 es el estándar oficial de Ethernet
  - 1985 - IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications
- Identificadores cortos sistemas: velocidad, señalización, medio físico. Ejemplos:
  - 10BASE-T: 10 Mbps, banda base, dos pares de cable trenzado
  - 100BASE-FX: 100 Mbps, banda base, fibra óptica multimodo
  - 1000BASE-T: 1000 Mbps, banda base, pares trenzados
  - 10GBASE-SR: 10 Gbps, banda base, fibra multimodo de corto alcance

## 7.5 Ejemplo tarjeta red Ethernet

```
lab000:~/ ethtool eno1
Settings for eno1:
Supported ports: [ TP ]
Supported link modes:  10baseT/Half 10baseT/Full
                       100baseT/Half 100baseT/Full
                           1000baseT/Full
Supported pause frame use: No
Supports auto-negotiation: Yes
Supported FEC modes: Not reported
Advertised link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full
                           1000baseT/Full
Advertised pause frame use: No
Advertised auto-negotiation: Yes
Advertised FEC modes: Not reported
Speed: 1000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
MDI-X: off (auto)
Cannot get wake-on-lan settings: Operation not permitted
Current message level: 0x00000007 (7)
                         drv probe link
Link detected: yes
```

## 7.5 Power over Ethernet (PoE)



- Además de enviar datos, el cable Ethernet alimenta al equipo conectado al conmutador
- Usos: alimentación de puntos de acceso, sensores

Pin	Function
1	Tx+
2	Tx-
3	Rx+
4	DC+ PoE
5	DC+ PoE
6	Rx-
7	DC- PoE
8	DC- PoE

# 8 Subcapa control de enlace



- Función: establecer *comunicación fiable*



- *Fiabilidad*: los datos se reciben correctamente, ordenados, sin pérdidas y sin duplicados
- Las tramas recibidas pueden contener errores
  - *Control de errores*
- Se pueden perder tramas
  - *Secuenciación de datos*

Un protocolo ofrece un servicio fiable si detecta tramas:

- Erróneas o perdidas y las retransmite
- Duplicadas y las descarta
- Desordenadas y las ordena

## 9. Control de errores

### 9.1. Detección de errores

Códigos de paridad

Checksum

CRC

### 9.2. Corrección de errores

# 9 Control de errores

---



- Valores típicos de *bit-error-ratio* (BER) (tramas  $\approx 10^4$  bits):
  - Fibra óptica:  $\sim 10^{-12}$
  - Cable de cobre:  $\sim 10^{-9}$  (muy pocos errores)
  - Aire (medio inalámbrico):  $\sim 10^{-5}$  (1/10 tramas erróneas)
- Detección: paridad, checksum, CRC, SHA1, etc.
  - Añade redundancia: alteraciones  $\leftrightarrow$  inconsistencias
  - Sobrecarga «baja»
  - Tasa código: relación entre bits de datos y bits totales
  - *Ninguna detección es 100 % fiable*
- Ante error detectado:
  - *Descartar* trama  $\rightarrow$  no hay fiabilidad (e.g. Ethernet, IP, UDP)
  - *Retransmitir* trama  $\rightarrow$  necesita secuenciación (e.g. TCP)
  - *Corregir* error  $\rightarrow$  sobrecarga elevada (e.g. Hamming) pero interesante ante coste de retransmisión muy alto

## 9.1.1 Códigos de paridad simple

### Bit de paridad

- Cada bloque de  $k$  bits se envía con un bit de paridad:  
 $1010100\ P$
- Par ( $P = \oplus$ ): número par de unos:  $1010100\ 1$
- Impar ( $P = \overline{\oplus}$ ): número impar de unos:  $1010100\ 0$
- Detecta cualquier número impar de bits erróneos
- No corrige errores
- Tasa código:  $\frac{k}{k+1}$
- Tasa redundancia:  $\frac{1}{k+1}$

3 bit data			Message with even parity		Message with odd parity	
A	B	C	Message	Parity	Message	Parity
0	0	0	000	0	000	1
0	0	1	001	1	001	0
0	1	0	010	1	010	0
0	1	1	011	0	011	1
1	0	0	100	1	100	0
1	0	1	101	0	101	1
1	1	0	110	0	110	1
1	1	1	111	1	111	0

## 9.1.1 Códigos de paridad por bloques



- Toma el mensaje como una matriz de  $m$  bits de ancho
- Calcula una nueva fila de  $m$  bits de paridad vertical
- Detecta errores en ráfaga de longitud  $\leq m$ 
  - Longitud de ráfaga de error: número de bits (erróneos o no) entre dos bits erróneos (incluidos en la longitud)
- Código rectangular o código producto: paridad en 2 dimensiones. Caso particular donde cada fila de la matriz tiene a su vez un bit de paridad

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

a. Design of row and column parities

Código rectangular de detección de errores.

Fuente: <http://www.myreadingroom.co.in/notes-and-studymaterial/68-dcn/801-simple-parity-check-code.html>

## 9.1.2 Checksum

---

- Redundancia: bits obtenidos al sumar palabras de datos
- El receptor verifica si la suma es correcta
  - E.g. emisor:  $3 + 7 + 5 + 1 + 3 = 19$  (checksum)
  - Comprobación:  $3 + 7 + 5 + 1 + 3 - 19 = 0?$  ( $\neq 0$ : error)
- Toma el mensaje como secuencia de números de  $m$  bits
- Calcula una nueva fila de  $m$  bits como la suma de la secuencia de números
  - Cualquier tipo de suma sirve, aunque distintos tipos de suma tendrán distintas propiedades de detección
  - Paridad por bloques: caso particular de checksum con  $\oplus$  bit a bit como operación de suma
  - TCP/IP: suma de enteros de 16 bits en complemento a 1 con inversión de bits en el resultado
- Detecta errores en ráfaga de longitud  $\leq m$

## 9.1.3 Cyclic Redundancy Check (CRC)



- Suma tiene limitaciones para detectar errores:

	mensaje			checksum
Original	6	23	4	33
Recibido	8	20	5	33

- Bits de redundancia obtenidos mediante división permiten detectar más errores
- Ejemplo: letra DNI = número DNI módulo 23  
(T R W A G M Y F P D X B N J Z S Q V H L C K E)
- CRC: resto de la división entera entre mensaje y un divisor
- Aritmética polinomial
  - Operaciones módulo 2:  $+ \equiv - \equiv \oplus$
  - Un divisor cabe en un dividendo si éste tiene tantos bits como el divisor:  $10 \div 11 = 1, \text{resto } 1$
  - Un polinomio es de grado  $n$  si tiene  $n + 1$  bits y su bit de más peso es 1

## 9.1.3 Cyclic Redundancy Check (CRC) (II)



- Emisor: dado un mensaje  $M$  y un polinomio generador  $G$  de grado  $n$ , genera un polinomio  $T$  divisible por  $G$ 
  - $CRC = (M \times 2^n) \text{ mód } G$  (añade  $n$  0s, divide y coge resto)
  - $T = M \times 2^n + CRC$  (concatena  $M$  y  $CRC$ )
- Receptor:  $R = T + E$ ,  
asume mensaje correcto si:  $R \text{ mód } G = 0$
- Detecta:
  - Errores en ráfaga de longitud  $\leq n$
  - Número impar de errores si  $G$  es múltiplo de  $x + 1$   
(ningún polinomio  $E$  con un número impar de términos es múltiplo de  $x + 1$ )
- Algunos  $G$  populares:
  - 16 bits: X25: (16,12,5,0), CRC-16: (16,15,2,0)
  - 32 bits: Ethernet: (32,26,23,22,16,12,11,10,8,7,5,4,2,1,0)
- Más información en Tanembaum y en este [enlace](#).

### 9.1.3 Cyclic Redundancy Check (CRC) (III)



*M*: 1101011011

$$(x^9 + x^8 + x^6 + x^4 + x^3 + x + 1)$$

$$G: 10011 \ (x^4 + x + 1)$$

*n*: 4

*CRC:* 1110 ( $x^3 + x^2 + x$ )

*T*: 11010110111110

$$(x^{13} + x^{12} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^3 + x^2 + x)$$



## 9.2 Corrección de errores

---

**Forward Error Correction (FEC):** envío de información con suficiente redundancia para que el receptor pueda corregir errores de transmisión

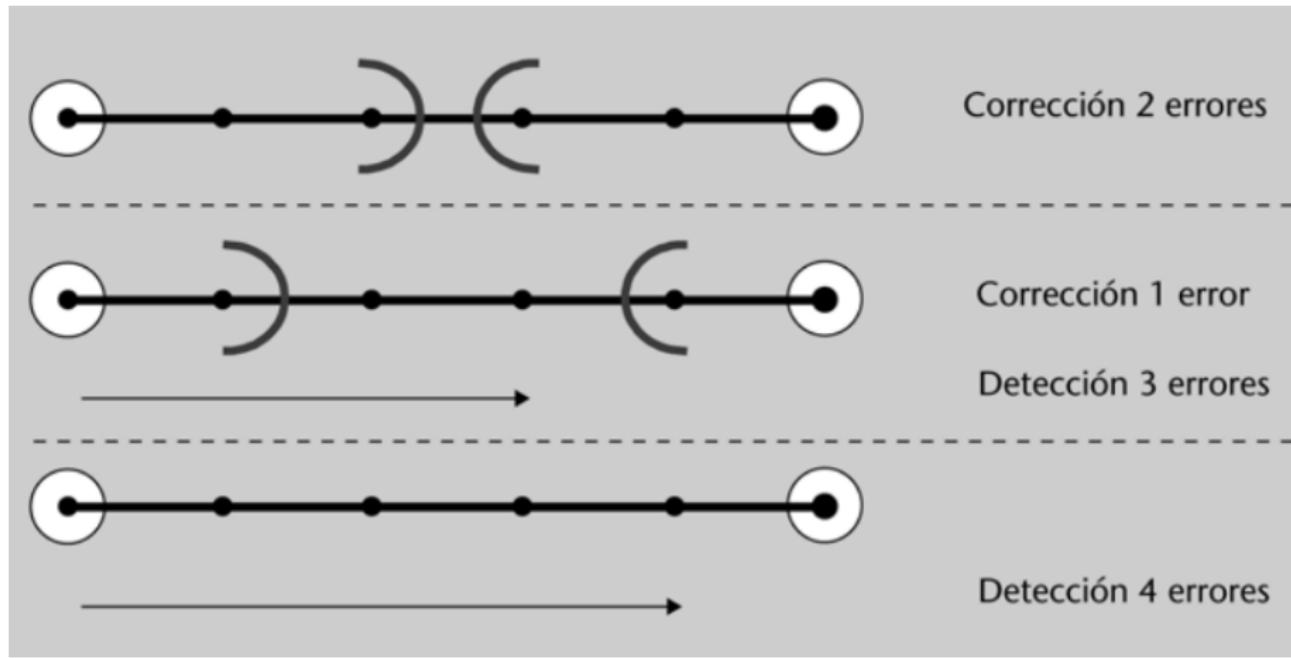
**Código de bloque:** asigna una palabra o vector de  $n$  bits a cada uno de los  $2^k$  posibles mensajes de  $k$  bits

Mensaje ( $k=2$ bits)	Palabra código ( $n=10$ bits)
00	0000000000
01	0000011111
10	1111100000
11	1111111111

**Distancia de Hamming:** número mínimo de bits que deben cambiar en una palabra código para convertirse en otra

## 9.2 Corrección de errores (II)

- Posibilidades de corrección/detección de errores con un código de distancia mínima 5



Fuente: Codificación de canal I: introducción y códigos de bloque. Francesc Tarrés y Margarita Cabrera.

## 9.2 Corrección de errores (III)

- La corrección de errores se basa en establecer palabras código lo suficientemente distantes como para que:
  1. Si hay error, se reciba una palabra código no válida
  2. Al recibir una palabra código no válida, la más cercana en distancia Hamming se considera la transmitida
- Con distancia  $d$  se pueden corregir  $\lfloor(d - 1)/2\rfloor$  bits
- Ejemplo: 0000000000, 0000011111, 1111100000, 1111111111
  - Distancia de Hamming: 5
  - Puede corregir 2 bits, e.g. 0000010011 → 0000011111
- Ejemplos: códigos Hamming, códigos Golay (NASA Voyager 1 y 2), etc.



## 10. Secuenciación de datos

10.1. Stop & wait

10.2. Ventana deslizante

Go-Back-N

Selective Repeat

Números de secuencia y tamaños de ventana

10.3. Ejemplo: HDLC

10.4. Ejercicios resumen de secuenciación

# 10 Secuenciación de datos

Contexto: protocolos que ofrecen servicio fiable deben recuperar tramas erróneas o perdidas.

Generalmente, esta funcionalidad se implementa mediante algoritmos *ARQ* (*Automatic Repeat reQuest*), que se basan en:

- *Acuse de recibo (ACK)*: trama de control enviada por el receptor de una trama para confirmar su recepción
  - *Piggyback*: ACK en trama de datos
- *Temporizador*: tiempo de espera antes de retransmitir una trama sin ACK

Algoritmos ARQ se usan en protocolos de

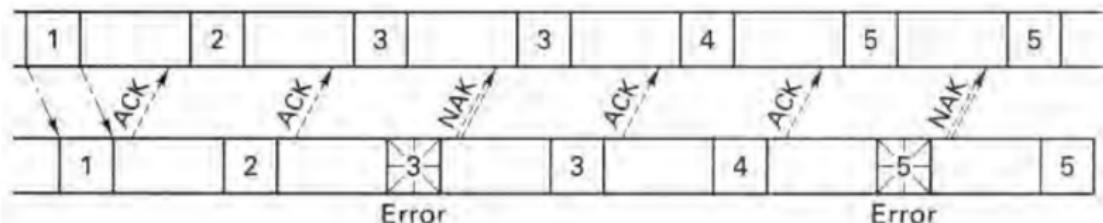
- Nivel de enlace: enlace lógico punto-a-punto
- Nivel de transporte: enlace lógico extremo-a-extremo

# 10 Secuenciación de datos (II)

Algoritmos ARQ:

- *Parada y espera (stop & wait)*: sencillo
- *Ventana deslizante*
  - *Vuelta atrás (go-back-n)*
  - *Repetición selectiva (selective repeat)*: más eficiente

# 10.1 Stop & wait



Fuente: Sklar. Digital Communications

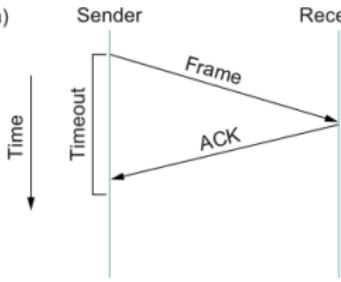
- Emisor envía trama  $i$ , inicia temporizador y espera ACK
  - Si recibe ACK de trama  $i \rightarrow$  parar temporizador e iniciar transmisión de trama  $i + 1$
  - Si vence el temporizador o recibe NAK  $\rightarrow$  reenviar trama
- Receptor espera trama con identificador  $i$ 
  - Trama  $i$  correcta  $\rightarrow$  enviar ACK de  $i$  y esperar trama  $i + 1$
  - Trama  $i$  errónea  $\rightarrow$  nada o enviar NAK de  $i$  (negative ACK)
- Puede funcionar con half-duplex

Para transmitir un número ilimitado de tramas, ¿cuántos identificadores de trama se necesitan? ¿Cuántos bits?

# 10.1 Stop & wait (II)

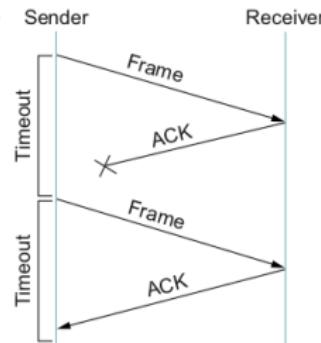
► 4 escenarios:

(a) Sender



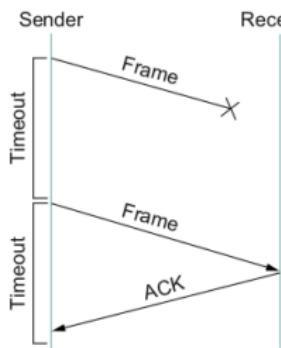
Receiver

(c) Sender

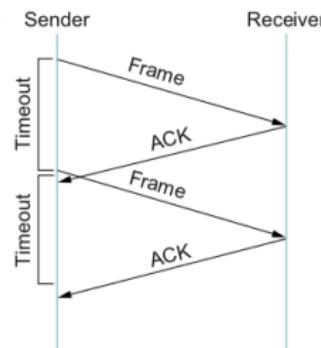


Receiver

(b) Sender



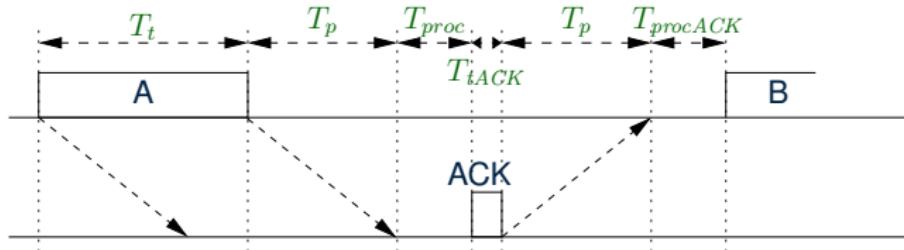
(d) Sender



# 10.1 Stop & wait (III)

Tiempo de ida y vuelta / *Round Trip Time* (RTT): tiempo necesario para enviar una trama y recibir su confirmación (segundos, s)

- RTT sin errores: transmisión  $T_t$  + propagación  $T_p$  + procesamiento  $T_{proc}$  de trama y ACK
- $RTT = T_t + T_p + T_{proc} + T_{tACK} + T_p + T_{procACK} = L_t/V_t + D/V_p + T_{proc} + L_{ACK}/V_t + D/V_p + T_{procACK}$



- Habitualmente:  $T_{proc} \approx T_{procACK} \approx 0$ ,  $T_t \gg T_{tACK} \approx 0$ :  
 $RTT = T_t + 2 \cdot T_p$

## 10.1 Stop & wait (IV)

- Utilización del enlace:

$$U = 100 \cdot \frac{T_t}{RTT} = 100 \cdot \frac{1}{1 + 2 \cdot T_p/T_t} \leq 100\%$$

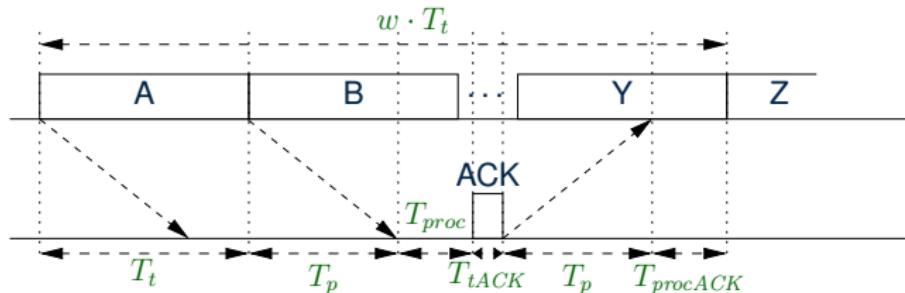
- ¿Y si viajan tramas de datos y ACK en ambos sentidos?

☞ Calcula la utilización de los siguientes enlaces:

- $T_t = 1 \text{ ms}, T_p = 5 \mu\text{s}$
- $T_t = 1 \text{ ms}, T_p = 50 \mu\text{s}$
- $T_t = 1 \text{ ms}, T_p = 500 \mu\text{s}$
- $T_t = 1 \text{ ms}, T_p = 5 \text{ ms}$
- $T_t = 1 \text{ ms}, T_p = 50 \text{ ms}$

## 10.2 Ventana deslizante

- ▶ *Stop & wait* es poco eficiente cuando no se cumple  $T_t \gg T_p$
- ▶ Ventana deslizante envía  $w$  tramas sin esperar el primer ACK:



- ▶ Utilización del enlace:

$$U(\%) = \begin{cases} 100 & \text{si } w \cdot T_t \geq RTT \rightarrow w \geq 1 + 2 \cdot \frac{T_p}{T_t} \\ 100 \cdot \frac{w \cdot T_t}{RTT} & \text{en caso contrario} \end{cases}$$

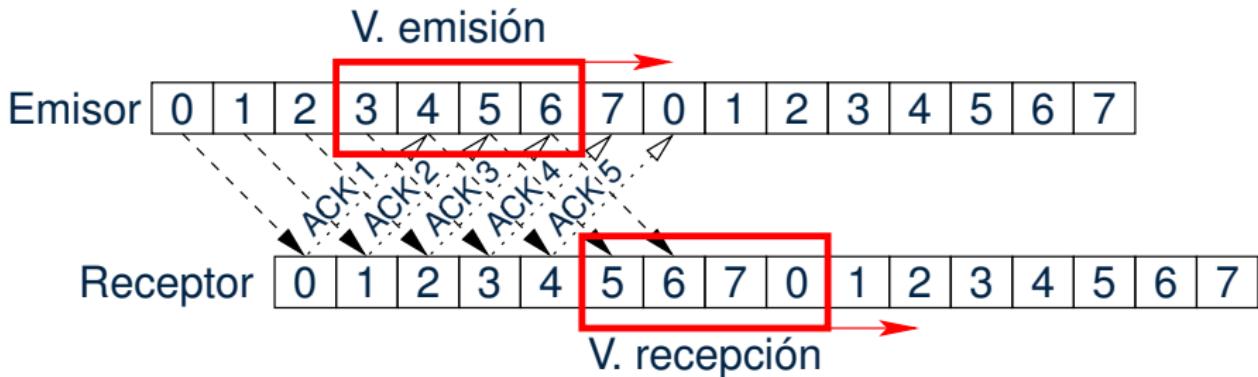
- ▶ Requiere canal full-duplex

## 10.2 Ventana deslizante (II)

✍ Calcula el mínimo tamaño de ventana deslizante que permite utilizar el 100 % de los siguientes enlaces:

- ▶  $T_t = 1 \text{ ms}, T_p = 5 \mu\text{s}$
- ▶  $T_t = 1 \text{ ms}, T_p = 50 \mu\text{s}$
- ▶  $T_t = 1 \text{ ms}, T_p = 500 \mu\text{s}$
- ▶  $T_t = 1 \text{ ms}, T_p = 5 \text{ ms}$
- ▶  $T_t = 1 \text{ ms}, T_p = 50 \text{ ms}$

## 10.2 Ventana deslizante (III)



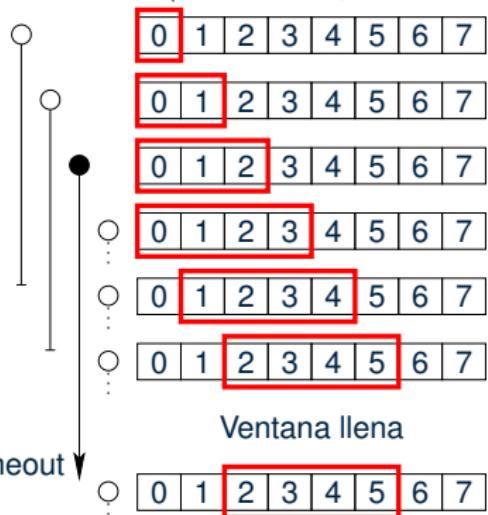
- El emisor asigna a cada trama un *número de secuencia*
- El emisor mantiene una *ventana de emisión* con las  $w$  tramas pendientes de confirmación
- El receptor mantiene una *ventana de recepción* para las  $w_r$  tramas que está dispuesto a aceptar
- $ACK_i$  indica que se ha recibido la trama  $i - 1$  y se espera la trama con número de secuencia  $i$

## 10.2.1 Go-Back-N

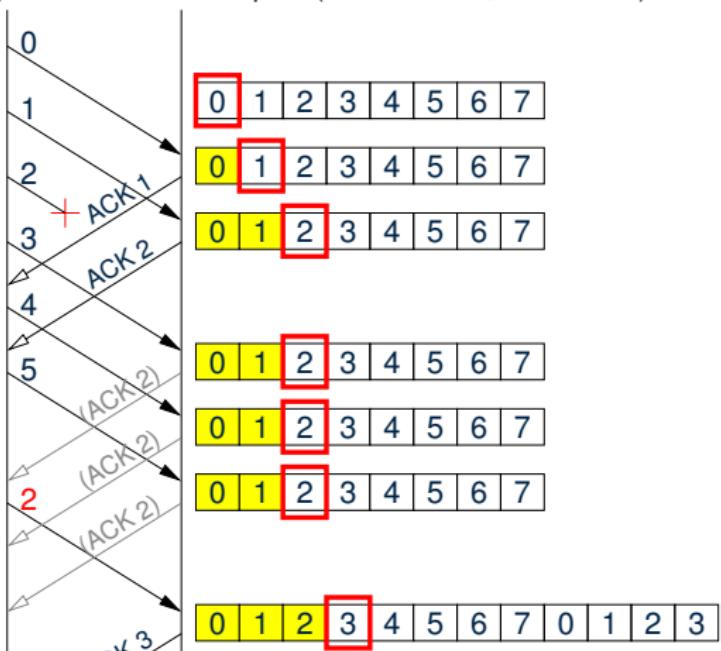


1474

Emisor (Núm. sec. 8, ventana 4)



Receptor (Núm. sec. 8, ventana 1)



Timeout

## 10.2.2 Selective Repeat

Emisor (Núm. sec. 8, ventana 4)

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Ventana llena

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Timeout

0

1

2

3

4

5

6

7

+ ACK1

ACK2

ACK2

ACK2

ACK2

ACK6

Receptor (Núm. sec. 8, ventana 4)

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

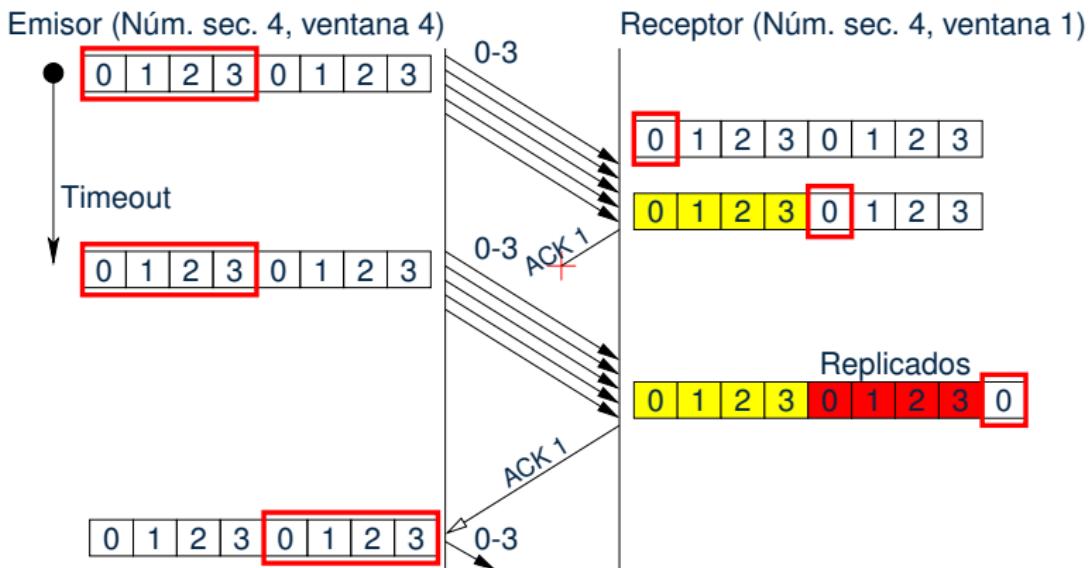
0	1	2	3	4	5	6	7	0	1	2	3
---	---	---	---	---	---	---	---	---	---	---	---

6-1

0	1	2	3	4	5	6	7	0	1	2	3
---	---	---	---	---	---	---	---	---	---	---	---

## 10.2.3 Núm. secuencia y tamaño vent.

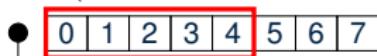
- Se necesitan  $NS$  números de secuencia:  $NS \geq w + w_r$
- Go-back-n:  $NS = w + 1$ ,  $n = \lceil \log_2(w + 1) \rceil$  bits
- Contraejemplo:



## 10.2.3 Núm. secuencia y tamaño vent. (II)

- Se necesitan  $NS$  números de secuencia:  $NS \geq w + w_r$
- *Selective Repeat*:  $NS = w + w_r$ ,  $n = \lceil \log_2(w + w_r) \rceil$  bits
- Contraejemplo:

Emisor (Núm. sec. 8, ventana 5)



Timeout



Receptor (Núm. sec. 8, ventana 5)



Sustituidos



## 10.3 Ejemplo: HDLC

ISO High-Level Data Link Control (6 ó 7 bytes + datos)

1	1	1 ó 2	n	2	1
01111110	Dirección	Control	Datos	FCS	01111110

**Dirección:** id. secundaria (no usada en punto-a-punto)

**Control (8 bits):** número secuencia (3 bits), núm. sec. esperada (3 bits), 2 bits control

**Control (16 bits):** número secuencia (7 bits), núm. sec. esperada (7 bits), 2 bits control

**FCS:** CRC de 16 bits

## 10.4 Ejercicio resumen de secuenciación



☞ Completa la siguiente tabla de tamaños de ventana.

	V. Emisión	V. Recepción
Stop&Wait		
Go-Back-n		1
Selective Repeat		$w_r$

☞ Queremos implementar un protocolo para la secuenciación de datos en capa de enlace con 3 bits para el número de secuencia. ¿Es posible un protocolo libre de fallos en los siguientes casos?

- ▶ Ventana de emisión 2 y ventana de recepción 6
- ▶ Ventana de emisión 1 y de recepción 8
- ▶ Ventana de emisión 5 y de recepción 5

# **Redes de Computadores**

## **Tema 4 – Capa de red**

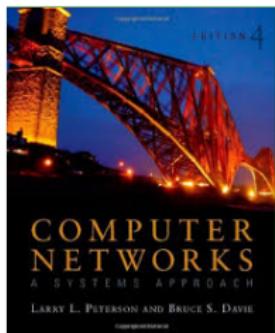
**Natalia Ayuso, Juan Segarra y Jesús Alastruey**



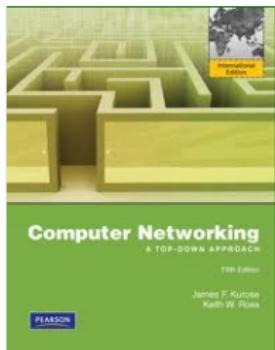
Departamento de  
Informática e Ingeniería  
de Sistemas

**Universidad** Zaragoza

1. Introducción
2. Modelos en conmut. paquetes
3. Encaminadores
4. Protocolo IP
5. Direcciones IPv4
6. IP versión 6
7. Túneles
8. Protocolos de encaminamiento
9. Estructura de Internet



Capítulos 3.1, 4



Capítulo 4

# 1 Introducción

---

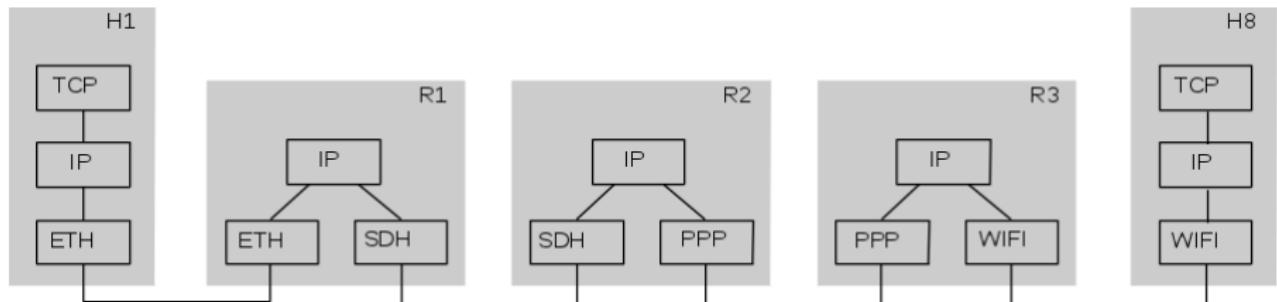
¿Por qué la capa 2 no sirve al aumentar el tamaño de la red?

- No puede interconectar redes físicas distintas
- Excesivo tamaño de tablas de encaminamiento en comutadores aprendices (requiere una entrada por cada nodo en la red)
- Crecimiento desordenado:  
*spanning tree* → rutas no óptimas
- Problemas con mensajes de difusión total (broadcast)

# 1 Introducción (II)

Solución: pasar de red «física» (capa 2) a redes lógicas interconectadas → capa de red (capa 3)

- Abstrae diferencias de funcionamiento de redes físicas
- Todas las redes deben tener un protocolo común para interconectarse: IP



Internet es un conjunto mundial de redes interconectadas con protocolos comunes (TCP/IP) y direccionamiento universal (IP)

### 2. Modelos en conmut. paquetes

- 2.1. Conmutación por datagrama
- 2.2. Encaminamiento fuente
- 2.3. Conmutación por circuito virtual
- 2.4. Multi Protocol Label Switching

## 2 Modelos en conmutación de paquetes

---

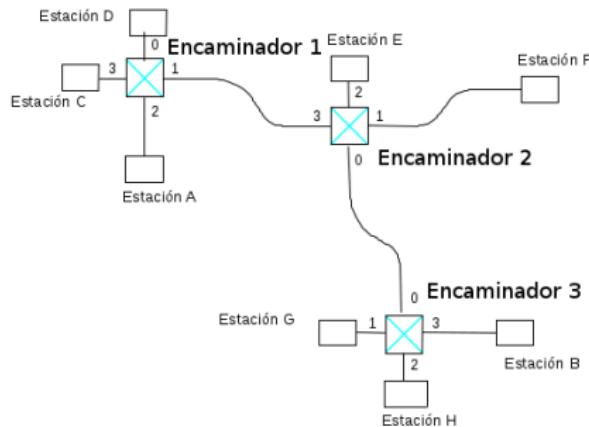


Dependiendo de la información usada para decidir el camino, existen varios modelos dentro de la conmutación de paquetes:

- *Comutación por datagrama*
  - En cada esquina preguntar a alguien por qué calle nos acercamos más al destino
- *Encaminamiento fuente*
  - Buscar la ruta y anotarla antes de iniciar el viaje para después seguirla
- *Comutación por circuito virtual*
  - Especificar un destino a una agencia e iniciar el viaje hasta destino con los trasbordos ya organizados
- Combinaciones de los modelos anteriores

## 2.1 Conmutación por datagrama

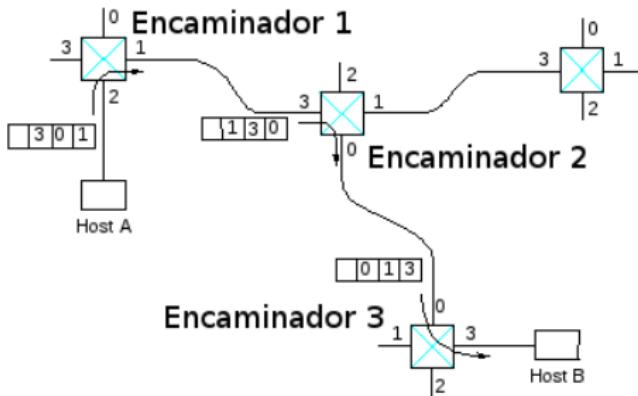
- Usado en Internet en el protocolo IP
- Cada paquete es encaminado de forma independiente
  - No hay establecimiento del camino
  - Paquetes con mismo origen-destino pueden ir por distintos caminos
  - Alta tolerancia a fallos
- Cada *encaminador* enruta en función de la dirección destino de cada paquete
  - Debe conocer el camino a ¡cualquier destino! [p. 66]
  - Mantiene una *tabla de encaminamiento* con esa info
  - Cada tabla de encaminamiento es distinta



## 2.2 Encaminamiento fuente



- El propio paquete lleva la ruta a seguir
  - El emisor debe conocer la topología de la red
  - La información de ruta (campo «siguiente puerto») se modifica en cada encaminador (rotación, punteros, etc.)
  - La cabecera tiene un tamaño variable sin límite



- El protocolo IP permite usar encaminamiento fuente añadiendo ciertas opciones en la cabecera de los paquetes

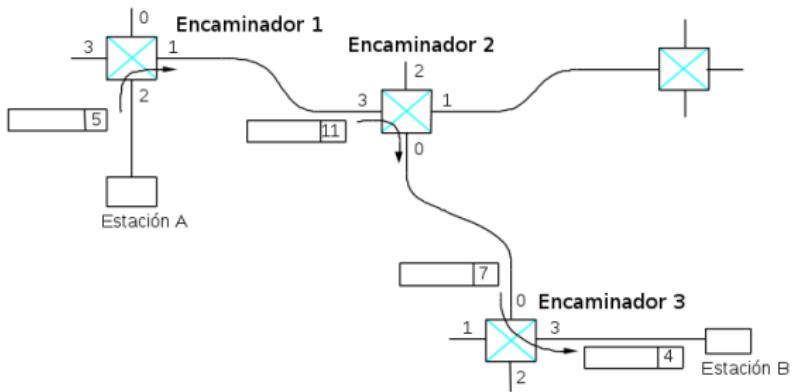
## 2.3 Comutación por circuito virtual

---



- Comutación de circuitos vía software
- Fase de creación (y destrucción) explícita de circuitos (paquetes especiales de ida y vuelta)
- Paquetes de datos se enrutan por circuito creado
- Cada encaminador mantiene *tabla de circuitos virtuales*
- Petición de conexión contiene dirección completa destino, pero cada paquete de datos solo un pequeño identificador
- Si un encaminador o un enlace de una conexión falla, la conexión se deshace y se necesita establecer una nueva
- Establecimiento de conexión proporciona oportunidad para reserva de recursos (QoS) [Tema 6]
- E.g. X.25, Frame relay, ATM

## 2.3 Conmutación por circuito virtual (II)



Después de establecer el circuito A-B:

1. A envía paquete con etiqueta de entrada a circuito A-B (5)
2. Encaminador 1 ve etiqueta 5 y enruta por puerto 1 con etiqueta 11 (tabla circuitos virtuales)
3. E2 ve etiqueta 11 y enruta por puerto 0 con etiqueta 7
4. E3 ve etiqueta 7 y enruta por puerto 3 con etiqueta 4

## 2.4 Multi Protocol Label Switching

- Sobre conmutación por datagrama, los encaminadores solicitan que para ciertos destinos los paquetes incorporen ciertas etiquetas
- Conmutación de etiquetas multiprotocolo
- Combinación de:
  - Circuitos virtuales (etiquetas cortas de longitud fija y ámbito local)
  - con flexibilidad y robustez de datagramas
- Características:
  - Necesita de protocolos encaminamiento IP para crear rutas
  - Es capaz de llevar cualquier protocolo de la capa de red
  - Situado conceptualmente entre capas 2 y 3

## 2.4 Multi Protocol Label Switching (II)

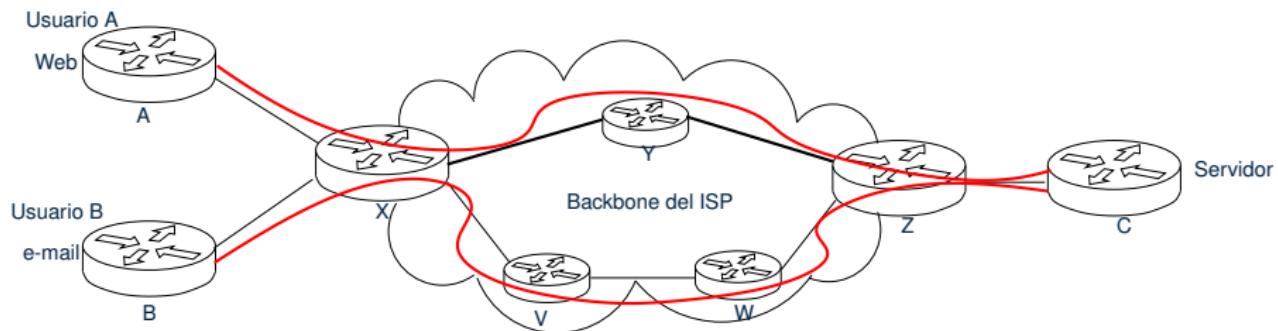
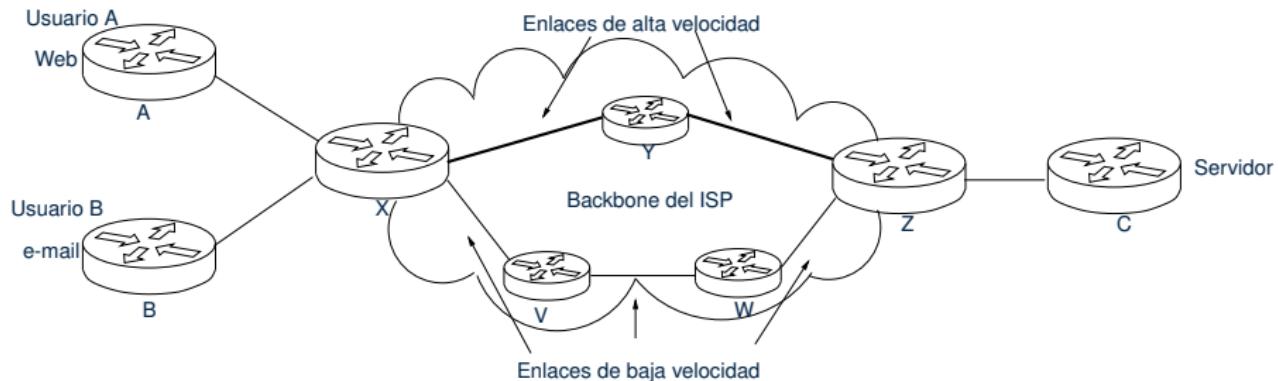
---



- Originalmente para mejorar prestaciones de Internet
- Utilización actual: encaminamiento explícito
  - Reexpedición basada en destino: por construcción, es el destino el que solicita etiquetas
  - Encaminamiento explícito
  - Creación de túneles [p. 63]

## 2.4 Multi Protocol Label Switching (III)

Ejemplo encaminamiento explícito:



## 3. Encaminadores

- 3.1. Enrutamiento desde encaminador
- 3.2. Enrutamiento desde nodo
- 3.3. Ejemplo tablas

# 3 Encaminadores



- Encaminador/*router*: conmutador de capa de red (capa 3)
  - Interconecta redes: inter-net 
  - Trabaja con direcciones lógicas (dir. IP)
  - Varios interfaces o puertos, *distintos tipos de redes*
  - Conoce ruta óptima hacia cada red destino
  - Sólo enruta hacia donde corresponde
  - Nunca enruta tramas broadcast (MAC: ff:ff:ff:ff:ff:ff)
- Cada puerto: dir. lógica (IP) y física (MAC)
- Ejemplo router ADSL: Ethernet + PPPoE (ADSL) + WiFi

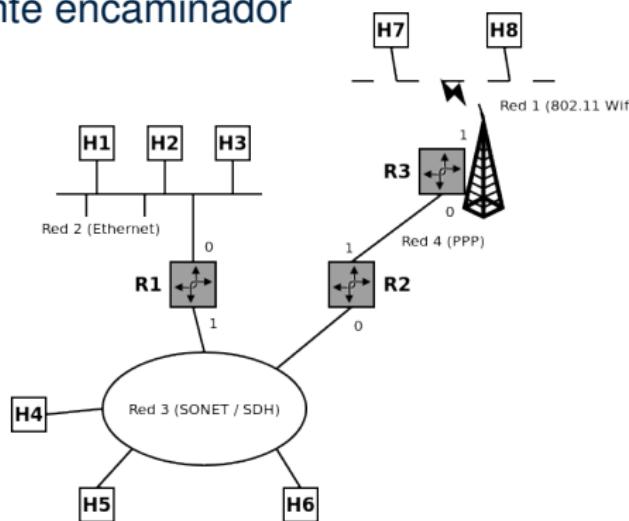


### 3.1 Enrutamiento desde encaminador

- Cada encaminador tiene una tabla con una *lista de destinos* y cómo acercarse/llegar a ellos
- Ante un paquete con una dir. destino, la busca en su tabla
  - Si está directamente conectado a la red destino → enruta hacia el nodo destino por interfaz correspondiente
  - Si no → enruta hacia siguiente encaminador según la tabla (sig. salto)

Tabla de encaminamiento de R2

Destino	Sig. salto	Interfaz
Red 1	R3	1
Red 2	R1	0
Red 3	—	0
Red 4	—	1



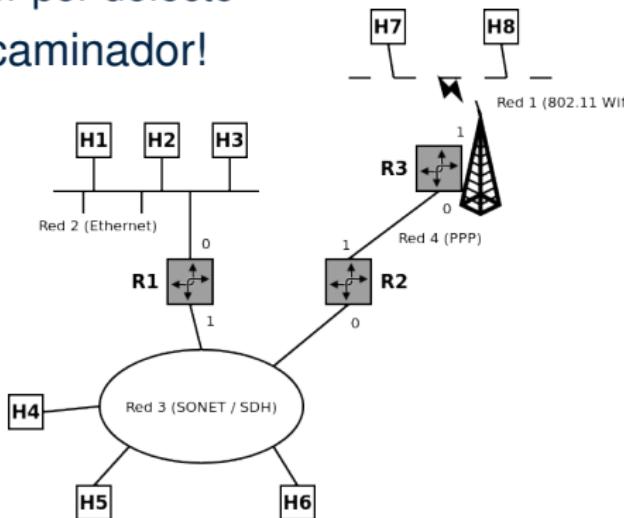
### 3.2 Enrutamiento desde nodo



- Cada nodo tiene una tabla con su red y su *encaminador (gateway) por defecto*
- Para enviar un paquete hacia dir. destino:
  - Si está en la propia red → envía a nodo destino
  - Si no → envía a encaminador por defecto
- ¡Mismo procedimiento que encaminador!

Tabla de encaminamiento de *H1*

Destino	Sig. salto	Interfaz
Red 2 default	— R1	0 0



### 3.3 Ejemplo tablas

---



```
lab000:~$ /sbin/route
Kernel IP routing table
Destination      Gateway          Genmask        Flags Metric Ref  Use Iface
0.0.0.0          155.210.152.254 0.0.0.0        UG    425   0      0 br0
155.210.152.0   0.0.0.0         255.255.255.0  U     425   0      0 br0
```

```
lab000:~$ ip route
default via 155.210.152.254 dev br0 proto dhcp src 155.210.152.177 metric 425
155.210.152.0/24 dev br0 proto kernel scope link src 155.210.152.177 metric 425
```

## 4. Protocolo IP

4.1. Cabecera IPv4

4.2. Fragmentación y reensamblado

4.3. Protocolo de Mensajes de Control de Internet (ICMP)

4.4. Procesado de un datagrama IP

# 4 Protocolo IP

---



- Internet Protocol (IP)
- Comunicación por datagrama [p. 7]
- Servicio *no fiable (unreliable)*
  - La red hace lo posible para la entrega de los paquetes, pero sin garantizarlo (*best effort*)
  - Pueden perderse paquetes
  - Pueden llegar desordenados respecto al envío
  - Se pueden entregar paquetes duplicados
  - El tiempo de entrega puede ser muy variable
- Actualmente funciona la versión 4 del protocolo IP, pero la transición a IPv6 está en marcha [p. 56]

## 4.1 Cabecera IPv4

0	4	8	16	19	31				
Ver	HLen	TOS		Longitud					
Ident		Flg		Offset					
TTL	Proto		Checksum						
Dir. origen									
Dir. destino									
Opciones (opcional, variable)			Relleno (variable)						

Versión: 4, desde comienzo de los años 80

Hlen: longitud de la cabecera, en palabras de 32 bits.

Usualmente 5 → 20 bytes, máximo 15 → 60 bytes

Type Of Service (TOS): calidad de servicio (QoS) [Tema 6]

Longitud: longitud total del paquete, en bytes

► 16 bits → máximo: 65535 bytes ( $2^{16} - 1$ )

Identificador, Flags, Offset: campos para fragmentación [p. 23]

## 4.1 Cabecera IPv4 (II)

---



**Time-To-Live:** valor fijado por emisor y decrementado por cada encaminador atravesado. Paquete descartado por encaminador que decrementa el valor a 0.

**Protocolo:** tipo de datos que contiene el paquete (ICMP, IGMP, TCP, UDP). Usado para demultiplexación en la capa superior

**Checksum:** verificación de la cabecera, no de los datos

**Direcciones:** IP origen y destino, 32 bits cada una

**Opciones:**

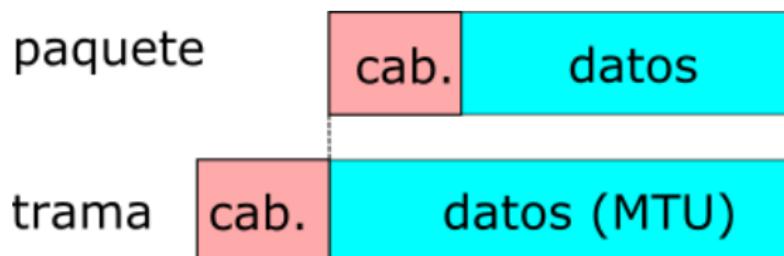
- Pedir que los encaminadores atravesados añadan información al paquete (su IP, la hora, etc.)
- Encaminamiento fuente: lista de encaminadores que pueden/deben ser atravesados
- Y otras

**Relleno:** para que los datos del paquete comiencen en posición múltiplo de 32 bits

## 4.2 Fragmentación y reensamblado



- *MTU, maximum transmission unit:* tamaño del mayor paquete que puede enviarse en los datos de una trama
- Depende del protocolo de la capa de enlace.
  - Ej. 1 500 B en 802.3, 4 464 B en 802.5, 7 981 B en 802.11



## 4.2 Fragmentación y reensamblado (II)



- Un paquete generado en una red puede no caber en otra
- Estrategia IPv4:
  - Fragmentar cuando  $MTU < \text{longitud paquete}$
  - El encargado de fragmentar es el encaminador directamente conectado a la red que requiere tramas más pequeñas
  - Los fragmentos son paquetes completos en sentido estricto: es posible fragmentar fragmentos
  - Reensamblado en nodo destino

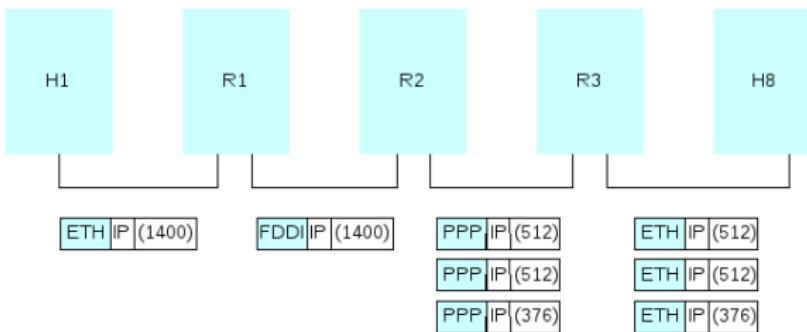


Fuente: Cloudfare: Broken packets: IP fragmentation is flawed

## 4.2.1 Ejemplo fragmentación

Ejemplo fragmentación:

- **Ident (16 bits)**: identificador de paquete (igual en todos los fragmentos)
- **Flags (3 b)**: reservado (0), Prohibido fragmentar, Siguen más fragmentos
- **Offset (13 b)**: posición de los datos en el paquete original (unidades de 8 B)



Inicio cabecera
Ident <b>x 0 0 0</b> Offset <b>0</b>
Resto cabecera
1400 bytes datos
↓
Inicio cabecera
Ident <b>x 0 0 1</b> Offset <b>0</b>
Resto cabecera
512 B datos ( <b>/8 = 64</b> )
Inicio cabecera
Ident <b>x 0 0 1</b> Offset <b>64</b>
Resto cabecera
512 bytes datos
Inicio cabecera
Ident <b>x 0 0 0</b> Offset <b>128</b>
Resto cabecera
376 bytes datos

## 4.3 ICMP

- Internet Control Message Protocol, RFC 792
- Encaminador o nodo destino informan al nodo emisor de un error, por ejemplo, paquete que no ha llegado a destino, paquete descartado ...
- No se envían mensajes ICMP sobre mensajes ICMP
- Encapsulado dentro de un paquete IP
- Cabecera ICMP: 8 bytes

0	8	16	31
Tipo	Código	Checksum	
Resto cabecera			

**Tipo/Código:** tipo/subtipo de mensaje ICMP

**Checksum:** del mensaje ICMP (cabecera + datos)

**Resto cabecera:** depende del tipo de mensaje

## 4.3.1 Ejemplos ICMP

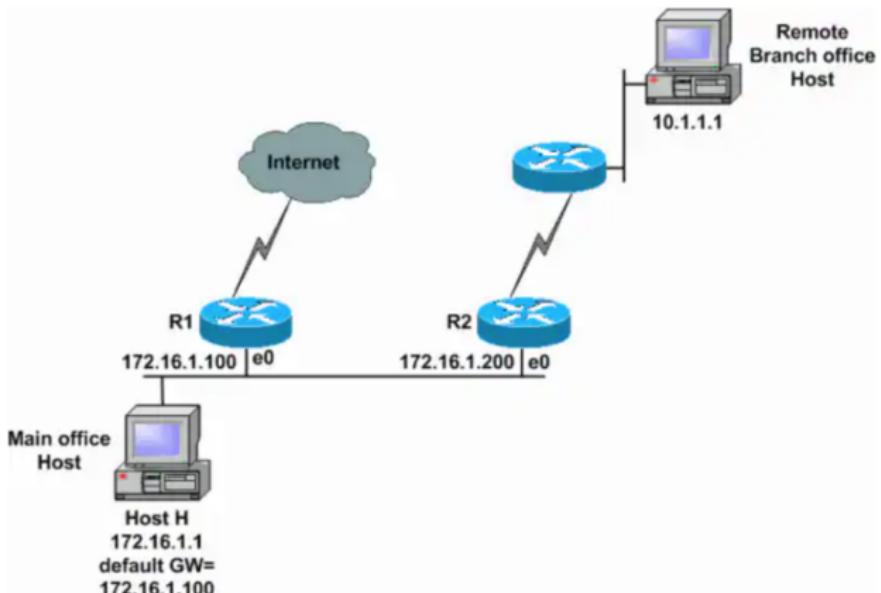
- Diagnóstico de red
  - ping: disponibilidad y latencia de un enlace
  - traceroute: ruta entre origen y destino
- Se requiere fragmentar pero DF+



Fuente: [Cloudflare: Broken packets: IP fragmentation is flawed](#)

## 4.3.1 Ejemplos ICMP (II)

- Redirect: router notifica a nodo que hay una ruta mejor hacia un destino



Fuente: Cisco: When Are ICMP Redirects Sent?

## 4.3.2 Ejercicio ping + fragmentación

---



👉 En una red 802.3, el MTU es de 1500 bytes. ¿Cuántos datos se pueden enviar en un paquete ICMP sin que haya fragmentación?

## 4.4 Procesado de un datagrama IP



1474

Al recibir una trama, el encaminador debe:

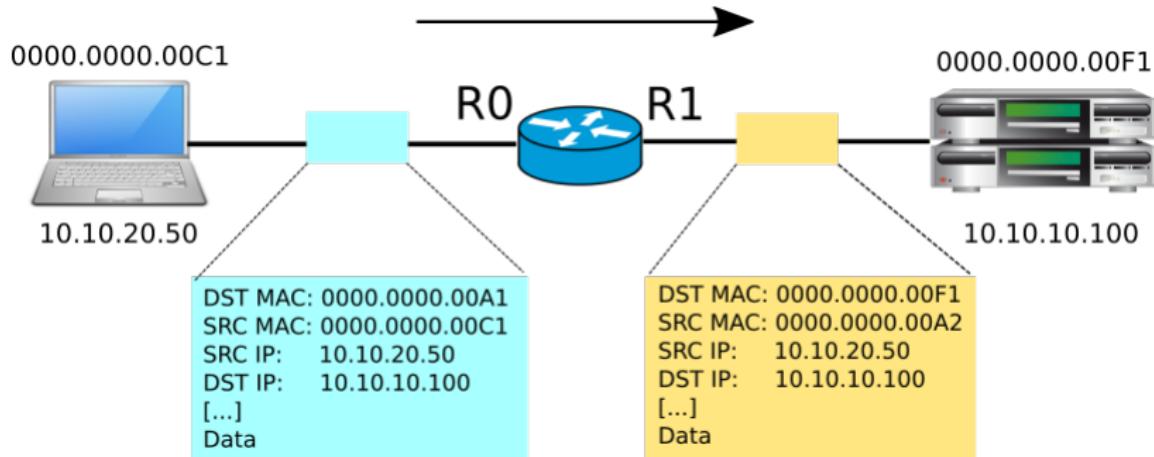
1. Validar trama (checksum/crc/etc.)
2. Validar cabecera IP (comprobar checksum)
3. Procesar opciones de la cabecera (si las hay)
4. Buscar dirección destino en tabla de encaminamiento
5. Decrementar TTL
6. Realizar fragmentación (si es necesario)
7. Calcular checksum (por cada fragmento)
8. Construir trama de capa inferior (cabecera, crc, etc.)
9. Transmitir a siguiente salto por interfaz correspondiente
10. Generar y enviar paquete ICMP (si es necesario)

El tiempo de procesado en cada salto *no es despreciable*

## 4.4 Procesado de un datagrama IP (II)



- Ejemplo de procesado de paquete en router:



## 5. Direcciones IPv4

- 5.1. Clases de direcciones IPv4
- 5.2. Direcciones especiales
- 5.3. Subredes
- 5.4. Superredes
- 5.5. ¿Cómo usar menos direcciones IP?
- 5.6. Correspondencia IP-MAC
- 5.7. Address Resolution Prot. (ARP)
- 5.8. DHCP
- 5.9. NAT
- 5.10. Ejemplo cortafuegos + NAT

# 5 Direcciones IPv4



- Direcciones de 32 bits
- Globalmente únicas, excepto las privadas
- Jerárquicas: red + nodo
- Asociadas a interfaces de red más que a nodos
- Internet Assigned Numbers Authority (IANA) reparte bloques de direcciones IPv4 a Registros Regionales de Internet (RIR) bajo demanda → *todos ya repartidos*
  - AfriNIC: África
  - APNIC: Asia-Pacífico
  - ARIN: EE.UU.-Canadá
  - LACNIC: Lat. América-Caribe
  - RIPE: Europa-Asia occidental
- DNS traduce nombres a direcciones IP [Tema 7]



# 5.1 Clases de direcciones IPv4



## ► Notación de puntos:

- 10.3.2.4
- 128.96.33.81
- 192.12.69.77

A:	0	Red	Nodo
B:	10	Red	Nodo
C:	110	Red	Nodo

## ► 5 formatos de dirección:

Clase	Prefijo	1er byte	Nº redes	Nº dir./red	Uso
Clase A	0	0-127	128	16777216	Unicast
Clase B	10	128-191	16348	65536	Unicast
Clase C	110	192-223	2097152	256	Unicast
Clase D	1110	224-239			Multicast
Clase E	1111	240-255			Experimental

- Históricamente, a una organización se le asignaba un id. red de una clase, por ejemplo, 155.210.0.0
- Actualmente, las direcciones se consideran sin clase (CIDR), por ej., puede asignarse el id. red 193.10.20.192

## 5.1.1 Direcciones sin clase

---



- Ignora la distinción entre clases A, B y C y las fronteras entre identificadores de red y nodo
- Notación dirección red: dirección red/número bits red
  - 155.210.0.0/16
  - 206.62.226.0/24
- El uso de direcciones sin clase requiere encaminamiento sin clase: *Classless Inter-Domain Routing*, RFC 1519
- Objetivos CIDR
  - Reducir tamaño tablas encaminamiento routers
  - Reducir agotamiento direcciones IPv4

## 5.2 Direcciones especiales



- Dirección de red: todo 0s en bits de nodo. Ej: 128.96.0.0
- Difusión/broadcast: todo 1s en bits nodo. Ej: 192.12.69.255
- 0.0.0.0: dirección no encaminable.
  - Significado dependiente del contexto, ej. ruta por defecto
- 127.0.0.0/8: bucle (*loopback*), el propio nodo
  - Ejemplo: 127.0.0.1
  - RFC 1122, Section 3.2.1.3
- 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16: privadas
  - No únicas globalmente → no sirven como dirección destino a nivel global (Internet)
  - Válidas a nivel local → redes privadas
  - Para conectarse a Internet se necesita traducción de direcciones privadas (NAT) [p. 50]
  - RFC 1918

## 5.2 Direcciones especiales (II)

- 169.254.0.0/16: link-local
  - Dirección que se autoasigna un nodo cuando no ha podido obtener otra mediante configuración manual (fichero) o automática (DHCP)
  - Válida únicamente para comunicación local
  - Bits de nodo: valores aleatorios para mitigar conflictos
  - RFC 3927
- Paquetes con destino difusión/privadas/link-local *no salen de la red local*

## 5.3 Subredes

---



- Direcciones unicast posibles en una red:  $2^{\text{nº bits nodo}} - 2$
- Problema: redes muy grandes
  - Broadcast excesivo, colisiones, retardos, pérdida de tramas
- Solución: añadir nivel de jerarquía red-subred-nodo
  - Se usan *bits de nodo* para identificar subredes
  - Cada subred necesita @ red, broadcast y encaminador
  - Fuera de una red no hay conciencia de sus subredes  
(los encaminadores externos propagan hasta la red)
- Gestionadas mediante *máscaras de subred*

## 5.3.1 Máscara de subred

- Información total & Máscara de bits = Información útil



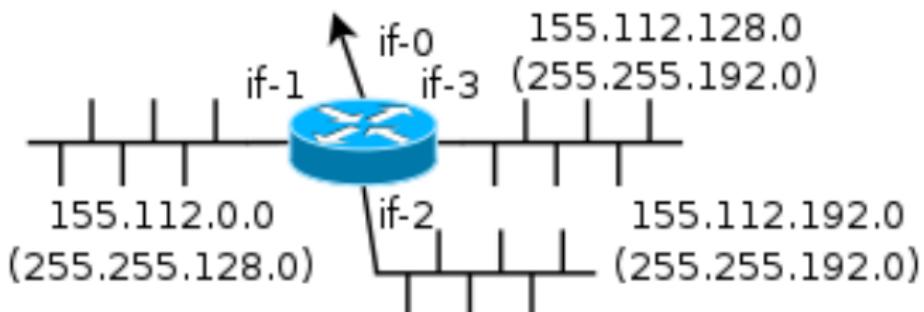
- Dirección IP & Máscara de subred = Dirección de subred

- Red 155.210.0.0/16

bytes id. red	bits id. red	bits id. nodo
IP 155.210.153.238 -	10011011.11010010.	10011001.11101110
Mask 255.255.248.0 -	11111111.11111111.	11111000.00000000
Sub 155.210.152.0 -	10011011.11010010.	10011000.00000000
	bits subred	bits nodo

- Subred 155.210.152.0/255.255.248.0 = 155.210.152.0/21

## 5.3.2 Ejemplo subredes



Destino	Máscara subred	Interfaz	(Destinos)
155.112.0.0	255.255.128.0	If-1	155.112.0xxxxxx.X
155.112.128.0	255.255.192.0	If-3	155.112.10xxxxxx.X
155.112.192.0	255.255.192.0	If-2	155.112.11xxxxxx.X
default	-	if-0	



Resolver con notación CIDR

## 5.3.3 Ejercicio subredes



✍ Para la red 155.210.0.0/16, ¿son válidas estas máscaras?

- a) 255.224.0.0 (11111111.111**0** 0000.0000000.0000000)
- b) 255.255.216.0 (11111111.11111111.11**0**11000.0000000)
- c) 255.255.255.0 (11111111.11111111.11111111.00000000)

## 5.4 Superredes

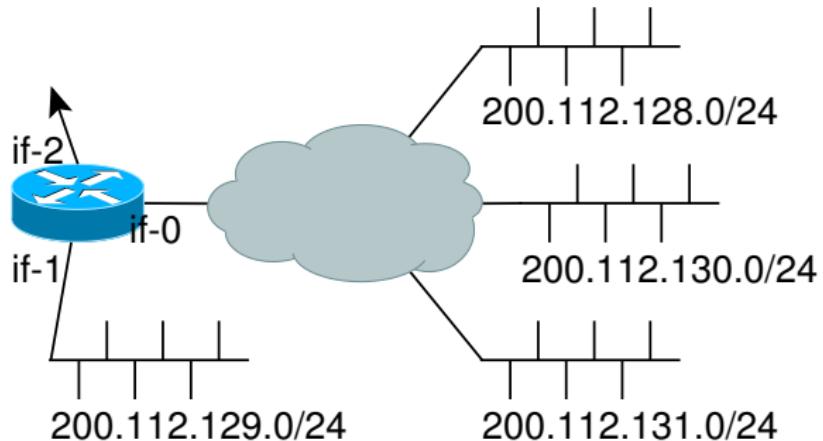


- RFC 4632
- Los encaminadores «necesitan» una entrada en la tabla por cada red destino, y hay más de 2 millones de redes
- Problema: con tantas entradas las tablas no son eficientes
- Objetivo: juntar varias redes contiguas en una única entrada en tabla
  - Redes 192.4.8.0/24 - 192.4.15.0/24 → 192.4.8.0/21

192.4.00001000.0  
192.4.00001001.0  
192.4.00001010.0  
...  
192.4.00001111.0

192.4.8.0/21

## 5.4.1 Ejemplo CIDR



Destino	Interfaz
200.112.128.0	If-0
200.112.129.0	If-1
200.112.130.0	If-0
200.112.131.0	If-0
default	If-2

Destino	Interfaz
200.112.129.0/24	If-1
200.112.128.0/22	If-0
default	If-2

- If-0: 200.112.128.0, 130.0, 131.0 ( $100000\frac{0}{1}\frac{0}{1}$ )
  - 22 bits de red comunes: 200.112.100000xx.xxxxxxxx
  - Prefijo común: 200.112.128.0/22
  - Redes 128, 130 y 131 se agrupan en una única entrada
- If-1: 200.112.129.0 ( $10000001$ ) más prioritaria en tabla por estar incluida en 200.112.128.0/22

## 5.5 ¿Cómo usar menos direcciones IP?



- Problema: agotamiento de direcciones IPv4 unicast
- Servidores necesitan estar localizables en todo momento para recibir peticiones
  - IP *estática* o fija: dirección IP que no cambia con el tiempo
- Clientes no necesitan estar localizables
  - IP *estática*, si hay para todos
  - IP *dinámica*: @IP que puede cambiar con el tiempo
- Asignación dinámica desde un servidor (DHCP, PPP)
  - Al apagar el equipo, otro puede usar su @IP
  - Al reiniciar el equipo, podemos usar una @IP distinta
  - Una @IP nunca es usada por varios equipos a la vez
- Compartir direcciones IP unicast (NAT)
  - Una dirección IP es usada por varios equipos a la vez



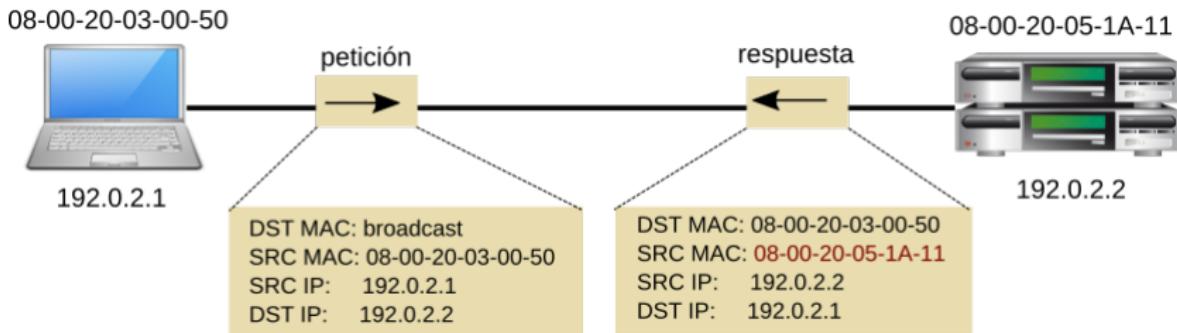
## 5.6 Correspondencia IP-MAC

---



- La capa de red funciona con direcciones IP
- La capa de enlace funciona con identificadores MAC
- ¿Cuándo se necesita asociar ambas?
  - Nodo necesita el id. MAC de su encaminador por defecto y de los nodos de la misma red
  - Encaminador necesita el id. MAC del siguiente salto
  - Encaminador final necesita el id. MAC del nodo destino
- ¿Cómo se asocian ambas direcciones?
  - Codificar id. MAC dentro de dir. red (e.g. EUI-64)
    - Usado en IPv6
    - No sirve en IPv4 porque los identificadores MAC tienen un tamaño mayor que las direcciones IPv4
  - Mediante tablas, e.g. Address Resolution Protocol - ARP
    - Usado en IPv4

## 5.7 Address Resolution Prot. (ARP)



- Mensajes de petición y respuesta sobre capa de enlace
  - 1. Nodo origen consulta @IP destino en tabla ARP.  
Si no está, petición ARP por difusión
  - 2. Nodos que reciben petición actualizan su tabla con la correspondencia IP-MAC origen
  - 3. Nodo destino responde con su id. MAC
  - 4. Nodo origen actualiza su tabla con IP-MAC destino

## 5.7 Address Resolution Prot. (ARP) (II)



- Entradas de tabla ARP:
  - No se añaden por ningún otro procedimiento
  - Son eliminadas tras un tiempo sin usarse
- Ejemplo consulta tablas ARP mediante orden arp:

```
lab000:~$ arp
Address          HWtype  HWaddress          Flags Mask   Iface
hendrix02.cps.unizar.es  ether   00:14:4f:ec:4d:54  C      br0
155.210.152.254       ether   f0:f7:55:f3:c7:c1  C      br0
camposancos.cps.unizar.  ether   00:1c:c0:ef:8d:8d  C      br0
```

## 5.7.1 Ejemplo petición-respuesta ARP



No.	Time	Source	Destination	Protocol	Length	Info
4338	158.72	IntelCor_8b:f9	Broadcast	ARP	42	who has 192.168.0.1? Tell 192.168.0.13
4339	158.72	1 CiscoSvp_f4:49:1f	IntelCor_8b	ARP	42	192.168.0.1 is at 18:59:33:f4:49:1f

Frame 4338: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0  
Ethernet II, Src: Intelcor\_8b:f9:de (68:5d:43:8b:f9:de), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
Destination: Broadcast (ff:ff:ff:ff:ff:ff)  
Source: Intelcor\_8b:f9:de (68:5d:43:8b:f9:de)  
Type: ARP (0x0806)  
Address Resolution Protocol (request)  
Hardware type: Ethernet (1)  
Protocol type: IP (0x0800)  
Hardware size: 6  
Protocol size: 4  
Opcode: request (1)  
Sender MAC address: Intelcor\_8b:f9:de (68:5d:43:8b:f9:de)  
Sender IP address: 192.168.0.13 (192.168.0.13)  
Target MAC address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Target IP address: 192.168.0.1 (192.168.0.1)

Frame 4339: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0  
Ethernet II, Src: CiscoSvp\_f4:49:1f (18:59:33:f4:49:1f), Dst: Intelcor\_8b:f9:de (68:5d:43:8b:f9:de)  
Destination: Intelcor\_8b:f9:de (68:5d:43:8b:f9:de)  
Source: CiscoSvp\_f4:49:1f (18:59:33:f4:49:1f)  
Type: ARP (0x0806)  
Address Resolution Protocol (reply)  
Hardware type: Ethernet (1)  
Protocol type: IP (0x0800)  
Hardware size: 6  
Protocol size: 4  
Opcode: reply (2)  
Sender MAC address: CiscoSvp\_f4:49:1f (18:59:33:f4:49:1f)  
Sender IP address: 192.168.0.1 (192.168.0.1)  
Target MAC address: Intelcor\_8b:f9:de (68:5d:43:8b:f9:de)  
Target IP address: 192.168.0.13 (192.168.0.13)

The diagram illustrates the flow of ARP requests and responses between hosts. Hosts a, b, c, d, e, f, g, h, i, and j are represented by green circles. The top section shows host e sending an ARP request (Frame 4338) to broadcast (ff:ff:ff:ff:ff:ff). The bottom section shows host j responding with an ARP reply (Frame 4339) to host e. Lines connect host e to the broadcast and host j to host e, indicating the path of the ARP message exchange.

Fuente: Sharetechnote.com: IP Network - ARP

## 5.8 DHCP

---



- Dynamic Host Configuration Protocol
- Un servidor DHCP proporciona información para que los equipos de su LAN configuren la red:
  - Dirección IP: no necesariamente la misma siempre
  - Máscara de subred
  - Encaminador por defecto
  - Servidor de nombres [Tema 7]
  - etc.
- Pasos de configuración:
  1. Al arrancar, el equipo busca un servidor DHCP:  
IPsrc: 0.0.0.0, IPdest: 255.255.255.255, UDP: 67
  2. El servidor responde anunciando su presencia:  
IPdest: 255.255.255.255
  3. El equipo pide datos al servidor
  4. El servidor proporciona los datos

# 5.9 Network Address Translation



1. Uso de @IP privadas en equipos locales
    - 10.X.X.X, 172.0001xxxx.X.X, 192.168.X.X
    - Válidas localmente pero NO globalmente
  2. Enmascarar tras @IP válida con encaminador NAT
    - Network Address (and port) Translator
    - El encaminador tiene una @IP unicast válida
    - Salida: NAT sustituye @IP origen por la propia
    - Entrada: NAT deshace la sustitución: deduce a quién entregar el paquete
- *Solución arquitectónicamente mala* 
- Modifica el funcionamiento básico de la capa de red (el NAT cambia direcciones en paquetes)
  - Implica restricciones de funcionalidad
- Carrier-grade NAT / Large-scale NAT: NAT dentro de NAT

# 5.9 Network Address Translation (II)

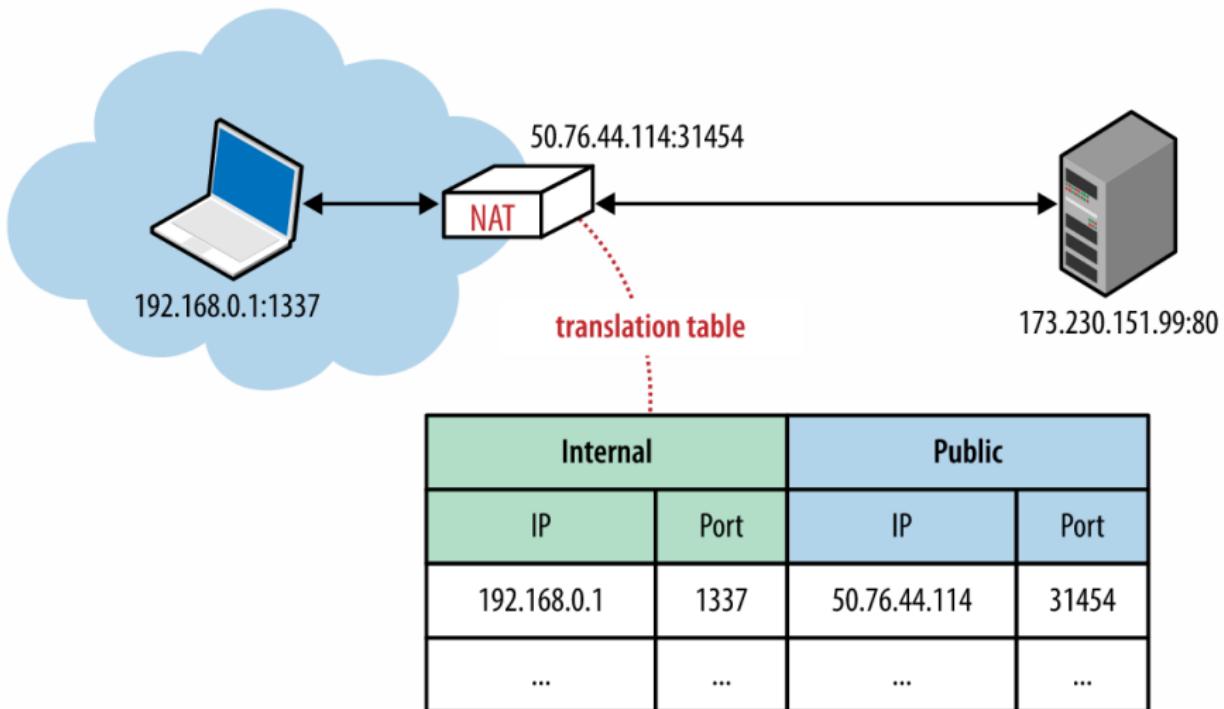
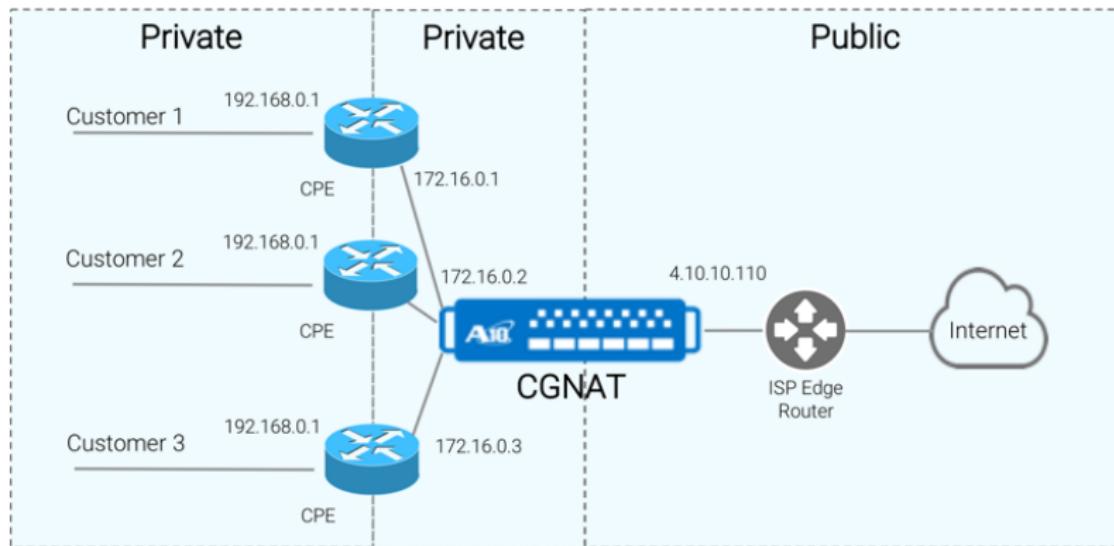


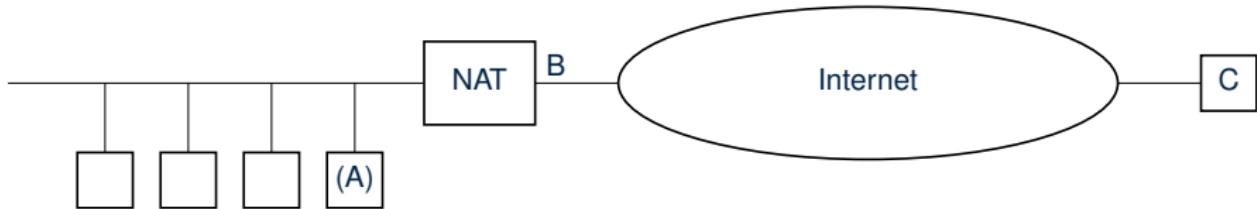
Imagen: High Performance Browser Networking: Building Blocks of UDP

# 5.9 Network Address Translation (III)

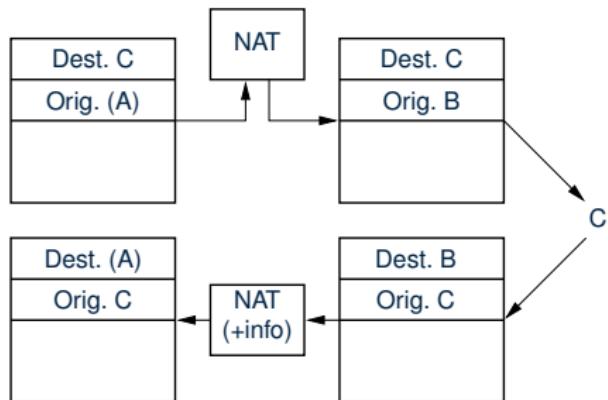
## Ejemplo NAT444



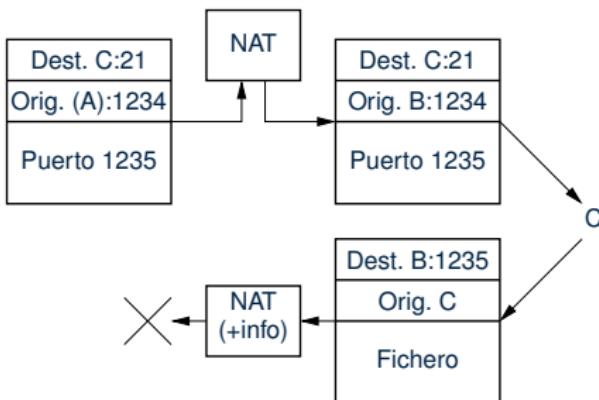
# 5.9 Network Address Translation (IV)



Ejemplo «funcional»



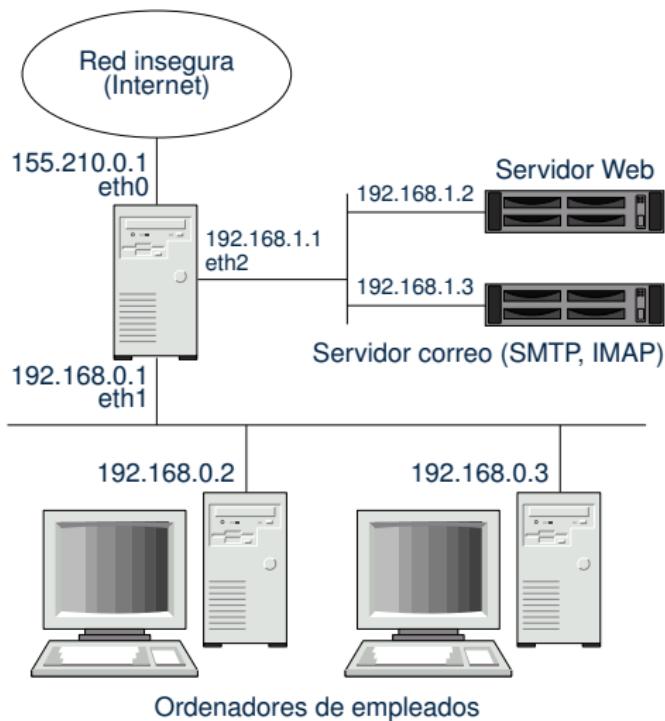
Ejemplo FTP «problemático»



# 5.10 Ejemplo cortafuegos + NAT



Ej. de manipulación de paquetes:



- Redirigir paquetes SMTP e IMAP al servidor de correo (NAT)
- Redirigir paquetes web, con un límite de 1 por segundo y ráfagas de 10 como máximo, al servidor web (NAT+filtro)
- Descartar todo el tráfico con protocolo distinto de TCP (filtro)
- Hacer de NAT a los ordenadores de empleados
- Inhabilitar el uso de la red en la propia máquina (filtro)
- Descartar conexiones desde 192.168.0.3 a puertos 6667 del exterior (filtro)

## 6. IP versión 6

6.1. Cabecera IPv6

6.2. Direcciones IPv6

6.3. Ejercicio de direcciones

## 6 IP versión 6

---

- RFC 8200
- Propuesta inicial en 1991 (IETF IPng: IP Next Generation)
- Nuevo protocolo → requiere actualizar encaminadores
- Intención de transición progresiva, que se ha ido retrasando «forzando» IPv4
- Direcciones de 128 bits sin clases y jerárquicas (red + interfaz)
- Al no haber NAT es posible tener seguridad en capa de red
  - Implementación obligatoria de IPsec
- No hay fragmentación (ICMP notifica tamaño demasiado grande)
- Estadísticas adopción:  
<https://www.google.com/intl/en/ipv6/statistics.html>

# 6.1 Cabecera IPv6



Cabecera de tamaño fijo (40 bytes):

1	4	12	32	48	56	64
Ver.	TrafClas	Flow Label	Payload Length	NextHdr	HopLimit	
Dirección origen						
Dirección destino						

- No hay checksum ni campos fragmentación ni opciones
- Campos ligeramente modificados: TOS → TrafficClass, Length → PayloadLength, Protocol → NextHeader, TTL → HopLimit
- Nuevo campo *FlowLabel*
- Opciones → Cabeceras de extensión (NextHeader)

## 6.1 Cabecera IPv6 (II)



Cabeceras de extensión (ejemplo):

<i>IPv6 Header</i> NextHeader= Security	<i>Security Header</i> NextHeader= Fragmentation	<i>Frag. Header</i> NextHeader= TCP	TCP Header	DATA
--	---	--	------------	------

- Enlazadas con campos NextHeader hasta capa superior
- Procesadas por nodo destino  
(excepto extensión *Hop-by-Hop*)
- Permite añadir funcionalidad sin cambiar protocolo y sin modificar encaminadores
- E.g. salto a salto, fragmentación, autenticación, encapsulado de seguridad de la carga útil

## 6.2 Direcciones IPv6



- RFC 4291
- Direcciones de 128 bits
- 3 tipos:
  - Unicast: interfaz único
  - Anycast: conjunto de interfaces
  - Multicast: conjunto de interfaces
- Paquete enviado a dirección **anycast/multicast** se envía al **interfaz más cercano/todos los interfaces** del conjunto
- No hay direcciones broadcast
- Representación:
  - 8 números de 16 bits en formato hexadecimal
    - 2001:0DB8:0000:0000:0000:A456:0024
  - 0 contiguos y a la izquierda se pueden omitir
    - 2001:DB8::A456:24
    - "::"solamente puede aparecer una vez

## 6.2 Direcciones IPv6 (II)

- Tipos de direcciones:

Tipo	Prefijo	Notación IPv6
Sin especificar	00...0 (128 bits)	::/128
Bucle local ( <i>Loopback</i> )	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Global Unicast	resto	

- Bloque de direcciones global unicast:



- Identificador de interfaz autoconfigurado a partir de MAC, asignado mediante DHCP, aleatorio (privacidad) o establecido manualmente

## 6.3 Ejercicio de direcciones

☞ Busca la configuración de los equipos de tu casa. Para cada uno de ellos (excepto el encaminador):

- ¿Cuál es su dirección IP? ¿Tiene IPv4, IPv6 o ambas?
- ¿Es pública o privada?
- ¿Está en alguna subred?
- ¿Está configurado manualmente o vía DHCP?

Para el encaminador:

- ¿Qué dirección IP tiene cada uno de sus interfaces?
- ¿Tiene IPv4, IPv6 o ambas?
- ¿Cuáles son públicas?
- ¿Es también un servidor DHCP?
- ¿Es también un NAT?

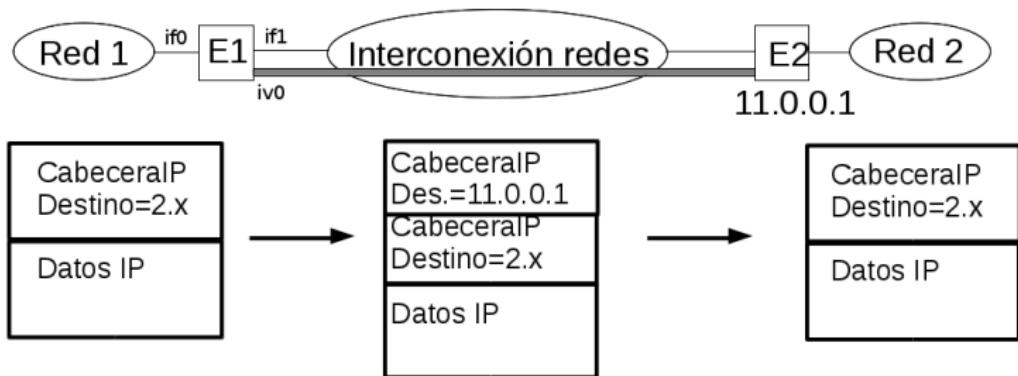
## 7. Túneles

### 7.1. Ejemplo VPN

# 7 Túneles



- Encapsulación de paquetes de la capa de red dentro de paquetes de la capa de red



- Tabla encaminamiento E1:

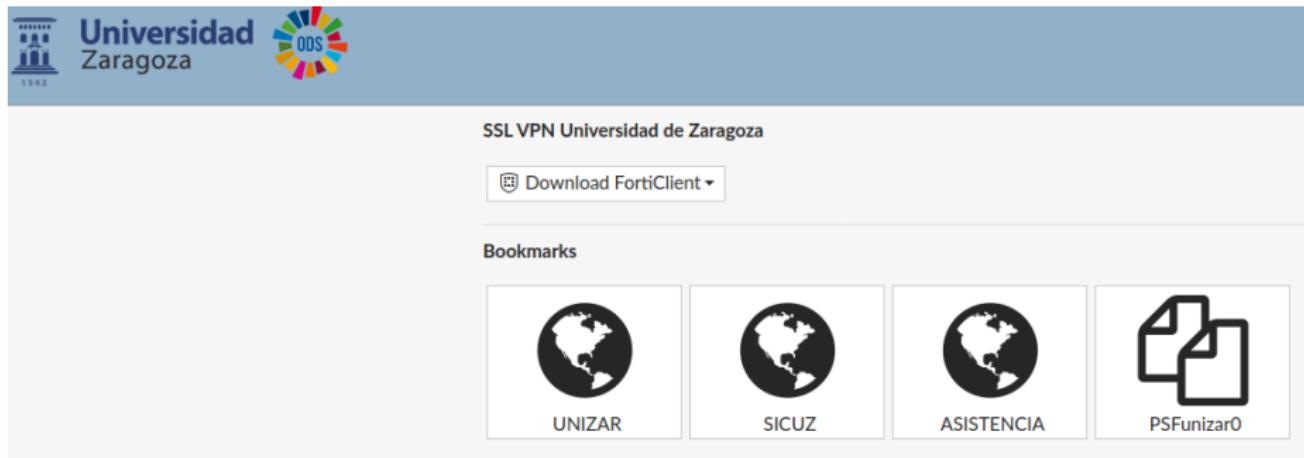
	Destino	Interfaz
	Red 1	Interfaz 0
	Red 2	Interfaz virtual 0
	Por defecto	Interfaz 1

- Túnel+cifrado: VPN (Virtual Private Network)

# 7.1 Ejemplo VPN



<https://remoto.unizar.es>



The screenshot shows the SSL VPN interface for the University of Zaragoza. At the top left is the university's logo and name. A blue header bar contains the text "SSL VPN Universidad de Zaragoza". Below the header is a button labeled "Download FortiClient ▾". Underneath this is a section titled "Bookmarks" containing four items: "UNIZAR" (with a globe icon), "SICUZ" (with a globe icon), "ASISTENCIA" (with a globe icon), and "PSFunizar0" (with a document icon).

SSL VPN Universidad de Zaragoza

Download FortiClient ▾

Bookmarks

UNIZAR SICUZ ASISTENCIA PSFunizar0

## 8. Protocolos de encaminamiento

- 8.1. Sistemas autónomos
- 8.2. Protocolo encaminamiento interior
- 8.3. Protocolo encaminamiento exterior
- 8.4. Algoritmos de encaminamiento
- 8.5. Estado de enlace
- 8.6. Ejemplo búsqueda de caminos

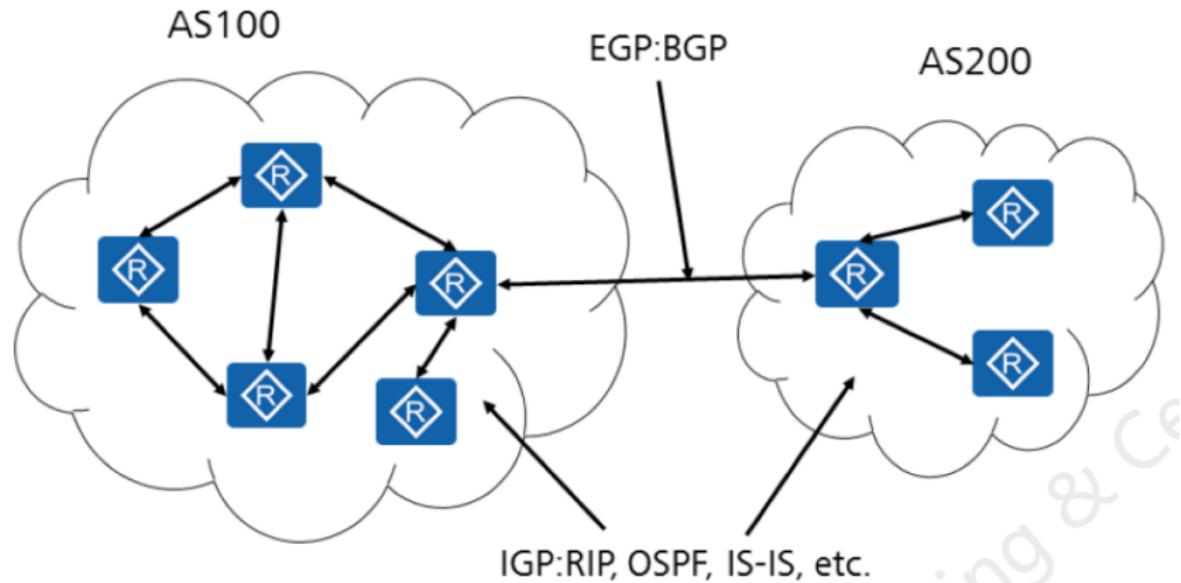
# 8 Protocolos de encaminamiento

- Encaminadores reciben paquetes y los reenvían hacia el destino
- Decisiones basadas en
  - Conocimiento topología de la red
  - Condiciones red (caídas, retardos ...)
- Encaminadores intercambian este tipo de información mediante protocolos de encaminamiento
- Objetivo: configuración automática de tablas de encaminamiento
- Ventajas: reacción y adaptación a cambios
  - Propagación rutas alternativas

# 8 Protocolos de encaminamiento (II)



- Dos tipos:
  - Interior: dentro de un Sistema Autónomo (SA)
  - Exterior: entre SAs



## 8.1 Sistemas autónomos

---



- Sistema Autónomo (SA) es un grupo de redes IP que poseen una política de rutas propia e independiente:
  - Un SA es un dominio administrativo independiente
  - Se asigna a cada SA un número de 32 bits (ASN)
  - Ejemplos: gran compañía, red columna vertebral
- *Gestión de caminos en dos niveles*: dentro/fuera de un SA
  - Dentro: optimizar caminos entre las redes que contiene
  - Fuera: buscar caminos que comuniquen los distintos SAs
- Ventajas por trabajar con subconjuntos de Internet:
  - *Escalabilidad*: que al crecer todo siga funcionando
  - *Eficiencia*: poco tráfico de configuración
  - *Tolerancia a fallos*: buscar rápido otras rutas en caso de fallo del enlace o de un encaminador

## 8.2 Protocolo encaminamiento interior

---



- Entre encaminadores dentro de un SA
- Todos los encaminadores del SA deben usar el mismo protocolo de búsqueda de caminos
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First

## 8.3 Protocolo encaminamiento exterior

---



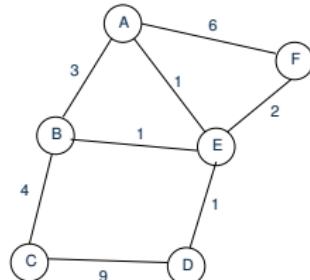
1474

- Entre encaminadores de distintos SAs
- Cada SA tiene uno o más encaminadores frontera, que anuncian
  - Redes internas
  - Redes externas alcanzables (sólo en SA de tránsito)
  - Información de caminos
- Todos los SAs deben usar el mismo protocolo:
  - BGP-4 (*Border Gateway Protocol 4*), RFC 4271
- Elección de rutas basada en políticas explícitas (preferencias basadas en precio, etc.)
- Para poder atravesar ciertos SAs y evitar otros, BGP trabaja con rutas completas, no solo el siguiente salto

## 8.4 Algoritmos de encaminamiento

Objetivo: encontrar el camino con el menor coste entre dos nodos en una red dada. Se basan en:

- Topología: grafo donde cada red es un nodo
- Métricas de coste en enlaces
  - Coste constante: e.g. capacidad enlace, nº saltos, etc.
  - Factor dinámico «simple»: e.g. nº paquetes en cola
  - Factor dinámico dependiente de capacidad y carga (e.g. retardo medio de los últimos *n* minutos)
  - Factores económicos: e.g. precio por transmisión
  - Etc.
- Con esta información, los encaminadores construyen su tabla de encaminamiento



## 8.4 Algoritmos de encaminamiento (II)

---



1474

Dos tipos:

- Distancia-vector
  - «Si el router X está a 5 saltos de la red Y, y yo soy adyacente a X, estoy a 6 saltos de la red Y»
  - Problemas de convergencia
  - RIP (LANs), IGRP (WANs pequeñas), EIGRP (WANs), BGP (Internet backbone)
- Estado-enlace
  - «Router X es adyacente al router Y por enlace activo»
  - Convergencia rápida
  - Requiere más memoria y CPU que distancia-vector
  - OSPF (LANs, WANs pequeñas)

## 8.5 Estado de enlace (link-state)



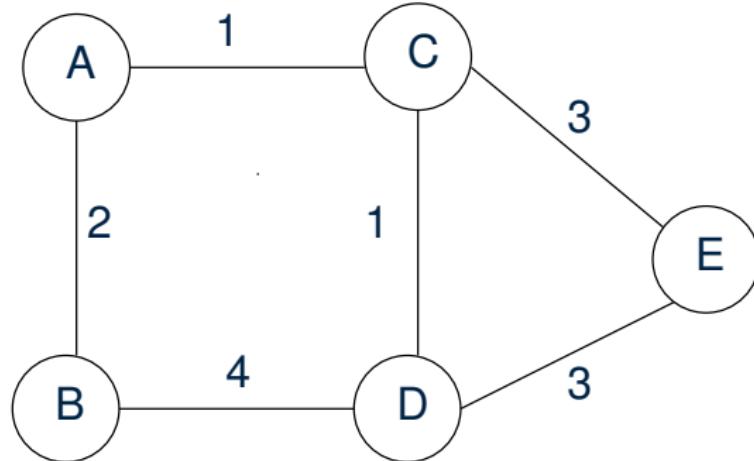
- Cada actualización contiene información sobre los *enlaces directamente conectados* (no toda la tabla de encam.)
- Envía actualización de estado a *todos los nodos* (no sólo a los vecinos inmediatos) mediante inundación fiable

Cada nodo aplica el algoritmo de *Dijkstra* (teoría de grafos) para calcular la mejor ruta hacia otros nodos:

1. Confirmar camino con distancia 0 al propio nodo
2. Crear lista vacía de posibles caminos y costes
3. Para el nuevo nodo confirmado:
  - 3.1 Añadir a la lista los nodos directamente conectados al nuevo nodo confirmado y el coste para llegar a ellos pasando por él
  - 3.2 Si hay varios caminos para llegar a un nodo, descartar los más costosos
  - 3.3 Confirmar el nodo con menor coste y quitarlo de la lista

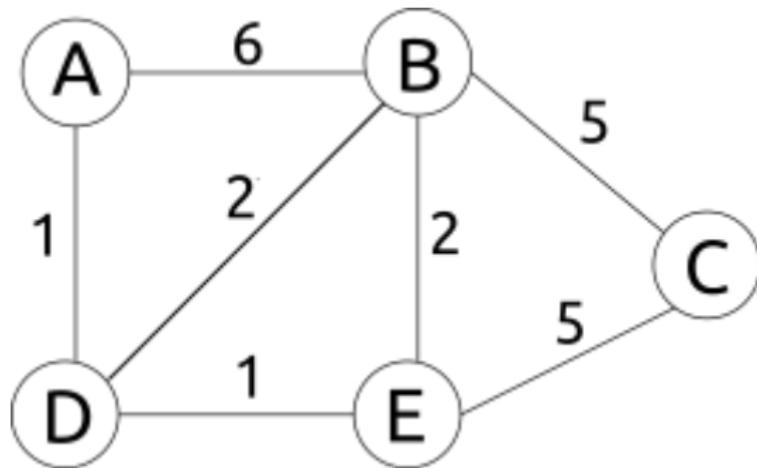
## 8.6 Ejemplo búsqueda de caminos

Obtener la tabla de encaminamiento del nodo A mediante el algoritmo de estado de enlace

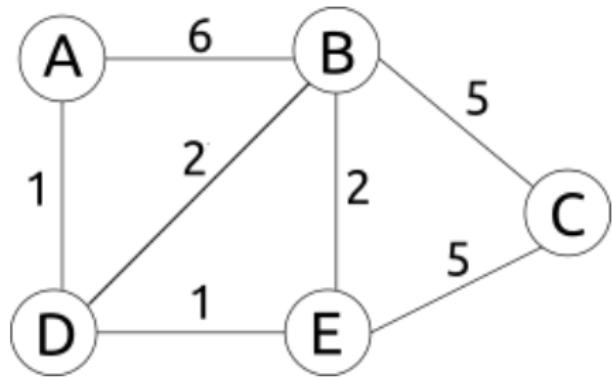


## 8.6 Ejemplo búsqueda de caminos

Obtener la tabla de encaminamiento del nodo A mediante el algoritmo de estado de enlace

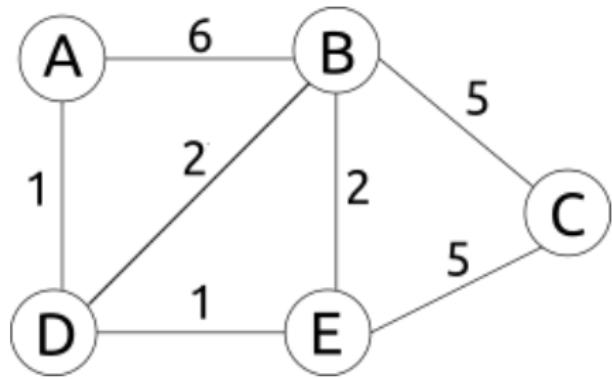


## 8.6 Ejemplo búsqueda de caminos



Vértice	Distancia	Último nodo
A		
B		
C		
D		
E		

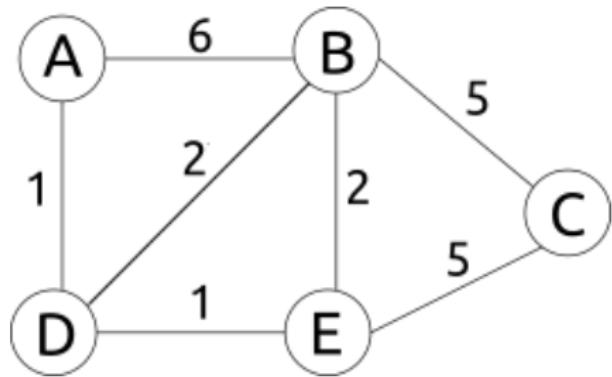
## 8.6 Ejemplo búsqueda de caminos



Vértice	Distancia	Último nodo
A	0	-
B	$\infty$	
C	$\infty$	
D	$\infty$	
E	$\infty$	

Visitados = [ - ], No visitados = [A,B,C,D,E]

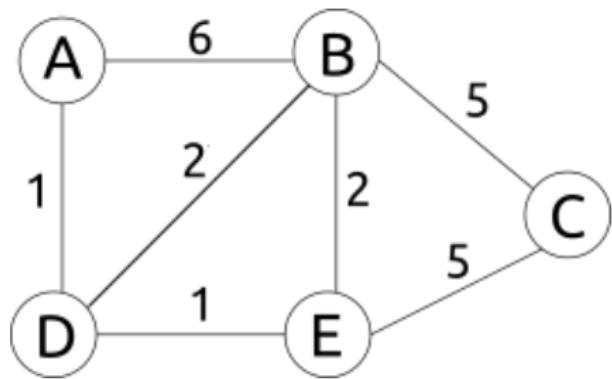
## 8.6 Ejemplo búsqueda de caminos



Vértice	Distancia	Último nodo
A	0	-
B	6	A
C	$\infty$	
D	1	A
E	$\infty$	

Visitados = [A], No visitados = [B,C,D,E]

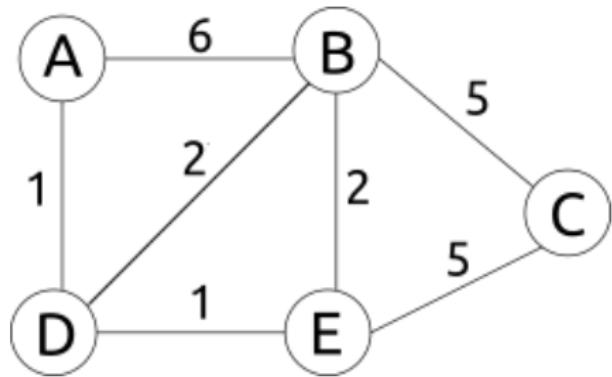
## 8.6 Ejemplo búsqueda de caminos



Vértice	Distancia	Último nodo
A	0	-
B	3	D
C	$\infty$	
D	1	A
E	2	D

Visitados = [A, D], No visitados = [B, C, E]

## 8.6 Ejemplo búsqueda de caminos



Vértice	Distancia	Último nodo
A	0	-
B	3	D
C	7	E
D	1	A
E	2	D

Visitados = [A,D,E], No visitados = [B,C]

## 9. Estructura de Internet

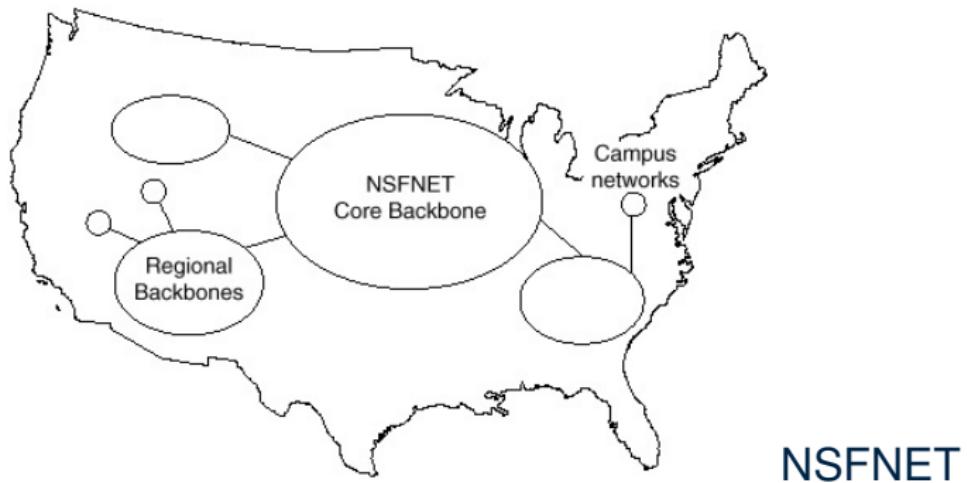
### 9.1. Sistemas autónomos

# 9 Estructura de Internet



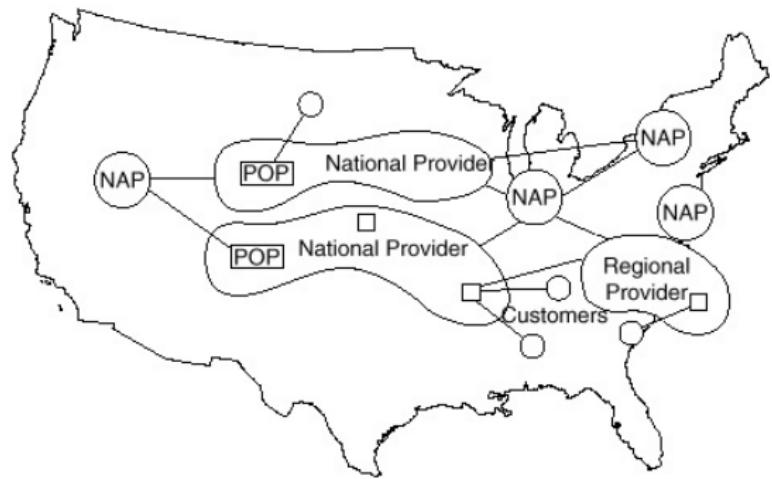
1474

## ► Pasado reciente (1986-1995)



# 9 Estructura de Internet (II)

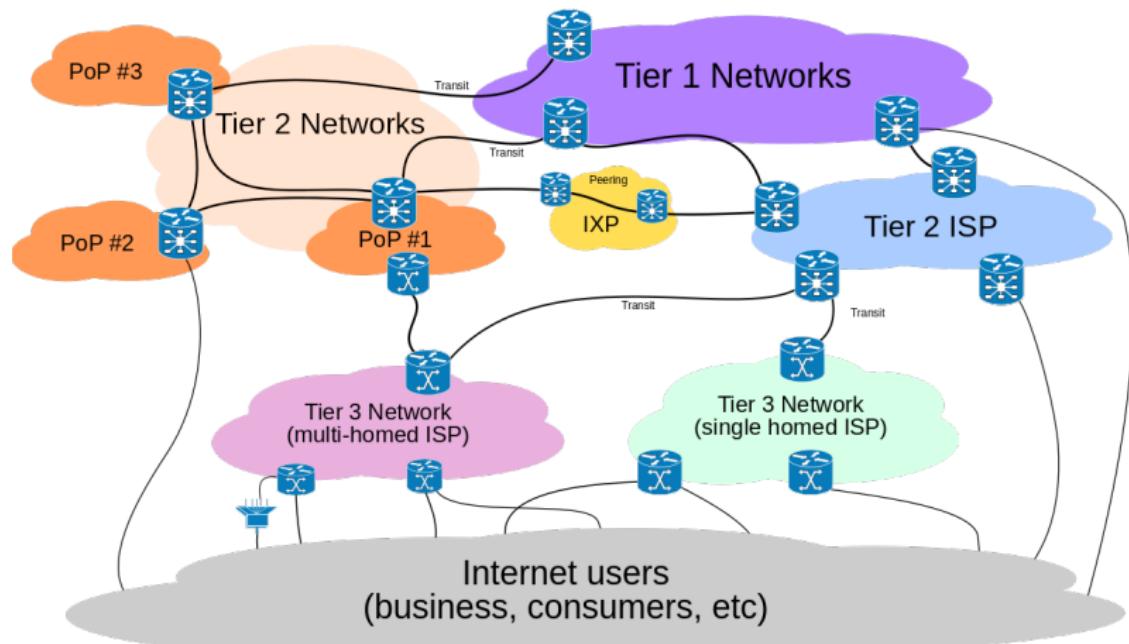
- Ahora es una estructura jerárquica operada por proveedores comerciales



# 9 Estructura de Internet (III)



## ► ISP roles y relaciones



[https://en.wikipedia.org/wiki/Internet\\_transit](https://en.wikipedia.org/wiki/Internet_transit)

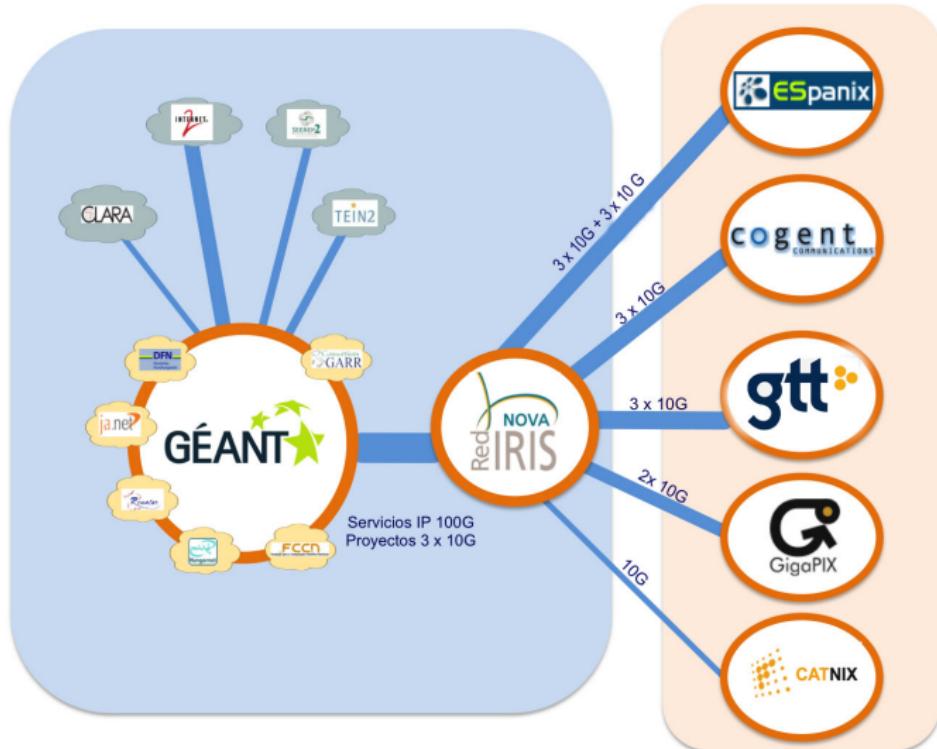
# 9.1 Sistemas autónomos

- E.g. RedIRIS (SA 766) agrupa las redes de las universidades y centros de investigación en España



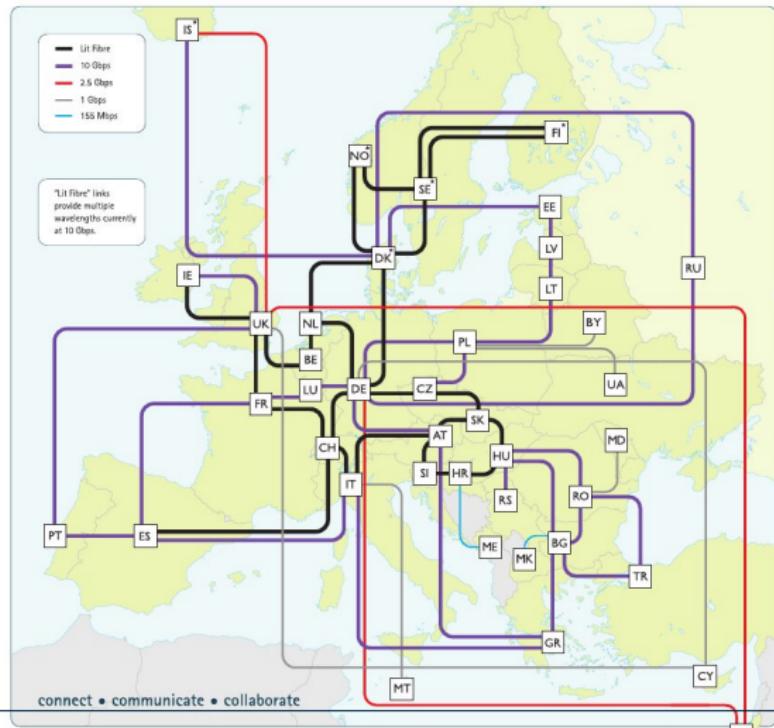
# 9.1 Sistemas autónomos (II)

- Conectividad externa de RedIRIS



# 9.1 Sistemas autónomos (III)

- E.g. de SA troncal: GÈANT (SA 20965) comunica los SAs de investigación europeos



# **Redes de Computadores**

## **Tema 5 – Capa de transporte**

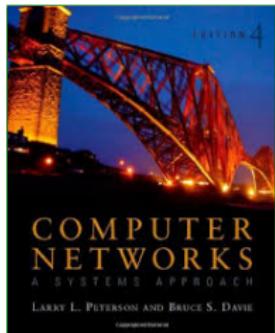
**Natalia Ayuso, Juan Segarra y Jesús Alastruey**



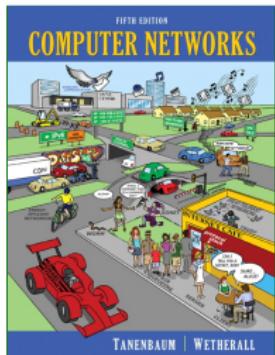
Departamento de  
Informática e Ingeniería  
de Sistemas

**Universidad** Zaragoza

1. Introducción
2. User Datagram Protocol (UDP)
  - 2.1. Cabecera UDP
  - 2.2. Puertos software
3. Transmission Control Protocol (TCP)
  - 3.1. Cabecera TCP
  - 3.2. Establecimiento/finalización de conexión
  - 3.3. Estados de una conexión TCP
  - 3.4. Envío de datos
  - 3.5. Control de flujo
  - 3.6. Opciones TCP
  - 3.7. Retransmisión adaptativa
  - 3.8. Conclusiones



Capítulo 5



Capítulo 6

# 1 Introducción

- Función: control de la comunicación extremo a extremo

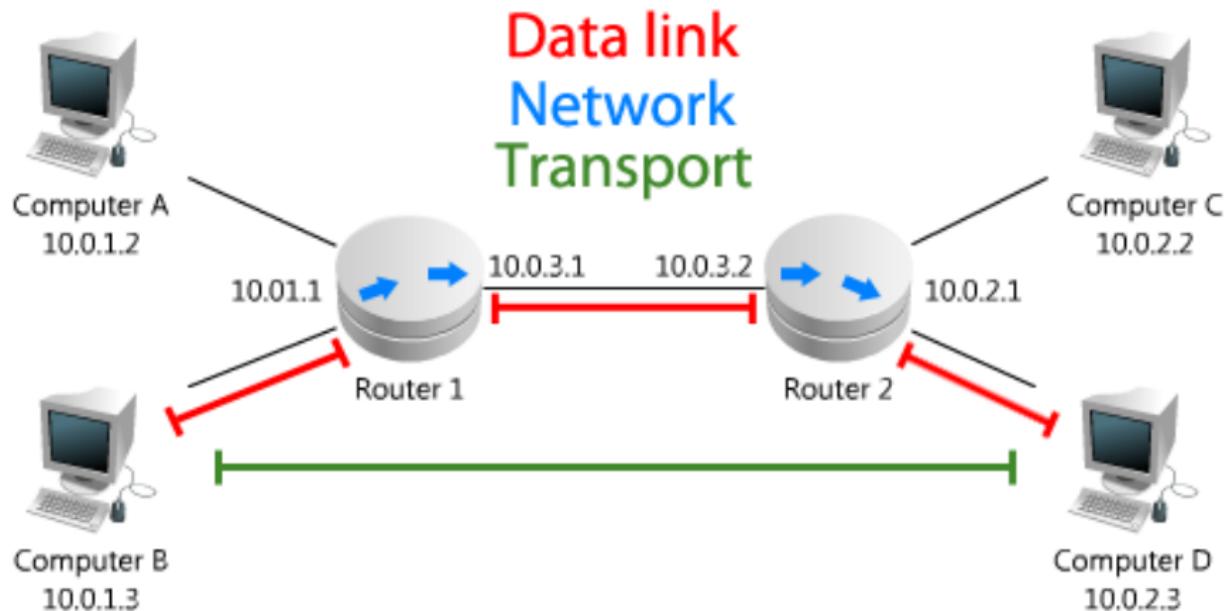


Imagen: OSI model – Layer 4: Transport (TCP and UDP with Scapy)

# 1 Introducción (II)

---

- Se supone un protocolo en la capa de red que puede ser:
  - No fiable: paquetes perdidos, duplicados, desordenados
  - Limita los paquetes a un tamaño finito (MTU)
  - Entrega los paquetes con retardo arbitrariamente largo
- Posibles servicios extremo a extremo:
  - Fiabilidad
  - Permitir mensajes de tamaño arbitrariamente grande
  - El receptor puede controlar el flujo de datos del emisor
  - Reserva de recursos (QoS)

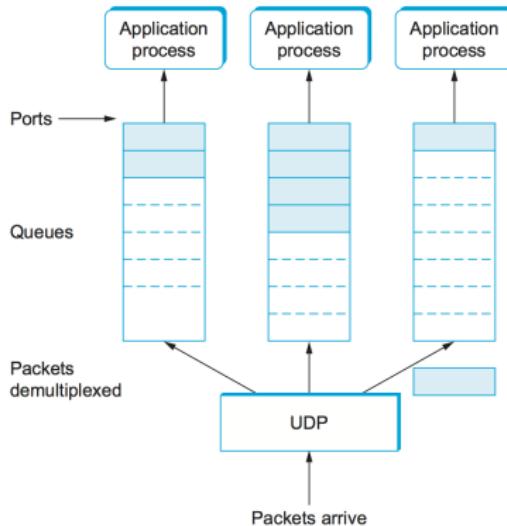
# 1 Introducción (III)

- UDP (User Datagram Protocol)
  - Protocolo de transporte bidireccional «sin conexión»
  - No garantiza un servicio extremo a extremo fiable
  - Usado para transmisión continua de *información actualizada*: juegos en red, DNS, SNMP, etc.
  - Utilizado en multicast
- TCP (Transmission Control Protocol)
  - Protocolo de transporte bidireccional «orientado a conexión»
  - Fiable (checksum + retransmisión)
  - Usado cuando se requiere fiabilidad: FTP, HTTP, Telnet, SMTP, etc.
- RTP (Real Time Protocol)
  - Orientado a aplicaciones de tiempo real: *streaming*, etc.
  - Funciona sobre UDP

## 2 User Datagram Protocol (UDP)



- Servicio de comunicación entre procesos
- Unidad de transferencia: datagrama
- Destinatario no envía confirmación de recepción



## 2.1 Cabecera UDP



**Checksum:** opcional sobre IPv4 y obligatorio sobre IPv6

- pseudo cabecera + cabecera UDP + datos
- pseudo cabecera = IP origen + IP destino +  
IP protocolo (17) + longitud del segmento UDP

**Longitud:** tamaño en bytes del datagrama

- cabecera + datos

**Puertos origen y destino:** números de 16 bits [0, 65535] que asocian comunicaciones de capa de transporte (*sockets*) a procesos en equipos origen y destino

## 2.2 Puertos software

---



- Abstracción para identificar procesos en un nodo
- Se usan en UDP y TCP
- Servidores usan puertos asociados al protocolo, por ejemplo: SSH: 22, DNS: 53, HTTP: 80
  - IANA gestiona correspondencia servicio-puerto
  - RFC 6335 define 3 rangos:
    - ≤1023: sistema, requieren privilegios de administrador
    - 1024–49151: usuario
    - ≥49152: dinámicos/privados
  - Ver fichero /etc/services
- Clientes suelen usar puertos locales aleatorios, asignados por el sistema, *sin usar bind()*

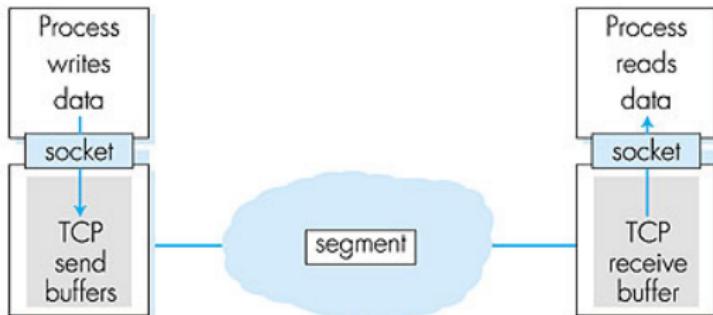
## 3. Transmission Control Protocol (TCP)

- 3.1. Cabecera TCP
- 3.2. Establecimiento/finalización de conexión
- 3.3. Estados de una conexión TCP
- 3.4. Envío de datos
- 3.5. Control de flujo
- 3.6. Opciones TCP
- 3.7. Retransmisión adaptativa
- 3.8. Conclusiones

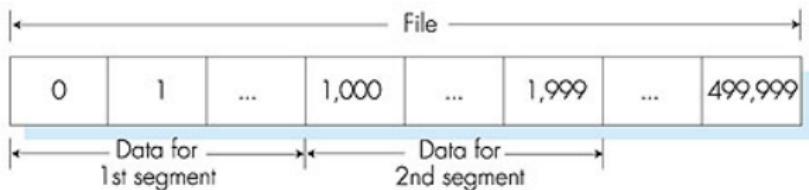
# 3 Transport Control Protocol (TCP)

- Comunicación entre procesos fiable, orientada a conexión
  1. Establecimiento de conexión
  2. Transferencia de datos
  3. Liberación de conexión
- Control de errores (checksum)
- Control de flujo: evita que receptor se desborde (ventana)
- Control de congestión: evita sobrecarga red (flags) [Tema 6]
- Unidad de transferencia: segmento TCP
- Secuenciación de datos *extremo a extremo*
  - Posición de los datos en el mensaje (nº de secuencia)
  - Confirmación de datos correctos (nº de ACK)
  - Retransmisión de un segmento si el emisor no recibe confirmación transcurrido un tiempo desde su envío (tiempo de expiración/retransmission timeout, RTO)

# 3 Transport Control Protocol (TCP) (II)



TCP send and receiver buffers



File of 500 KB with MSS of 1 KB

### 3.1 Cabecera TCP



0	4	7	16	31
Puerto origen		Puerto destino		
Número de secuencia				
Acknowledgment Number				
Long. cab.	0	NS CWR ECE URG ACK PUSH RST SYN FIN	Ventana	
Checksum			Puntero urgente	
Opciones (opcional)				

Puertos origen y destino: identifican extremos de la conexión

Nº de secuencia (SEQ): posición del primer byte de datos del segmento en el mensaje original.

Acknowledgment Number (ACK): siguiente nº de secuencia que se espera recibir.

### 3.1 Cabecera TCP (II)

- SEQ y ACK son relativos a un valor inicial aleatorio (ISN, Initial Sequence Number)

Longitud de la cabecera: medida en palabras de 32 bits

Flag ACK: validez del campo Acknowledgment Number

Flags SYN, FIN, RESET: establecer/finalizar/abortar conexión

Flag URG: avisa de datos urgentes (puntero urgente)

Flag PUSH: fuerza el vaciado de *buffers* en origen y destino

Flags NS, CWR, ECE: control de *Explicit Congestion Notification* [Tema 6]

Checksum: pseudocabecera (=UDP) + cabecera TCP + datos

Ventana: bytes que el receptor puede aceptar

Opciones: parámetros adicionales de la conexión [p. 26]

### 3.2 Establecimiento de conexión

#### ► 3-way handshake

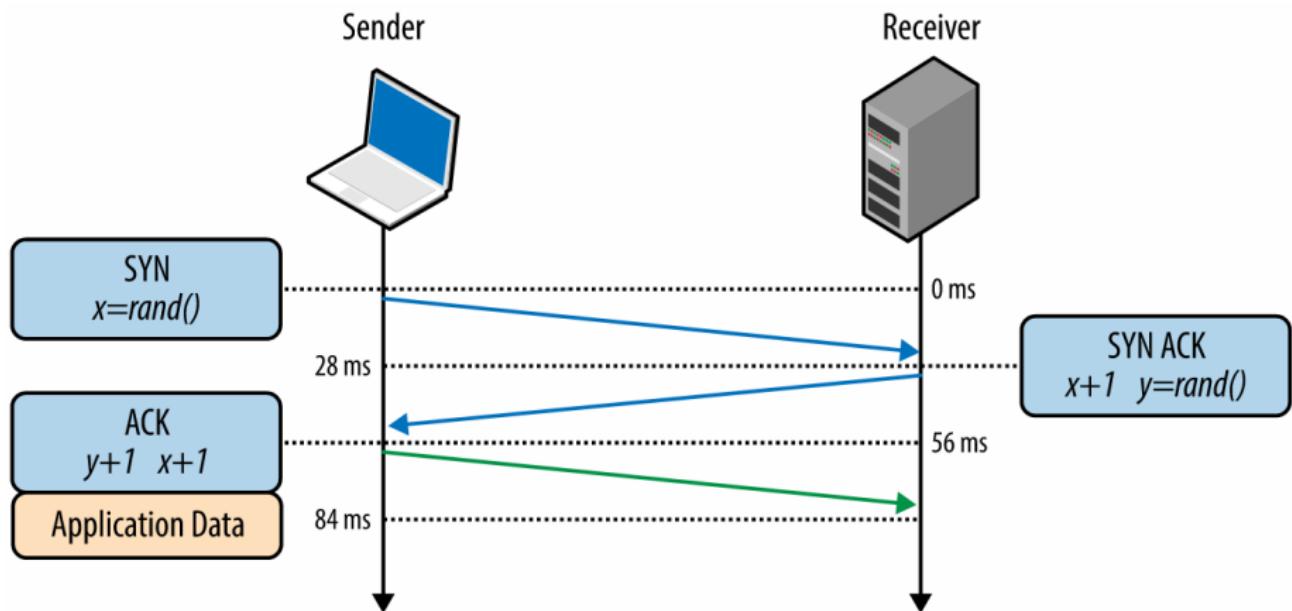
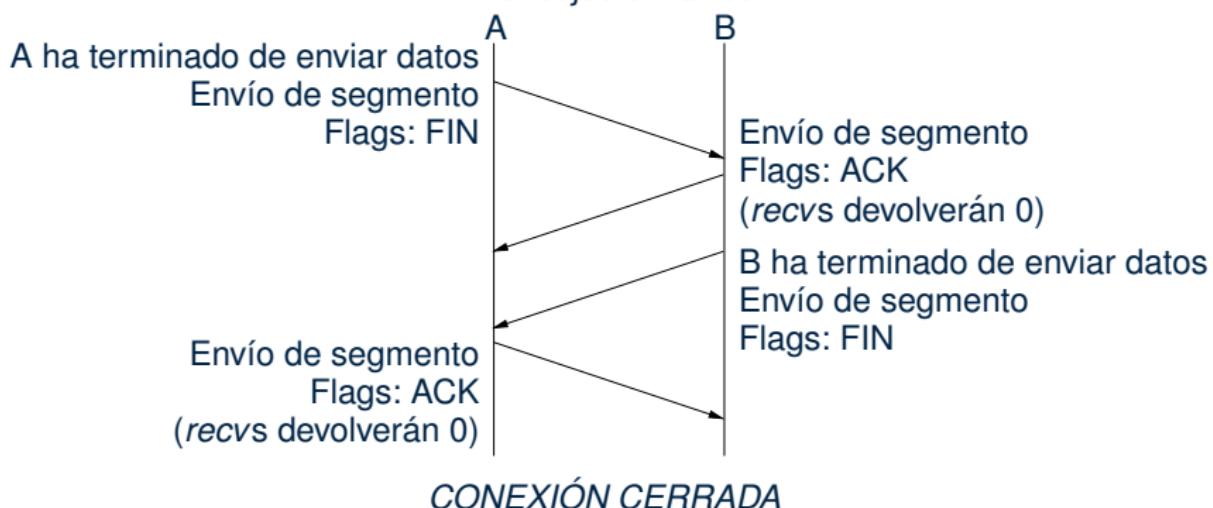


Imagen: High Performance Browser Networking: Building Blocks of TCP

## 3.2 Finalización de conexión

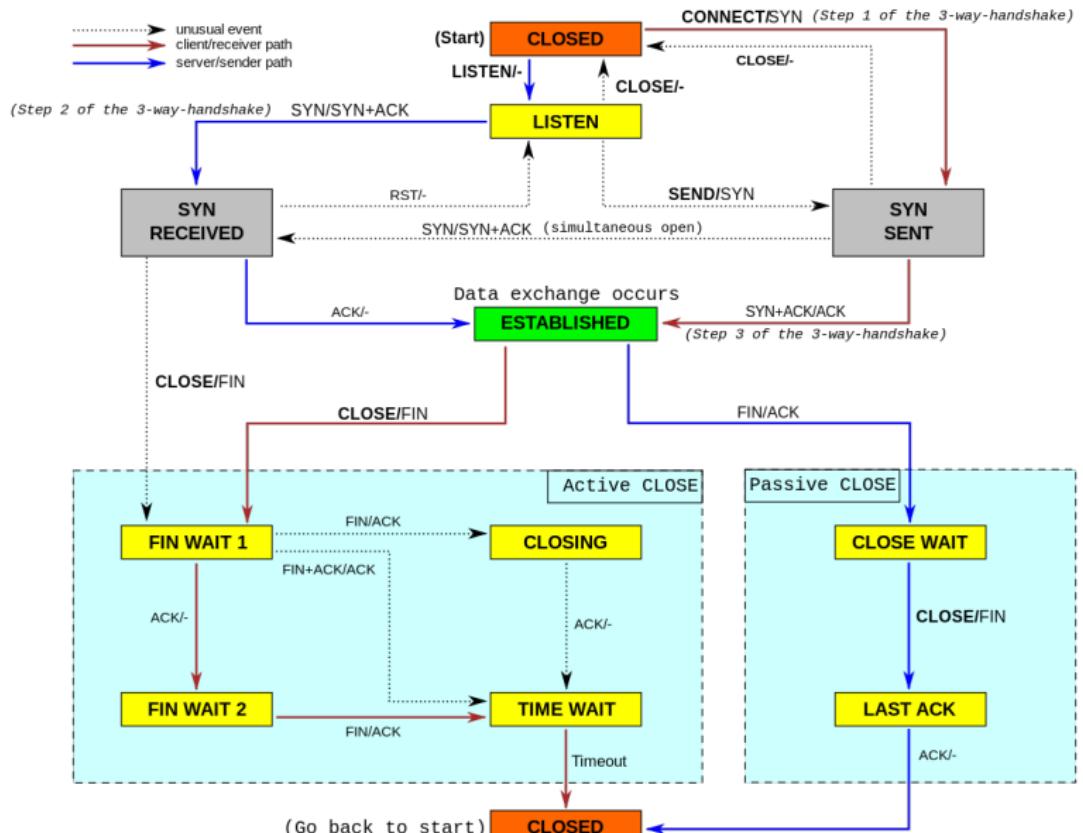
- Desconexión ordenada o consensuada
  - `int close(int fd);`
  - `int shutdown(int sockfd, int how); //SHUT_RD/WR/RDWR`

*Mensajes en la red*



- Desconexión desordenada o unilateral (flag RESET)

### 3.3 Estados de una conexión TCP



### 3.4 Envío de datos

- *Maximum Segment Size (MSS)*: máximo volumen de datos TCP que pueden enviarse en un segmento.
  - Normalmente se inicializa al mayor tamaño que no requiere fragmentación IP:  $MSS = MTU - \text{long\_cabeceras\_TCP/IP}$
  - Por ejemplo, para Ethernet:  $MSS = 1500 - 40 = 1460 \text{ bytes}$

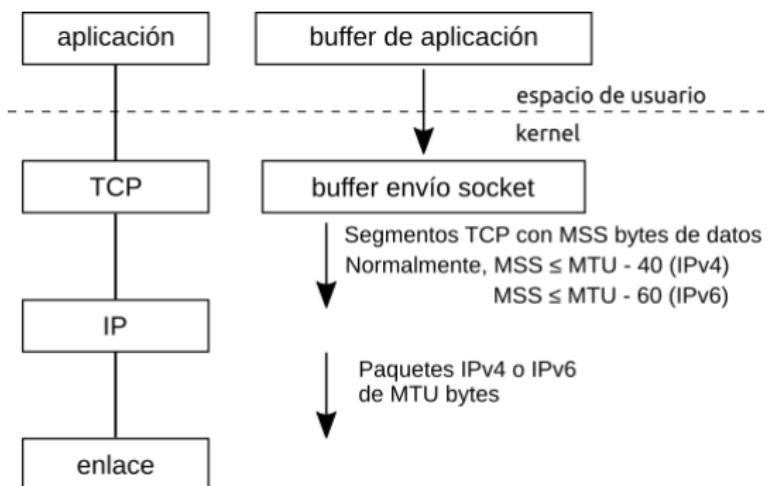


Figura: Buffers involucrados cuando una aplicación escribe en un socket TCP.  
Adaptado de Figura 2.11 en Stevens. Unix Network Programming. Vol.1, 2<sup>a</sup> ed.

## 3.4 Envío de datos (II)

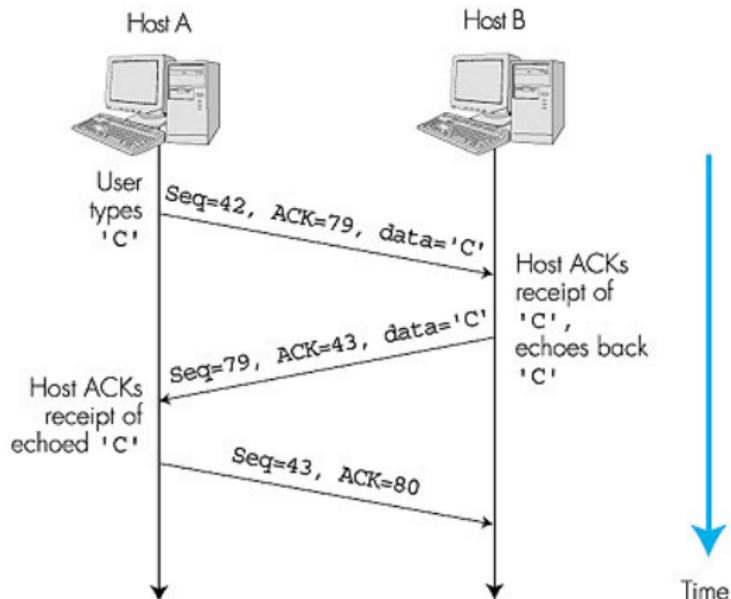


Figura: Envío de datos de cliente a servidor y acuses de recibo en sentido contrario.  
Fuente: Sequence and acknowledgement numbers for simple Telnet application over TCP

### 3.4 Envío de datos (III)

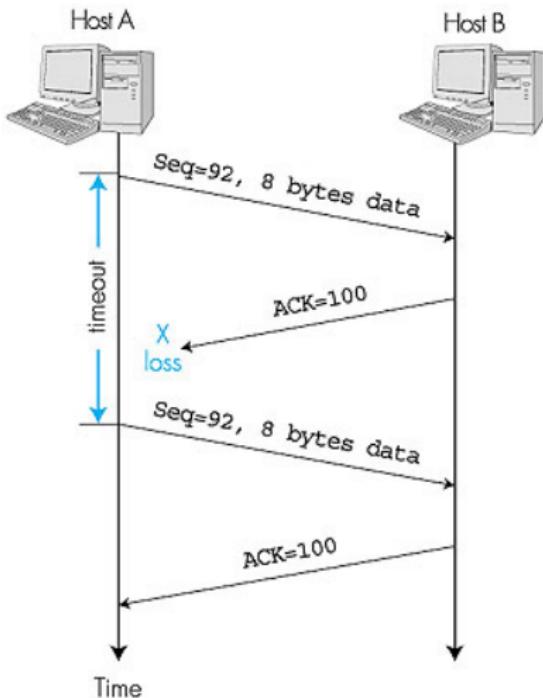


Figura: Envío de datos de cliente a servidor y acuses de recibo en sentido contrario.  
Fuente: Retransmisión por pérdida de ACK

## 3.4 Envío de datos (IV)

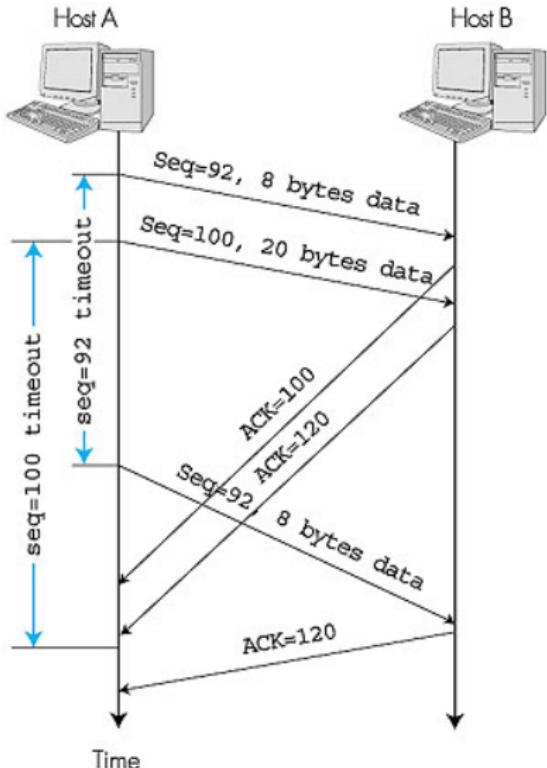


Figura: Envío de datos de cliente a servidor y acuses de recibo en sentido contrario.  
Fuente: [Retransmisión y Timeout](#)

## 3.4 Envío de datos (V)

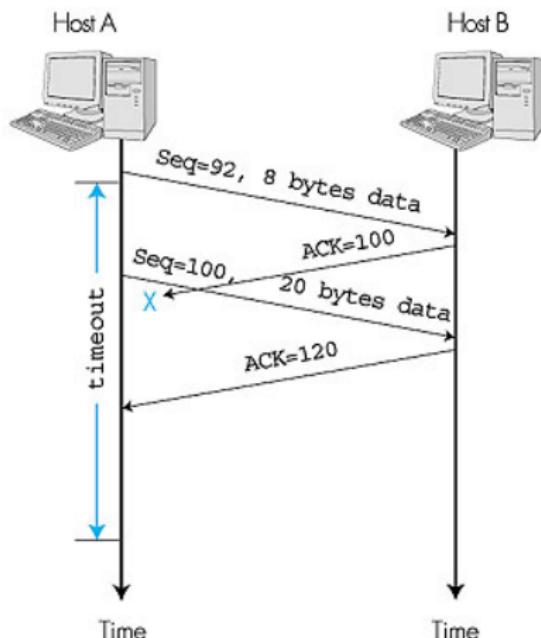
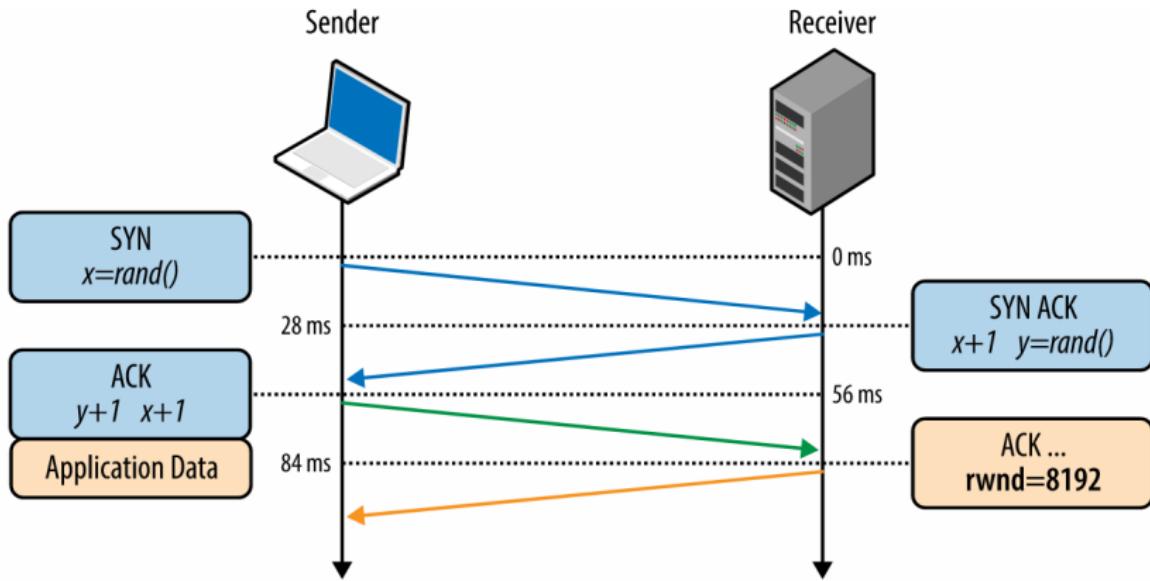


Figura: Envío de datos de cliente a servidor y acuses de recibo en sentido contrario.

Fuente: Retransmisión y ACK acumulado

## 3.5 Control de flujo

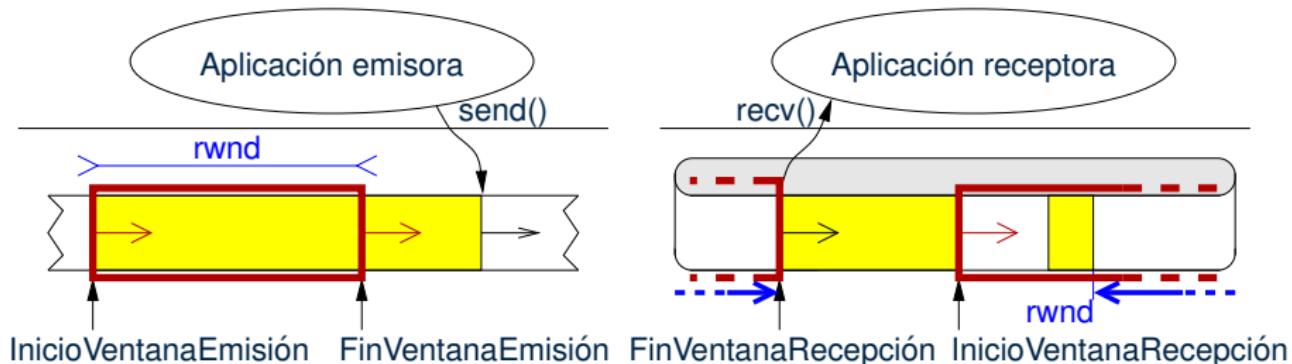
- Objetivo: impedir que el emisor desborde al receptor
- Ventana de recepción, *rwnd*: en cada ACK el receptor indica el número de bytes que puede aceptar
  - Rango números de secuencia:  $ackn : ackn + rwnd - 1$



Fuente: High Performance Browser Networking: Building Blocks of TCP

### 3.5 Control de flujo (II)

- Ventana deslizante de tamaño variable,  $rwnd$
- Receptor informa del espacio libre de su buffer en el campo *Ventana*:  $rwnd = BufferSize - Ocupado$
- Emisor tiene un límite de datos enviados sin confirmar:  $rwnd \geq FinVEmis - IniVEmis$
- Si  $rwnd = 0$ , envíos de tamaño mínimo cada cierto tiempo



### 3.5.1 Ejercicios con control de flujo



#### ► Variables en juego:

- $\text{AdvertisedWindow} = \text{MaxRcvBuffer} - ((\text{NextByteExpected} - 1) - \text{LastByteRead})$
- $\text{LastByteSent} - \text{LastByteAcked} \leq \text{AdvertisedWindow}$
- $\text{EffectiveWindow} = \text{AdvertisedWindow} - (\text{LastByteSent} - \text{LastByteAcked})$
- $\text{LastByteWritten} - \text{LastByteAcked} \leq \text{MaxSendBuffer}$

✍ Determina las características de un enlace TCP considerando que el emisor siempre tiene datos a escribir y el receptor lee los datos en cuanto llegan y el RTT se puede aproximar a 0:

- Be=32 KiB, Br=16 KiB: ¿*rwnd, ewnd*?
- Be=16 KiB, Br=32 KiB: ¿*rwnd, ewnd*?

### 3.5.1 Algoritmo de Nagle

---

```
if (hay más de MSS bytes de datos) y  
    (la ventana lo permite) then  
        enviar MSS bytes  
    else  
        if (se espera algún ACK) then  
            acumular datos en buffer  
        else  
            enviar datos del buffer  
        end if  
    end if
```

---

- Recepción de ACK → activación envío
- Puede desactivarse: opción TCP\_NODELAY en setsockopt()

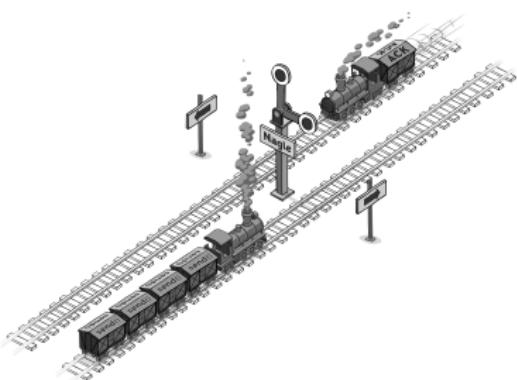


Imagen: <http://ithare.com/tcp-peculiarities-as-applied-to-games-part-ii/>

## 3.6 Opciones TCP

- RFC 7323
- Las funcionalidades poco usadas y las nuevas se implementan como opciones en el protocolo básico
- Cada extremo TCP puede incluir opciones en un segmento SYN. Las más comunes son:
- *Escalado ventana de recepción TCP (rwnd):*
  - $rwnd$ : 16 bits en cabecera  $\rightarrow rwnd_{max} = 64KiB$
  - Conexiones alta velocidad ( $v_t$ ) o latencia elevada ( $RTT$ ) requieren ventanas mayores:  $V_t \cdot RTT$ 
    - Por ejemplo, 100 Mbps con RTT de 96 ms  $\rightarrow rwnd = 1.2 MiB$
  - $rwnd = rwnd << nbits = rwnd \cdot 2^{nbits}$ , con  $nbits = [0 - 14]$ ,  
 $\rightarrow rwnd_{max} = 65535 \cdot 2^{14} = 1 GiB$

## 3.6 Opciones TCP (II)



- *Marca de tiempo TCP (timestamp)*: 32 bits que permiten
  - Medir RTT: *Round-Trip Time Measurement (RTTM)*
  - Extender el rango de números de secuencia (*nseq*) para no reusar valores antes de *Maximum Segment Lifetime (MSL)*:  
*PAWS - Protection Against Wrapped Sequences*
- *PAWS*
  - Con *nseq* de 32 bits →  $2^{32} = 4 \text{ GiB}$  y MSL entre 1 y 2 min
    - 100 Mbps → reúso en 5.7 min, 10 Gbps → reúso en 3.4 s!
  - Orden de segmentos especificado por combinación de campos <marca de tiempo, número de secuencia>
- Nota: *Maximum Segment Life (MSL)* es el tiempo que un segmento puede existir en la red
  - Según **RFC 793**: 2 minutos

### 3.7 Retransmisión adaptativa



- El retardo en la recepción de un ACK es muy variable
- El tiempo de expiración (*retransmission timeout, RTO*) debe ajustarse al RTT:
  - Retransmitir demasiado tarde infrautiliza la red
  - Retransmitir demasiado pronto añade sobrecarga

Objetivo del emisor: para cada envío, establecer su tiempo de expiración (RTO) en función del RTT estimado (*EstimRTT*)

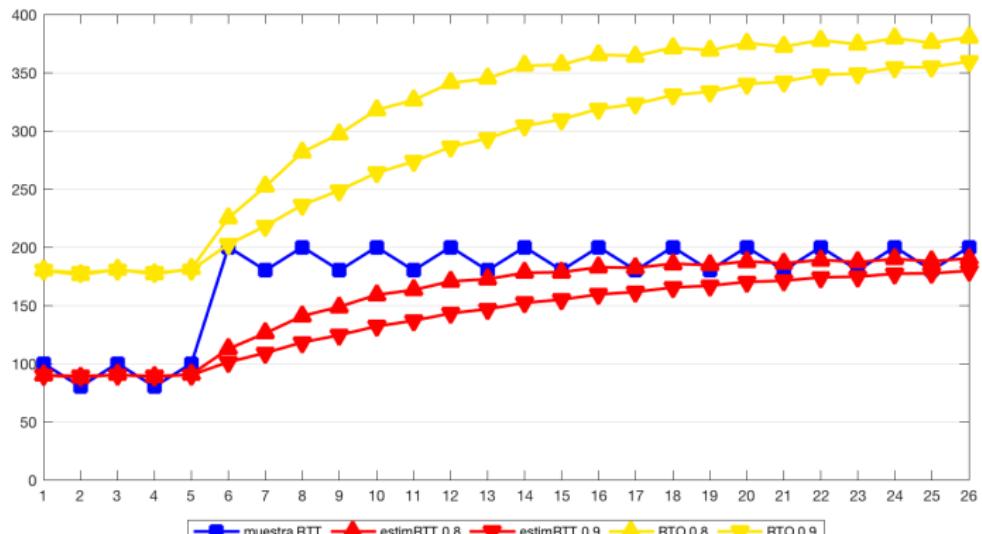
1. Cuando llega ACK, mide el tiempo que ha tardado desde su correspondiente envío: *MuestraRTT*
  2. Estima el RTT previsto para el envío: *EstimRTT*
  3. Establece el timeout para el envío: *RTO*
- NO se toma muestra cuando se ha producido un reenvío
  - ¿Valor de RTO?

### 3.7.1 Algoritmo TCP original

- RFC 793: cálculo sencillo del tiempo de expiración

$EstimRTT = \alpha \cdot EstimRTT + (1 - \alpha) \cdot MuestraRTT$   
con  $\alpha \in [0.8, 0.9]$

$RTO = \beta \cdot EstimRTT$ , con  $\beta \in [1.3, 2.0]$



## 3.7.2 Algoritmo Jacobson/Karels

---



- Incorpora varianza de RTT a la estimación

$$DifRTT = MuestraRTT - EstimRTT$$

$$EstimRTT += \delta \cdot DifRTT$$

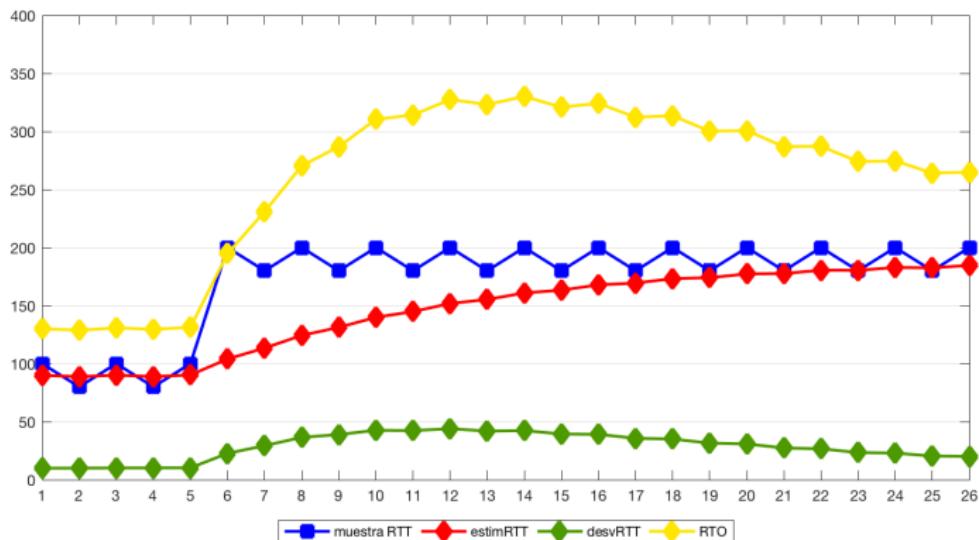
$$DesvRTT += \delta(|DifRTT| - DesvRTT)$$

$$RTO = \mu \cdot EstimRTT + \phi \cdot DesvRTT$$

$$(\delta \in [0, 1], \mu = 1, \phi = 4)$$

### 3.7.2 Algoritmo Jacobson/Karels (II)

Ejemplo Jacobson/Karels: paso de RTTs de 80-100 ms a 180-200 ms,  $\delta = 1/8$ . Se ignora el efecto de paquetes perdidos.



## 3.8 Conclusiones

---



Característica	UPD	TCP
<b>Fiabilidad</b>		
<b>Velocidad</b>		
<b>Datos acotados</b>		
<b>Conexión</b>		
<b>Orden</b>		
<b>Overhead</b>		
<b>Multicast</b>		

- En una conexión TCP, si el número de secuencia de un segmento de esta conexión es  $m$ , entonces ¿el número de secuencia del siguiente segmento será  $m + 1$ ?