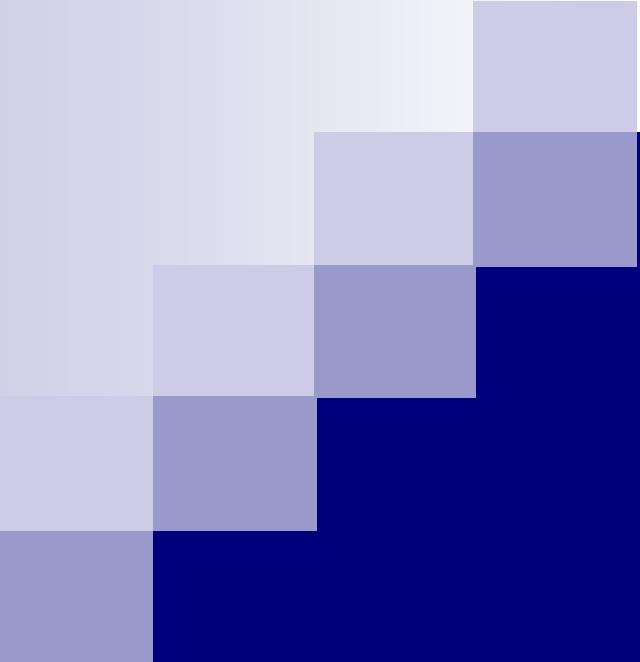


# Redes de Computadoras



**Universidad de Cuenca**

1867



# Capítulo 1 Introducción

# Contenido

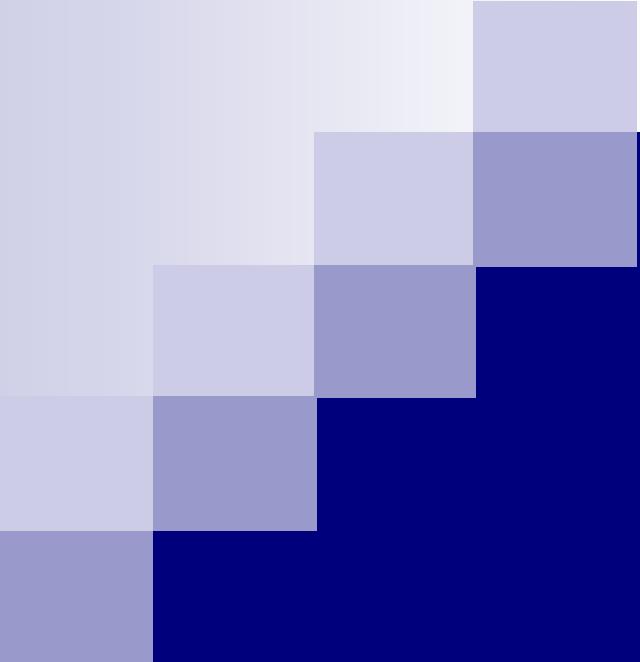
1.1 Generalidades

1.2 Usos de las redes de computadoras

1.3 Hardware de redes

1.4 Software de redes

1.5 Modelos de referencia



## 1.1 Generalidades

# Redes de computadoras

- Modelo antiguo: una sola gran computadora haciendo todas las tareas de una empresa
- Modelo actual: muchas computadoras pequeñas, **interconectadas** hacen, en conjunto, las mismas tareas

# Modelo antiguo



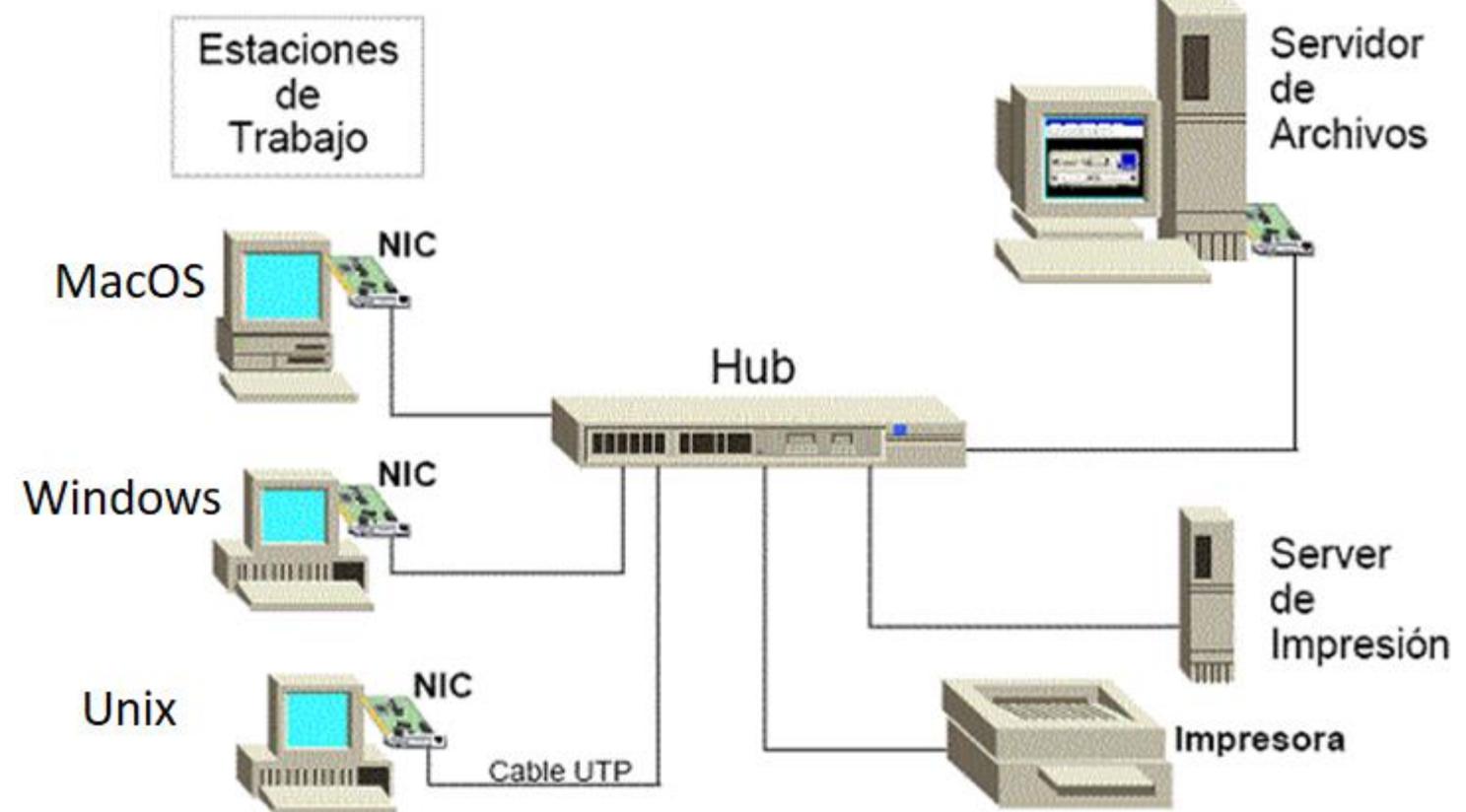


Ilustración 1- 1: Elementos de una red LAN

# Medios de transmisión de datos

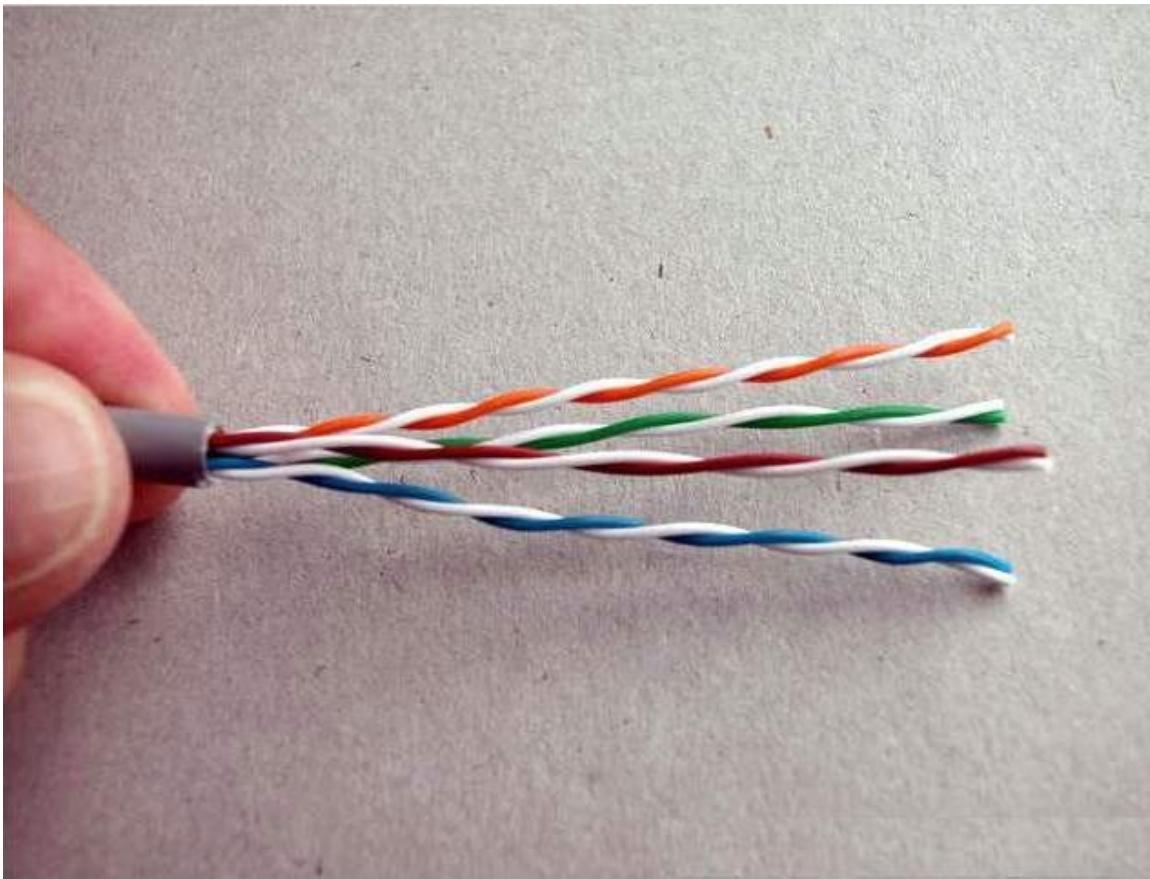
## ■ **Medios guiados. Utilizan materiales sólidos, hilos o alambres**

- Cable de cobre: coaxial, par trenzado
- Fibra óptica: multimodo y monomodo

## ■ **Medios no guiados. No utilizan alambres**

- Infrarrojo
- Láser
- Microonda: bandas licenciadas y bandas libres ISM (Industrial, Scientific and Medical)
- Satélite

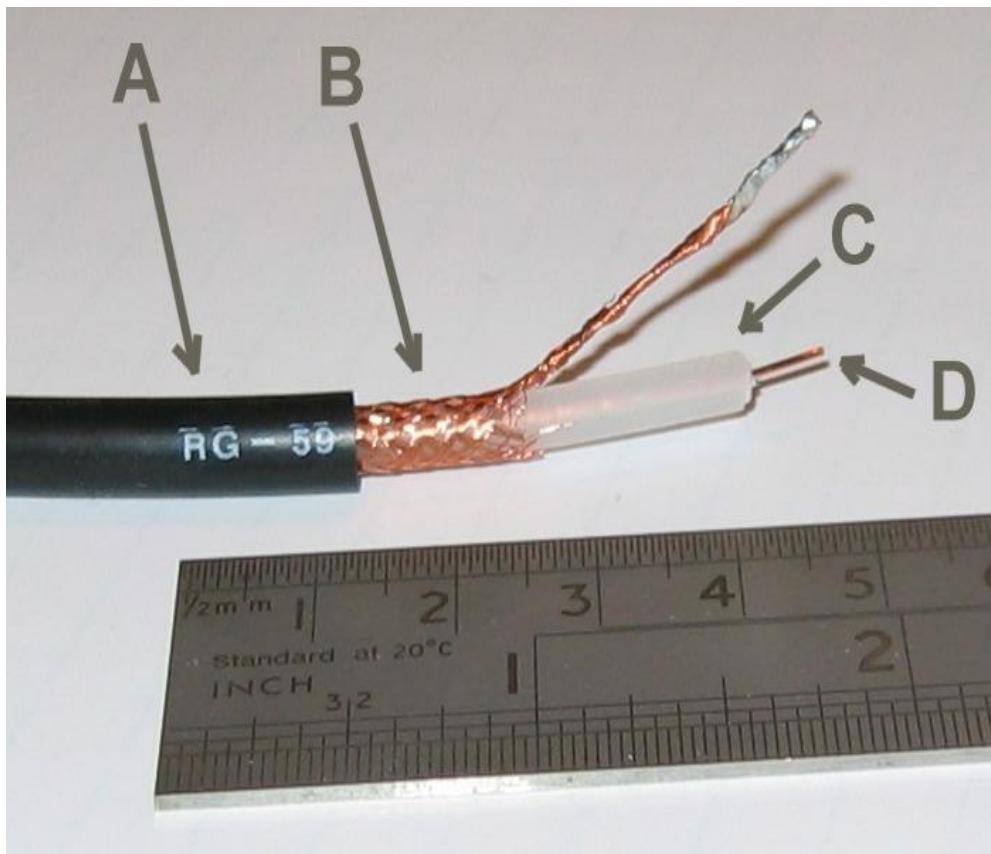
# Cable par tranzado UTP



Cats: 3, 4, 5, 6, 7

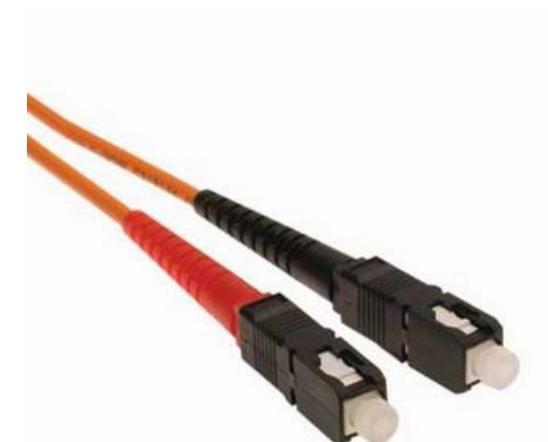
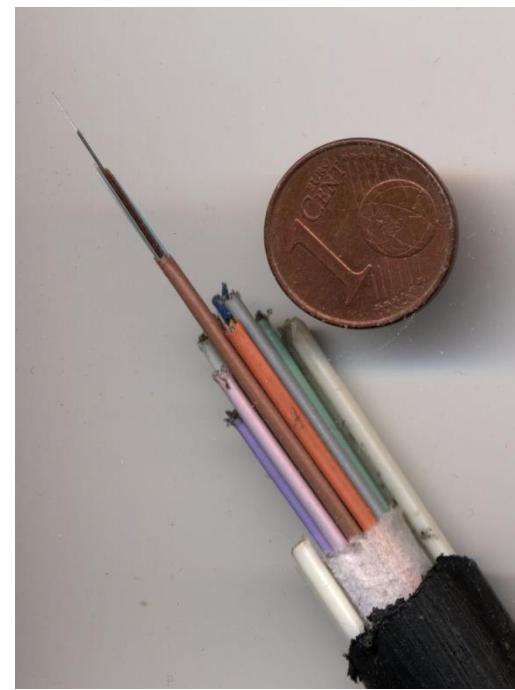
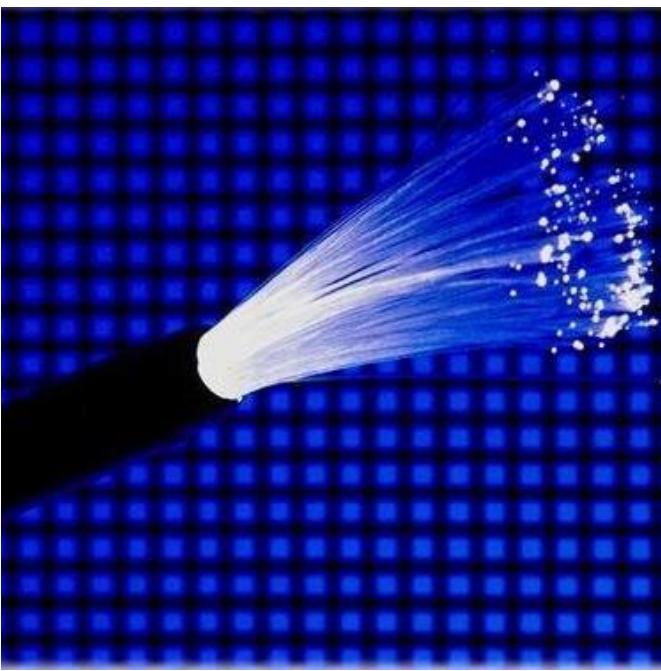


# Cable coaxial RG-59. Transmisión en banda ancha (CATV)

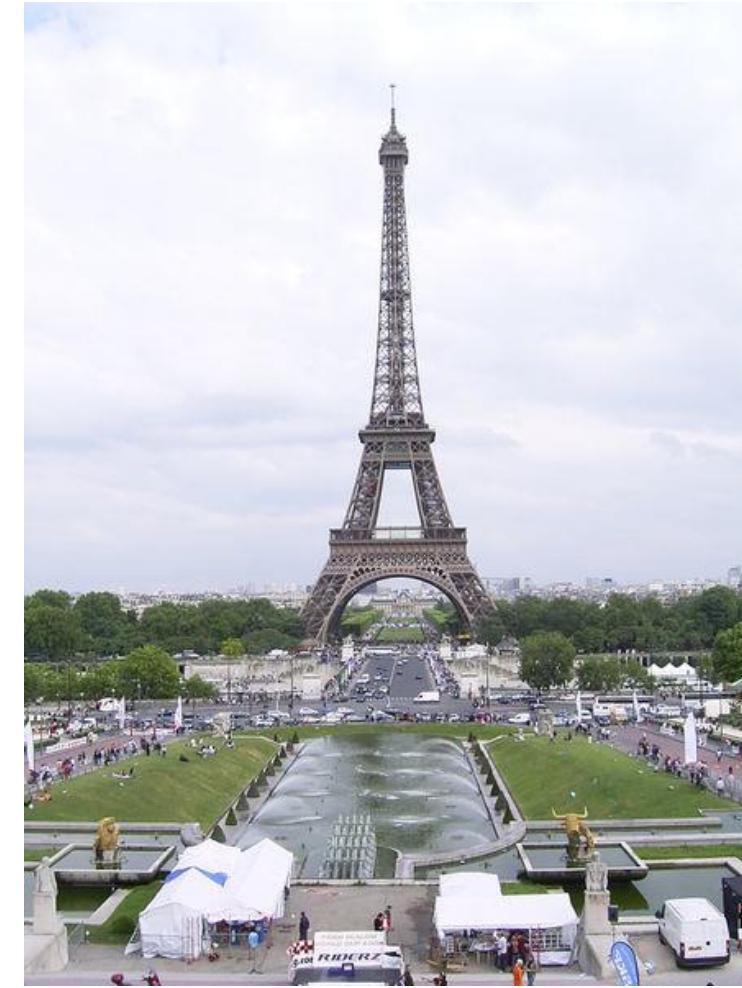


Conectores BNC

# Cable de fibra óptica



# Enlace inalámbrico



Exposición Universal 1889 300 m.  
Antena de radio 24 m

## Estación terrena. Enlace de microondas con satélite geoestacionario

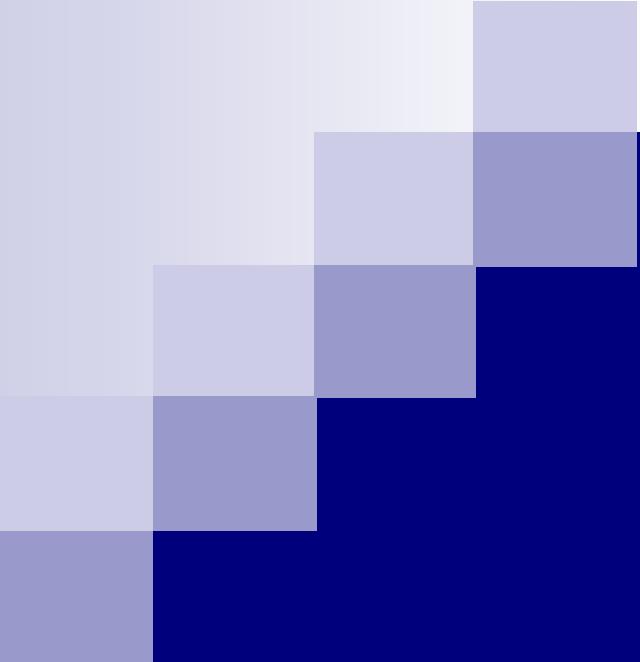


# Sistema distribuido SD

- Conjunto de computadoras que se presenta ante el usuario como un sistema único
- Es un **software** construido sobre una red de computadoras
- Ejemplo: WWW World Wide Web

# Red de computadoras RC

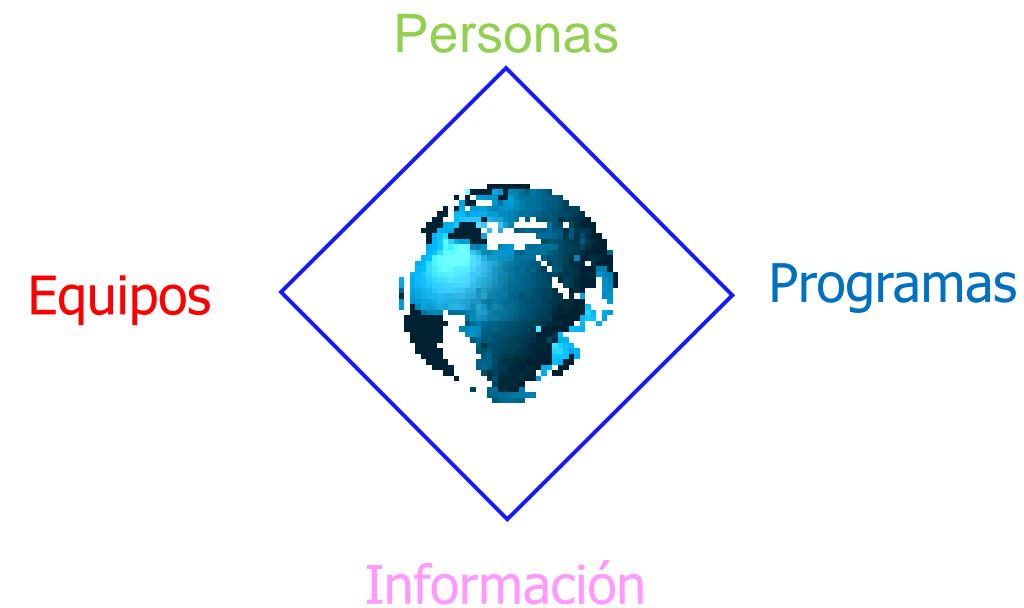
- Usuarios conscientes que hay múltiples computadoras
- Conexión explícita con una de ellas
- No hay el sw que dé transparencia a la red
- La diferencia entre SD y RC está en el sw



## 1.2 Usos de las redes de computadoras

## 1.2.1 Aplicaciones de negocios

- La idea es compartir recursos en una oficina o alrededor del mundo:



# ¿Qué información?

- Nómina
- Clientes
- Proveedores
- Contabilidad
- Inventarios
- Sistemas de salud
- Afiliados
- Unidades educativas
- Ciudadanos
- Sistemas de votación
- Sistemas de información geográfica
- Redes de agua potable
- Redes eléctricas
- Redes telefónicas
- Red vial

# ¿En dónde se usan las redes?

## Sector privado

- Bancos
- Plantas industriales
- Aerolíneas y agencias de viajes
- Comercios
- Sector turístico
- Centros educativos

## Sector público

- Ministerio de Salud
- IESS
- Registro civil
- CNE Consejo Nacional Electoral
- CNT Consejo Nacional de Telecomunicaciones
- ANT Agencia Nacional de Tránsito
- Empresas eléctricas: CentroSur
- Fuerzas armadas
- Policía Nacional
- Municipalidades

# Una red de computadoras permite la comunicación entre personas:

Correo electrónico



Videoconferencia



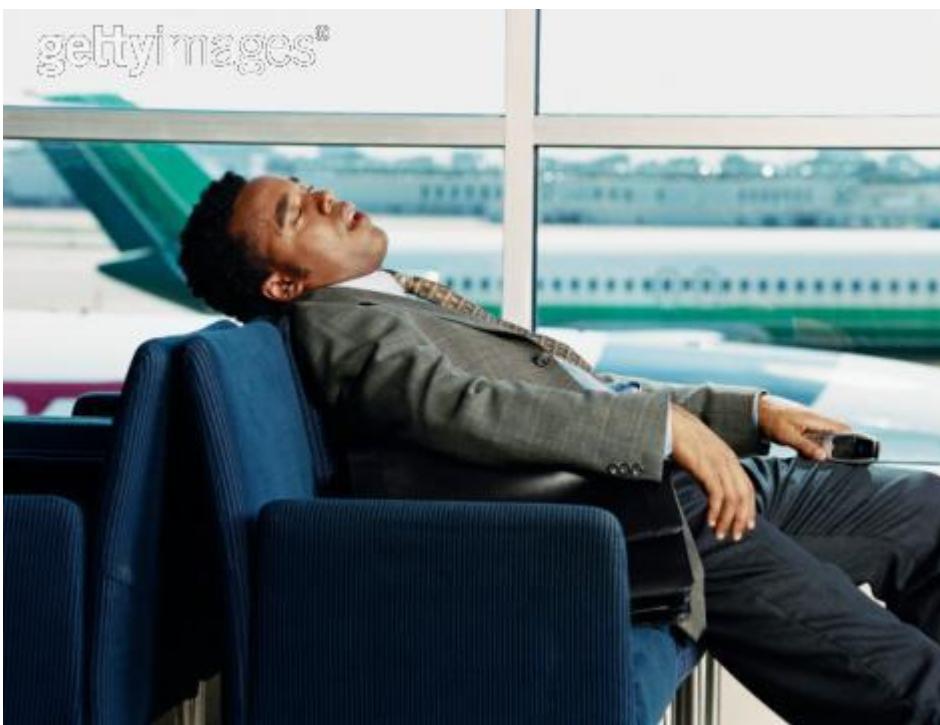
Redes sociales



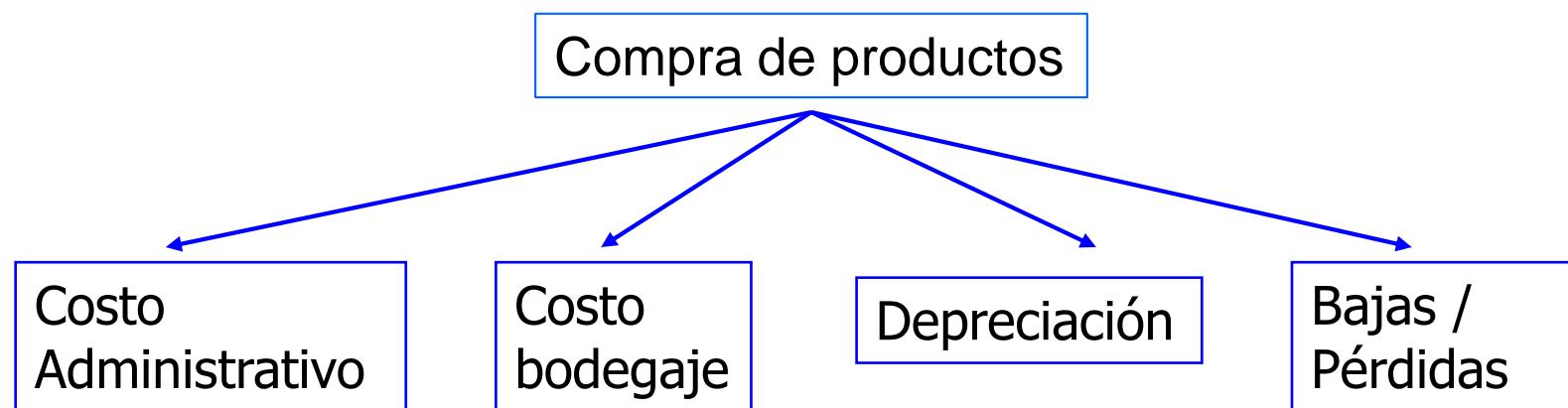
Voz sobre IP



- Las comunicaciones y los sistemas de transporte están en competencia. quien gane hará obsoleto al otro



- Una meta herramienta para proveedores y consumidores es e-commerce
- Pedidos en **tiempo real** conforme se requiera, sin tener grandes inventarios



# Algunas formas de comercio electrónico :

B2C	Business to client	Pedido de un artículo
-----	--------------------	-----------------------



B2B	Business to Business	compra entre fabricantes
-----	----------------------	--------------------------



G2C	Government to Client	Trámites burocráticos
-----	----------------------	-----------------------



C2C	Client to Client	Subasta de productos usados:
-----	------------------	------------------------------



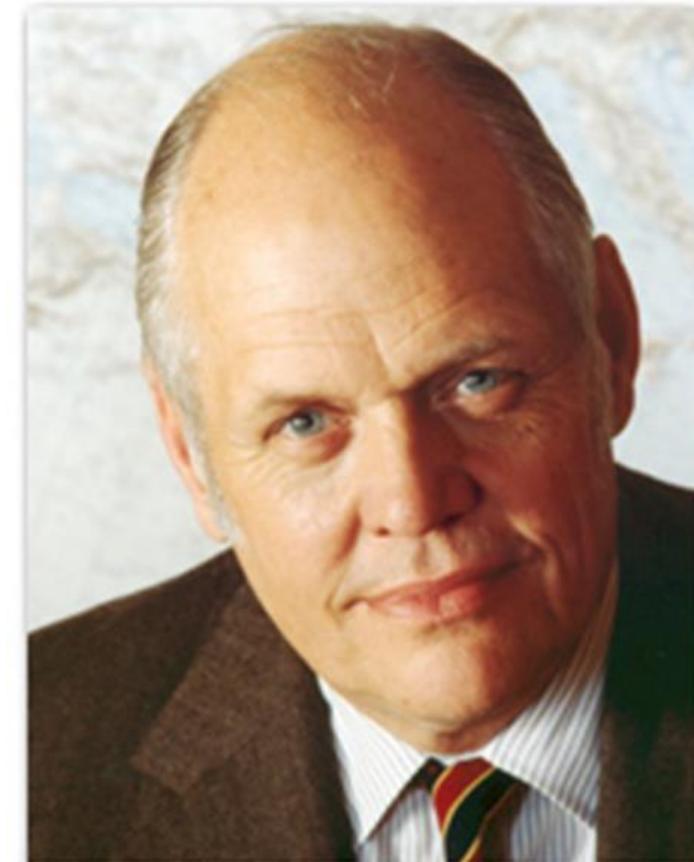
## 1.2.2 Aplicaciones domésticas

Mercado de las PCs a gran escala:

Ken Olsen, Presidente Ejecutivo de Digital Equipment Corporation DEC. 1977

*“No hay razón alguna para que un individuo tenga una computadora en casa”*

Hoy Digital ya no existe

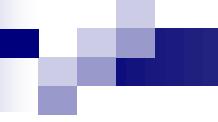




Thomas J. Watson  
Presidente de IBM en 1945

De por qué IBM no entraba al negocio de las computadoras:

*“Cuatro o cinco computadoras deberán ser suficientes en todo el mundo hasta el año 2000”*



- La razón más importante de tener una computadora es el acceso a Internet

- Acceso a información remota (web)
  - Comunicación de persona a persona
  - E-commerce
  - E-government
  - E-learning
  - E-banking
  - Banca móvil
  - Entretenimiento interactivo

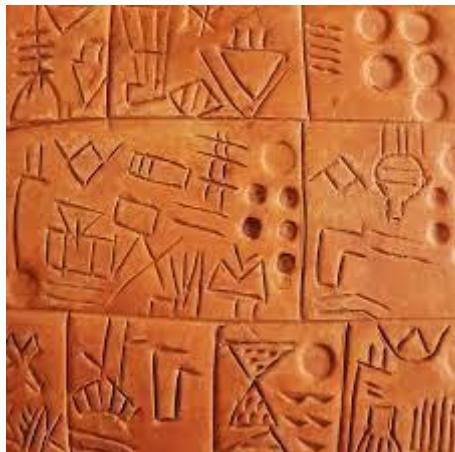
- Periódicos en línea que pueden personalizarse: política, economía, ciencia y tecnología, deportes, cocina, farándula, etc.
- Solucionan el problema de distribución, punto débil en la cadena de producción
- Hay necesidad de suscribirse al servicio



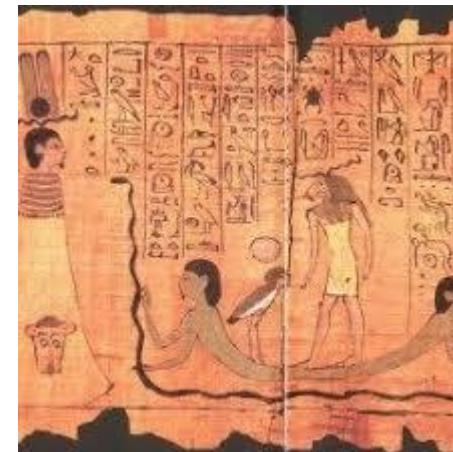
# Libros digitales

- Libros digitales.
- Dicen que los libros **impresos** llegarán a ser obsoletos

# Soporte de la escritura



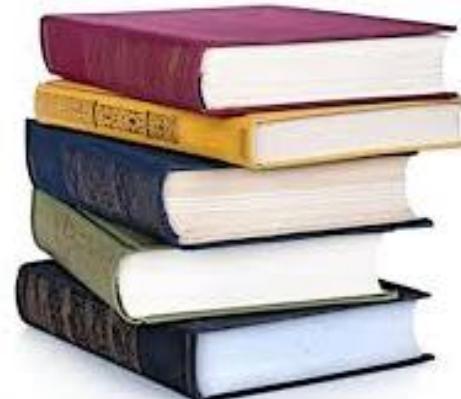
Tablilla de arcilla  
Edad antigua 3500 AC



Papiros. Edad Antigua  
3000 AC – 500 DC



Manuscritos en pergamino  
Edad Media. Siglos V al XV



Libros impresos  
Año 1450



Libros digitales  
Año 200?

# Vídeo bajo demanda: streaming (transmisión en vivo)



# Más aplicaciones

- Teletrabajo
- Telemedicina
- Televigilancia. Cámaras IP: plantas industriales, casa, guardería, seguridad ciudadana
- Monitoreo vehicular: repartidores, recolectores, vendedores, buses, taxis, etc.
- Parqueo tarifado

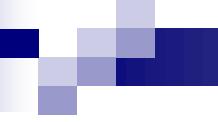
## 1.2.3 Usuarios móviles



# Dispositivos móviles

- Oficina portátil: celular + laptop.
- Ejecutivos se conectan por red celular o WiFi para navegar en Internet
  - Metro
  - Tren a 300 km/h
  - Vehículo a 120 km/h
  - Barco
  - Avión a 900 km/h





## ■ WiFi en muchos lugares públicos:

- Hoteles
- Aeropuertos
- Malls
- Auditorios
- Campus universitarios
- Plazas y parques públicos

# Smart phone

- Acceso a Internet
- Localización de comercios
- Determinación de la mejor ruta a un destino
- Mapas móviles
- Monitoreo de pacientes críticos

- Virus. Desde 1972. El mayor número de virus se da en Windows. En Mac y Linux casi no lo hay
- Es necesario la seguridad:

- Confidencialidad



- Autenticación



- Autorización

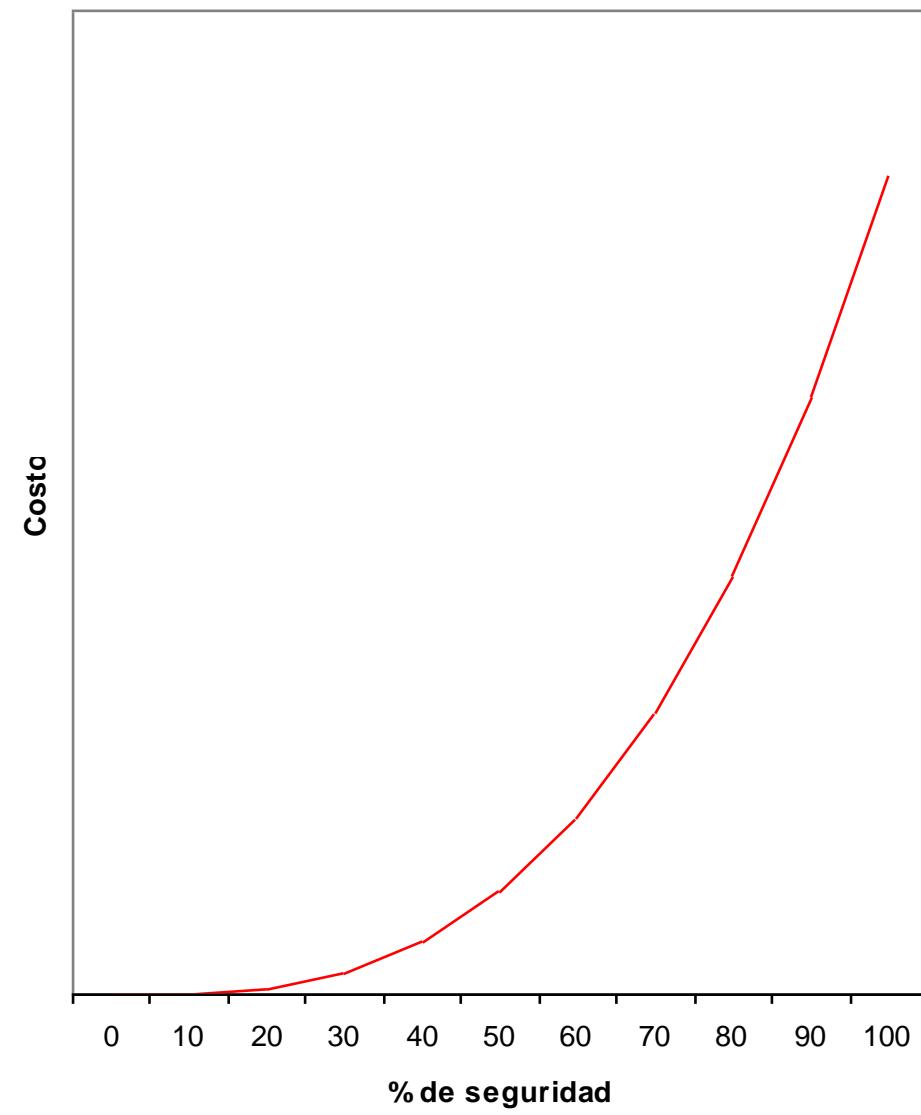


- Irrebatibilidad



- La seguridad cuesta cada vez más

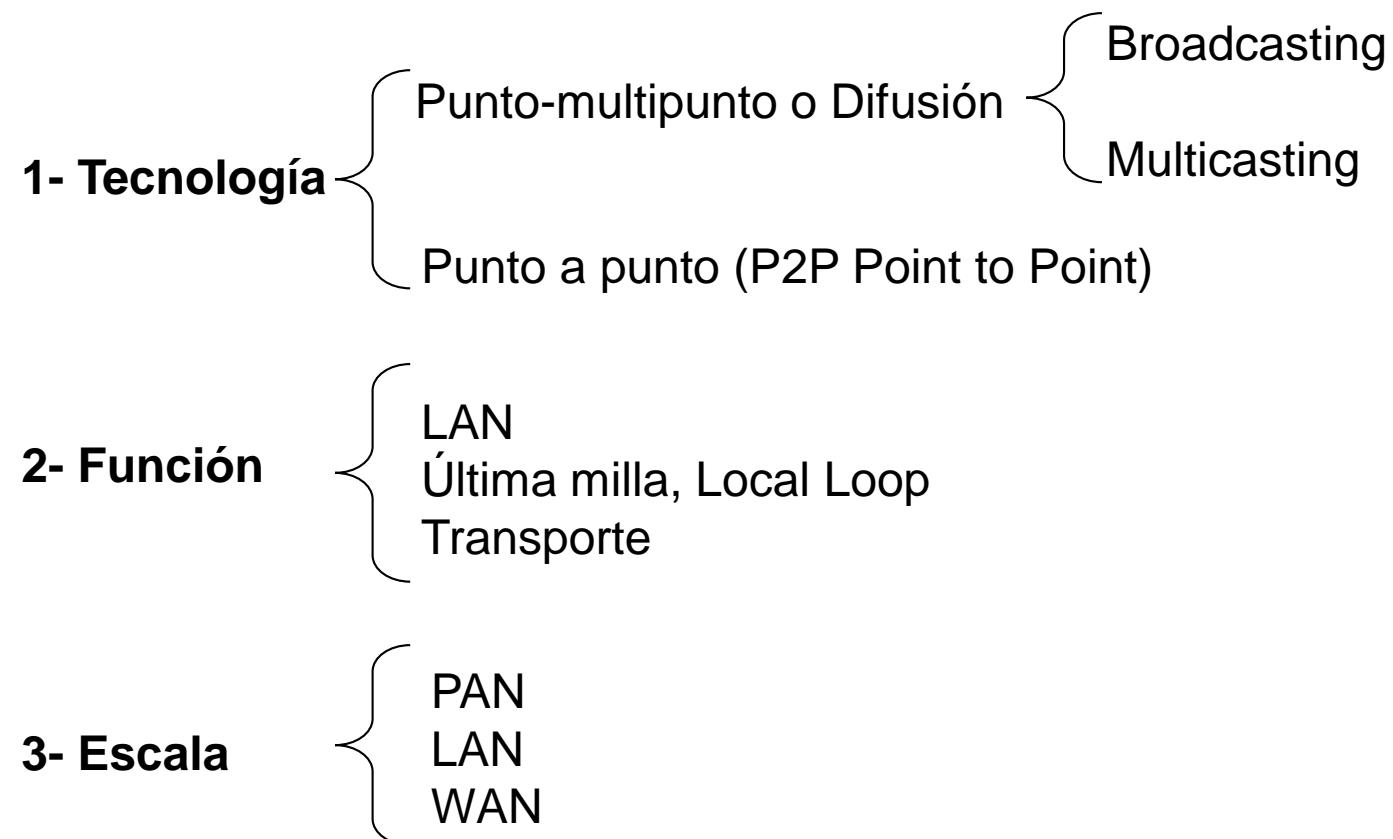
### Seguridad vs. costo





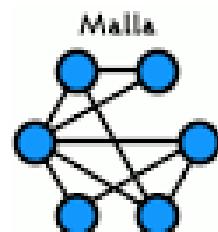
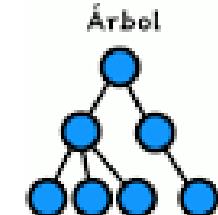
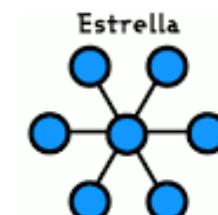
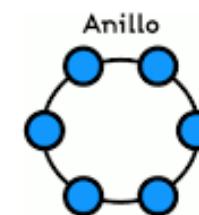
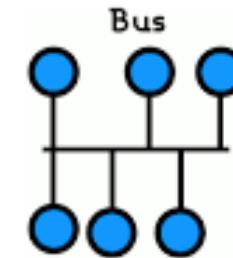
## 1.3 Hardware de redes

# Criterios de clasificación de las redes:



# Topologías

- Forma en la que están conectados los dispositivos en una red
- Tecnología punto-multipunto: topología bus
- Tecnología punto a punto. Topologías:
  - anillo,
  - estrella,
  - árbol,
  - malla

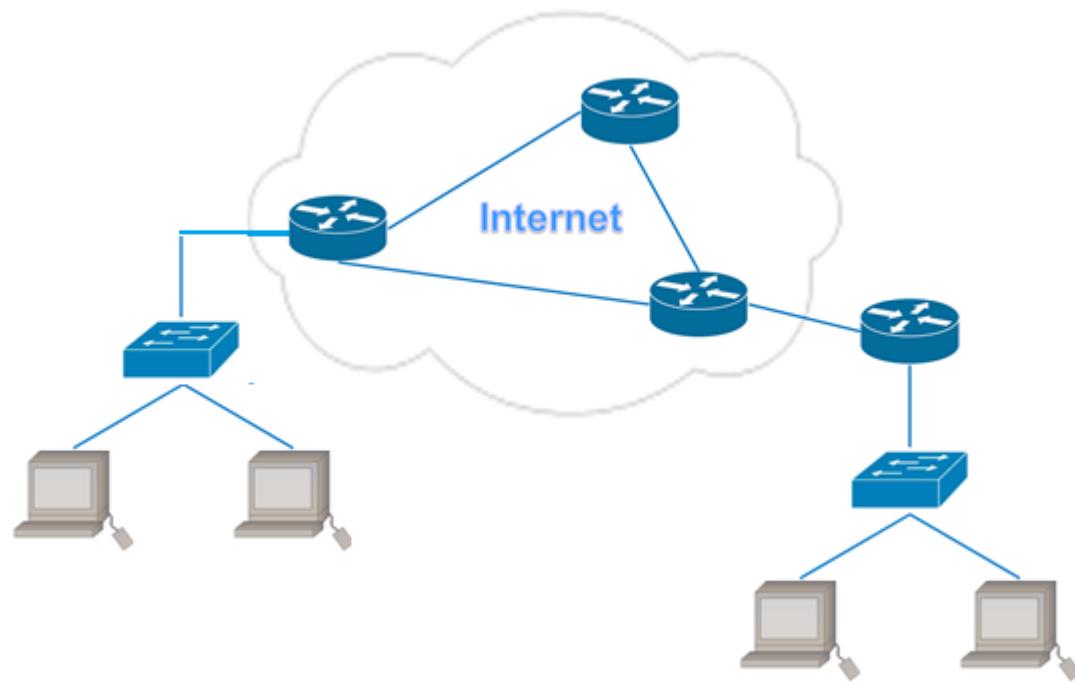


# Broadcast o enlaces de difusión

- Topología punto - multipunto
- Un sólo canal de difusión
- Medio compartido por todas las máquinas
- El mensaje llega a todas las máquinas
- Paquete con la dirección destino
- Solo el destinatario procesa el paquete
- Las otras máquinas lo ignoran

- **Difusión o broadcasting**: envío a todos los destinos a través de una dirección especial
- **Multidifusión o multicasting**: envío sólo a un subconjunto de máquinas

# Point-to-Point P2P



- Topología punto a punto
- Conexiones entre pares de máquinas
- El paquete pasa por máquinas intermedias
- Pueden haber varias rutas
- Se necesita determinar la mejor ruta
- Redes grandes usan P2P

# Clasificación de redes por su función

- LAN
- Última milla o de acceso
- De transporte

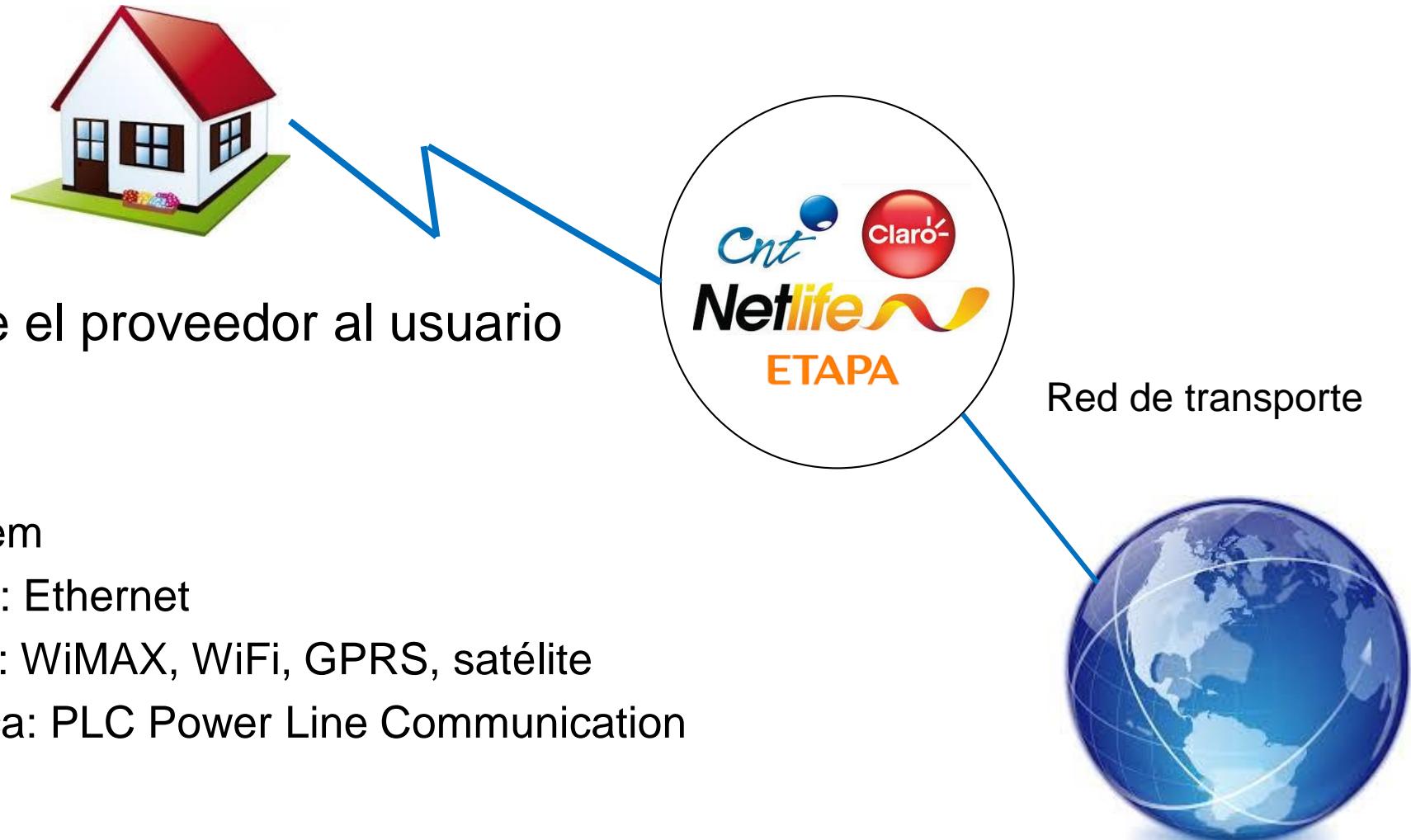
# Redes LAN

- Dan servicios al usuario final



# Local loop, última milla, red de acceso

- Enlace desde el proveedor al usuario
- Tecnologías:
  - ADSL
  - Cable módem
  - Fibra óptica: Ethernet
  - Inalámbrica: WiMAX, WiFi, GPRS, satélite
  - Red eléctrica: PLC Power Line Communication



# Clasificación de redes por escala

- PAN Personal Area Network.
- LAN Local Area Network
- WAN Wide Area Network: Internet

# Redes PAN (Personal Area Network)

Alcanza hasta 10 m



# Red LAN

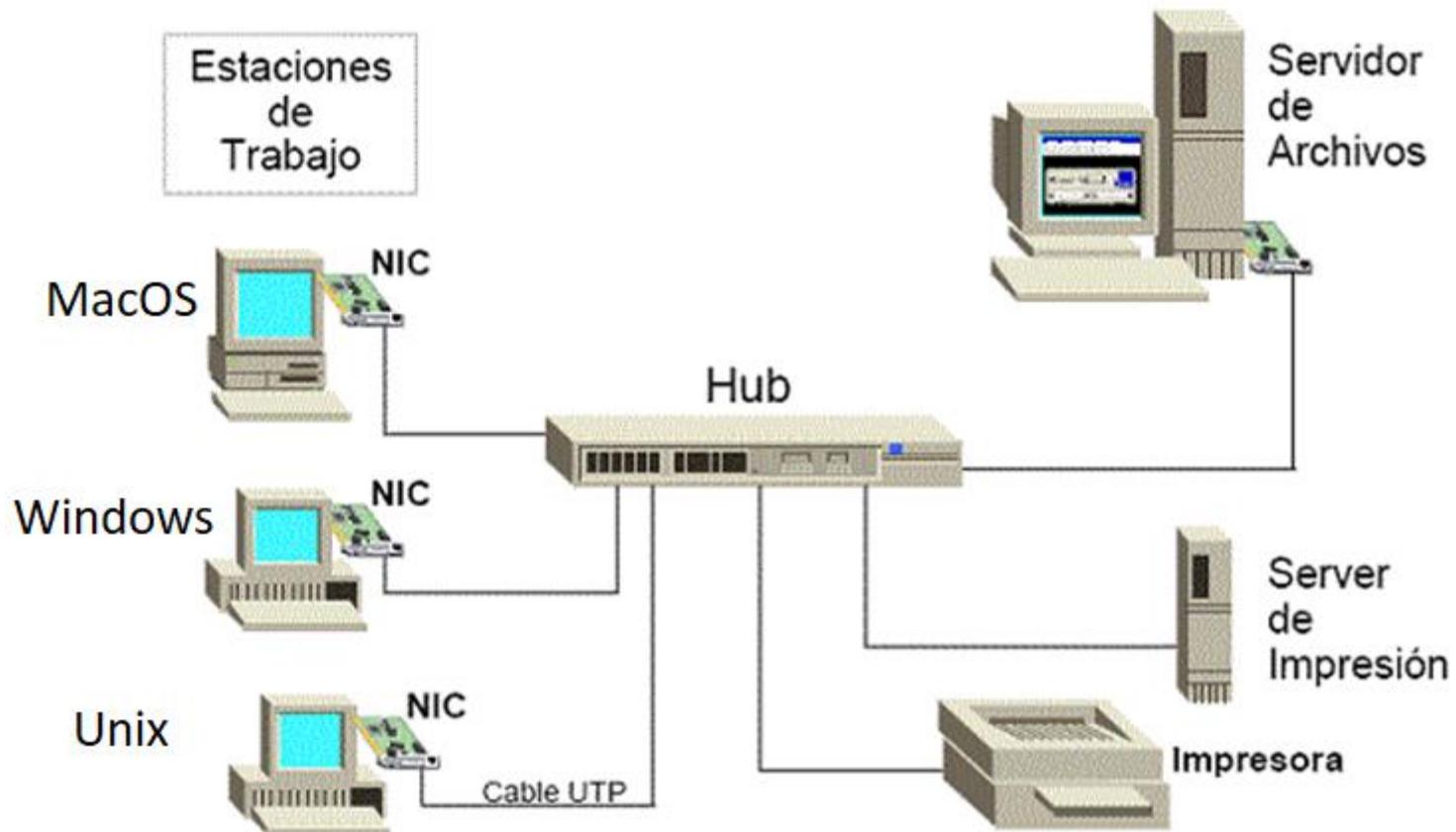
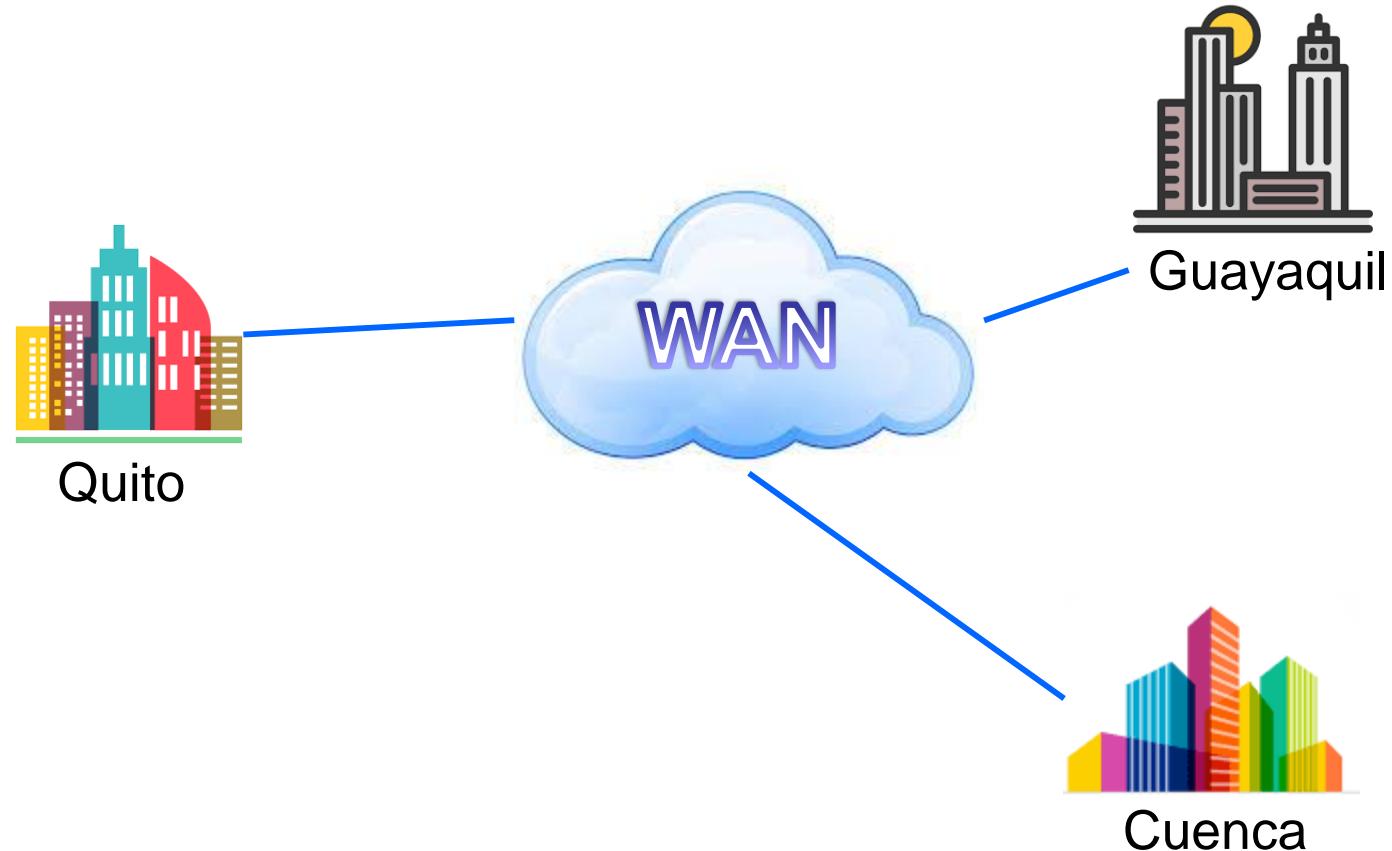


Ilustración 1- 1: Elementos de una red LAN

# Wide Area Network

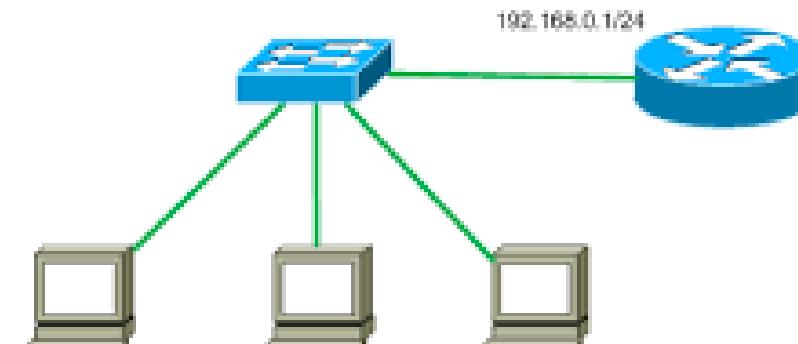






## 1.3.1 Redes LAN

- De propiedad privada
- Un oficina, hogar, edificio, campus (2 km.)
- Diferencias de las LANs de otros tipos de redes:
  - Tamaño
  - Tecnología de transmisión
  - Topología



# Topología bus

- Las máquinas pueden transmitir en **cualquier** momento
- Riesgo de **colisiones**
- Estándar IEEE 802.3
- Nombre comercial: Ethernet (1972)
- Desarrollado por Xerox, DEC, Intel

### 1.3.3 Redes de área amplia WAN

- Interconecta redes LAN que se encuentran distantes entre sí
- Gran área geográfica
- Los clientes poseen los hosts
- Las empresas de telecomunicaciones poseen la subred
- Subred = líneas de transmisión y elementos de conmutación
- Red = hosts + subred

## 1.3.4 Redes inalámbricas.

- Categorías de redes inalámbricas:

- PANs Interconexión sistemas (partes de una computadora)
    - Bluetooth
  - WLANs
    - Access Point
    - Ad hoc
  - WANs inalámbricas

# Utilidad de WLANs

- Lugares en donde es difícil instalar Ethernet:
  - Hogares
  - Pequeñas oficinas
  - Edificios antiguos y patrimoniales
  - Hoteles
  - Cafeterías
  - Bibliotecas
  - Salas de conferencias
  - Centros comerciales
- Estándar WLAN: IEEE 802.11

# WANs Inalámbricas

- Red **telefónica celular**
- Distancias mucho más grandes que las WLAN

Tecnología	Mbps	distancia
WLAN	54-10.000	Menos de 100 m
Red celular	Hasta 300	Kms

# Generaciones de la telefonía celular

- 1G Primera generación
  - Analógica.
  - Fin de los años 60s
  - voz.
- 2G
  - Digital
  - Voz y datos
  - 1989
- 3G
  - Videoconferencia
  - 144 kbps a 120 km/h.
  - 384 kbps a pie
- 4G
  - 2010
  - 100 Mbps a 200 km/h
- 5G
  - 2019

## 1.3.5 Redes domésticas. IoT Internet of Things



## 1.3.6 Interredes

- Existen redes con hardware y software diferente
- Los *gateways* permiten interconectar redes distintas
- Un conjunto de redes diferentes interconectadas se llama **interred**



## 1.4 Software de redes

- Diseño de las primeras redes:
  - El hardware era el punto principal
  - El software era un añadido
- Hoy el software permite que las redes sean amigables

## 1.4.1 Jerarquía de protocolos

- Una red de computadoras se compone de hardware y software
- Para facilitar su diseño, hacemos una abstracción de la red
- Imaginamos a la red como si fuera una pila de capas
- Una capa contempla el hardware y otras contemplan el software
- Cada capa ofrece **servicios** a las capas superiores
- Para que una capa pueda dar servicio, esta realiza internamente ciertas funciones

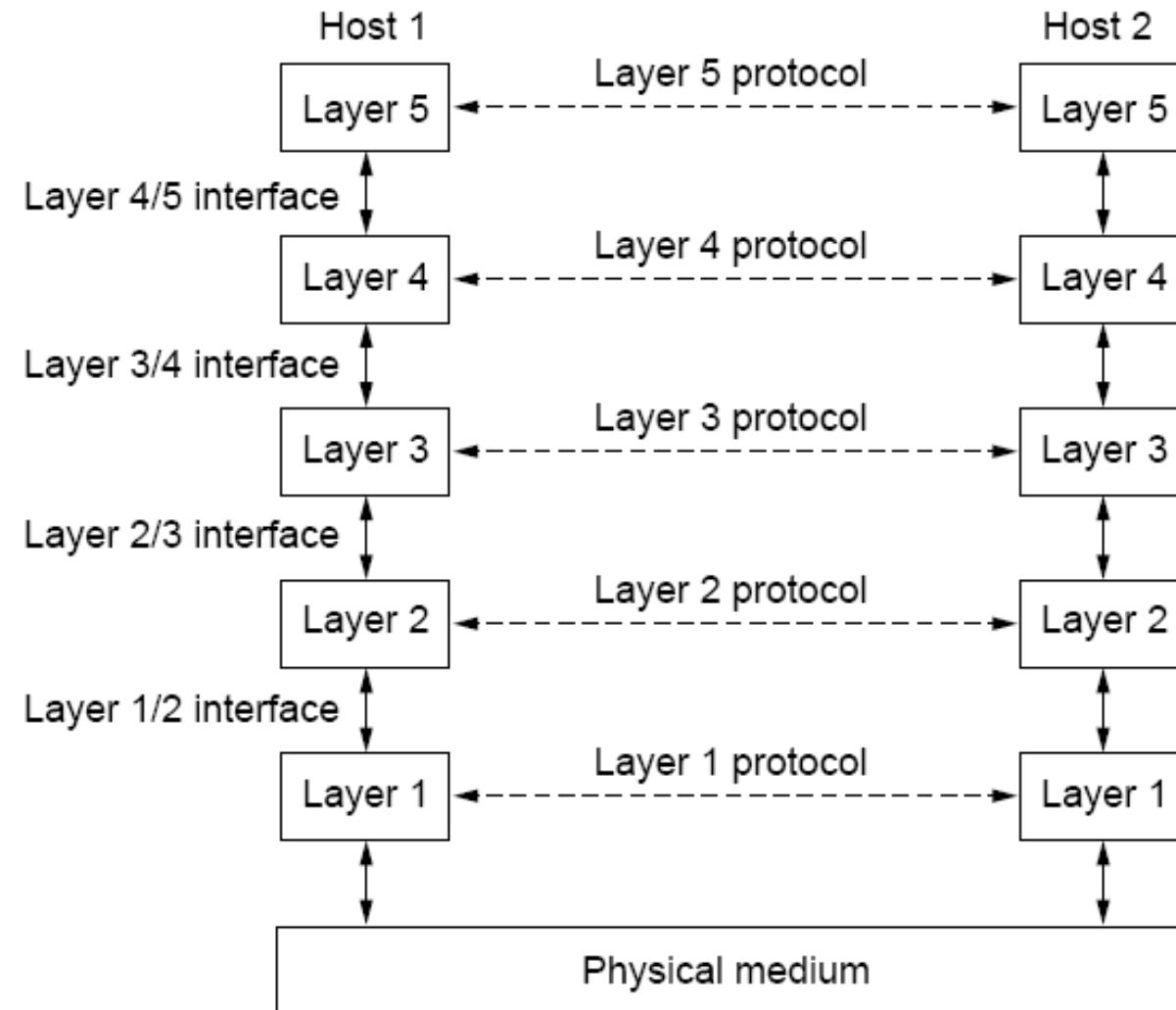


Fig. 1-13. Layers, protocols, and interfaces.

- La capa  $n$  de una máquina intercambia datos con su igual de otra máquina
- El acuerdo de conversación se llaman protocolo de comunicación
- En cada capa se utiliza un protocolo de comunicación
- Por eso se llama jerarquía de protocolos
- Los datos no se envían directo desde la capa  $n$  de una máquina a su igual de otra máquina
- En el emisor, cada capa pasa los datos a su capa inferior
- En el receptor, cada capa pasa los datos a su capa superior

## ■ ¿Qué es una arquitectura de red?

- conjunto de capas
  - servicios, y
  - protocolos
- ## ■ La arquitectura permite que un fabricante desarrolle SW o construya HW para cada capa

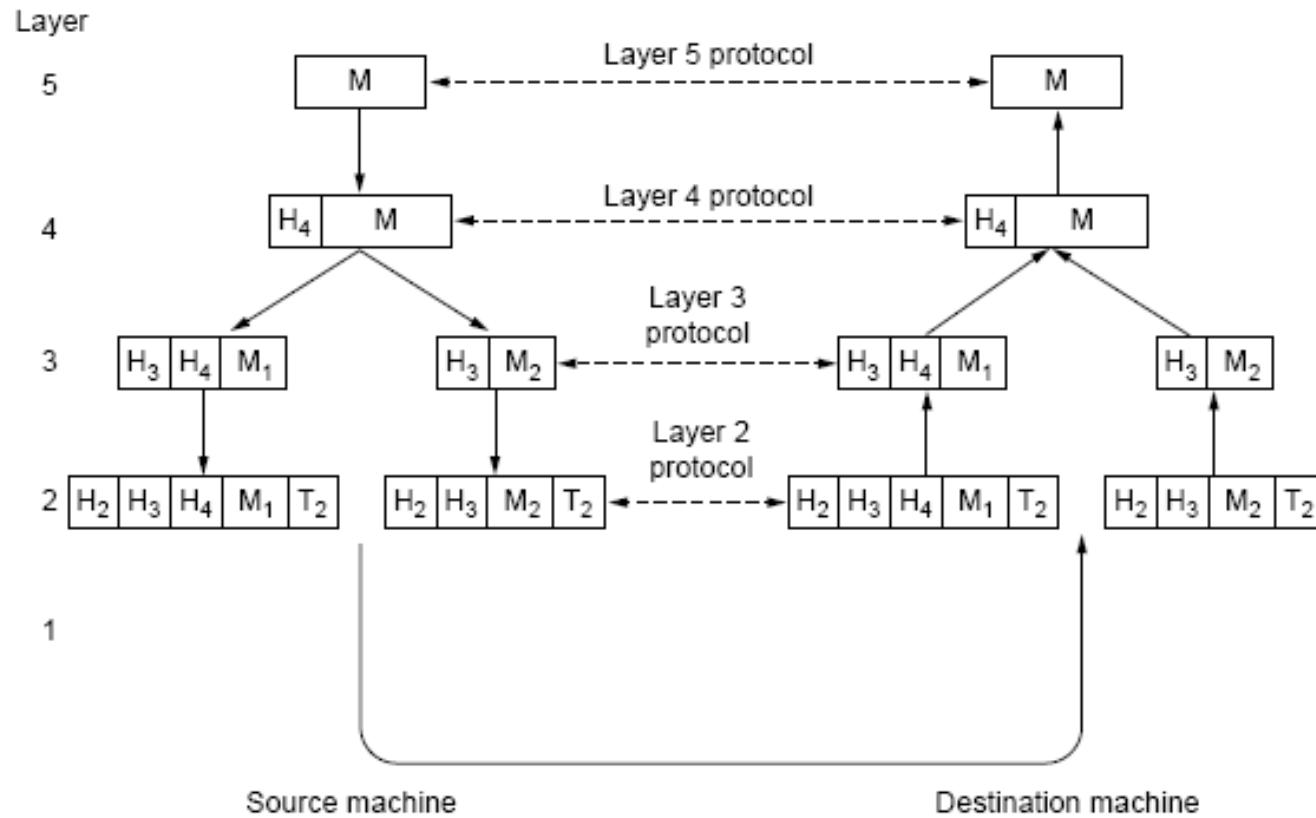


Fig. 1-15. Example information flow supporting virtual communication in layer 5.

- **Encabezado:** Información de control. Identifica: número, tamaño, etc.
- Hay límites de tamaño del mensaje en una capa
- Una capa en el emisor **desintegra** el mensaje en paquetes pequeños
- En la capa correspondiente del receptor los paquetes se integran
- El encabezado colocado en la capa *n* no es pasado a la capa *n+1*

## 1.4.2 Aspectos de diseño de las capas

1. Esquema de direccionamiento
2. Reglas de transferencia de datos
3. Manejo de prioridades
4. Control de errores
5. Numeración de mensajes
6. Control de flujo
7. Desensamblado/ensamblado
8. Multiplexación/demultiplexación
9. Decisiones de enrutamiento

## ■ Direcciónamiento

Una capa necesita una manera de identificar emisores y receptores

Una red tiene muchas computadoras. Éstas tienen muchos procesos

Se requiere una forma de que un proceso en una máquina especifique con qué proceso quiere conversar

# Aspectos de diseño de las capas

- Reglas de transferencia de datos
  - Comunicación simplex
  - Comunicación semidúplex
  - Comunicación dúplex
- Prioridades
- Control de errores
  - Los medios físicos de comunicación no son perfectos. Hay ruido
  - Hay códigos de detección y corrección de errores
- Numeración de mensajes
  - Se controlan pérdidas, duplicación, secuencia
  - Permite unir los pedazos en forma apropiada
- Control de flujo
  - Evita que un emisor rápido **sature** a un emisor lento

# Más aspectos de diseño de las capas

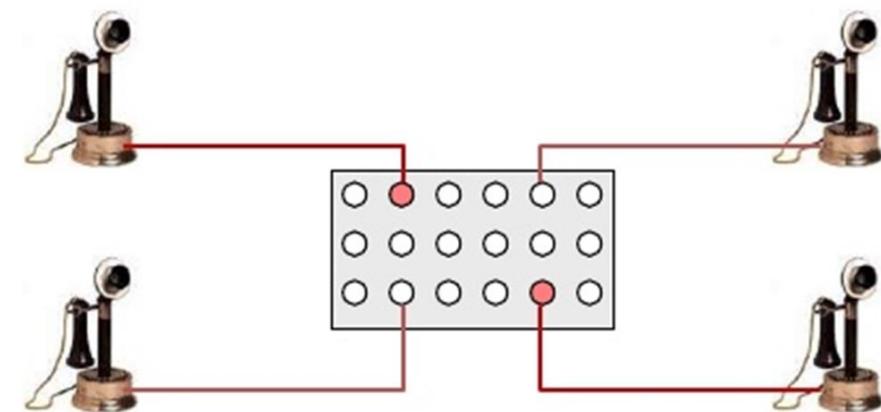
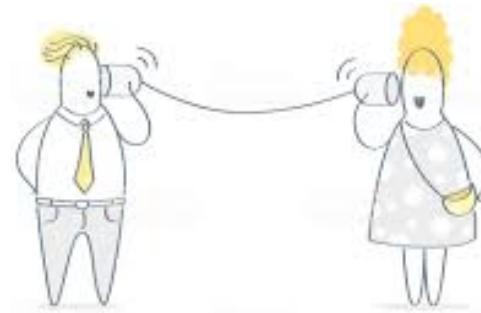
- Desensamblado/re-ensamblado
  - Solución al problema de mensajes muy largos o mensajes muy cortos
- Multiplexión/demultiplexión
  - Uso de una misma conexión para múltiples conversaciones
  - Uso de múltiples conexiones para una misma conversación
- Decisiones de enrutamiento

## 1.4.3 Tipos de servicios ofrecidos por las capas

- Servicios **orientados** a conexión
- Servicios **sin** conexión

# Servicios orientados a la conexión

- **Inspirado** en el sistema telefónico
- El usuario establece la conexión lógica, lo usa, y lo cierra
- La conexión se comporta como un **tubo**
  - Los paquetes llegan en el mismo orden en que fueron enviados
- Usado para el envío de archivos



Comutación de circuitos

# Servicios sin conexión

- Inspirado en el sistema postal
- Cada mensaje lleva la dirección del destino
- Cada mensaje se encamina **independientemente** de los demás a través de la red
- Los mensajes podrían no llegar en el mismo orden en que se enviaron



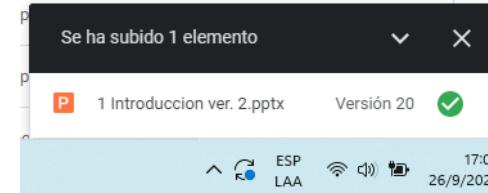
# Servicios sin conexión

- Vienen en dos “sabores”:

- Servicio de datagramas **no confiable**

- Servicio de datagramas **confirmado**

- Igual que una carta certificada: Se envía un mensaje y se espera la confirmación de recepción de ese mensaje



- Los efectos negativos de la confiabilidad son: **sobrecargas y retardos**
- Aplicaciones en tiempo real no toleran retardos inherentes al servicio confiable

## 1.4.4 Primitivas de servicio

- Una capa da un servicio a su capa superior (control errores, control de flujo, orientación o no a la conexión...)
- Una capa solicita un servicio a su capa inferior a través de un conjunto de operaciones
- Estas operaciones se llaman ***primitivas de servicio***
- Una capa da servicio a su capa superior a través de las primitivas de servicio
- Las primitivas son ***llamadas al sistema operativo***
- La **pila de protocolos** está en el **sistema operativo**

- Capas adyacentes de un mismo host se comunican a través de primitivas de servicio
- Capas iguales en sistemas diferentes se comunican a través de un protocolo de comunicación
- **Protocolo de comunicación:** grupo de reglas que definen el formato y significado de los paquetes que se intercambian entre capas iguales de hosts diferentes

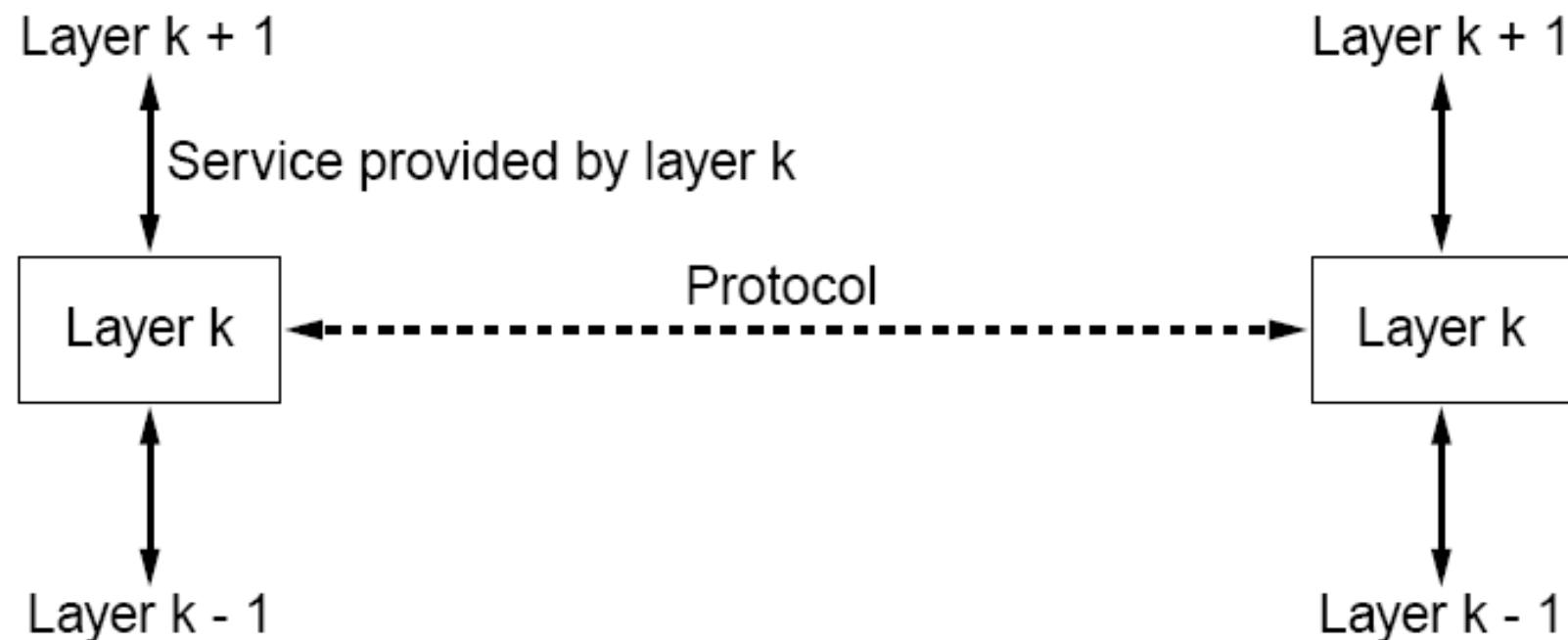
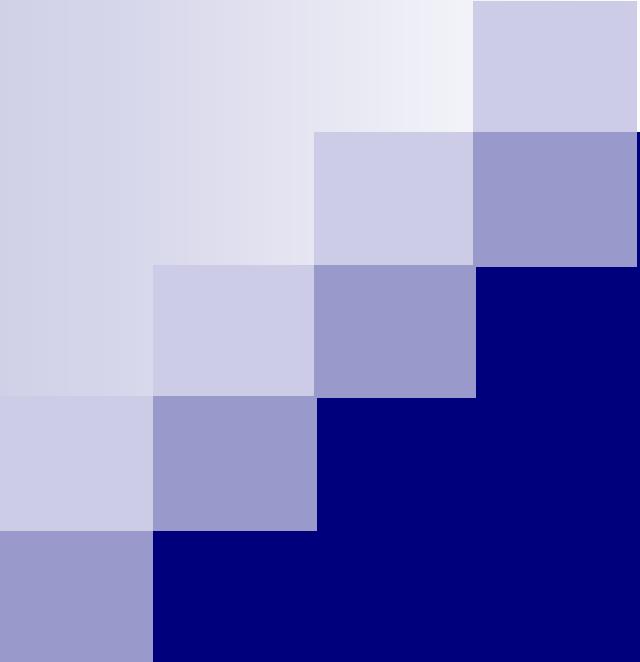


Fig. 1-19. The relationship between a service and a protocol.



## 1.5 Modelos de referencia



digital



CPUs, monitores, teclados, impresoras, cables, conectores debían ser del mismo fabricante. IBM y DEC no eran compatibles entre sí

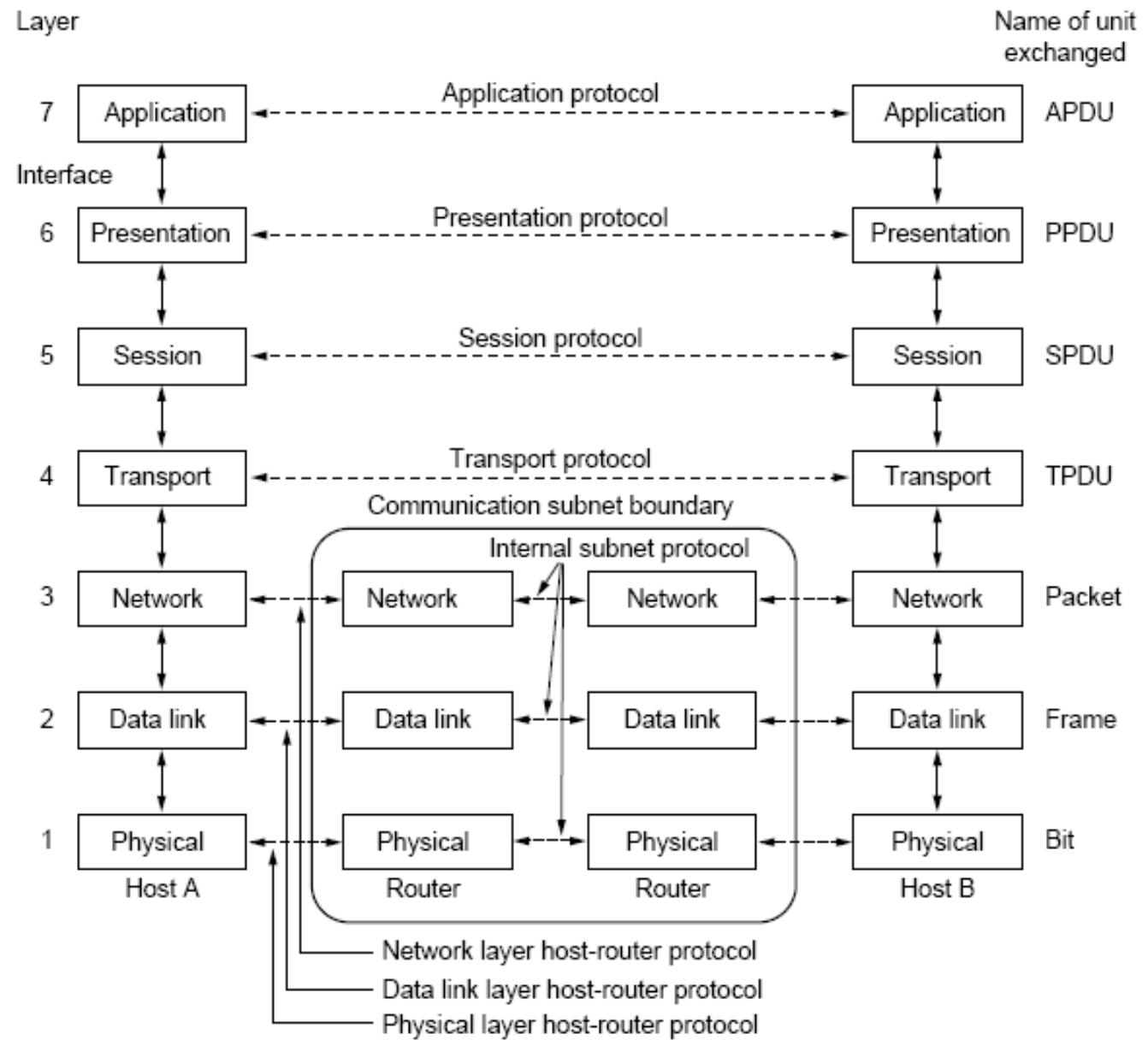
- Surgió la necesidad de que los fabricantes entiendan a las redes de una misma manera
- También era urgente de que los equipos de comunicación sean compatibles entre sí

## 1.5.1 Modelo de referencia OSI

- Open Systems Interconnection (1984)
- OSI es un marco de referencia para el estudio, diseño y construcción de arquitectura de redes
- Sistemas abiertos a la comunicación con otros sistemas
- Primer paso hacia la estandarización internacional
- Propuesto por ISO International Organization for Standardization
- El modelo tiene 7 capas



Ginebra – Suiza (1947)  
Estandariza actividades industriales, comerciales y administrativas



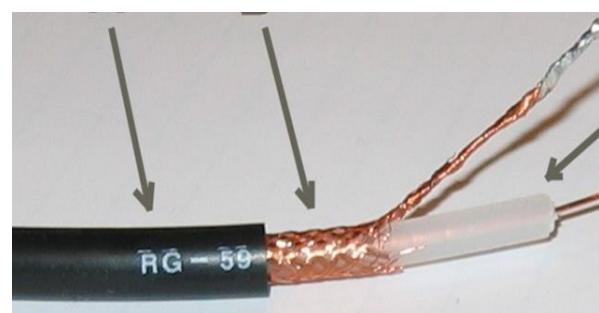
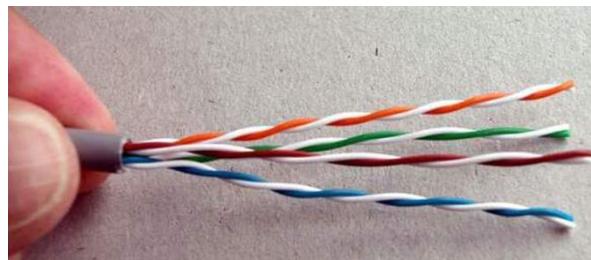
# Cuatro principios para llegar a OSI

1. Se crea una capa para que realice una función
2. Cada capa realiza una sola función
3. La función permite definir luego protocolos
4. Los límites de las capas se eligen tal que minimicen el flujo de información entre capas



# Capa física

- Bits 1 y 0 se deben convertir en señales electromagnéticas para ser transmitidas por un canal de comunicación



# Capa física

- Los medios físicos cumplen con aspectos funcionales, eléctricos, mecánicos

- Funcionales

- Transmisión en uno o en dos sentidos
- Número de pines de un conector
- Función de cada pin
- Secuencia de cada función

- Eléctricos

- Frecuencia de la señal
- Niveles de voltaje
- Tiempo de un bit

- Mecánicos

- Calibres
- Dimensiones
- Resistencia mecánica

# Capa de enlace de datos

- Convierte un medio de transmisión rústico en una línea de comunicación **entre vecinos inmediatos**
- Esto se logra fragmentando los datos en tramas para:
  - Controlar errores
  - controlar el flujo

- En redes de difusión (usan medios compartidos), esta capa controla el acceso al canal de cada computadora
- Este control se realiza a través de la Subcapa MAC *Media Access Control*
- Cada computadora en una red de difusión necesita una interfaz de red
- Cada tarjeta de red debe tener una dirección **física** única
- Esta dirección consta de 48 bits
- Esta dirección es dada por su fabricante y viene impresa en la tarjeta

# Capa de red

- Da un esquema de direccionamiento **lógico**: IP address
- Determina las rutas estáticas y dinámicas
- Elije la mejor ruta con base en diferentes criterios:
  - Saltos
  - Retardos
  - Ancho de banda
  - Tarifas

- En las redes de difusión, el enrutamiento es simple
  - No existen tareas de enrutamiento
  - Los paquetes se difunden a todos los sistemas
  - La capa de red es muy delgada, o no existe

# Capa de transporte

- Controla la comunicación **extremo a extremo**
- Asegura que la información llegue correctamente
- Para ello usa encabezados en mensajes y mensajes de control

# Capa de sesión

- Controla el diálogo: a quién le toca transmitir
- Sincronización: Adición de puntos de referencia para transmisiones largas

# Capa de Presentación

- Existen diferentes tipos de datos: números enteros y de punto flotante, y caracteres alfanuméricos
- Cada tipo de datos tienen un esquema de representación interna binaria
- Un mismo tipo de datos podría tener diferente codificación según cada fabricante
- La Capa de Presentación permite que computadoras con diferentes representaciones internas de datos se puedan comunicar

# Capa de aplicación

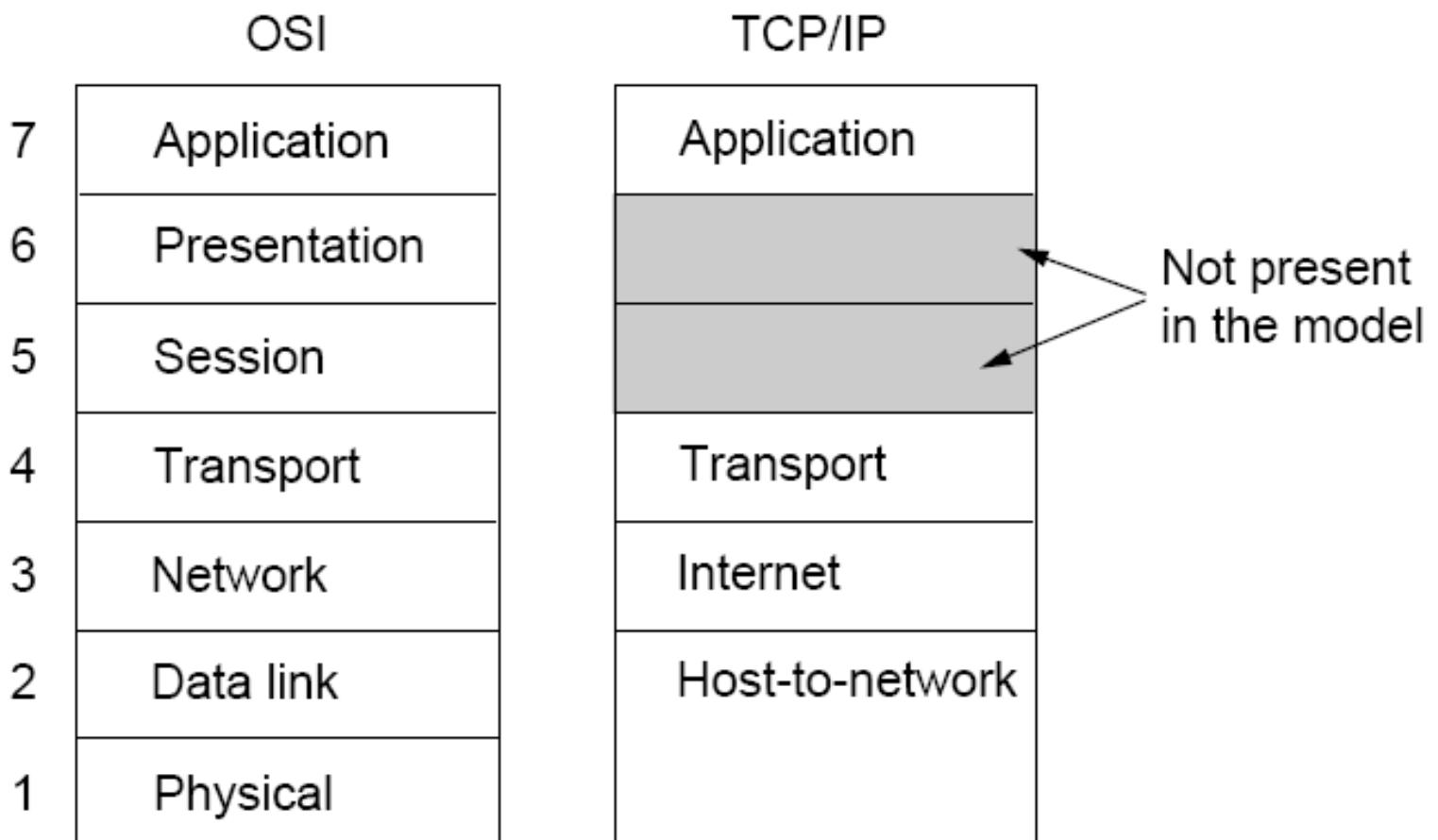
- Pone a disposición diferentes aplicaciones de red que el usuario requiere:
  - Correo electrónico
  - Transferencia de archivos
  - Acceso remoto
  - Servicio de chat
  - Videoconferencia

## 1.5.2 El modelo de referencia TCP/IP

- Año 1974
- Usado en ARPANET (Advanced Research Projects Agency Network) y luego en Internet
- ARPANET de DARPA (Defense Advanced Research Projects Agency)
- Conectó a universidades y oficinas de gobierno utilizando PSTN
- Luego se usaron enlaces de radio y satelitales

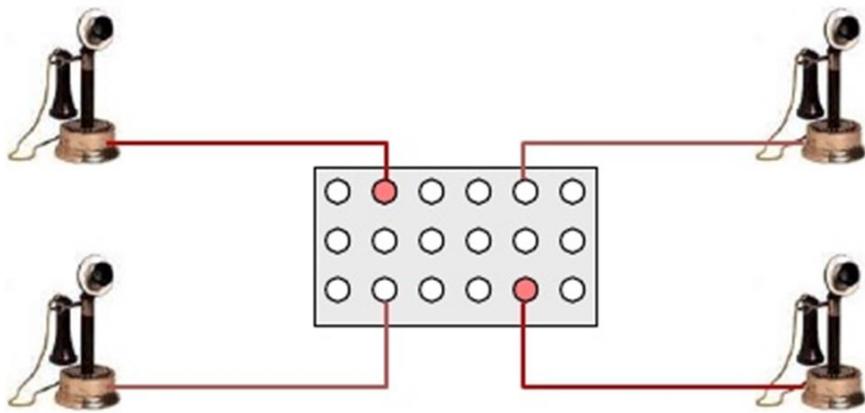
# Requerimientos exigidos a TCP/IP

- Sólido frente a uso de diversos medios físicos de transmisión
- Robusto y flexible frente a pérdidas de hardware de la subred:  
Las conversaciones se mantienen



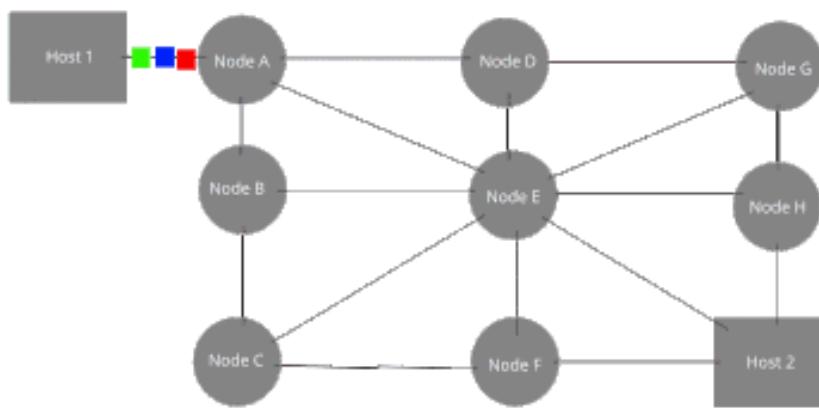
# La capa de internet

- Provee una red de conmutación de paquetes **sin conexión, no fiable**
- Cada paquete viaja por la red en forma independiente. Tal vez lleguen en desorden o incluso no lleguen
- El protocolo de comunicación usado es IP
- Se encarga del enrutamiento
  - Un enrutamiento adecuado evita la congestión
- Esta capa es similar a la capa de red de OSI



## Comutación de circuitos

The original message is Green, Blue, Red.



## Comutación de paquetes sin conexión

## Comutación de paquetes orientado a la conexión

# La capa de transporte

- Igual que la capa de transporte de OSI
- Permite la comunicación extremo a extremo entre hosts origen y destino
- Extremo: host + número de puerto proceso: socket
- Socket origen y socket destino
- Dos protocolos:
  - TCP
  - UDP

# TCP Transmission Control Protocol

- Lenta pero segura (confiable)
- Orientado a la conexión
- Entrega sin errores
- Divide el flujo de bytes que proviene de cada aplicación de red en mensajes discretos
- Controla el flujo y errores por cada conexión

# UDP User Datagram Protocol

- No hay establecimiento de conexión
- No confiable: errores en los datos
- No hay secuenciación ni control de flujo
- Para consultas únicas de solicitud-respuesta
- Usado para entrega **rápida** más que **precisa**: voz y video

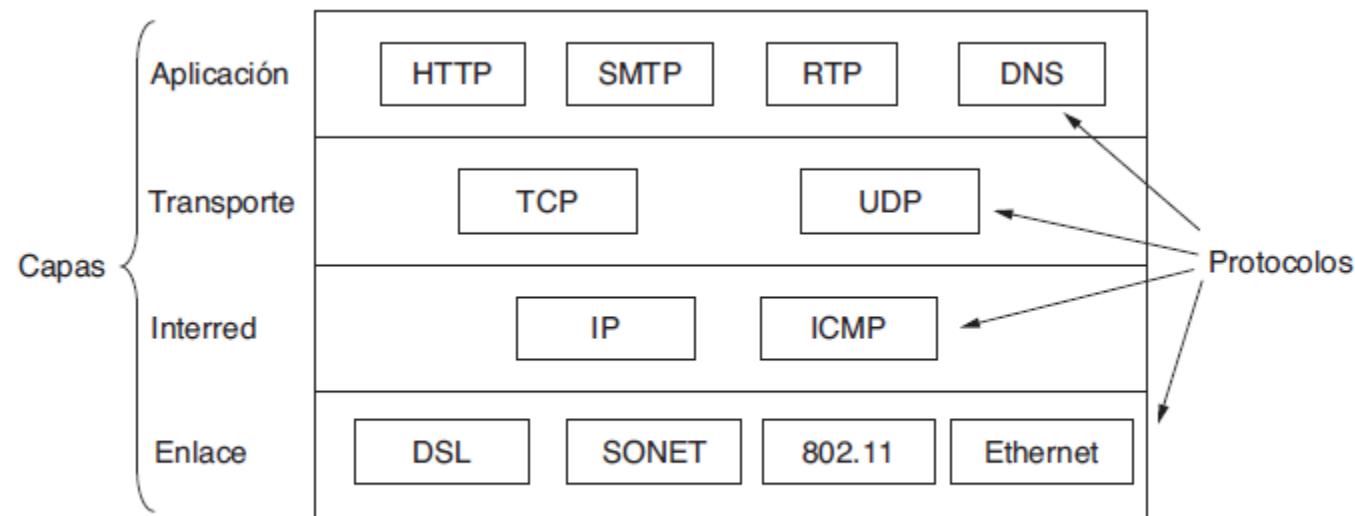


Figura 1-22. El modelo TCP/IP con algunos de los protocolos.

# La capa de aplicación

- TCP/IP no tiene capas de sesión ni presentación
  - De poco uso para la mayoría de las aplicaciones
- Aplicaciones:
  - Telnet
  - FTP
  - SMTP
  - HTTP
  - Servicio de chat
  - Video bajo demanda: TVoD
  - Videoconferencia
  - Radio sobre IP: ROIP
  - Televisión sobre IP: IPTV

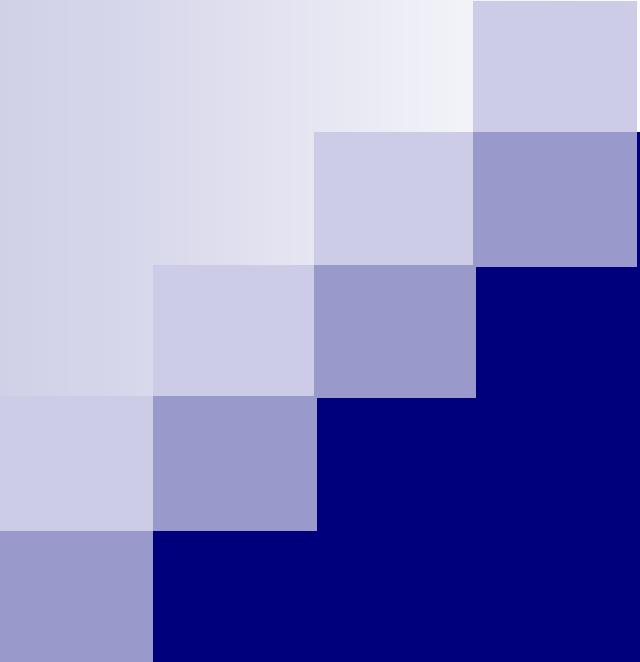
## 1.5.3 Comparación entre OSI y TCP/IP

- Ambos usan concepto de pila de protocolos
- Contribución más grande de OSI: conceptualiza y explicita diferencias entre:
  - Servicios
  - Interfaces
  - Protocolos

- Con OSI el modelo llegó primero y luego los protocolos
- Con TCP/IP los protocolos llegaron primero y después el modelo
- OSI: diseñadores sin experiencia
  - A inicios la capa 2 sólo trataba redes punto a punto. Cuando llegaron las redes LAN tuvieron que crear las subcapa MAC.

- OSI no pensó en la interconexión de redes
- OSI no pensó inicialmente en las redes de difusión
- TCP/IP no es adecuado para describir otras redes que no sean TCP/IP

- Una diferencia evidente es el número de capas
- OSI soporta comunicación orientada y no orientada a conexión en capa 3
- TCP/IP sólo soporta el servicio no orientado a conexión en la capa 3
- OSI soporta sólo la comunicación orientada a la conexión en la capa 4
- TCP/IP soporta ambos en la capa 4



# 1.6 Hardware de redes por capas

- Capa física
  - Cables
  - Conectores
  - Antenas
  - Repetidores
  - Hubs
  - Módems
- Capa de enlace de datos
  - Tarjetas de red
  - Access Point
  - Switches
  - Switches de capa 3
- Capa de red
  - Routers
- Capa de transporte
  - Firewalls
- Capa de aplicación
  - Computadoras

# Capa Física

- **Conectores**
- Permite conectar la computadora al medio físico de transmisión



- Repetidor
- Recibe la señal, la depura del ruido, la amplifica y la retransmite
- Puede ser cableada o inalámbrica



# Hubs

- Permite que un computador que se conecta a otros computadores necesite solo una tarjeta de red
- Un hub interconecta varias computadoras para formar una LAN
- Todas las computadoras reciben la información transmitida
- El hub determina el dominio de colisión
- Dominio de colisión es un segmento físico de una red, compartido por un grupo de computadoras, donde es posible que las tramas colisionen
- Trabaja mejor con pocas computadoras conectadas



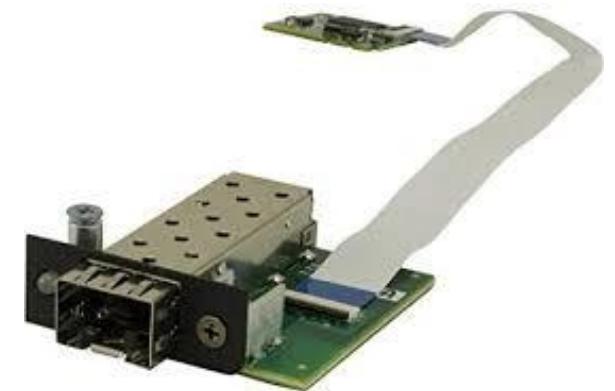
# Módem

- Transforma la señal digital en analógica y viceversa
- La señal analógica puede transmitirse en un medio guiado o inalámbrico
- El medio guiado puede ser línea telefónica, cable coaxial o fibra óptica



# Capa de enlace de datos

- Tarjeta de red
- Dispositivo que permite conectar un computador a una red LAN



## ■ Switch

- Al igual que un hub, un switch permite interconectar un grupo de computadoras para formar una LAN
- A diferencia de un hub, cada puerto del switch es un dominio de colisión diferente
- Maneja las direcciones físicas que tienen las tarjetas de red



## ■ Access Point

- Es un switch inalámbrico para formar una wireless LAN
- Los APs pueden conectarse entre sí comportándose como repetidores para permitir roaming
- Soporta hasta 255 estaciones



- Switch de capa 3
- Switch + router
- Maneja direcciones físicas y direcciones IP
- Permite la conexión de redes con diferente dirección de red
- No tiene puertos serie para conectarse a redes WAN
- No puede conectarse directamente a un proveedor de Internet
- Para la conexión a Internet se necesita de un router



# Capa de red

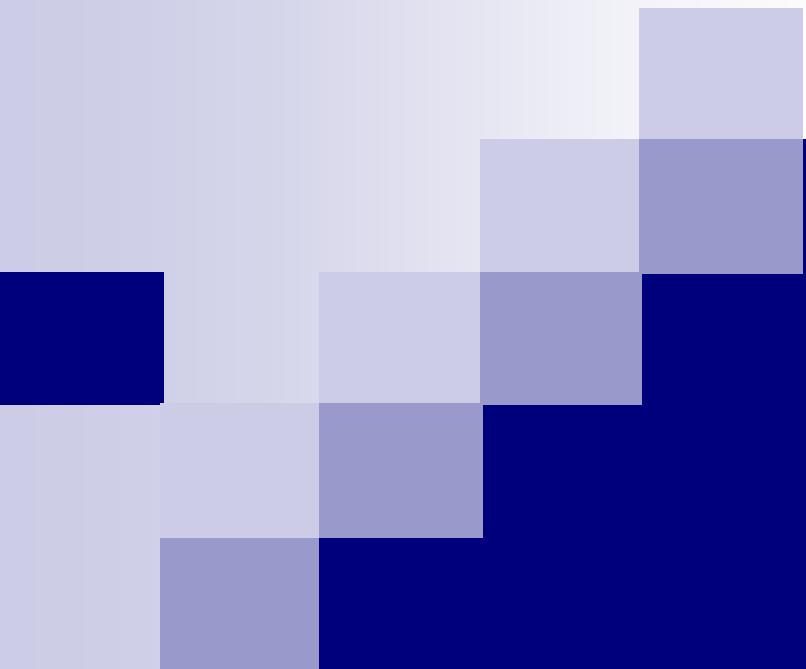
- **Routers**
- Maneja direcciones IP
- Permite la conexión de redes con diferente dirección de red
- Un router determina dominios de broadcast
- El broadcast está marcado para cada dirección de red IP
- Tiene puertos serie para conectarse a redes WAN
- Puede conectarse directamente a un proveedor de Internet



# Capa de transporte

- **Hardware Firewall**
- Dispositivo de control de tráfico
- Generalmente trabajan en capa 4
- Controla el tráfico entrante y saliente de una corporación
- Bloquea el tráfico no autorizado





## 2 La Capa Física

# Contenido

- 2.1 Transmisión de datos
- 2.2 Codificación o señalización de datos
- 2.3 Base teórica de la comunicación de datos
- 2.4 Medios de transmisión guiados
- 2.5 Medios de transmisión no guiados o inalámbricos
- 2.6 Satélites de comunicación



## 2.1 Transmisión de datos

Libro: Comunicaciones y redes de computadores

Autor: William Stallings

Capítulo 3. Transmisión de datos

# Teoría de la Información

- La Teoría de la Información estudia la cantidad de información que contiene un mensaje que se transmite a través de un canal de comunicación
- La Teoría de la Información es una rama de las matemáticas y de las Ciencias de la Computación
- Esta teoría estudia la manera de representar o codificar datos y la capacidad de los medios de transmisión
- Para estudiar la Capa Física se necesita recurrir a la Teoría de la Información

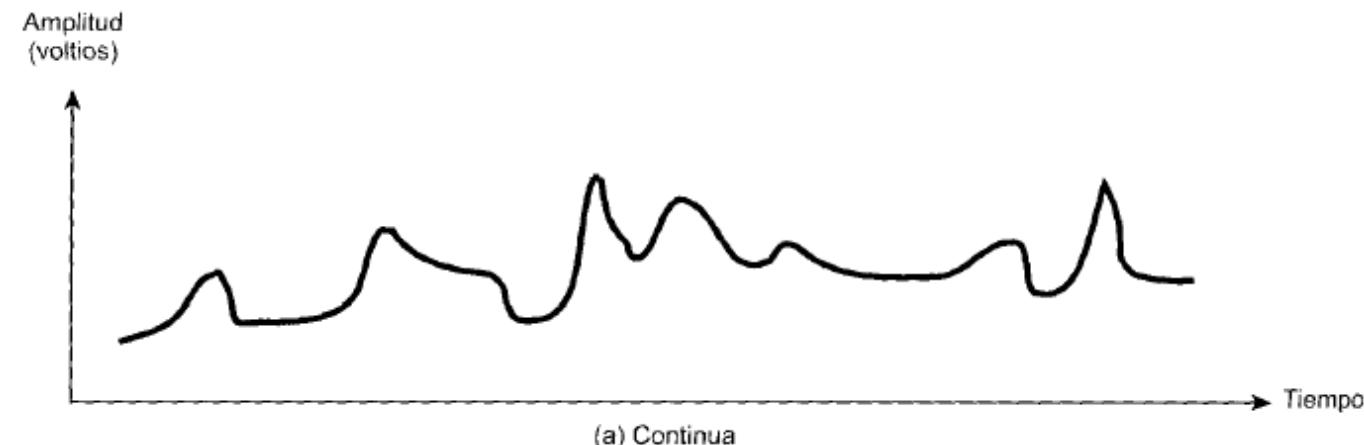
- Existen varios formatos en los que se presenta la información:
- Texto, sonido, imágenes y video
- ¿Cómo se representa la información para transmitir la información por la red?
- Con señales electromagnéticas
- Estas señales pueden ser analógicas o digitales según el medio que se use
- ¿Qué es una señal analógica y una señal digital?
- El éxito en la transmisión depende de dos cosas:
  - la calidad de la señal y
  - las características físicas del medio

## 2.1.1 Conceptos y terminología

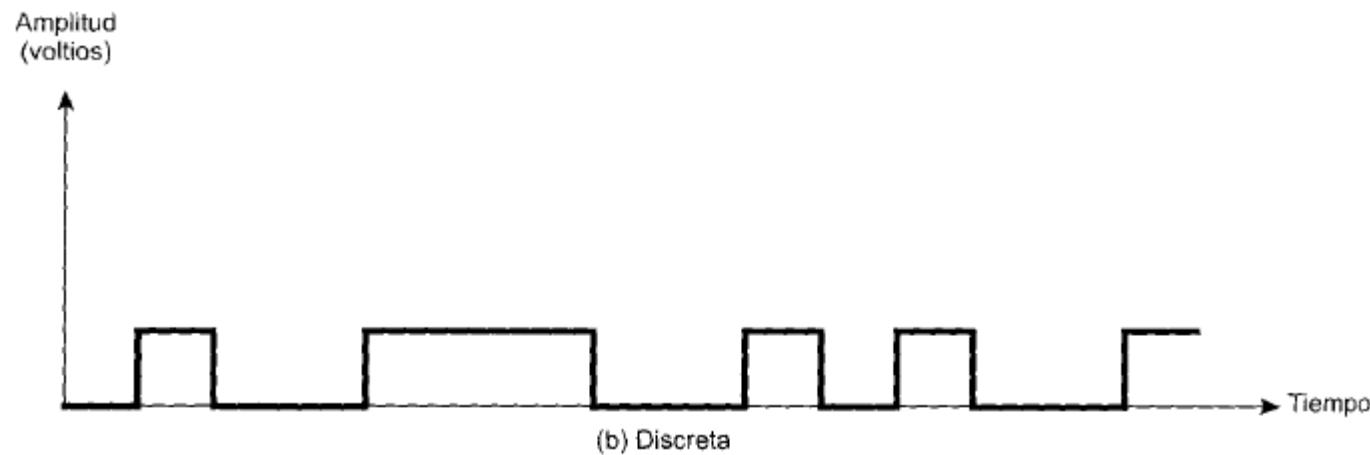
- Hay medios de transmisión guiados y no guiados
- En cualquier caso la transmisión usa ondas electromagnéticas
- La manera de transmitir puede ser:
  - Simplex: transmisión en una dirección
  - Duplex: transmisión simultánea en dos direcciones
  - Half-duplex: transmisión en dos direcciones en forma alternada

# Clasificación de las señales

- Una señal electromagnética en el dominio del tiempo puede ser continua o discreta
- Señal continua: la intensidad varía suavemente

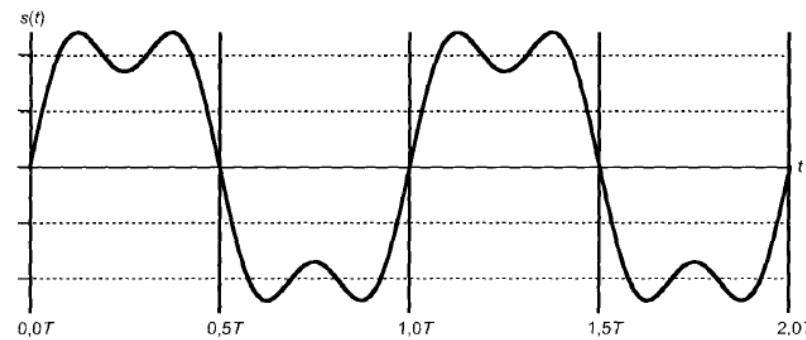


# Señal discreta

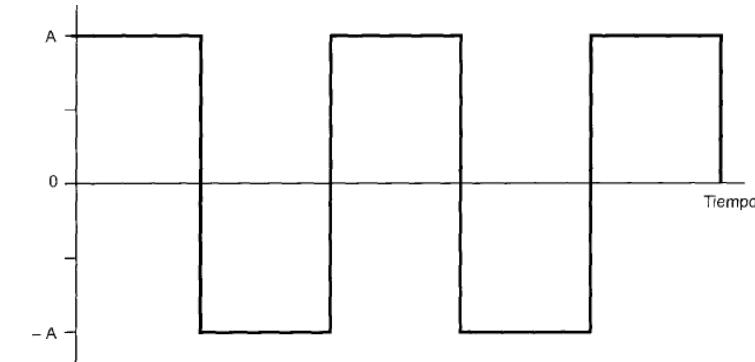


- Toma solo ciertos valores
- La señal se mantiene constante por cierto tiempo, luego cambia a otro valor constante
- Se representa con pulsos de voltaje
- Cada pulso es un elemento de señal

- Por otro lado, una señal puede ser periódica o aperiódica
- Señal periódica: tiene un patrón que se repite en el tiempo con una periodicidad  $T$  o frecuencia  $f$

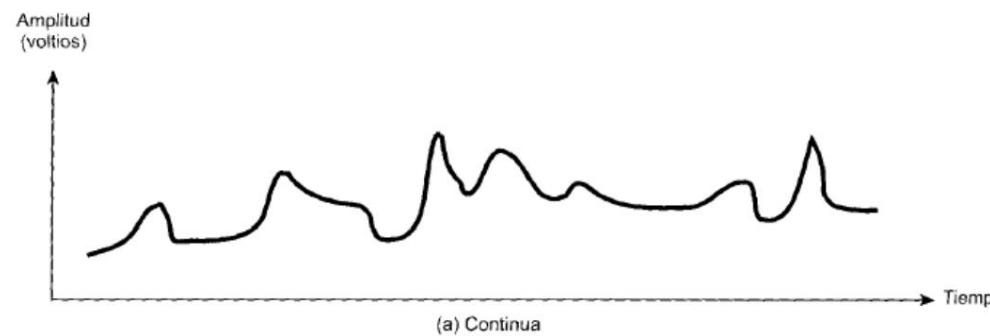


Continua y periódica

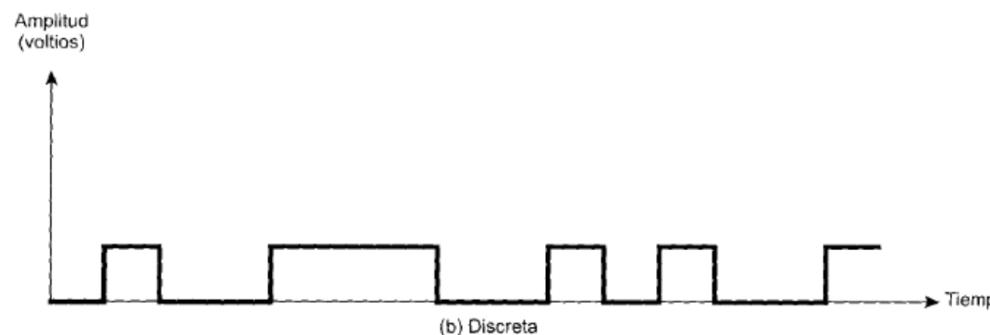


Discreta y periódica

- Señal aperiódica: no tiene un patrón que se repita en el tiempo



Continua y aperiódica



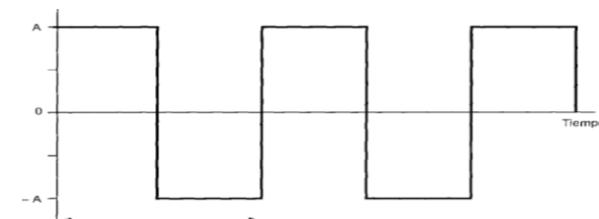
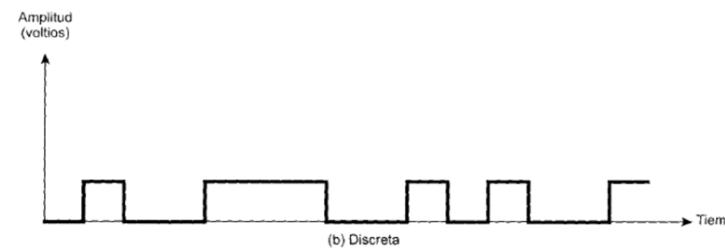
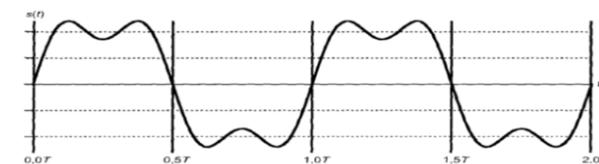
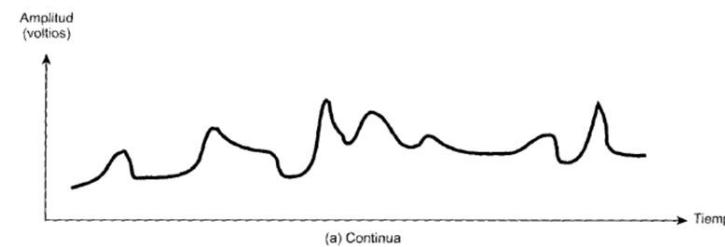
Discreta y aperiódica

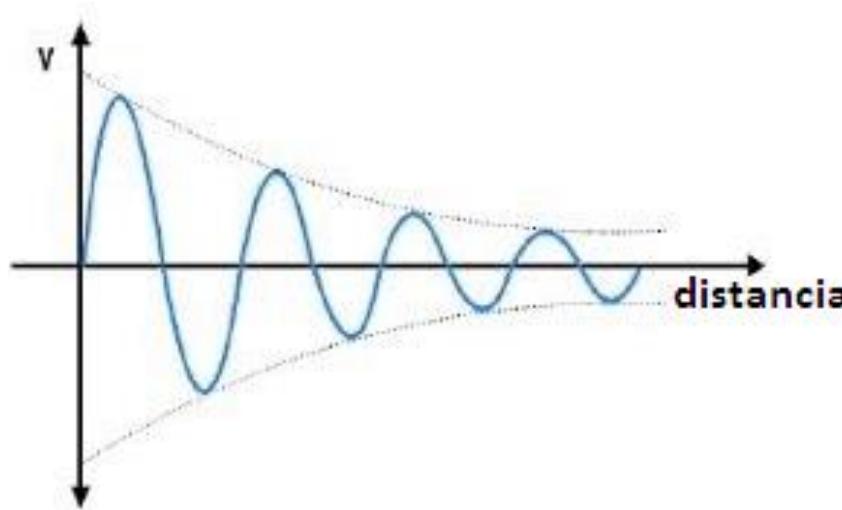
- ¿Cuáles son los parámetros de una función continua, por ejemplo una señal de eléctrica?
  - Amplitud A: valor máximo de la señal expresado en voltios (V), amperios (A)
  - Frecuencia f: número de veces que se repite el patrón en un segundo (Hz)
  - El período T: tiempo de duración del patrón que se repite ( $T = 1/f$ ) (s)
  - Fase  $\phi$ : desplazamiento de la señal con relación a algún punto de referencia ( $^{\circ}$ , rad)

- Una señal puede representarse de tres maneras:

- En función del tiempo
  - En función del espacio
  - Si es periódica, en función de la frecuencia

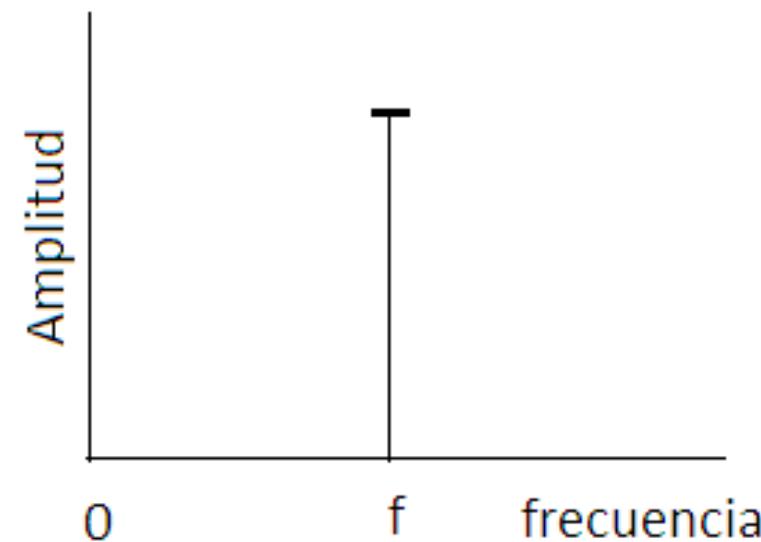
## ■ Representación de la señal en función del tiempo





- Representación de la señal en función de la distancia recorrida
- Longitud de onda  $\lambda$  es la distancia de un ciclo expresada en metros
- La amplitud  $A$  se atenúa
- $\lambda$  permanece constante

- Representación de una señal periódica en función de la frecuencia



## 2.1.2 Transmisión de datos analógicos y digitales

- Los términos *analógico* y *digital* corresponden a *continuo* y *discreto* respectivamente
- Estos dos términos se aplican a tipos de: datos, señales y transmisión
- **Dato:** unidad básica de información que es cuantitativa o cualitativa
- **Señal:** representación de los datos
  - ¿Cómo se representan los datos?
  - Colores, imágenes, sonidos, luces, ondas electromagnéticas
- **Transmisión:** propagación de datos en forma de señales utilizando un medio físico

# Datos

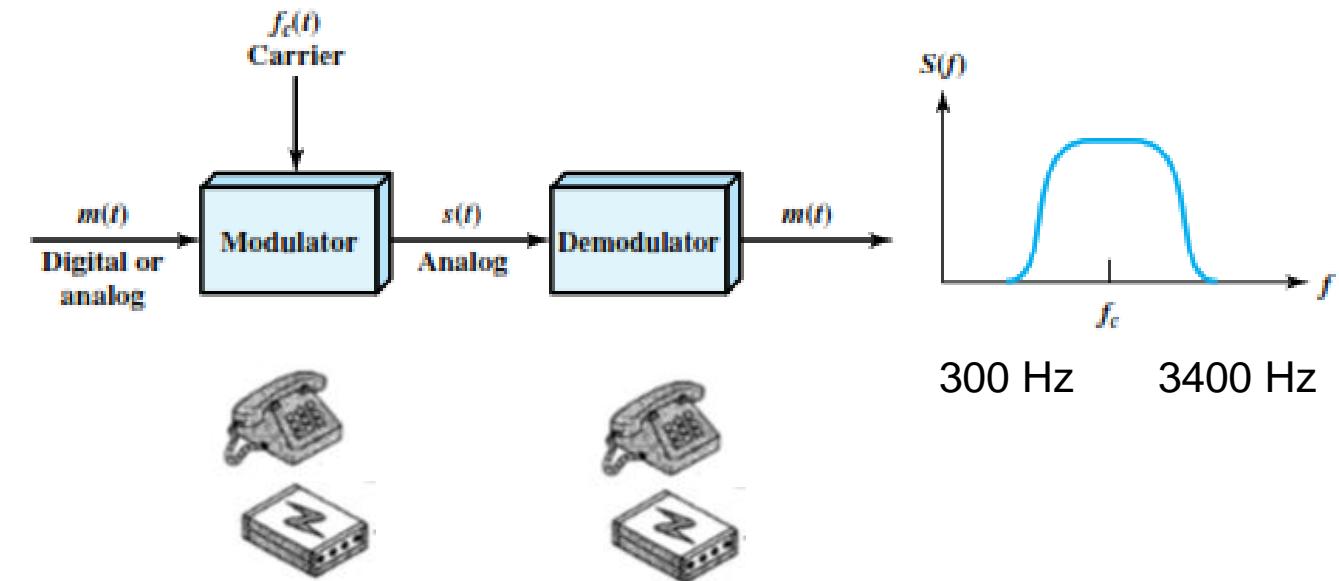
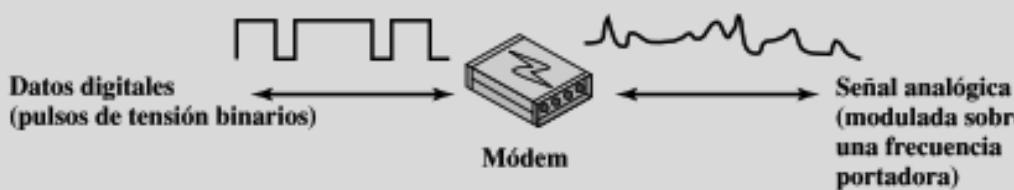
- Dato analógico toma valores continuos en el tiempo. Ejemplos:
- Sonido, video, temperatura, presión atmosférica, velocidad, estatura de una persona, ...
- Dato digital toma valores discretos. Ejemplos:
- Cantidad de personas, páginas de un libro, cosas, ideas, momentos felices  
...

# Señales

- **Señal analógica** se transmite como onda electromagnética en diferentes tipos de conductores o en el espacio abierto
  - La voz ocupa un espectro de 300 Hz a 4000 Hz y se representa con una señal electromagnética que ocupa el mismo espectro. Precisamente por este hecho se llama *señal analógica*
- Las señales analógicas son adecuadas para transmitir datos a grandes distancias
- Los niveles de voltaje de la señal analógica de la voz en una línea telefónica analógica están entre -6V a +6V

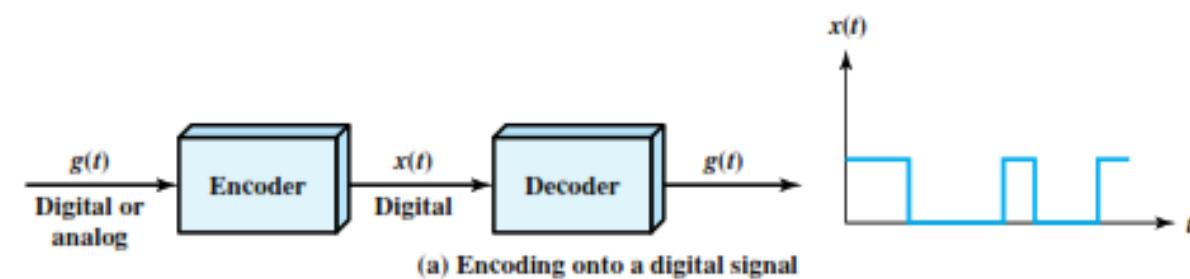
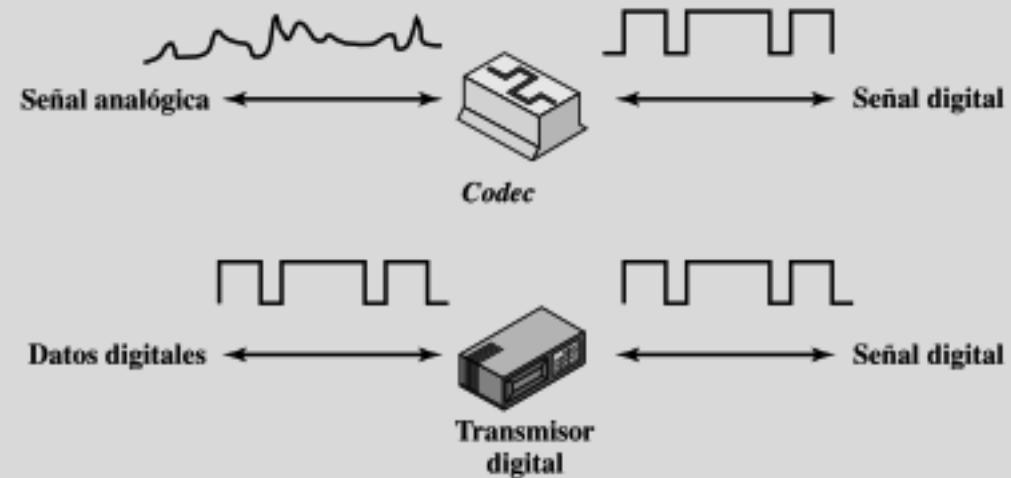
- **Señal digital** es una secuencia de pulsos eléctricos que se transmiten solo a través de medios **guiados**
- Estas señales se transmiten solo a cortas distancias
- Los niveles de voltaje de Fast Ethernet son +1.2 V y -1.2 V

**Señales analógicas:** representan los datos mediante una onda electromagnética que varía continuamente



En ambos casos, el teléfono o el modem modifican (modulan) algún parámetro ( $A$ ,  $f$ ,  $\phi$ ) de la portadora (tono)

**Señales digitales: representan a los datos mediante una secuencia de pulsos de tensión**



Hay varios esquemas de codificación digital

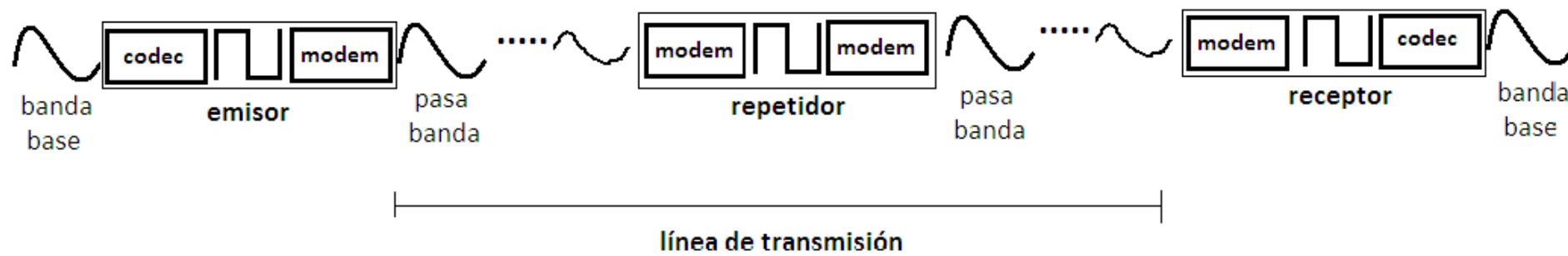
# Transmisión

	Transmisión analógica	Transmisión digital
Señal analógica	Se usan amplificadores para datos analógicos o digitales	Si los datos son digitales se usan repetidores para demodular la señal analógica con el fin de recuperar los datos digitales para luego volverlos a modular
Señal digital	<b>No</b> se usan amplificadores para amplificar señales digitales  No hay transmisión analógica	Se usan repetidores para recuperar la señal digital para volverla a transmitir

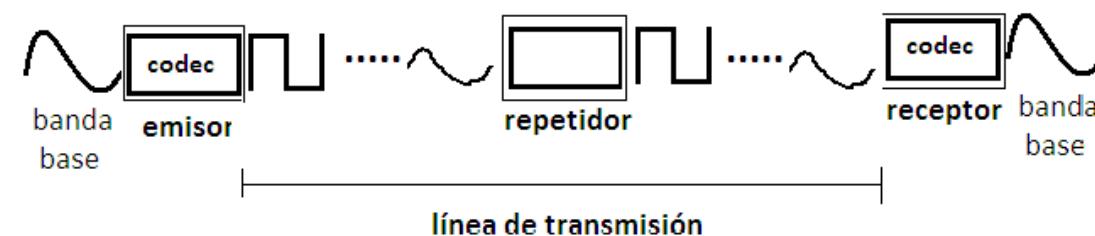
El propósito de la transmisión digital es limpiar el ruido FO y el espacio abierto solo admiten señales analógicas

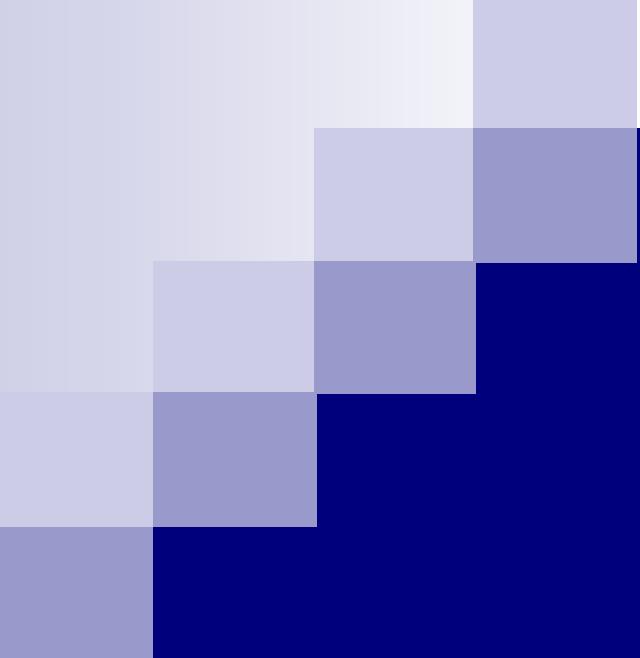
# Transmisión digital

## Señal analógica



## Señal digital





## 2.2 Codificación o señalización de datos

Libro: Comunicaciones y redes de computadores

Autor: William Stallings

Capítulo 3. Transmisión de datos

- Datos digitales, señales digitales
- Datos digitales, señales analógicas
- Datos analógicos, señales digitales
- Datos analógicos, señales analógicas

## 2.2.1 Datos digitales, señales digitales

- Los datos digitales se representan con el sistema binario
- Esquema de señalización o codificación de datos: es la relación entre los bits de datos y los elementos de señal
- En el esquema más sencillo, cada bit de datos se corresponde con un elemento de señal o pulso
- Un 0 binario se representa mediante un nivel de voltaje
- Un 1 binario se representa con otro nivel de voltaje

- Señal unipolar: todos los elementos de señal o pulsos tienen el mismo signo (todos positivos o todos negativos)
- Señal polar: hay pulsos de señal positivos y negativos
- La velocidad de **transmisión** de los datos se expresa en bps
- La velocidad de **modulación** o señalización es la velocidad de los pulsos de voltaje en un segundo (baudio)



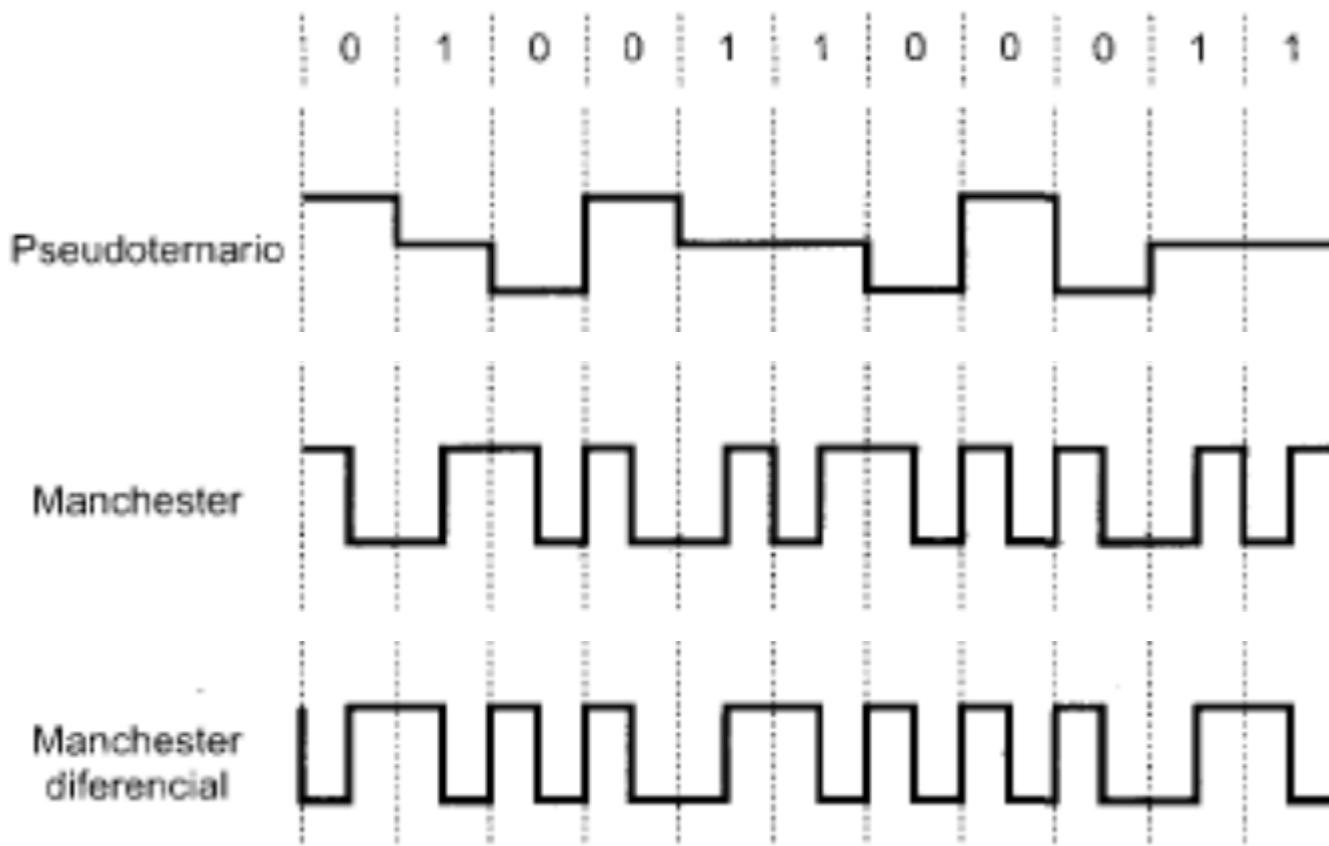
0: Nivel de voltaje alto  
1: Nivel de voltaje bajo



0: Sin transición de voltaje  
1: Transición de voltaje



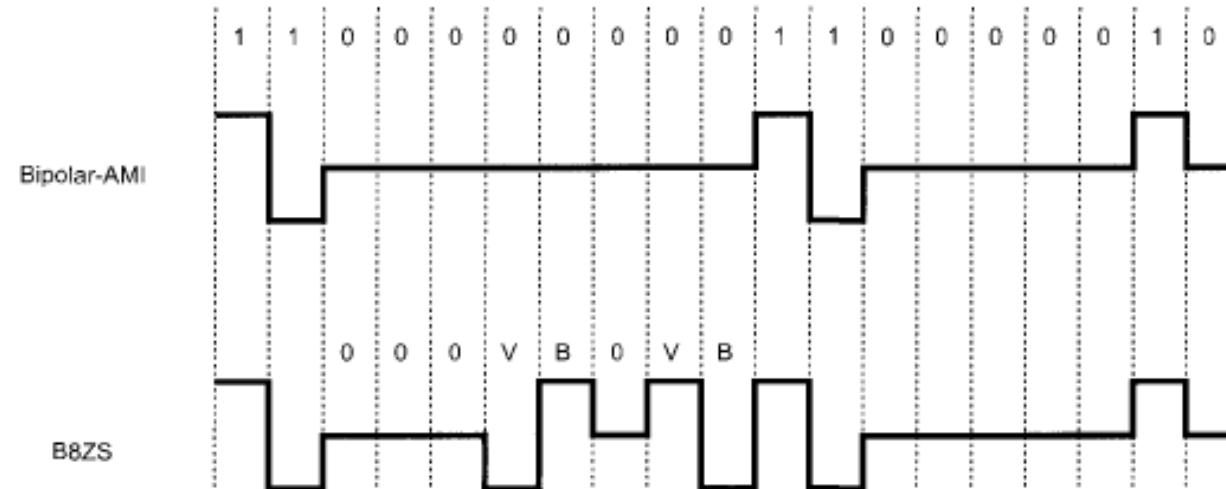
0: Sin señal  
1: Voltaje negativo o positivo alternante



1: Sin señal  
0 : Voltaje positivo  
o negativo alternante

0: Transición alto-bajo  
1: Transición bajo-alto

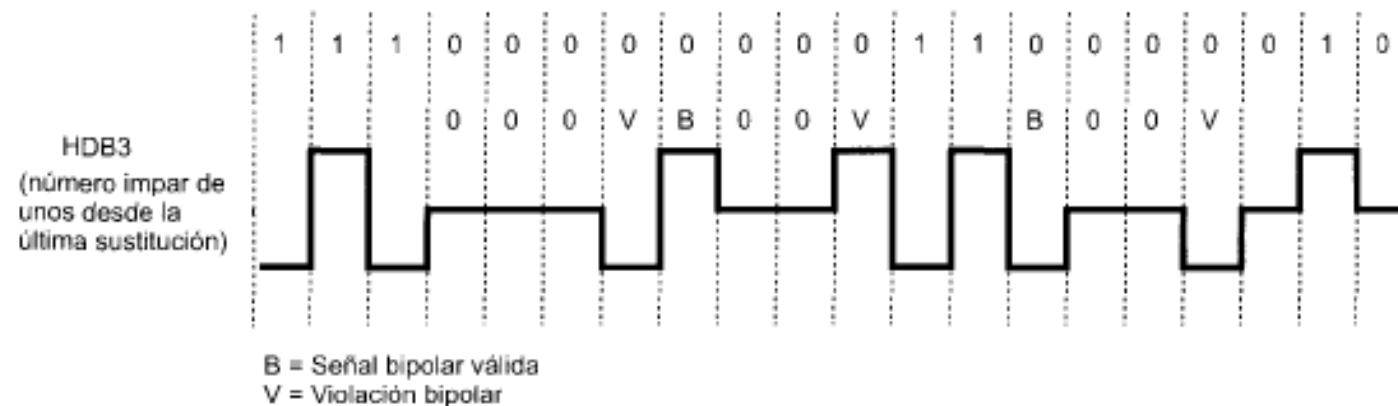
0: Transición al inicio  
1: Sin transición al inicio



Evita el problema de la pérdida de sincronía por la presencia de muchos 0s consecutivos

Si hay ocho 0s consecutivos, estos se codifican así:

- Si el último voltaje anterior al octeto es +: 000+-0--
- Si el último voltaje anterior al octeto es -: 000--0+-



**Tabla 5.4.** Reglas de sustitución en HDB3.

Número de pulsos bipolares (unos) desde la última sustitución		
Polaridad del pulso anterior	Impar	Cero o Par
-	000-	+00+
+	000+	-00-

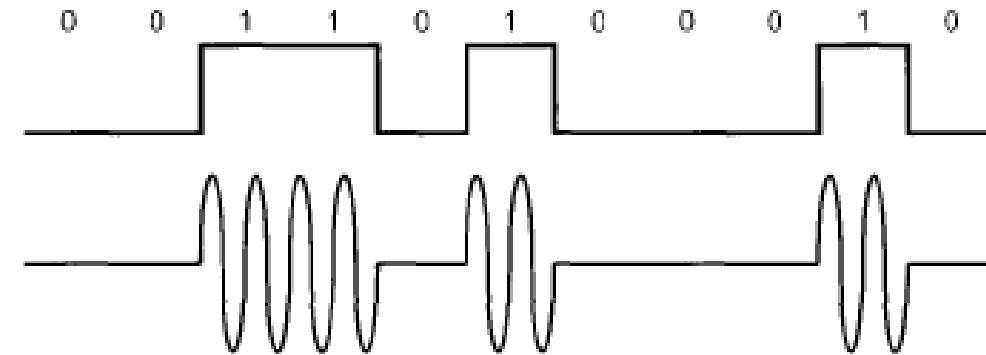
Se reemplazan las cadenas de cuatro 0s por cadenas que tienen uno o dos impulsos

## 2.2.2 Datos digitales, señales analógicas

- La representación, señalización o modulación analógica de datos digitales se lo hace modificando alguno de los parámetros de una onda portadora senoidal
- Los parámetros son amplitud, frecuencia y fase
- Los módems producen señales en el rango de frecuencia de voz (300 a 4000 Hz) para líneas telefónicas, o en frecuencias más altas para microondas

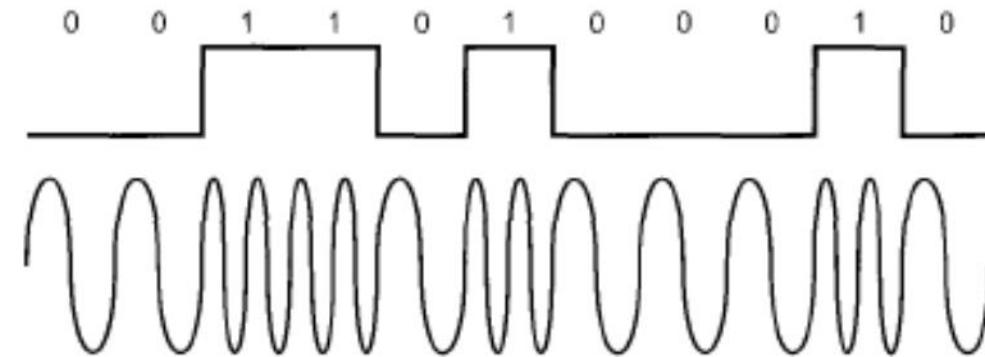
- Hay tres técnicas básicas de representación, señalización, codificación o **modulación** de datos digitales:
  - Cambio de amplitud (ASK Amplitude-Shift Keying)
  - Cambio de frecuencia (FSK Frequency-Shift Keying)
  - Cambio de fase (PSK Phase-Shift Keying)
- La señal resultante ocupa una ancho de banda centrado en la frecuencia de la portadora  $f_c$

# ASK



- Una de las amplitudes es cero o ausencia de portadora
- $s(t) = \begin{cases} A\cos(2\pi f_c t) & 1 \text{ binario} \\ 0 & 0 \text{ binario} \end{cases}$
- En líneas telefónicas se usa hasta 1200 bps
- ASK se utiliza en fibra óptica

# FSK



- $s(t) = \begin{cases} A\cos(2\pi f_1 t) & 1 \text{ binario} \\ A\cos(2\pi f_2 t) & 0 \text{ binario} \end{cases}$
- $f_1$  y  $f_2$  son desplazamiento de la frecuencia portadora o de carrier  $f_c$  de igual magnitud pero sentidos opuestos
- FSK es más resistente a errores que ASK

## ■ FSK en una transmisión full-duplex en una línea telefónica

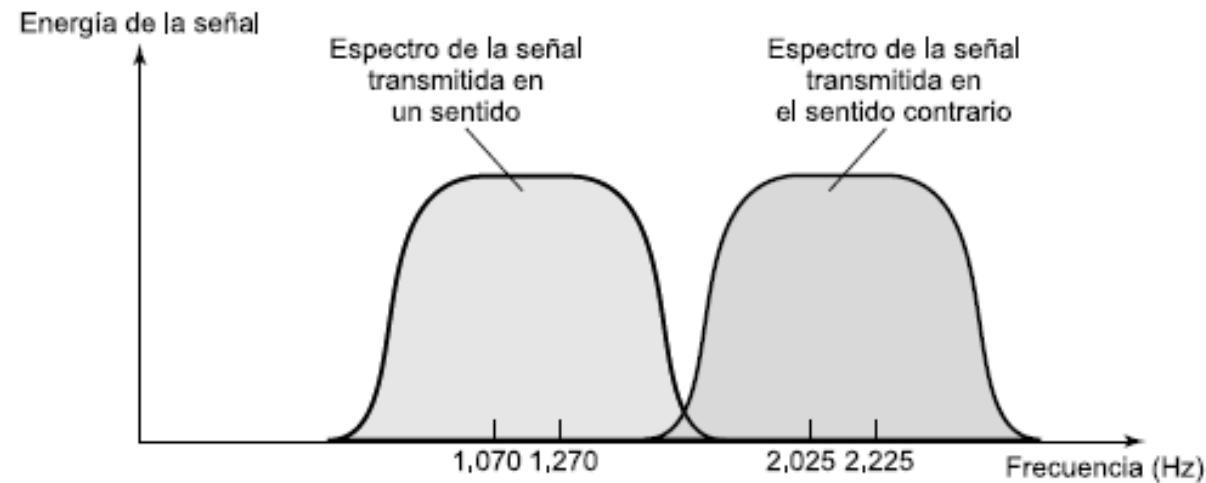


Figura 5.8. Transmisión FSK *full-duplex* en una línea de calidad telefónica.

# PSK

- 0 binario se representa con una señal con la misma fase de la señal anterior
- 1 binaria tiene su señal desfasada  $\pi$  radianes respecto de la señal anterior
- $s(t) = \begin{cases} A\cos(2\pi f_c t + \pi) & 1 \text{ binario} \\ A\cos(2\pi f_c t) & 0 \text{ binario} \end{cases}$
- Si el desplazamiento ya no es  $\pi$  sino  $\pi/4$  cada elemento de señal representa dos bits de datos
- Así, se usa más eficazmente el ancho de banda

$$\blacksquare s(t) = \begin{cases} A\cos\left(2\pi f_c t + \frac{1}{4}\pi\right) & 11 \\ A\cos\left(2\pi f_c t + \frac{3}{4}\pi\right) & 10 \\ A\cos\left(2\pi f_c t + \frac{5}{4}\pi\right) & 00 \\ A\cos\left(2\pi f_c t + \frac{7}{4}\pi\right) & 01 \end{cases}$$

- Cada ángulo podría tener más de una amplitud representándose 3 bits por cada elemento de señal

## 2.2.3 Datos analógicos, señales digitales

- El proceso se denomina *digitalización*
- Dispositivo usado es el *codec*
- Estas señales luego se modulan para convertirlas en señales analógicas para transmitirlas a grandes distancias
- Hay dos técnicas de digitalización, codificación o modulación:
  - Modulación por codificación de impulsos
  - Modulación delta

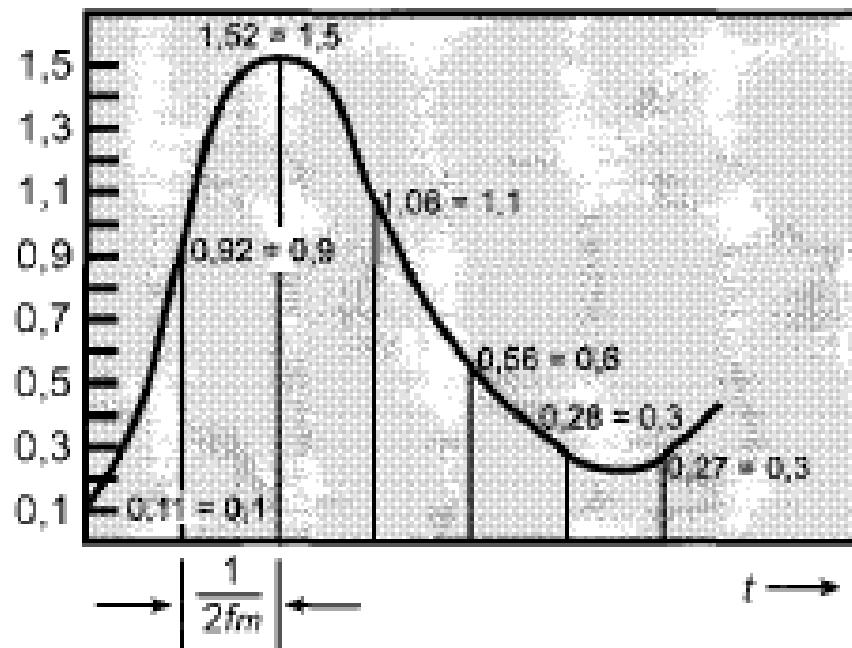
# Modulación por codificación de impulsos

- PCM Pulse Code Modulation
- Se basa en el **Teorema de Muestreo** de Henry Nyquist-1928
- Estas muestras se llaman PAM *Pulse Amplitude Modulation*
- A cada muestra PAM se asigna un código digital

# Teorema de muestreo de Nyquist

- Si se pasa una señal analógica en banda base<sup>1</sup> a través de un filtro **paso-bajo** de ancho de banda B, esta se puede reconstruir matemáticamente por completo, tomando solamente un número de muestras a intervalos regulares igual al doble de la frecuencia máxima en cada segundo (2B muestras/seg).
- La señal analógica se puede reconstruir por completo a partir de estas muestras
- No tiene sentido muestrear más rápido ya que las componentes de frecuencias mayores a B ya fueron filtradas

<sup>1</sup> Señal en banda base: Señal en baja frecuencia antes de sufrir modulación



## Muestras PAM

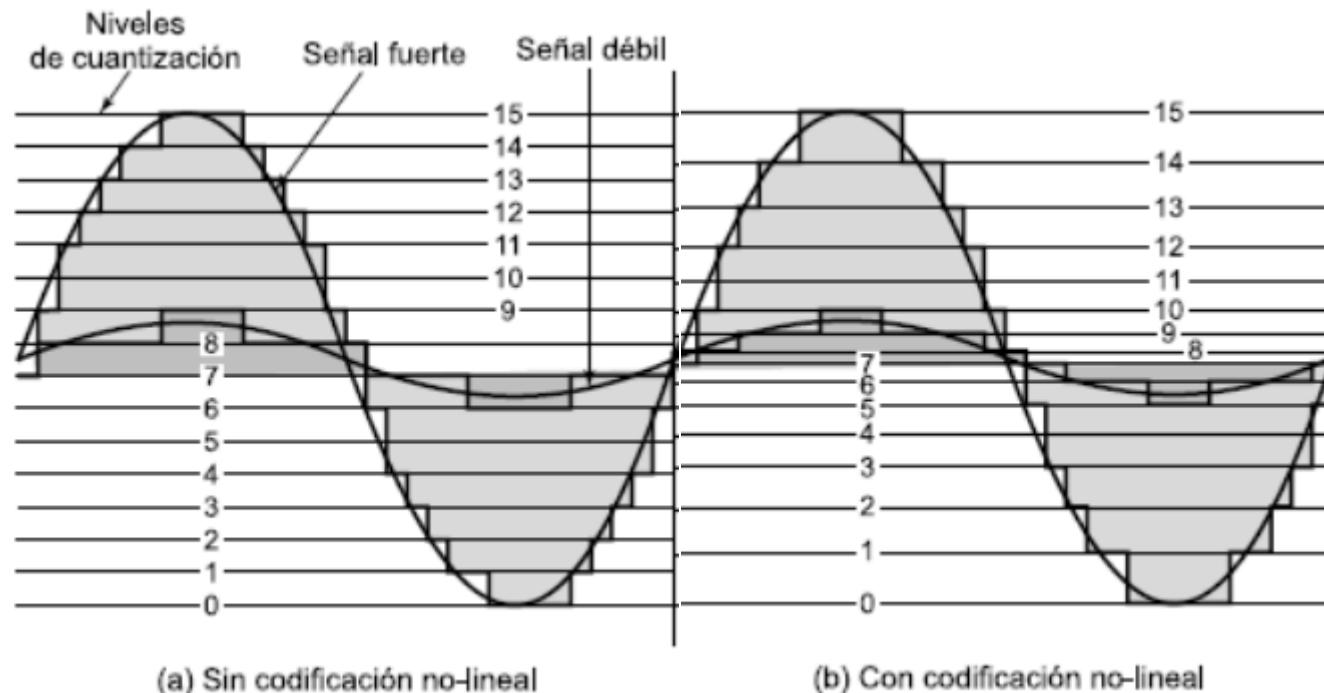
Dígitos	Equivalentes en binario	Forma de onda PCM
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	
10	1010	
11	1011	
12	1100	
13	1101	
14	1110	
15	1111	

Con 4 bits se alcanzan 16 niveles de cuantización

- A mayor número de niveles, mejor calidad de digitalización
- Ejemplo
  - Cada muestra se representa con 8 bits, ¿cuántos niveles de cuantización se tienen?
  - Con 8 bits se alcanzan  $2^8 = 256$  niveles
  - ¿Cuál es la velocidad de transmisión que se requiere?
  - Rango de frecuencias de la voz es 300 a 4000 Hz
  - La frecuencia más alta es 4000 Hz
  - Es suficiente  $2 \times 4000 = 8000$  muestras en un segundo (Nyquist)
  - El ancho de banda requerido es  $8000 \text{ muestras} \times 8 \text{ bits} = 64 \text{ kbps}$
- En el receptor se invierte el proceso para obtener la señal analógica

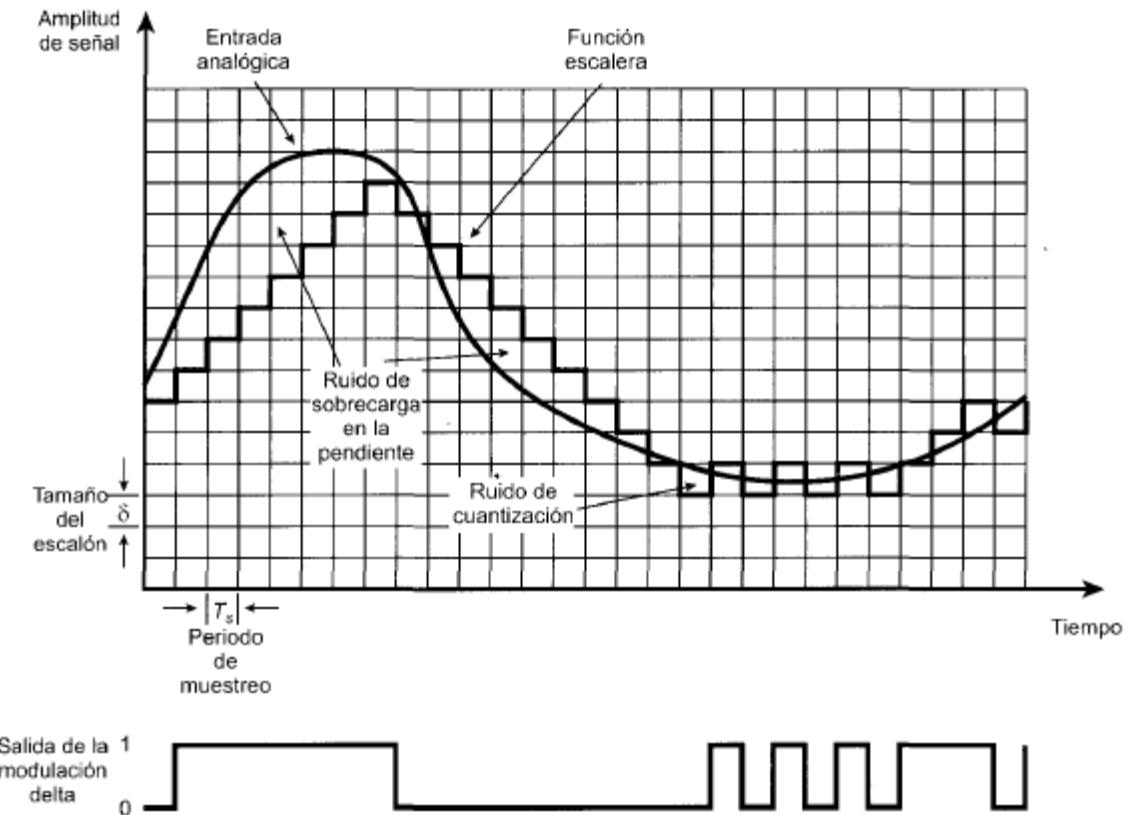
# Codificación no lineal

- PCM se refina con técnicas de cuantificación no lineal
- Los niveles no están igualmente separados
- Se usan más niveles de cuantificación para señales de poca amplitud



# Modulación Delta DM

- Es una mejora a PCM
- Cada muestra sube o baja un nivel  $\delta$
- DM tiene un comportamiento binario
- Cada muestra se representa con un 0 o 1
- Ventajas de DM: sencillez y poco uso de ancho de banda



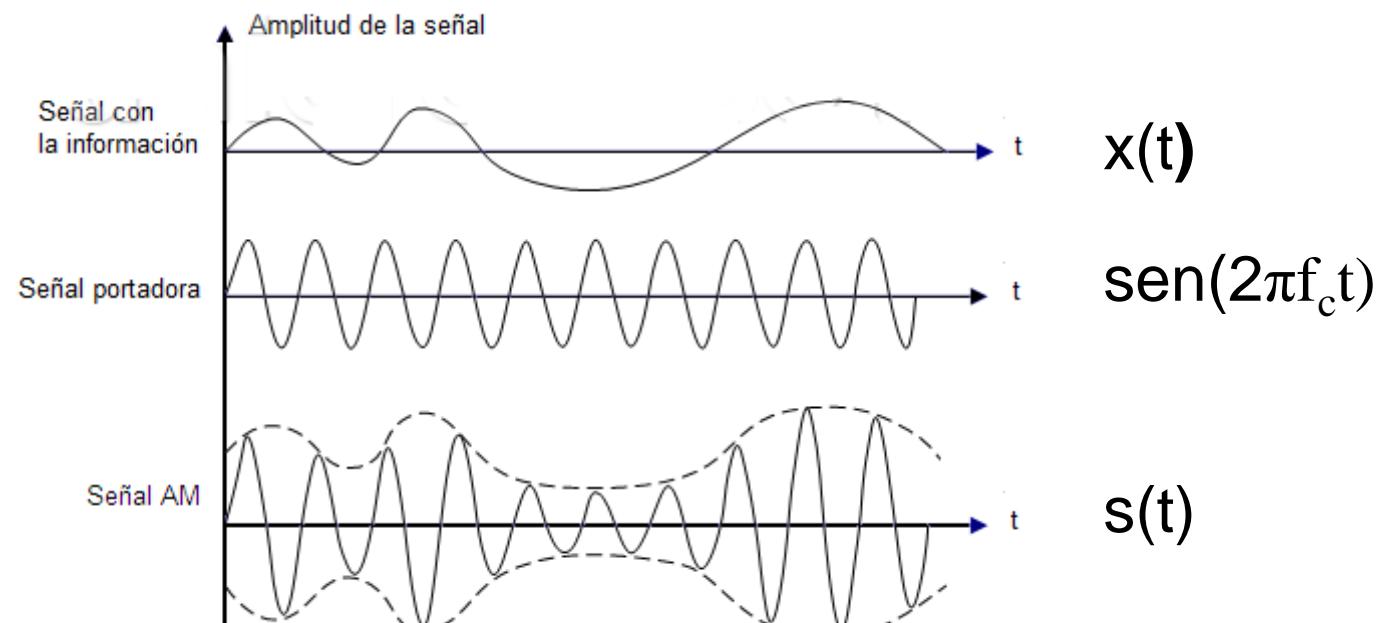
## 2.2.4 Datos analógicos, señales analógicas

- Razones para hacerlo
  - 1. Para transmitir por medios inalámbricos
    - Es imposible transmitir en banda base (4 KHz) porque las antenas tendrían que ser muy grandes.  $\lambda = cT = c/f$
    - Medios inalámbricos utilizan frecuencias elevadas: 10 KHz, MHz, GHz
  - 2. Para realizar multiplexión por división de frecuencia FDM
- Existen tres técnicas de modulación: AM, FM, PM



(c) 1998 JAMES P. HAWKINS - WA2WHV  
95

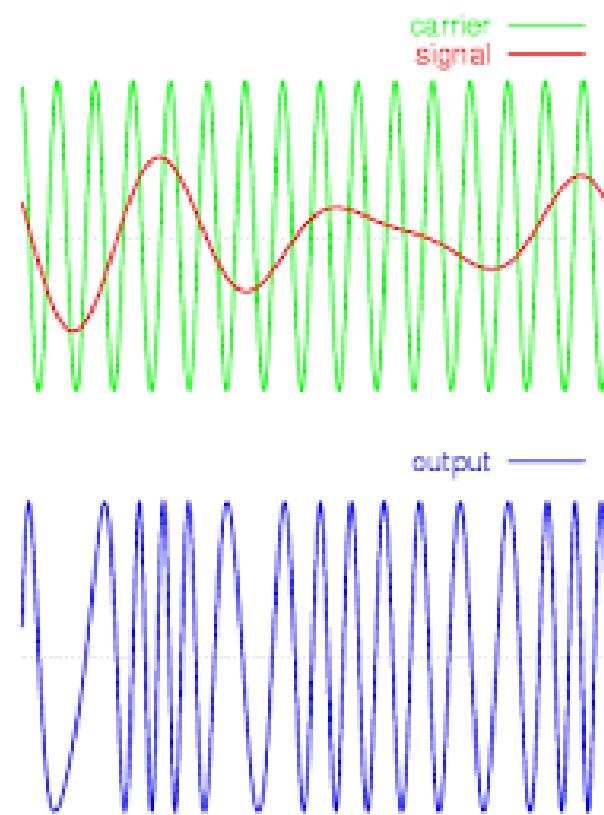
# AM

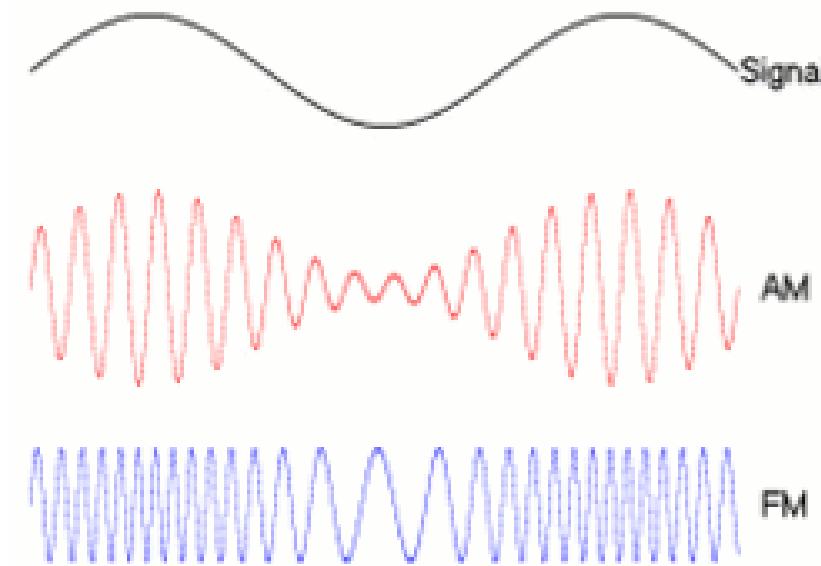


$$s(t) = [1 + n_a x(t)] \sin(2\pi f_c t)$$

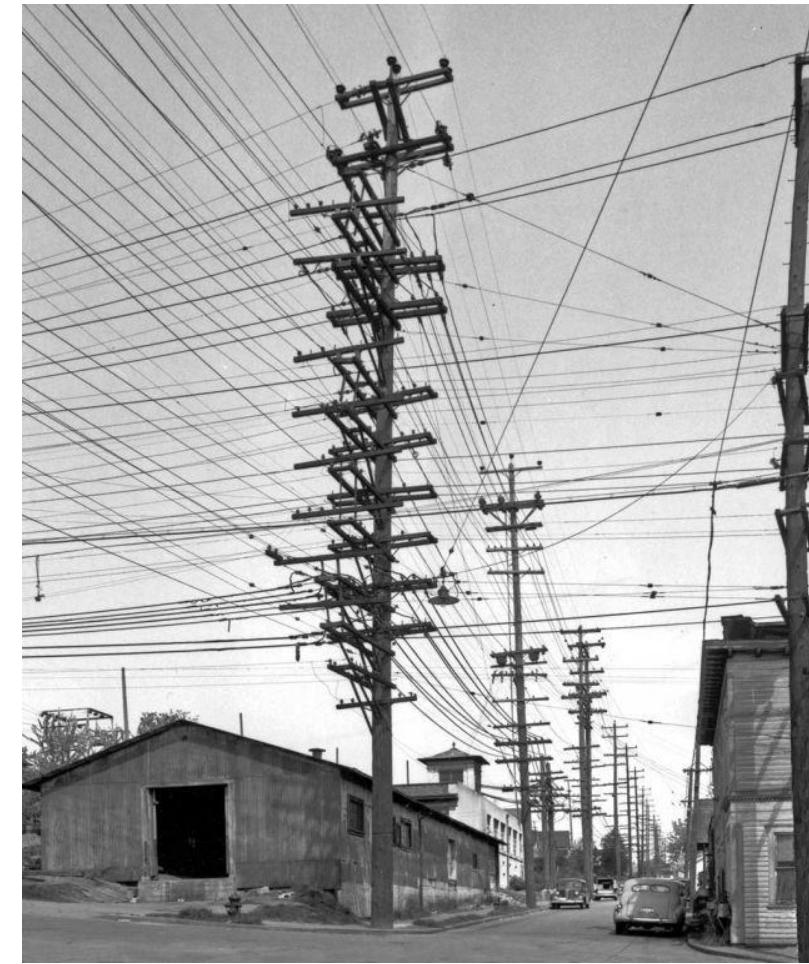
$n_a$  = coeficiente entre amplitud de  $x(t)$  y amplitud de la portadora

# FM





# Transmisión sin multiplexión

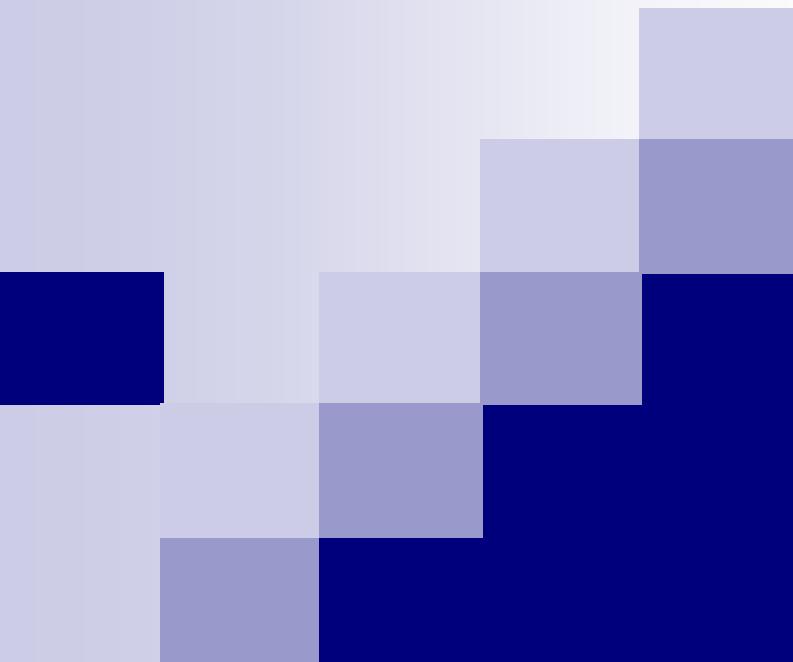


Solo una comunicación por cada conductor

# Contenido – Andrew S. Tanenbaum

## Redes de Computadoras

- Base teórica de la comunicación de datos
- Medios de transmisión guiados
- Transmisión inalámbrica
- Satélites de comunicaciones



## 2.3 Base teórica de la comunicación de datos

Redes de Computadoras

Andrew S. Tanenbaum

- Es posible transmitir información variando algunas propiedades eléctricas:  $V, I$  aplicadas a un medio físico
- La capacidad de transmisión de datos de los medios físicos de transmisión tiene unos límites máximos que dependen sus características intrínsecas

# Latencia en la comunicación

- Se debe a retardos de transmisión y retardos de propagación
- Retardo de transmisión es el tiempo que se demora el hardware en convertir los datos en una señal electromagnética
- Retardo de propagación es el tiempo que demora viajar la señal desde el origen hasta el destino. A mayor distancia mayor tiempo de propagación
- Si la señal electromagnética viaja por un material sólido como el cobre o la fibra óptica, la velocidad de propagación disminuye cuando aumenta su frecuencia en Hz
- En el vacío una señal electromagnética viaja a la velocidad de la luz, independientemente de su frecuencia

# Perturbaciones en la transmisión

- En sistema de comunicación, la señal recibida es diferente a la transmitida
- Esto se debe a factores que distorsionan la señal
  - Atenuación de la señal. Afecta la distancia y la frecuencia
  - Distorsión de la señal. Algunas componentes de la señal viajan más rápido que otras componentes
  - Ruido introducido por factores externos o por el mismo medio de transmisión

# Atenuación de la señal

- La señal debe tener suficiente potencia para que el receptor la “entienda”
- El nivel de la señal debe ser mayor al nivel del ruido
- La señal se atenúa a medida que aumenta la distancia y la frecuencia
- Si la señal llega con demasiada potencia, la electrónica del receptor se puede saturar o sobrecargar, pudiendo producir un daño permanente

## 2.3.1 Análisis de Fourier



Francia 1768 - 1830

- Jean-Baptiste Fourier
- Cualquier función **periódica**  $g(t)$  se puede construir sumando una cantidad, posiblemente **infinita**, de **senos y cosenos**:

# Serie de Fourier

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft) \quad (1)$$

- $f = 1/T$  frecuencia fundamental o primer armónico
- El período  $T$  de  $g(t)$  es el período de la primera armónica  
Frecuencias de los demás armónicos son múltiplos de  $f$
- Para reconstruir  $g(t)$  se necesita  $f$  y las amplitudes  $a_n$  y  $b_n$

Las  $a_n$  se calculan multiplicando ambos lados de (1) por  $\sin(2\pi kft)$ , e integrando de 0 a T

Ya que:

$$\int_0^T \sin(2\pi kft) \sin(2\pi nft) dt = \begin{cases} 0 & \text{para } k \neq n \\ T/2 & \text{para } k = n \end{cases}$$

Solo permanece un término de la integral:  $a_n$

$$a_n = \frac{2}{T} \int_0^T g(t) \sin(2\pi nft) dt$$

c y la integral de  $b_n$  desaparecen

Las amplitudes  $b_n$  se calculan multiplicando ambos lados de la ecuación (1) por  $\cos(2\pi kft)$ , e integrando de 0 a T.

Ya que:

$$\int_0^T \cos(2\pi kft) \cos(2\pi nft) dt = \begin{cases} 0 & \text{para } k \neq n \\ T/2 & \text{para } k = n \end{cases}$$

Sólo perdura un término de la sumatoria:  $a_n$

$$b_n = \frac{2}{T} \int_0^T g(t) \cos(2\pi nft) dt$$

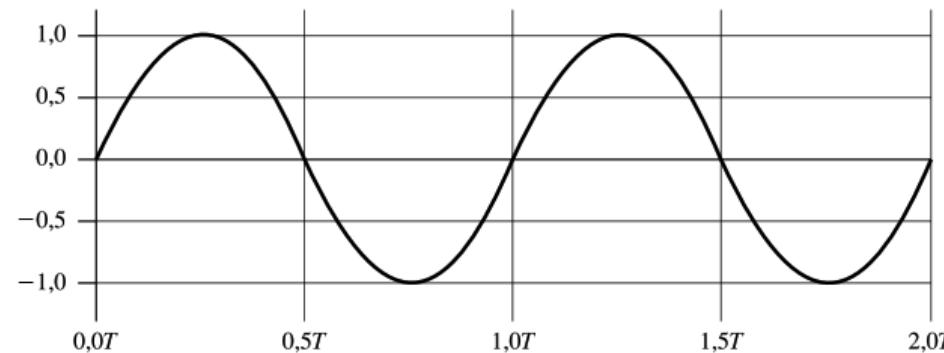
c y la sumatoria de  $a_n$  desaparecen

La amplitud **c** se calcula integrando directamente de 0 a T.

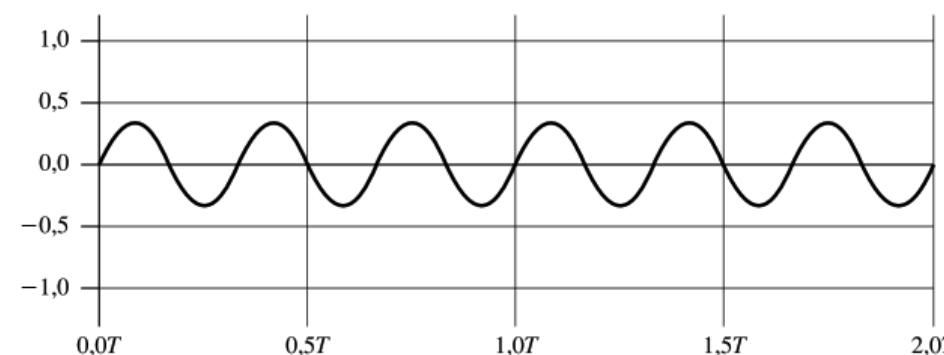
$$c = \frac{2}{T} \int_0^T g(t) dt$$

Las sumatoria de  $a_n$  y  $b_n$  desaparecen.

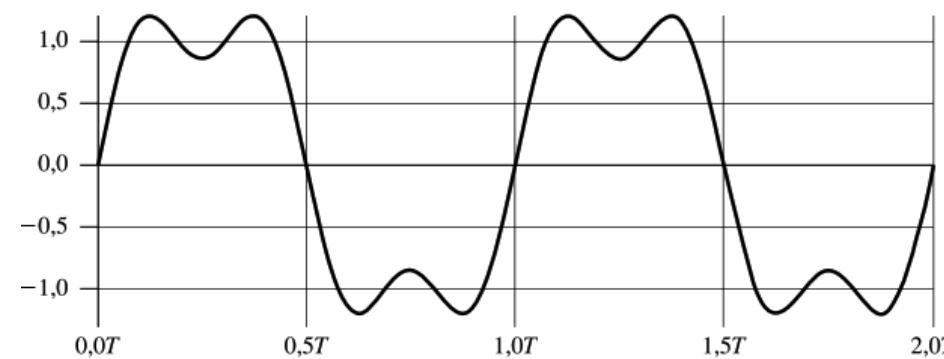
Los estudiantes deben demostrar las expresiones para encontrar  $a_n$ ,  $b_n$  y  $c$



$$\sin(2\pi ft)$$

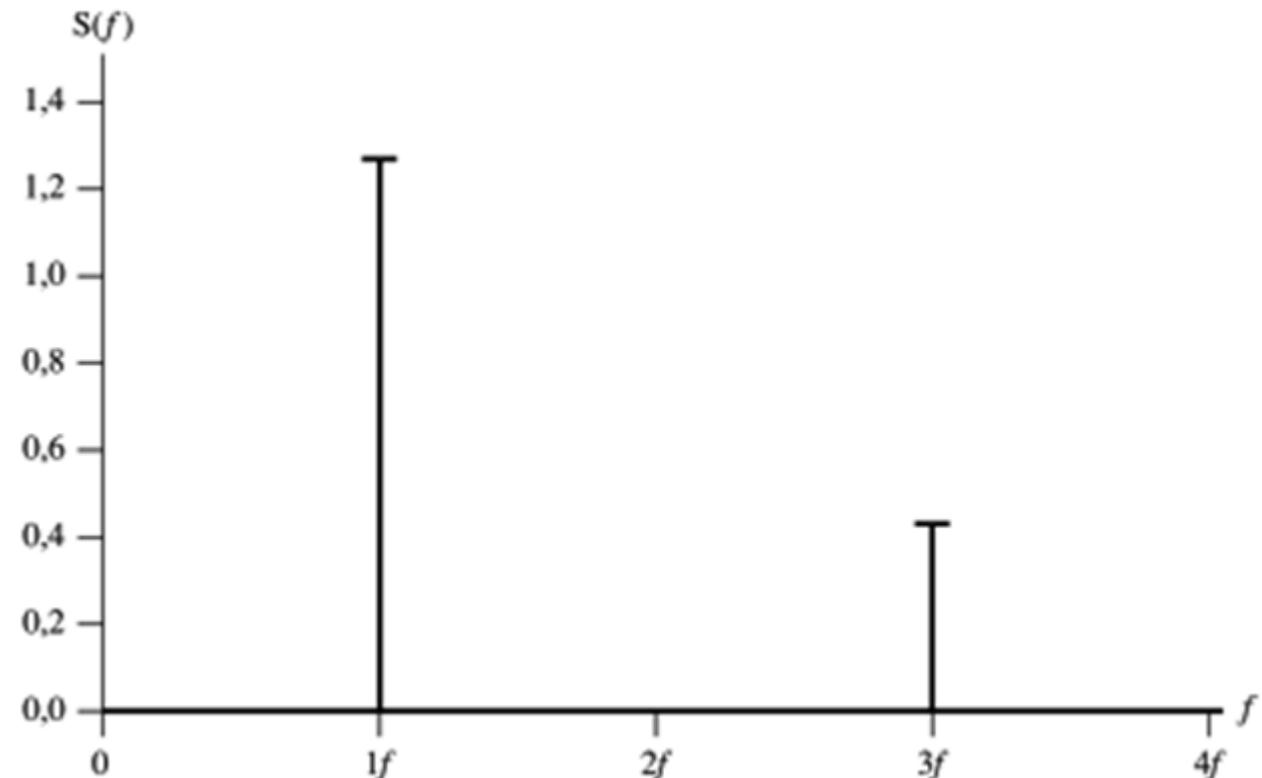


$$(1/3)\sin(2\pi(3f)t)$$



$$(4/\pi)[\sin(2\pi ft) + (1/3)\sin(2\pi(3f)t)]$$

# Dominio de la frecuencia

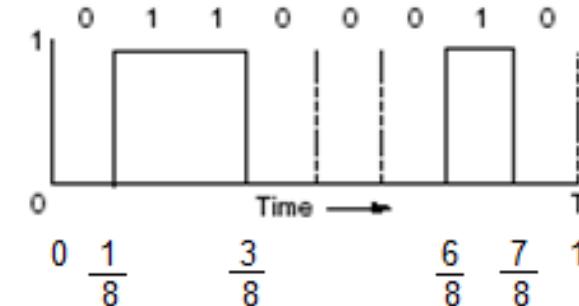


$$s(t) = \frac{4}{\pi} [\sin(2\pi ft) + \frac{1}{3} \sin(2\pi(3f)t)]$$

## 2.3.2 Señales de ancho de banda limitado

- Una señal que tiene una duración finita, se puede expresar como una serie de Fourier suponiendo que la señal se repite infinitamente
- Ejemplo: Transmisión del carácter ASCII “b” 01100010
- Amplitudes  $\left(\sqrt{a_n^2 + b_n^2}\right)$  de la raíz media cuadrática *rms* de los armónicos
- La energía en cada frecuencia o armónico es proporcional a su amplitud

carácter ASCII “b” 01100010



$$g(t) = 0 \text{ en } [0, 1/8T], [3/8T, 6/8T] \text{ y } [7/8T, T]$$

$$g(t) = 1 \text{ en } [1/8T, 3/8T] \text{ y } [6/8T, 7/8T]$$

$$a_n = \frac{2}{T} \int_0^T g(t) \sin(2\pi n f t) dt$$

$$a_n = \frac{2}{T} \int_{1/8T}^{3/8T} \sin(2\pi n f t) dt + \frac{2}{T} \int_{6/8T}^{7/8T} \sin(2\pi n f t) dt$$

$$a_n = \frac{1}{\pi n} [\cos(\pi n/4) - \cos(3\pi n/4) + \cos(6\pi n/4) - \cos(7\pi n/4)]$$

$g(t) = 1$  en  $[1/8T, 3/8T]$  y  $[6/8T, 7/8T]$

$$b_n = \frac{2}{T} \int_0^T g(t) \cos(2\pi n f t) dt$$

$$b_n = \frac{2}{T} \int_{\frac{1}{8}T}^{\frac{3}{8}T} \cos(2\pi n f t) dt + \frac{2}{T} \int_{\frac{6}{8}T}^{\frac{7}{8}T} \cos(2\pi n f t) dt$$

$$b_n = \frac{1}{\pi n} [\sin(3\pi n/4) - \sin(\pi n/4) + \sin(7\pi n/4) - \sin(6\pi n/4)]$$

$g(t) = 1$  en  $[1/8T, 3/8T]$  y  $[6/8T, 7/8T]$

$$c = \frac{2}{T} \int_0^T g(t) dt$$

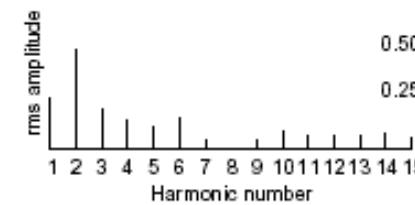
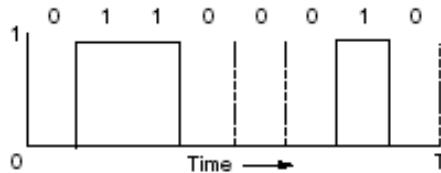
$$c = \frac{2}{T} \int_{\frac{1}{8}T}^{\frac{3}{8}T} dt + \frac{2}{T} \int_{\frac{6}{8}T}^{\frac{7}{8}T} dt$$

$$c = 3/4$$

componente  
o armónico

<i>n</i>	<i>frecuencia</i>	<i>a<sub>n</sub></i>	<i>b<sub>n</sub></i>	<i>amplitud de RMS</i>
1	<i>1f</i>	-0,225	0,093	0,243
2	<i>2f</i>	-0,159	-0,477	0,502
3	<i>3f</i>	-0,075	-0,181	0,196
4	<i>4f</i>	0,159	0	0,159

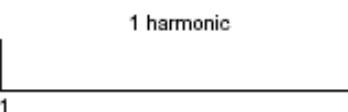
$$RMS \text{ o valor eficaz} = \sqrt{\frac{{a_n}^2 + {b_n}^2}{2}}$$



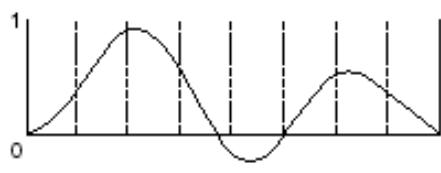
(a)



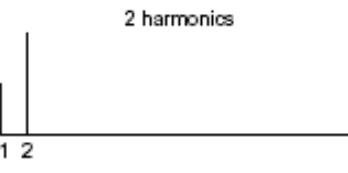
(b)



0.243

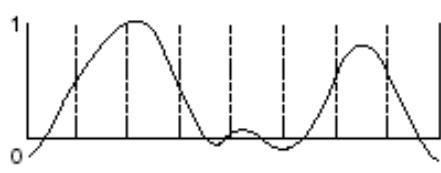


(c)

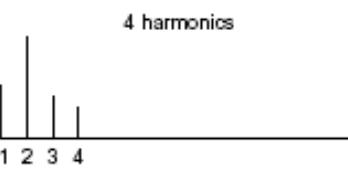


2 harmonics

0.243; 0,502

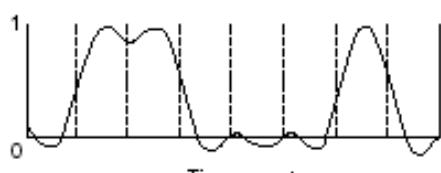


(d)

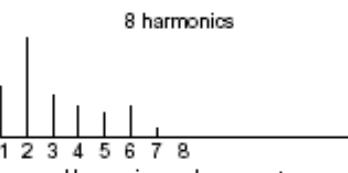


4 harmonics

0.243; 0,502; 0.196; 0.159



(e)



8 harmonics

- Con 7 armónicos es suficiente para que el receptor distinga los 1s y 0s binarios del carácter ASCII “b”
- Es suficiente el ancho de banda  $7f - 1f = 6f$
- Tener más ancho de banda es innecesario

# Series de Fourier

Patrón que se repite en un período  $T$ : 8 bits

$f_{corte} f_c = 3000 \text{ Hz}$  línea telefónica

Regla de 3:  $\frac{8b}{bps} \frac{T}{1s} \quad T = \frac{8 \text{ bits}}{bps} \quad f \text{ del } 1^{\text{er}} \text{ armónico} = \frac{1}{T} \quad \# \text{ armónicos} = \frac{f_c}{f \text{ del } 1^{\text{er}} \text{ armónico}}$

Bps	T (mseg)	Primer armónico (Hz)	Núm. de armónico transmitidos
300	26.67	37.5	80
600	13.33	75	40
1200	6.67	150	20
2400	3.33	300	10
4800	1.67	600	5
9600	0.83	1200	2
19200	0.42	2400	1
38400	0.21	4800	0

Para una misma  $f_c$  (ancho de banda) a mayor velocidad bps menor número de armónicos y menor calidad de la señal

- Una onda se atenúa más mientras mayor es la distancia recorrida
- Una onda se atenúa más mientras mayor es su frecuencia
- Si todas las componentes de Fourier de una señal cuadrada se atenuaran en la misma proporción, la señal se atenuara sin distorsión, manteniendo la misma forma cuadrada
- Pero, los medios de transmisión atenúan las componentes de Fourier en diferente grado, causando distorsión en la señal

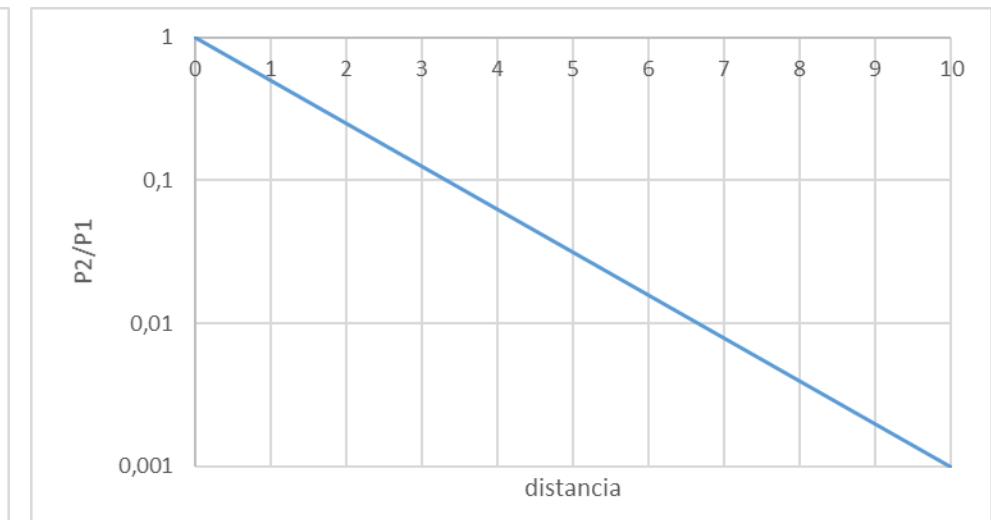
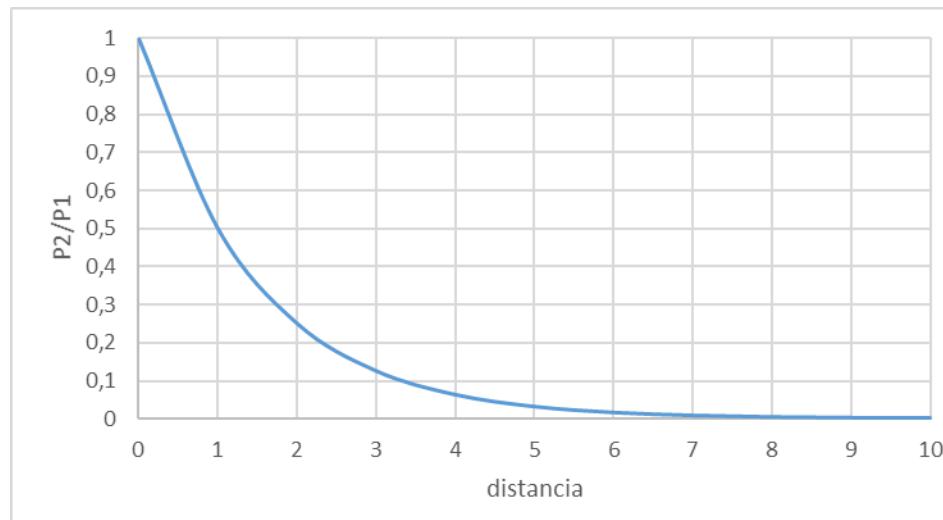
## *Ancho de banda efectivo o ancho de banda*

- Es el rango de frecuencias de 0 hasta  $f_c$  en la que la señal, sumando todos los armónicos hasta  $f_c$ , llega atenuada a la mitad de la amplitud original
- $f_c$  se llama frecuencia de corte
- Este rango de frecuencias tiene la mayor parte de la energía de la señal
- Este es un rango pequeño
- La amplitud de la señal es directamente proporcional a su potencia:  $P = VI$

- En medios **guiados**, la atenuación de la onda es constante, es decir lineal, o sea logarítmica
- En estos casos la atenuación se expresa en decibelios (dB) por unidad de longitud
- Expresiones logarítmicas facilitan las operaciones matemáticas:
  - la multiplicación se transforma en suma
  - la división en resta
  - la potenciación en multiplicación
  - la radicación en división

## A distancias iguales, atenuaciones iguales de la potencia de la señal

distancia	P2/P1
0	1
1	0,5
2	0,25
3	0,125
4	0,0625
5	0,03125
6	0,015625
7	0,0078125
8	0,00390625
9	0,00195313
10	0,00097656



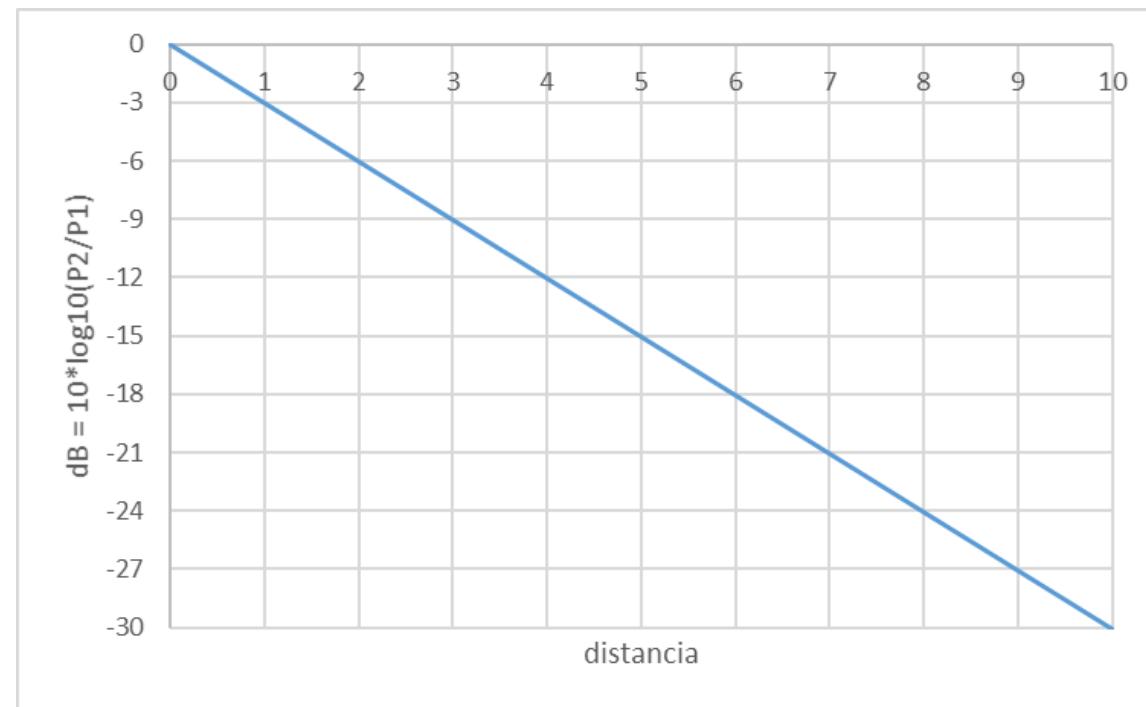
Escala logarítmica del eje vertical.  
Atenuación lineal

## A distancias iguales, atenuaciones iguales

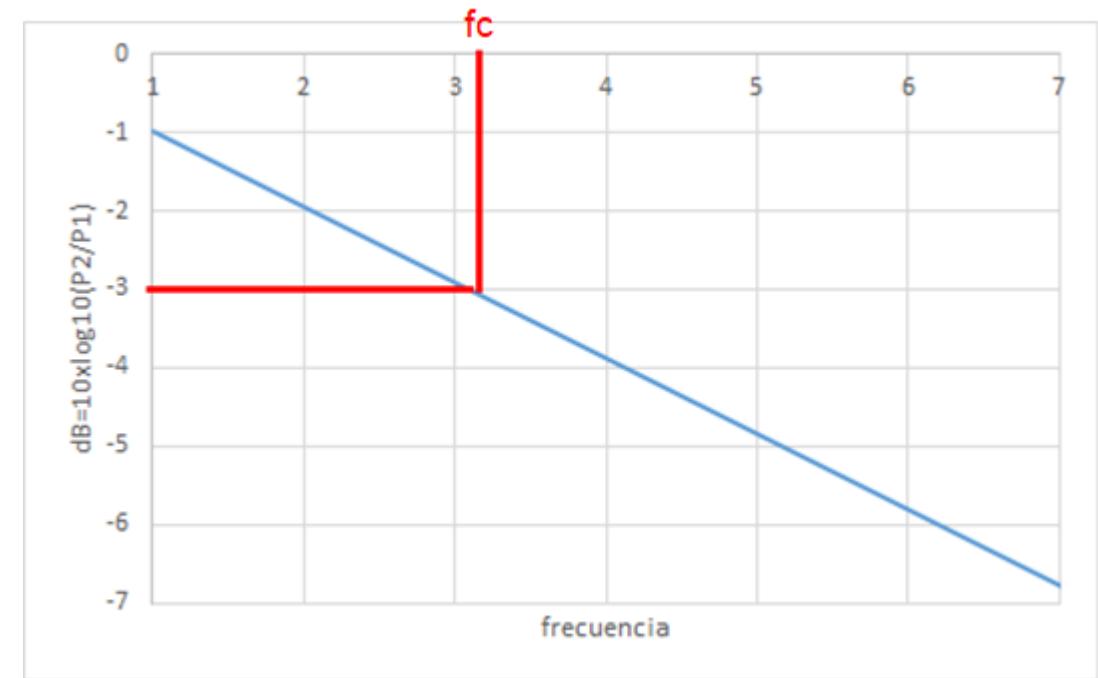
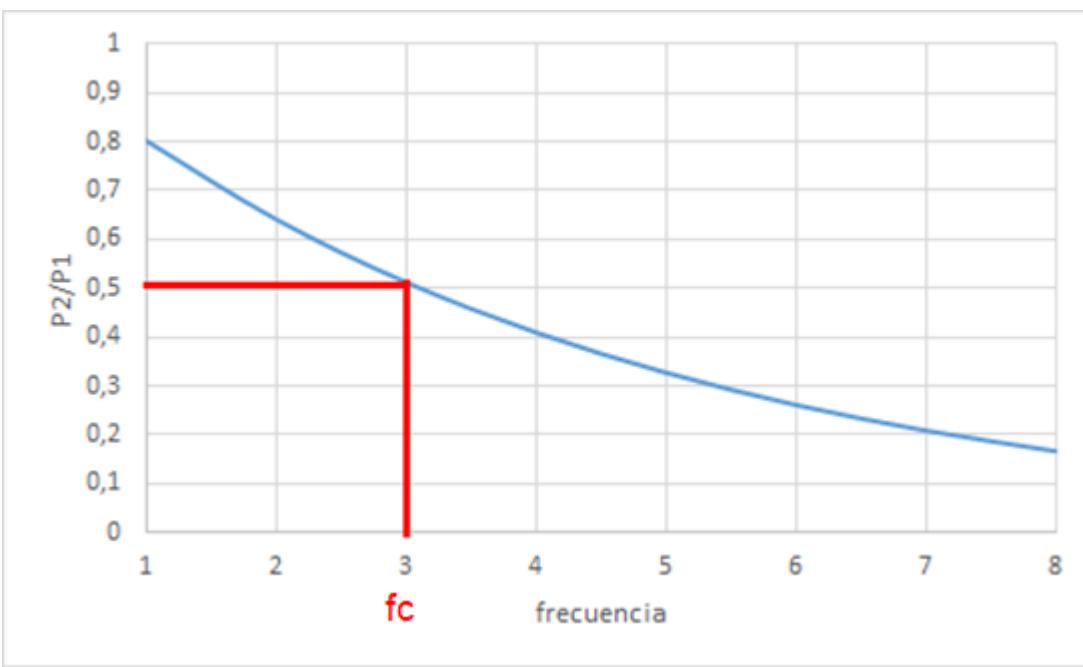
$$\text{Si } P_2 = \frac{P_1}{2}$$

$$10 \log_{10} \frac{P_2}{P_1} = 10 \log_{10} \frac{P_1/2}{P_1} = 10 \log_{10} \frac{1}{2} = -3 \text{ dB}$$

distancia	P2/P1	dB
0	1	0
1	0,5	-3,0103
2	0,25	-6,0206
3	0,125	-9,0309
4	0,0625	-12,041
5	0,0313	-15,051
6	0,0156	-18,062
7	0,0078	-21,072
8	0,0039	-24,082
9	0,002	-27,093
10	0,001	-30,103



Para una distancia determinada:



$$P_1 = V_1 I \quad V = IR \quad P_1 = \frac{V_1^2}{R} \quad P_2 = \frac{V_2^2}{R} \quad \frac{P_2}{P_1} = \frac{V_2^2}{V_1^2} \quad \frac{V_2}{V_1} = \sqrt{\frac{P_2}{P_1}}$$

$$Si \quad \frac{P_2}{P_1} = \frac{1}{2} \quad \frac{V_2}{V_1} = \sqrt{\frac{1}{2}} \quad V_2 = \frac{V_1}{\sqrt{2}} = 0.707V_1$$

- El ancho de banda es una **propiedad física** del medio de transmisión
- Depende del tipo de material, grosor, longitud, construcción
- Todo medio de transmisión soporta una limitada banda de frecuencias
- Un medio de transmisión se comporta como un filtro electrónico paso-bajo
- Esto limita la tasa de datos a ser transportada
- En otras ocasiones, se utilizan filtros para limitar el ancho de banda a cada cliente: sistema telefónico (4000 Hz)

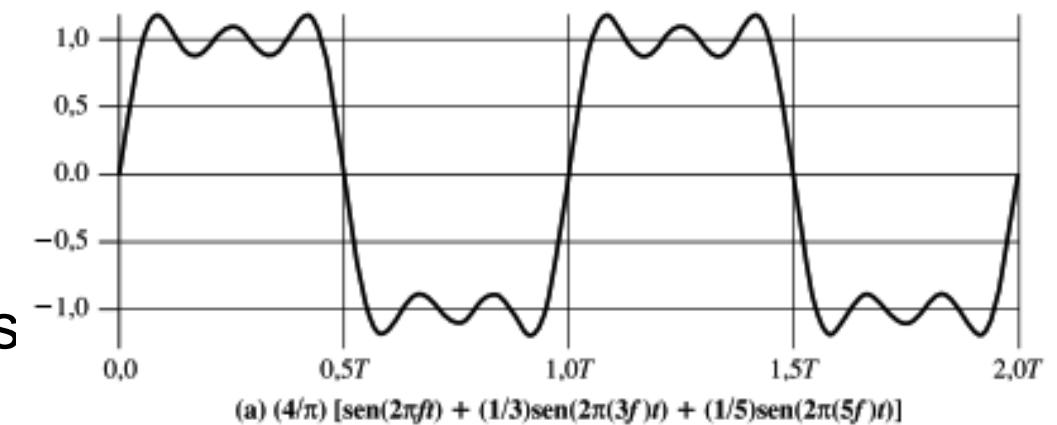
# Relación entre velocidad de transmisión y ancho de banda

## ■ Supuestos:

- Sistema de transmisión **digital**
- Transmisión de 1s y 0s alternantes (onda cuadrada)
- ¿Cuál es la velocidad de transmisión?
- No se considera el ruido del medio

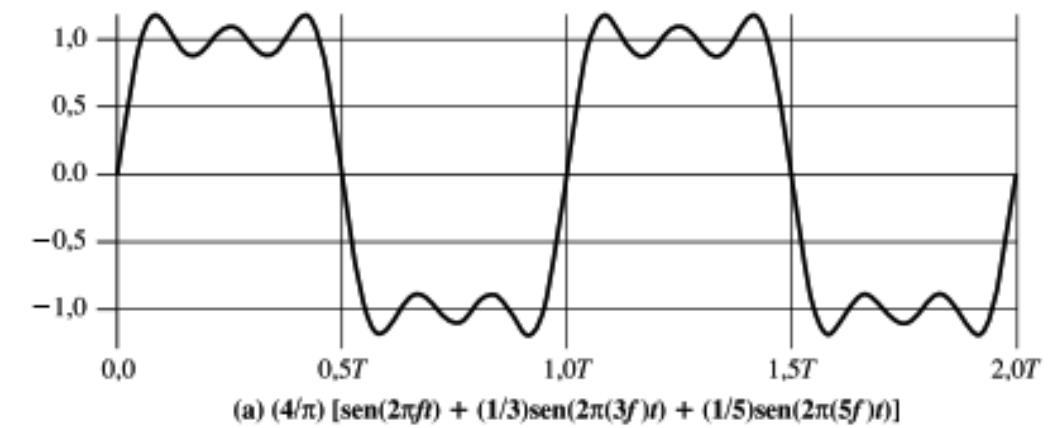
# CASO I

- $s(t) = (4/\pi)[\sin(2\pi ft) + (1/3)\sin(2\pi 3ft) + (1/5)\sin(2\pi 5ft)]$
- Pocos armónicos: onda distorsionada
- Frecuencia fundamental  $f = 1 \text{ MHz}$
- $T = 1/f = 1 \mu\text{s}$
- $B = f_{\max} - f_{\min} = 5f - f = 4 \text{ MHz}$
- En  $T$  se representan 2 bits: 1 bit =  $0.5 \mu\text{s}$
- La velocidad es **2 Mbps**



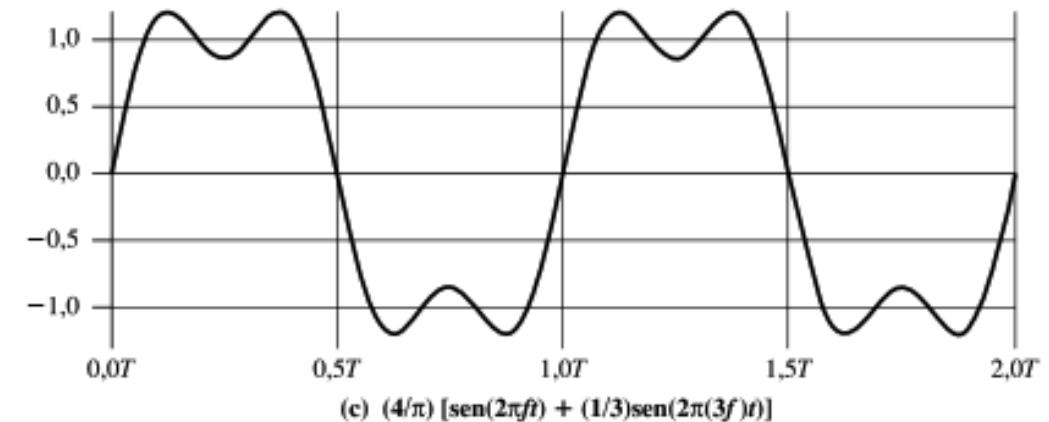
## CASO II

- $s(t) = (4/\pi)[\sin(2\pi ft) + (1/3)\sin(2\pi 3ft) + (1/5)\sin(2\pi 5ft)]$
- Pocos armónicos: onda distorsionada
- Frecuencia fundamental  $f = 2 \text{ MHz}$
- $T = 1/f, T = 0,5 \mu\text{s}$
- $B = f_{\max} - f_{\min} = 5f - f = 8 \text{ MHz}$
- 1 bit =  $0.25 \mu\text{s}$
- La velocidad es **4 Mbps**
- A mayor frecuencia fundamental  $\Rightarrow$  más velocidad bps



## CASO III

- $s(t) = (4/\pi)[\sin(2\pi ft) + (1/3) \sin(2\pi 3ft)]$
- Onda más distorsionada
- $f = 2 \text{ MHz}$
- $B = f_{\max} - f_{\min} = 3f - f = 4 \text{ MHz}$
- $T = 1/f = 0,5 \mu\text{s}$
- 1 bit = 0.25  $\mu\text{s}$
- La velocidad es **4 Mbps**
- A menor número de armónicos -> menor ancho de banda -> menor calidad de la señal



# Distorsión de la señal por retardo

- La velocidad de propagación de la señal en el medio varía con la frecuencia
- Las distintas componentes en frecuencia llegan al receptor en diferentes instantes de tiempo
- Esto distorsiona a la señal
- Esto es crítico en la transmisión de datos digitales produciendo alteraciones entre bits posteriores

# Distorsión de la señal por ruido

- Este factor es el que más afecta la transmisión de señales
- Hay cuatro clases de ruido
  - Ruido térmico
  - Ruido de intermodulación
  - Diafonía
  - Ruido impulsivo

# Ruido térmico

- También llamado ruido blanco
- Se debe al calentamiento del medio y dispositivos electrónicos
- Este ruido tiene todas las componentes de frecuencia
- Es imposible eliminarlo

# Expresión matemática

- $N_0$  densidad de potencia del ruido térmico [watos / Hz]
- $N_0 = kT$
- $k = \text{constante de Boltzmann} = 1,3803 \times 10^{-23} \frac{\text{Joule}}{\text{°K}}$
- $T = \text{temperatura en } {}^\circ \text{ Kelvin } (1^\circ K = 1^\circ C + 273)$
- $\frac{\text{Joule}}{\text{°K}} = \frac{\text{Joule}}{\text{°K} \cdot \text{s} \cdot \text{Hz}} = \frac{\text{watio}}{\text{°K} \cdot \text{Hz}} \quad \text{s} \cdot \text{Hz} = 1$
- $k = 1,3803 \times 10^{-23} \frac{\text{watio}}{\text{°K} \cdot \text{Hz}}$
- $N = N_0 B$

# Ejemplo

- $T = 17^{\circ}\text{C} = 273 + 17 = 290^{\circ}\text{K}$
- $B = 10 \text{ MHz}$
- $N = N_0 B \quad N_0 = kT$
- $N = kTB$
- $N = \left(1,3803 \times 10^{-23} \frac{\text{watio}}{^{\circ}\text{K} \cdot \text{Hz}}\right) (290 \ ^{\circ}\text{K})(10^7 \text{Hz})$
- $N = 400,287 \times 10^{-16} \text{ watt}$
- $dBW = 10 \log_{10} \left( \frac{N}{1 \text{ Watt}} \right) = 10 \log_{10} \left( \frac{400,287 \times 10^{-16} \text{ watt}}{1 \text{ Watt}} \right)$
- $N = -13,39 \text{ dBW}$

# Ruido de intermodulación

- Producido por la aparición de señales a frecuencias que son la suma o diferencia de las componentes de la señal  $f_1$  y  $f_2$ , o múltiplos de estas
- Esta aparición se debe a la excesiva potencia de la señal

# Ruido por diafonía

- Debido al acoplamiento entre líneas próximas que transportan señales
- Una evidencia de esto es cuando oímos otra conversación por el teléfono

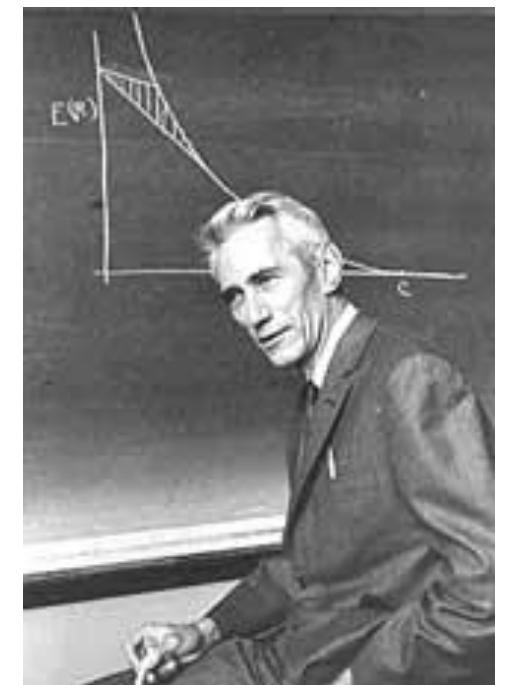
# Ruido impulsivo

- Picos de gran amplitud
- Puede aparecer en diferentes rangos de frecuencia
- Se generan por descargas eléctricas atmosféricas, ruido de motores, explosiones, impactos, etc.

## 2.3.3 Teoría de la información

- Parte de las matemáticas y de las ciencias de la computación. Estudia:
  - Representación de los datos
  - canales de comunicación
  - compresión de datos
  - criptografía o cifrado
  - detección y corrección de errores
- Estudia la cantidad de información y ruido que contiene una señal transmitida

- El estudio de la Teoría la Información lo inició Claude Shannon-1948.
- La información se trata como una **magnitud física**
- Teorema de muestreo de Nyquist
- Teorema de la tasa de datos máxima de un canal
  - Sin ruido. Nyquist
  - Con ruido. Shannon



# Teorema de la tasa de datos máxima de un canal perfecto de Nyquist

- 1924, Henry Nyquist, ingeniero de AT&T.
- **Descubrimiento:** Hasta un canal perfecto (sin ruido), tiene una capacidad de transmisión **finita**.
- La limitación está impuesta por el ancho de banda del canal
- La señal de banda base es una señal original de baja frecuencia no modulada, filtrada con un filtro paso-bajo de un ancho de banda  $B$  (de **0** a  $B$  Hz)
- $B$  es la frecuencia máxima del filtro paso-bajo



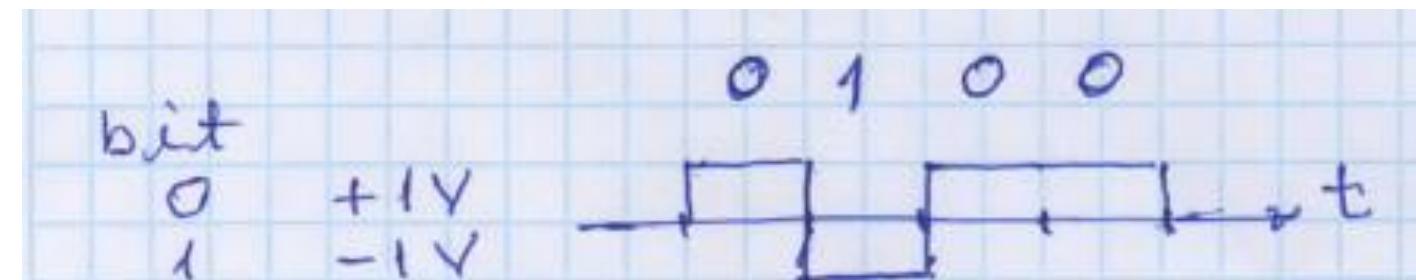
1889 - 1976

Nyquist: Tasa de datos máxima  $C = 2B \log_2 N$  bps

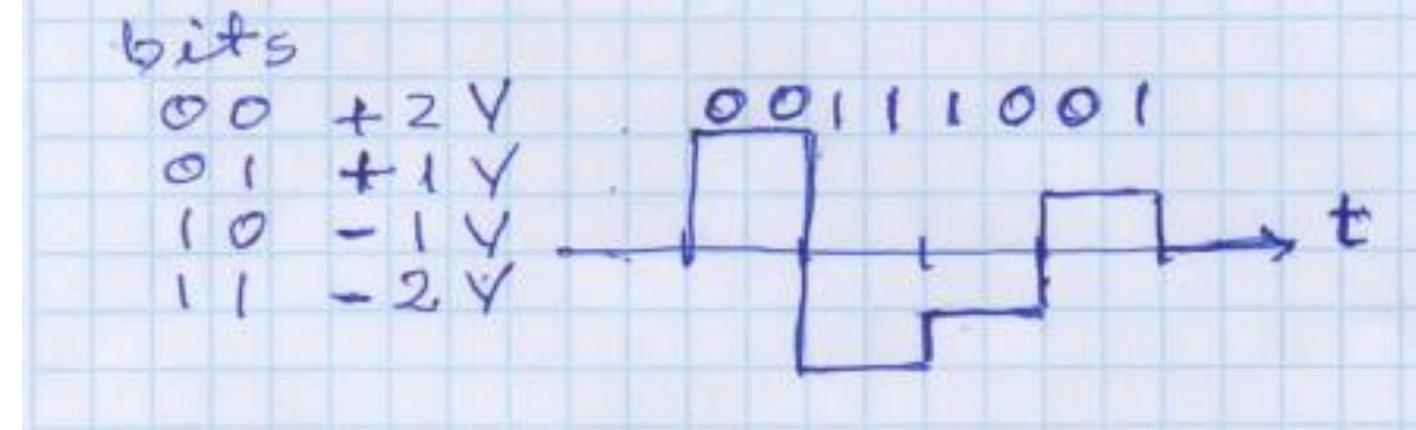
B = ancho de banda

N = número de niveles de tensión

$$N = 2, C_{\max} = 2B$$



$$N = 4, C_{\max} = 4B$$



## Caso I

- $s(t) = (4/\pi)[\sin(2\pi ft) + (1/3)\sin(2\pi 3ft) + (1/5)\sin(2\pi 5ft)]$
- Pocos armónicos: onda distorsionada
- Frecuencia fundamental  $f = 1 \text{ MHz}$
- $T = 1/f = 1 \mu\text{s}$
- $B = f_{\max} - f_{\min} = 5f - f = 4 \text{ MHz}$
- 1 bit =  $0.5 \mu\text{s}$
- La velocidad es **2 Mbps**
- Tasa de datos **máxima** de Nyquist  $C = 2 \times 5 \times 10^6 = 10 \text{ Mbps}$ ,  $N = 2$  niveles de tensión

- Con el mismo ancho de banda  $B = 4$  MHz
- Si  $N = 4$ , cada nivel representa 2 bits. Se cambia la codificación de los datos
- Tasa de datos máxima de Nyquist  $C = 2B\log_2 N = 2 \times 4 \times 10^6 \log_2 4$
- $C = 16$  Mbps
- Si se aumenta el número de niveles, el número de armónicos no es capaz de dibujar la onda y el receptor no podría interpretar los 0s y 1s

# Transmisión en canal con ruido térmico

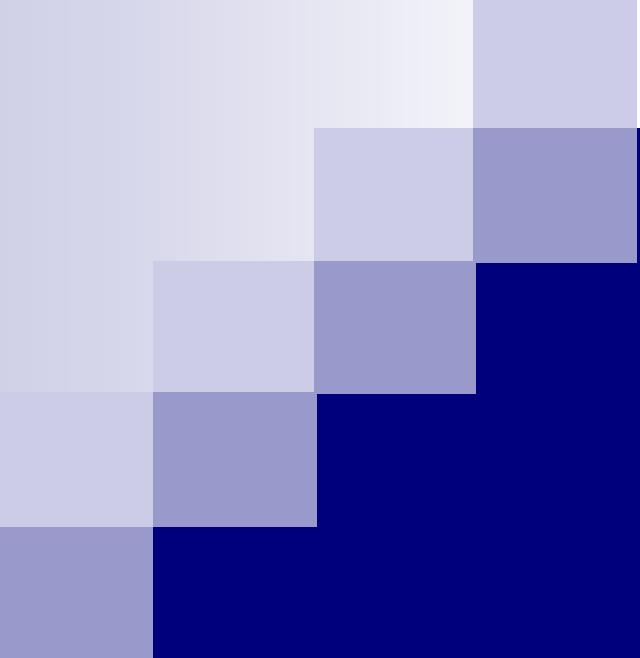
- Descubrimiento de Claude Shannon, 1948
- Este ruido se debe al movimiento de las moléculas del medio
- SNR = potencia de la señal / potencia del ruido
- $\text{SNR}_{\text{dB}} = 10 \log_{10}(\text{S/N})$  decibeles

SNR	$\text{SNR}_{\text{dB}}$
10	10
100	20
1000	30

- Según Shannon, la tasa de datos máxima =  $B \log_2(1 + \text{SNR})$ 
  - $B$  = ancho de banda
- No importa cuántos niveles de señales se usen
- Recordar que:  $\log_a x = \log_{10} x / \log_{10} a$

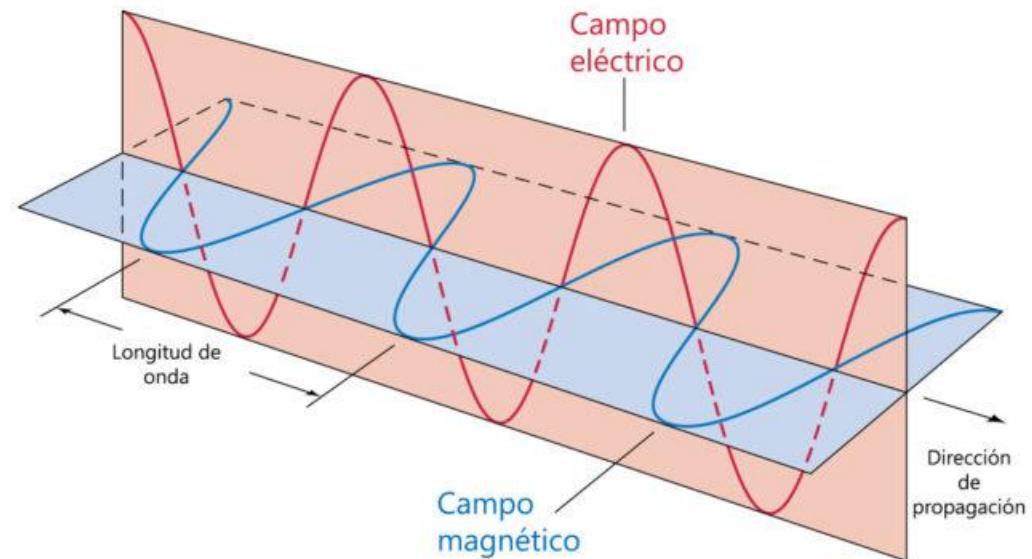
## Ejemplo: Capacidad máxima de un canal ruidoso C

- Espectro de un canal entre 1 MHz y 5 MHz
- $SNR_{dB} = +3 dB$  (la potencia de la señal es el doble de la potencia del ruido)
- $C = B \log_2(1 + SNR)$
- $SNR_{dB} = 10 \log_{10}(SNR)$
- $SNR = 10^{\frac{SNR_{dB}}{10}} = 10^{\frac{3}{10}} = 2$
- $B = 5 \text{ Mhz} - 1 \text{ MHz} = 4 \text{ MHz}$
- $C = 4 \times 10^6 \times \log_2(1 + 2) = 4 \times 10^6 \frac{\log_{10}(3)}{\log_{10}(2)} = 4 \times 10^6 \frac{0,48}{0,3}$
- $C = 6,36 Mbps$



## 2.4 Medios de transmisión guiados

- La información se propaga por **ondas electromagnéticas**
- Estas se pueden transmitir:
  - En forma confinada a través de un medio físico sólido,
  - a través del espacio abierto



# Características de un medio

- Ancho de banda
- Retardo
- Facilidad de instalación y mantenimiento
- Costo
- Vida útil

# Medios guiados

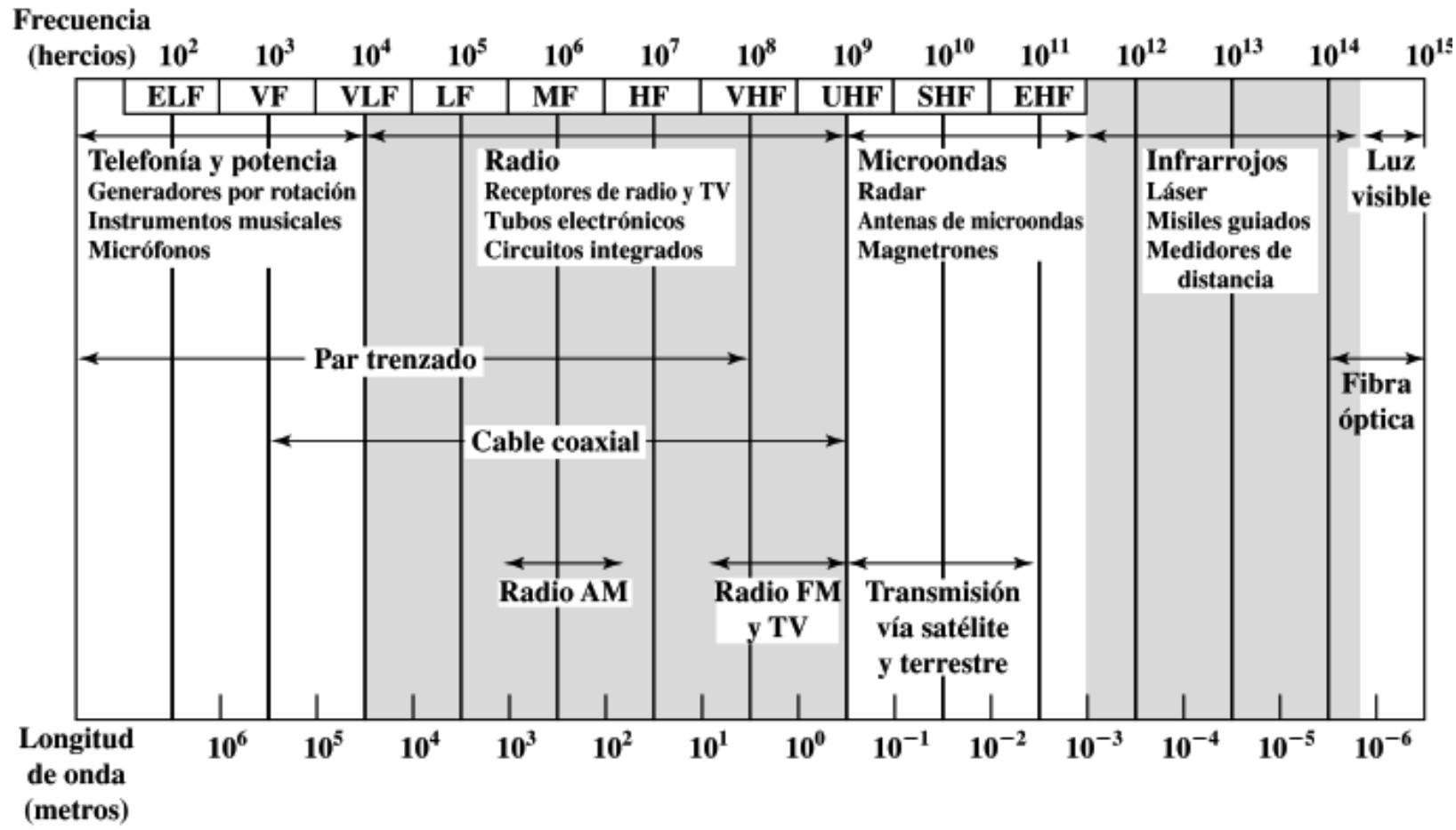
## ■ Cable de cobre

- Par trenzado

- UTP Unshielded Twisted Pair (no apantallado o blindado)
  - STP Shielded Twisted Pair

- Cable coaxial

## ■ Fibra óptica



AM 535-1605 kHz  
FM 88-108 MHz

ELF (*Extremely low frequency*) = Frecuencia extremadamente baja  
VF (*Voice frequency*) = Frecuencia vocal  
VLF (*Very low frequency*) = Frecuencia muy baja  
LF (*Low frequency*) = Frecuencia baja  
MF (*Medium frequency*) = Frecuencia media

HF (*High frequency*) = Frecuencia alta  
VHF (*Very high frequency*) = Frecuencia muy alta  
UHF (*Ultrahigh frequency*) = Frecuencia ultra alta  
SHF (*Superhigh frequency*) = Frecuencia super alta  
EHF (*Extremely high frequency*) = Frecuencia extremadamente alta

**Figura 4.1.** Espectro electromagnético para las telecomunicaciones.

## 2.4.1 Par trenzado

- Es uno de los más antiguos y más usados
- Se comenzó a utilizar a finales del S. XIX
- 2 alambres de cobre aislados de 1 mm de grosor
- Trenzados helicoidalmente para evitar que dos alambres paralelos se comporten como **antena**<sup>1</sup>
- Aplicación más común: sistema telefónico. Recorre varios km desde el hogar u oficina hasta la compañía telefónica

1 Antena: dispositivo capaz de recibir y emitir ondas electromagnéticas



(a)

(b)

- Usado para transmisión analógica y digital
- El **B** depende de:
  - tipo de trenzado
  - grosor del cable
  - distancia
  - apantallado
- Fácil instalación y mantenimiento
- Bajo costo

## ■ Tipos de cables de par trenzado:

- No blindado. **UTP** Unshielded Twisted Pair
- **FTP** Foiled Twisted Pair. Un blindaje con papel aluminio para todos los pares al mismo tiempo
- **SSTP** Shielded Screen Twisted Pair. Par trenzado apantallado blindado (IBM). Pantalla individual para cada par trenzado
- El conector RJ-45 para FTP y SSTP también es apantallado



**Table 4.3** Twisted Pair Categories and Classes

	<b>Category 3 Class C</b>	<b>Category 5 Class D</b>	<b>Category 5E</b>	<b>Category 6 Class E</b>	<b>Category 7 Class F</b>
<b>Bandwidth</b>	16 MHz	100 MHz	100 MHz	200 MHz	600 MHz
<b>Cable Type</b>	UTP	UTP/FTP	UTP/FTP	UTP/FTP	SSTP
<b>Link Cost (Cat 5 = 1)</b>	0.7	1	1.2	1.5	2.2

**UTP** = Unshielded twisted pair

**FTP** = Foil twisted pair

**SSTP** = Shielded screen twisted pair

# **UTP**



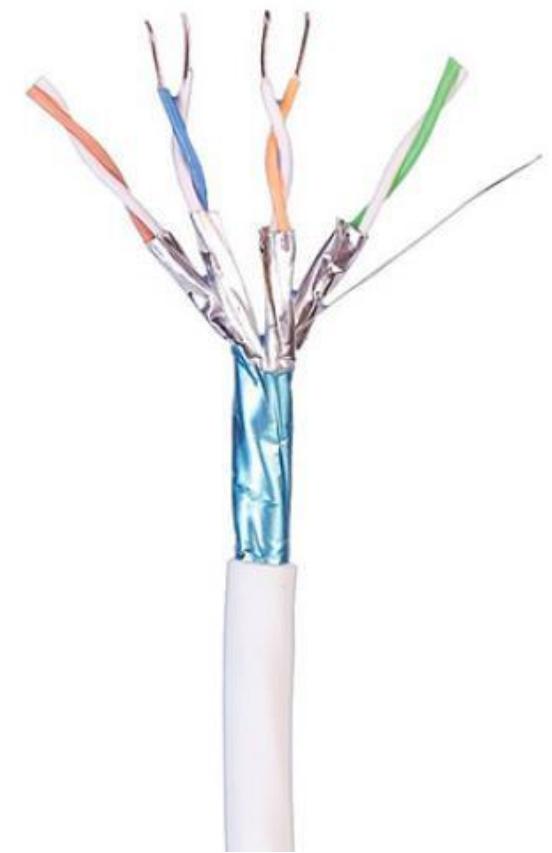
# **FTP**



# **STP**



# **SFTP**



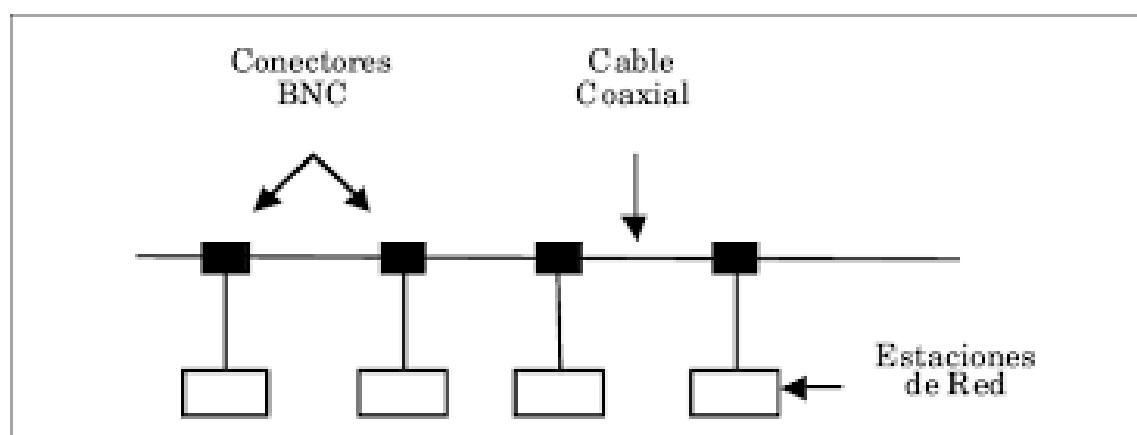
## ■ Fabricación de cable UTP ([video en Internet](#))

## 2.4.2 Cable coaxial

- Se comenzó a utilizar en la década de 1920
- Inicialmente utilizado para transmisiones de radio, telegrafía y telefonía
- Mejor blindaje
- Excelente inmunidad al ruido debido a su estructura
- Mayor **B.** Hasta 1 GHz
- Tramos más largos
- Mayores velocidades bps
- Para transmisión analógica y digital

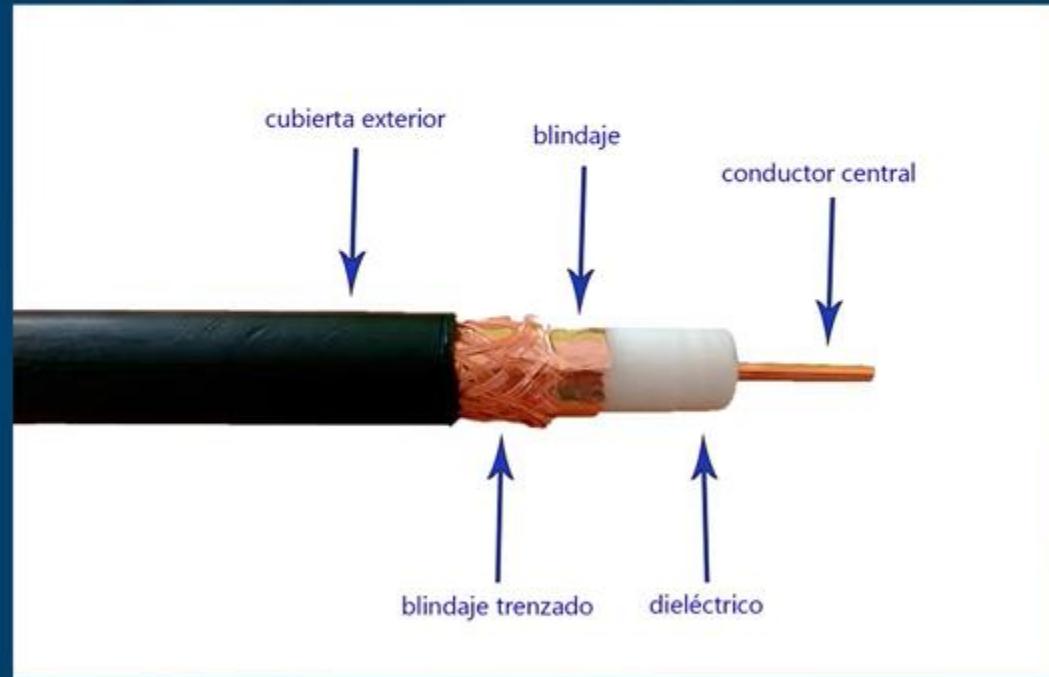
# Tipos de cable coaxial

- **50 ohmios:** transmisión digital
- **75 ohmios:**
  - transmisión analógica
  - TV cable
  - Internet por cable (cable módem)
  - MAN
- Fue muy usado en telefonía de larga distancia. Hoy se usa fibra óptica



# CABLE COAXIAL

consta de un núcleo de hilo de cobre rodeado por un aislante, un apantallamiento de metal trenzado y una cubierta externa.



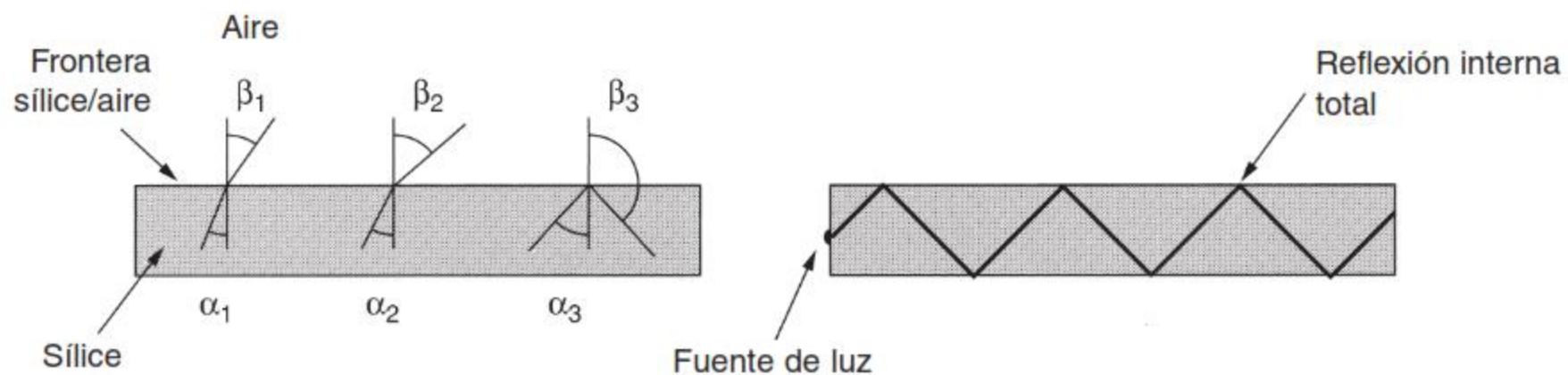
## 2.4.3 Fibra óptica

- C 50 Tbps
- Límite práctico de señalización actual es 10 Gbps (100 Gbps en laboratorio)
  - Debido a la incapacidad de **convertir** con más rapidez las señales eléctricas a ópticas y viceversa
- Fabricación ([video en internet](#))

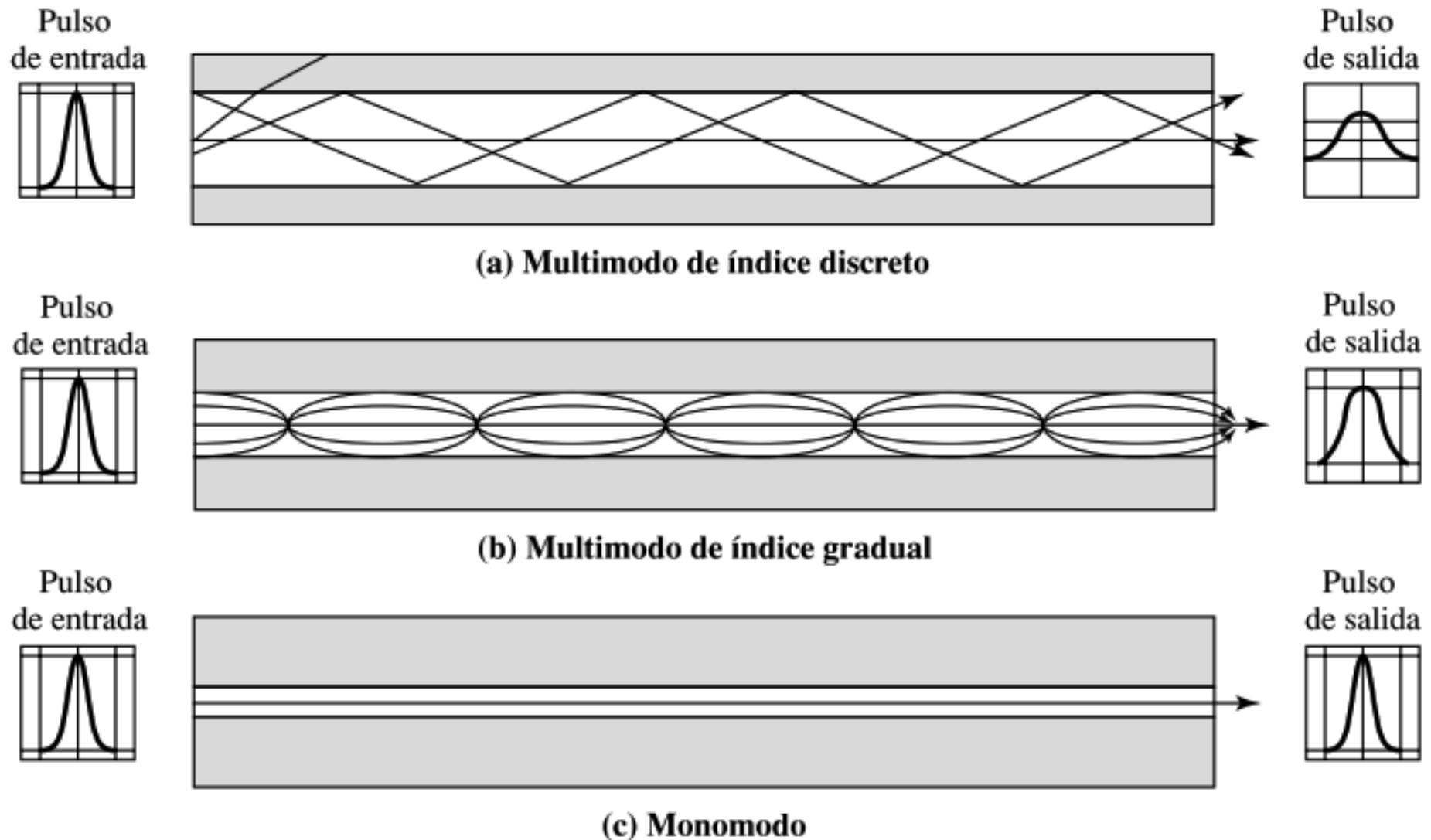
# Componentes del sistema de transmisión unidireccional

- **Medio de transmisión:** fibra de vidrio ultra delgada
- **Fuente de luz:** genera un pulso de luz al recibir una señal eléctrica
- **Detector de luz:** genera pulso eléctrico al recibir luz
  - Pulso de luz: bit 1
  - Ausencia de luz: bit 0

- El rayo de luz se propaga confinado dentro de la fibra por la ley física de la **refracción**
- **Refracción:** Cambio de dirección de un haz de luz al pasar de un medio físico a otro.



Índice de refracción de un medio  $n = c/v$   
 $c$ : velocidad de la luz en el vacío  
 $v$ : velocidad de la luz del medio



# Fibras multimodo

- Fibras con mayor grosor conducen más de un haz de luz
- Cada haz rebotará con ángulos diferentes o **modos** diferentes

# Fibra multimodo de índice gradual

- A menor densidad de la FO mayor la velocidad de la luz
- La densidad de la fibra **disminuye** conforme se aleja del eje axial
- La luz avanza más rápido conforme se aleja del eje
- La luz describe curvas helicoidales
- La luz periférica llega al receptor al mismo tiempo que la luz axial

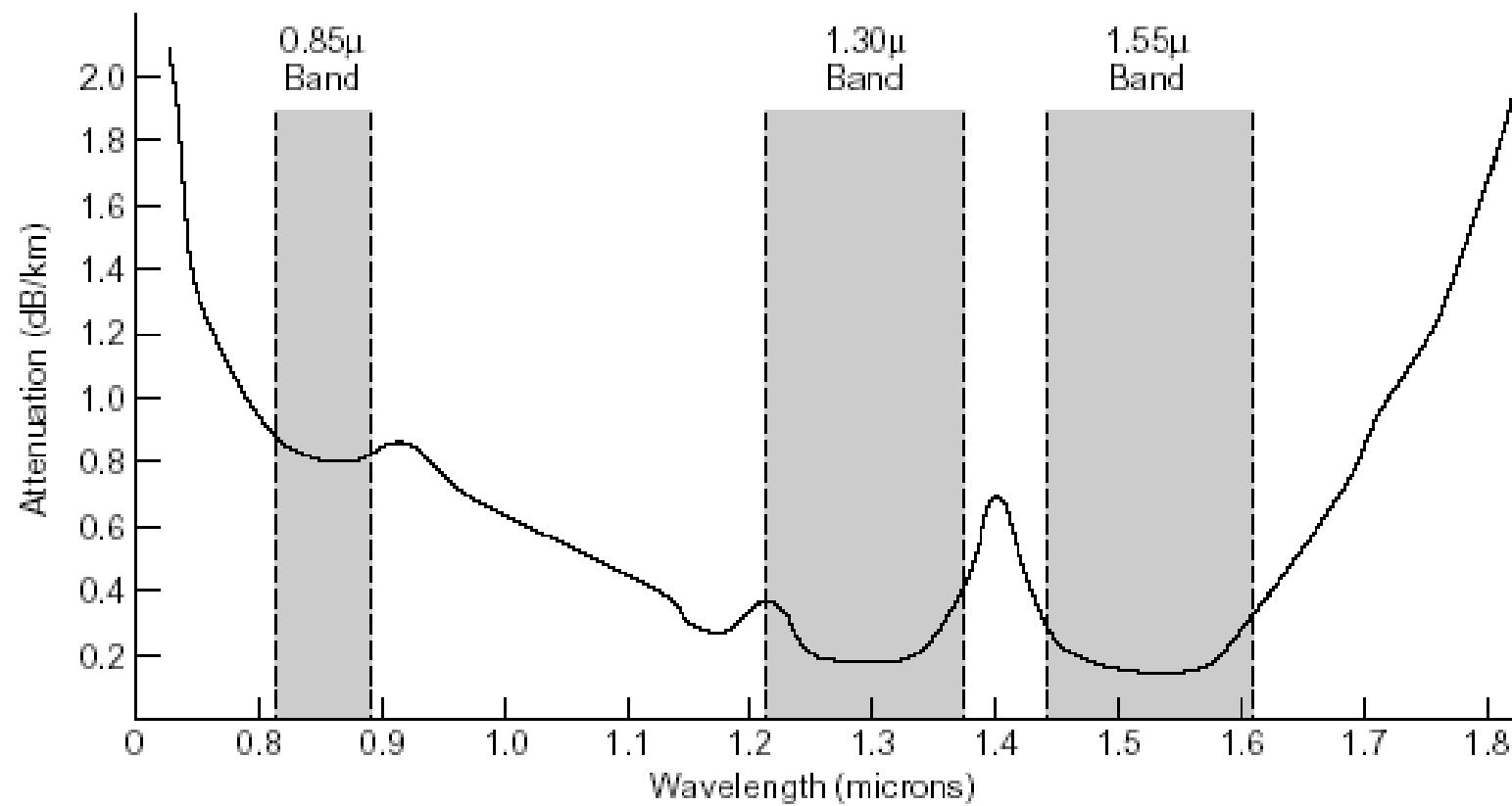
## Fibra monomodo

- **Guía de onda:** diámetro se reduce a unas cuantas  $\lambda$  (8,3 a 10  $\mu\text{m}$ )
- El haz se propaga sin reflejarse en los límites de la fibra
- Distancias desde 40 Km a 1000 Km y más, para transmisiones submarinas, sin necesidad de regeneración de la señal
- Hay varios tipos de fibra monomodo

- La fibra es transparente como el aire
- Se fabrica con **silicio**
- El silicio se encuentra en la arena y en las rocas de granito, material de muy bajo costo
- Hay en cantidades ilimitadas

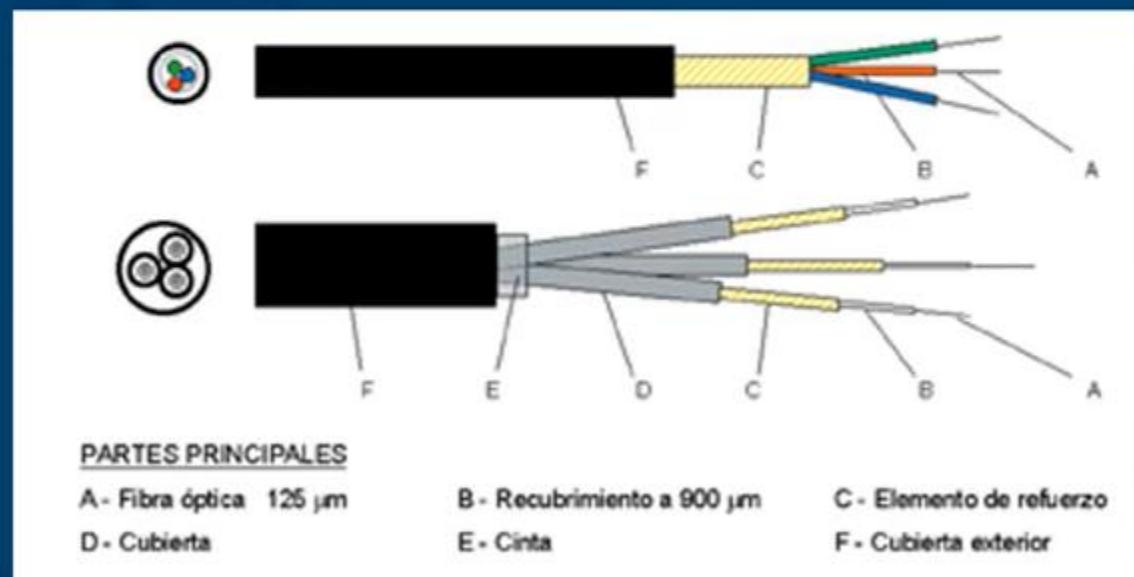
# Problemas en la propagación de la luz

- Atenuación de la luz depende de  $\lambda$  de onda y de la pureza del vidrio
- Luz visible:  $\lambda$  de 0.4 a 0.7  $\mu$
- Para las comunicaciones se utilizan tres bandas de longitud de onda



# CABLE DE FIBRA OPTICA

transportan señales digitales de datos en forma de pulsos modulados de luz. Es apropiado para transmitir datos a velocidades muy altas y con grandes capacidades.



## PARTES PRINCIPALES

A- Fibra óptica 125  $\mu\text{m}$

D- Cubierta

B- Recubrimiento a 900  $\mu\text{m}$

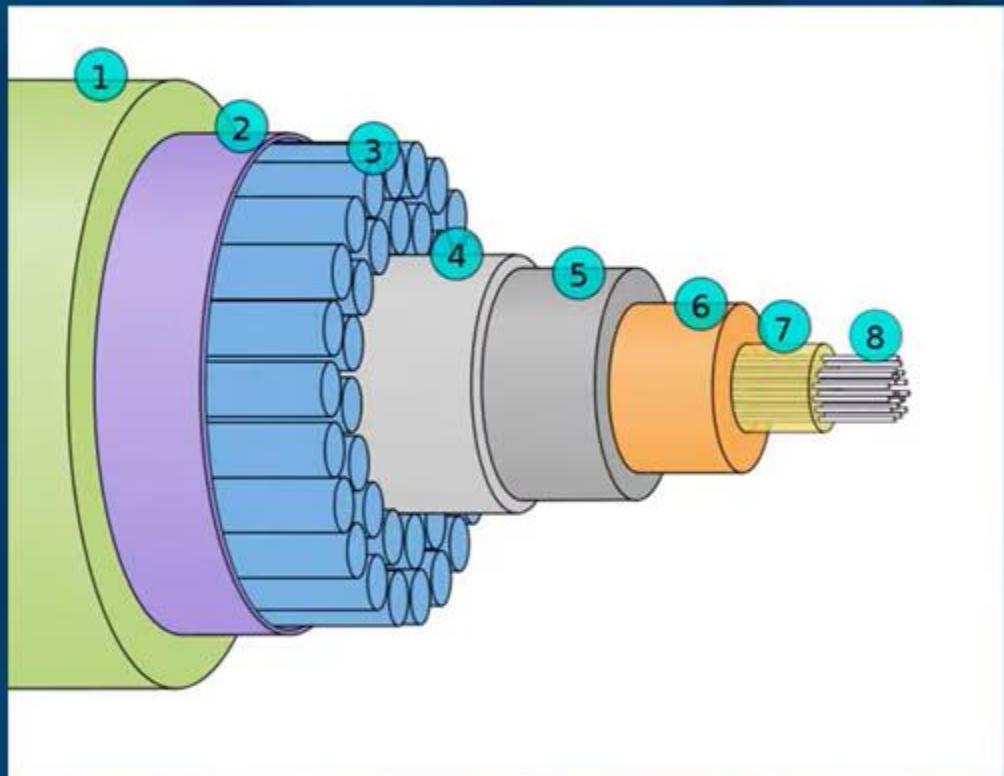
E- Cinta

C- Elemento de refuerzo

F- Cubierta exterior



## Cable submarino



- 1 - Polietileno
- 2 - Cinta tipo Mylar
- 3 - Cables Trenzados de acero
- 4 - Aluminio que sirve como barrera contra el agua
- 5 - Policarbonato
- 6 - Tubo de aluminio o cobre
- 7 - Un protector para el agua llamado Petroleum jelly (vaselina)
- 8 - La fibra óptica

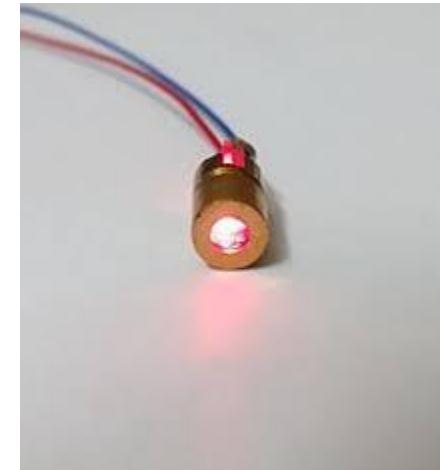


Fibra	Diámetro núcleo
multimodo	50 $\mu$ (un cabello)
monomodo	8 a 10 $\mu$ .

- El revestimiento de vidrio mantiene la luz dentro del núcleo
- Fibras transoceánicas van por el fondo del mar

# Características de transmisión

- FO actúa como guía de onda para frecuencias de luz de  $10^{14}$  a  $10^{15}$  Hz
- LED (Light Emitting Diode)
- Injection Laser Diode (ILD)
- Multiplexión por división de  $\lambda$  WDM



# Fuentes de luz

Elemento	LED	Láser semiconductor
Tasa de datos	Baja 1 Gbps	Alta 10 Gbps
Tipo de fibra	Multimodo	Multimodo o monomodo
Distancia	Corta 550 m	Larga 40 Km – 1000 Km
Tiempo de vida	Largo	Corto
Sensibilidad a la temperatura	Menor	Considerable
Costo	Bajo	Elevado

**Figura 2-8.** Comparación de diodos semiconductores y LEDs como fuentes de luz.

# Ventajas de la fibra óptica

- Mucho más rápido que la transmisión en cobre
- Hasta diez veces o más rápido que la transmisión satelital (100 Mbps)
- Baja atenuación. Repetidores cada 40 Km a 1000 Km
  - 5 Km. en el cobre
- Inmunidad contra:
  - Sobrecargas de energía
  - interferencia electromagnética
  - sustancias corrosivas
- Delgada y liviana. Importante para las PSTN

material	distancia	pares	Kg.
cobre	1 km	1000	8000
fibra	1 km	1	100

- Bajo costo de instalación
- No hay fugas de luz
- Difícil intervenirlas
- Seguridad contra intrusos y espías

# Aplicaciones de la Fibra Óptica

- Troncales de gran recorrido
- Última milla, bucle del abonado, bucle local, bucle de suscriptor, local loop, red de acceso
- LANs

# Debilidades de la fibra óptica

- Tecnología que recién se está haciendo familiar
- Requiere habilidades que muchos ingenieros carecen
- Muy delicada. Se puede doblar o romper
- Transmisión unidireccional. La comunicación duplex requiere dos fibras o dos bandas de frecuencia en una fibra
- Interfaces de fibra más caras que las eléctricas
- Equipos para instalación de FO más caros

Cómo fusionar dos fibras ópticas con la fusionadora de PROMAX modelo PROLITE-40B

## I Cortadora compacta de fibra AF-001

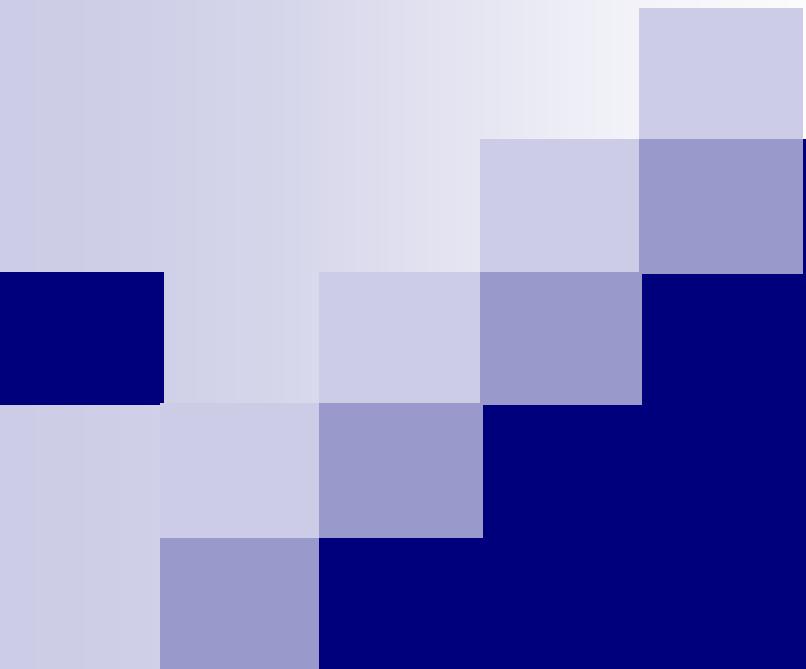
### I Incluida en los kits de conectorización y en los kits de fusión de fibra óptica



01:31

ES 17:21  
11/11/2019

137



## 2.5 Transmisión inalámbrica

- En el futuro solo habrían dos clases de comunicaciones:
  - Fibra óptica para los dispositivos fijos
  - Inalámbrica para dispositivos móviles
    - Laptops, tablets, celulares, GPSs, etiquetadoras de precios de artículos, lector de tarjetas de crédito y débito
    - Dispositivos fijos de difícil acceso: estaciones meteorológicas (sensores de caudal de agua, sísmicos, lluvia, temperatura, velocidad del viento)
- Las comunicaciones inalámbricas se iniciaron en Hawaii con la *Red Aloha 1970s* (Norman Abramson)



Estaciones remotas que transmite las lecturas inalámbricamente





## Estado de Hawaii

## 2.5.1 Espectro electromagnético

- Electrones en movimiento producen ondas electromagnéticas que viajan por el espacio
- Estas ondas fueron modeladas matemáticamente por el británico nacido en Edimburgo, James Maxwell en 1865
- Henrich Hertz, judío nacido en Hamburgo, las experimentó en 1887



*James Maxwell  
Hamburgo-Alemania  
1831-1879*



*Henrich Hertz  
Edimburgo - Escocia  
1857 - 1894*

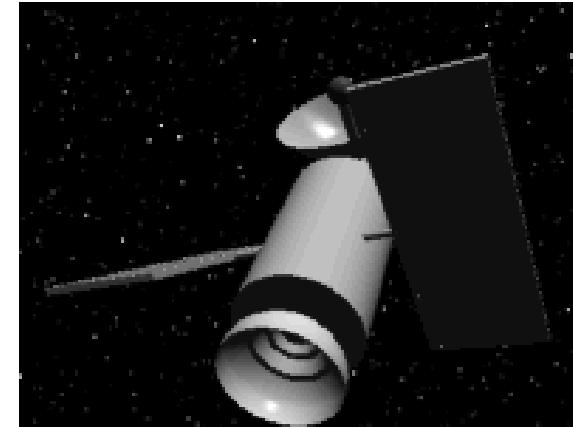
- Una antena puede difundir y recibir ondas electromagnéticas
- En el vacío las ondas viajan a la velocidad **c**, independientemente de **f**
- En el vacío:  $\lambda f = c$
- En cobre o fibra óptica, la velocidad es  $\frac{2}{3}c$  y es dependiente de **f**

# Antena

- Es uno o más conductores eléctricos usados para radiar y/o captar ondas electromagnéticas
- Una antena puede ser a la vez emisora y receptora
- Corriente eléctrica a frecuencia de radio (10KHz-100GHz) irradia energía en forma de ondas electromagnética a través de una antena emisora
- Una antena receptora capta estas ondas y las pasa al receptor para que las convierta en corriente eléctrica



Estación terrena



Satélite de comunicaciones

[Link](#)

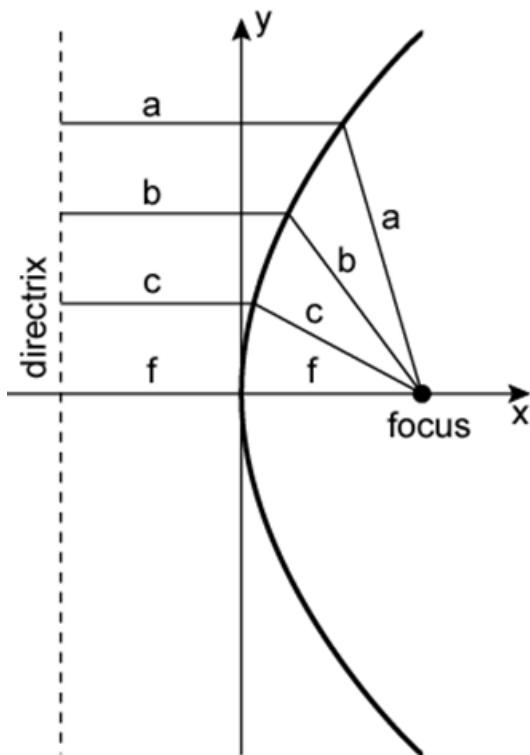
# Patrón de radiación

- Se irradia potencia en todas las direcciones
- No hay la misma emisión de potencia en todas las direcciones
- Antena Isotrópica (teóricamente) es un punto en el espacio que Irradia la misma potencia en todas las direcciones

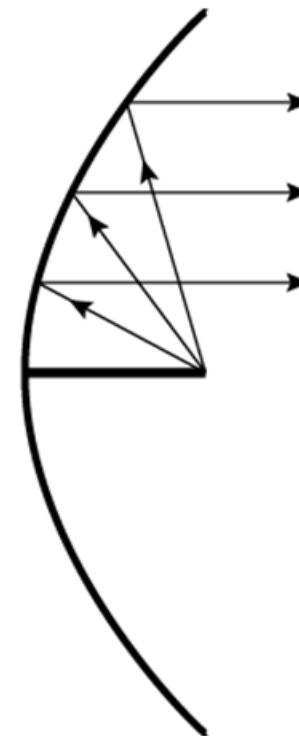
# Antenas parabólicas de reflexión

- Usadas para transmitir en el rango de frecuencias de radio en el rango de 1 GHz – 100 GHz
- Las ondas de radio que están en este rango de frecuencias se llaman **microonda**
- Usadas en transmisiones satelitales
- Parábola: lugar geométrico equidistante a una línea y un punto
  - El punto es el foco
  - La línea es la directriz

# Antena parabólica



(a) Parabola



(b) Cross-section of parabolic antenna  
showing reflective property

- La fuente colocada en el foco produce ondas reflejadas en la parábola paralelamente al eje
- Las ondas pueden ser: un haz de luz, sonido, calor u ondas de radio
- En recepción, la señal es concentrada en el foco, que se conecta el detector

# Spread Spectrum

- Inicialmente utilizado en aplicaciones militares
- Seguro contra la interceptación
- Usa banda ancha
- Hay dos tipos:
  - FHSS Frequency-Hopping Spread Spectrum
  - DSSS Direct Sequence Spread Spectrum

# FHSS

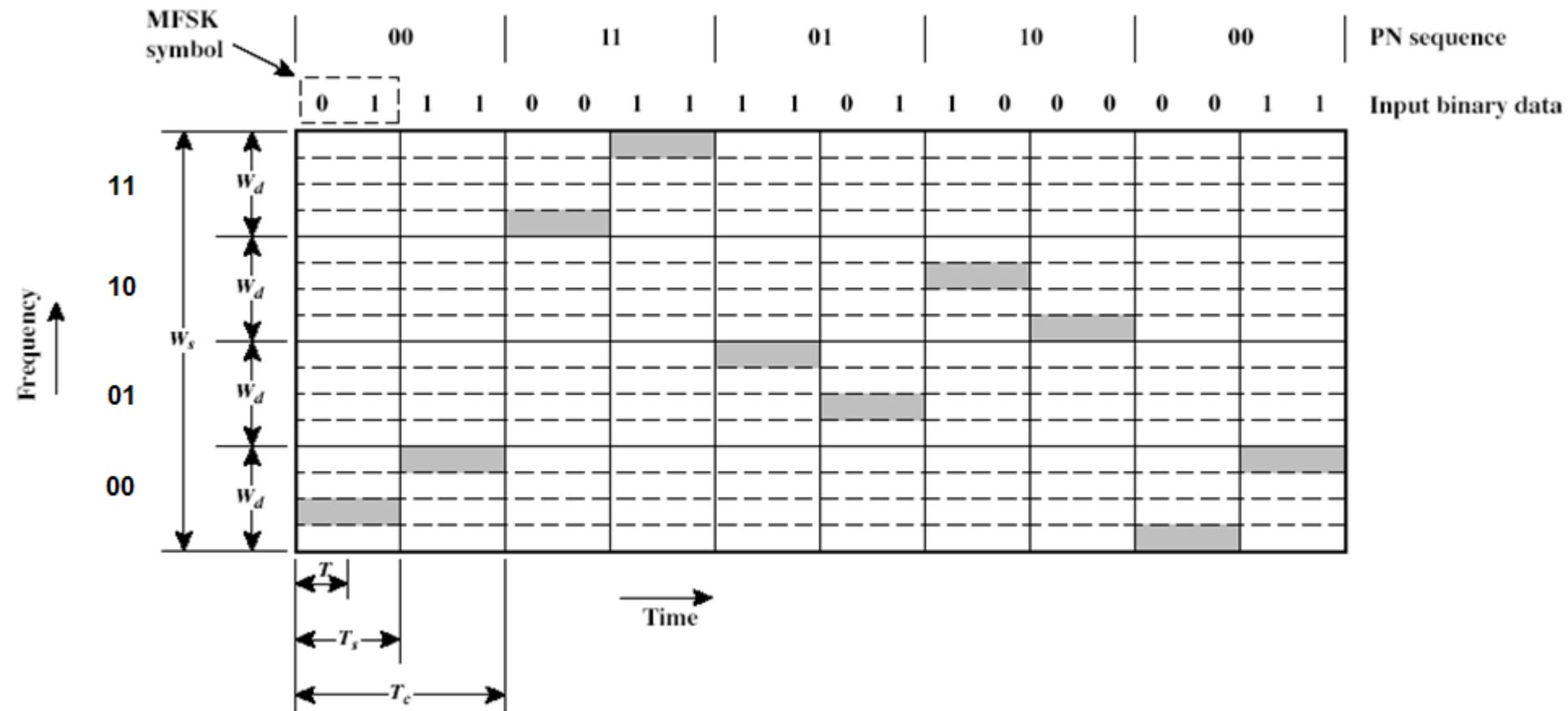
- Inventado por la actriz de Hollywood austriaca Hedy Lamarr en 1940 a la edad de 26 años
- Primera mujer que apareció desnuda en el film checoslovaco *Éxtasis* (1933)



Viena 1914 - 2000

- Comercialmente se usa en 802.11 y Bluethooth
- Resistente al efecto de la propagación por múltiples caminos
  - Las señales reflejadas llegan después de que el receptor ha saltado a otra frecuencia y ya no acepta señales de la frecuencia anterior

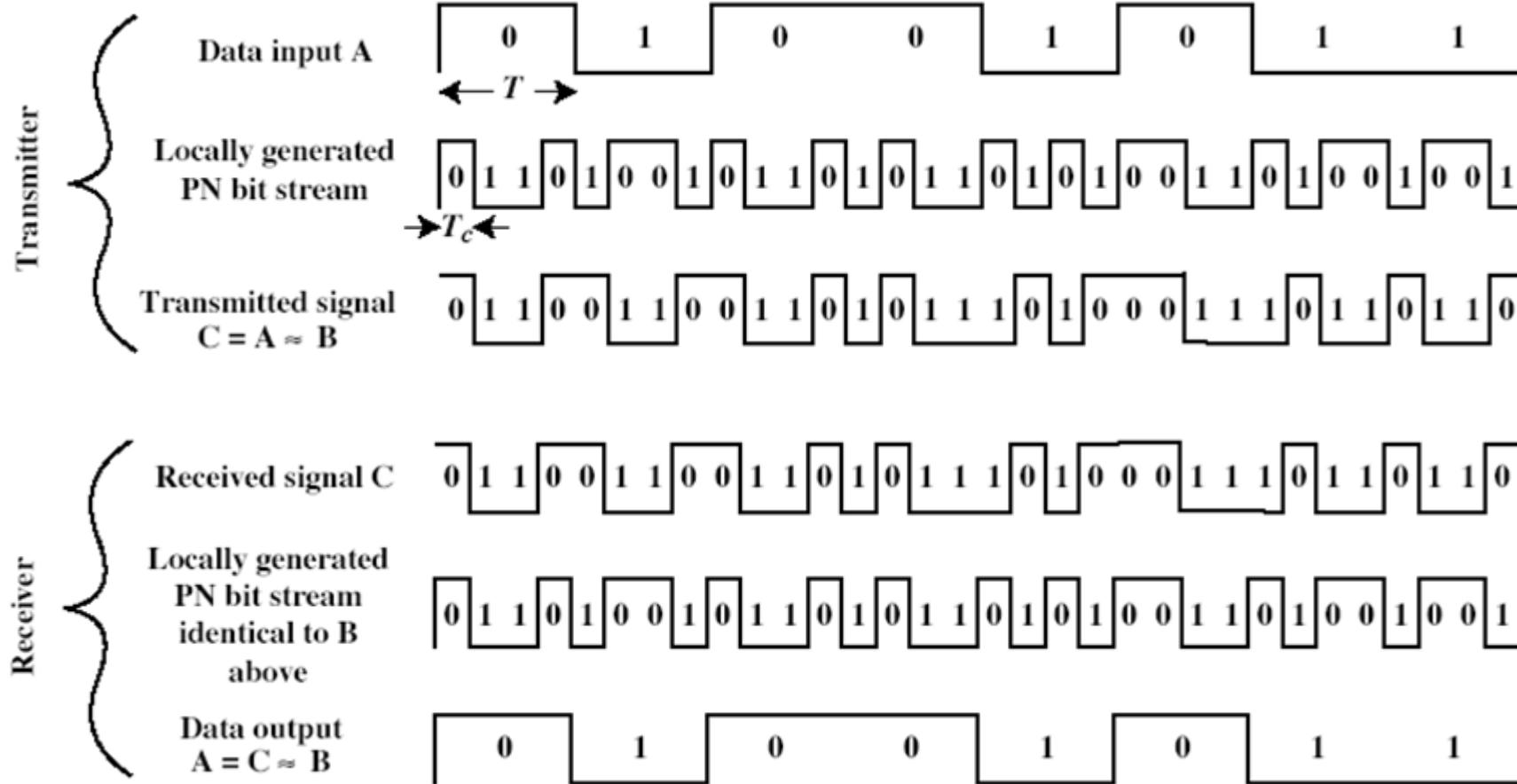
# FHSS Frequency-Hopping Spread Spectrum



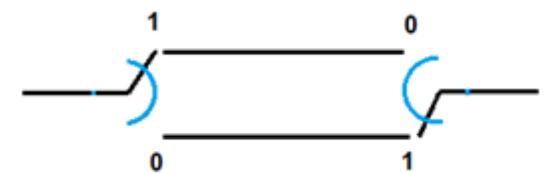
# DSSS Direct Sequence Spread Spectrum

- Lo usa la telefonía celular y algunas LANs inalámbricas

# DSSS



a	b	$a \text{ XOR } b$
0	0	0
0	1	1
1	0	1
1	1	0



# **Alan Turing**

## Interceptación de mensajes alemanes



[video](#)

# Medios inalámbricos

- Ondas de radio de baja frecuencia  
AM, FM, TV, 10 KHz – 1 GHz
- Ondas de radio de alta frecuencia  
Microondas 1 GHz – 100 GHz
- Infrarrojo 1 THz
- Ondas de luz 100 THz – 1000 THz

## 2.5.2 Ondas de radio de baja frecuencia

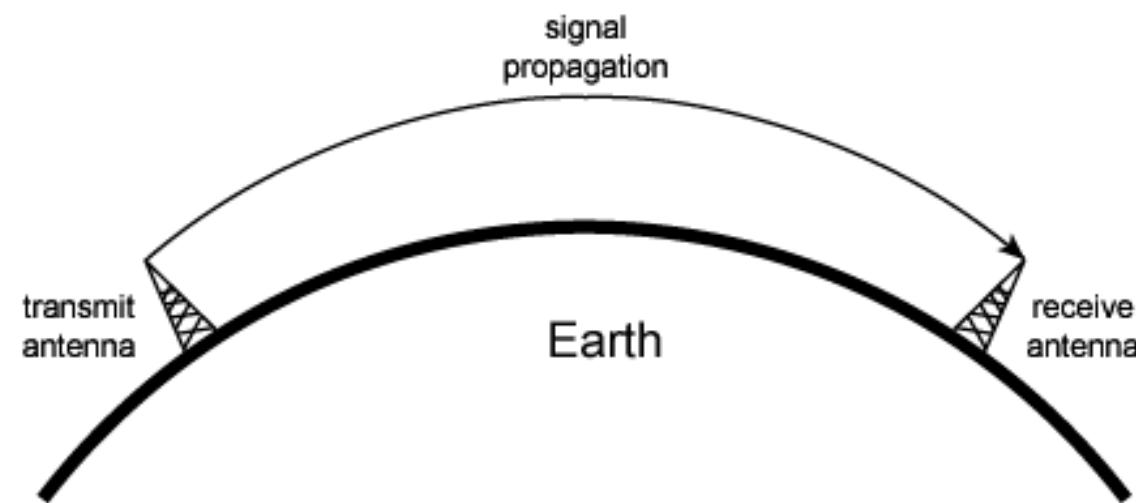
- 10 KHz – 1 GHz
- Relativamente fáciles de generar
- Viajan a grandes distancias
- Omnidireccionales
- No es necesario ***línea de vista*** entre transmisor y receptor

# Propiedades de las ondas de radio

- A **bajas** frecuencias las ondas penetran cuerpos opacos
- A frecuencias **altas**:
  - las ondas viajan en línea recta
  - rebotan en los obstáculos
  - son absorbidas por la lluvia
- En todas las frecuencias, las ondas son interferidas por el ruido de motores

## Baja frecuencia. 10 KHz – 1 GHz

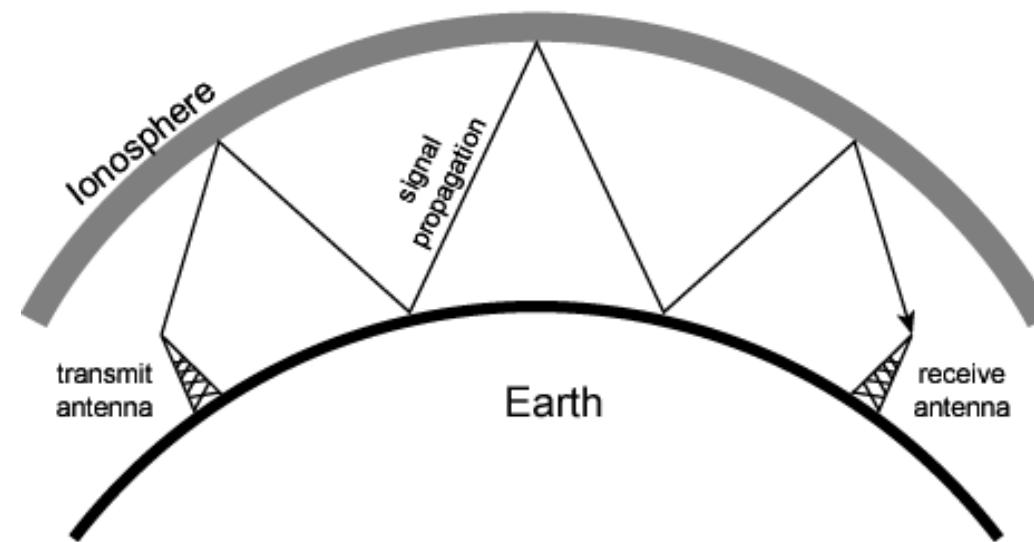
- Debajo de 2 MHz: VLF y MF
- Propagación superficial de onda
- Ondas de radio de baja frecuencia: Radio AM 550 a 1705 KHz
- Se pueden detectar a 1000 km



(a) Ground-wave propagation (below 2 MHz)

## Baja frecuencia. 10 KHz – 1 GHz

- 2 MHz a 30 MHZ: HF y VHF
- Propagación aérea
- Ondas de radio de baja frecuencia: FM: 88 a 108 MHz
- Se pueden detectar a 500 km



(b) Sky-wave propagation (2 to 30 MHz)

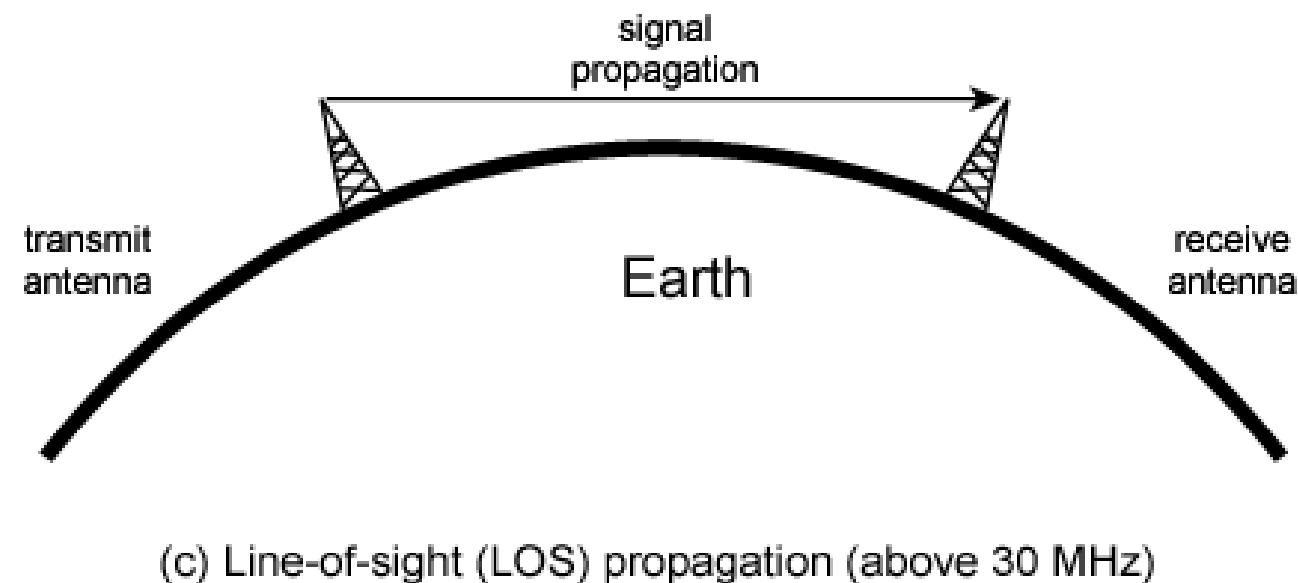
# Ionósfera

- Capa de la atmósfera que se extiende entre los 80 y los 500 km de altitud
- Es una capa ionizada (electrones libres)
- En esta se produce la aurora boreal



## Baja frecuencia. 10 KHz – 1 GHz

- Sobre los 30 MHz
- Las ondas no se reflejan en la Ionósfera
- Propagación en trayectoria de línea de vista





SW Shortwave frecuencias superiores a AM (1700 KHz – 30 MHz)  
Emisoras de radio de alcance internacional

## 2.5.3 Transmisión por **microondas**. 1 GHz a 100 GHz

- Las ondas viajan en línea recta
- No atraviesan muy bien edificios
- Necesitan línea de vista
- Se enfoca un haz estrecho con antena parabólica
- Telefonía celular: 850 MHz – 1.900 MHz



- Se necesitan **repetidores** periódicos
- Mientras más altas las torres más separadas pueden estar
- Torres de 100 m de altura pueden estar 80 Km separadas
- A 4 GHz las ondas son absorbidas por el agua
  - Cualidad aprovechada por hornos de microondas
  - Para las comunicaciones es un problema

- Bandas licenciadas. Los gobiernos asignan espectros para:
  - La radio AM y FM
  - Televisión
  - Teléfonos celulares
  - Policía
  - Marina
  - Navegación
  - Milicia
  - Entidades de gobierno, etc.

# Organismos

- A nivel mundial, ITU-R coordina la asignación de frecuencias, para la fabricación de dispositivos de comunicación
- En USA, FCC (Federal Communications Commission)
- En Ecuador es la Arcotel Agencia de Regulación y Control de las Telecomunicaciones

# Concesión del espectro

- Las portadoras -telefónicas- **concursan** para ganar una concesión
  - Los funcionarios deciden a quien dar. Se presta para sobornos
- Otra forma es:
  - **sortear** entre las empresas
  - **subastar** el espectro al mejor postor

# Bandas ISM Industrial, Scientific and Medical

- Otra forma de asignar frecuencias es **no asignar por completo**
- Política aplicada con las bandas ISM
- Son bandas no licenciadas
- Se deja que todos transmitan a voluntad
- Se regula la potencia tal que las estaciones no se interfieran entre sí

- Estas bandas no requieren autorización
- Aplicaciones:
  - Redes LAN 802.11
  - Teléfonos inalámbricos
  - Mouses inalámbricos, etc.
  - Dispositivos inalámbricos para abrir puertas
  - Juguetes a control remoto
- Se minimiza interferencias con técnicas de espectro expandido: FHSS, DSSS

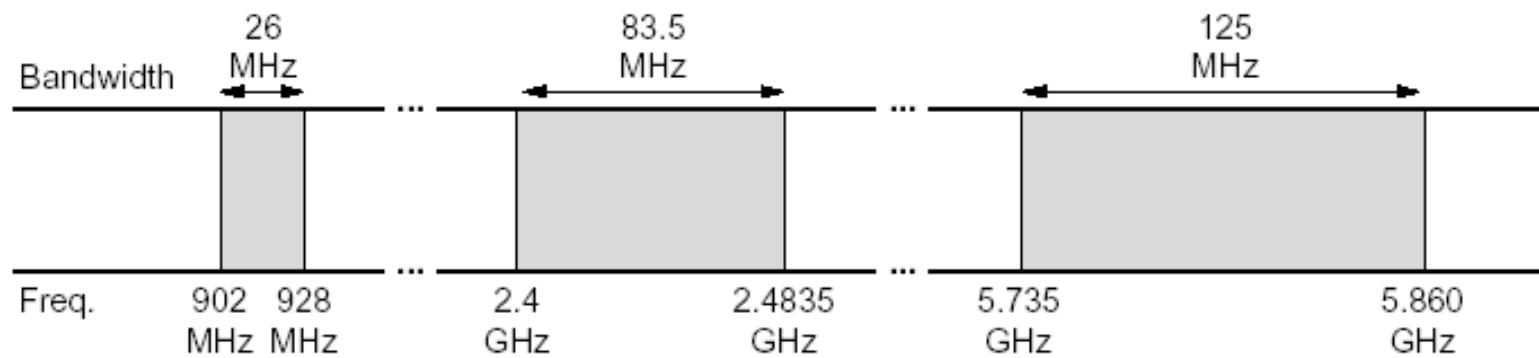


Fig. 2-13. The ISM bands in the United States.

**902 - 928 MHz** Está saturado

**2.4 GHz** LAN 802.3, Bluetooth

**5.7 GHz** No explotada, equipos caros

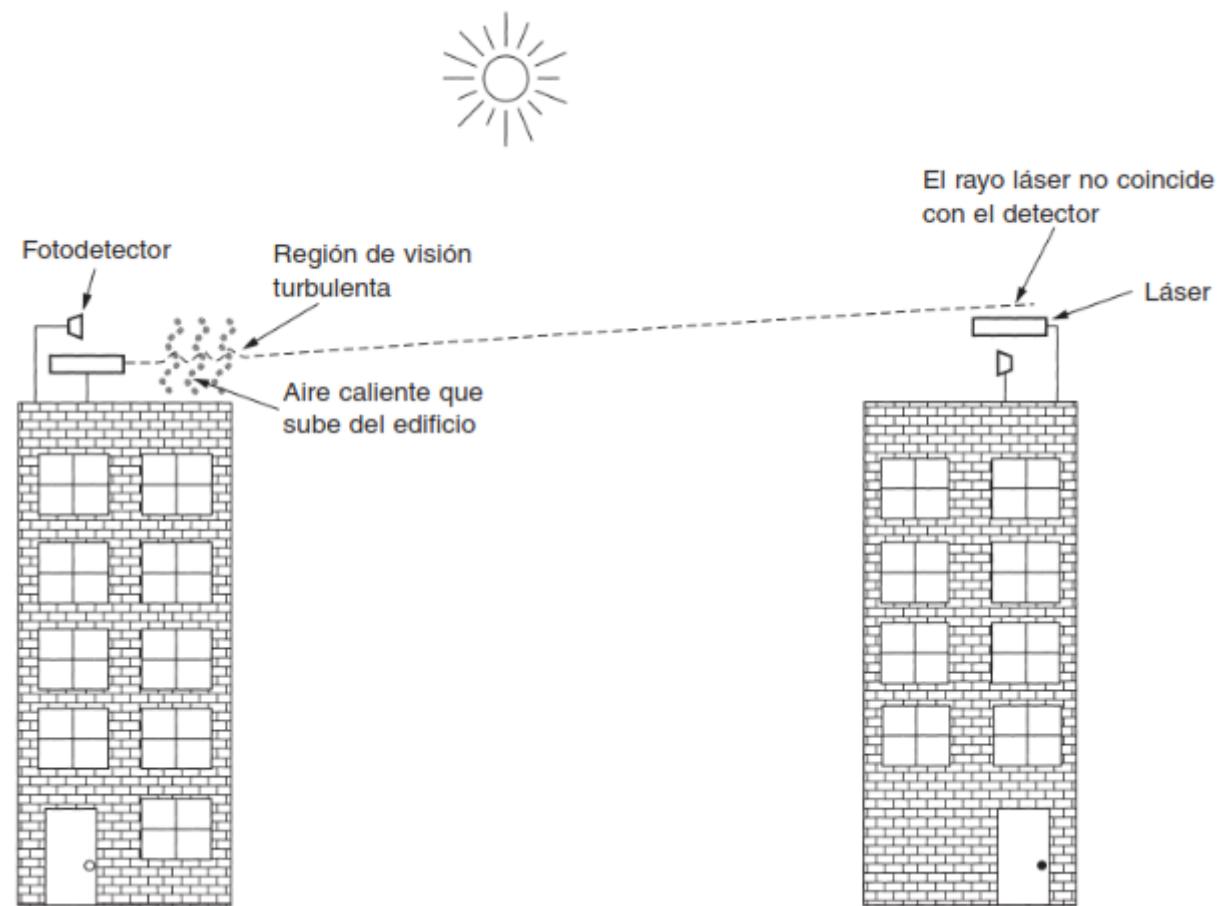
## 2.5.4 Ondas infrarrojas

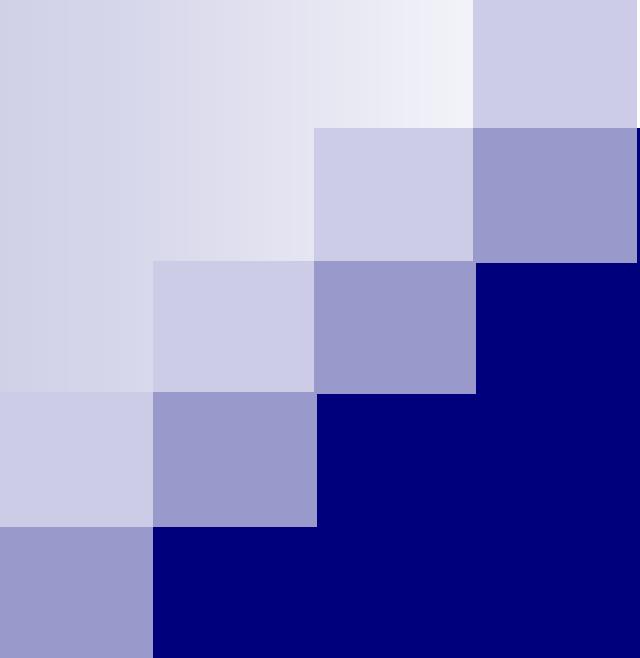
- Para comunicación de corto alcance
- WLANs. Casi y no se usan
- Control remoto de electrodomésticos
- Económicos y direccionales
- No atraviesan objetos sólidos
- No hay necesidad de licencia del gobierno
- Uso limitado

## 2.5.5 Transmisión por ondas de luz

- LANs de dos edificios pueden conectarse por medio de láser
- Sistema de comunicación **simplex** unidireccional
- Gran ancho de banda
- Costo muy bajo
- Relativamente fácil de instalar
- No requiere licencia

- Haces muy estrechos
- Apuntar un rayo láser de  $\emptyset = 1mm$ . a un blanco del tamaño de la punta de un alfiler a 2 Km de distancia requiere de puntería
- Se requiere de una mira utilizadas en los rifles
- No penetran la lluvia ni la niebla densa
- Funciona bien en días soleados





## 2.6 Satélites de comunicaciones

- En los años 50s, se desarrolló un sistemas de comunicación por rebote de señales en **globos aerostáticos meteorológicos**
- La marina de los U.S. descubrió una especie de globo climático en el cielo, la Luna
- Desarrolló un sistema de comunicación que rebotaba las señales en ella

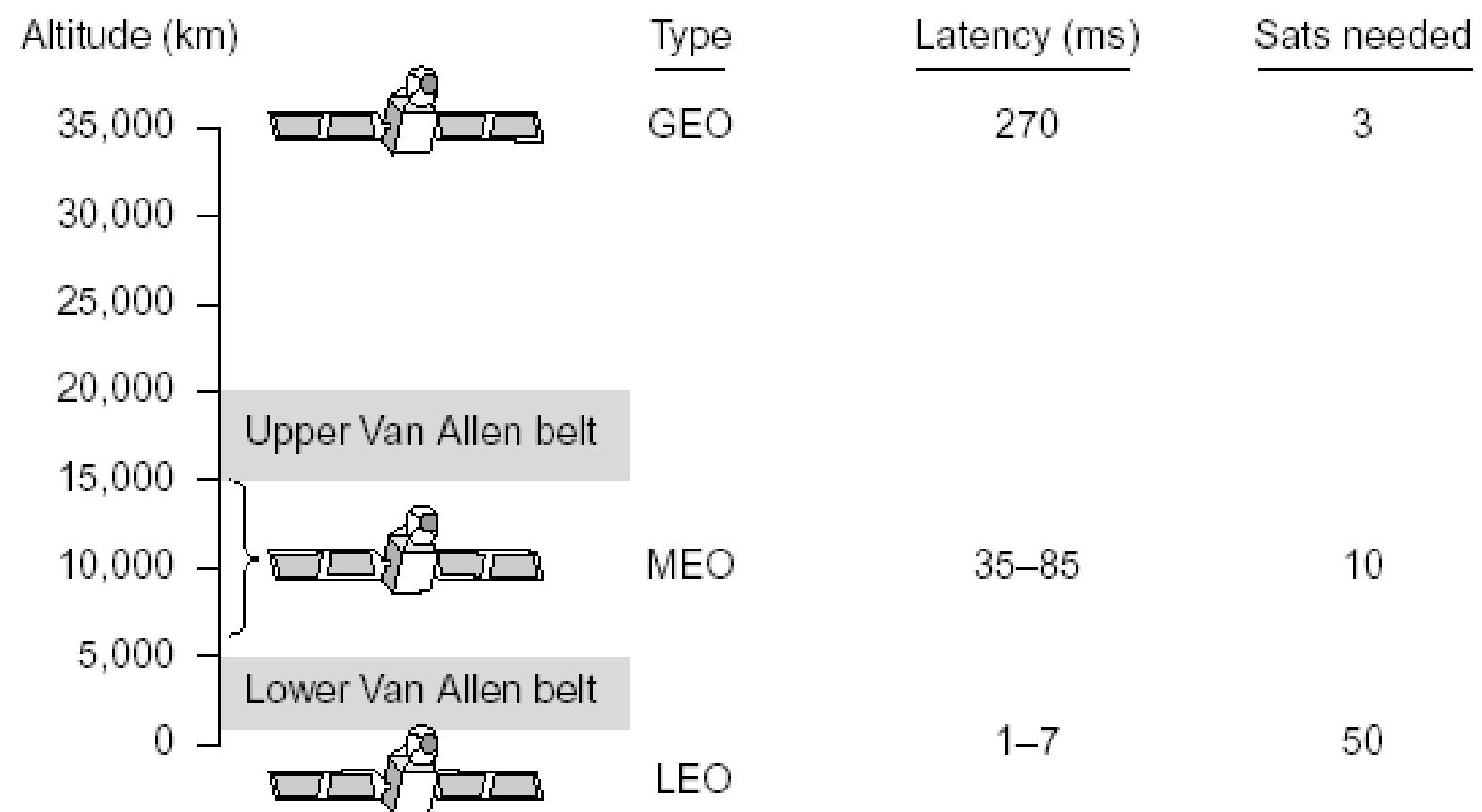


- El primer satélite artificial fue el Sputnik - спутник (satélite), URSS en 1957
- **Amplifica** la señal antes de reenviarla
- Un satélite artificial es un repetidor de microondas
- Tiene varios **transponders**: transmitter-responder
- Reciben la señal en una frecuencia, la amplifican, la retransmiten en otra f
- Cada transponder se encarga de una parte del espectro
- Los haces de retorno cubren una superficie grande de la Tierra: *footprint*



Johannes Kepler  
Alemania 1571-1630

- *Fuerza centrífuga = Fuerza gravitacional*
- $ma_c = G \frac{Mm}{r^2}$
- $m \frac{v^2}{r} = G \frac{Mm}{r^2}; v = \frac{2\pi r}{T}$
- Kepler: El período orbital del satélite  $T = 2\pi \sqrt{\frac{r^3}{GM}}$  segundos
- $r = R + h$ : radio de la órbita expresada en metros
- Radio de la Tierra  $R = 6370 \text{ Km}$
- Altura del satélite snm  $h = 35800 \text{ Km}$
- Constante gravitacional  $G = 6.674 \times 10^{-11} m^3 Kg^{-1} s^{-2}$
- Masa de la tierra  $M = 5.92 \times 10^{24} Kg$
- A mayor altura mayor período
- A 35.800 Km snm,  $T = 24$  : órbita geoestacionaria de Clarke
- A 384.000 Km  $T = 1 \text{ mes}$ , caso de la Luna



## 2.6.1 Satélites geoestacionarios

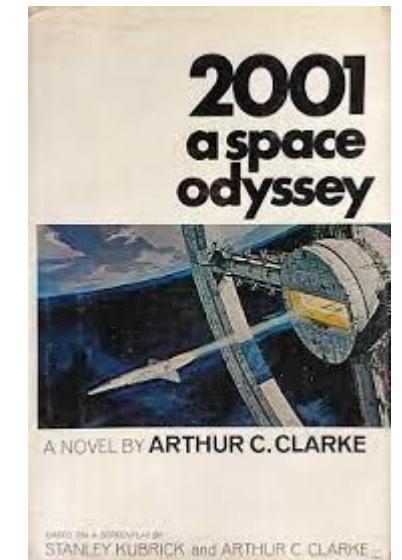
- Ideado en 1945 por el escritor de ciencia ficción Arthur C. Clarke
- Orbitan a una altura de 35.800 Km
- Ubicados en la zona ecuatorial
- No es necesario rastrearlos



## *Arthur C. Clarke*

Gran Bretaña 1917 – 2008

Novela: 2001: Una odisea espacial (1968)



*“La única manera de descubrir los límites  
de lo posible es aventurarse hacia lo  
imposible”*

# Albert Einstein

1879 – 1955

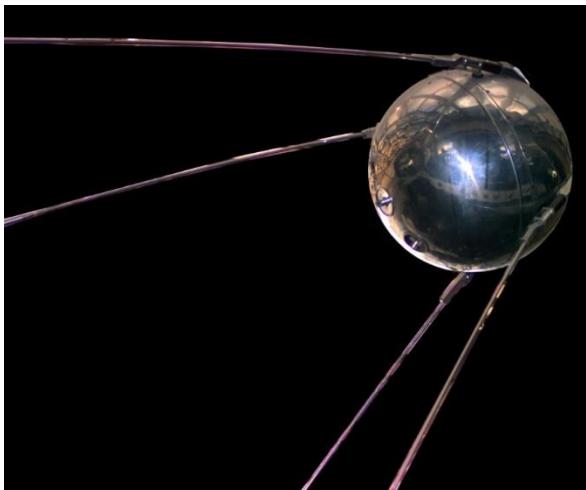
*“La imaginación es  
más importante que  
las matemáticas”*



- Clarke, luego de analizar el tema, concluyó que era muy difícil poner en órbita a una satélite porque los amplificadores de tubos catódicos son frágiles, consumen mucha energía y son muy pesados y voluminosos
- La invención del transistor (Laboratorios Bell) en 1948 lo hizo posible
- Un satélite pesa 4000 Kg. y consumen varios kw de potencia







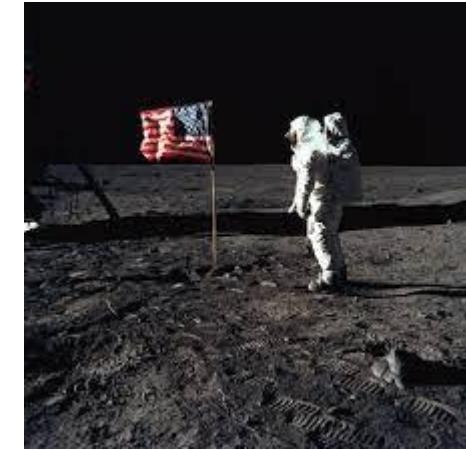
Primer satélite Sputnik.  
URSS 1957



Yuri Gagarin, dio la vuelta al mundo el 12 de abril de 1961



Primer satélite de comunicaciones  
Telstar. NASA 1962

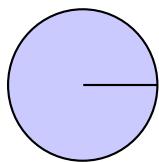


16 de julio de 1969

- ITU asigna a los países posiciones orbitales de cada satélite – geopolítica
- La separación entre satélites GEO es de  $2^{\circ}$  en el plano ecuatorial
- ITU también asigna bandas de frecuencia a las que operan los satélites
- Un satélite puede tener varios transponders
- La gravedad del Sol y la Luna desplaza a los satélites de sus órbitas. Se contrarresta con motores integrados al satélite
- El haz de los satélites geoestacionarios cubre la 1/3 de la superficie de la Tierra (huella o footprint)
- Un satélite puede emitir varios haces que abarquen pequeñas regiones



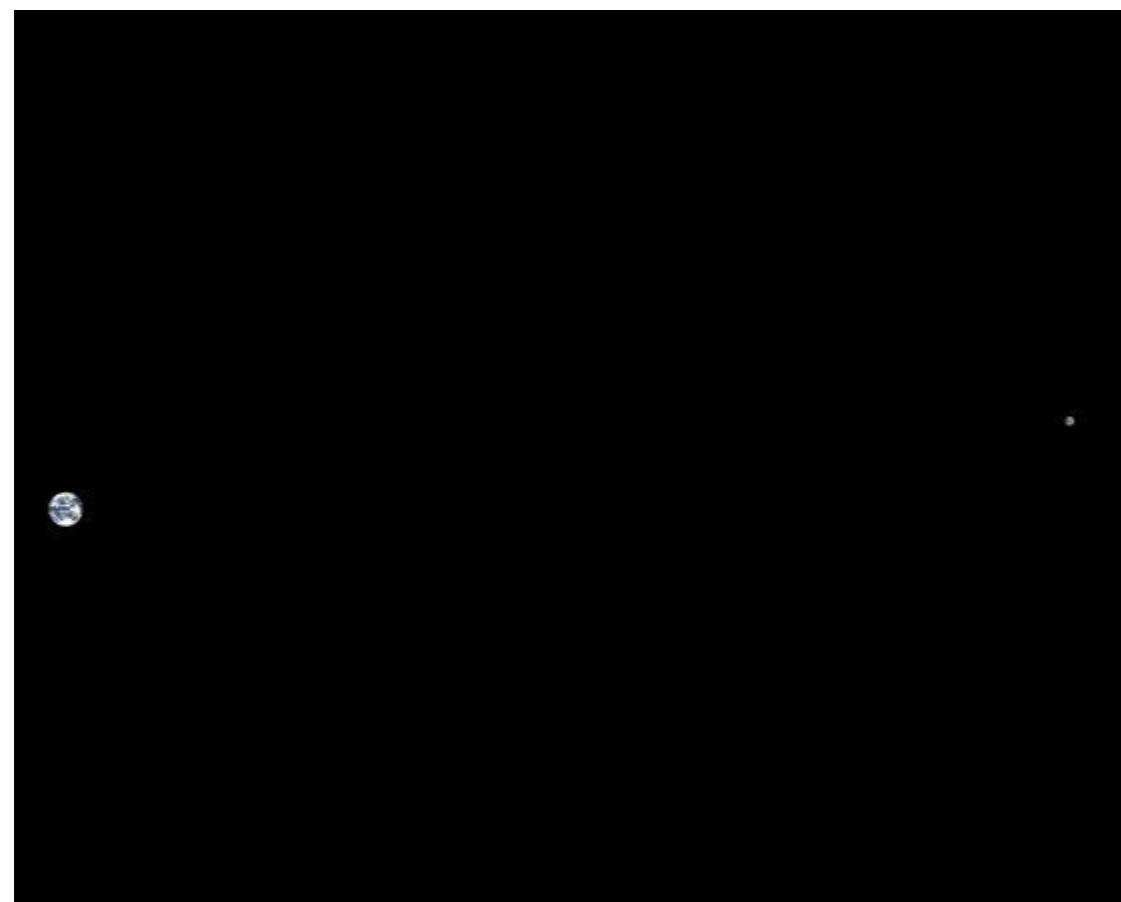
6674 km



35800 km

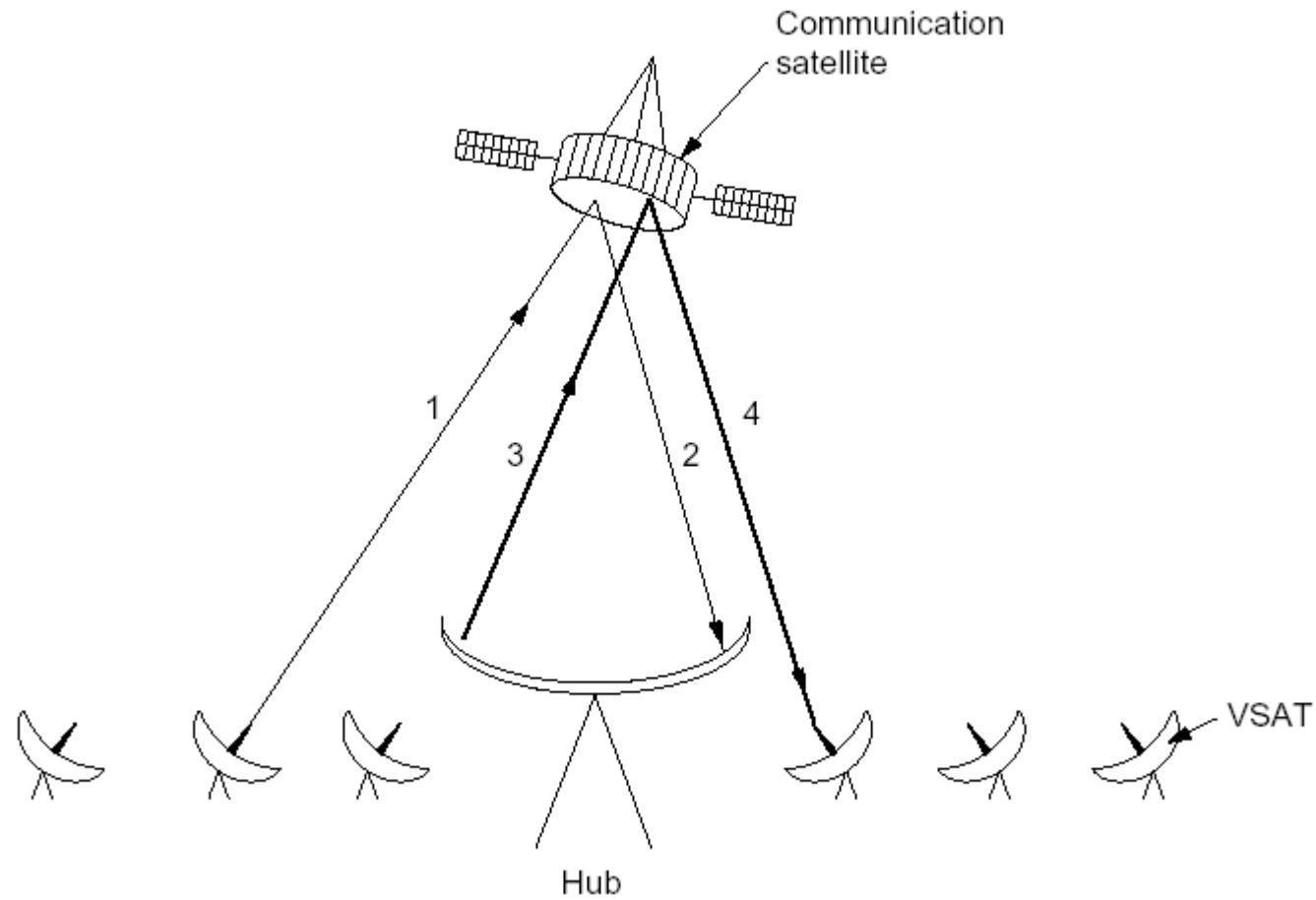


# Distancia de la Luna a la Tierra



# Antenas satelitales VSATs

- Usan satélites GEO
- Son micro estaciones de bajo costo y bajo consumo
- Consumen **1 watio** de potencia
- El tiempo de propagación de la señal desde la antena al satélite es 270 ms
- El diámetro de una VSAT terminal es 0.76 m
- La antena de una estación terrena mide 10 m de Ø
- Estas antenas son usadas principalmente en la televisión satelital



## ISP o Proveedor de tv satelital

# Antena parabólica VSAT



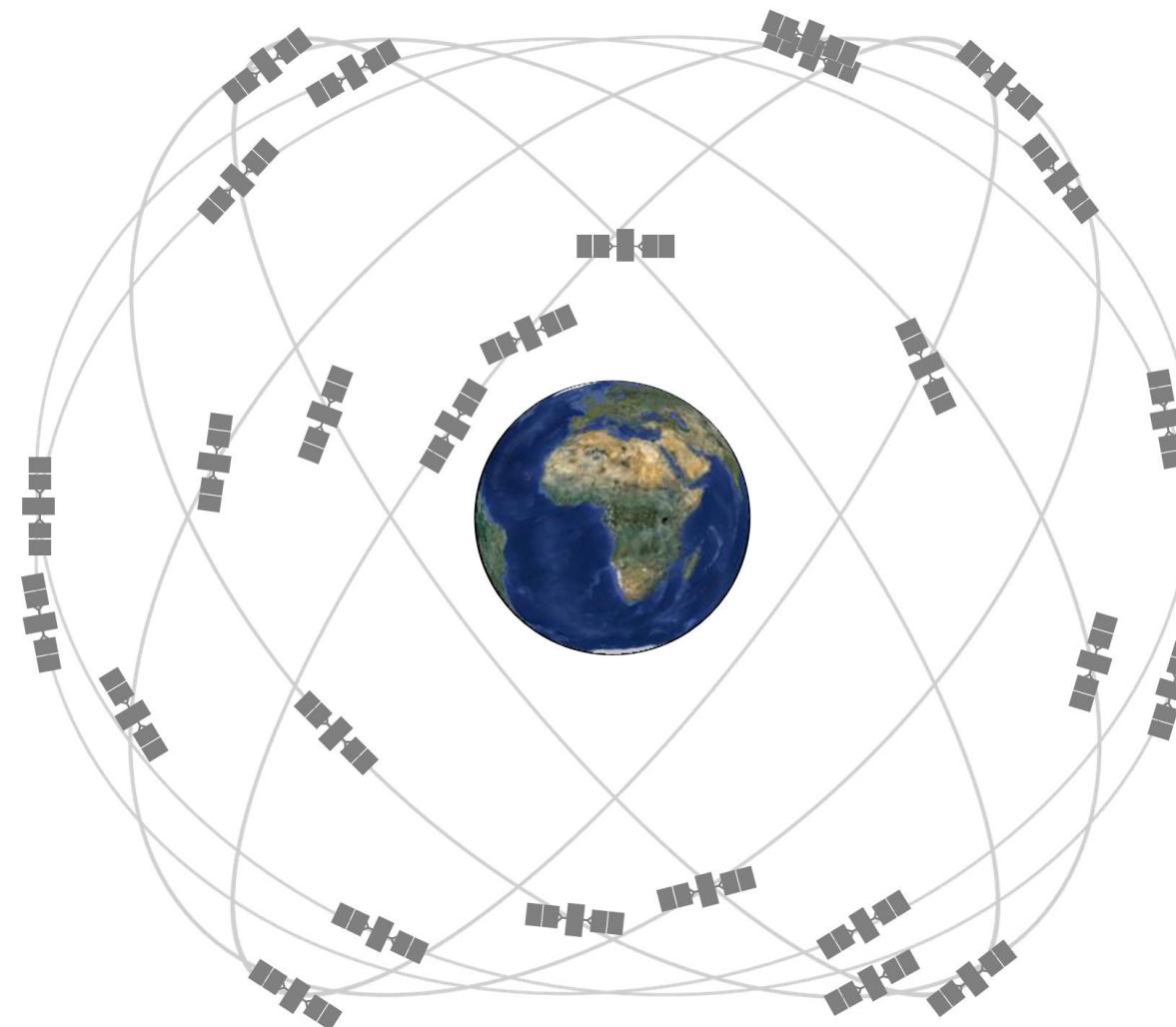


- Las VSATs se aplican en las zonas rurales
- Antenas VSATs terminales se alimentan con paneles fotovoltaicos
- El Internet hoy es un servicio básico, como el agua potable y el alcantarillado
- El 51% de la población mundial tiene acceso a Internet
- Más de la mitad de la población del mundo vive a una hora de distancia del teléfono más cercano
- El 80% de la población mundial tiene acceso al agua potable. 1500 millones de personas no tienen agua potable
- El 55% de la población mundial tienen acceso al alcantarillado sanitario

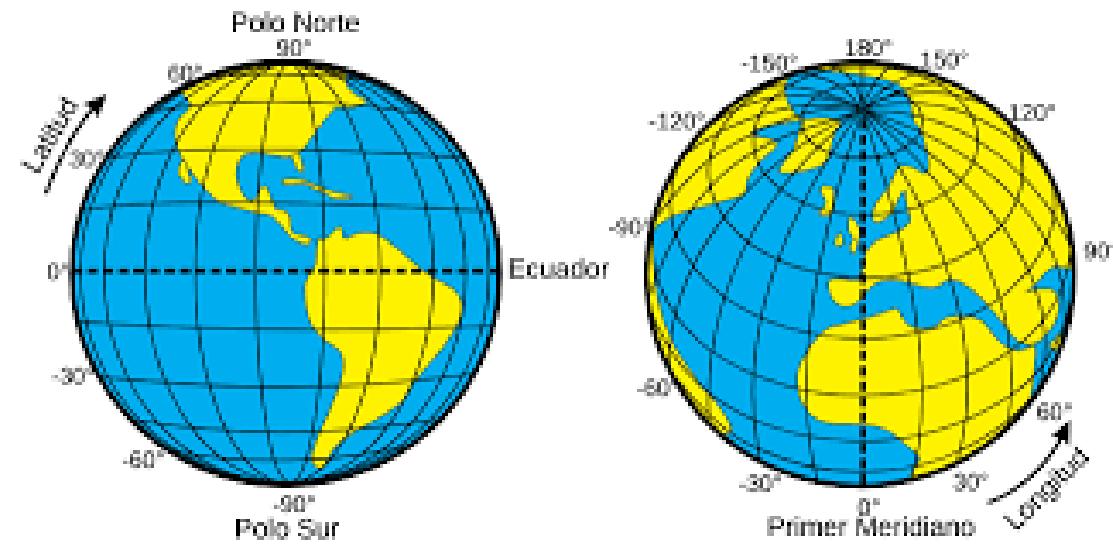
- Los enlaces de microonda tienen un retardo de 3  $\mu$ s/Km
- Los enlaces de cable coaxial o fibra óptica tienen un retardo de 5  $\mu$ s/Km
- Los satélites son esencialmente medios de **difusión**
- Cuesta igual enviar una señal a una o a miles de estaciones dentro de la footprint (huella) del transponder
- Una transmission inalámbrica está sometida a ruido
- Hay el problema de **seguridad** y **privacidad**. Es esencial la encriptación

## 2.6.2 Satélites de órbita terrestre media MEO

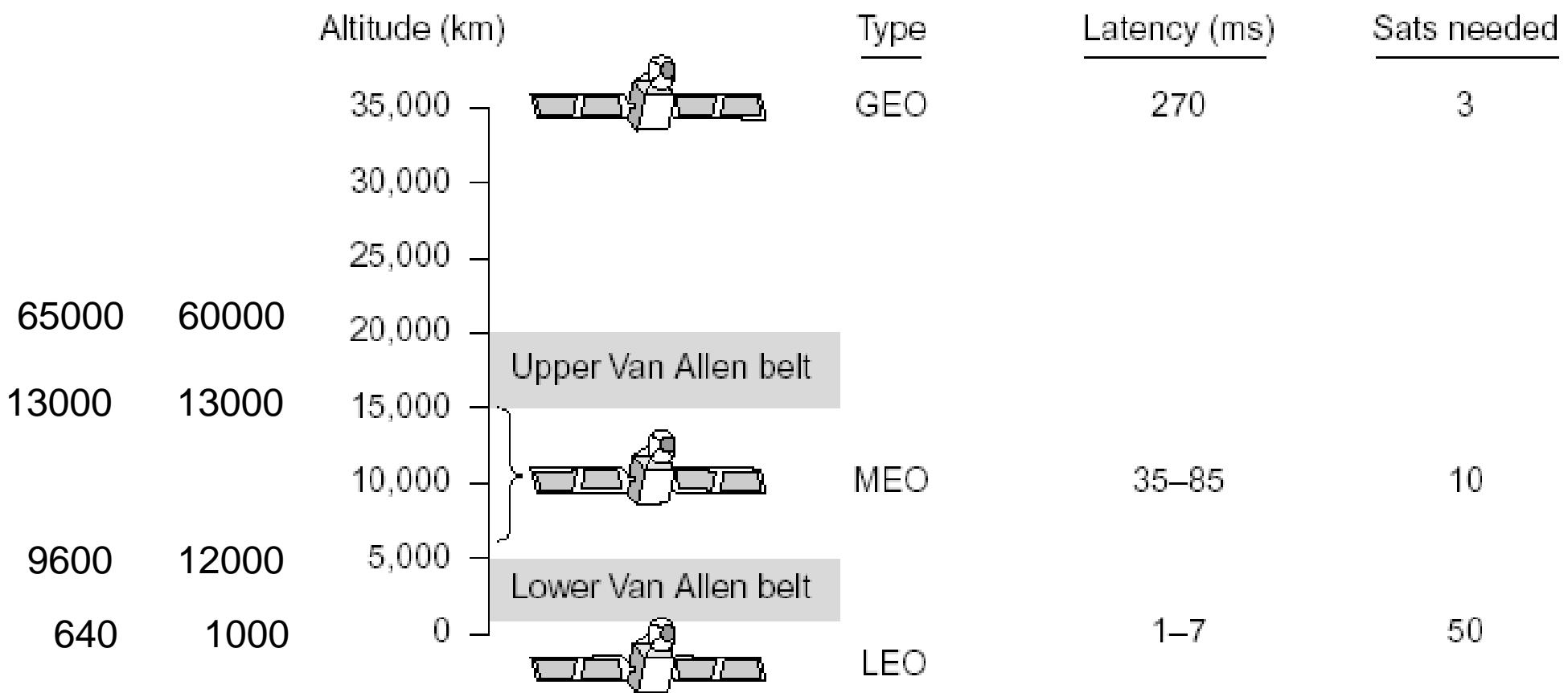
- MEO Medium Earth Orbit
- El servicio de GPS es una aplicación de los satélites MEO
- Entrega datos de latitud, longitud, altitud y tiempo
- GPS Global Positioning System: 24 satélites a una altura de 20180 km
- El período orbital de 12 horas
- Para fijar una ubicación de un punto geográfico se necesitan cuatro satélites
- Estando más cerca de la Tierra, tienen un *footprint* más pequeño
- Se requieren transmisores menos potentes

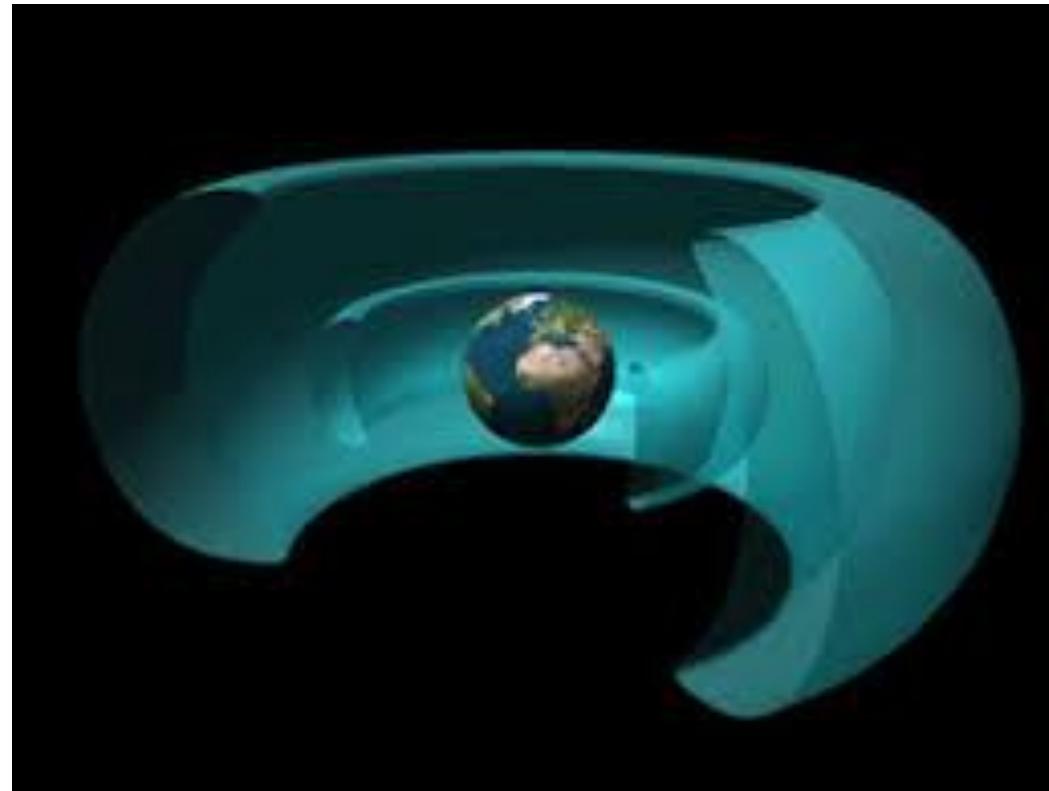


## GPS: latitud, longitud, altitud



## ChatGPT Wikipedia





Zonas de la magnetosfera  
Concentración de protones y electrones  
2 Anillos toroidales

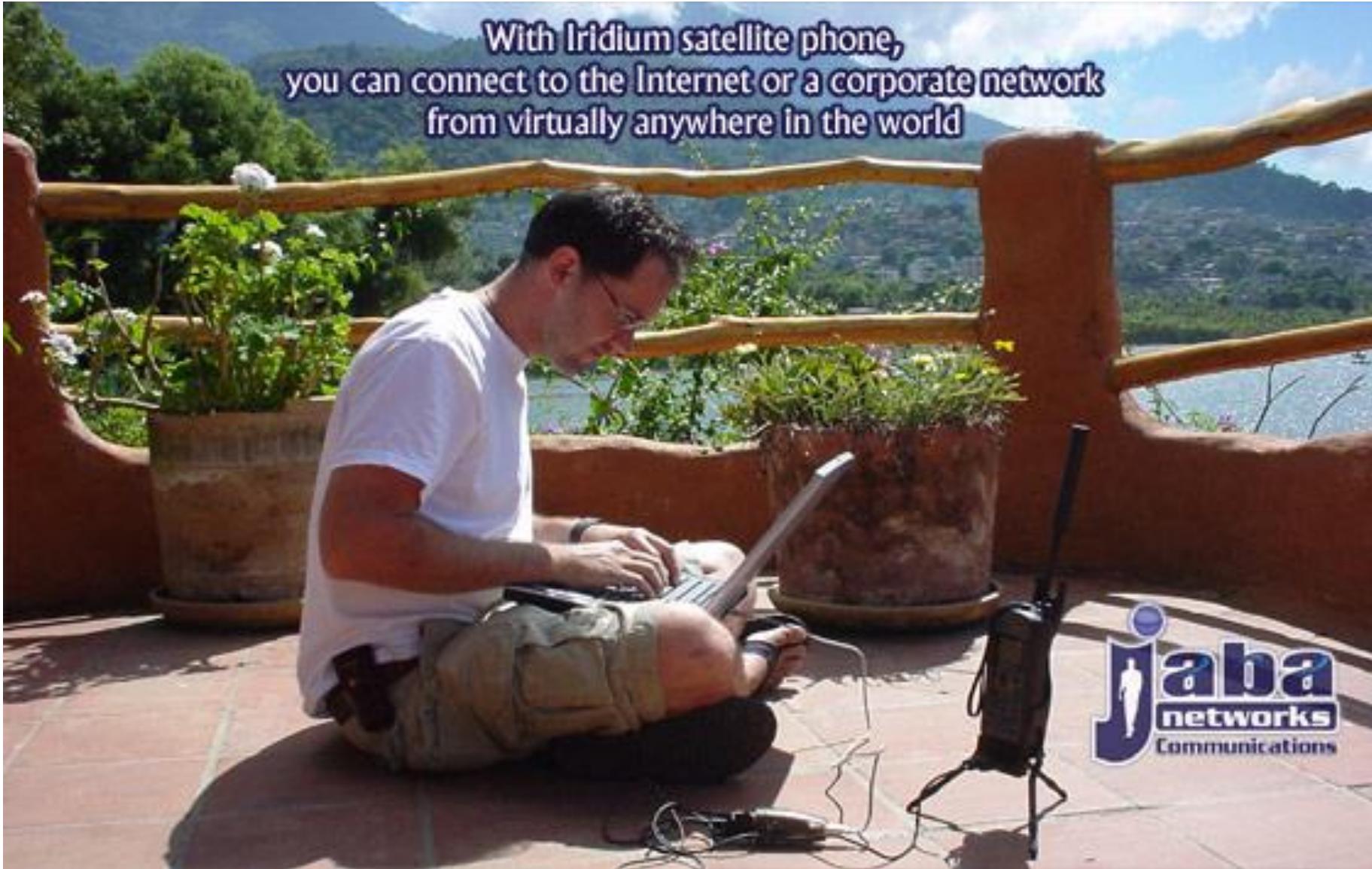
## 2.6.3 Satélites de órbita terrestre baja LEO

### ■ Aplicación de LEO: telefonía satelital

- Iridium:
- Globalstar

# Iridium

- Teléfonos por satélite
- Motorola solicitó a FCC permiso para lanzar **77** satélites LEO
- Iridio es el elemento 77 de la tabla periódica
- El servicio empezó en 1998
- La demanda comercial fue insignificante debido a la telefonía celular
- El proyecto Iridium, de \$ 5.000 millones quebró en menos de un año
- Fue uno de los fracasos corporativos más espectacular de la historia
- Luego alguien lo compró en \$ 25 millones en una especie de venta de garaje



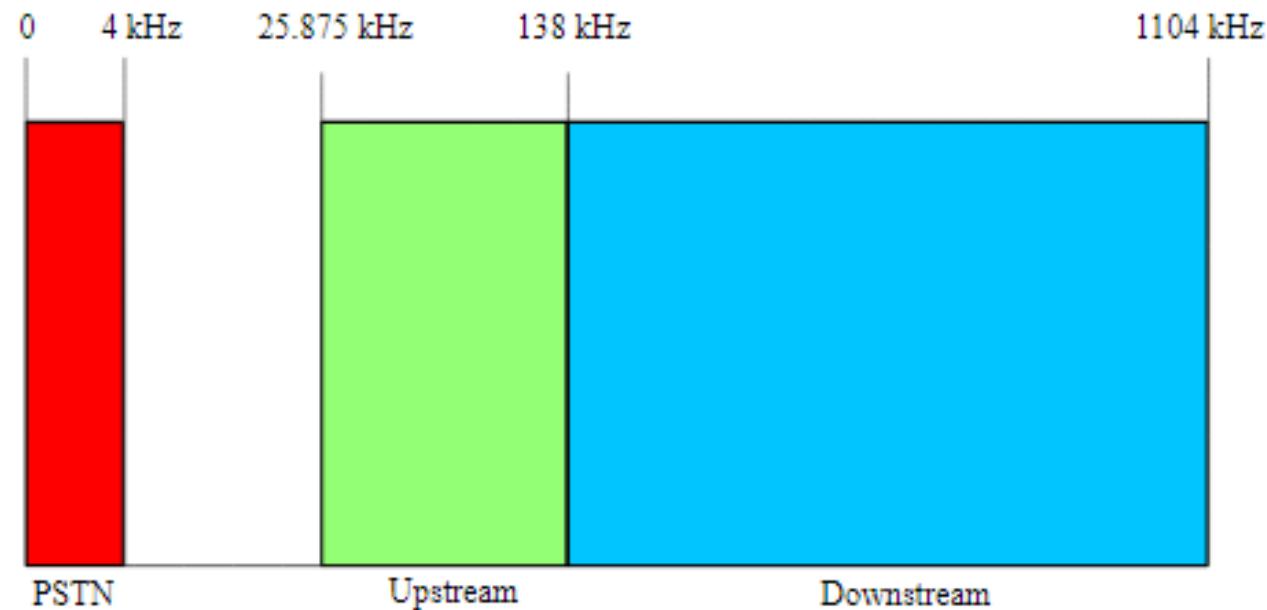
## Red de satélites Iridium



## 2.6.4 Sistema telefónico, fibra óptica y satélites

- El sistema telefónico evolucionó poco en los primeros 100 años (1876 - )
- Razones:
  - No había competencia
  - Regulaciones que **garantizaban** las utilidades sobre las inversiones
- En 1984 con la entrada de la competencia en U.S. y Europa, las telefónicas comenzaron a mejorar el servicio de Internet
- En 1987 introdujeron el servicio de banda ancha ADSL Asymmetric Digital Suscriber Line
- Para ello aplicaron la técnica de Frequency Division Multiplexing
- Posteriormente reemplazaron sus viejas redes cobre con fibra óptica

## ■ ADSL



- Ahora con fibra óptica se ofrece banda ancha a usuarios finales
- Si no hay red de acceso con ADSL o FO, un usuario instala una VSAT en el techo de la casa y se conecta a Internet a través del satélite
- El país asiático Indonesia tiene su propio satélite para el tráfico telefónico interno
- El lanzamiento de un satélite resultó más económico que el enlace de miles de cables bajo al mar entre las 13.667 islas que conforman el archipiélago
- El satélite permite despliegue rápido del servicio

# Indonesia



- No hay una tecnología de comunicaciones que sea óptima para todas las circunstancias
- Unas veces es más adecuado par trenzado con ADSL, otras veces es mejor la FO, y en otras la única opción será el satélite



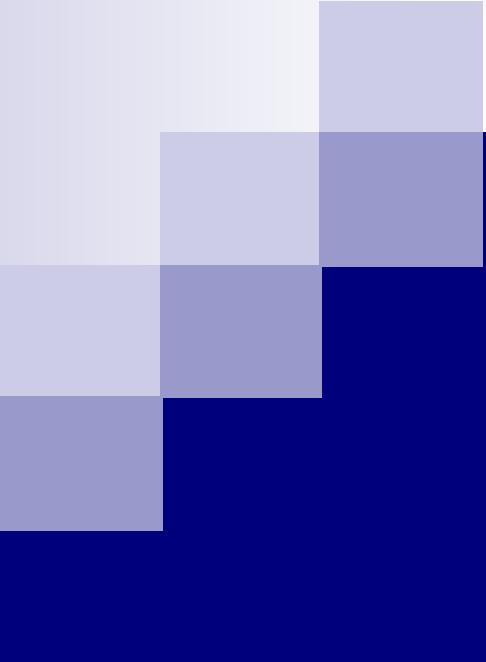
### 3. LA CAPA DE ENLACE DE DATOS

# Contenido

- 3.1 Cuestiones de diseño de la capa de enlace de datos
- 3.2 Detección y corrección de errores
- 3.3 Protocolos elementales de enlace de datos
- 3.4 Protocolos de ventana deslizante

## ■ Limitaciones de los medios de comunicación:

- Tienen una **limitada** tasa de datos
- **Retardan** la propagación de los bits
- Se producen **errores** en la transmisión de los datos



## 3.1 Cuestiones de diseño de la capa de enlace de datos



## ■ Las funciones de la capa 2 son dos:

1. Maneja los **errores** de transmisión
2. Regula el **flujo** de datos: emisores rápidos y receptores lentos

- Para realizar las funciones, la capa 2 encapsula los **paquetes** de la capa 3 en **tramas**
- Trama:
  - Encabezado (head)
  - Carga útil: **paquete** (packet)
  - Cola (tail)
- Los controles de flujo y errores los realizan las capas 2 y 4

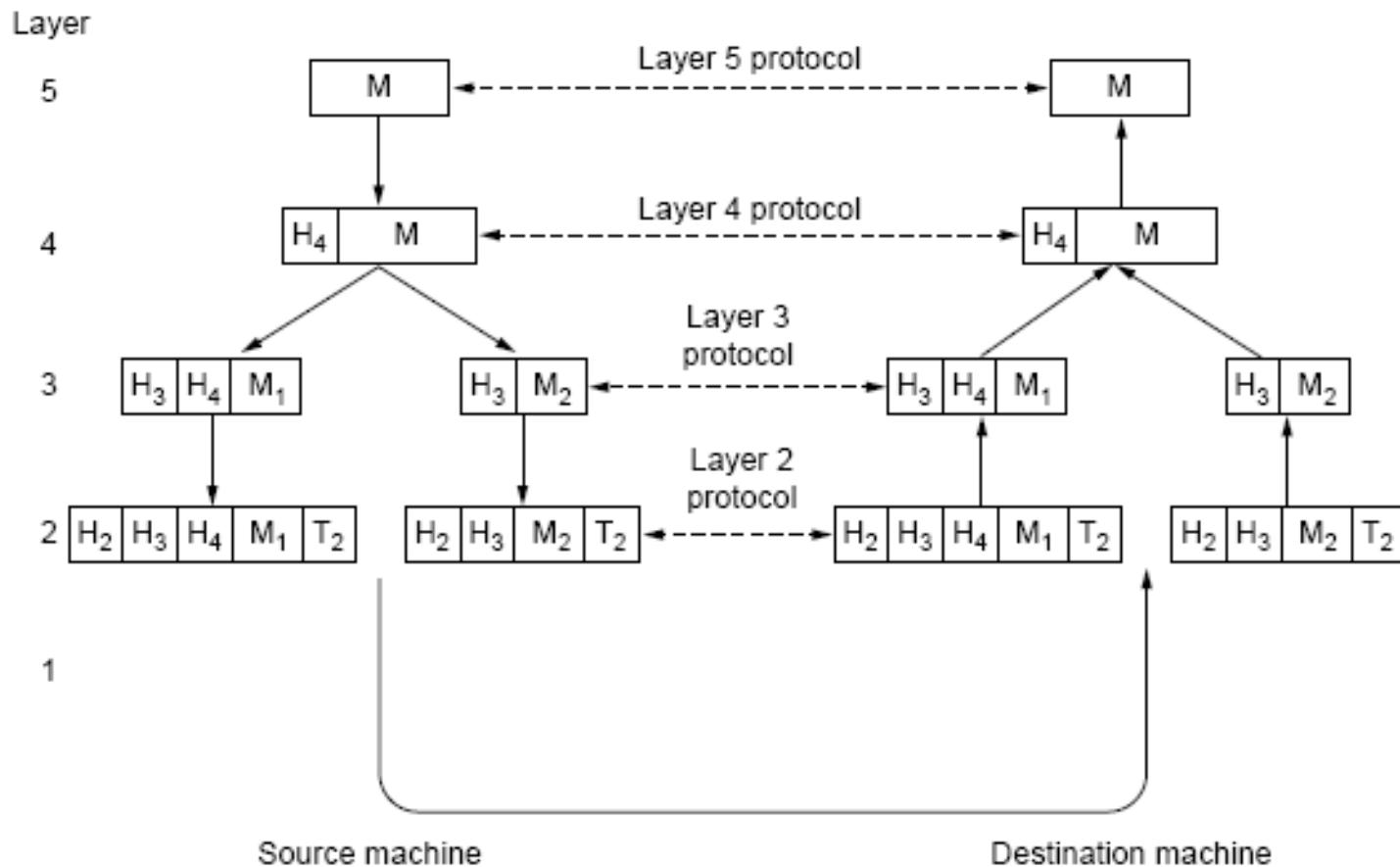


Fig. 1-15. Example information flow supporting virtual communication in layer 5.

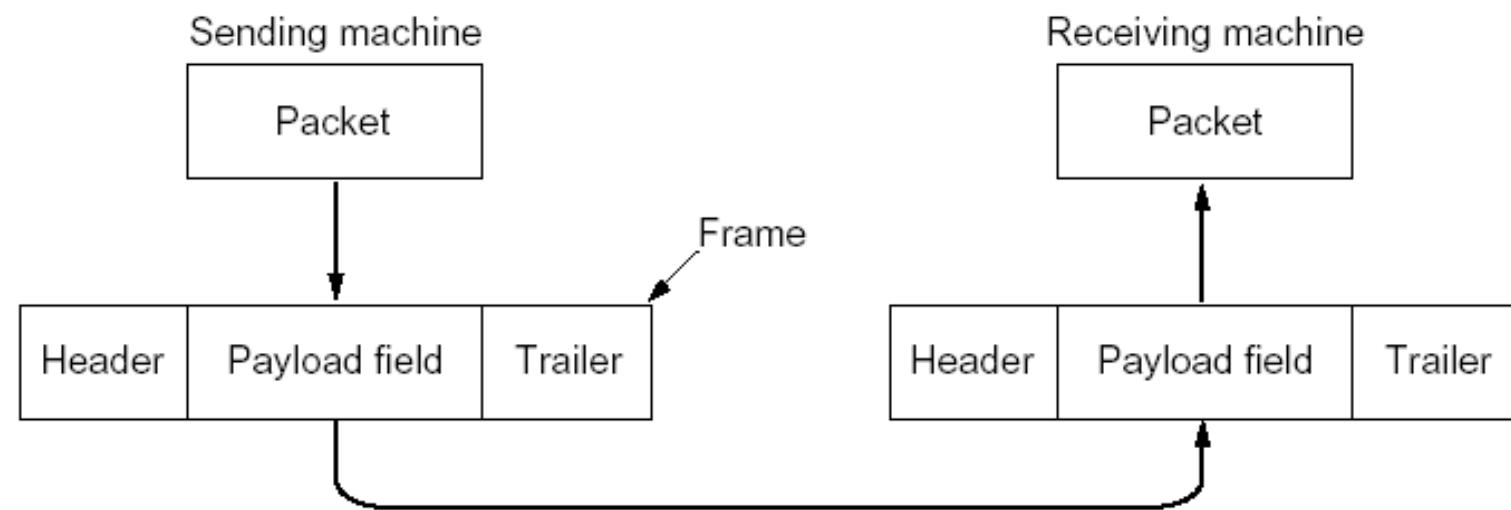


Fig. 3-1. Relationship between packets and frames.

### 3.1.1 Servicios a la capa de red

- Servicio principal: **transferir datos** de la capa 3 en el origen a la capa 3 en el destino

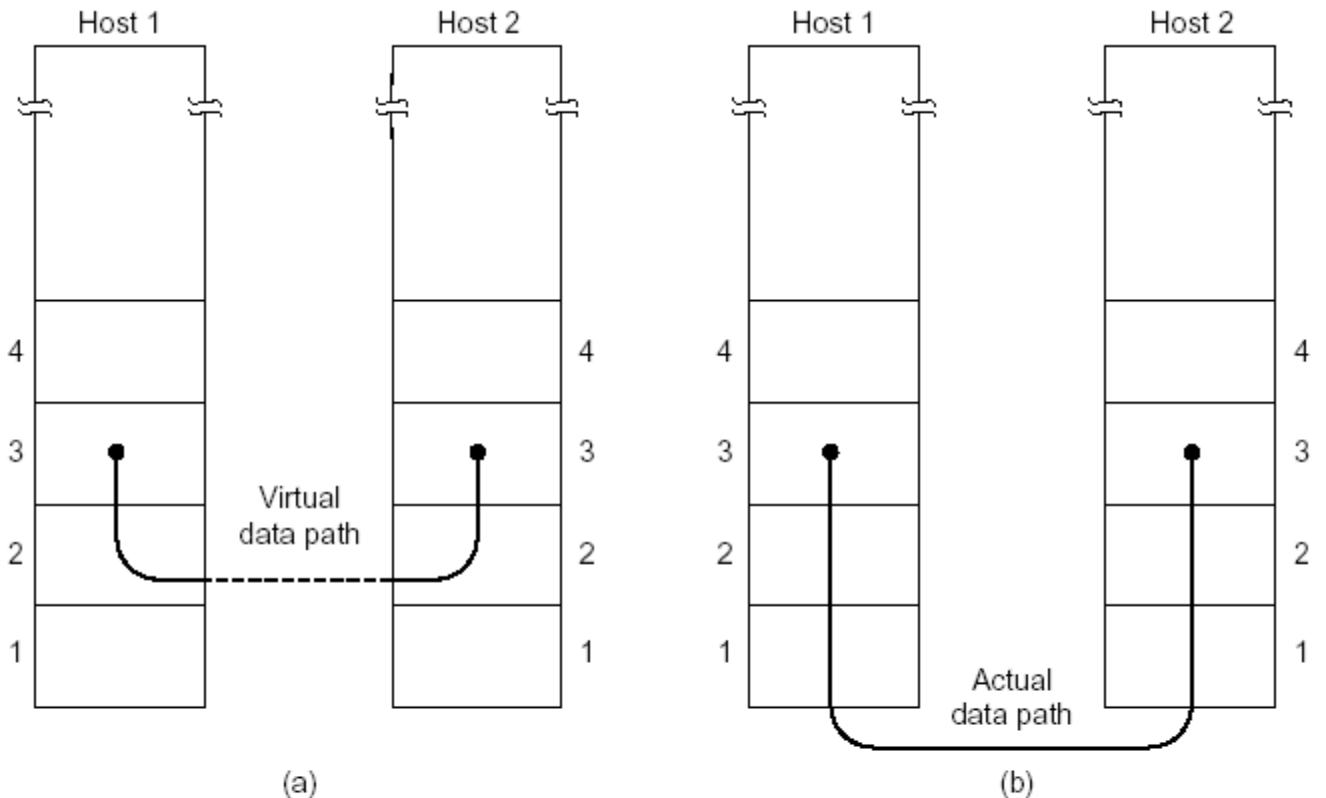


Fig. 3-2. (a) Virtual communication. (b) Actual communication.

# Formas de dar servicio de la capa de enlace

- La capa 2 tiene tres maneras de dar el servicio a la capa 3
  - Sin conexión sin ACK.
  - Sin conexión con ACK.
  - Orientado a la conexión con ACK.

# Servicio sin conexión sin ACK

- El origen envía tramas independientes al destino sin pedir confirmación (acknowledgment)
- No hay control de daño o pérdida de tramas debido al ruido en el medio
- En el destino, los datos se aceptan con o sin errores
- Servicio apropiado con medios de transmisión con baja tasa de errores
- Adecuado en:
  - redes LAN como Ethernet
  - aplicaciones de tiempo real: VoIP, videoconferencia

# Servicio sin conexión con ACK

- El receptor avisa al emisor de qué manera llegó la trama
- Si la trama no ha llegado o llega dañada, el emisor lo reenvía
- ¿Cómo sabe el emisor que la trama no ha llegado a su destino?
- Usado en canales inestables: inalámbricos
- Con FO este servicio es innecesario, ¿por qué?
- Porque los datos llega y sin errores
- Utilizado en mensajería de texto de telefonía móvil (WhatsApp): se envía un mensaje y se recibe su confirmación

# Servicio orientado a la conexión con ACK

- Es un servicio confiable
- Se simula una conexión **numerando** las tramas
- La numeración permite garantizar:
  - Recibir todas las tramas
  - En forma ordenada
  - Una sola vez
  - Sin errores
- Se simula una conexión porque las tramas llegan en orden y una sola vez
- El algoritmo podría implementarse en hardware

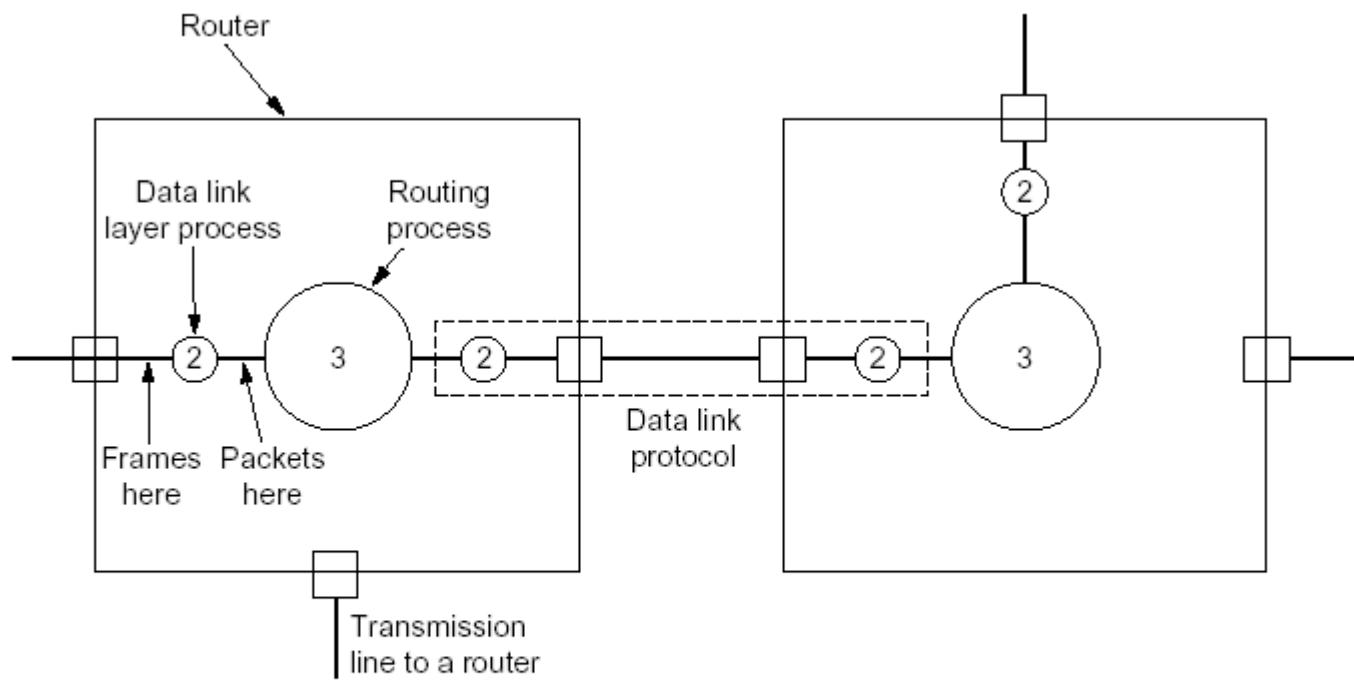


Fig. 3-3. Placement of the data link protocol.

## 3.1.2 Entramado

- Métodos para indicar el inicio y fin de una trama:
  - Inserción de intervalos de tiempo entre las tramas
  - Conteo de caracteres
  - Banderas de inicio y fin, con relleno de **caracteres**
  - Banderas de inicio y fin, con relleno de **bits**
  - Violación de codificación de la capa física

# Inserción de intervalos de tiempo

- Pausas entre tramas

- Similar a los espacios de tiempo entre palabras

- Riesgo:

- Los intervalos podrían ser eliminados
  - Por error podrían insertarse intervalos dentro de una trama

## Conteo de caracteres

- Longitud de la trama: campo en la cabecera para indicar el número de caracteres de la trama

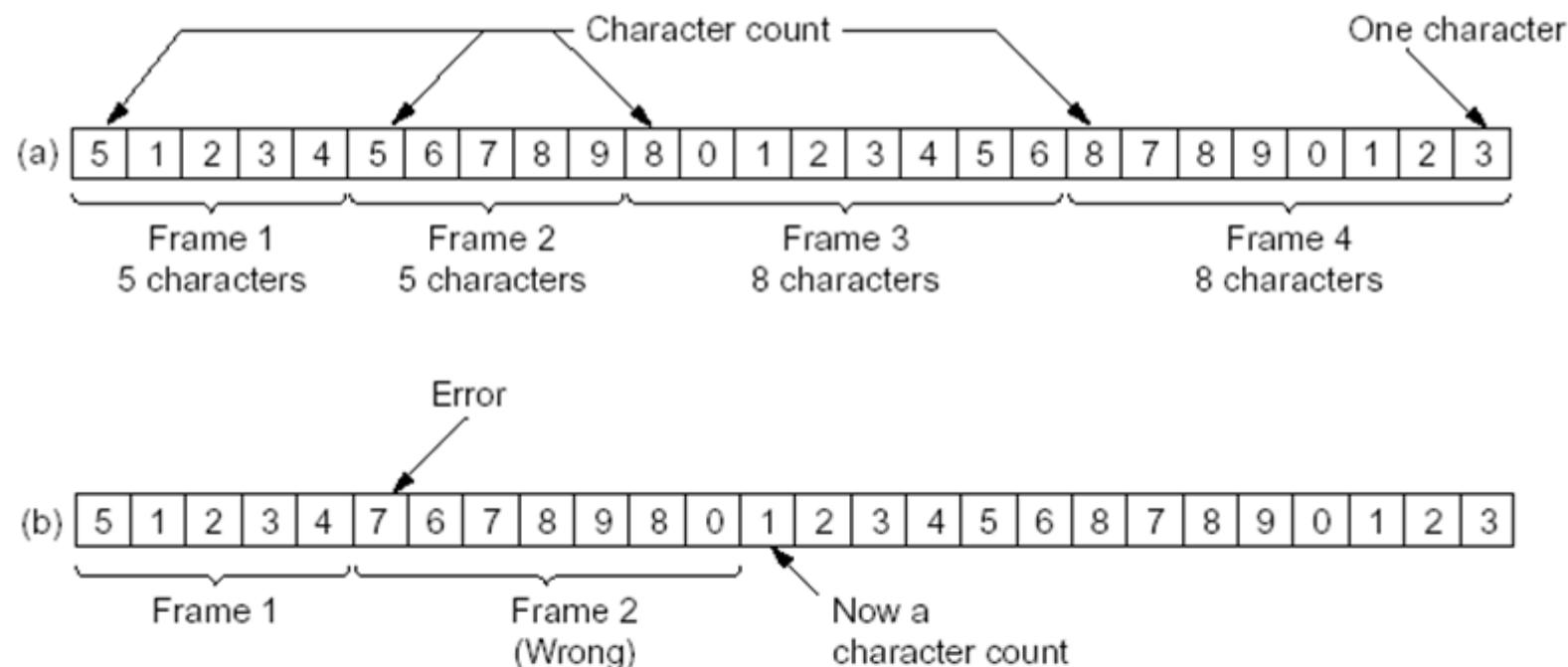
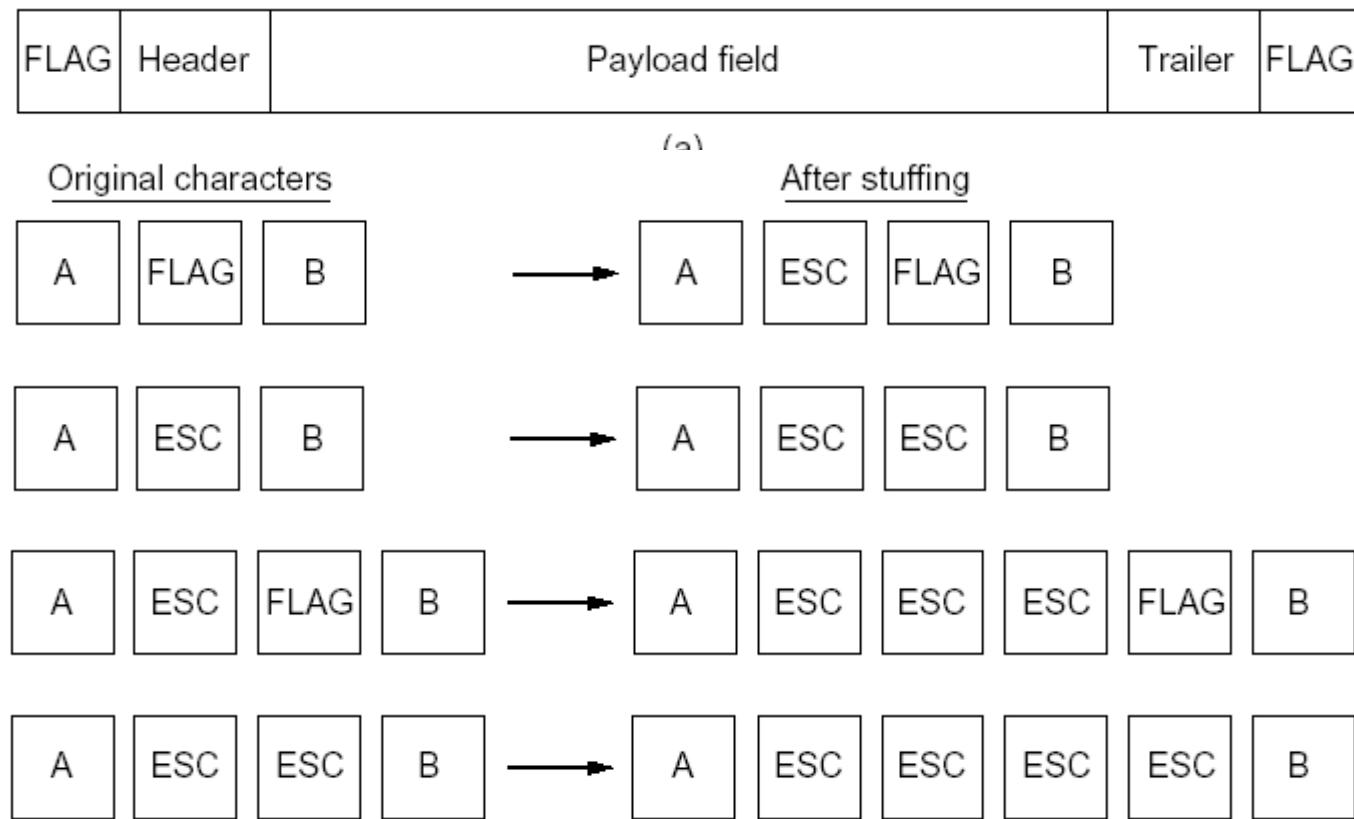


Fig. 3-4. A character stream. (a) Without errors. (b) With one error.

# Banderas con relleno de caracteres

- Inicio y fin de una trama con un byte especial
- 01111110
- La bandera sincroniza al receptor con el emisor
- Si el receptor pierde sincronía, puede esperar la siguiente bandera
- Dos banderas consecutivas señalan el fin de una trama e inicio de la siguiente
- Podría ocurrir que el patrón de bits de la bandera aparezca entre los datos
- Para solucionar se usa el byte ESC



# Banderas con relleno de bits

- Cada trama inicia y termina con 01111110 (seis 1s)
- Si la capa 2 encuentra **cinco** 1s consecutivos en los datos, **inserta un 0**
- Si el receptor ve cinco 1s de entrada seguido de un 0, extrae el 0 de relleno

# Bits de relleno en los datos

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

↑  
Stuffed bits

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

# Violaciones de codificación de la capa física

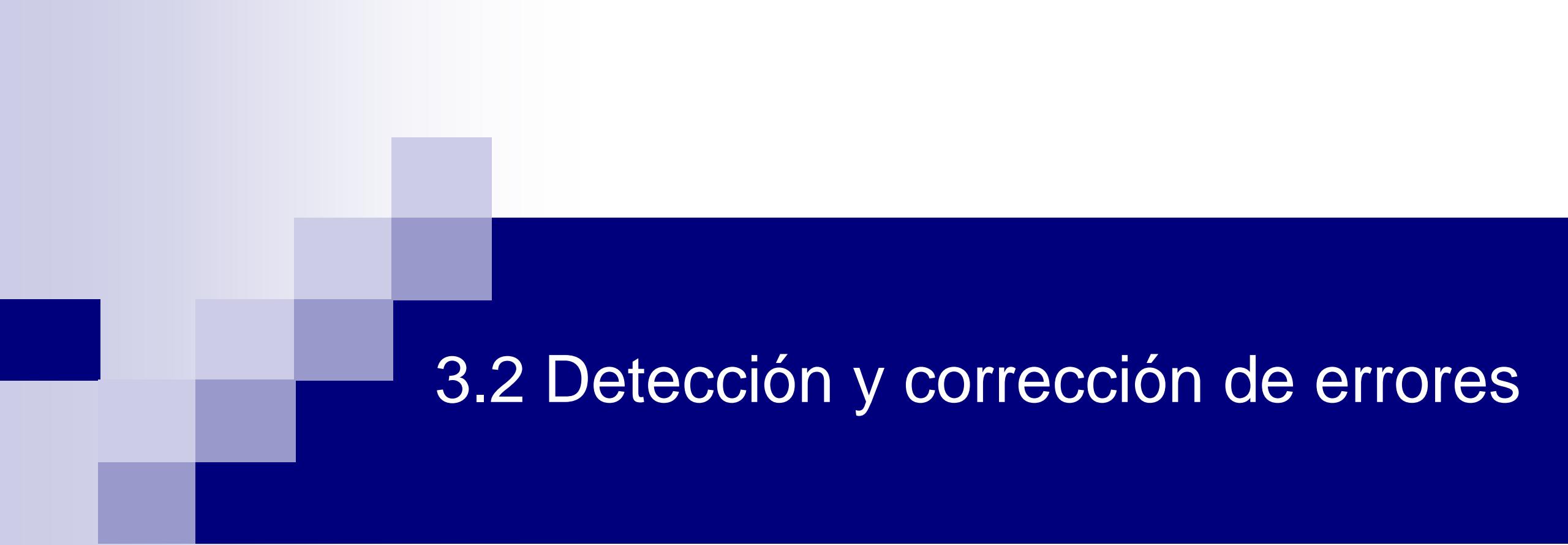
- Algunas LANs representan los bits de **datos** usando codificación **Manchester**
  - Alto – bajo: bit 1
  - Bajo – alto: bit 0
- alto-alto y bajo-bajo no se usa para datos
- Algunos protocolos usan alto-alto y bajo-bajo para **delimitar** tramas

### 3.1.3 Formas de controlar los errores

- Primero es necesario implementar un sistema de detección los errores
- Luego se retroalimenta al emisor con tramas de control ACK – NACK
- Son tramas de confirmación positiva o negativa
- ACK *Acknowledgement*
- ACK: envíe la siguiente trama
- NACK: Retransmita la trama dañada
- Si la trama de datos no llega al receptor, no hay retroalimentación:
  - Inicio de temporizador en el emisor
  - Expira el temporizador – retransmite
  - Tramas numeradas para no duplicar si el receptor si recibió la trama pero ACK se perdió

### 3.1.4 Control de flujo

- Si existen emisores rápidos y/o receptores lentos
- Hay saturación, sobrecarga o desbordamiento del receptor
- Para evitarlo se controla el flujo de tramas del emisor hacia el receptor:
  - Basado en retroalimentación al emisor. Se usa en la Capa de Enlace de Datos
  - Basado en tasa. El protocolo limita la tasa a la que el emisor puede transmitir los datos. Se asigna un determinado ancho de banda. Usado en la Capa de Red



## 3.2 Detección y corrección de errores

- Fundamentalmente los errores se producen en:
  - Última milla, si es de cobre (*local loop*, bucle local, bucle de abonado, o planta externa del proveedor del servicio)
  - Comunicación inalámbrica
- Debido a ráfagas de ruido, los errores aparecen en **ráfagas de bits**, no individualmente
- Desventaja de errores en ráfaga: más difíciles de detectar y corregir
- Ventaja: dañan una o máximo dos tramas

### 3.2.1 Códigos de corrección de errores

- Hay dos estrategias:
- Códigos de **corrección** de errores
  - Incluir información **redundante** en la trama para que el receptor corrija el error
- Códigos de **detección** de errores
  - Incluir suficiente información para que el receptor sepa que ha ocurrido un error

- Fibra óptica: tasa baja de error porque el medio es inmune al ruido
  - Es suficiente usar códigos de detección de errores
- Wireless: tasa alta de errores debido a ruido ambiental, descargas eléctricas atmosféricas, interferencia con otras señales
  - Es mejor usar códigos de corrección de errores

- En dos palabras codificadas, el número de bits en los que difieren es **h**

10001001

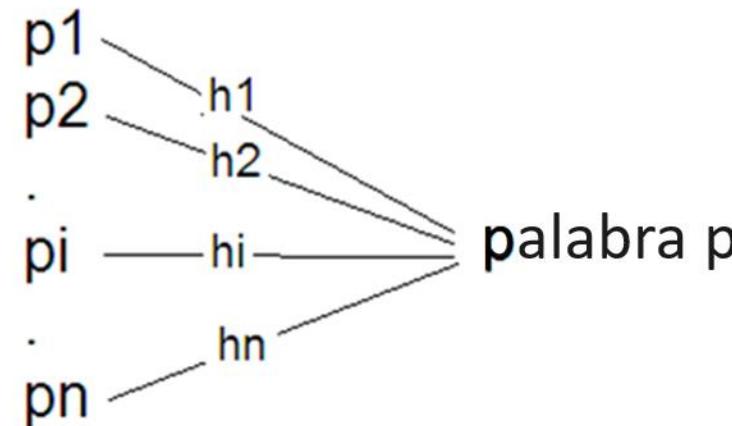
10110001

XOR 00**111**000

- En el ejemplo, **h** = 3
- **h** es la **distancia de Hamming**

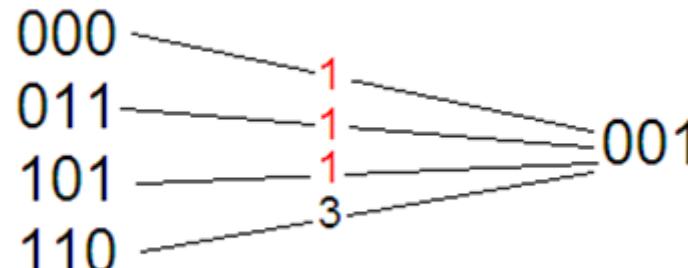
- Se puede construir una lista completa de palabras válidas
- Las palabras deben ser lo suficientemente distintas entre sí
- Para ello,  $h$  entre dos palabras no será menor a una **distancia mínima  $m$**
- La facilidad para detectar y corregir errores depende de  $h$
- Si una palabra tiene un  $h < m$  con alguna de las palabras legales, se detectará el error
- Si  $h < m$ , esta cantidad  $h$  de bits no es posible convertir una palabra válida en otra palabra válida

## Lista legal



Si  $h_i < m$ , se detecta error

## Lista legal    $m = 2$



$1 < 2$ , se detecta error

# Corrección de errores. 1era. forma

- Para corregir  $e$  bits errados o menos se necesita un código de distancia  $m = 2e + 1$
- Las palabras legales están tan separadas que alterando hasta  $e$  bits todavía esta más cerca de la palabra original

# Ejemplo de corrección de errores

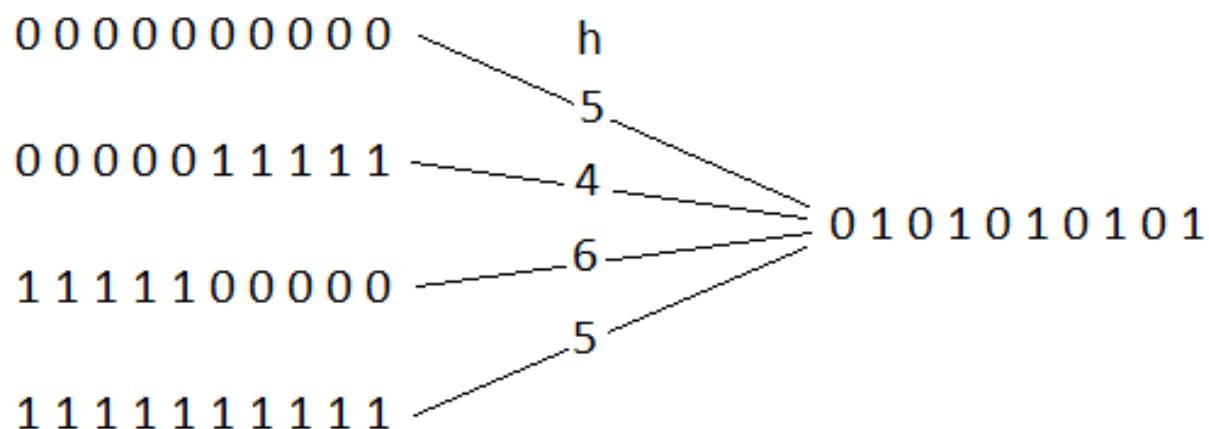
- En vez de: 00      Se tiene: 0000000000  
                          01                            0000011111  
                          10                            1111100000  
                          11                            1111111111
  
- $m = 5$
- $m = 2e + 1$
- $e = (m - 1)/2 = (5 - 1)/2 = 2$

Se pretende corregir hasta  $e = 2$  errores.

$$m = 2e + 1$$

$$m = 5$$

Códigos legales



Las distancias  $h$  son mayores a  $e = 2$

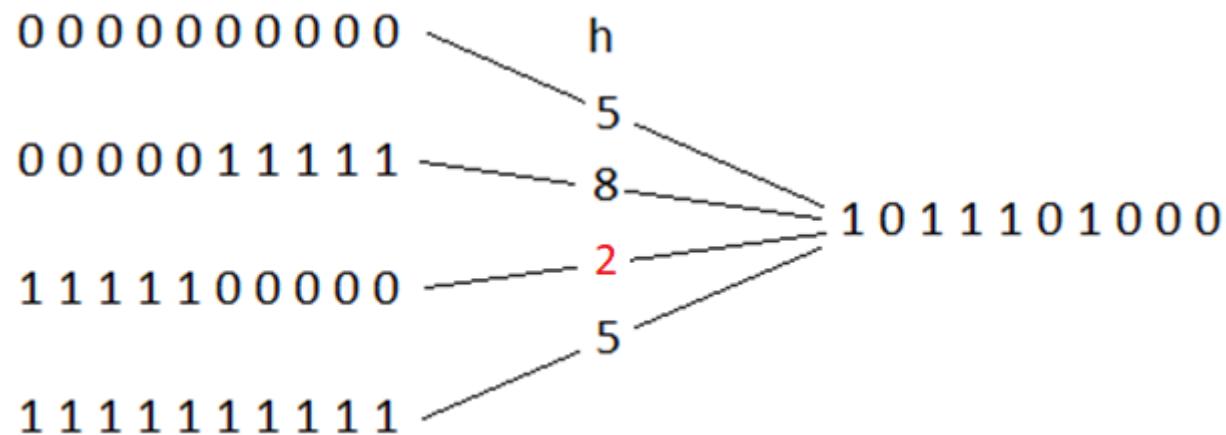
**No** se pueden corregir los errores

Se pretende corregir hasta  $e = 2$  errores.

$$m = 2e + 1$$

$$m = 5$$

Códigos legales



# Corrección de errores. Códigos de Hamming. 2da. forma

- Se desea corregir errores individuales
- Se debe determinar el número de bits redundantes  $r$
- El mensajes tiene  $m$  bits
- $n$  longitud de la trama
- $n = m + r$

# ¿Cuál es el número mínimo de bits redundantes $r$ ?

- $m$  = longitud del mensaje
- Hay  $2^m$  mensajes legales
- Si  $m = 2$  hay 4 mensajes legales: 00, 01, 10, 11
- Haremos que por cada mensaje legal haya  $n$  palabras **ilegales** a una distancia 1 del mensaje
- Entonces, el total de mensajes ilegales tiene que ser  $n2^m$
- Palabras legales + ilegales:  $2^m + n2^m = (n + 1)2^m$
- Debe hacerse cumplir que: legales + ilegales  $\leq 2^n$
- $(n + 1)2^m \leq 2^n$
- $n = m + r$ ;  $(m + r + 1)2^m \leq 2^{m+r}$
- $m + r + 1 \leq 2^r$
- Por tanto, si  $m = 2$  entonces  $r = 3$

# Ejemplo

$$m = 2; r = 3; n = m + r = 5$$

legales  $2^m = 4$  ilegales  $n2^m = 20$  con errores de 1 bit

- 00000      10000 01000 00100 00010 00001
- 01011      11011 00011 01111 01001 01010
- 10100      00100 11100 10000 10110 10101
- 11111      01111 10111 11011 11101 11110
- Legales + ilegales =  $2^m + n2^m = (n + 1)^* 2^m = 24$
- Legales + ilegales  $\leq 2^n$
- $24 \leq 32$

# Ejemplo de Códigos de Hamming

ASCII	Posición k												
	1	2	3	4	5	6	7	8	9	10	11		
	1	0	0	1	0	0	1	0	0	0	0		

$2^n$

1

2

4

8

$$3 = 1 + 2$$

	Posición k											
	1	2	3	4	5	6	7	8	9	10	11	
ASCII	1	0	0	1	0	0	0	0	0	0	0	

$2^n$

---

1

---

1

---

---

2

---

1

---

---

4

---

---

8

---

$$5 = 1 + 4$$

	Posición k											
	1	2	3	4	5	6	7	8	9	10	11	
ASCII	1	0	0	1	0	0	0	0	0	0	0	

$2^n$

---

1                  1                  0

---

2                  1

---

4                  0

---

8

$$6 = 4 + 2$$

	Posición k											
	1	2	3	4	5	6	7	8	9	10	11	
ASCII	1	0	0	1	0	0	0	0	0	0	0	

$2^n$

---

1                  1                  0

---

2                  1                  0

---

4                  0                  0

---

8

	1	2	3	4	5	6	7	8	9	10	11
ASCII	1			0	0	1		0	0	0	

$2^n$

---

1	1	0	1	0	0
---	---	---	---	---	---

---

2	1	0	1	0	0
---	---	---	---	---	---

---

4	0	0	1	0	0
---	---	---	---	---	---

---

8	0	0	0	0	0
---	---	---	---	---	---

## Paridad par de 1s

	1	2	3	4	5	6	7	8	9	10	11
ASCII	1	0	0	1	0	0	0	0	0	0	0
<hr/>											
$2^n$	1	0	1	0	1	0	0	0	0	0	0
	2		1		0	1			0	0	
	4			0	0	1					
	8						0	0	0		

## Paridad par de 1s

	1	2	3	4	5	6	7	8	9	10	11
ASCII				1	0	0	1	0	0	0	0
<hr/>											
$2^n$											
	1	0	1	0	1	0	0	0	0	0	0
	2	0	1	0	1	0	0	0	0	0	0
	4			0	0	1					
	8						0	0	0	0	0

## Paridad par de 1s

	1	2	3	4	5	6	7	8	9	10	11
ASCII				1	0	0	1	0	0	0	0
<hr/>											
$2^n$											
	1	0	1	0	1	0	0	0	0	0	0
	2	0	1	0	1	0	0	0	0	0	0
	4			1	0	0	1				
	8						0	0	0	0	0

# Código de Hamming

Posición k											
1	2	3	4	5	6	7	8	9	10	11	
0	0	1	1	0	0	1	0	0	0	0	
<hr/>											
$2^n$											
1	0	1	0	1	0	0	0	0	0	0	
2	0	1	0	1	0	0	0	0	0	0	
4			1	0	0	1					
8						0	0	0	0	0	

## Simulación de error en los datos:

Dato enviado: 1001000

Dato recibido con error: 1001001

Posición k											
1	2	3	4	5	6	7	8	9	10	11	
1	1	1	1	0	0	1	1	0	0	0	1

$2^n$

1 1 1 0 1 0 1

2 1 1 0 1 0 1

4 1 0 0 1

8 1 0 0 1

Posición de los bits de paridad: 1    2    4    8  
Bits recibidos:                        0    0    1    0  
Bits calculados en el receptor:    1    1    1    1  
    error error                      error

Sumar las potencias de 2 en donde hubo error:  
 $1 + 2 + 8 = 11$

El bit 11 está errado. Hay que invertir su valor

Error:                    1001001  
Dato corregido: 1001000

- Los Códigos de Hamming sólo corrigen errores **individuales**
- Hay un truco para corregir códigos de ráfaga
- $k$  palabras se disponen en matriz
- No se transmite por filas, sino por columnas
- Un error de ráfaga de longitud  $k$ , dañará a lo mucho 1 bit de cada una de las  $k$  palabras

Char.	ASCII	Check bits
H	1001000	00110010000
a	1100001	10111001001
m	1101101	11101010101
m	1101101	11101010101
i	1101001	01101011001
n	1101110	01101010110
g	1100111	01111001111
	0100000	10011000000
c	1100011	11111000011
o	1101111	10101011111
d	1100100	11111001100
e	1100101	00111000101

Order of bit transmission

## 3.2.2 Códigos de detección de errores

- Bit de paridad
  - Paridad par: cantidad par de 1s
  - Paridad impar: cantidad impar de 1s
- Códigos con bit de paridad tienen distancia mínima  $m = 2$

10100011

11100010

XOR 01000001 h = 2

- Cualquier error de **un solo bit** será detectado
- Es un sistema para detectar errores individuales

# Código de redundancia cíclica **CRC**

- Una cadena de bits se trata como un polinomio, con coeficientes 0 y 1
- Una trama de  $k$  bits se considera como un polinomio de  $k$  términos de grado  $k - 1$
- $b_{k-1}x^{k-1} + b_{k-2}x^{k-2} + \dots + b_1x + b_0$
- $101011 = 1x^5 + 0x^4 + 1x^3 + 0x^2 + 1x + 1$

- La suma y la resta se reducen a un XOR

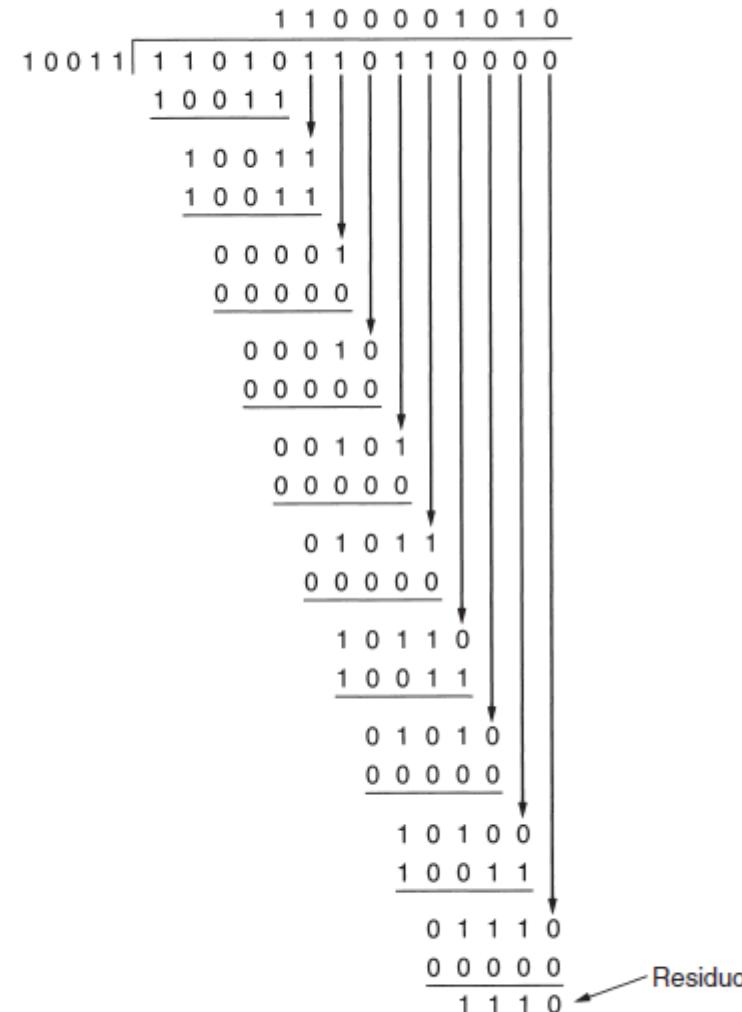
$$\begin{array}{r} 10011011 \\ \pm 11001010 \\ \hline 01010001 \end{array}$$

- Emisor y receptor acuerdan el generador  $G(x)$
- El mensaje  $M(x)$  de  $m$  bits, debe ser más grande que  $G(x)$
- Se añade la suma de verificación CRC al final de  $M(x)$
- $T(x) = M(x) + \text{CRC}$
- $T(x)$  trama que se transmite
- $T(x)$  debe ser **divisible** para  $G(x)$
- Si hay residuo, hay un error de transmisión

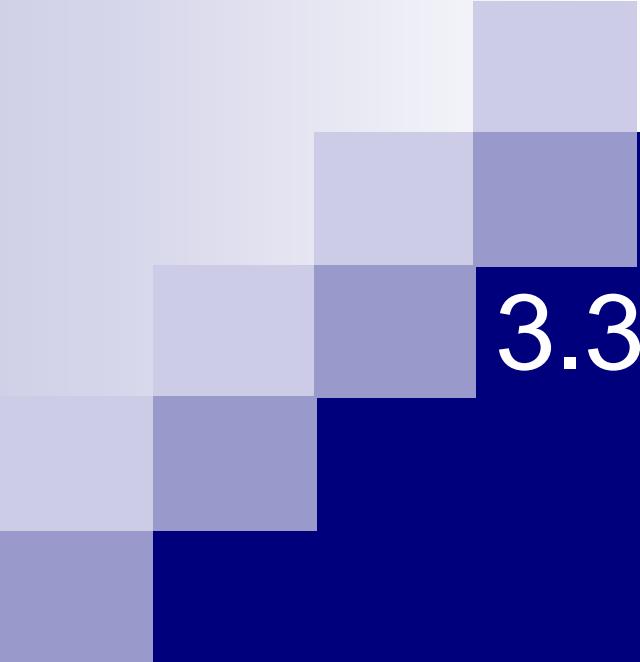
**M(X)** Trama : 1101011011

**G(X)** Generador: 10011

Mensaje tras anexar 4 bits cero: 11010110110000



**T(X)** Trama transmitida: 11010110111110



## 3.3 Protocolos elementales de enlace de datos

## Archivo *protocol.h*

```
#define MAX_PKT 1024                                /* determina el tamaño del paquete  
                                                    en bytes */  
  
typedef enum {false, true} boolean;                  /* tipo booleano */  
typedef unsigned int seq_nr;                        /* números de secuencia o  
                                                    confirmación */  
typedef struct {unsigned char data[MAX_PKT];} packet; /* definición de paquete */  
typedef enum {data, ack, nak} frame_kind;           /* definición de frame_kind */  
  
typedef struct {  
    frame_kind kind;                            /* las tramas se transportan en  
                                                esta capa */  
    seq_nr seq;                               /* ¿qué clase de trama es? */  
    seq_nr ack;                               /* número de secuencia */  
    packet info;                             /* número de confirmación de  
                                                recepción */  
} frame;                                         /* paquete de la capa de red */  
  
/* Espera que ocurra un evento; devuelve el tipo en la variable event. */  
void wait_for_event(event_type *event);  
  
/* Obtiene un paquete de la capa de red para transmitirlo por el canal. */  
void from_network_layer(packet *p);  
  
/* Entrega información de una trama entrante a la capa de red. */  
void to_network_layer(packet *p);  
  
/* Obtiene una trama entrante de la capa física y la copia en r. */  
void from_physical_layer(frame *r);
```

```
/* Pasa la trama a la capa física para transmitirla. */
void to_physical_layer(frame *s);

/* Arranca el reloj y habilita el evento de expiración de temporizador. */
void start_timer(seq_nr k);

/* Detiene el reloj e inhabilita el evento de expiración de temporizador. */
void stop_timer(seq_nr k);

/* Inicia un temporizador auxiliar y habilita el evento ack_timeout. */
void start_ack_timer(void);

/* Detiene el temporizador auxiliar e inhabilita el evento ack_timeout. */
void stop_ack_timer(void);

/* Permite que la capa de red cause un evento network_layer_ready. */
void enable_network_layer(void);

/* Evita que la capa de red cause un evento network_layer_ready. */
void disable_network_layer(void);

/* La macro inc se expande en linea: incrementa circularmente k. */
#define inc(k) if (k < MAX_SEQ) k = k + 1; else k = 0
```

## ■ Tres casos con niveles de complejidad crecientes

- Protocolo simplex sin restricciones
- Protocolo simplex de parada y espera
- Protocolo simplex para un canal ruidoso

### 3.3.1 Protocolo simplex sin restricciones

- Es sencillo
- Es irreal
- Los datos se transmiten sólo en una dirección
- Las capas de red del emisor y receptor siempre están listas
- Se ignora el tiempo de procesamiento
- Hay un espacio infinito de buffer
- Canal no produce errores ni pierde tramas

```

typedef enum {frame_arrival} event_type;
#include "protocol.h"

void sender1(void)
{
    frame s;                                /* búfer para una trama de salida */
    packet buffer;                           /* búfer para un paquete de salida */

    while (true) {
        from_network_layer(&buffer); /* consigue algo que enviar */
        s.info = buffer;                /* lo copia en s para transmisión */
        to_physical_layer(&s);        /* lo envía a su destino */
    }                                         /* Mañana, y mañana, y mañana,
                                                Se arrastra a este mísero paso de día
                                                a día
                                                Hasta la última silaba del tiempo
                                                recordado
                                                -Macbeth, V, v */
}

void receiver1(void)
{
    frame r;
    event_type event;                      /* ocupado por wait, pero no se usa aquí */

    while (true) {
        wait_for_event(&event);           /* la única posibilidad es frame_arrival */
        from_physical_layer(&r);         /* obtiene la trama entrante */
        to_network_layer(&r.info);       /* pasa los datos a la capa de red */
    }
}

```

- No se usan números de secuencia ni confirmaciones de recepción
- Único evento posible: llega tramas sin daños

### 3.3.2 Protocolo simplex de parada y espera

- El receptor requiere un tiempo para recibir y procesar la trama que llega
- Se evita que el emisor **sature** al receptor
- Se supone que el canal de comunicaciones está libre de errores
- El tráfico de datos es simplex
- **Solución:** receptor retroalimenta al emisor
- Luego de enviar una trama, el emisor espera que llegue una trama ACK
- La trama ACK autoriza al emisor para que envíe la siguiente trama
- Este protocolo se llama de «parada y espera»

```

void sender2(void)
{
    frame s;                                /* búfer para una trama de salida */
    packet buffer;                           /* búfer para un paquete de salida */
    event_type event;                        /* frame_arrival es la única posibilidad */

    while (true) {
        from_network_layer(&buffer);       /* consigue algo que enviar */
        s.info = buffer;                   /* lo copia en s para transmisión */
        to_physical_layer(&s);           /* adiós a la pequeña trama */
        wait_for_event(&event);          /* no procede hasta que recibe la señal de
                                         continuación */
    }
}

void receiver2(void)
{
    frame r, s;                                /* búferes para las tramas */
    event_type event;                           /* frame_arrival es la única posibilidad */

    while (true) {
        wait_for_event(&event);           /* la única posibilidad es frame_arrival */
        from_physical_layer(&r);         /* obtiene la trama entrante */
        to_network_layer(&r.info);       /* pasa los datos a la capa de red */
        to_physical_layer(&s);           /* envía una trama ficticia para informar
                                         al emisor */
    }
}

```

- En realidad hay flujo de información en ambas direcciones en una alternancia estricta

### 3.3.3 Protocolo simplex para un canal ruidoso

- El canal comete errores o pierde tramas
- Puede ser que el ACK enviado por el receptor se pierda
  - El emisor lo vuelve a enviar duplicándose la trama
  - Para evitar problemas se **numera las tramas**

- El número de secuencia es de 1 bit (0 ó 1)
- Éste se incrementa módulo 2
- Se transmite también en una sola dirección
- Los protocolos en los que el emisor espera confirmación se llaman ARQ (Automatic Repeat reQuest)

```

#define MAX_SEQ 1                                /* debe ser 1 para el protocolo 3 */
typedef enum {frame_arrival, cksum_err, timeout} event_type;
#include "protocol.h"

void sender3(void)
{
    seq_nr next_frame_to_send;                /* número de secuencia de la siguiente
                                                trama de salida */
    frame s;                                  /* variable de trabajo */
    packet buffer;                           /* búfer para un paquete de salida */

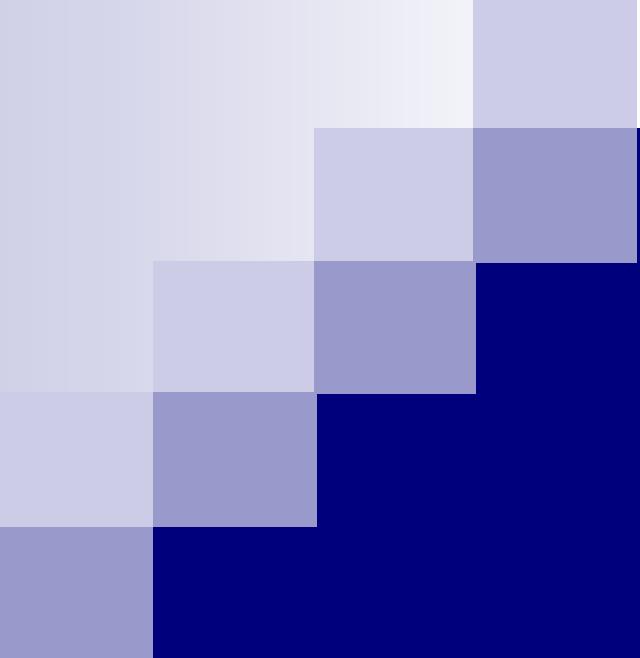
    next_frame_to_send = 0;                   /* inicializa números de secuencia de
                                                salida */
    from_network_layer(&buffer);           /* obtiene el primer paquete */
    while (true){
        s.info = buffer;
        s.seq = next_frame_to_send;

        to_physical_layer(&s);             /* construye una trama para transmisión */
        start_timer(s.seq);                /* inserta un número de secuencia en la
                                                trama */
        /* la envia a su destino */
        /* si la respuesta tarda mucho, expira el
           temporizador */
        wait_for_event(&event);            /* frame_arrival, cksum_err, timeout */
        if (event == frame_arrival){
            from_physical_layer(&s);       /* obtiene la confirmación de recepción */
            if (s.ack == next_frame_to_send){
                stop_timer(s.ack);          /* desactiva el temporizador */
                from_network_layer(&buffer); /* obtiene siguiente a enviar */
                inc(next_frame_to_send);    /* invierte next_frame_to_send */
            }
        }
    }
}

```

```
void receiver3(void)
{
    seq_nr frame_expected;
    frame r, s;
    event_type event;

    frame_expected = 0;
    while (true){
        wait_for_event(&event);           /* posibilidades: frame_arrival, cksum_err */
        if (event == frame_arrival){      /* ha llegado una trama válida. */
            from_physical_layer(&r);    /* obtiene la trama recién llegada */
            if (r.seq == frame_expected){/* esto es lo que hemos estado esperando. */
                to_network_layer(&r.info);/* pasa los datos a la capa de red */
                inc(frame_expected);     /* para la próxima se espera el otro número
                                            de secuencia */
            }
            s.ack = 1 - frame_expected;  /* indica la trama cuya recepción se está
                                            confirmando */
            to_physical_layer(&s);       /* envía confirmación de recepción */
        }
    }
}
```



## 3.4 Protocolos de ventana deslizante

- Una forma de transmitir datos **dúplex** es con **2** circuitos simplex
- Otra forma es usar un circuito para datos en ambas direcciones
- El campo *kind* de la cabecera de la trama indica si es de datos o ACK

## Técnica de superposición o *piggybacking*

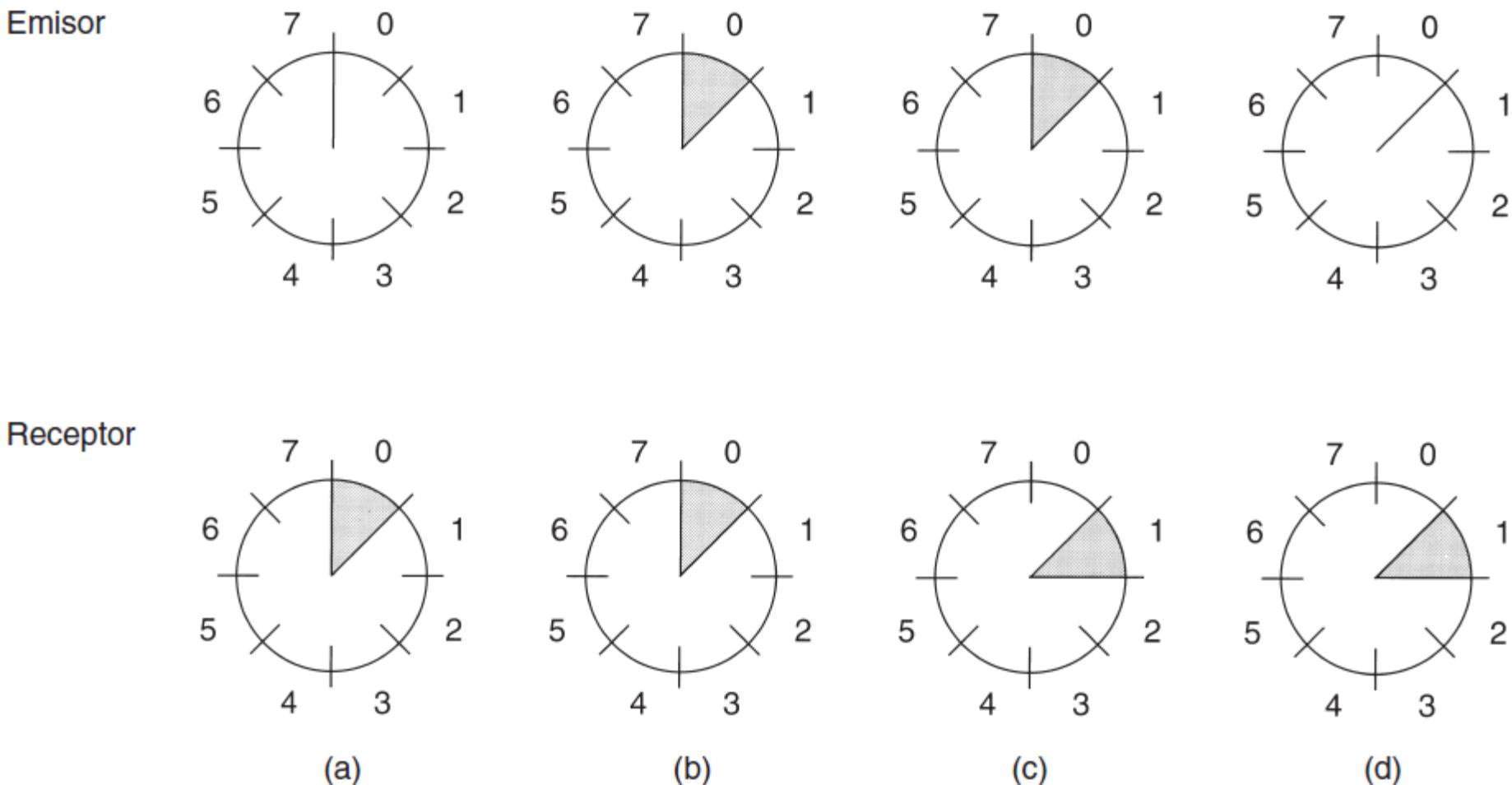
- Se anexa un ACK a una trama de datos de retorno, usando el campo ack del encabezado
- ACK viaja gratis
- Se aprovecha mejor el BW del canal
- Hay menos interrupciones de “*ha llegado una trama*”

- **Problema:** Podría expirar el temporizador del emisor y retransmitir la trama porque el receptor no tiene datos que enviar
- **Solución:** El receptor espera un tiempo. Si el receptor no tiene datos que enviar, envía solo un ACK

# Tres protocolos de ventana deslizante

- Son bidireccionales:
  - Protocolo de ventana corrediza de un bit
  - Protocolo de retroceso N
  - Protocolo de repetición selectiva
- Cada trama tiene un número de secuencia
- El **emisor** registra las tramas enviadas aun no confirmadas
- El **receptor** tiene una ventana de tramas a aceptar

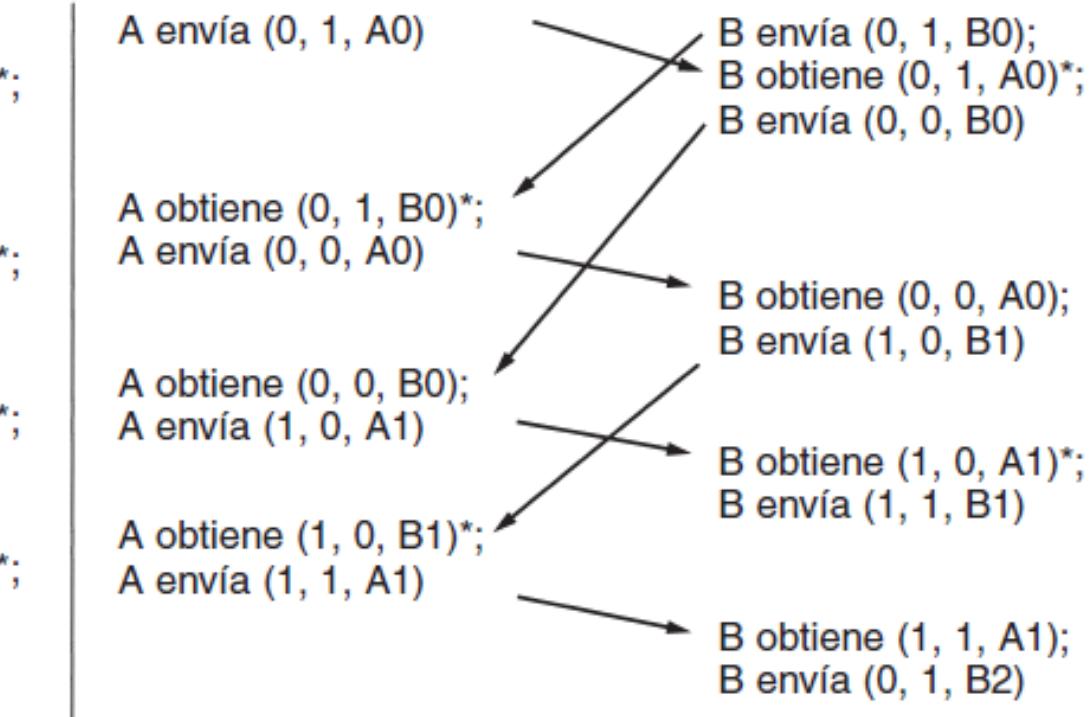
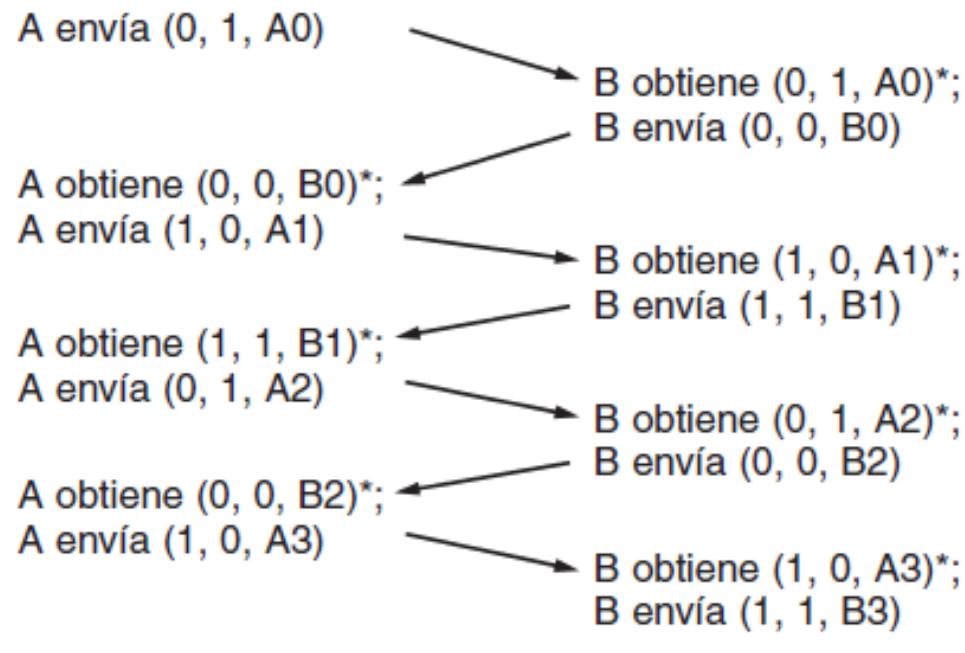
- Estos protocolos entregan en orden los paquetes a la capa 3
- En el emisor:
  - Si llega un paquete nuevo de la capa 3, el extremo superior de la ventana avanza 1
  - Al llegar un ACK, el extremo inferior de la ventana avanza 1



**Figura 3-13.** Ventana corrediza de tamaño 1, con un número de secuencia de 3 bits. (a) Al inicio. (b) Tras la transmisión de la primera trama. (c) Tras la recepción de la primera trama. (d) Tras recibir la primera confirmación de recepción.

### 3.4.1 Protocolo de ventana corrediza de 1 bit

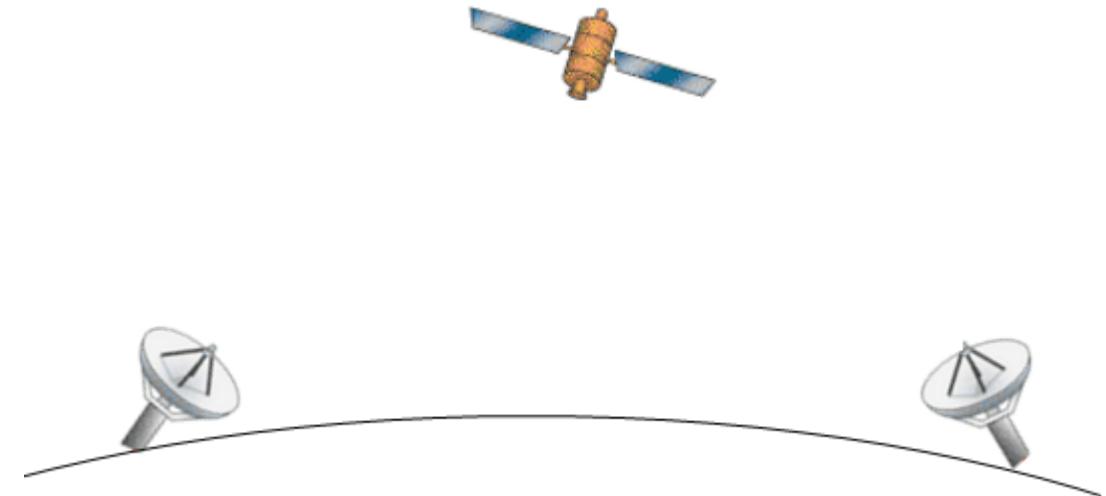
- Utiliza parada y espera
- Las tramas se numeran con 0, 1, 0, 1, ...
- Sólo una de las máquinas puede iniciar la transmisión
- Si las 2 máquinas iniciaran la transmisión al mismo tiempo hay funcionamiento anormal



**Figura 3-15.** Dos escenarios para el protocolo 4. (a) Caso normal. (b) Caso anormal. La notación es (secuencia, confirmación de recepción, número de paquete). Un asterisco indica el lugar en que una capa de red acepta un paquete.

### 3.4.2 Protocolo que usa retroceso N: Canalización

- Enlace satelital de  $C = 50 \text{ kbps}$
- Satélite geoestacionario 36.000 km
- Ida y vuelta al emisor es  $4 \times 36.000 \text{ km}$
- Retardo de propagación de ida y vuelta  $P$
- $$P = 4 \times \frac{36.000 \text{ km}}{300.000 \text{ km/s}} = 480 \text{ ms}$$
- Longitud de la trama  $L = 1000 \text{ bits}$
- Tiempo de transmisión de la trama  $T$
- $$T = \frac{L}{C} = \frac{1000 \text{ bits}}{50 \text{ kbps}} = 20 \text{ ms};$$
- Tiempo de transmisión y recibir ACK de una trama  $TP = T + P = 20 \text{ ms} + 480 \text{ ms} = 500 \text{ ms}$
- El emisor espera la confirmación de la trama enviada antes de enviar las siguiente trama
- Con ACK corto, el uso del canal  $U1 = T / (TP) \times 100 = (20 \text{ s} / 500 \text{ ms}) \times 100 = 4.0\%$



# Solución

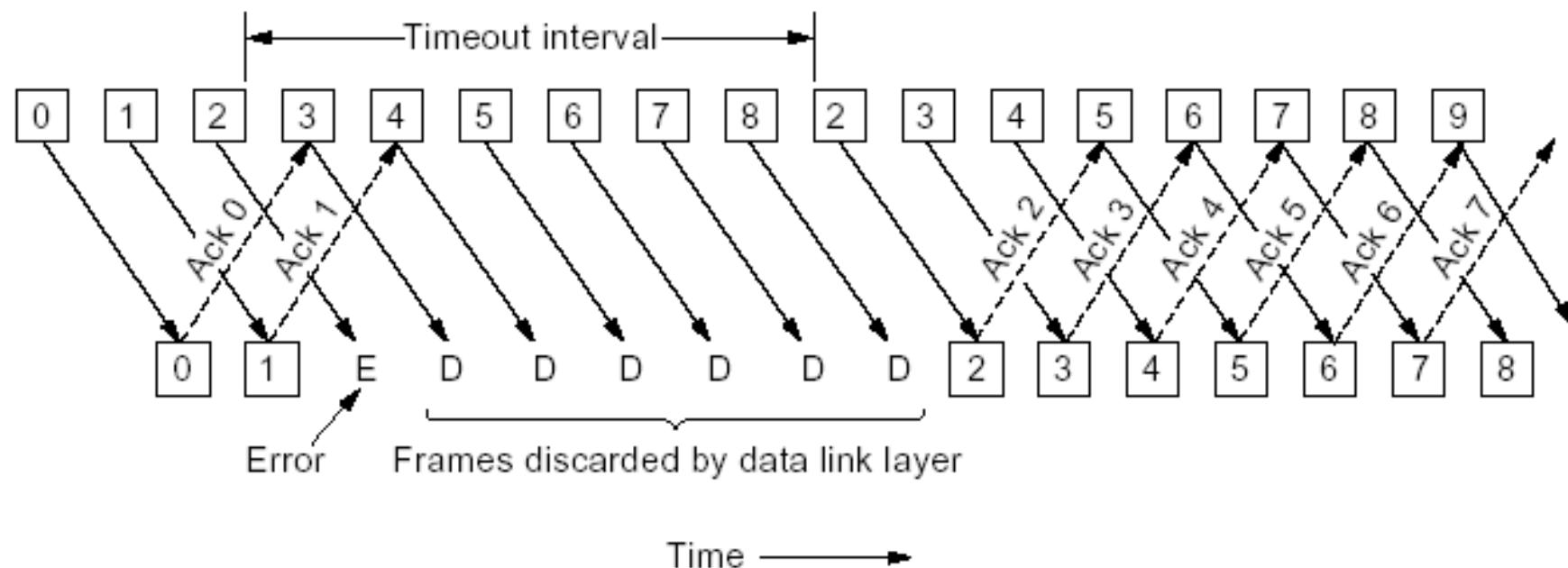
- **Canalización.** El emisor envía  $w$  tramas durante el tiempo total  $\mathbf{TP} = \textcolor{blue}{T} + \mathbf{P} = 500 \text{ ms}$
- ¿Cuántas tramas puede enviar el emisor, o cuál es el tamaño de la ventana  $w$ ?
- $w = \mathbf{TP} / \textcolor{blue}{T} = 500 \text{ ms} / 20 \text{ ms} = 25 \text{ tramas}$
- El emisor ocupa el canal por 500 ms para enviar 25 tramas
- El emisor espera los ACKs de todas ellas antes de enviar las siguientes 25 tramas
- Tiempo para transmitir y recibir ACKs de 25 tramas  $\mathbf{TT}$
- $\mathbf{TT} = \mathbf{TP} + 24\textcolor{blue}{T} = 500 \text{ ms} + 24 \times 20 \text{ ms} = 980 \text{ ms}$
- $24\textcolor{blue}{T}$  es el tiempo que demora en llegar al receptor los ACKs de las tramas restantes
- Uso del canal  $U2 = \text{tiempo de transmisión de } w \text{ tramas} / \mathbf{TT}$
- $U2 = (500 \text{ ms} / 980 \text{ ms}) \times 100 = 51\%$
- Mucho mayor uso del canal

- Al terminar de enviar la trama 26 en  $t = 520$  ms, llega la confirmación de recepción de la trama 1
- Entonces las confirmaciones llegan cada 20 ms

# Métodos para manejar **errores** en canalización

1. Retroceso n
2. Repetición selectiva

# Retroceso n



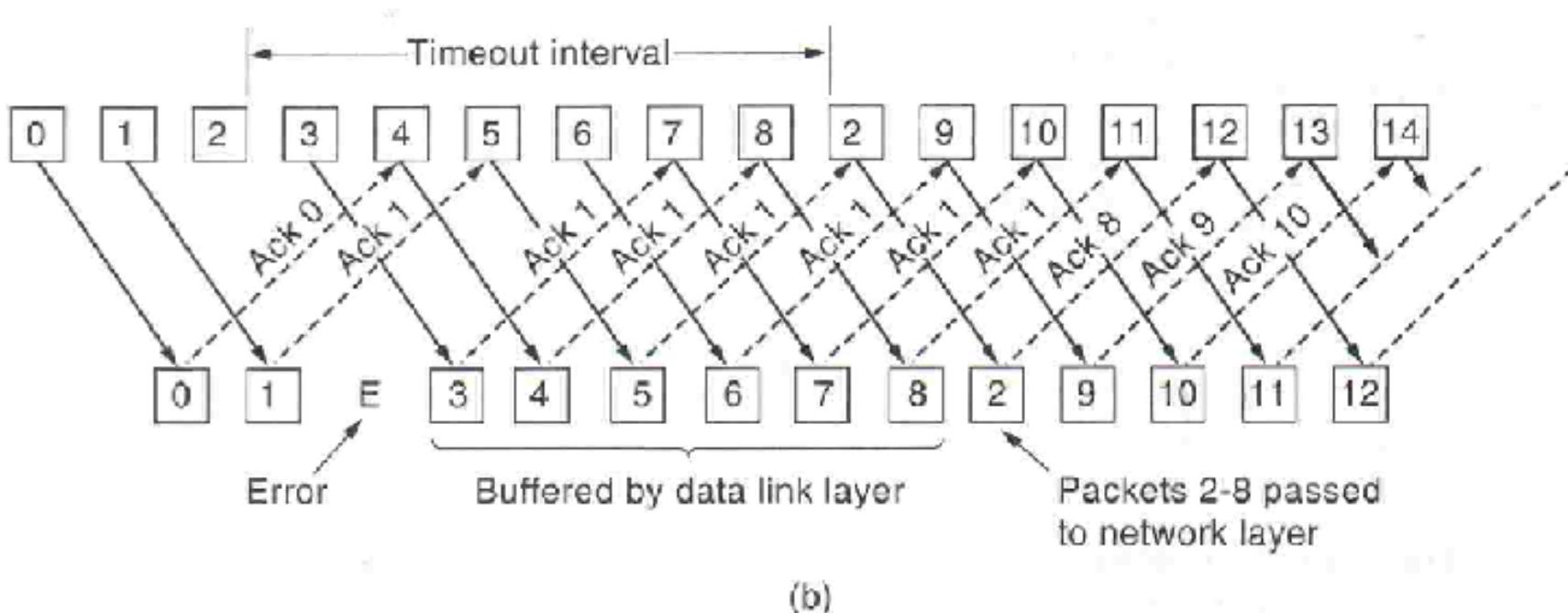
## Retroceso n continuación

- Tamaño de la ventana del receptor = 1
- El receptor descarta las tramas subsiguientes
- El canal de comunicación no se aprovecha si la tasa de errores es alta

# Repetición selectiva

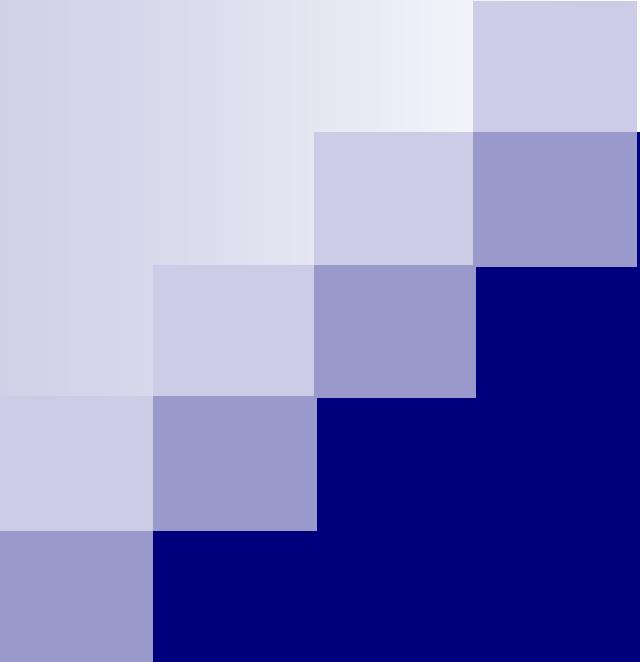
- Ventana del receptor  $> 1$
- Se descarta la trama dañada recibida
- Siguientes tramas recibidas correctamente se almacenan en buffer
- Emisor transmite sólo la última trama sin ACK

# Repetición selectiva



# Repetición selectiva

- El receptor podría enviar un NACK al detectar error para estimular la retransmisión antes de que expire el temporizador del emisor
- El emisor y receptor mantienen una ventana



## 3.5 EJEMPLOS DE PROTOCOLOS DE ENLACE DE DATOS

# Transmisión serie y paralelo

- Transmisión serie. Los bits se transmiten a través de un único camino
- Transmisión paralelo. Los bits se transmiten en forma simultánea a través de varios caminos
- En telemática se estudia solamente la transmisión serie



# Muestreo de la línea de transmisión

- En la transmisión de datos es importante la sincronización de los relojes del emisor y receptor
- El receptor debe saber la velocidad a la que se están transmitiendo los datos y en qué momentos inicia un bit
- Esto le permite muestrear la línea a intervalos constantes de tiempo para leer los bits
- Para ellos se utilizan dos técnicas de transmisión:
  - asíncrona y
  - síncrona

# Transmisión asíncrona

- La transmisión se realiza de carácter en carácter
- Cada carácter o byte se trata independientemente
- El primer bit de cada carácter alerta al receptor de la llegada del carácter
- El % de bits suplementarios para alertar al receptor y validar el byte es alto
- La validación de cada carácter se lo hace a través del bit de paridad
- La paridad puede ser par o impar

# Transmisión síncrona

- La transmisión se realiza en bloques de tramas
- El % de bits suplementarios para sincronizar y validar tramas es mucho menor
- Por tal razón la transmisión es más eficiente

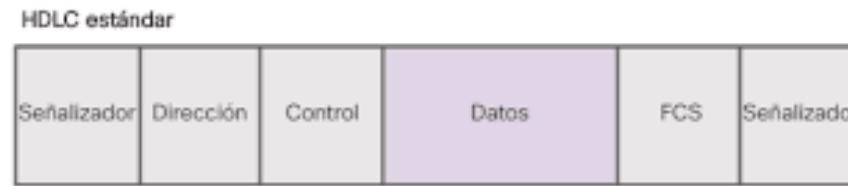
# Líneas síncronas y asíncronas

- Línea síncrona. Línea telefónica arrendada o dedicada. Siempre está a disposición
- Línea asíncrona. Línea telefónica básica o dial-up. La línea podría estar ocupada

- Las redes WAN utilizan líneas punto a punto
- Las líneas punto a punto pueden ser de cobre o de FO
- Dos de los protocolos de enlace de datos para líneas punto-a-punto y punto-multipunto son:
- HDLC High-level Data Link Control
- PPP Point-to-Point Protocol, que es una mejora de SLIP

# HDLC High-level Data Link Control

- Dentro de la carga útil usa relleno de bits (Sección 3.1.2 Entramado)  
Protocolo orientado a bit
- Se aplica en los datos insertando un 0 si hay 5 1s seguidos. Esto ayuda a la sincronización en la capa física
- Permite tramas con número fraccionario de bytes
- Usa transmisión con ventana deslizante y ACKs
- Cada trama empieza y termina con la bandera o señalizador 01111110
- La bandera permite además sincronizar los relojes de emisor y receptor



Sólo admite entornos de protocolo único.

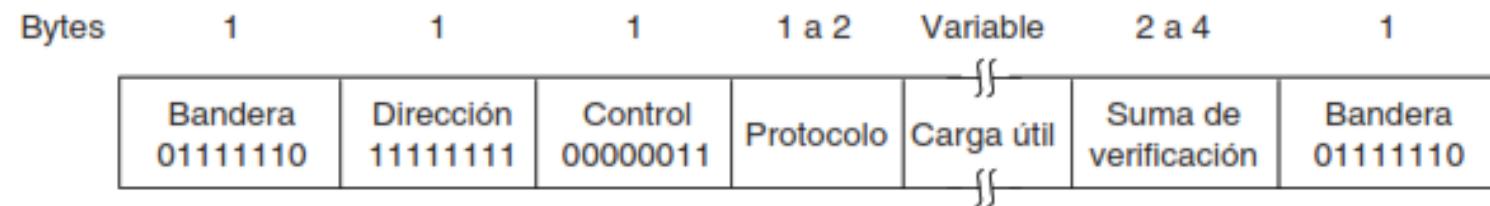


Usa un campo de datos de protocolo para admitir entornos multiprotocolo.

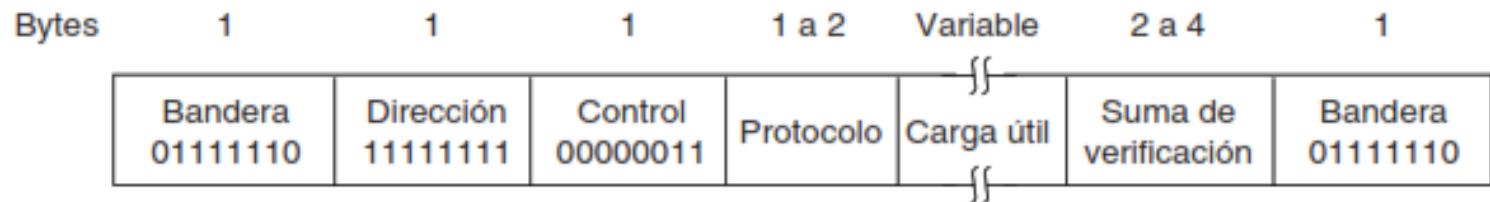
- El campo de dirección identifica a la estación que recibirá la trama
- Este campo solo se necesita en transmisiones punto-multipunto (computador-terminales)
- 11111111 es una dirección de broadcasting. En este caso, el campo dirección identifica a la estación que ha transmitido la trama

# PPP Point-to-Point Protocol

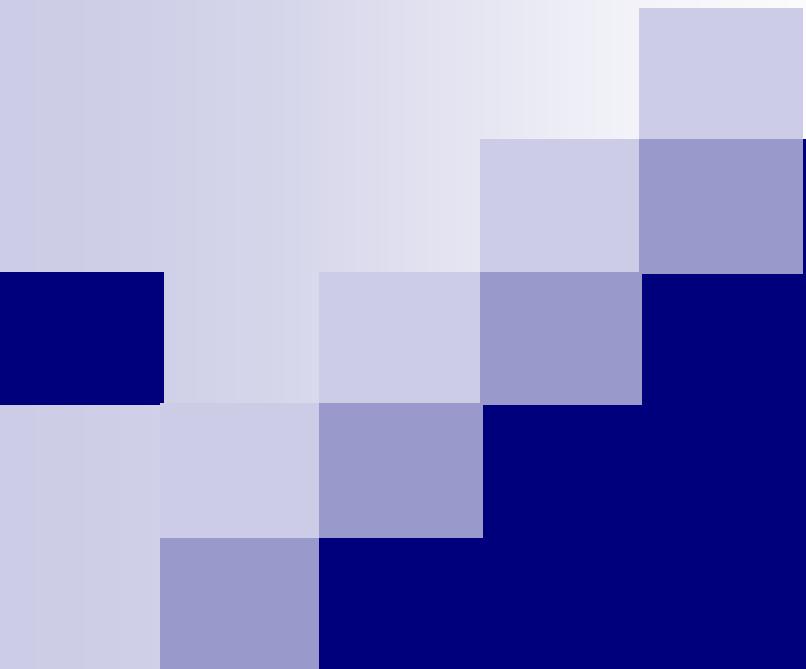
- Es una mejora de SLIP Serial Line Internet Protocol
- Está orientado a bytes
- Las tramas tienen un número entero de bytes
- Dentro de la carga útil utiliza relleno de caracteres



- Solo se necesita una bandera entre tramas
- Si no hay tramas para enviar se envían una secuencia de banderas
- Dirección 11111111 indica que todas las estaciones deben aceptar la trama



- Control 00000011 indica tramas de control y son no numeradas
- Protocolo: IPv4, IPv6, IPX (Netware Novell), AppleTalk
- Carga útil de longitud variable. Longitud predeterminada 1500 bytes
- Usa Checksum para verificar la integridad de los datos (no usa CRC; este se lo usa en capas superiores)
- PPP es usado con muchos tipos de capas físicas



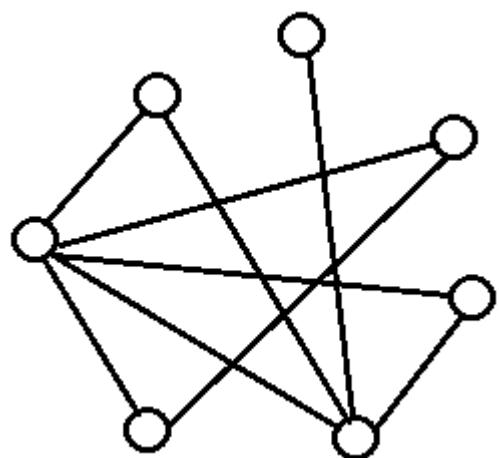
## 4. LA SUBCAPA DE CONTROL DE ACCESO AL MEDIO

# Contenido

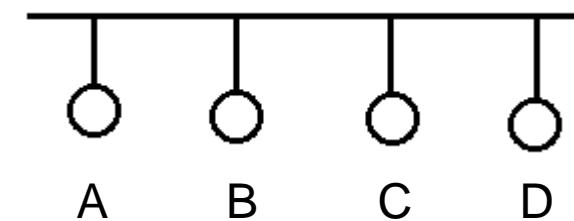
- 4.1 El problema de la asignación del canal
- 4.2 Protocolos de acceso múltiple: Aloha
- 4.3 Protocolos de acceso múltiple con detección de portadora
- 4.4 Protocolos libres de colisiones

# Canales de difusión

- Tipos de redes:
  - Conexiones punto a punto
  - Canales de difusión
- Un canal de difusión es un único medio de comunicación compartido entre múltiples usuarios



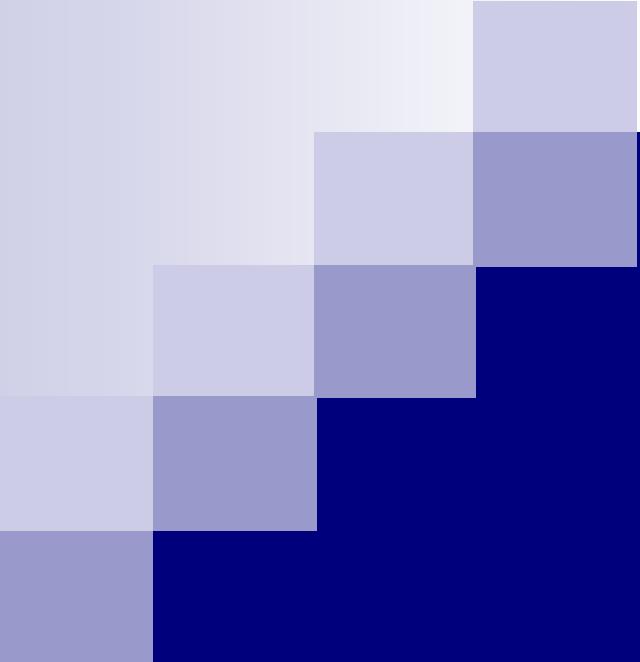
Red P2P



Red de difusión

- La clave es determinar quien utiliza el canal
- Hay muchas maneras de hacerlo: protocolos
- Los protocolos pertenecen a la subcapa MAC: Medium Access Control
- MAC: parte inferior de la capa 2

- LANs utilizan canales de difusión
  - MAC es fundamental en las LANs
- WANs usan canales Punto a Punto, excepto las redes satelitales



## 4.1 El problema de la asignación del canal



## ■ Dos tipos de asignación del canal:

- Estática
- Dinámica

# Asignación estática del canal

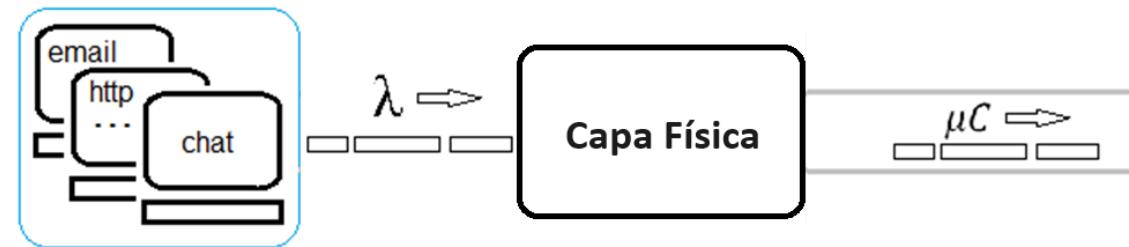
- Lo tradicional es FDM
- El ancho de banda se divide en N subcanales
- Es práctico para número pequeño y fijo de usuarios
- A cada usuario se le asigna un subcanal
- No hay interferencia entre usuarios

- No es práctico FDM cuando:
  - $n$  de usuarios grande y varía continuamente
  - $N$  es el número de subcanales
  - Tráfico en ráfagas: desborda el canal
- Si  $n < N$ , se desperdician canales
- Si  $n > N$ , a algunos de ellos se les negará el acceso

- El tráfico de las redes de computadores es en ráfagas
- Tráfico pico / tráfico promedio = 1.000
- La mayoría de los canales están desocupados casi todo el tiempo

## Asignación estática del canal. Desempeño de FDM. Teoría de colas

- T tiempo promedio que necesita la estación para despachar la cola: s./trama
- C capacidad del canal en bps
- $\lambda$  tasa de llegada de tramas a la cola: tramas/s
- $1/\mu$  tamaño promedio de las tramas: bits/trama
- Tasa de salida de tramas de la cola (tasa de servicio de *Poisson*)
- $Tasa\ de\ salida = \frac{capacidad\ del\ canal}{tamaño\ promedio\ de\ las\ tramas}$
- $Tasa\ de\ salida = \frac{C}{1/\mu} = \mu C \left[ \frac{bits/s}{bits/trama} \right] = \mu C \left[ \frac{tramas}{s} \right]$



- Tasa promedio de disminución de la cola:

$$\text{Tasa de salida} - \text{tasa de llegada} = \mu C - \lambda = [\text{tramas/s}]$$

- Tiempo para despachar la cola:

$$T = \frac{1}{\text{tasa promedio de disminución de la cola}} [\text{s/trama}], T = \frac{1}{\mu C - \lambda} [\text{s/tramas}]$$

- Si  $\lambda = 0$ ,  $T = \frac{1}{\mu C} = \frac{1/\mu}{C} = \frac{\text{tamaño promedio de las tramas}}{\text{capacidad del canal}}$
- Si  $\lambda = \mu C$ , la cola se mantiene del mismo tamaño
- Si  $\lambda > \mu C$ , la cola aumenta de tamaño
- En los dos últimos casos, nunca se termina de despachar la cola

## Ejemplo

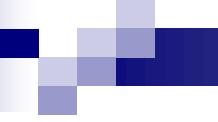
- C capacidad del canal = 100 Mbps
- $\lambda$  tasa de llegada = 5.000 tramas/s
- $1/\mu$  tamaño promedio de las tramas = 10.000 bits/trama
- $\mu C = \frac{C}{1/\mu} = \frac{100 \times 10^6 \text{ bps}}{10 \times 10^3 \text{ b/trama}} = 10.000 \text{ tramas/s}$
- Tiempo para despachar la cola  $T = \frac{1}{\mu C - \lambda} = \frac{1}{10.000 \frac{\text{tramas}}{\text{s}} - \frac{5.000 \text{ tramas}}{\text{s}}}$
- $T = 200 \mu\text{s/trama}$
- Con  $\lambda = 0$ :
- $T = \frac{1}{\mu C} = 100 \mu\text{s/trama}$

# División del canal en N subcanales: FDM



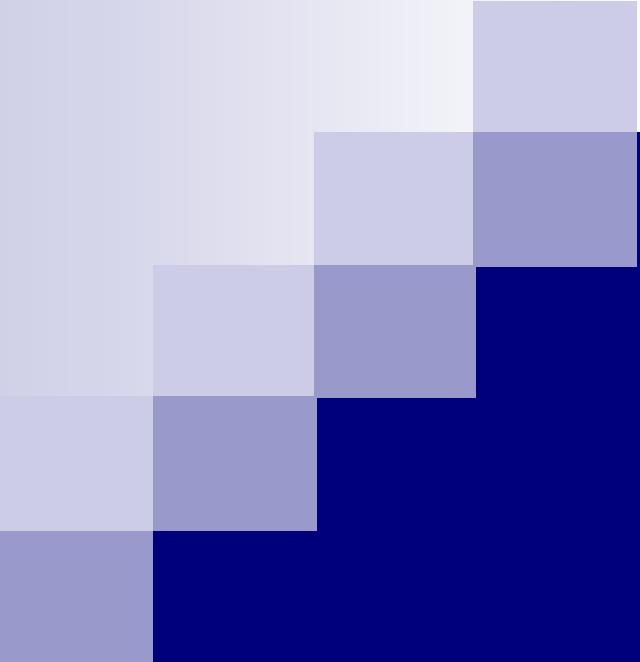
- Análogo al sistema telefónico
- Capacidad de cada subcanal =  $\frac{C}{N}$
- $\lambda$  tasa de llegada =  $\frac{\lambda}{N}$
- Tiempo para despachar la cola  $T_{FDM} = \frac{1}{\frac{\mu C}{N} - \frac{\lambda}{N}} = \frac{N}{\mu C - \lambda}$
- $T_{FDM} = NT$

- El retardo medio en FDM con  $N$  subcanales es  $N$  veces peor que un canal sin subcanales
- Lo mismo ocurre si reemplazamos una red de 100 Mbps por diez de 10 Mbps



## ■ Dos estrategias básicas de adquisición del canal:

- Métodos por contienda
- Métodos libres de colisión

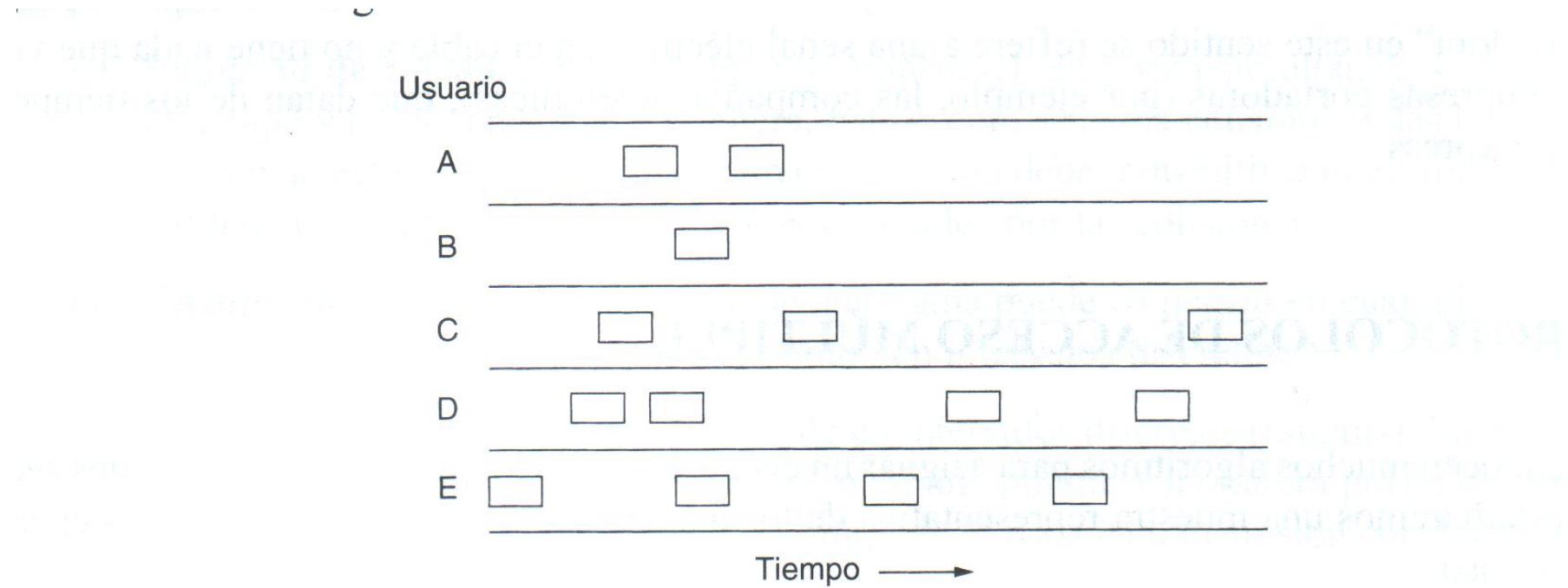


## 4.2 Protocolos de Acceso Múltiple: Aloha

- Asignación **dinámica** del canal
- Década de 1970
- Universidad de Hawaii
- Dos versiones: continuo y ranurado
- Ranurado divide el tiempo en ranuras discretas en las que cabe una trama
- Ranurado sincroniza el tiempo

# Aloha continuo o puro

- Los usuarios transmiten en cualquier momento que tengan datos para enviar
- Puede haber colisiones
- Tramas que colisionan se dañan
- La estación escucha el canal para saber si hay colisión **después** de transmitir la trama
- En una LAN la retroalimentación es **inmediata** (distancias cortas)
- Vía satélite hay un retardo promedio de  $\frac{1}{4}$  segundo
- Si la trama colisiona, el emisor espera un tiempo **aleatorio** y lo reenvía
- Estos sistemas se llaman de **contienda**



**Figura 4-1.** En ALOHA puro, las tramas se transmiten en momentos completamente arbitrarios.

## Aloha continuo

- Es más eficiente con tramas de **longitud uniforme**
- Si en una colisión sólo se daña un bit de una trama, hay que volver a transmitir

# Eficiencia del canal Aloha (1/7)

- ¿Qué fracción de tramas transmitidas no colisionan?
- ¿Cuál es la velocidad real o efectiva de transporte  $S$ ?
- $C$  capacidad del canal: bps
- Tiempo de trama  $T$ : tiempo para transmitir una trama de tamaño fijo
- $T = \frac{\text{tamaño de la trama}}{\text{velocidad de la luz}}$
- $T = \frac{L}{C} = \left[ \frac{\text{bits/trama}}{\text{bits/segundo}} \right] = [\text{segundos/trama}]$

## Eficiencia del canal Aloha (2/7)

- $N = \frac{\text{nuevas tramas generadas por los usuarios}}{\text{en 1 tiempo de trama}} = \frac{n}{1T}$
- Si  $N = 1$ , se genera  $n = 1$  trama nueva en  $1T$
- $1 = \frac{1}{T}$
- $T = 1 = \frac{L}{C}; \quad L = C$
- Es equivalente a generar una trama nueva de tamaño  $L = C$  en un segundo
- Esto equivale a que las estaciones generen, en conjunto, un total de  $C$  bits en un segundo (bps)
- Si  $N > 1$ , es como generar una trama de tamaño  $L > C$  en un segundo
- Lo deseable es  $0 < N < 1$

## Eficiencia del canal Aloha (3/7)

- Todas las estaciones en conjunto, y en promedio transmiten  $G$  tramas nuevas más la retransmisión de tramas colisionadas en el tiempo de trama  $T$
- $G = \frac{\text{transmisión de tramas nuevas} + \text{retransmisión de tramas colisionadas}}{\text{tiempo de trama}}$
- $G$  es la frecuencia promedio de la ocurrencias en un intervalo de tiempo
- Vemos que  $G \geq N$
- Con carga baja ( $N \approx 0$ ) hay pocas colisiones y retransmisiones:  $G \approx N$
- Con carga alta  $G > N$  hay muchas colisiones y retransmisiones

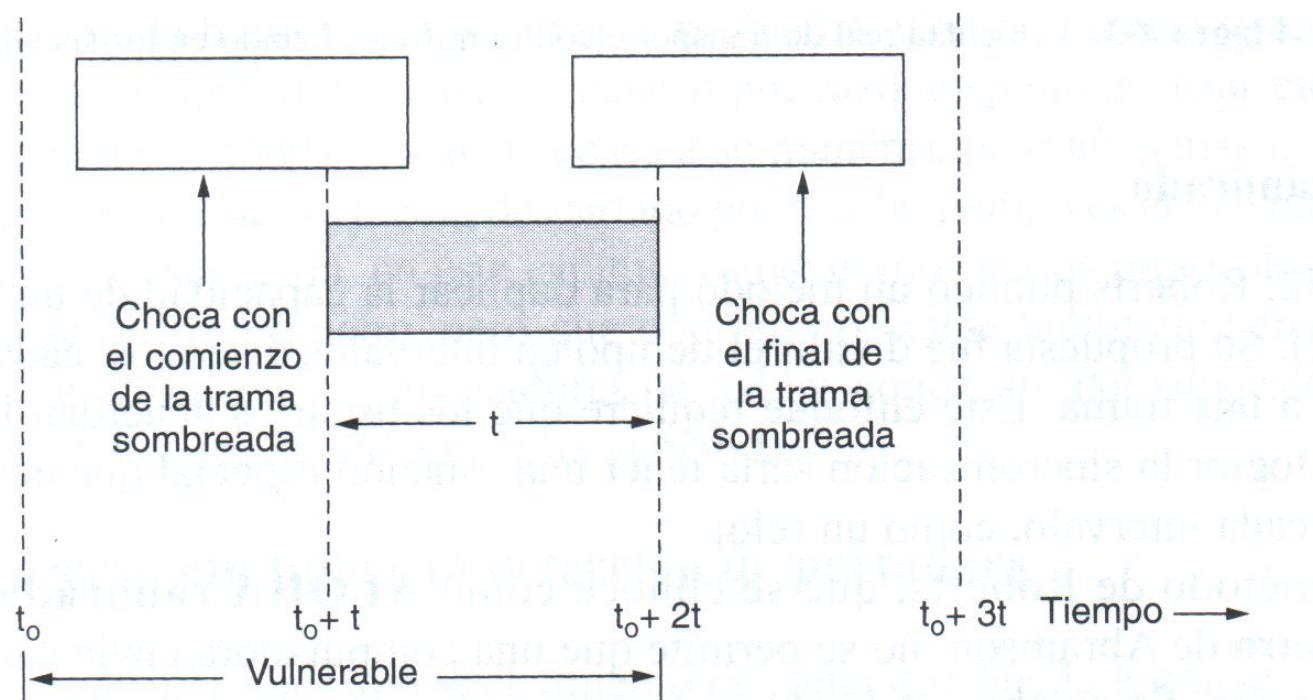
## Eficiencia del canal Aloha (4/7)

- Una trama no choca si no se envían otras tramas en el tiempo de trama
- $P_0$  = probabilidad de que la transmisión de una trama **no** colisione

$$P_0 \leq 1$$

- Velocidad efectiva de transporte es:

$$S = P_0 * G$$



**Figura 4-2.** Periodo vulnerable para la trama sombreada.

## Eficiencia del canal Aloha (5/7)

- Vulnerabilidad de trama =  $2 * (\text{tiempo de trama})$
- Probabilidad  $P(k)$  de que se generen  $k$  tramas en el tiempo de trama  $T$
- $P(k) = \frac{G^k e^{-G}}{k!}$ ; *distribución de probabilidad de Poisson*
- Esta es una distribución discreta que expresa, a partir de una frecuencia de ocurrencia promedio  $G$  (promedio de tramas transmitidas y retransmitidas en  $T$ ), la probabilidad de que ocurra un determinado número de eventos  $k$  (tramas generadas) en cierto intervalo de tiempo (tiempo de trama  $T$ )
- $k$  es una variable aleatoria, independiente de cualquier intervalo anterior
- El promedio de tramas generadas en el tiempo de vulnerabilidad es:  $2G$

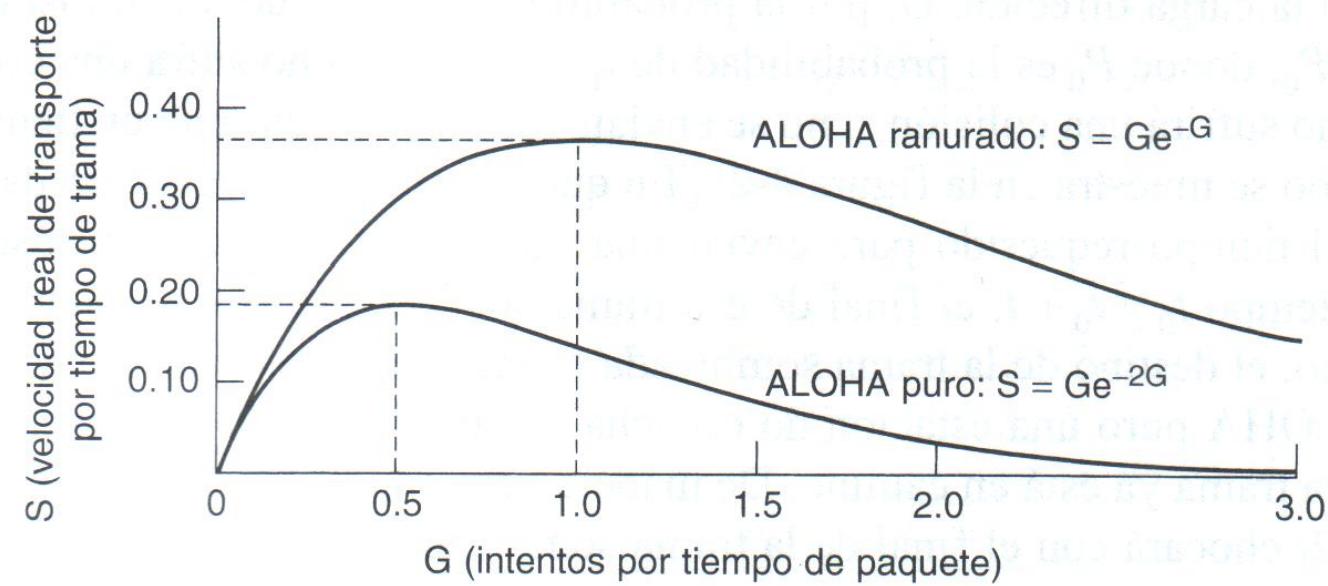
## Eficiencia del canal Aloha (6/7)

- Probabilidad de que no haya colisiones ocurre cuando no se generen tramas ( $k = 0$ ) en el **tiempo de trama** es
- $P(0) = \frac{G^0 e^{-G}}{0!} = e^{-G}$
- Probabilidad de que no haya colisiones en el **tiempo de vulnerabilidad** es:
- $P_0 = e^{-2G}$
- Si la velocidad efectiva de transporte es  $S = P_0 G$ , entonces
- $S = e^{-2G} G$
- Velocidad efectiva de transporte máxima:  $S_{\max} = ?$

## Eficiencia del canal Aloha (7/7)

- $S = e^{-2G} G$
- Velocidad máxima efectiva de transporte:  $S_{max}$
- $\frac{dS}{dG} = 0$
- $\frac{dS}{dG} = \frac{d}{dG}(e^{-2G} G) = e^{-2G} - 2Ge^{-2G} = (1 - 2G)e^{-2G} = 0$
- $1 - 2G = 0$
- $G = 0.5$

- Reemplazando  $G = 0.5$  en  $S = e^{-2G}G$
- $S_{max} = 0.5e^{-2(0.5)} = 0.184$
- La eficiencia máxima  $S_{max}$  de un canal Aloha puro es del **18,4%**



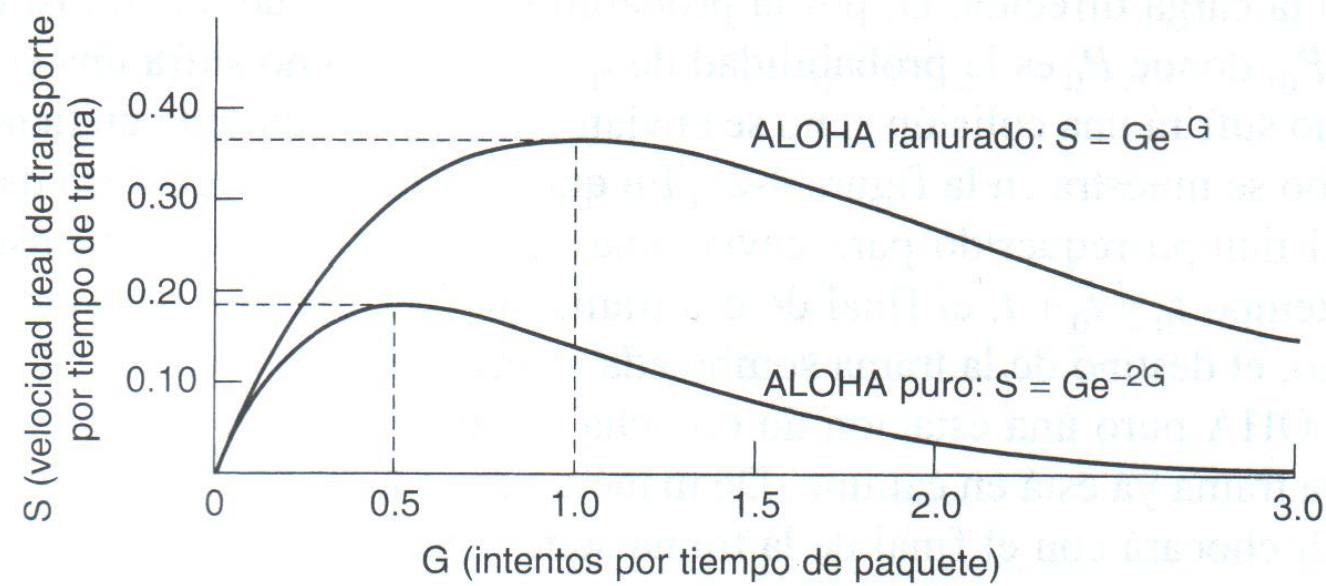
**Figura 4-3.** Velocidad real de transporte contra tráfico ofrecido en los sistemas ALOHA.

# Aloha ranurado

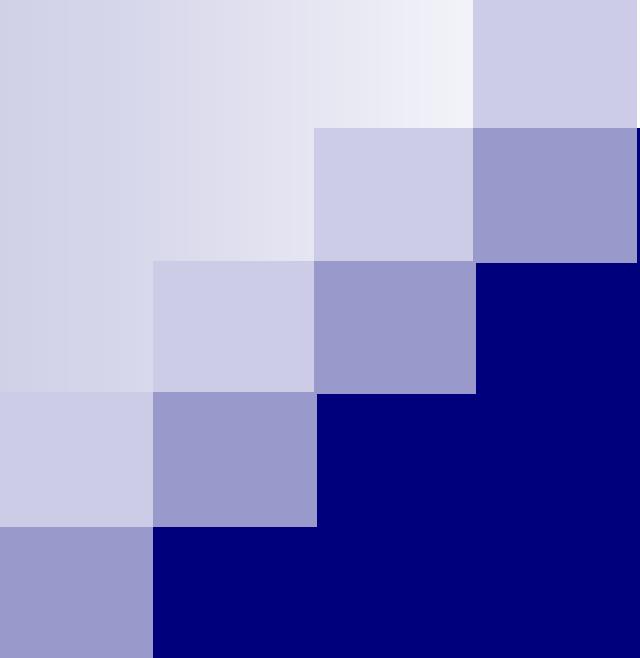
- No se envían tramas en cualquier momento
- Se divide el tiempo en intervalos discretos o ranuras
- Cada intervalo es igual al tiempo de trama
- Se espera el inicio de la siguiente ranura para enviar una trama
- Se necesita sincronizar los límites de cada ranura
- Para ello una estación central emite señales de reloj
- Se logra duplicar la capacidad del Aloha puro

# Aloha ranurado

- Período vulnerable = **1** tiempo de trama
- $G$  = número medio de tramas nuevas y retransmitidas en 1 tiempo de trama
- $P(k) = \frac{G^K e^{-G}}{k!}$  Distribución de probabilidad de Poisson
- Probabilidad de que la trama no colisione en el período vulnerable:  
 $P(k = 0) = P_0 = e^{-G}$
- Velocidad efectiva de transporte:  $S = P_0 G = G e^{-G}$
- $\frac{dS}{dG} = \frac{d}{dG}(G e^{-G}) = e^{-G} - G e^{-G} = (1 - G)e^{-G} = 0$
- $G = 1$
- $S_{max} = e^{-1} = 0.367; S_{max} = 36.7\%$



**Figura 4-3.** Velocidad real de transporte contra tráfico ofrecido en los sistemas ALOHA.



## 4.3 Protocolos de acceso múltiple con detección de portadora

- En LANs las estaciones detectan lo que hacen las otras y deciden transmitir sobre esta información
- Esto es «detección de portadora»: CS *carrier sense*
- Son más eficientes que Aloha *puro* (estaciones que transmiten en cualquier momento. No detectan lo que hacen las otras estaciones)
- Protocolos que detectan portadora antes de transmitir son:
  - CSMA persistente
  - CSMA no persistente
  - CSMA persistente-p
  - CSMA/CD

# CSMA persistente (1/4)

- CSMA Carrier Sense Multiple Access
- Una estación antes de transmitir **escucha el canal** para saber si otra está transmitiendo
  - Si es así, escucha persistentemente el canal hasta que esté desocupado
  - Si no, transmite inmediatamente una trama
- Si hay colisión, espera un tiempo aleatorio y vuelve a escuchar el canal

## CSMA persistente (2/4)

- El retardo de propagación del carrier es clave en el desempeño del protocolo
- A mayor retardo de propagación, peor desempeño del protocolo
- Si justo luego de que A empieza a transmitir, B escucha el canal
  - El carrier de A no habrá llegado aun a B
  - B no detecta ningún carrier
  - B transmite
  - Habrá una colisión

## CSMA persistente (3/4)

- Aun si el tiempo de propagación es cero habrá colisiones
  - Si dos estaciones que desean transmitir escuchan que el canal está desocupado: transmitirán
  - Se producirá una colisión

## CSMA persistente (4/4)

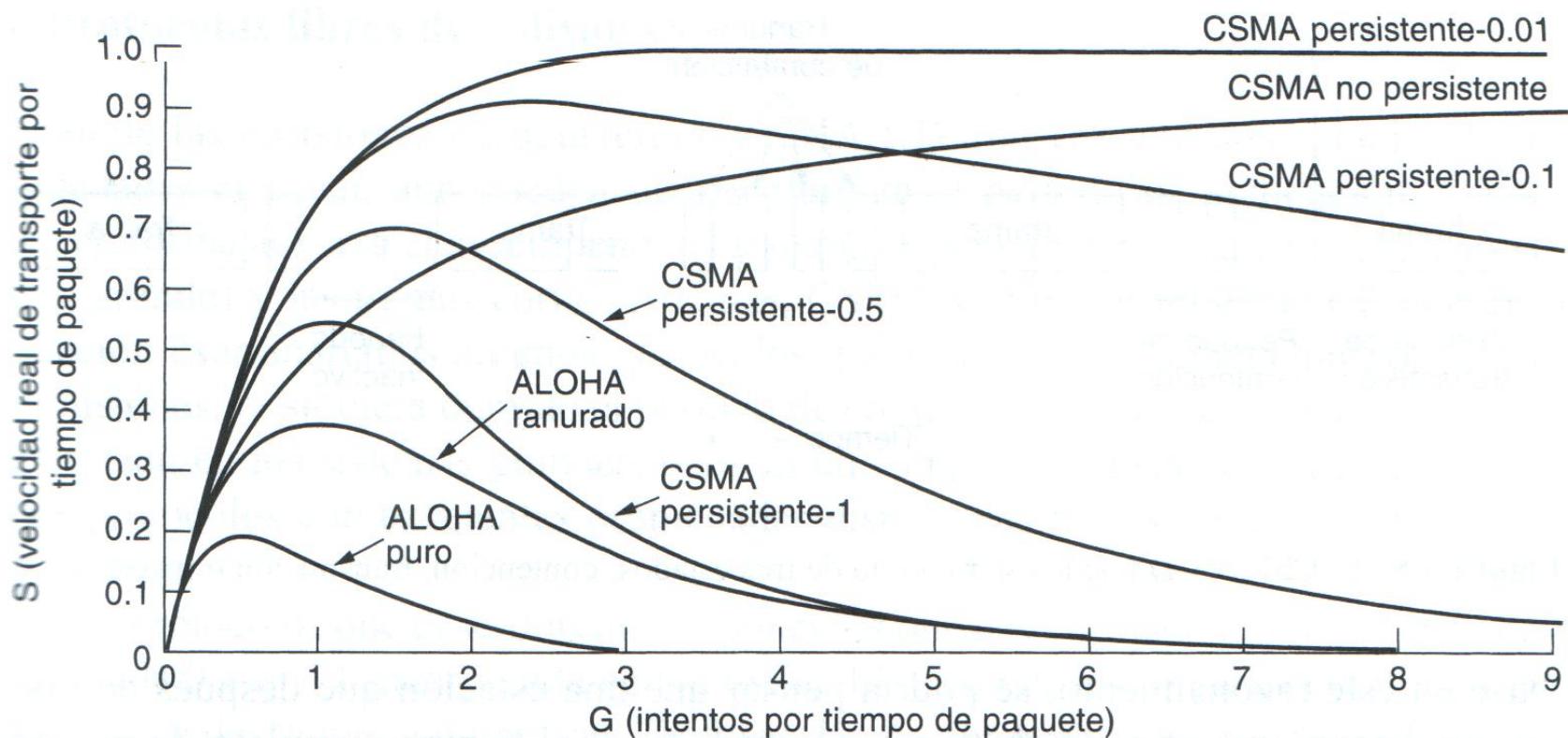
- **Persistencia:** Dos estaciones listas para transmitir cuando una tercera transmite, esperan hasta que ésta termine y entonces comienzan a transmitir simultáneamente en forma inmediata
- Hay una colisión
- Si no fueran tan impacientes habrían menos colisiones

## CSMA **no** persistente

- La estación escucha el canal, si este está desocupado la estación transmite
- Si el canal está en uso, la estación no lo escucha persistentemente
- Espera un tiempo aleatorio y vuelve a escuchar el canal
- Este algoritmo usa mejor el canal pero produce **mayores retardos** que CSMA persistente

# CSMA persistente-p

- Se aplica a canales ranurados
- Cuando una estación está lista para enviar, escucha el canal
- Si el canal está inactivo:
  - La estación transmite con probabilidad  $p$ ,
  - O espera a la siguiente ranura

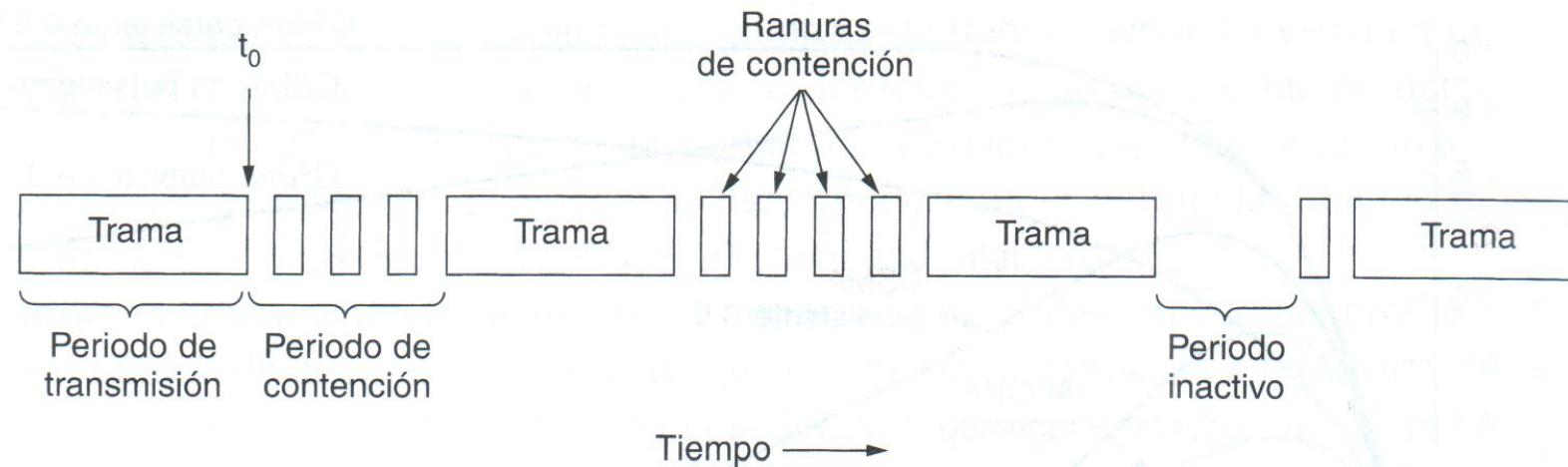


**Figura 4-4.** Comparación de la utilización del canal contra la carga para varios protocolos de acceso aleatorio.

## CSMA/CD (1/4)

- Carrier Sense Multiple Access / Collision Detection
- La estación escucha el canal antes de transmitir
- Las estación aborta su transmisión si detecta colisión: CD
- Se ahorra tiempo y ancho de banda
- Es la base de LAN Ethernet

## Transmisiones abortadas



**Figura 4-5.** El CSMA/CD puede estar en uno de tres estados: contención, transmisión o inactivo.

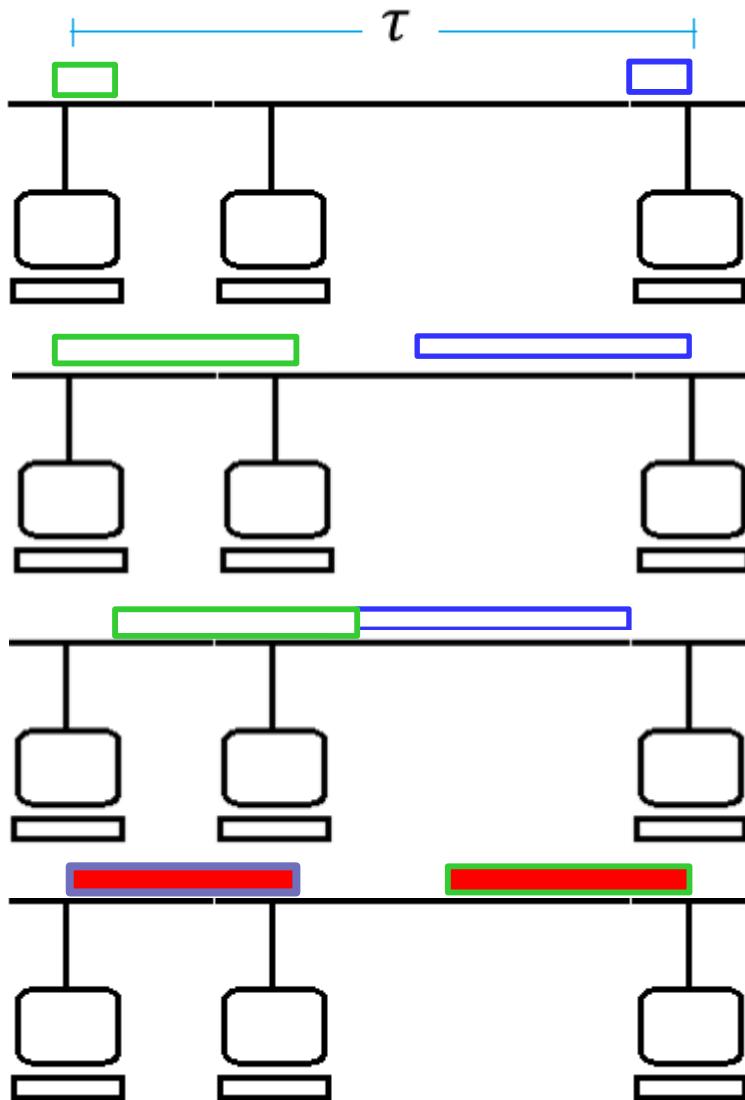
## CSMA/CD (2/4)

- En  $t_0$  la estación termina de transmitir su trama
- Cualquier estación con una trama por enviar, ahora puede intentar hacerlo
- Si dos estaciones deciden transmitir simultáneamente, hay colisión
- Las colisiones se detectan comparando la potencia de la señal enviada con la recibida

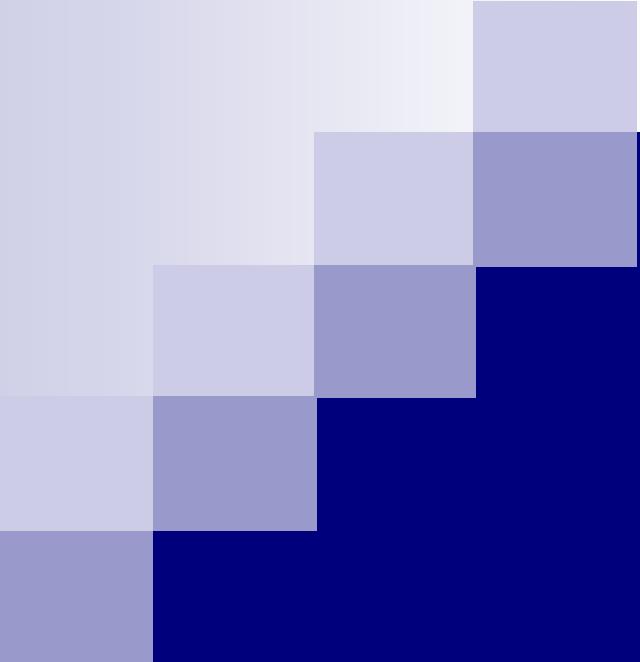
## CSMA/CD (3/4)

- La estación que detecta colisión, aborta la transmisión
- Espera un tiempo aleatorio
- Intenta de nuevo
- Hay períodos de:
  - Contienda
  - Transmisión
  - Inactividad

## Contienda en CSMA/CD (4/4)



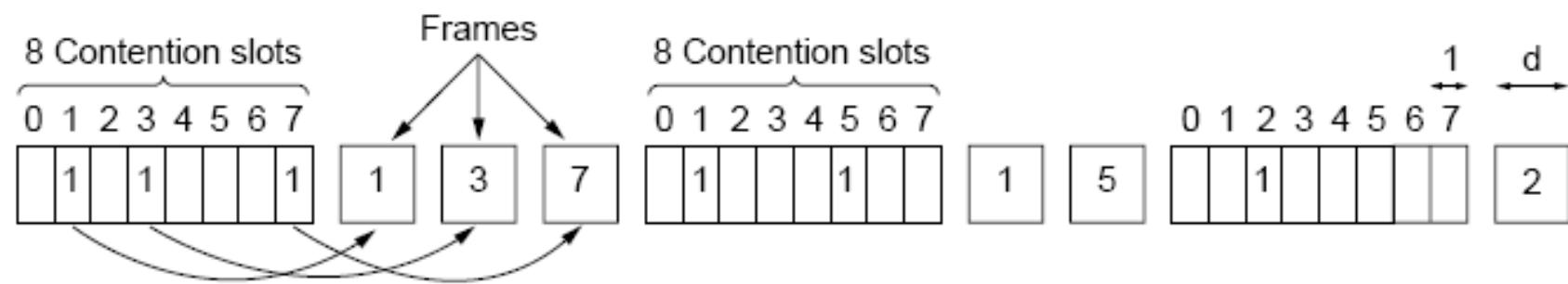
- Dos estaciones inician su transmisión en  $t_0$
- Una estación detecta la colisión en el tiempo promedio de propagación  $\tau$  de la señal
- $\tau$  es la clave para fijar el tiempo de contienda y la tasa de transporte
- El tiempo máximo de detección de colisión es  $2\tau$
- Entonces el tiempo de contienda debe ser  $2\tau$



## 4.4 Protocolos libres de colisiones

- Estos protocolos son:
  - Protocolo de mapa de bits
  - Conteo descendente binario
- N estaciones, cada una con una dirección única de 0 a N-1

# Un protocolo de mapa de bits



- Llamado también protocolo de reserva
- Si la estación  $j$  tiene una trama por enviar, transmite un bit 1 en la ranura  $j$
- Cada estación sabe cuáles estaciones quieren transmitir
- Entonces, las estaciones comienzan a transmitir en orden numérico

# Conteo descendente binario

- Se usan direcciones de estación binarias
- Las estaciones que van a transmitir difunden su dirección
- Se asume que los retardos son insignificantes
- La estación con dirección más alta tiene derecho a usar el canal

# Protocolos de contienda limitada

- Un criterio de desempeño de estos protocolos es el **retardo**
- Retardo en carga baja es distinto al retardo en carga alta

Protocolo	Rendimiento
De contienda	Mejor rendimiento con carga baja
Libres de colisión	Mejor rendimiento con carga alta
De contienda limitada	Contienda en cargas bajas Libre de colisiones en cargas altas

- Los protocolos de contienda estudiados son **simétricos**
- **Simetría**: las estaciones tienen la misma probabilidad  $p$  para transmitir
- Un sistema, podría mejorarse dando diferente  $p$  a cada estación

# Desempeño de los protocolos simétricos

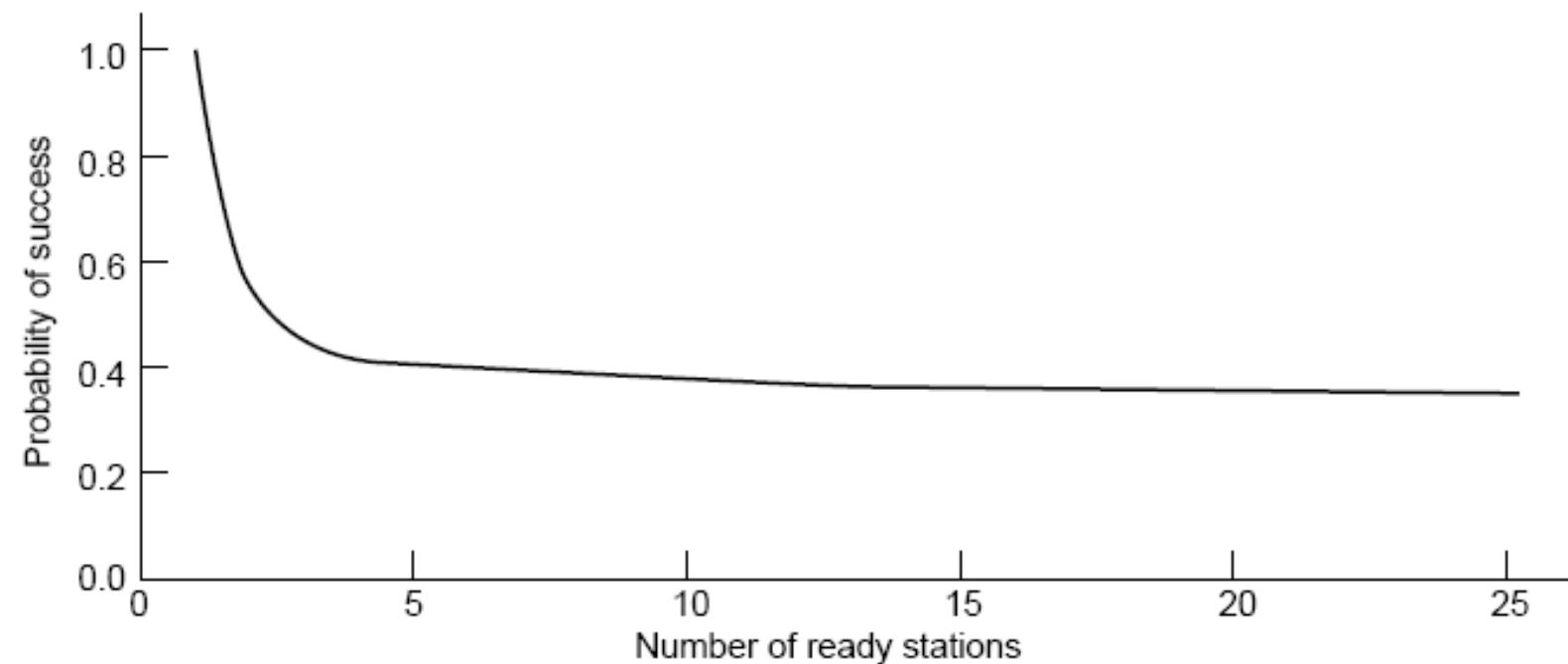
- $k$  = número de estaciones que compiten por acceder al canal
- Cada una tiene una probabilidad  $p$  de transmitir en una ranura
- La probabilidad de que alguna estación transmita **con éxito** en una ranura es:

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

- ¿Con qué valor de  $p$  se alcanza una probabilidad máxima de transmitir con éxito  $Pe$ ?

- El valor óptimo de  $P_e$  se obtiene derivando e igualando a cero
- $\frac{d}{dp}(P_e) = \frac{d}{dp} [kp(1 - p)^{k-1}] = 0$
- $p = 1/k$
- $P_e[\text{con } p \text{ óptimo}] = \left(1 - \frac{1}{k}\right)^{k-1}$

$\Pr[\text{éxito con } p \text{ óptimo}]$



- Para un número pequeño de estaciones la probabilidad de éxito es buena
- Pronto esta probabilidad cae a  $1/e = 0,36\dots$

# Protocolos de contienda limitada

- Se reúnen las estaciones en **grupos pequeños** para aumentar la probabilidad de éxito en la transmisión
- Estaciones del grupo 0 compiten en la ranura 0, estaciones del grupo 1 compiten en la ranura 1, así sucesivamente . . .
- Hay 2 extremos:
  - Todas las estaciones en un solo grupo: Aloha ranurado
  - 1 estación en cada grupo. Protocolo libre de colisiones

## Protocolo de recorrido de árbol adaptable

- Es una forma dinámica de asignar estaciones a una ranura
- Ideado por el ejército de US para hacer pruebas de sífilis a los soldados en la II Guerra Mundial
- Se toma una muestra de sangre de N soldados
- Se vacía una parte de cada muestra en un solo tubo de ensayo

- La muestra se examina

- Si no hay anticuerpos todos los soldados están sanos
  - Si hay anticuerpos se preparan dos muestras: una de los soldados de 1 a  $N/2$  y la otra muestra del resto de soldados

- El proceso se repite hasta dar con los soldados infectados

Nivel

0

1

2

1

2

3

4

5

6

7

A

C

D

E

F

G

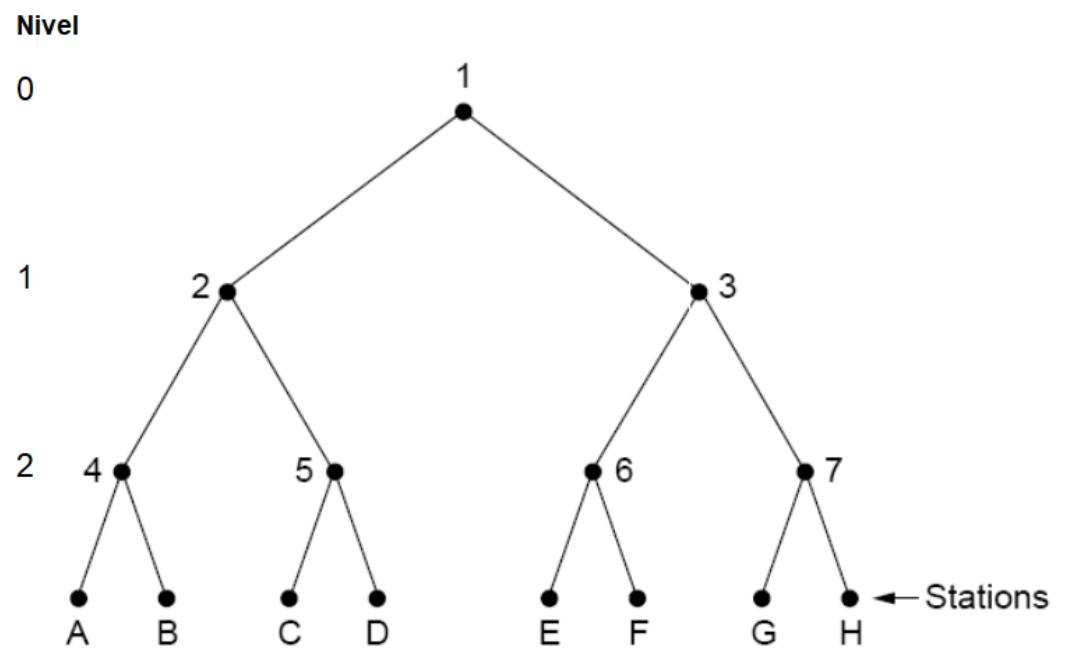
H

Stations

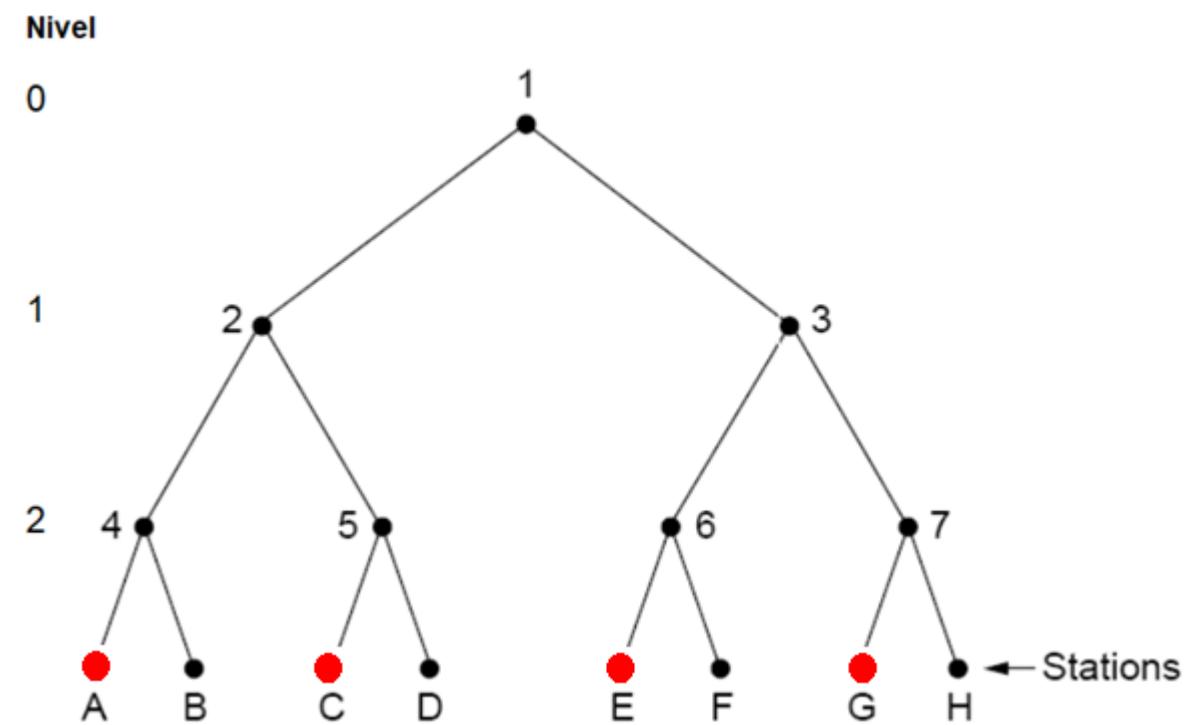
Fig. 4-9. The tree for eight stations.

- Las estaciones son hojas de un árbol binario
- Todas las estaciones pueden competir para conseguir el canal en la ranura 1
- Si hay colisión, entonces sólo las estaciones que están debajo del nodo 2 pueden competir por el canal en la primera ranura
- Si una estación consigue el canal, la ranura 2 se reserva para las estaciones que están debajo de nodo 3
- Si ocurre una colisión, se examina todo el árbol con los hijos izquierdo y derecho en forma recursiva para localizar las estaciones listas
- Si vuelve a ocurrir una colisión, continúa la búsqueda

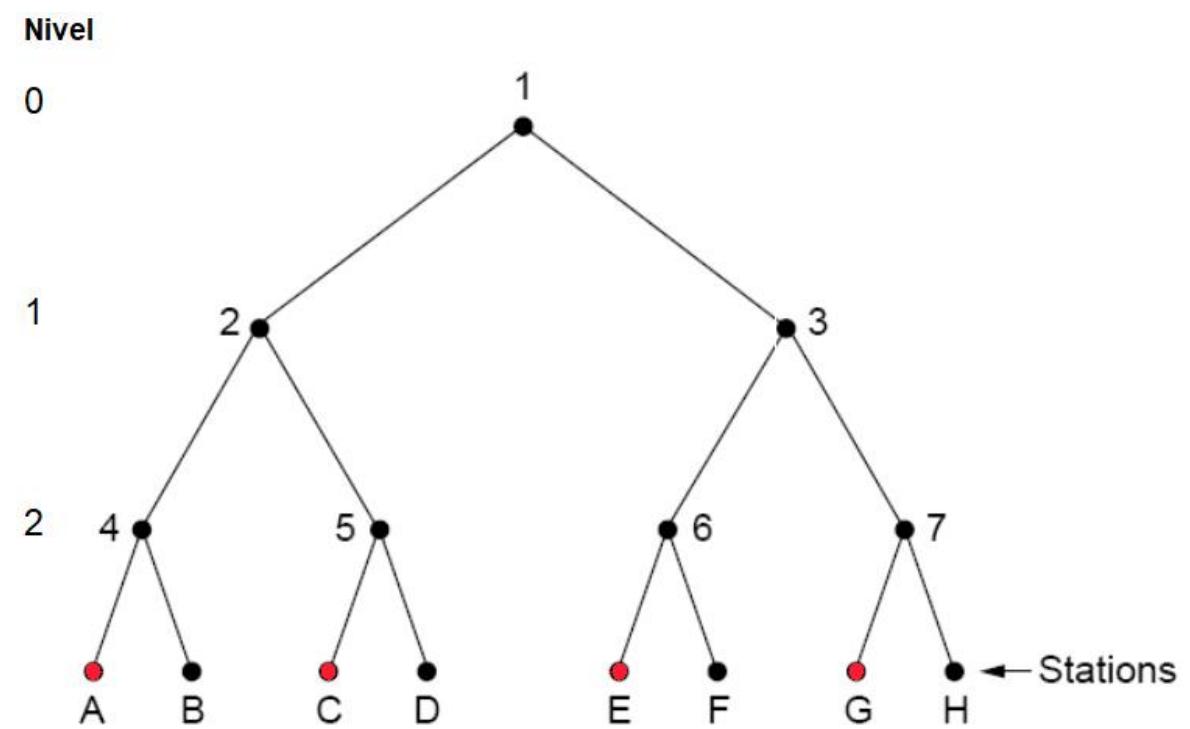
- A más carga, se inicia la búsqueda de la estación a transmitir desde un nivel  $i$  más abajo
- Un nodo en nivel  $i$  tiene una fracción  $2^{-i}$  estaciones por debajo de él, expresado en por unidad
- El nivel 0 tiene  $2^{-0} = 1$  estaciones
- El nivel 1 tiene  $2^{-1} = 0.5$  estaciones
- El nivel 2 tiene  $2^{-2} = 0.25$  estaciones



- $q$  es el número de estaciones que desean transmitir distribuidas uniformemente en el árbol
- Un nodo de nivel  $i$  tiene una fracción de  $q$  que es  $2^{-i}q$
- Si  $q = 4$
- El nodo de nivel 0 tiene  $2^{-0}4 = 4$  estaciones que desean transmitir
- Un nodo de nivel 1 tiene  $2^{-1}4 = 2$  estaciones que desean transmitir, etc.

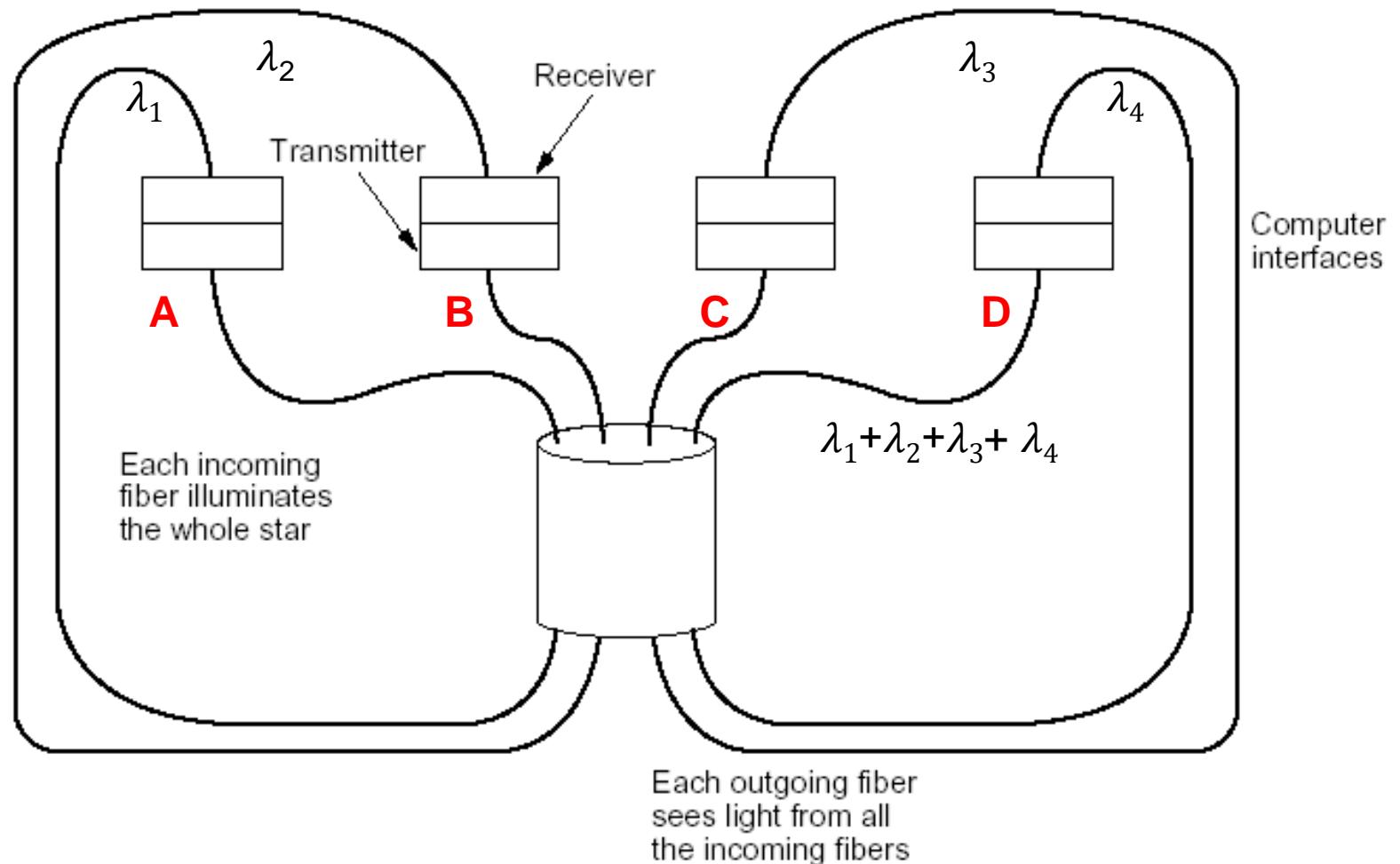


- El nivel óptimo para comenzar a analizar el árbol es aquel cuyo número de estaciones que desean transmitir sea 1
- $2^{-i}q = 1$
- $\frac{1}{2^i}q = 1$
- $q = 2^i$
- $i = \log_2 q$
- Si  $q = 4$ ,  $i = 2$



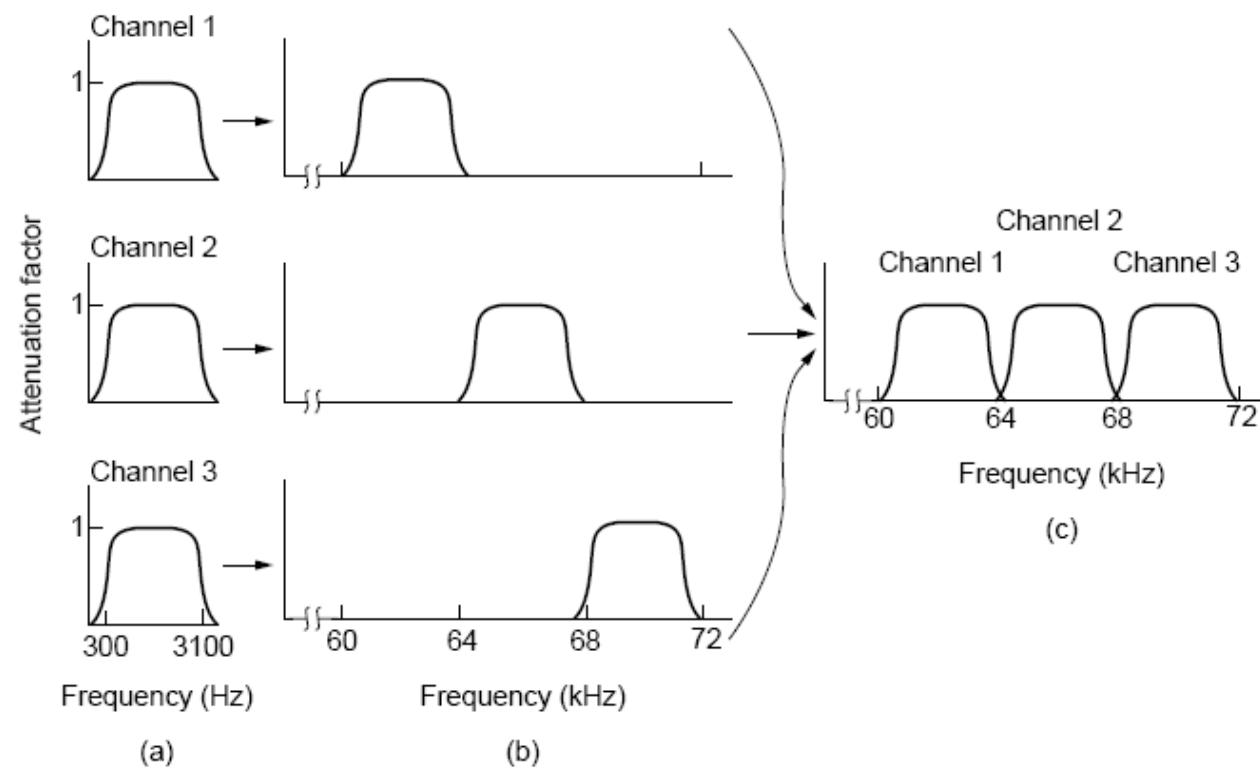
# Protocolos de acceso múltiple por división de longitud de onda

- El canal se divide en sub-canales con FDM y/o TDM
- Se permite muchas conversaciones simultaneas
- Esquema usado en LANs de fibra óptica
- Se fusionan dos fibras de cada estación a un cilindro de vidrio



# Multiplexión

- Se divide el espectro en bandas de longitud de onda



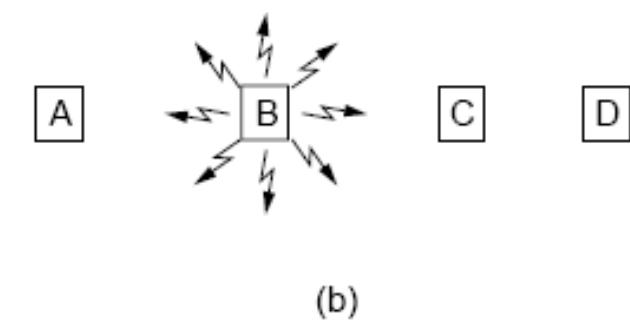
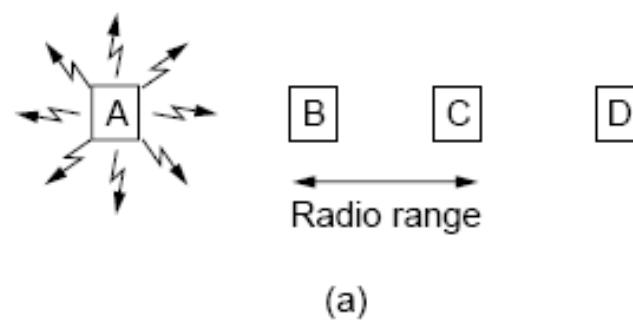
- Con WDMA Wave-length Division Multiple Access, se asignan 2 canales a cada estación
- La estación A solo recibe información en  $\lambda_1$ , B en  $\lambda_2$ , etc.
- Si la estación A desea enviar información a C, usa  $\lambda_3$
- Hay un canal estrecho para control, y un canal ancho para el envío de datos
- El canal de control se usa para solicitar y recibir solicitudes de conexión

# Protocolos de LANs inalámbricas

- Laptops requieren de un lugar fijo para trabajar. Ellas son **portátiles, no móviles**
- Un sistema móvil implica que puede trabajar mientras se desplaza
- La **movilidad** necesita de un sistema de comunicación **inalámbrico**
- Una WLAN requieren de estaciones base o puntos de acceso colocados adecuadamente
- Las estaciones base se interconectan con cobre o fibra

- La potencia de la señal de APs y portátiles es baja
- Así se evita que una WLAN interfiera a otra WLAN
- El alcance es de pocas decenas de metros
- Cada oficina es una **celda** y el edificio un sistema celular
- A diferencia de telefonía celular, cada celda sólo tiene **un canal** que cubre todo el ancho de banda disponible
- 11 a 54 Mbps

- WLANs requieren protocolos MAC especiales
- El problema en WLAN es la limitación de alcance
- Un enfoque equivocado es usar CSMA: oír si hay transmisiones y transmitir si nadie más lo hace
- Pero lo que importa es la **interferencia** en el **receptor** no en el emisor



# Problema de estación oculta

- En el gráfico, no importa cuáles son AP ni cuáles son portátiles
- A puede alcanzar a B pero no a C
- B puede alcanzar a A y C pero no a D
- Si A transmite a B, C no podrá escuchar a A por problema de limitación de alcance
- C deducirá falsamente que puede transmitir a B
- Si C transmite se **colisionará** en B con A

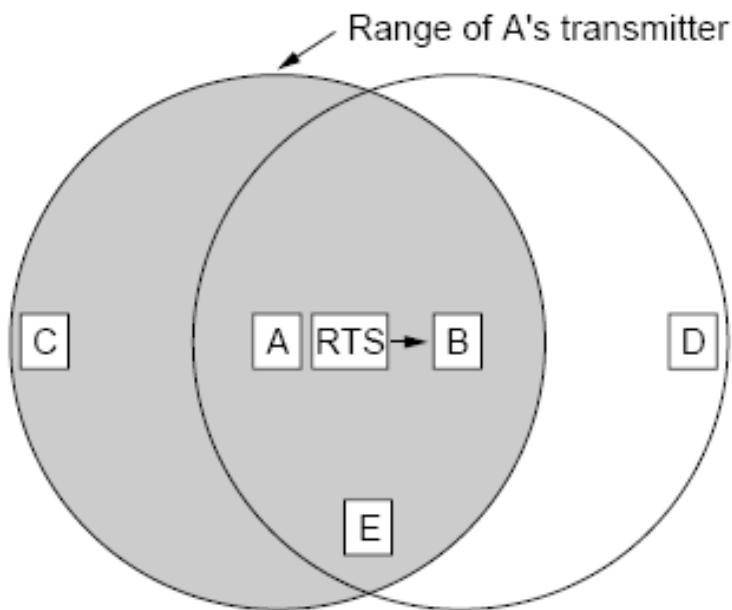
# Problema de estación expuesta

- Ahora B transmite a A
- C escucha la transmisión
- C concluye erradamente que no puede transmitir a D (?!)
- *Pero la trama de C que se envía a D colisiona con la trama de B que se envía a A, porque B sí alcanza a C (RMOG). No hay tal “problema de estación expuesta”*

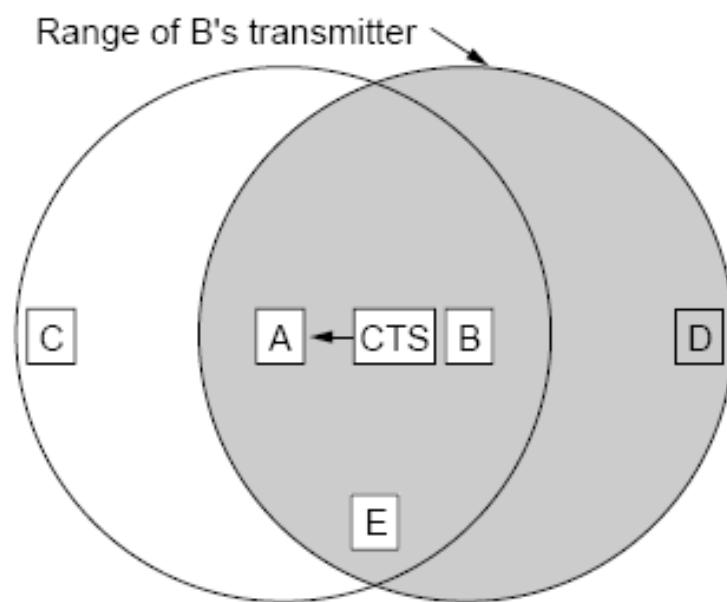
- Antes de transmitir, una estación necesita saber si hay portadora **alrededor del receptor**
- CSMA indica si hay actividad alrededor de la estación que detecta la portadora
- Con cable, las señales se propagan y llegan a todas las estaciones
- En sistemas de radio de corto alcance **se pueden aceptar transmisiones simultaneas**

# MACA

- Multiple Access with Carrier Avoid. 1990
- Acceso Múltiple con Prevención de Portadora
- Uno de los primeros protocolos hechos para WLAN
- El emisor pide al receptor enviar una trama corta, para que las estaciones cercanas al **receptor** no transmitan



(a)



(b)

1. A envía una trama corta de 30 bytes RTS (Request to send) a B.
2. RTS contiene la longitud de la trama de datos que se enviará luego
3. B contesta con CTS (Clear to Send) con la longitud de la trama que recibirá de A
4. A inicia la transmisión a B

- Las estaciones que escuchan RTS enviada por A permanecen en silencio hasta que CTS se transmite hacia A
- Las estaciones que escuchan CTS enviada por B permanecen en silencio hasta que la trama de datos se transmita hacia B
- Pero aún pueden ocurrir colisiones. A y D pueden enviar RTS a B al mismo tiempo

# MACAW

- MACA Wireless
- Es una mejora a MACA (Multiple Access with Carrier Avoid)
- Se introduce la trama ACK tras una trama exitosa
- Se agregó la **detección de portadora** para evitar que una estación transmita un RTS al mismo tiempo que otra

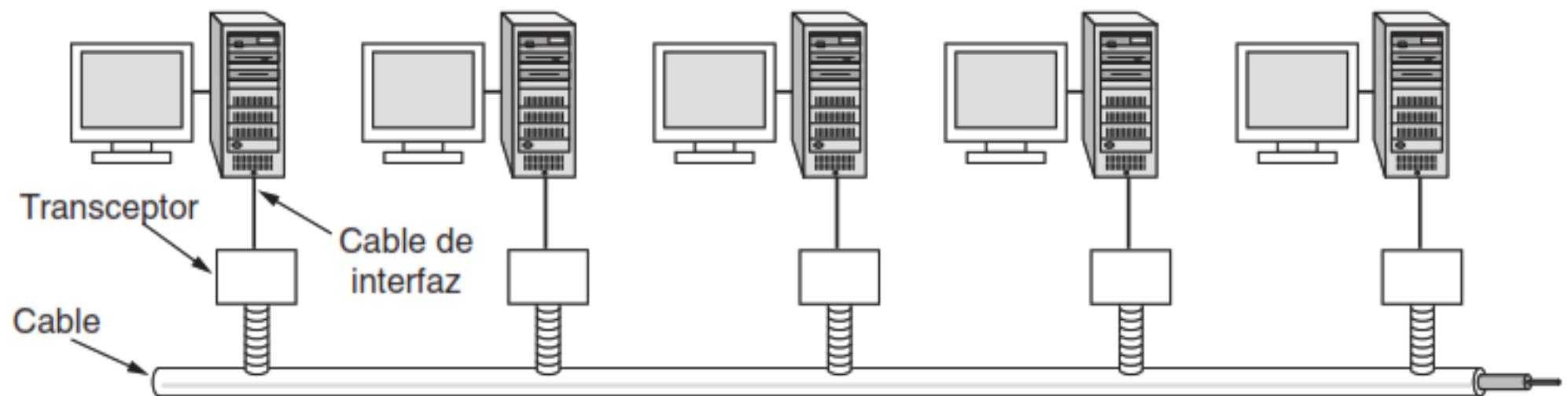
## 4.5 Ethernet

- Existen dos tipos muy diferentes de Ethernet: Ethernet clásica y Ethernet commutada.
- Ethernet clásica resuelve el problema del acceso múltiple con el uso de técnicas que hemos estudiado. Opera hasta a 10 Mbps. Ya no se utiliza
- Ethernet commutada que utiliza switches para conectar varias computadoras. Opera a 100 Mbps, 1 Gbps, 10 Gbps
- 100 Mbps Fast Ethernet
- 1 Gbps Gigabit Ethernet
- 10 Gbps 10 Gigabit Ethernet
- Ethernet se recoge en la norma IEEE 802.3

# Capa física de Ethernet clásica

- La primera LAN fue construida por Bob Metcalf y David Boggs en 1976 en la empresa Xerox PARC (Ciudad de Palo Alto - California)
- Utilizaba un cable coaxial grueso
- Operaba inicialmente a 3 Mbps y luego a 10 Mbps
- Usaba la codificación Manchester
- Antes se pensaba que el espacio estaba lleno de *ether* por medio del cual se propagaban las ondas electromagnéticas
- En 1983 se convirtió en el estándar IEEE 802.3





# Protocolo de la subcapa MAC de Ethernet clásica



Figura 4-14. Formatos de trama. (a) Ethernet (DIX). (b) IEEE 802.3.

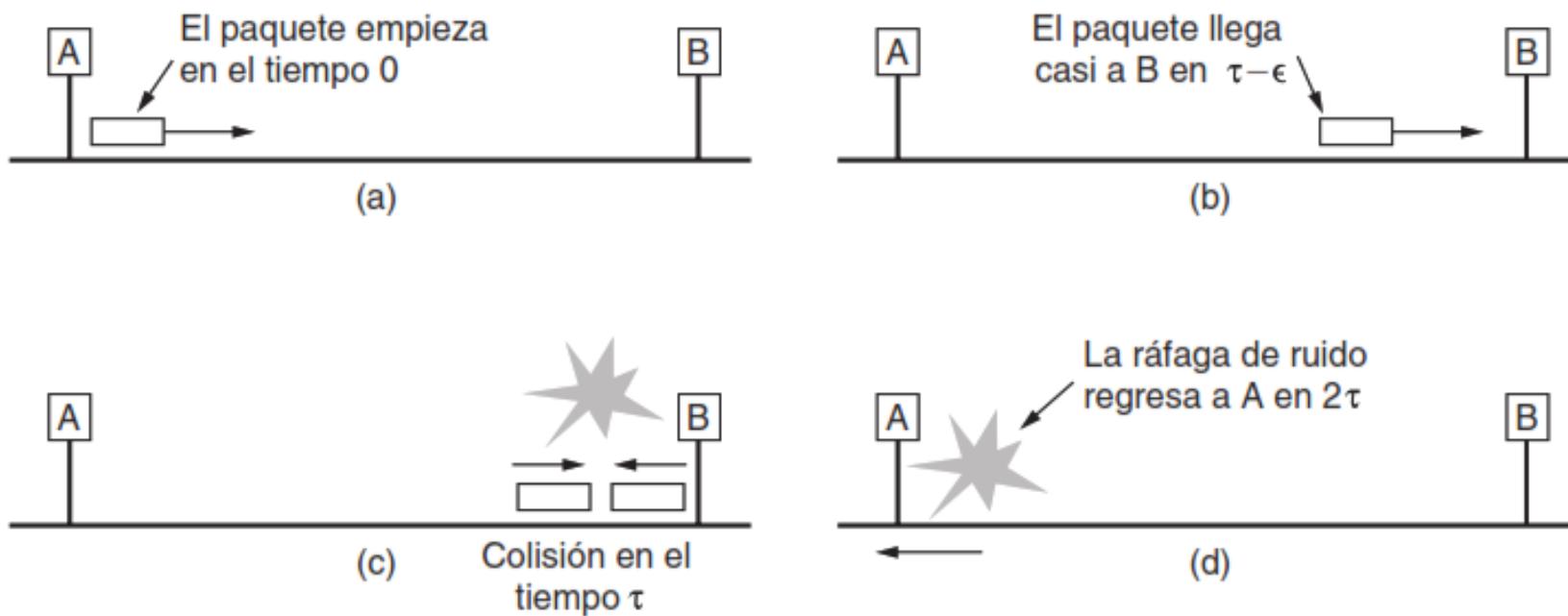
- Cada byte del Preámbulo contiene el patrón 10101010
- Los dos últimos bits del último byte del Preámbulo se establecen en 11 para indicar el inicio de la trama SOF
- El primer bit de la dirección destino es:
  - 0 para direcciones ordinarias
  - 1 para direcciones de grupo: Multidifusión (multicast)

- La dirección que consiste únicamente de bits 1 es para difusión (broadcast)
- Las direcciones origen (48 bits) son únicas a nivel mundial
- Los primeros 3 bytes de la dirección NIC Network Interface Card identifican a su fabricante
- En el 3er. Campo, todo valor mayor a 1536 identifica el ***tipo***: protocolo de cada de red (ejemplo IPv4)
- Todo valor menor o igual a 1536 indica ***longitud*** de los datos
- Después están los datos de hasta 1500 bytes, límite establecido por la cantidad de RAM del transceptor de aquella época (año 1978)

- Si la trama no transporta datos, el campo de datos es de 0 bytes
- La trama debe tener una longitud mínima de 64 bytes considerando desde el campo destino hasta el campo de suma de verificación
- Para ello se incluye el campo de **relleno** de 46 bytes para alcanzar el tamaño mínimo



- La razón más importante de tener una trama mínima es evitar que el emisor termine de transmitir la trama antes de que el primer bit llegue a la estación más lejana
- Un emisor sabe que se produjo una colisión cuando escucha el ruido de la colisión mientras está transmitiendo la trama
- Si el emisor no escuchó ruido mientras transmitió la trama, este supondrá que la transmisión fue exitosa
- Pero podría ocurrir que luego de terminar la transmisión haya una colisión
- Para evitar esto, las tramas deberán tardar más de  $2\tau$  en transmitirse
- $2\tau$ : tiempo de propagación de ida y vuelta para el peor caso en el cable



**Figura 4-15.** La detección de una colisión puede tardar hasta  $2\tau$ .

- El campo ***suma de verificación*** utiliza el polinomio generador ya estudiado

# CSMA/CD

- Ethernet clásica usa CSMA/CD persistente-1
- Si hay una colisión la estación aborta la transmisión y vuelven a transmitir luego de un intervalo aleatorio
- El modelo sigue siendo el de la figura 4.5

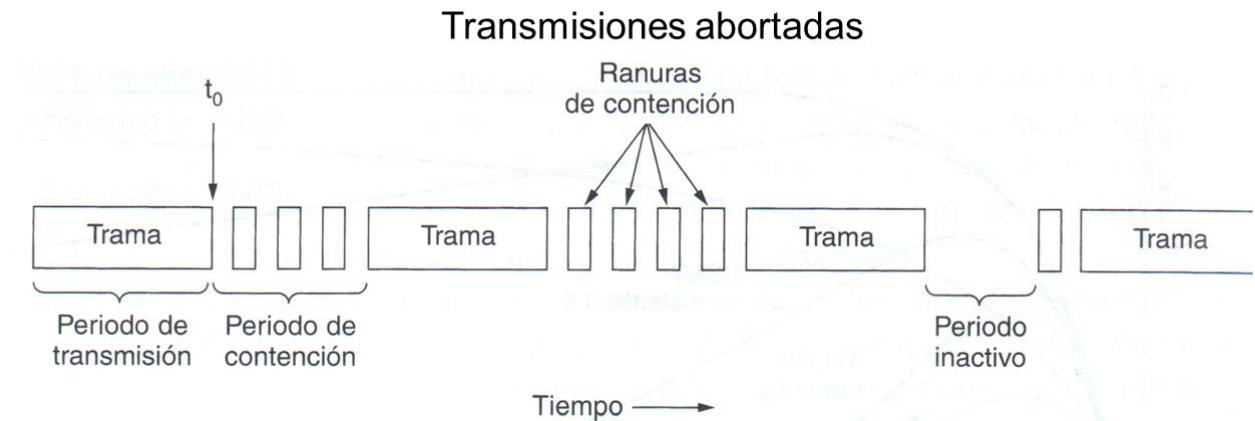
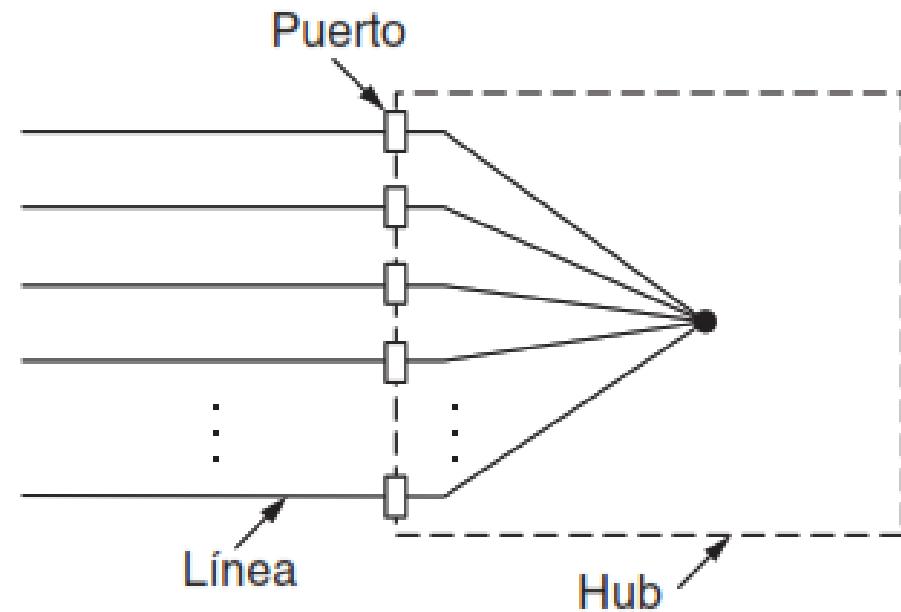


Figura 4-5. El CSMA/CD puede estar en uno de tres estados: contención, transmisión o inactivo.

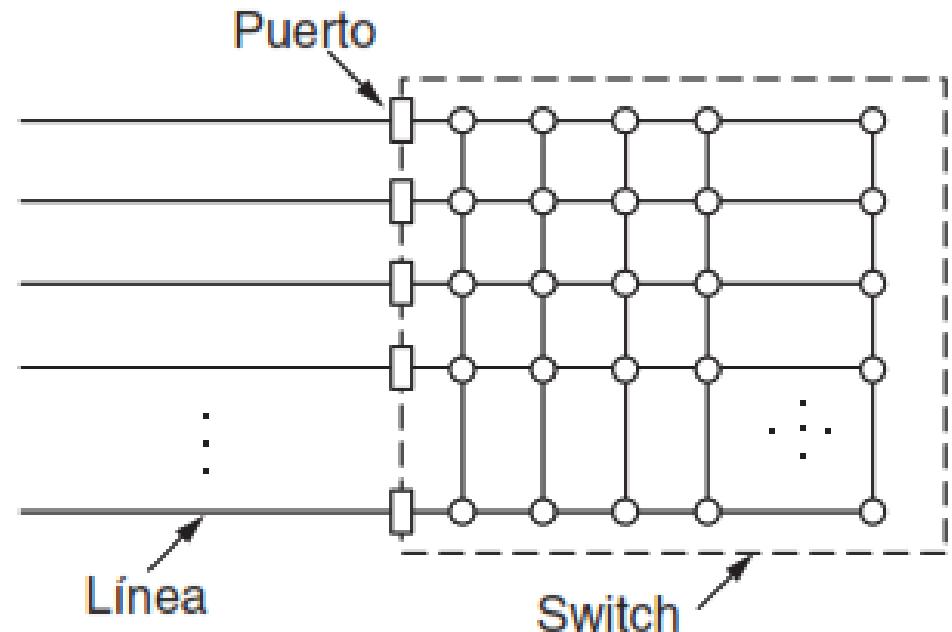
- Tras la colisión, el tiempo se divide en ranuras de  $2\tau$
- El algoritmo a seguir se llama **retroceso exponencial binario**
- Después de la primera colisión, las estaciones esperan 0 o 1 ranura al azar antes de intentarlo de nuevo
- Si vuelven a colisionar, cada una escoge entre 0, 1, 2 o 3 ranuras al azar
- De haber una tercera colisión, el número de ranuras a esperar es desde 0 hasta  $2^3 - 1$
- Después de  $i$  colisiones se elige entre 0 y  $2^i - 1$  ranuras
- CSMA/CD ni Ethernet proveen confirmaciones de recepción ya que el canal tiene tasas de error bajas
- Si ocurre un error se detecta con CRC y se recupera en las capas superiores
- En canales inalámbricos en donde hay más errores se usan ACKs

# Ethernet commutada

- Ethernet dejó de usar un solo cable extenso por problemas en las conexiones
- Ahora cada estación tiene un cable dedicado que llega a un hub
- Un hub conecta eléctricamente todos los cables como si estuvieran soldados
- En esta configuración es más fácil agregar o quitar una estación
- También las cables rotos afectan a una estación y se pueden detectar con facilidad



- Pero los hubs no incrementan la capacidad debido a que son lógicamente iguales a un solo cable extenso de antes
- A medida que se agrega una nueva estación, se tiene que compartir la capacidad de transmisión con un número mayor de estaciones
- Entonces surgió Ethernet commutada
- El corazón del sistema es un switch
- Un switch se ve como un hub



- Hubs y switches cuentan con puertos para conectores RJ-45 para cable de par trenzado
- Con un switch también es fácil añadir o quitar una estación
- Un switch solo envía la trama a los puertos para los cuales están destinadas
- Los demás puertos ni siquiera saben de que existe la trama
- Para ello, el switch verifica la dirección Ethernet de la estación destino
- Al interno el switch opera a muchos Gbps con un protocolo propietario oculto
- Si más de una estación desea trasmitir tramas al mismo tiempo, incluso a un mismo destino, no habrá colisión porque el switch encola las tramas
- Además, una estación puede transmitir y recibir tramas al mismo tiempo porque el cable es full-duplex

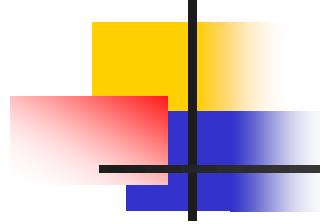
# Fast Ethernet

- Con el pasar de los años, 10 Mbps se hizo insuficiente
- Hubo la necesidad de crear una versión de Ethernet mucho más rápida
- La nueva versión se recogió en la norma IEEE 802.3u
- Totalmente compatible con Ethernet clásica IEEE 802.3
- Básicamente lo que se hizo fue reducir los tiempo de bit de 100 nseg. a 10 nseg.
- Así que la velocidad inicial se multiplicó por 10, alcanzando 100 Mbps
- Se reemplazó el cable coaxial por el cable UTP Cat 3 existente ya en los edificios, más hubs y switches



## 5. LA CAPA DE RED

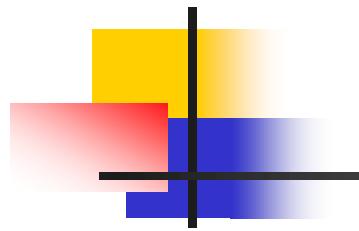
Al final del capítulo se tomará la prueba 2 sobre los capítulos  
4 y 5



# Contenido

---

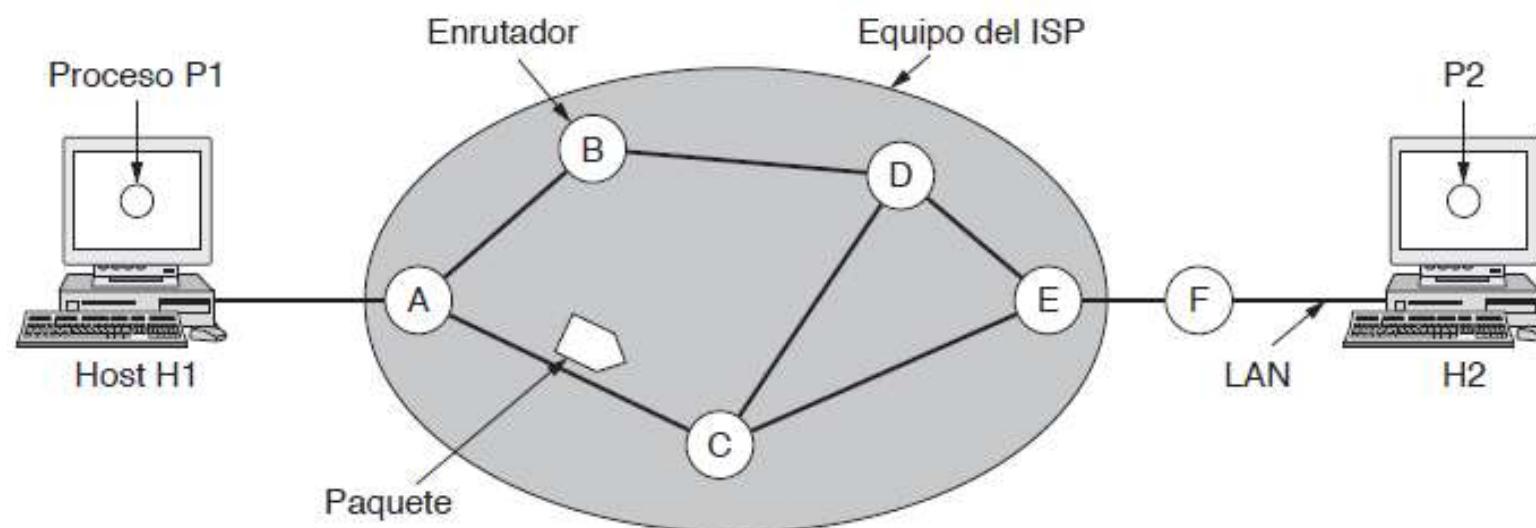
- 5.1 Aspectos de diseño de la capa de red
- 5.2 Algoritmos de enrutamiento
- 5.3 Algoritmos de control de congestión
- 5.4 Calidad de servicio QoS
- 5.5 Interconexión de redes
- 5.6 La capa de red de Internet

- 
- La Capa 2 solo lleva tramas del un extremo al otro extremo del enlace P2P
  - La Capa 3 lleva paquetes del origen al destino, aún si están en redes diferentes (extremo a extremo)
  - Esto podría requerir atravesar varios enrutadores
  - La capa 3 debe conocer: la topología y el tráfico en la subred
    - La **topología** permite conocer las distintas rutas para alcanzar el destino
    - Conocer el **tráfico** permite no sobrecargar unas rutas y subutilizar otras



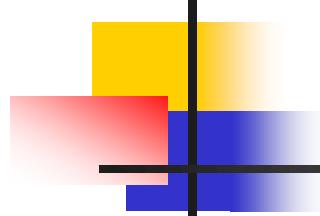
## 5.1 Aspectos de diseño de la capa de red

## 5.1.1 Conmutación de paquetes, almacenamiento y reenvío



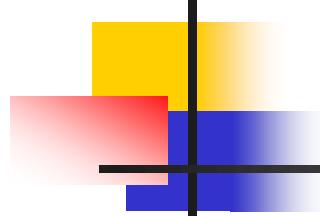
**Figura 5-1.** El entorno de los protocolos de la capa de red.

- H1 se conecta punto a punto a **A** de la empresa portadora, con un modem
- La LAN, se conecta a una red portadora a través del ruteador F y una línea alquilada
- F es similar a los routers de la portadora
- En un ruteador de la subred, el paquete se almacena un momento hasta que llegue por completo, se compruebe errores y haya línea de salida disponible
- El paquete se reenvía al siguiente enrutador hasta alcanzar el destino



## 5.1.2 Servicios proporcionados a la capa de transporte

- La capa de red da un esquema de direccionamiento uniforme a LANs y WANs
- Los servicios que da la capa 3 son:
- Servicio orientado a la conexión, y
- Servicio sin conexión



## 5.1.3 Implementación del servicio **sin** conexión

- Los paquetes se colocan en la subred y se enrutan de manera independiente
- Los paquetes se llaman **datagramas**
- Un mensaje a enviar podría ser dividido en varios paquetes
- Cada enrutador tiene una tabla de enrutamiento a cada destino final
- La información de una tabla de enrutamiento puede variar dinámicamente
- El algoritmo que maneja las tablas se llama **algoritmo de enrutamiento**

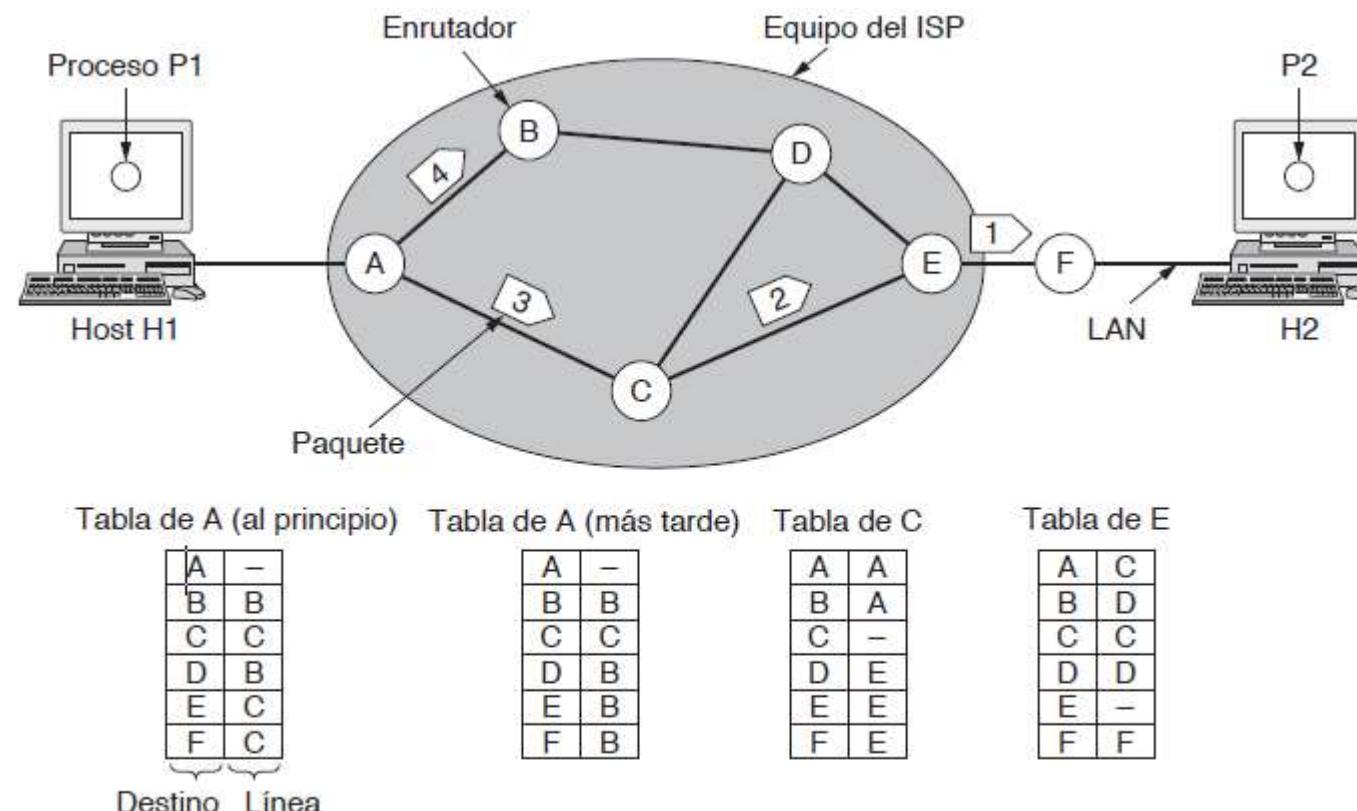
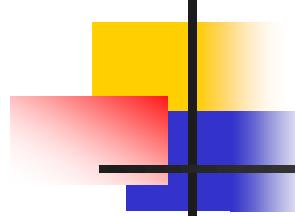


Figura 5-2. Enrutamiento dentro de una red de datagramas.



## 5.1.4 Implementación del servicio orientado a conexión

- Antes de enviar paquetes **se fija un camino** entre ruteadores finales
- Este se llama **circuito virtual**, como si fuera un circuito físico exclusivo
- El propósito es no tener que buscar una ruta para cada paquete a enviar

Circuito	Recursos físicos de la subred
Físico	Dedicados, ej: llamada telefónica convencional
Virtual	Compartidos con otros circuitos de otros usuario

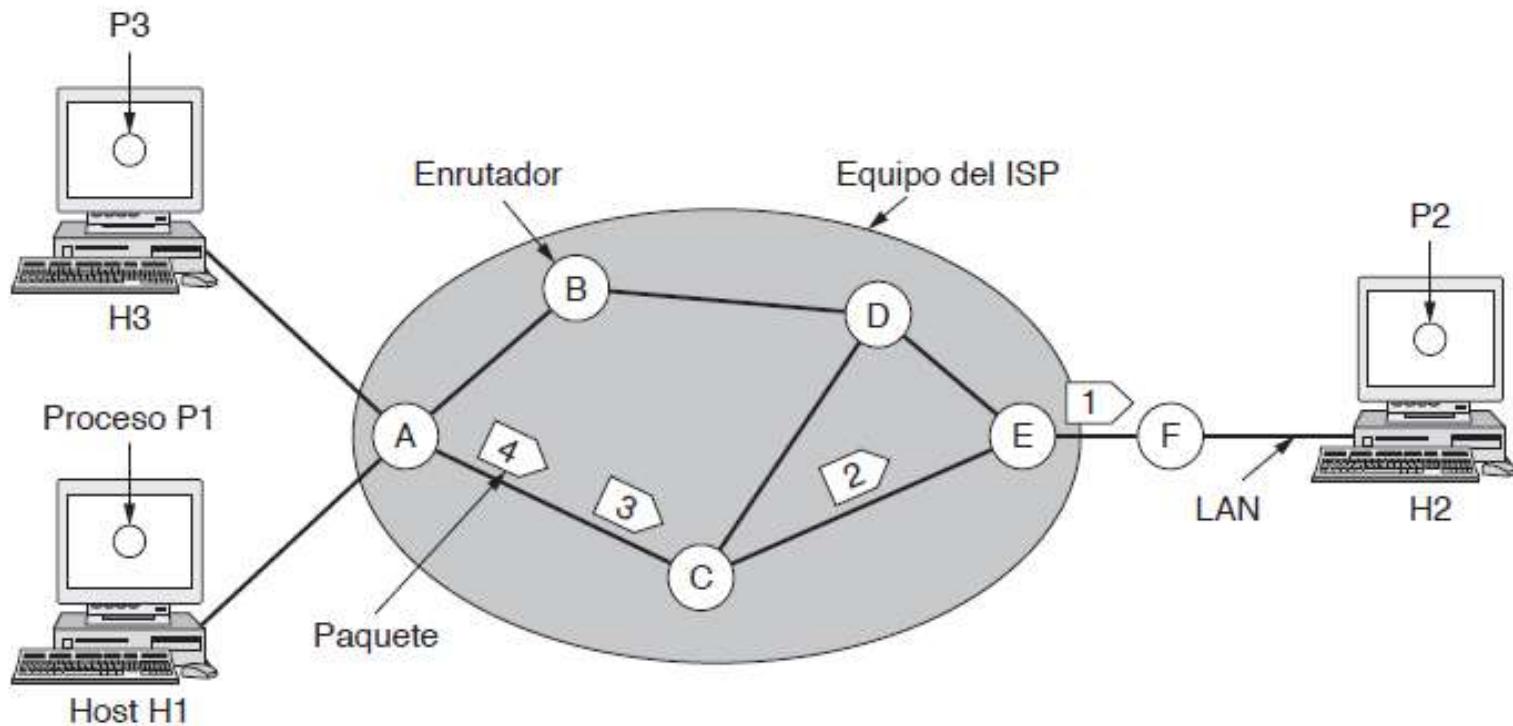


Tabla de A	
H1   1	C   1
H3   1	C   2
Dentro      Fuera	

Tabla de C	
A   1	E   1
A   2	E   2

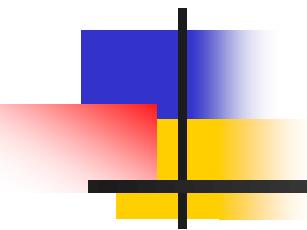
Tabla de E	
C   1	F   1
C   2	F   2

Figura 5-3. Enrutamiento dentro de una red de circuitos virtuales.

- Tabla del router A:
- La primera fila indica que el CV 1 viene de H1, se envía a C con el número CV 1
- La segunda fila indica que el CV 1 viene de H3, se envía a C con el número CV 2
- Todo el tráfico fluye por la misma ruta o conexión
- Cada paquete lleva un ID que indica a cuál circuito virtual pertenece

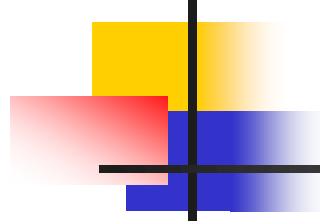
## 5.1.5 Comparación entre las subredes de circuitos virtuales y datagramas

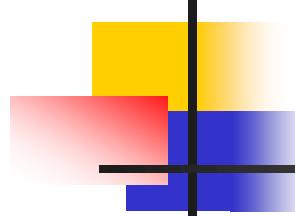
Asunto	Red de datagramas	Red de circuitos virtuales
Configuración del circuito.	No necesaria.	Requerida.
Direccionamiento.	Cada paquete contiene la dirección de origen y de destino completas.	Cada paquete contiene un número de CV corto.
Información de estado.	Los enrutadores no contienen información de estado sobre las conexiones.	Cada CV requiere espacio de tabla del enrutador por cada conexión.
Enrutamiento.	Cada paquete se enruta de manera independiente.	La ruta se elige cuando se establece el CV; todos los paquetes siguen esa ruta.
Efecto de fallas del enrutador.	Ninguno, excepto para paquetes perdidos durante una caída.	Terminan todos los CVs que pasaron por el enrutador defectuoso.
Calidad del servicio.	Difícil.	Fácil si se pueden asignar suficientes recursos por adelantado para cada CV.
Control de congestión.	Difícil.	Fácil si se pueden asignar suficientes recursos por adelantado para cada CV.



## 5.2 Algoritmos de enruteamiento

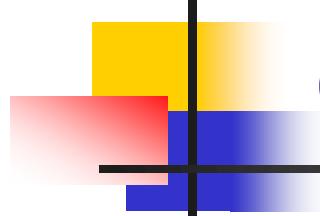
---

- 
- **Algoritmo de enrutamiento:** Construye las rutas y actualiza las tablas de enrutamiento de cada ruteador (RIP)
  - **Reenviar:** es la acción de buscar en la tabla de enrutamiento la línea de salida para una paquete (IP)
  - Con datagramas, la decisión de enrutamiento se toma para cada paquete
  - Con CVs, esta decisión se toma una vez al establecer el CV



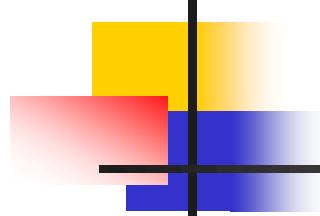
## Propiedades de un algoritmo de enrutamiento

1. **Exactitud.** El algoritmo permite alcanzar el destino deseado y no otro
2. **Sencillez.** El algoritmo trabaja usando el mínimo de recursos y en forma rápida
3. **Robustez.** El algoritmo es capaz de soportar fallas de hosts, ruteadores y líneas de comunicación
4. **Estabilidad.** Determina las rutas para todas las posibles topologías
5. **Equidad.** Se debe permitir enviar a todos los hosts conectados a la red
6. **Optimización.** Usar al máximo de capacidad de transmisión



## Clases principales de algoritmos de enrutamiento

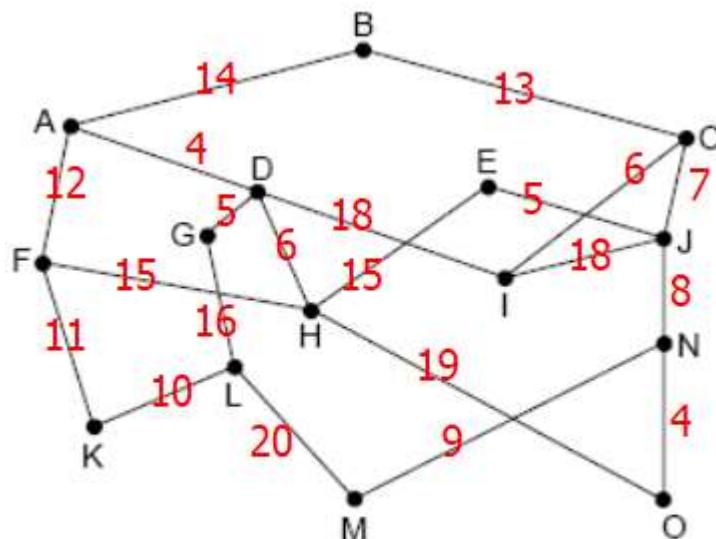
1. **No adaptativos o estáticos.** Las tablas de enrutamiento se determinan *a priori*. Estas no se adaptan a variaciones de tráfico y topología de la red
2. **Adaptativos.** Las tablas varían dinámicamente según variaciones de topología y tráfico
  - Esta información se toma de los ruteadores vecinos o de todos ellos



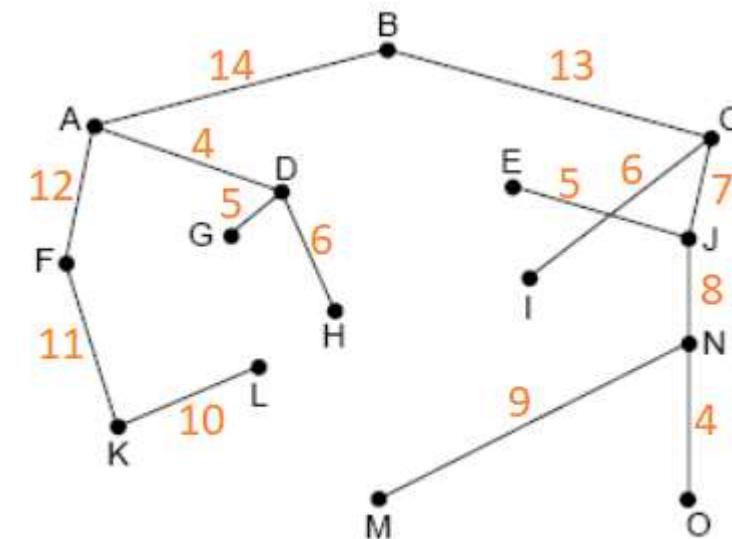
## 5.2.1 Principio de optimización

- Si el router  $J$  está en la ruta óptima de los routers  $I$  al  $K$ , entonces la ruta óptima de  $J$  a  $K$  también está en la misma ruta
- Las mejores rutas desde un origen a todos los destinos forman un **árbol sumidero** con raíz el origen

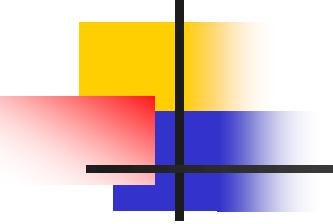
# Árbol generador. Algoritmo PRIM Árbol de peso mínimo



(a)

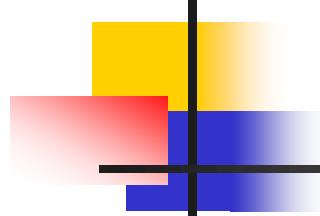


(b)



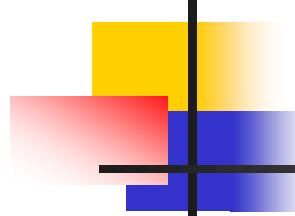
## ■ Algoritmos de enrutamiento **estático**:

- Enrutamiento por la ruta más corta
- Enrutamiento por inundación



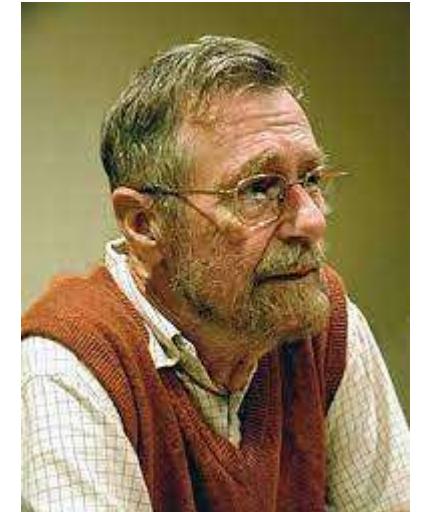
## 5.2.2 Enrutamiento por la ruta más corta

- Se arma un grafo de la subred
  - Cada nodo es un router
  - Cada línea es una línea de comunicación
- La mejor ruta entre dos routers es la **distancia más corta** que indica el grafo
  - Menor número de saltos
  - Menor distancia geográfica en km
- Aparte de la distancia hay otras métricas: retardo, ancho de banda, tráfico medio, costo de comunicación. El algoritmo podría combinar estos criterios

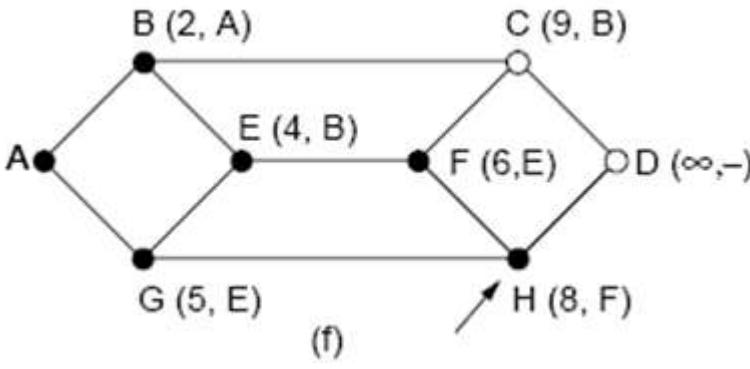
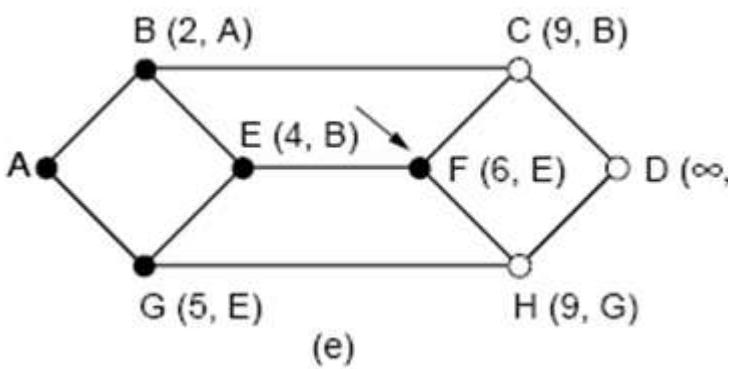
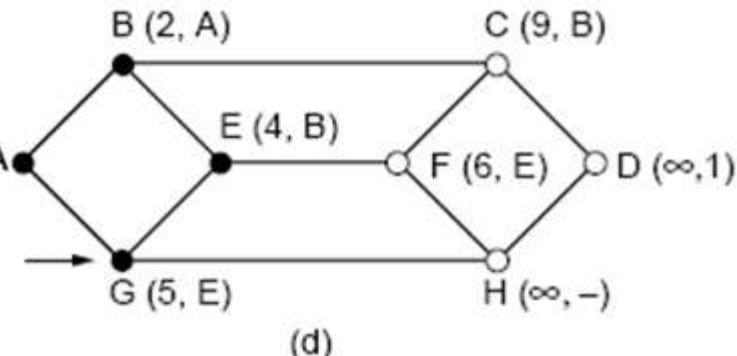
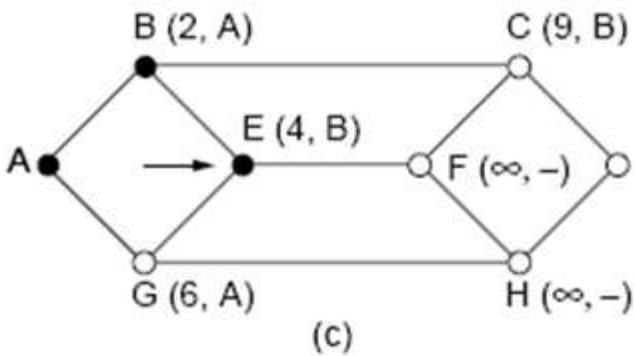
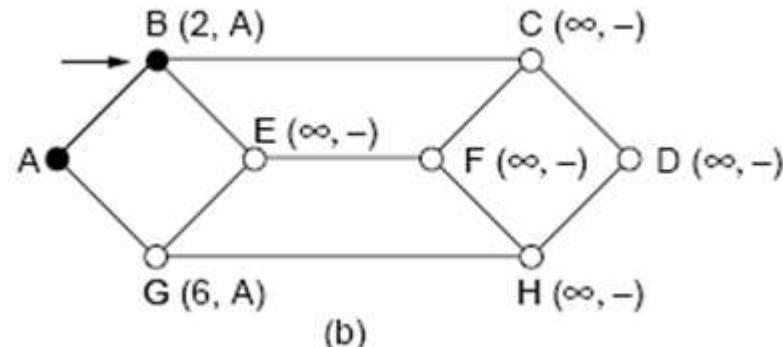
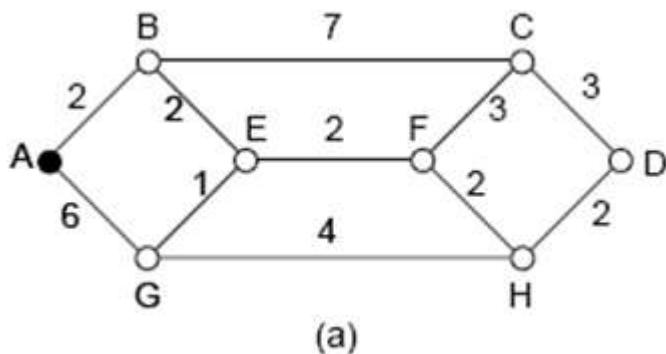


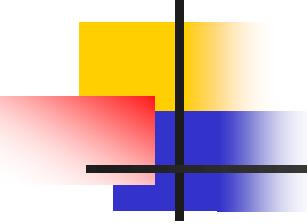
## Algoritmo de la ruta más corta de Dijkstra

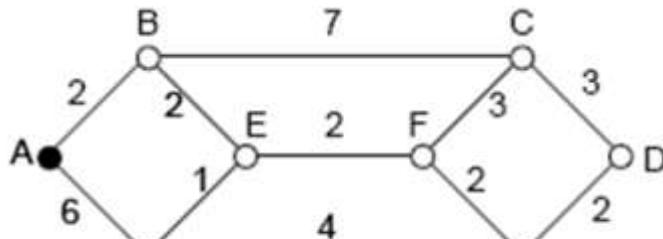
- Diseñado por el holandés Edsger Wybe Dijkstra en 1959
- El algoritmo calcula las distancias y los caminos más cortos desde un nodo a todos los demás nodos de un grafo
- Cada nodo se rotula con su distancia más corta al origen, y el nombre del nodo anterior
- Al inicio, cada etiqueta puede ser sólo tentativa
- La ponderación en cada enlace es, por ejemplo, distancia



1930 - 2002

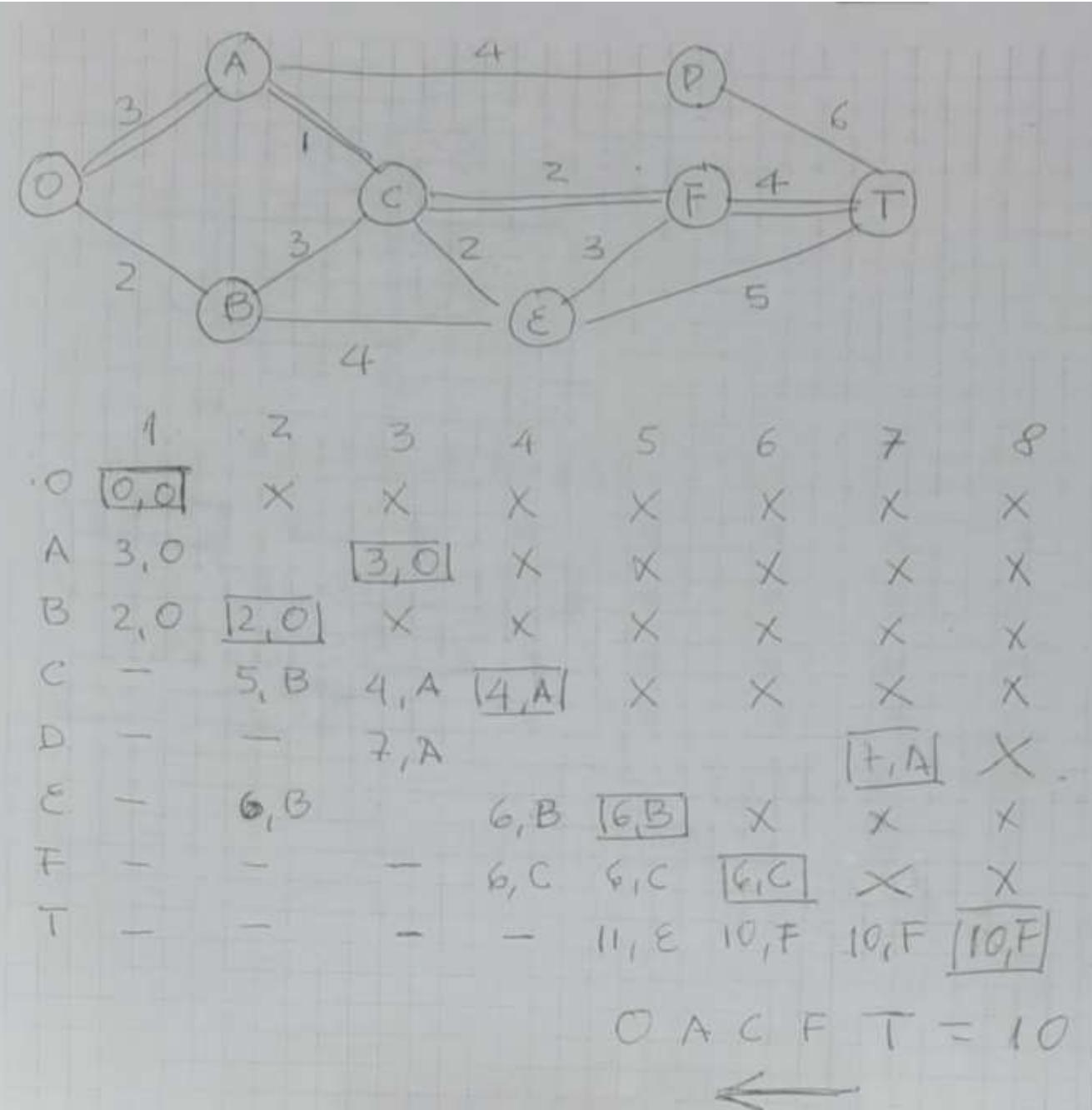


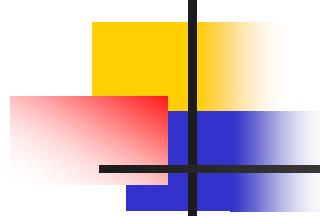
- 
- Para un nodo definitivo (negro) analizamos las distancias de los adyacentes
  - Escogemos el nodo adyacente con menor distancia. A este lo llamamos definitivo (negro)
  - El proceso se repite hasta llegar al nodo destino



	1	2	3	4	5	6	7	8	9
A	(0, A)	*							
B	2, A	$\rightarrow$ (2, A)	*						
C		9, B					$\rightarrow$ (9, B)	*	
D							10, H	12, C	$\rightarrow$ (10, H) *
E	4, B	$\rightarrow$ (4, B)	*						
F		6, E		$\rightarrow$ (6, E)	*				
G	6, A		5, E	$\rightarrow$ (5, E)	*				
H			9, G	8, F	$\rightarrow$ (8, F)	*			

Ruta: A, B, E, F, H, D

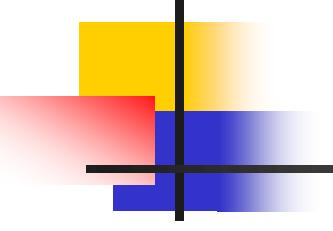


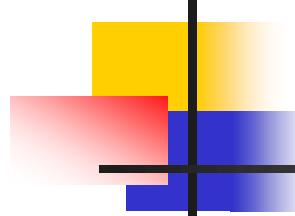


## 5.2.3 Inundación

---

- Cada paquete que llega se envía por todas las líneas, excepto por la que llegó
- La inundación genera grandes cantidades de paquetes **duplicados**
- Para limitarlo se pone un contador de saltos en el encabezado de cada paquete
- El contador se inicializa con un cierto valor máximo
- El contador se disminuye con cada salto
- El paquete se descarta si el contador llega a cero

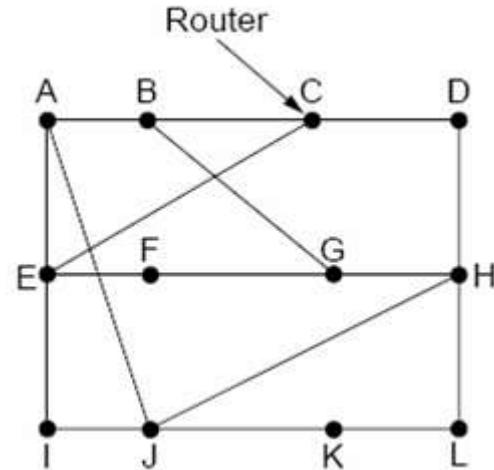
- 
- Una variación de la inundación es la **inundación selectiva**
  - Los enrutadores no envían cada paquete por todas las líneas, sino solo por las salidas que se supone apuntan en la dirección correcta
  - Los algoritmos estáticos no toman en cuenta la carga **actual** de la red
  - Algoritmos de enrutamiento **dinámicos**:
    - Enrutamiento por vector de distancia
    - Enrutamiento por estado del enlace



## 5.2.4 Enrutamiento por vector de distancia

- Años 1957 - 1962
- Cada router tiene un vector con la mejor distancia a cada destino y la línea de salida para llegar ahí
- La tabla se **actualiza** intercambiando información con los **vecinos**
- Vector de distancia fue el algoritmo original de enrutamiento de ARPANET
- Hoy en Internet se usa **RIP**

# Algoritmo de vector distancia



Retardo medido desde J a sus vecinos:

Retardos **medidos por los vecinos** de J: A, I, H, K hacia los demás nodos

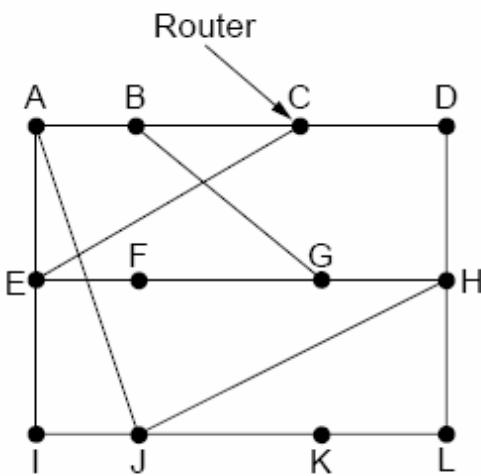
To	A	I	H	K
A	0	24	20	21
B	12	36	31	28
C	25	18	19	36
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
L	29	33	9	9

JA delay is 8  
JI delay is 10  
JH delay is 12  
JK delay is 6

New estimated delay from J Line



New routing table for J



Retardo medido desde J a sus vecinos:

New estimated delay from J

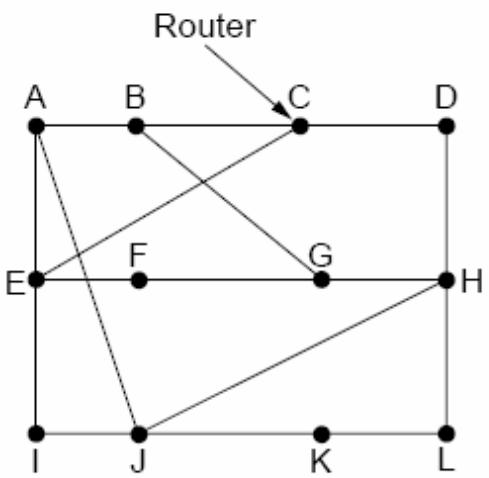
To	A	I	H	K	Line
A	0	24	20	21	
B	12	36	31	28	
C	25	18	19	36	
D	40	27	8	24	
E	14	7	30	22	
F	23	20	19	40	
G	18	31	6	31	
H	17	20	0	19	
I	21	0	14	22	
J	9	11	7	10	
K	24	22	22	0	
L	29	33	9	9	

JA delay      JI delay      JH delay      JK delay

is                is                is                is

8      10      12      6

New routing table for J

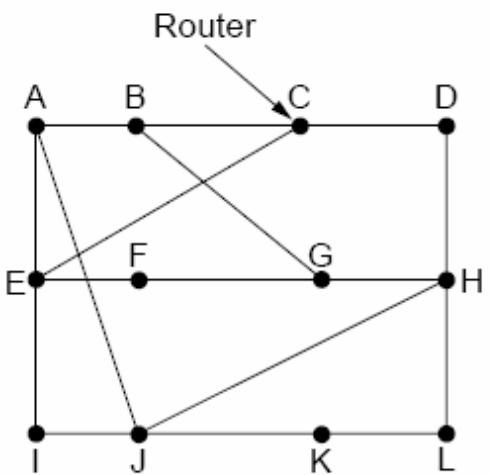


Retardo medido desde J a sus vecinos:

New estimated delay from J

To	A	I	H	K	Line
A	0	24	20	21	
B	12	36	31	28	
C	25	18	19	36	
D	40	27	8	24	
E	14	7	30	22	
F	23	20	19	40	
G	18	31	6	31	
H	17	20	0	19	
I	21	0	14	22	
J	9	11	7	10	
K	24	22	22	0	
L	29	33	9	9	
	JA delay is	JI delay is	JH delay is	JK delay is	
	8	10	12	6	

New routing table for J



Retardo medido desde J a sus vecinos:

Distancia de J a G: **26**

Distancia de J a I: **41**

Distancia de J a H: **18**

Distancia de J a K: **37**

New estimated delay from J

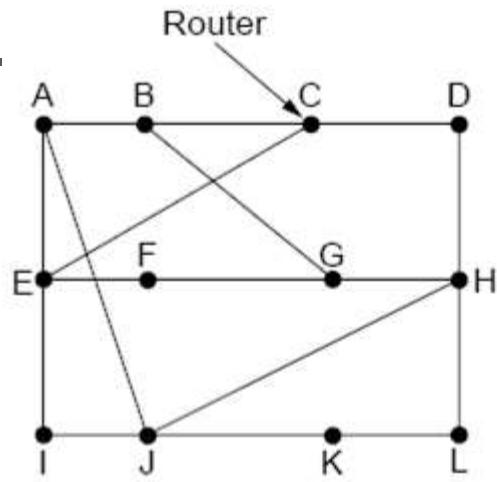
Line

To A I H K

	A	I	H	K
A	0	24	20	21
B	12	36	31	28
C	25	18	19	36
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
L	29	33	9	9

JA delay is 8   JI delay is 10   JH delay is 12   JK delay is 6

New routing table for J



Retardo medido desde J a sus vecinos:

Distancia de J a G:

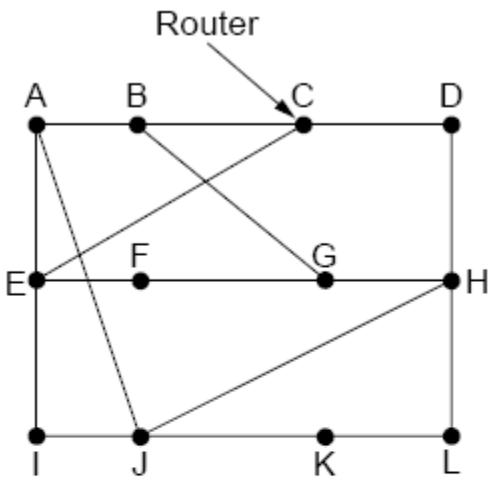
New estimated delay from J

Line

New routing table for J

To	A	I	H	K	Line
A	0	24	20	21	
B	12	36	31	28	
C	25	18	19	36	
D	40	27	8	24	
E	14	7	30	22	
F	23	20	19	40	
G	18	31	6	31	
H	17	20	0	19	
I	21	0	14	22	
J	9	11	7	10	
K	24	22	22	0	
L	29	33	9	9	
JA delay		JI delay	JH delay	JK delay	
is		is	is	is	
8		10	12	6	
		26	40	18	37

Calcular las distancias y la ruta desde J a los destinos A, B, C y D



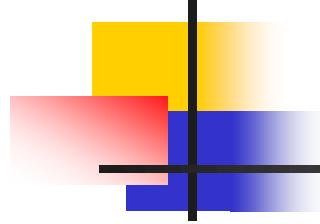
Retardo medido desde J a sus vecinos:

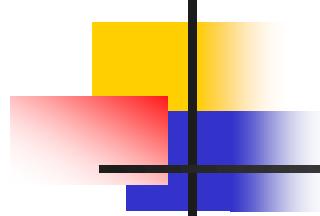
New estimated delay from J

To	A	I	H	K	Line
A	0	24	20	21	8 A
B	12	36	31	28	20 A
C	25	18	19	36	28 I
D	40	27	8	24	20 H
E	14	7	30	22	17 I
F	23	20	19	40	30 I
G	18	31	6	31	18 H
H	17	20	0	19	12 H
I	21	0	14	22	10 I
J	9	11	7	10	0 -
K	24	22	22	0	6 K
L	29	33	9	9	15 K

JA delay is 8   JI delay is 10   JH delay is 12   JK delay is 6

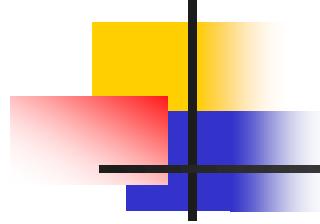
New routing table for J

- 
- La métrica usada es el número de paquetes en cola a lo largo de la ruta
  - Inconveniente del enrutamiento por vector de distancia: Lentitud



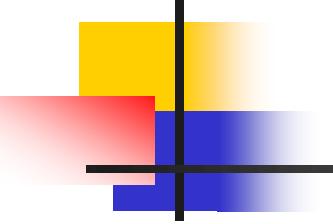
## 5.2.5 Enrutamiento por *estado del enlace*

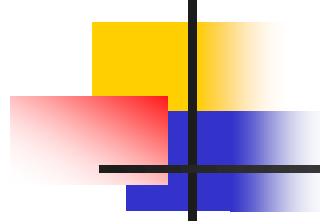
- Vector de distancia se usó en ARPANET hasta el año de 1979
- Fue cambiado por *estado del enlace* debido a dos problemas:
  1. Lentitud
  2. La métrica es el retardo en la cola, no toma en cuenta el ancho de banda de los enlaces al escoger rutas



# Pasos en el que se basa el *estado del enlace*

1. Descubrir las direcciones de sus vecinos
2. Medir el **retardo** a cada uno de sus vecinos
3. Construir un paquete con esta información
4. Compartir este paquete con **todos** los demás enrutadores
5. Calcular la ruta más corta a cada router

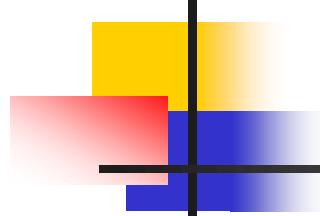
- 
- La topología y retardos se determinan experimentalmente
  - Se aplica el algoritmo de *Dijkstra* para encontrar la ruta más corta a los demás enrutadores



## 1. Conocer a sus vecinos

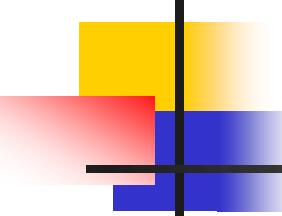
---

- Un enrutador envía un paquete *HELLO* por cada línea a sus enrutadores vecinos
- Cada vecino responde con su nombre para indicar quien es
- Los nombres deben ser globalmente únicos



## 2. Medición del costo de la línea

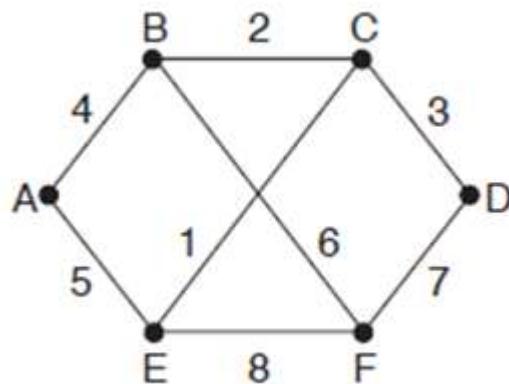
- Se determina el retardo a cada uno de sus vecinos enviando un mensaje *echo request*, y esperando un mensaje *echo replay* del protocolo *ICMP Internet Control Message Protocol*
- Estos mensajes son usados por el comando *ping*
- *Ping* es una herramienta disponible en varios sistemas operativos como *IOS CISCO, WINDOWS*



### 3. Construcción de los paquetes de estado del enlace

- El paquete contiene:
  1. Identidad del emisor
  2. Número de secuencia
  3. Edad
- Estos paquetes se generan periódicamente, o cuando ocurre una caída/reactivación de línea o vecino

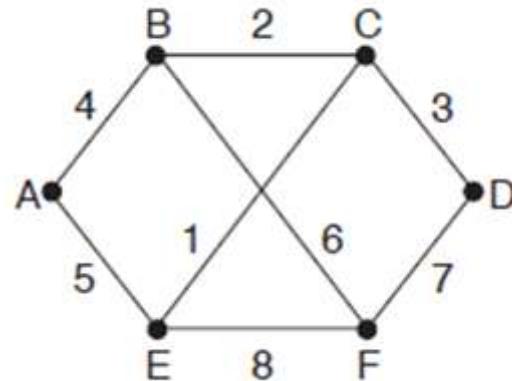
## Paquetes de estado de enlace construidos por cada enrutador



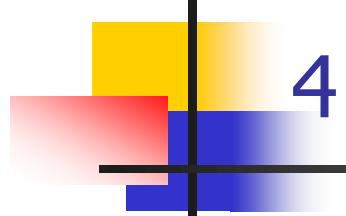
A	B	C	D	E	F
Sec.	Sec.	Sec.	Sec.	Sec.	Sec.
Edad	Edad	Edad	Edad	Edad	Edad
B 4	A 4	B 2	C 3	A 5	B 6
E 5	C 2	D 3	F 7	C 1	D 7
F 6	F 6	E 1	F 8	F 8	E 8

Los paquetes de estado de enlace son enviados a todos los enrutadores de la red

Construya la topología de la red con base en los mensajes recibidos desde todos los ruteadores

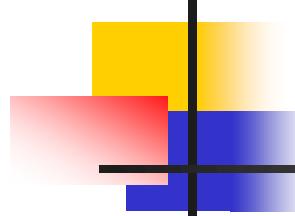


A	B	C	D	E	F
Sec.	Sec.	Sec.	Sec.	Sec.	Sec.
Edad	Edad	Edad	Edad	Edad	Edad
B   4	A   4	B   2	C   3	A   5	B   6
E   5	C   2	D   3	F   7	C   1	D   7
	F   6	E   1		F   8	E   8



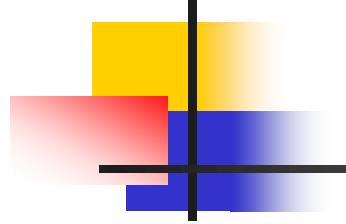
## 4. Distribución de paquetes de estado del enlace

- Los routers usan estos paquetes para construir la topología de la red
- Los routers podrían estar usando versiones diferentes de la topología
- Cada paquete tiene un número secuencial
- Cada enrutador registra el enrutador origen y la secuencia
- El paquete se distribuye por inundación
- Si es duplicado se descarta



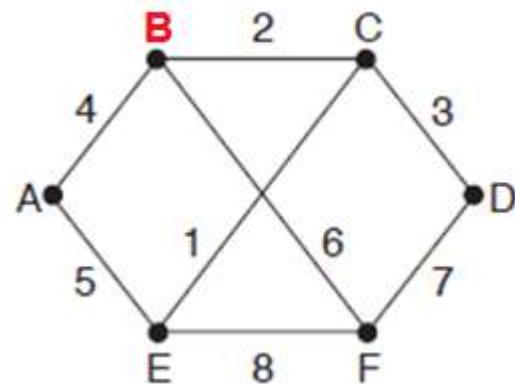
# Problemas de los paquetes de estado de enlace

- Problema 1
  - Cuando el número de secuencia vuelve a cero
  - Solución: usar un número de secuencia de 32 bits
  - Con un paquete por segundo, el tiempo para volver a empezar será de **136 años**
- Problema 2
  - Si falla el router el secuencial vuelve a 0, el paquete con este número será rechazado
- Problema 3
  - Si se daña un bit y en vez de 000000000000100 se tiene 100000000000100 el paquete con secuencial 5 será rechazado por obsoleto

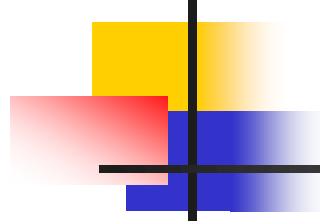


## ■ Solución:

- La **edad** en cada paquete de estado inicia en 60
- La edad se disminuye en 1 en cada seg.
- Cuando la edad es 0, se descarta la información de ese enrutador

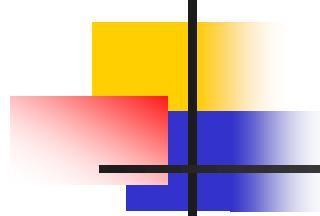


Origen	Sec.	Edad	Banderas de reenvío			Banderas de ACK		
			A	C	F	A	C	F
A	21	60	0	1	1	1	0	0
F	21	60	1	1	0	0	0	1
E	21	59	0	1	0	1	0	1
C	20	60	1	0	1	0	1	0
D	21	59	1	0	0	0	1	1



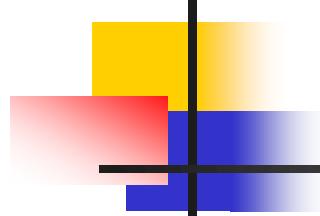
## Figura 5.13

- Se presenta la tabla del router B
- Descripción de la primera fila:
- B recibe de A un paquete de estado de enlace
- B reenvía el paquete recibido por las líneas de salida hacia C y F
- B devuelve a A un acuso de recibo por la línea que recibió el paquete
- Una fila es de un paquete de estado de enlace
- Análogamente con las otras filas



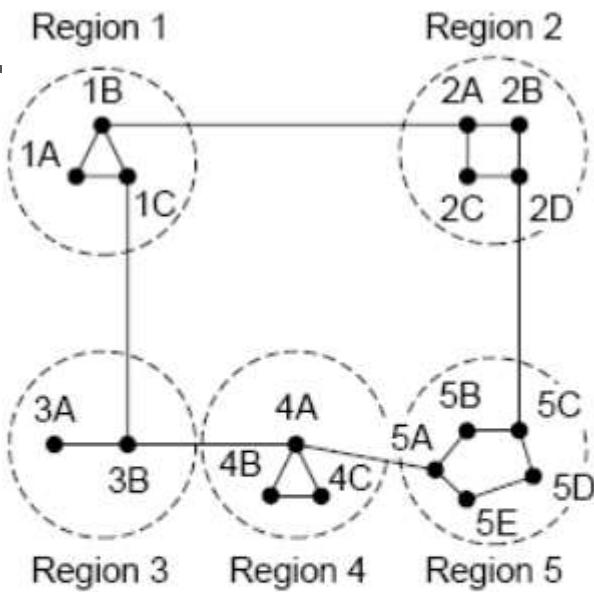
## 5. Cálculo de las nuevas rutas

- Cuando el router tiene todos los paquetes de estado del enlace, construye el grafo
- Se usa *Dijkstra* para construir la ruta más corta a todos los destinos
- Con muchos routers, hay problema de memoria
- Este algoritmo funciona bien en la práctica
- Muy usado en redes actuales
- El protocolo OSPF usa este protocolo
  - Usado en redes dorsales de Internet



## 5.2.6 Enrutamiento jerárquico

- En redes grandes un router no puede tener información a todos los destinos
- Se usa enrutamiento **jerárquico**, similar a la red telefónica
- Los enrutadores se agrupan en regiones
- Un enrutador conoce los detalles para enrutar a destinos **dentro de su región**
- Un enrutador no conoce la estructura interna de otras regiones
- Redes grandes se organizan en regiones, closters, zonas, grupos, etc.
- El enrutamiento jerárquico reduce el tamaño de las tablas de enrutamiento



(a)

Full table for 1A

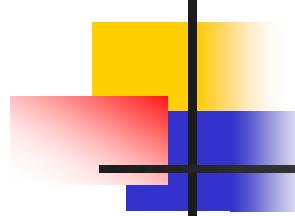
Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchical table for 1A

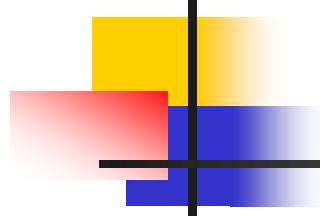
Dest.	Line	Hops
1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)



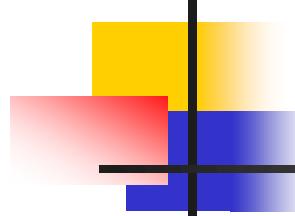
## 5.2.7 Enrutamiento por difusión

- A veces, los hosts necesitan enviar mensajes a varios o a todos los host
- El envío simultáneo de un paquete a todos los hosts se llama **difusión**
- Hay varios métodos
- El origen envía un paquete distinto a todos los destinos
  - Desperdicia ancho de banda
  - Requiere que el origen tenga una listas de los destinos
- Otro método es la inundación
  - El problema es que genera demasiados paquetes
  - Desperdicia ancho de banda



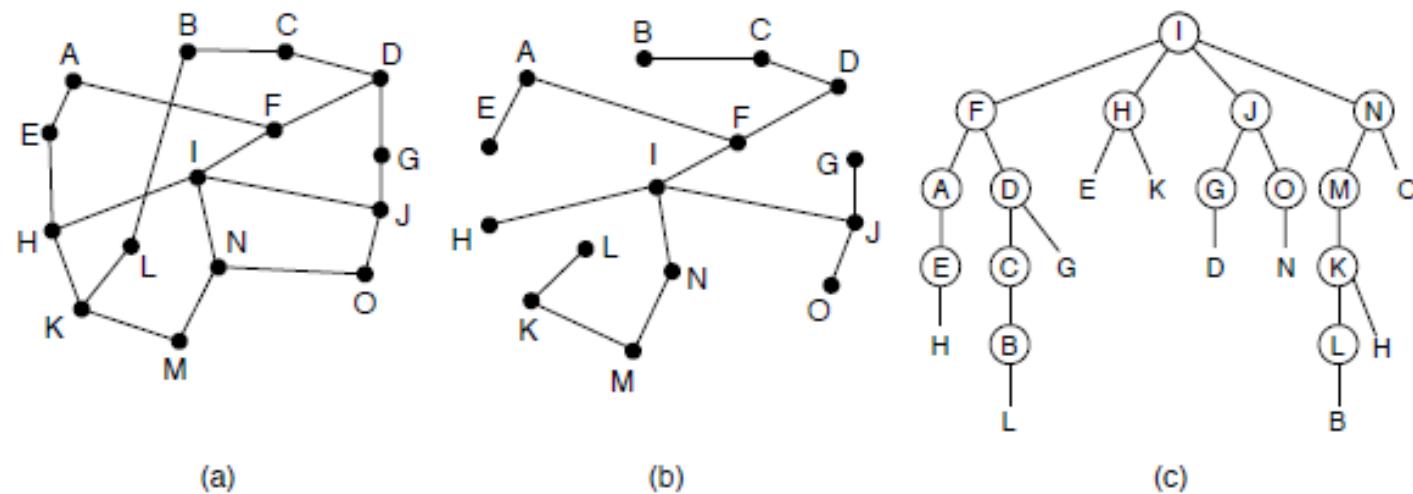
# Enrutamiento por difusión: Método del enrutamiento multidestino

- Cada paquete contiene una lista con los destinos deseados
- El enrutador revisa los destinos y determina las líneas de salida que necesita
- El enrutador **genera una copia** del paquete para cada línea de salida

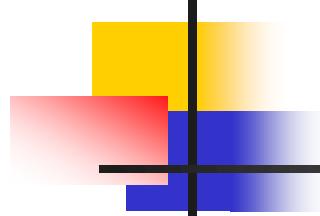


## Enrutamiento por difusión: Difusión por árbol sumidero

- El *árbol sumidero* incluye a todos los ruteadores de la red
- No contiene ciclos
- Uso óptimo del ancho de banda
- Necesidad: cada ruteador debe conocer su árbol sumidero

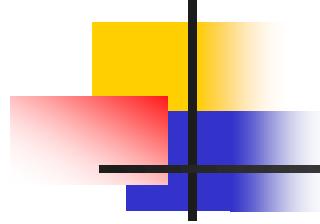


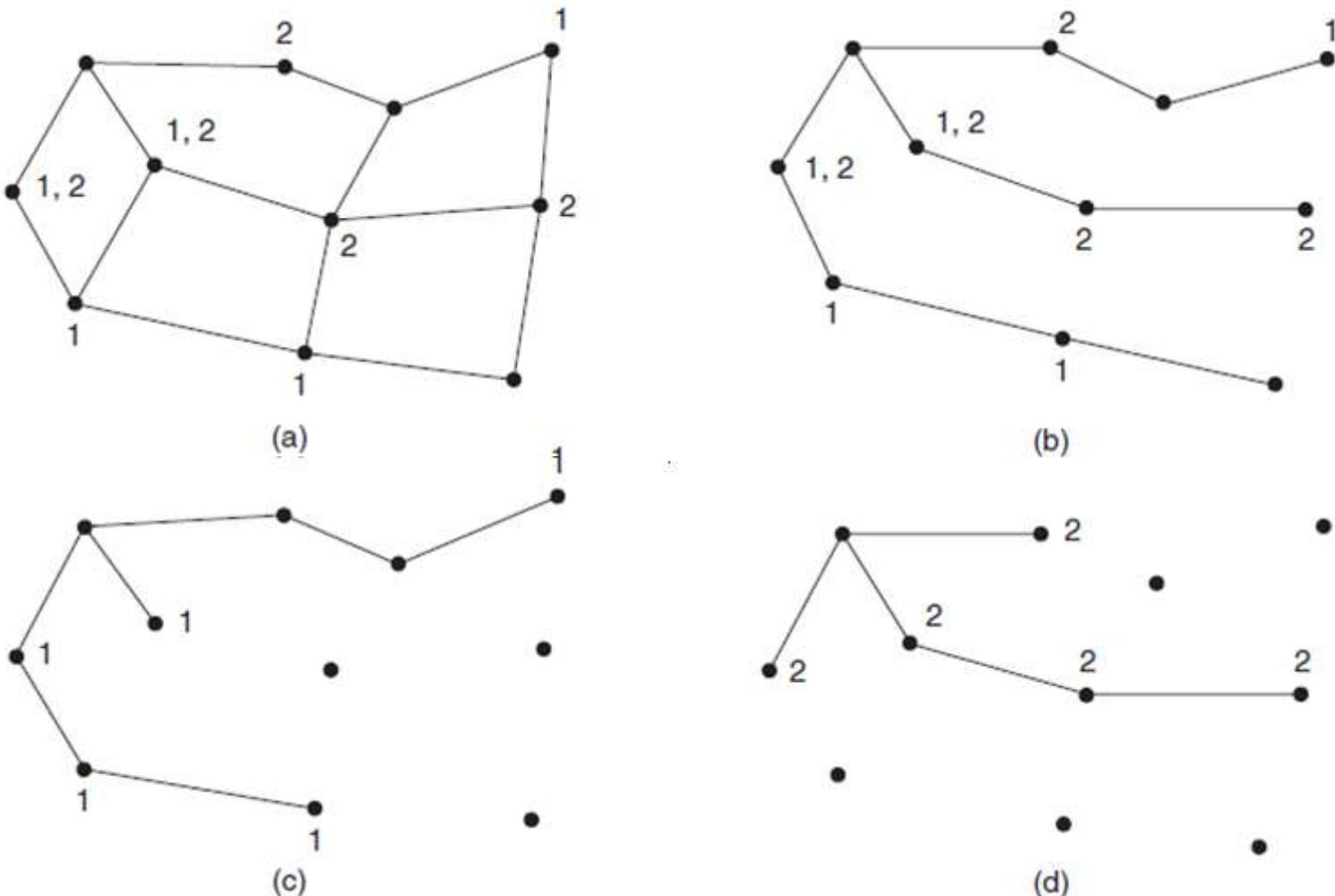
**Figura 5-15.** Reenvío por ruta invertida. (a) Una red. (b) Un árbol sumidero. (c) El árbol construido mediante reenvío por ruta invertida.



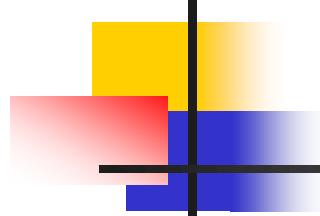
## 5.2.8 Enrutamiento por multidifusión

- Algunas aplicaciones requieren que procesos separados trabajen en grupo
  - Base de datos distribuida: Registro Civil. Cada regional tiene la información de una parte de los ciudadanos
  - Base de datos fragmentada: IESS. Datos generales. Aportes patronales y de los afiliados. Préstamos hipotecarios, quirografarios, prendarios. Historias clínicas
  - Bases de datos híbridas
- Con frecuencia un proceso envía un mensaje a **todos** los demás miembros de un grupo
- Los miembros de otros grupos no deben ver esos mensajes

- 
- Se necesita enviar mensajes a grupos claramente definidos
    - Esto es **multidifusión**
  - Se necesita crear y destruir grupos, y que los procesos entren y salgan de los grupos
    - Un proceso requiere enviar mensajes a un grupo
    - Otro proceso ya no requiere enviar mensajes
  - Los enrutadores deben saber cuáles hosts pertenecen a qué grupos
  - Los enrutadores deben enviar periódicamente las listas de sus hosts
  - Una manera de hacerlo es en routers CISCO a través de ACLs

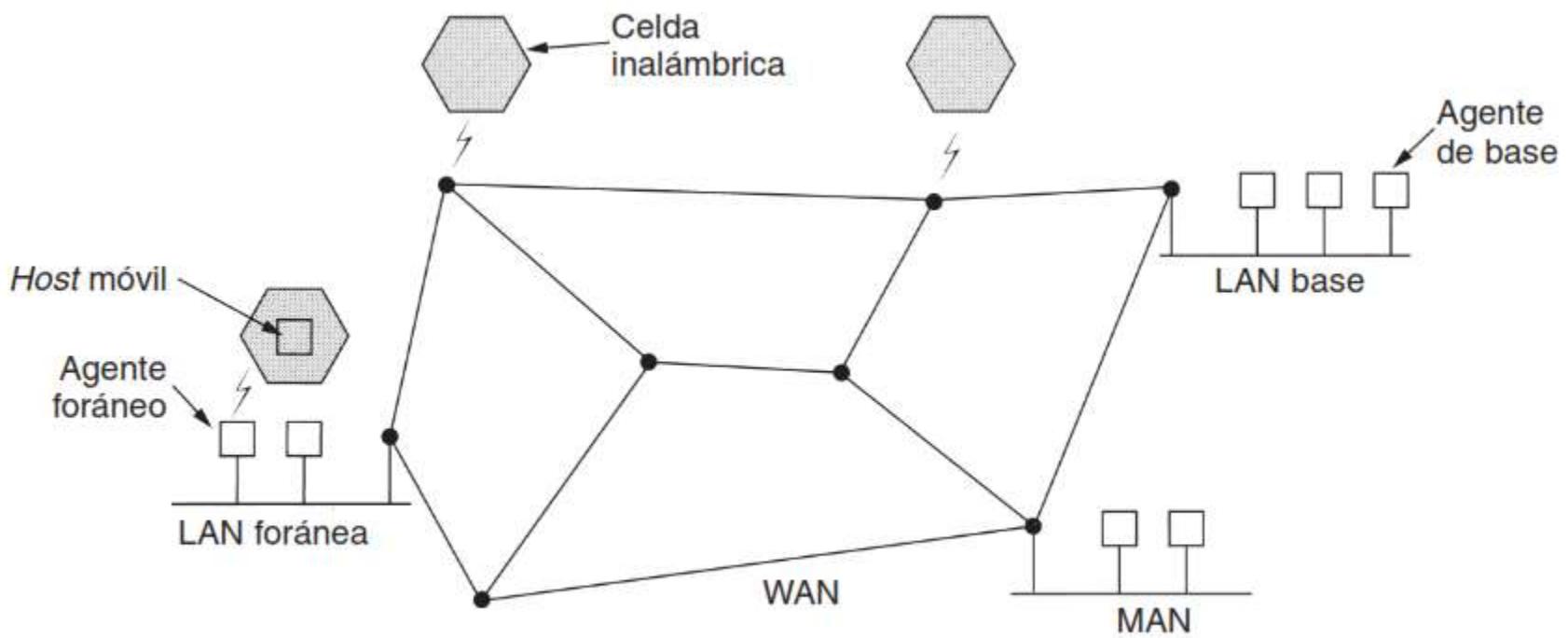


**Figura 5-16.** (a) Una red. (b) Un árbol de expansión para el enrutador del extremo izquierdo. (c) Un árbol de multidifusión para el grupo 1. (d) Un árbol de multidifusión para el grupo 2.

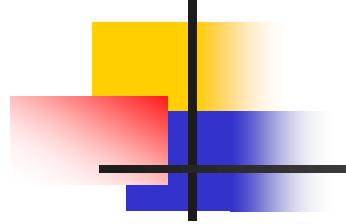


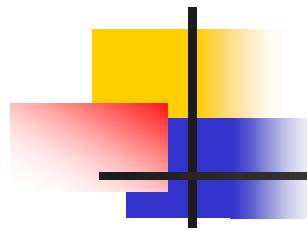
## 5.2.9 Enrutamiento para hosts migratorios y ambulantes

- Para enviar un paquete a un portátil, la red primero tiene que localizarlo
- Tipos de hosts que se conectan a una red
  - Hosts estacionarios (PCs) se conectan a una red cableada (cobre o FO)
  - Hosts **migratorios** (laptops), se mueven de un lugar fijo a otro. Se conectan a una red cableada o inalámbrica
  - Hosts **ambulantes** (smart phone, lector de tarjetas de crédito, dispositivo para control de inventarios) trabajan en movimiento. Se conectan inalámbricamente
- Los hosts migratorios tienen:
  - **Localidad base** que no cambia
  - Dirección base permanente



**Figura 5-18.** WAN a la que están conectadas LANs, MANs y celdas inalámbricas.

- 
- Una red se divide en áreas
  - **Área**: puede ser una LAN o una WLAN (celda inalámbrica)
  - Cada área tiene un agente de base y un agente foráneo
  - **Agente de base**: registra los hosts que pertenecen a su área base
  - **Agente foráneo**: registra los hosts migratorios que **visitan** el área foránea
  - Si un host portátil entra en un área foránea, debe registrarse con el agente foráneo

- 
- Procedimiento de registro de un host portátil migratorio
    1. Periódicamente el agente foráneo difunde un paquete con su dirección. El host portátil espera a que le llegue uno de esos paquetes
    2. El host móvil envía su dirección al agente foráneo
    3. El agente foráneo se comunica con el agente de base del host portátil para informar que dicho host con tal dirección ha llegado a la red foránea
    4. El agente de base, luego de confirmar que el host portátil sí pertenece a su área de base, le indica al agente foráneo que registre al host portátil
    5. El agente foráneo registra al host portátil

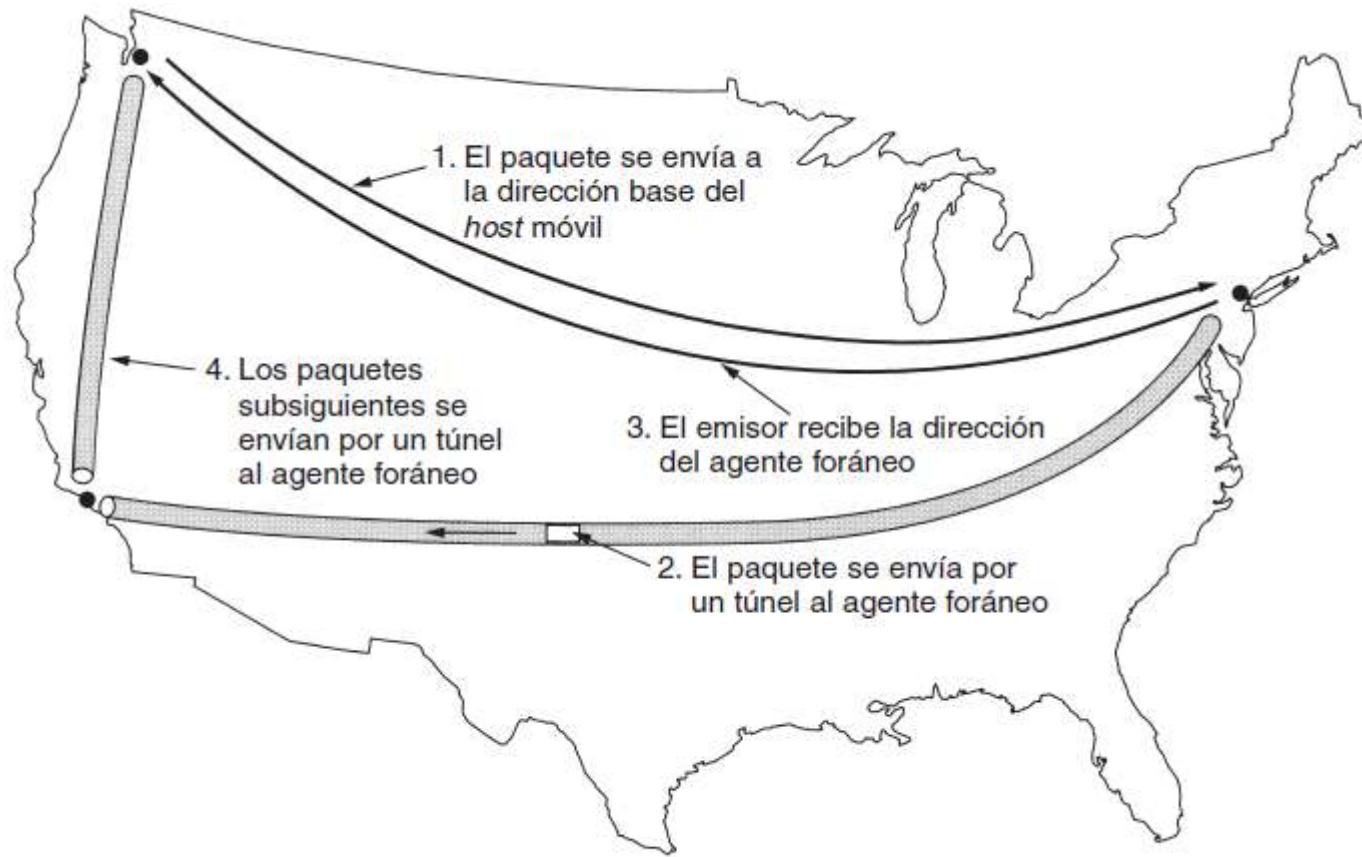
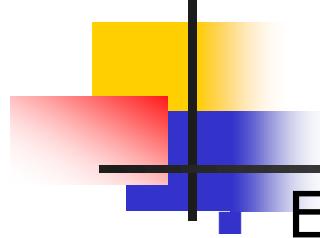
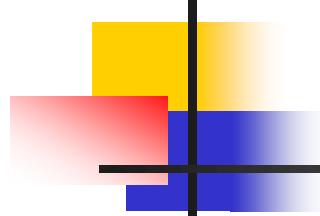


Figura 5-19. Enrutamiento de paquetes para *hosts* móviles.



El emisor en Seattle, envía un paquete a la red base del host portátil en NY

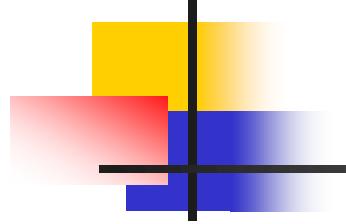
- El agente base en NY sabe que la ubicación temporal del portátil es LA
- El agente base reenvía el paquete en un túnel al agente foráneo en LA.
  - Túnel es enviar un paquete con una IP destino, encapsulado dentro de otro paquete con otra IP destino
- El agente foráneo en LA reenvía el paquete al host migratorio
- El agente base en NY indica al emisor en Seattle que en el futuro envíe paquetes directo al agente foráneo en LA
- El emisor envía directamente los paquetes en un túnel al agente foráneo

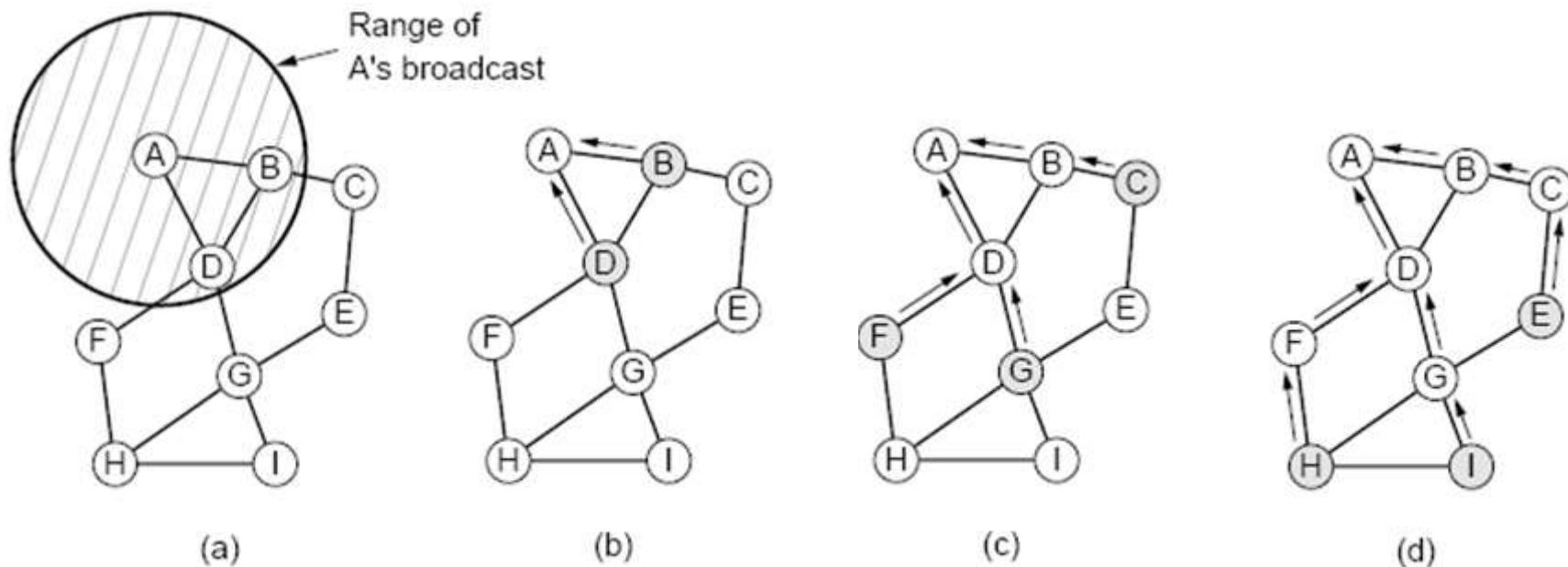


## 5.2.10 Enrutamiento en redes *ad hoc*

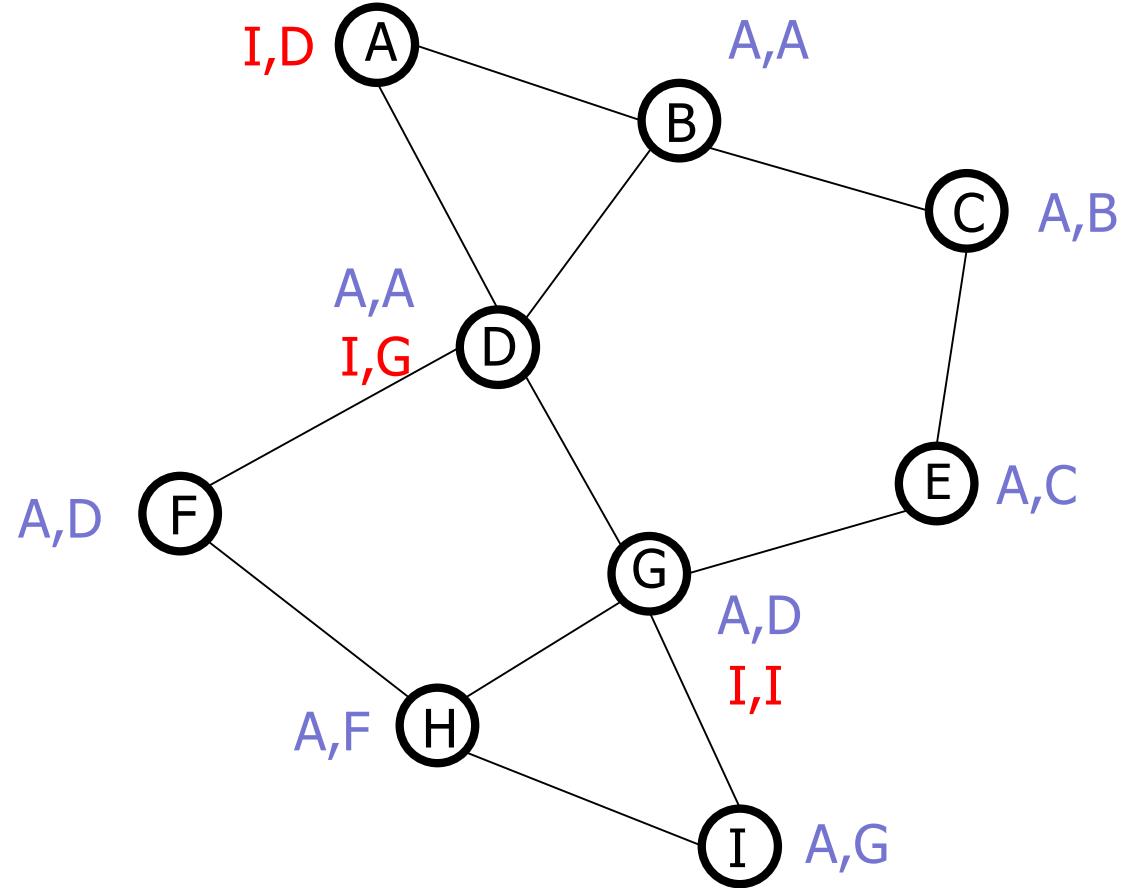
---

- Es utilizado por laptops y dispositivos móviles
- Los hosts hacen las funciones de enrutadores
- Situaciones en las cuales se utilizan redes ad-hoc:
  - Vehículos militares en un campo de batalla
  - Una flota de barcos en el mar
  - Trabajadores en un área de desastre
  - Reunión de personas con PCs portátiles en un área sin 802.11

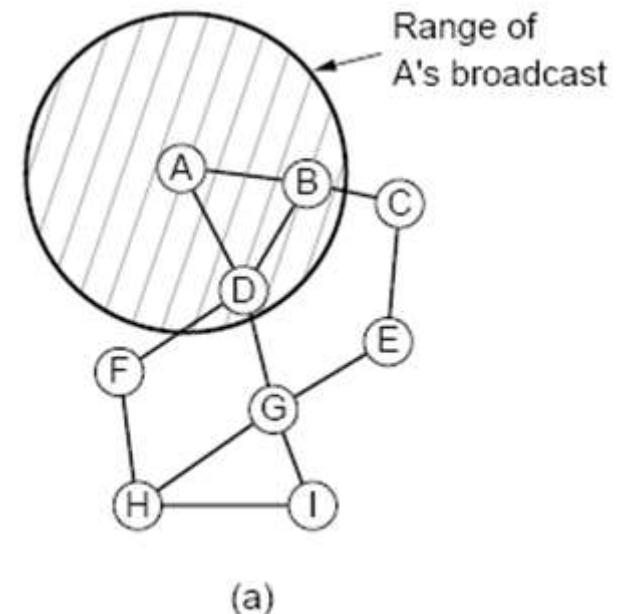
- 
- La topología de la red cambia todo el tiempo
  - Las rutas cambian todo el tiempo
  - Algoritmo de enrutamiento AODV Ad hoc On-Demand Distance Vector
  - Se define una ruta a un destino sólo cuando alguien desea enviar un paquete
  - Dos nodos se conectan si se pueden comunicar directamente
  - Si uno de los dos tiene un emisor más potente, A está conectado a B pero B no está conectado a A



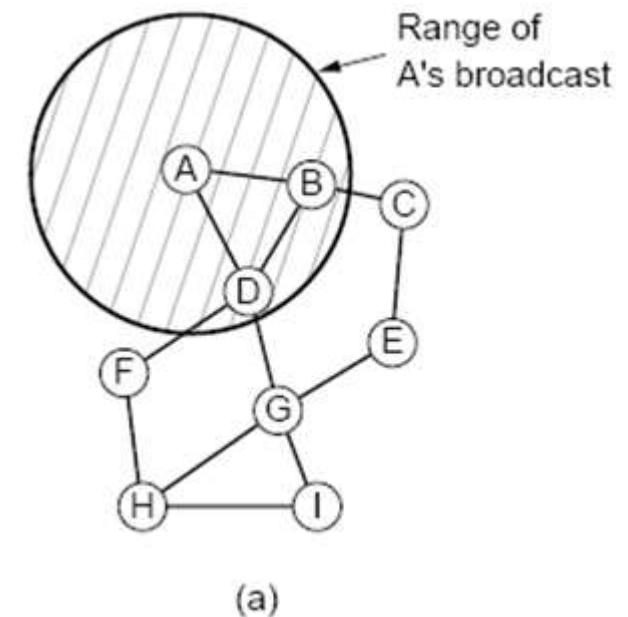
Los nodos que reciben el broadcast crean una ruta que apunta al nodo A

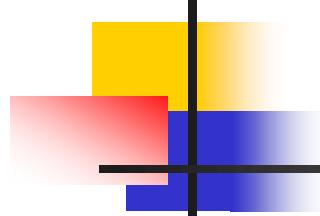


- A desea enviar un paquete a I
- AODV mantiene una tabla en cada nodo
- A busca en su tabla pero no encuentra una entrada para I
- Ahora debe descubrir la ruta a I
- A difunde un paquete de solicitud de ruta
- El paquete llega a B y D



- B y D lo vuelven a difundir
- Cada nodo que recibe la difusión crea una ruta que apunta a A
- El paquete de solicitud de ruta alcanza a I
- I crea un paquete de respuesta de ruta
- Este paquete sólo se envía al nodo del cual vino la solicitud

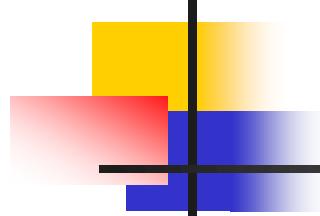




## Mantenimiento de rutas en redes *ad-hoc*

---

- Es posible mover y apagar nodos
- La topología puede cambiar
- El algoritmo necesita manejar estas situaciones
- Cada nodo difunde de manera **periódica** un mensaje de saludo (*Hello*)
- Cada vecino próximo responde al saludo
- Si un vecino no responde, sabe que este ya no está conectado a él
- Esta información se usa para eliminar rutas que ya no funcionan



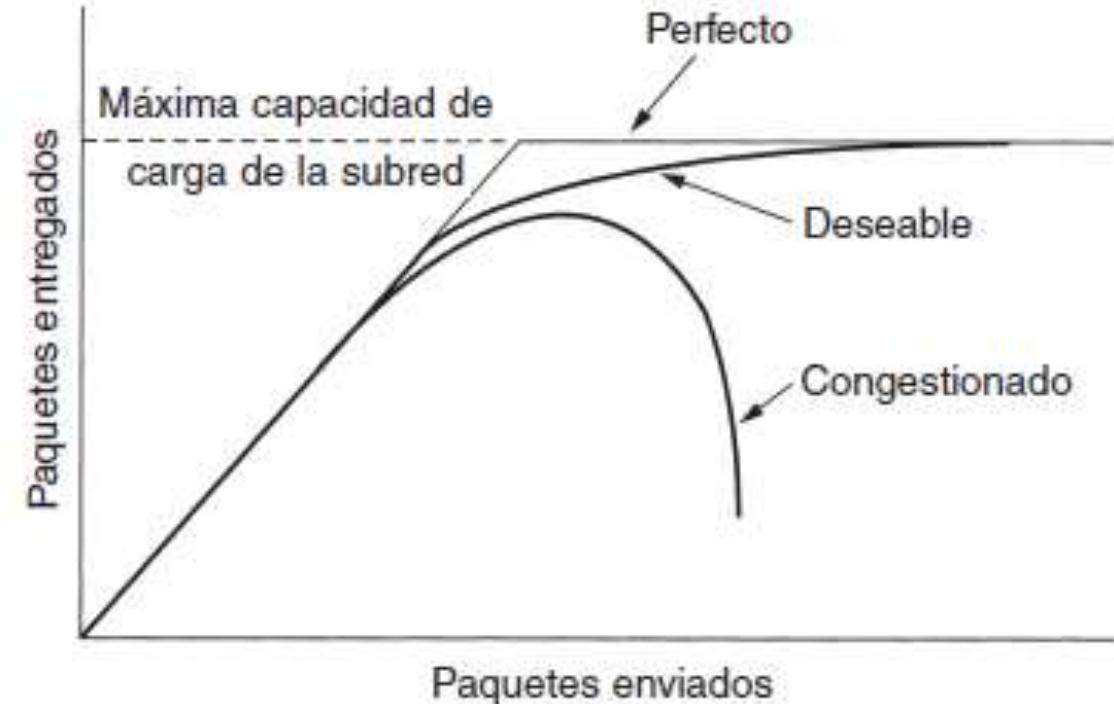
## 5.2.11 Búsqueda de nodos en redes de igual a igual

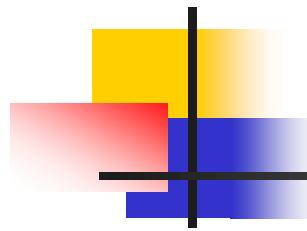
- Caso de las redes ad-hoc
- Los sistemas de igual a igual son totalmente distribuidos
- Todos los nodos son simétricos
- No hay nodo central o jerarquía
- Cada usuario tiene algo de información de interés para otros usuarios



## 5.3 Algoritmos de control de congestión

- Cuando hay demasiados paquetes en la subred, este disminuye su desempeño
- Esto se llama congestión
- Hay algoritmos de enrutamiento que trabajan en función de la congestión en la red

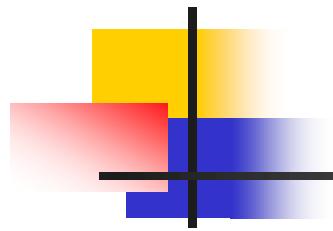


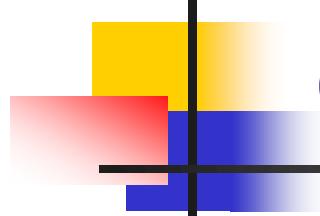


## Razones de la congestión



1. Llegan muchos paquetes por diferentes líneas de entrada pidiendo la misma línea de salida
2. Memoria de almacenamiento insuficiente en ruteadores
3. Routers con memoria infinita sin aumentar la velocidad del procesador ni el ancho de banda de las líneas de salida, la congestión empeora

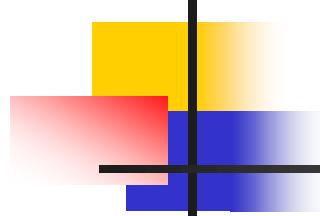
- 
- Situación:
    - Memoria insuficiente
    - Procesadores lentos
    - Líneas de poco ancho de banda
  - Actualizar solo una parte del sistema mueve el cuello de botella a otra parte
  - Solución: Se deben equilibrar todas las partes del sistema



## Control de congestión y control de flujo

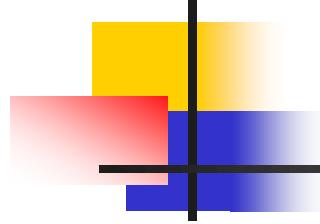
---

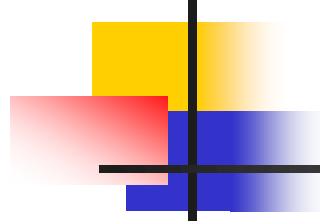
- El control de congestión es un asunto global
  - Procura que la subred sea capaz de transportar el tráfico ofrecido
- El control de flujo se relaciona con el tráfico entre un emisor rápido y un receptor lento
  - Implica realimentación del receptor al emisor
- Algunos algoritmos de control de congestión retroalimentan a varios emisores pidiéndoles reducir la velocidad
- Un host puede recibir un mensaje de reducir velocidad porque el receptor es lento o porque la red no la puede manejar



## 5.3.1 Principios generales del control de congestión

- La teoría de control divide en dos grupos las soluciones:
  - Soluciones de ciclo abierto: lo resuelven con un buen diseño
  - Soluciones de ciclo cerrado: se basan en retroalimentación
    - 1. Monitorear el sistema
    - 2. Informar a los lugares que puedan actuar
    - 3. Cuando se presenta la congestión, los lugares actúan para corregir el problema

- 
- Métricas para monitorear la subred en busca de congestión:
    - % de paquetes descartados en los routers
    - Promedio de la longitud las colas
    - Cantidad de paquetes que venció su tiempo de vida y se tienen que volver a transmitir
    - Retardo promedio de paquetes



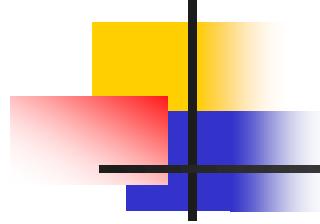
## Información de la congestión



- El router que detecta la congestión envía un paquete informando el problema al origen u orígenes del tráfico
  - Estos paquetes aumentan la congestión
- Otra manera: hosts o routers envían periódicamente paquetes de sondeo de congestión
  - Esta información permite desviar tráfico fuera del área congestionada

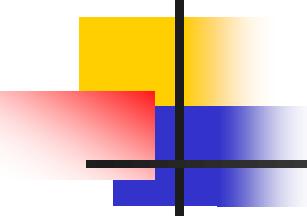
- La retroalimentación no debe hacerse apenas se inicia la congestión, ni tampoco mucho tiempo después de iniciada la congestión





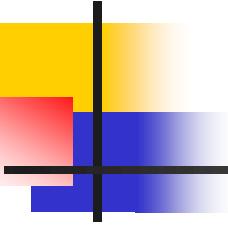
## ■ Algoritmos de ciclo cerrado se dividen en dos:

- Retroalimentación explícita: regresan paquetes desde el punto de congestión para avisar al origen
- Retroalimentación implícita: el origen deduce que hay congestión según el tiempo para recibir ACKs

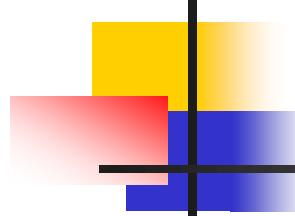
- 
- Soluciones frente a la congestión:
    - Reducir la carga
      - Negar el servicio a algunos usuarios
      - Degradar el servicio
    - Aumentar recursos
    - Dividir el tráfico entre varias rutas



## 5.4 Calidad de Servicio QoS



- El control de congestión no es suficiente en redes multimedia
- Se necesita controlar la calidad de servicio
- No todas las aplicaciones de red requieren la misma calidad de servicio
- Hay maneras para dar QoS ajustadas a las necesidades de las aplicaciones



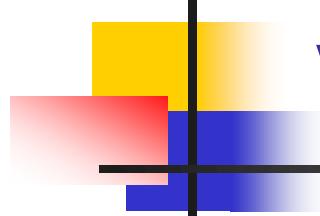
## 5.4.1 Requerimientos

- **Flujo:** Conjunto de paquetes que van desde un origen a un destino
- Los paquetes de un flujo pueden o no ir en una misma ruta según hayan CVs o datagramas
- El flujo se caracteriza según 4 parámetros:
  - Ancho de banda
  - Retardo
  - Variación o fluctuación de retardo
  - Pérdida de paquetes o confiabilidad
- Cada uno de estos parámetros son requeridos según cada aplicación de red

Aplicación	Ancho de banda	Retardo	Variación del retardo	Pérdida
Correo electrónico.	Bajo	Bajo	Baja	Media
Compartir archivos.	Alto	Bajo	Baja	Media
Acceso a Web.	Medio	Medio	Baja	Media
Inicio de sesión remoto.	Bajo	Medio	Media	Media
Audio bajo demanda.	Bajo	Bajo	Alta	Baja
Video bajo demanda.	Alto	Bajo	Alta	Baja
Telefonía.	Bajo	Alto	Alta	Baja
Videoconferencias.	Alto	Alto	Alta	Baja

Figura 5-27. Nivel de los requerimientos de calidad del servicio de la aplicación.

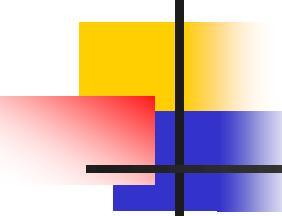
**Bajo:** parámetro bajamente requerido  
**Alto:** parámetro altamente requerido



## Variación de retardo o fluctuación

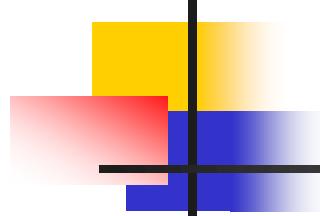
---

- Si la fluctuación de video a través de la red es entre 1 y 2 segundos, el resultado es terrible
- En audio, una fluctuación de unos cuantos mseg. es claramente audible



## Pérdida de paquetes o confiabilidad

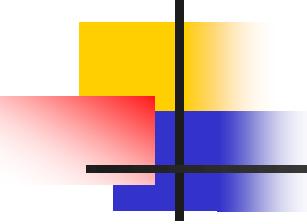
- Necesidad de tener un flujo sin pérdida ni errores
- Se alcanza con:
  - ACK, y
  - La suma de verificación – Checksum en CRC Cyclic Redundancy Check
- Aplicaciones en tiempo real no hacen Checksum

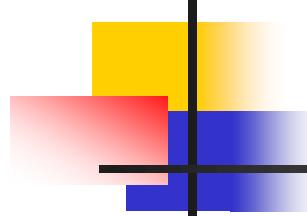


## 5.4.2 Técnicas para alcanzar QoS

---

- Hay muchas maneras de dar el servicio con calidad
- Ninguna técnica por sí sola proporciona QoS de una manera total
- Con frecuencia se combinan varias técnicas

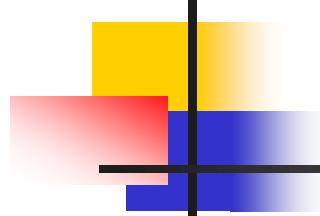
- 
- Algunas de las técnicas son:
    - Sobre aprovisionamiento o sobre dimensionamiento
    - Almacenamiento en búfer
    - Modelado de tráfico
    - Algoritmo de cubeta con goteo
    - Algoritmo de cubeta con *tokens*
    - Reservación de recursos
    - Control de admisión
    - Enrutamiento proporcional, y
    - Calendarización de paquetes



## 1/9. Sobreaprovisionamiento

---

- Para que los paquetes fluyan con facilidad, la solución fácil es dar **suficiente**: capacidad de CPU; memoria; y, ancho de banda
- Pero, esta solución es costosa
- ¿Cuánto es suficiente?. Esta técnica es práctica cuando se tiene experiencia
- Una comparación con el sistema telefónico convencional: siempre hay tono
- Con el sistema celular: siempre comienza a timbrar en el otro móvil



## 2/9. Almacenamiento en búfer

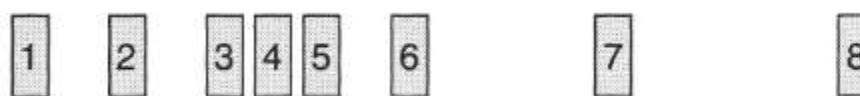
---

- Los flujos pueden almacenarse en el receptor (destino final) antes de ser desplegados al usuario
- Esto no afecta la confiabilidad o el ancho de banda
- Esta técnica atenúa la **fluctuación**
- Para **video** o **audio** bajo demanda esta técnica es muy útil
- Usado por empresas como YouTube, Netflix, Prime Video, . . .

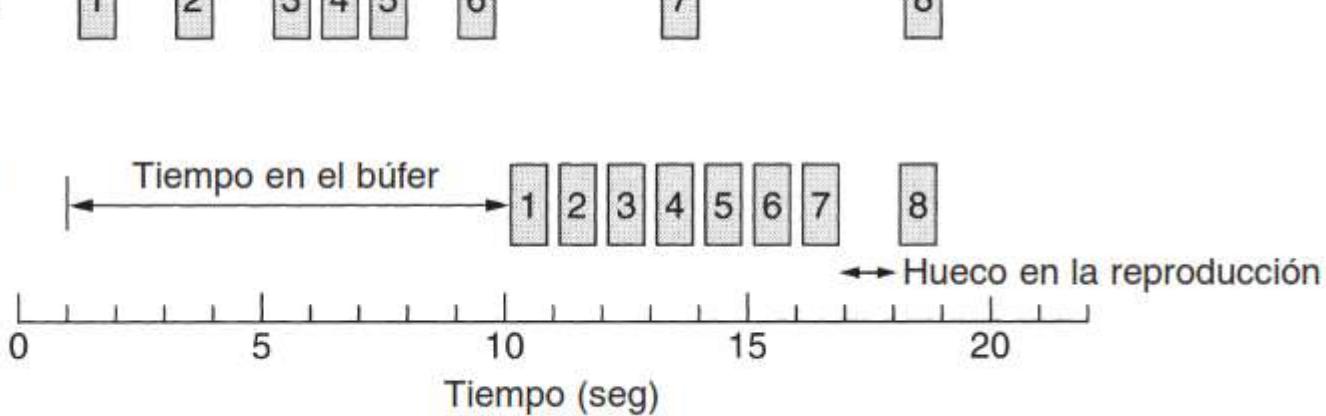
El paquete sale del origen



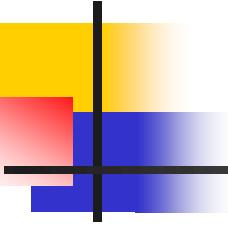
El paquete llega al búfer



El paquete se elimina del búfer



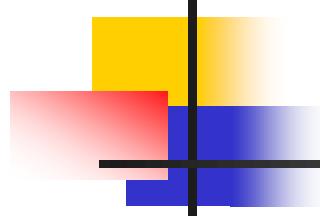
**Figura 5-31.** Refinamiento del flujo de paquetes almacenándolos en el búfer.



- Sitios web comerciales con transmisión continua de video o audio usan reproductores que almacenan en el búfer por aproximadamente **10 segundos** antes de comenzar a reproducir

## 3/9 Modelado de tráfico

- - Hay casos en que el origen envía paquetes con un **espaciado no uniforme**
  - Esto se debe a que el servidor maneja muchos flujos al mismo tiempo, o porque se permite avance rápido y rebobinado, etc.
  - El almacenamiento en el búfer es adecuado en VoD pero no para videoconferencia (Skype, Zoom, Google Meet, ...)
  - Surge entonces el modelado de tráfico
  - El cliente le dice al ISP el patrón de tráfico que va a generar
  - Si el ISP lo puede manejar se llega a un acuerdo de servicio
  - Entonces se establece la conexión
  - Mientras el cliente cumpla con lo acordado, la empresa entrega a tiempo

- 
- Estos acuerdos **no** son importantes para enviar archivos, **sí** para datos en **tiempo real** (videoconferencia)
  - El ISP debe saber si el cliente está cumpliendo con su parte del acuerdo
  - Para ello la empresa **supervisa** el tráfico
  - La supervisión es más fácil hacerlo con circuitos virtuales que con datagramas

## 4/9 Algoritmo de cubeta con goteo

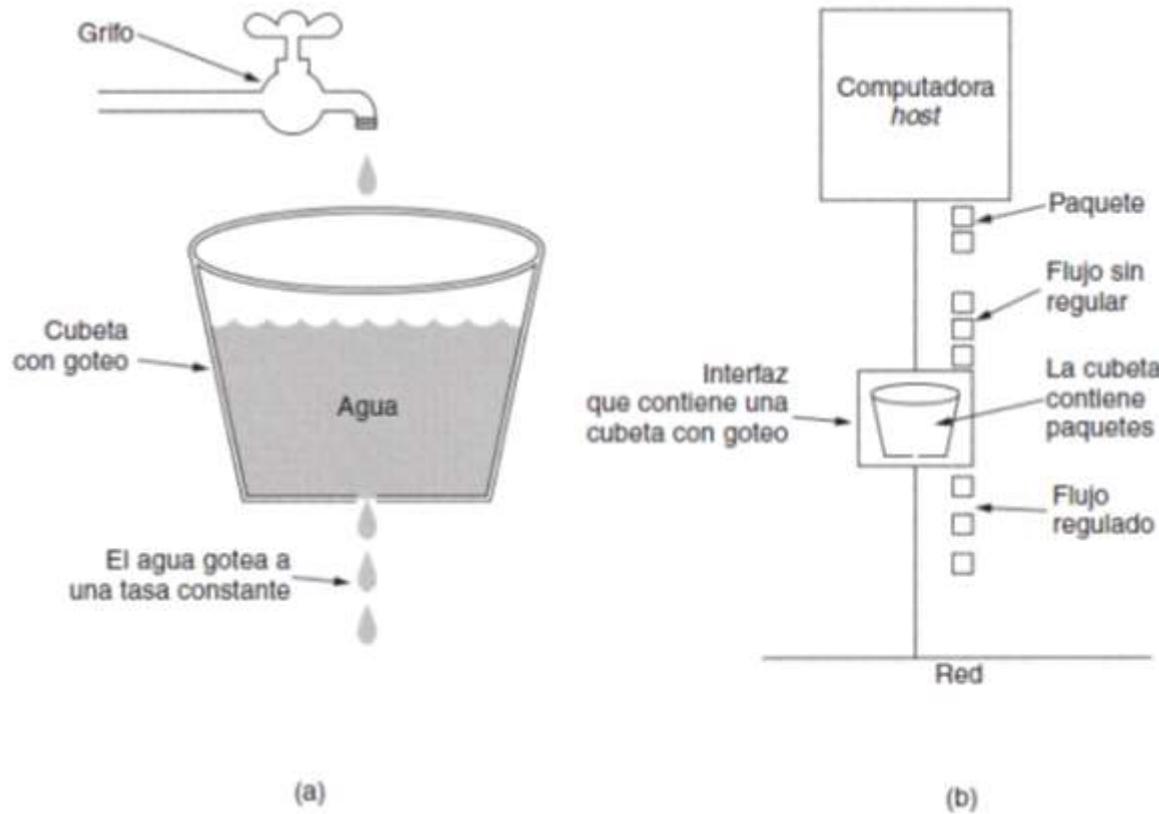
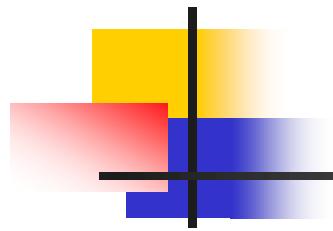
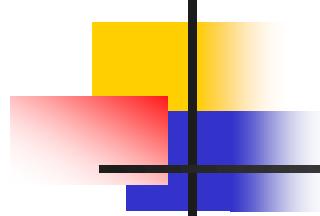


Figura 5-32. (a) Una cubeta con goteo, llena de agua. (b) Cubeta con goteo, llena de paquetes.

- Imagínese un balde de agua con un agujero en el fondo
- Sin importar la rapidez con la que entra agua al balde, el flujo de salida tiene una tasa constante  $\rho$  cuando hay agua en el balde
- Si el balde está vacío,  $\rho = 0$ . Si está lleno, el agua adicional se riega
- Puede aplicarse el mismo concepto a los paquetes
- Es un sistema de cola de un servidor con un tiempo de servicio constante
- Esta técnica convierte un **flujo desigual** de paquetes de los procesos de usuario dentro del host en un **flujo continuo** hacia la red

- 
- De esta manera se **moderan las ráfagas** y se reduce la congestión
  - Cuando los paquetes son del mismo tamaño (redes ATM), este algoritmo puede usarse tal cual
  - Si los paquetes son de tamaño variable, se permite un número fijo de **bytes** por pulso de reloj
  - Esta técnica requiere de una **cola finita**



## 5/9 Algoritmo de cubeta con *tokens*

---

- En ocasiones es necesario permitir ráfagas de tráfico
- El algoritmo de cubeta con goteo no lo permite
- Entonces se requiere un algoritmo más flexible
- Éste es el algoritmo de cubeta con tokens

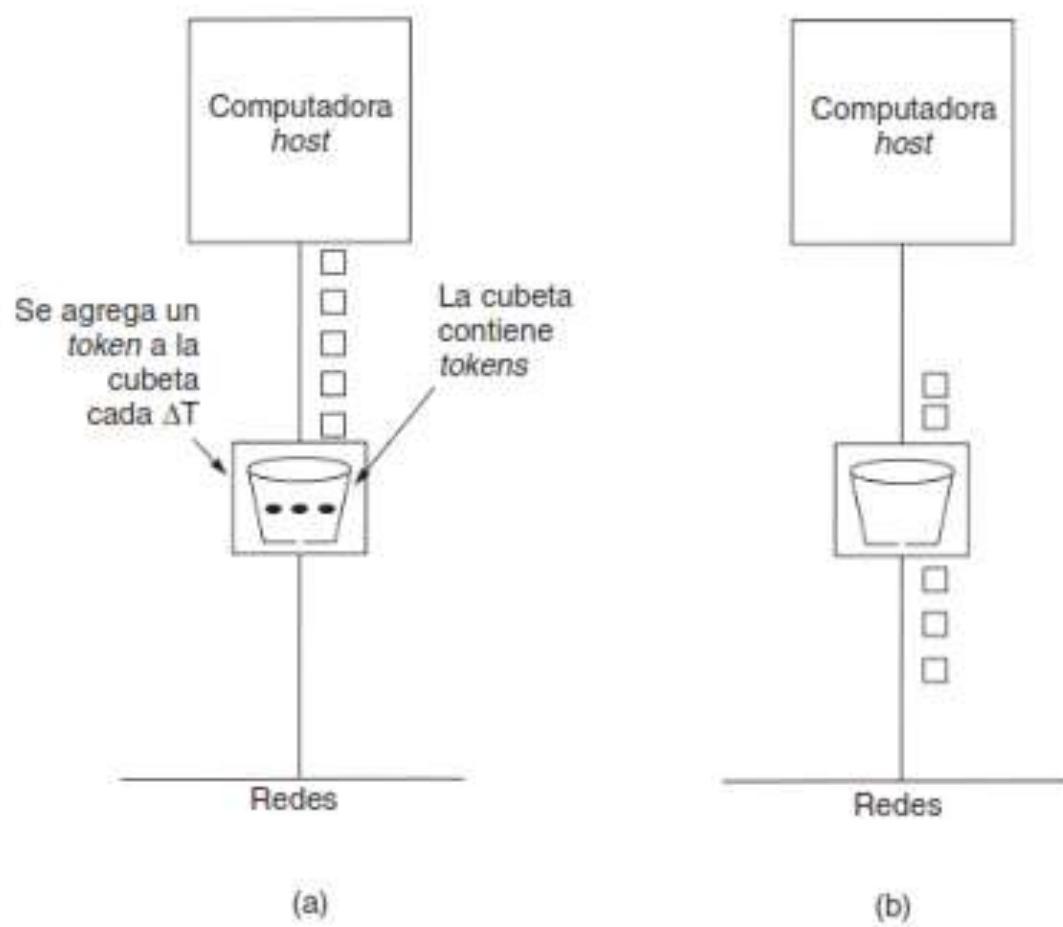
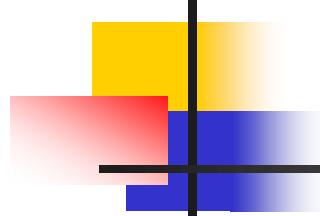


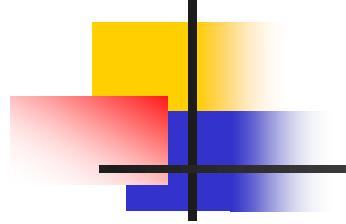
Figura 3-34. Algoritmo de cubeta con *tokens*. (a) Antes. (b) Después.

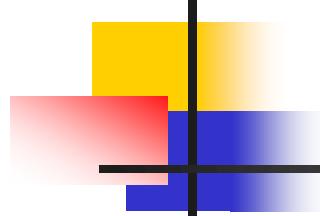
- El balde contiene tokens generados por un reloj a razón de uno cada  $\Delta T$
- Para transmitir un paquete, éste toma un token y lo destruye
- Se permite que un host acumule  $n$  tokens hasta que se llene el balde
- Esto permite que puedan enviarse ráfagas de hasta  $n$  paquetes
- Una variante es que un token da el derecho a transmitir  $k$  **bytes**
- Se envía un paquete si hay suficientes tokens que cubran su longitud en bytes
- Los algoritmos de cubeta con goteo y con tokens sirven para regular el tráfico de hosts y de ruteadores



## 6/9 Reservación de recursos

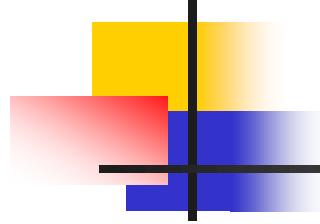
- Para reservar recursos es necesario que los paquetes de un mismo flujo sigan la misma ruta
- Así se asegura la capacidad necesaria sea disponible
- Se pueden reservar 3 tipos de recursos: bps, memoria, tiempo de CPU
- Éste es buen inicio para garantizar QoS
- Cuando se reserva ancho de banda no se sobrecarga la línea de salida

- 
- También se puede reservar algunos búfer para un flujo específico
  - Así, el flujo no tiene que competir con otros flujos para obtener espacio en búfer
  - Por otro lado, para procesar un paquete se necesita tiempo de CPU
  - Un ruteador sólo puede procesar cierta cantidad de paquetes por segundo
  - Para procesar a tiempo los paquetes es necesario que el CPU no se sobrecargue
  - Se puede reservar tiempo de CPU



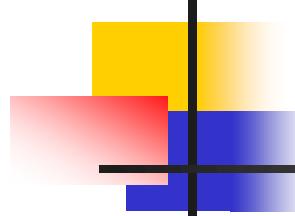
## 7/9 Control de admisión

- Si el flujo entrante en los enrutadores:
  - Está bien modelado y
  - sigue una misma ruta
- Entonces se puede solicitar reservar sus capacidades
- El ruteador analiza la solicitud con base en:
  - su capacidad de memoria, CPU y líneas de salida
  - y los compromisos adquiridos con otros flujos
- Entonces admite o rechaza un nuevo flujo

- 
- Aunque hay aplicaciones que conocen sobre los requerimientos de ancho de banda, saben poco sobre búferes o tiempos de CPU
  - Se necesita otra forma de **describir** los flujos
  - Muchas partes pueden estar involucradas en la negociación del flujo: emisor, receptor, enrutadores
  - Los flujos deben especificarse de forma precisa con parámetros negociables

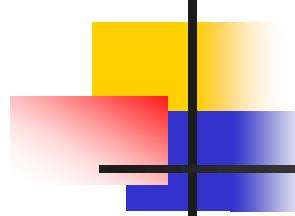
Parámetro	Unidad
Tasa de la cubeta con tokens.	Bytes/seg
Tamaño de la cubeta con tokens.	Bytes
Tasa pico de datos.	Bytes/seg
Tamaño mínimo de paquete.	Bytes
Tamaño máximo de paquete.	Bytes

**Figura 5-32.** Ejemplo de una especificación de flujo.



## 8/9 Enrutamiento proporcional

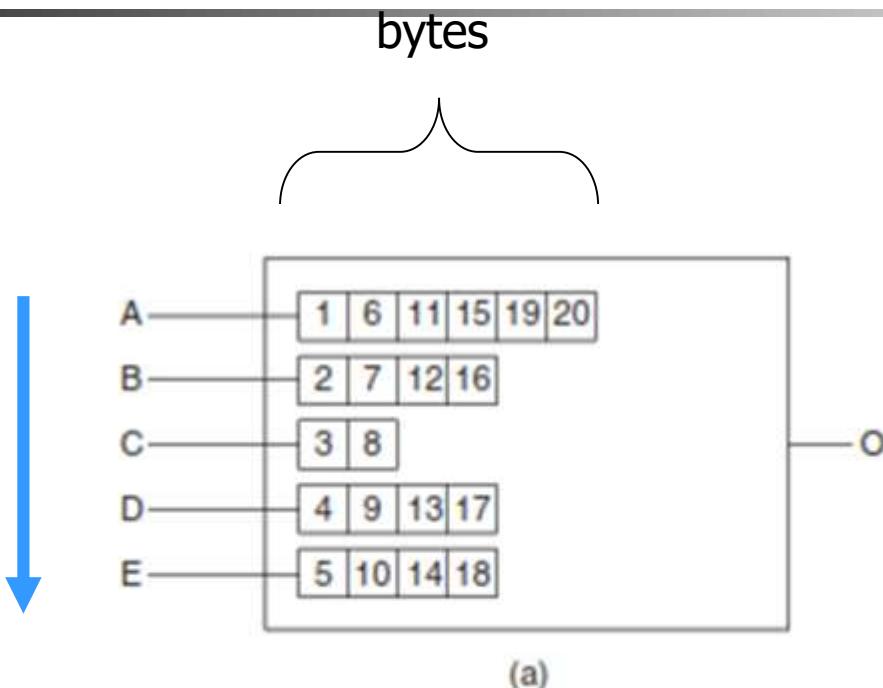
- La mayoría de los protocolos de enrutamiento envían **todo** el tráfico por la mejor ruta que encuentran para destino
- Para dar una mayor QoS se **divide** el tráfico para a destino a través de diferentes rutas
- Un método de hacerlo es dividir el tráfico:
  - en fracciones iguales, o
  - en proporción de la capacidad de los enlaces salientes



## 9/9 Calendarización de paquetes

- Hay flujos acaparan la capacidad de un router, limitando a los otros flujos
- Para evitar esto, existe el algoritmo llamado *encolamiento justo*
- Con este algoritmo los ruteadores tienen una cola para cada flujo
- El enrutador explora las diferentes colas de manera circular – *round robin*
- Este algoritmo tiene el problema de que da más ancho de banda a los hosts que utilizan paquetes más grandes
- Un mejora es hacer la exploración circular byte por byte en vez de paquete por paquete

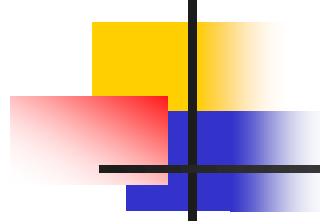
Numeración  
de bytes



Paquete	Tiempo de terminación
C	8
B	16
D	17
E	18
A	20

(b)

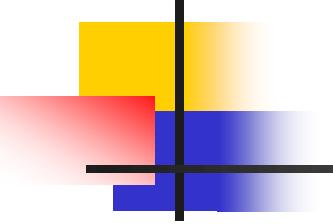
**Figura 5-36.** (a) Un enrutador con cinco paquetes encolados en la línea  $O$ . (b) Tiempos de terminación de los cinco paquetes.

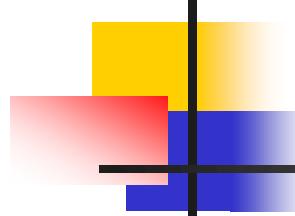
- 
- Un problema con este algoritmo es que da la misma prioridad a todos los hosts
  - Por ello se usa el algoritmo "*encolamiento justo ponderado*"
  - A cada flujo se le da diferente peso





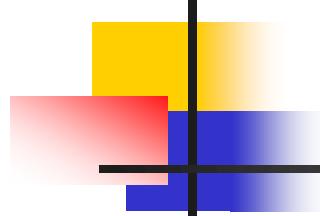
## 5.5 Interconexión de redes

- 
- Hasta ahora hemos considerado una sola red en donde cada máquina usa el mismo protocolo en cada capa
  - Pero existe una variedad de redes que usan diferentes protocolos en cada capa que son incompatibles entre sí
  - Estas redes pueden ser: PAN, LAN, WLAN, WAN, telefónica móvil, . . .
  - Estas distintos tipos de redes se pueden interconectar entre sí para formar una interred o internet
  - La interconexión a nivel mundial de redes se llama Internet
  - El propósito de interconectar distintas redes es permitir que usuarios de una red se puedan comunicar con usuarios de otras redes



## Diferencias entre redes

- Modulación, codificación o representación de bits en la capa física
- Formato de tramas en la capa de enlace
- En la capa de red
  - Tipo de servicio: sin conexión y orientado a conexión
  - Multidifusión o no
  - Tamaños máximos de paquetes
  - Diferentes QoS. Difícil garantizar ancho de banda o retardo en aplicaciones de tiempo real
  - Mecanismo de seguridad: capas de red y aplicación
- Mecanismos de contabilidad dan lugar a distintas formas de facturación
- Hay que conciliar estas diferencias para lograr la interconexión de redes



## Interconexión de redes

- Se lo puede hacer de dos maneras:
  1. A través de dispositivos que traducen los paquetes de la una red en paquetes de la otra red
  2. O, construir una capa común sobre las otras capas de cada red para ocultar las diferencias de las redes
- Esta capa común se separó luego en los protocolos TCP e IP

<i>Aplicación</i>
<i>Transporte</i>
<i>IP</i>
<i>Ethernet</i>
Puerto RJ-45

<i>IP</i>		
<i>Ethernet</i>	<i>HDLC</i>	<i>802.11</i>
Puerto RJ-45	Puerto serie	Antena

Dirección de red o subred 1

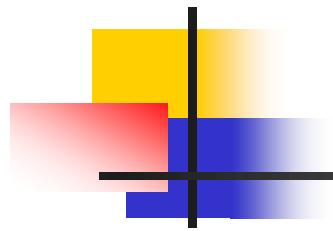
<i>Aplicación</i>
<i>Transporte</i>
<i>IP</i>
<i>802.11</i>
Antena

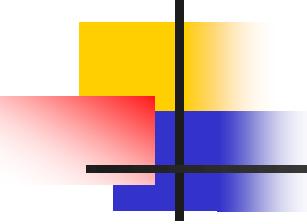
*IP*

<i>IP</i>	
<i>HDLC</i>	<i>802.11</i>
Puerto serie	Antena

Dirección de red o subred 2

Dirección de red o subred 3

- 
- IP ha extendido su alcance pasando desde las redes de computadoras hacia la telefonía móvil y red de sensores
  - Los hubs operan en la capa física y solo transfieren bits desde una cable a otro
  - Los switches operan en la capa de enlace y permiten la interconexión de PCs con diferentes velocidades 10, 100, 1000 Mbps
  - Los equipos de interconexión de distintos tipos de redes operan en la capa de red y son los enrutadores
  - Las tramas en cada tipo de red manejan diferentes tamaños máximos, así que posiblemente se tengan que fragmentar en tramas más pequeñas

- 
- Recalcamos que la interconexión de redes solo funciona cuando hay una capa de red común (IP)
  - IP es el protocolo de red universal, pero hay otros: IPX de Novell Netware, SNA de IBM, AppleTalk, . . . que casi ya no se usan
  - IP ha dado lugar a IPv4 e IPv6 que no son compatibles
  - IPv6 es la base de Internet 2, una red de alta velocidad utilizada para investigación y pruebas

# Tunelización

- Podría llamarse también encapsulación
- Esto se lo puede hacer si los hosts origen y destino tienen el mismo tipo de red con una red diferente en el medio

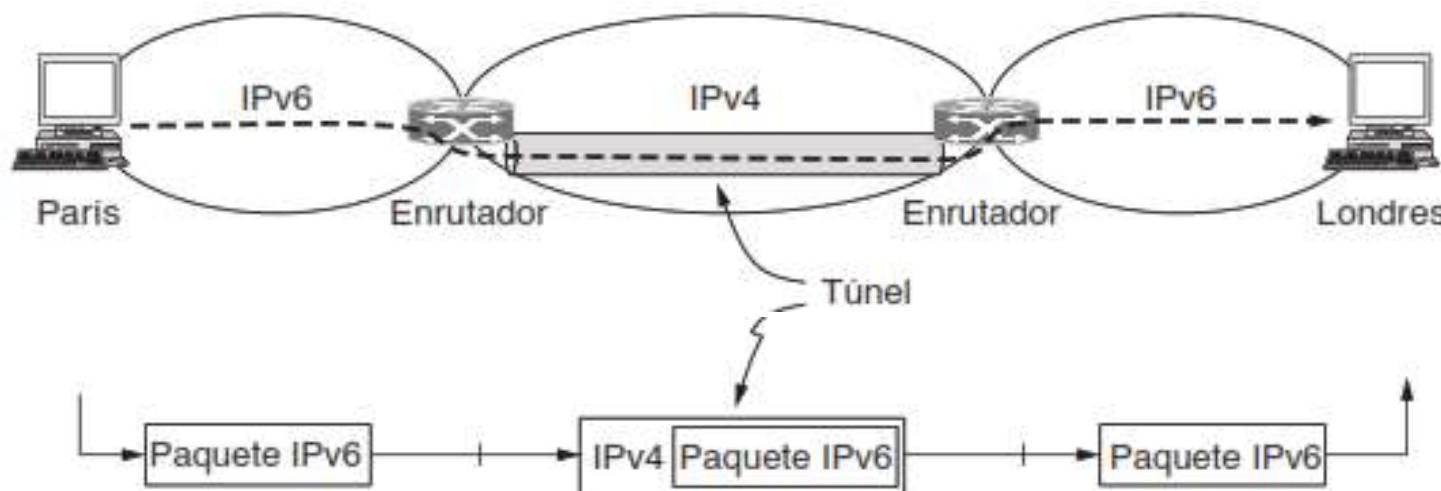
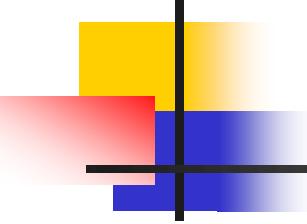


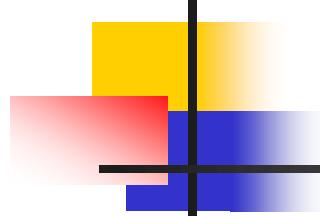
Figura 5-40. Tunelización de un paquete de París a Londres.

- Para hacer tunelización se requiere que los enrutadores sean multiprotocolo a nivel de capa de red: IPv4 e IPv6
- Una analogía interesante es la siguiente:



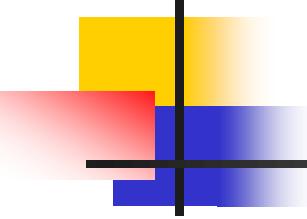
Figura 5-41. Tunelización de un auto de Francia a Inglaterra.

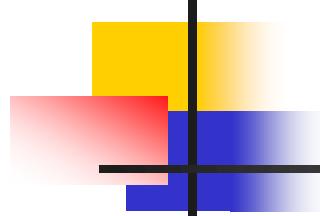
- 
- La tunelización se usa para conectar hosts y redes aisladas a través de otras redes
  - La red resultante se llama *red overlay* y se superpone sobre la red base
  - Esta característica se usa para crear redes *VPN Virtual Private Networks*
  - Una VPN es una red overlay que se usa para dar seguridad



## Enrutamiento entre redes

- Las redes interconectadas pueden utilizar diferentes protocolos de enrutamiento, lo cual es un problema
- Una red podría usar enrutamiento por estado de enlace y la otra enrutamiento por vector distancia
- Esto dificulta la determinación de rutas más cortas
- También, los operadores de las distintas redes pueden tener diferentes criterios de mejor ruta: la que tiene menos retardos, o la más económica
- Por otro lado, un operador puede que no quiera dar a conocer sus rutas

- 
- Todo esto conducen a un algoritmo de enrutamiento de dos niveles:
    - Dentro de cada red se usa un protocolo intradominio
    - En la interred se usa un protocolo interdominio
  - En internet se usa el protocolo BGP Border Gateway Protocol

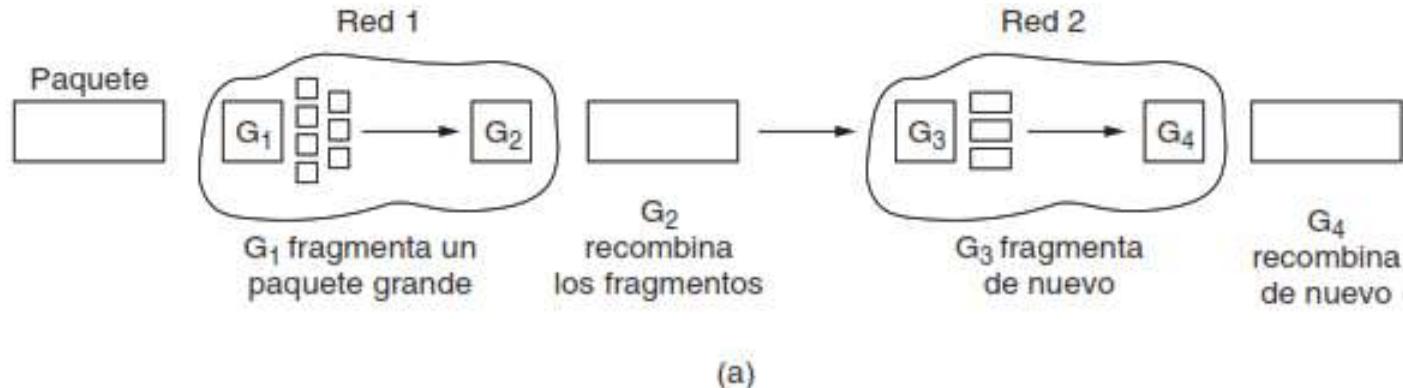


# Fragmentación de paquetes

- El tamaño máximo de paquetes tiene varias razones:
  1. El hardware (por ejemplo el tamaño de una trama ethernet)
  2. El sistema operativo (por ejemplo todos los búfers tienen tamaño 512 bytes)
  3. Necesidad de reducir las retransmisiones debido a errores
  4. Necesidad de evitar que un paquete use por demasiado tiempo el canal

- Las cargas útiles máximas para:
  - Ethernet es 1500 bytes
  - 802.11 es 2272 bytes
  - IP es 65515 bytes
- Los host prefieren transmitir paquetes grandes para ahorrar bytes de los encabezados. Problemas: más fragmentación; retransmisión de grandes paquetes debido a errores; luego la tarea de ensamblar los fragmentos
- El tamaño máximo de paquetes para llegar a un destino se llama *MTU de ruta Path Maximum Transmission Unit*
- Conocer MTU sería útil si la fuente conociera de antemano la ruta

# Fragmentación de paquetes. Dos maneras:



Todos los fragmentos de un mismo paquete siguen la misma ruta

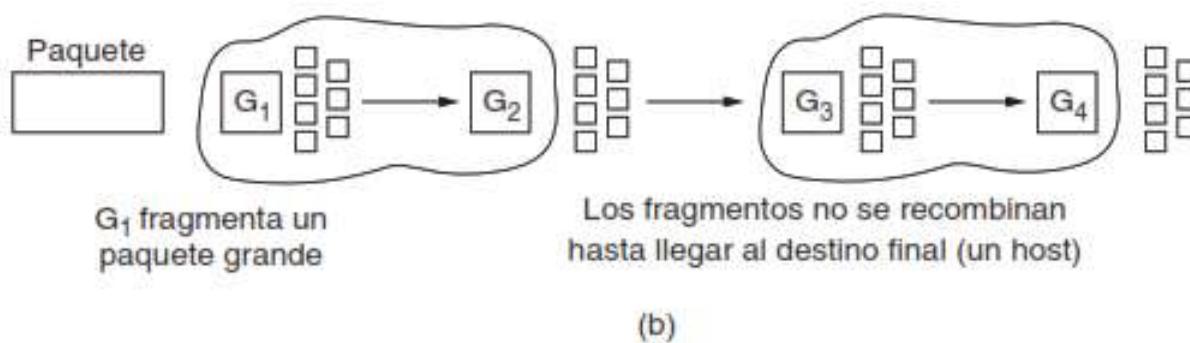
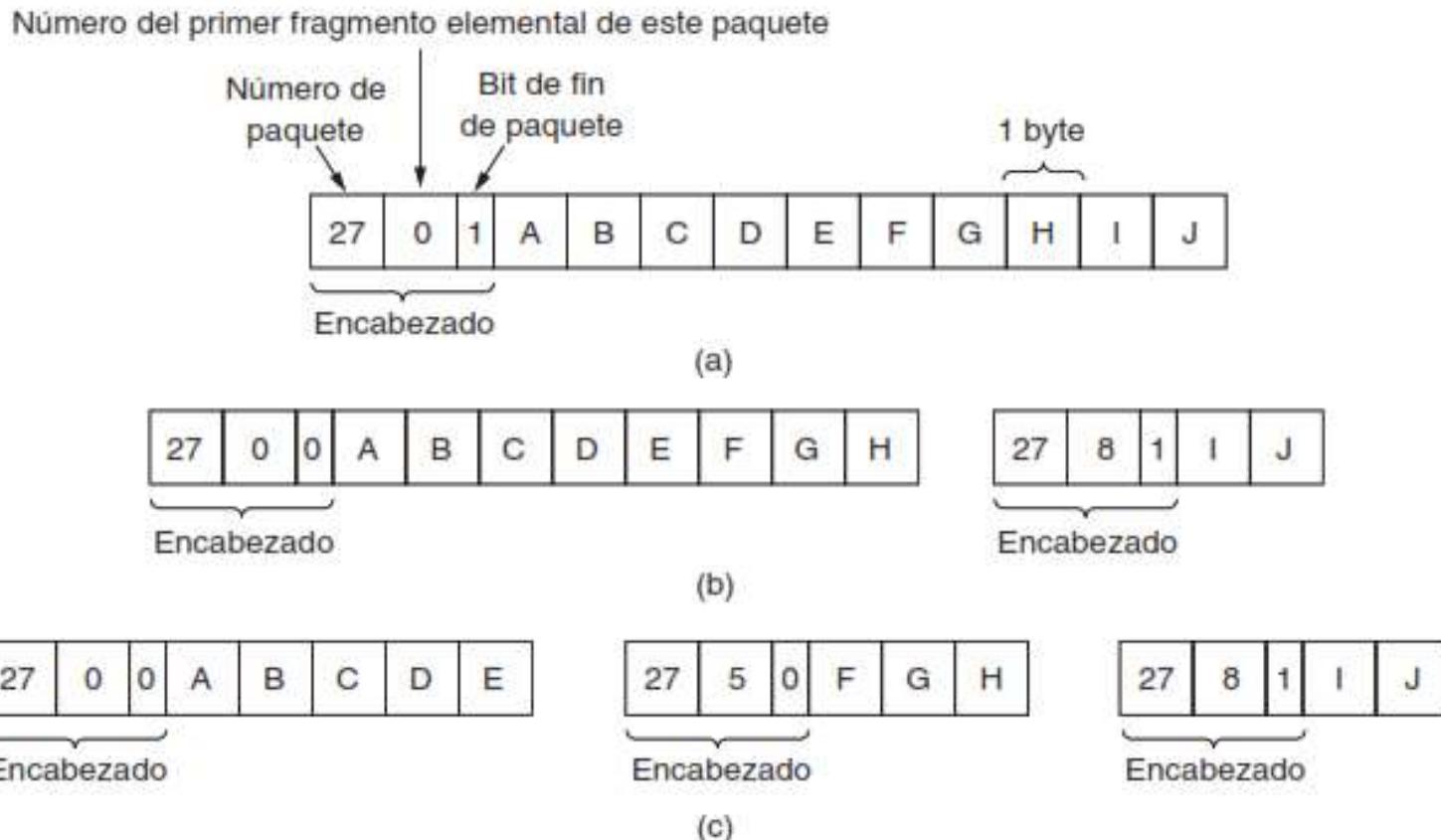


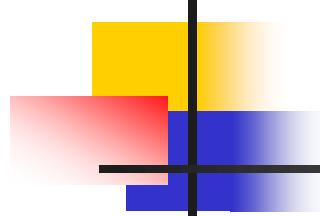
Figura 5-42. (a) Fragmentación transparente. (b) Fragmentación no transparente.

Los fragmentos de un mismo paquete podrían seguir diferentes rutas

# Fragmentación sucesiva a lo largo de una ruta

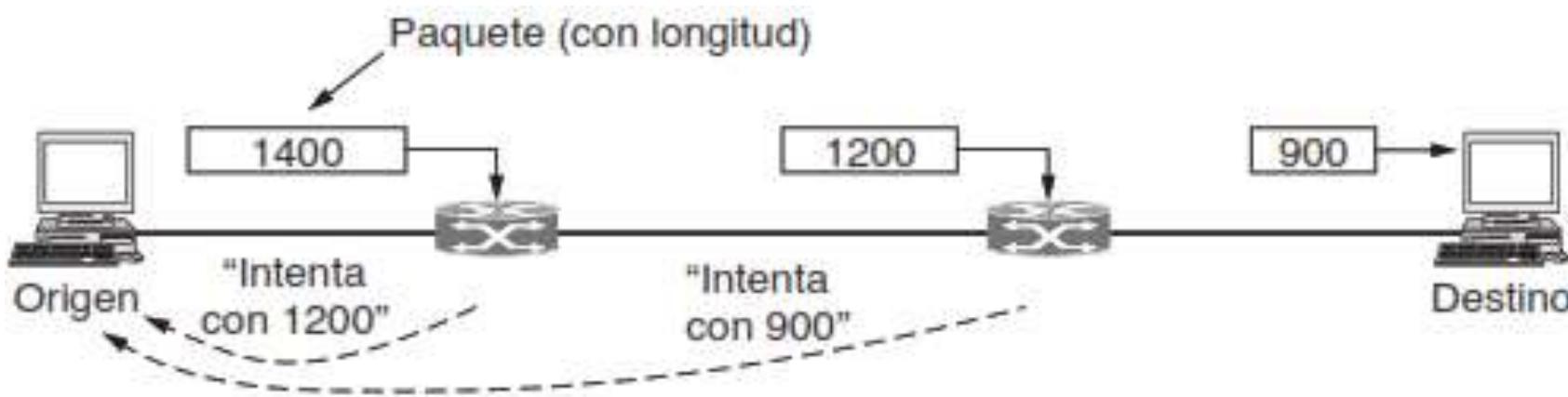


**Figura 5-43.** La fragmentación cuando el tamaño de datos elemental es de 1 byte. (a) El paquete original que contiene 10 bytes de datos. (b) Los fragmentos después de pasar por una red con un tamaño máximo de paquete de 8 bytes de carga útil más encabezado (c) Fragmentos después de pasar a través de una puerta de enlace de tamaño 5.

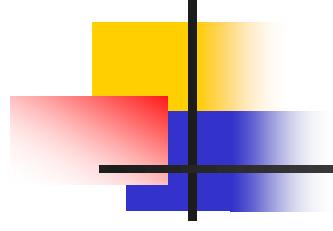


## Descubrimiento del MTU de la ruta

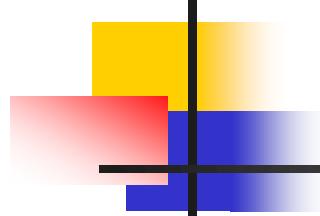
- Para evitar la fragmentación en la red, se realiza la fragmentación en el host origen
- Se envía los paquetes indicando en el encabezado que no se pueden fragmentar
- Si un router recibe un paquete demasiado grande, lo descarta y avisa al origen indicando cuál es el tamaño máximo
- Cuando el origen recibe el aviso, fragmenta el paquete y lo vuelve a enviar
- Si otro router más adelante tiene un MTU aun más pequeño, se repite el proceso
- La desventaja del MTU de ruta es que puede haber retardos al inicio



**Figura 5-44.** Descubrimiento de MTU de la ruta.

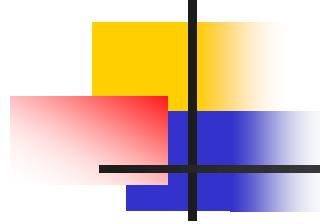


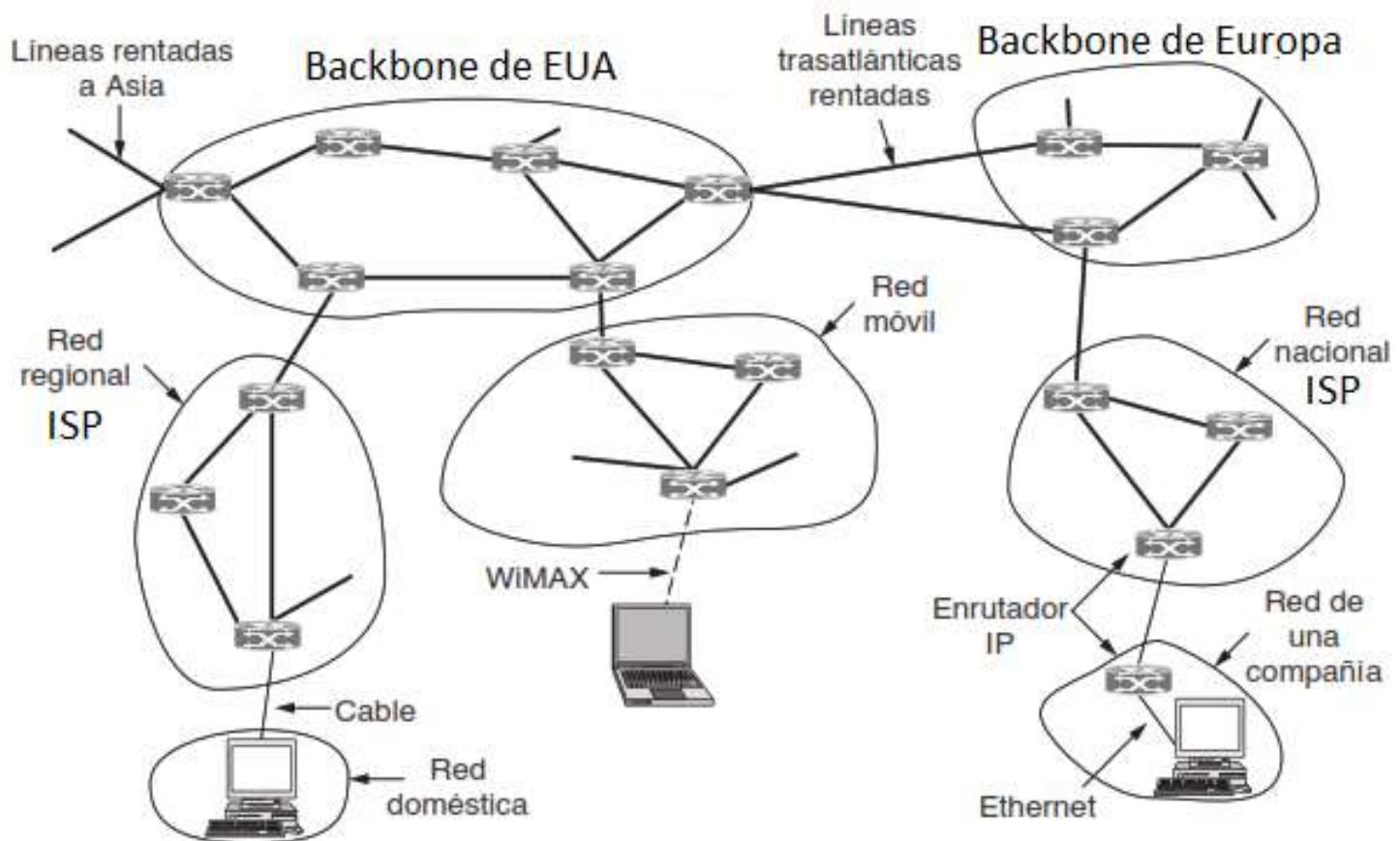
## 5.6 La capa de red de Internet



# Principios de diseño de la capa de red de Internet

1. Asegurarse de que el diseño funciona de verdad en la práctica
2. Mantener la simplicidad. "*cuando tenga duda utilice la solución más simple*". Si una característica del diseño no es esencial, descártela
3. Si hay varias maneras para realizar una tarea, elija solo una
4. Diseñe modularmente
5. Haga un solo diseño que sea flexible a diferentes tipos de hardware, facilidades de transmisión y aplicaciones
6. Incluir parámetros si son inevitables
7. No hacer un diseño que contemple hasta los casos más extraños y raros

- 
8. Diseño estricto en el cumplimiento de estándares cuando se envíe paquetes y tolerante si llegan paquetes que no los cumple
  9. Diseño debe ser escalable para millones de hosts
  10. Diseño que permite un desempeño eficiente de la red al menor costo

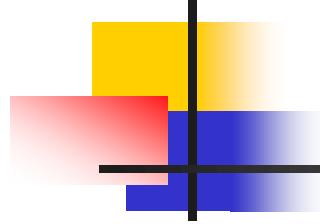


Nivel 1  
Routers y líneas muy rápidas

Redes de nivel medio

Universidades, empresas,  
instituciones, hogares

**Figura 5-45.** Internet es una colección interconectada de muchas redes.

- 
- El protocolo de la capa de red IP es lo que logra mantener interconectadas todas las redes que forman Internet
  - Justamente, IP fue diseñado con este propósito
  - En teoría, los paquetes IP pueden ser de hasta 64 KB (65536 bytes)
  - En la práctica no sobrepasan los 1500 bytes que es la capacidad de una trama Ethernet
  - Hay muchas rutas posibles entre dos hosts (figura 5.45)
  - IP tiene la tarea de decidir qué ruta seguir

# El protocolo IPv4

- Un datagrama tiene dos partes: encabezado y carga útil
- El encabezado tiene una parte fija de 20 bytes y una parte opcional de longitud variable
- Los bits se transmiten de izquierda a derecha y de arriba abajo

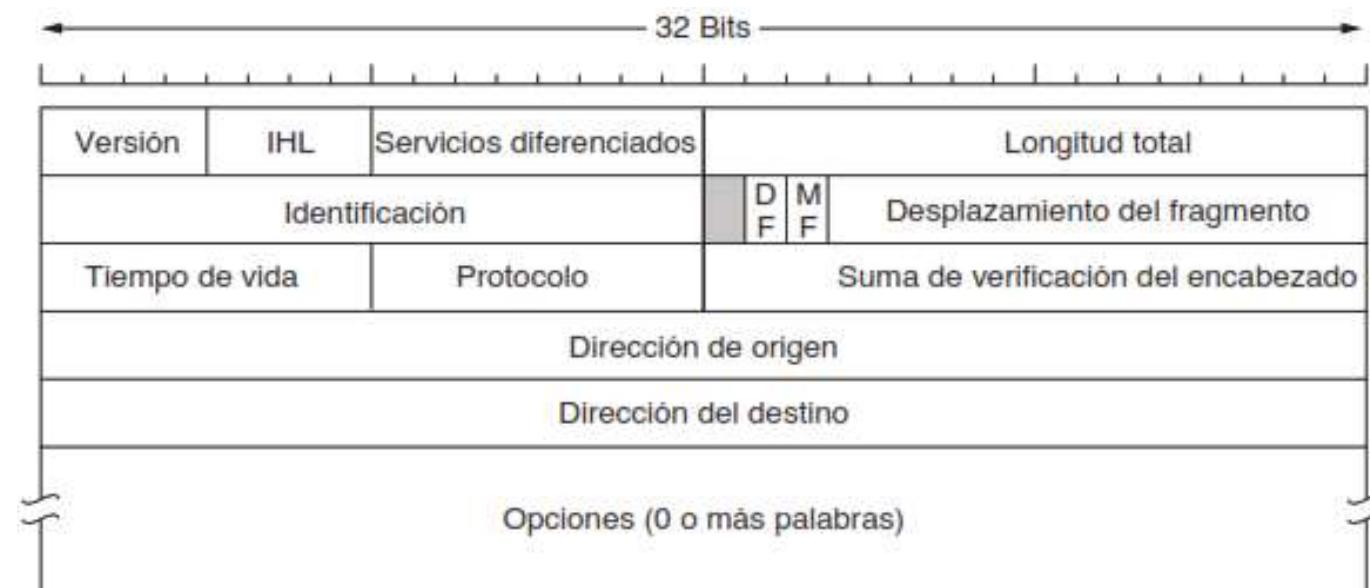
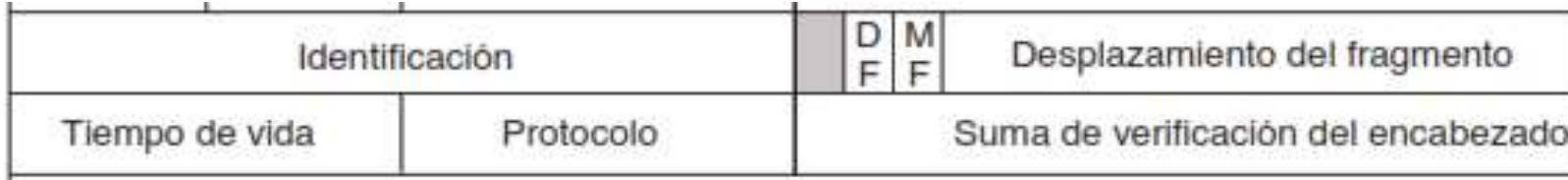


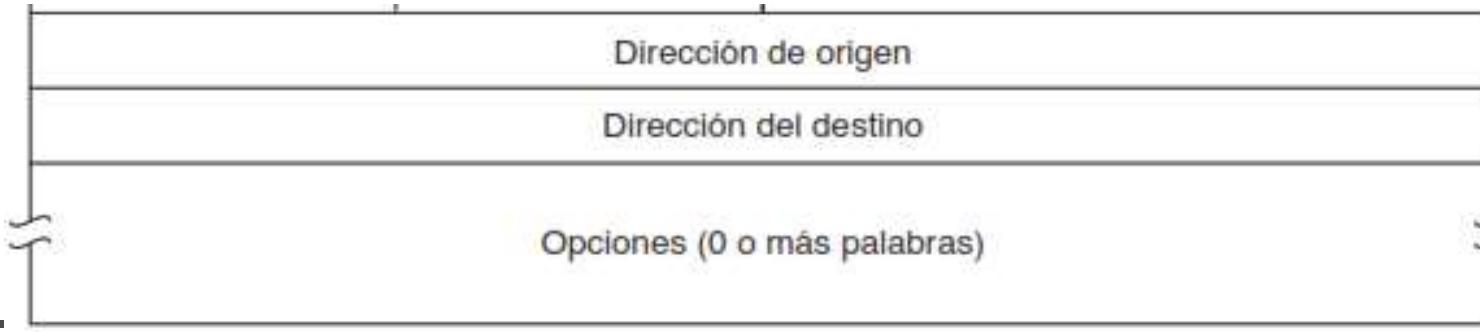
Figura 5-46. El encabezado de IPv4 (Protocolo de Internet).



- **Versión.** Indica la versión de IP. La que hoy domina Internet es IPv4
- La siguiente versión es IPv6, definida ya hace muchos años, apenas se usa
- IPv5 fue un protocolo experimental para tiempo real que no tuvo éxito
- **IHL.** Ya que la longitud del encabezado es variable, IHL indica esta longitud expresada en palabras de 32 bits. Lo mínimo es 5 (20 bytes = 160 bits)
- **Servicios diferenciados.** Para voz digital, lo rápido es más importante que lo preciso. Para envío de archivos lo preciso es más importante que lo rápido
- **Longitud total.** Incluye todo el datagrama: encabezado y datos. La longitud máxima es  $2^{16} - 1 = 65535$



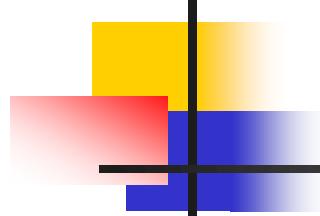
- **Identificación.** Indica a qué paquete pertenece un fragmento. Los fragmentos de un mismo paquete tienen la misma identificación
- **Bit sin uso**
- **DF.** *Don't fragment*, es una orden para los enrutadores
- **MF.** *More fragments*, Todos los fragmentos excepto el último tienen encendido este bit. Esto es necesario para saber que ha llegado el último fragmento
- **Desplazamiento del fragmento.** Indica a qué parte del paquete actual pertenece este fragmento
- **Tiempo de vida.** TTL, Contador de saltos de un paquete en la red para evitar que vague sin fin cuando se corrompen las tablas de ruteo
- **Protocolo.** de transporte al que se debe entregar el paquete: TCP o UDP
- **Suma de verificación del encabezado.** Checksum para detectar errores



- **Dirección de origen y dirección del destino.** Indican las direcciones IP de las interfaces de red del emisor y receptor
- **Opciones.** Creado pensando en las versiones futuras de IP para que lo puedan utilizar. Actualmente no se utiliza. Algunas de ellas son:

Opción	Descripción
Seguridad.	Especifica qué tan secreto es el datagrama.
Enrutamiento estricto desde el origen.	Proporciona la ruta completa a seguir.
Enrutamiento libre desde el origen.	Proporciona una lista de enrutadores que no se deben omitir.
Registrar ruta.	Hace que cada enrutador adjunte su dirección IP.
Estampa de tiempo.	Hace que cada enrutador adjunte su dirección y su etiqueta de tiempo.

Figura 5-47. Algunas de las opciones del protocolo IP.



## Direcciones IP

---

- Una dirección IP no se refiere a un host sino una interfaz de red
- En la práctica un host tiene una interfaz de red y un router tiene varias
- A diferencia de las direcciones de ethernet, las direcciones IP son jerárquicas: porción de red y porción de host
- Las direcciones jerárquicas tienen ventajas y desventajas
- La ventaja es que los enrutadores pueden enrutar con base a la porción de red o prefijo, así las tablas de ruteo son mucho más pequeñas
- Desventajas: La dirección IP de un host depende de su ubicación en la red. Una dirección ethernet puede usarse en cualquier parte del mundo
- Otra desventaja es que un sistema jerárquico desperdicia direcciones

# Subredes

- Para evitar conflictos, los números de red se administran a través de una empresa sin fines de lucro: ICANN
- ICANN ha delegado parte de las direcciones a varias autoridades regionales, las cuales reparten las direcciones a los ISPs
- Un bloque de direcciones internamente puede ser repartido entre varias redes, pero actuar como una sola red ante el mundo exterior
- Cada parte es una subred

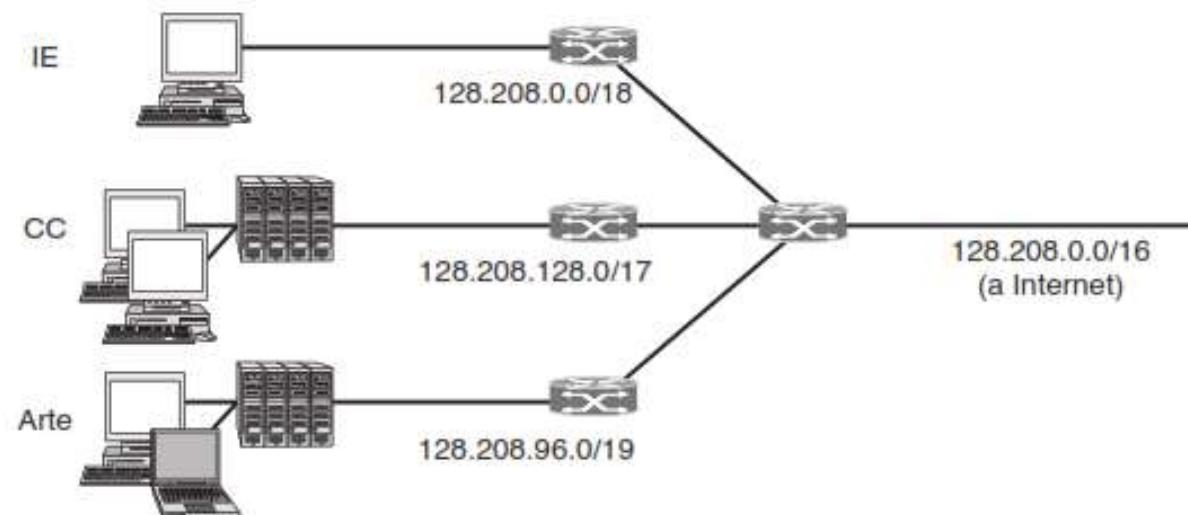
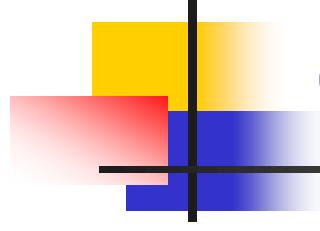


Figura 5-49. División de un prefijo IP en redes separadas mediante el uso de subredes.



# Capítulo 6

## La capa de transporte CT



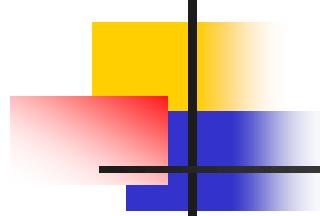
# Contenido

---

- 6.1 El servicio de transporte
- 6.2 Elementos de los protocolos de transporte
- 6.3 El protocolo de Internet UDP
- 6.4 El protocolo de Internet TCP

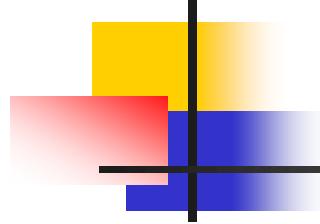


## 6.1 El servicio de transporte



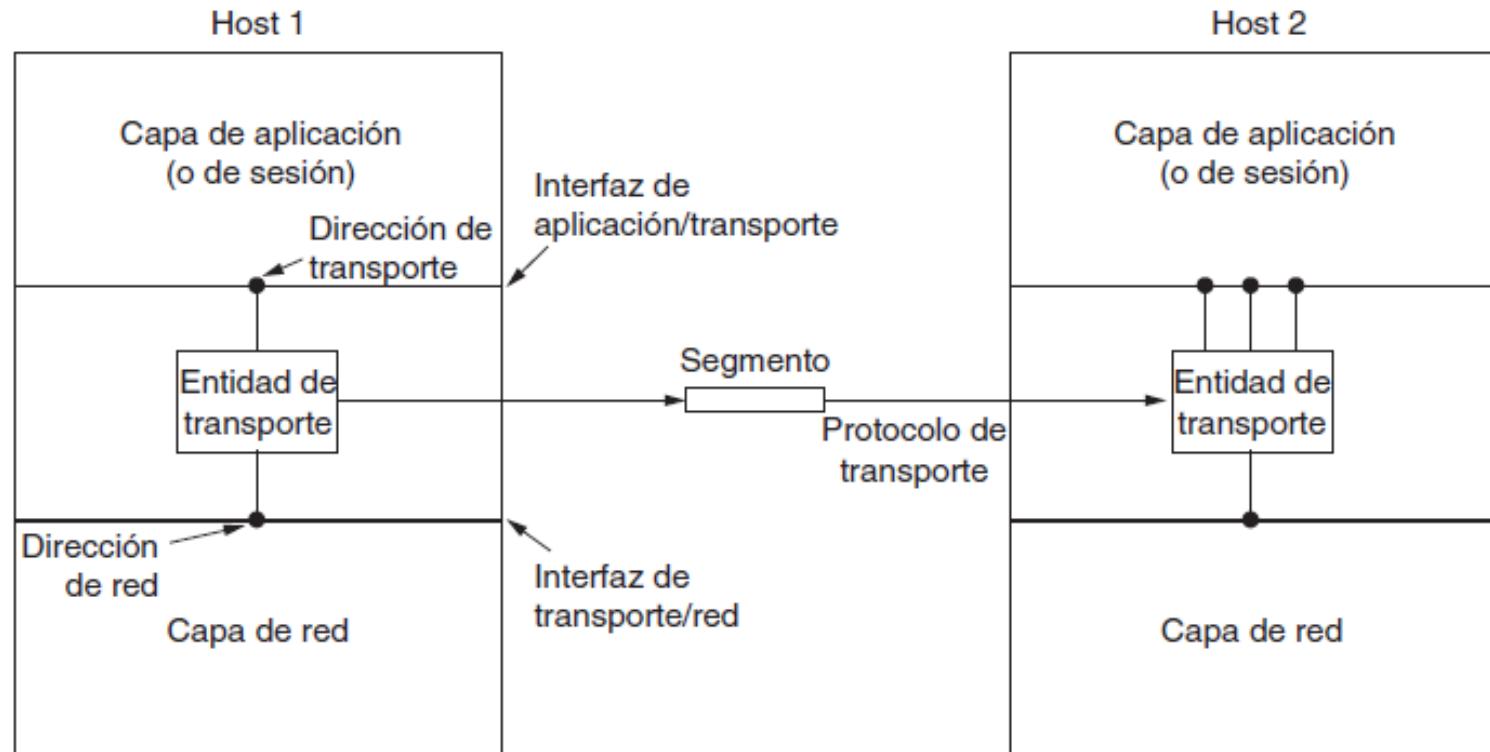
## 6.1.1 Servicios proporcionados a las capas superiores

- Conocer el funcionamiento de la CT es especialmente útil para quienes desarrollan aplicaciones de red
- Esta capa da un transporte de datos confiable **extremo a extremo** independientemente de la o las redes físicas
- Los usuarios de los servicios de la CT son los procesos de la capa de aplicación
- Para esto, la CT recibe los servicios de la capa de red CR

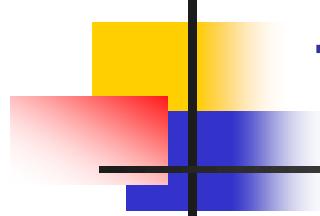


## Entidad de transporte

- La CT está implementada en hardware y en software
- Esta entidad puede estar en:
  - El kernel del sistema operativo
  - Un proceso de usuario independiente
  - Una librería de las aplicaciones de red, o
  - La tarjeta de red

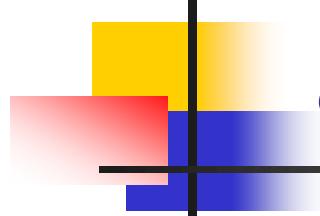


**Figura 6-1.** Las capas de red, transporte y aplicación.



## Tipos de servicio de la CT

- Similar a los servicios de la CR, hay dos tipos de servicio que se ofrecen a la capa de aplicación CA:
  - orientado a la conexión
  - sin conexión

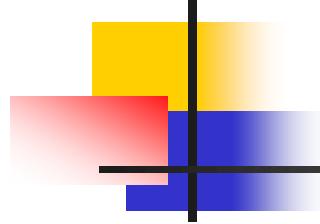


## ¿Dos capas que ofrecen los mismos servicios?

- La capa de red CR y la CT dan servicios sin conexión y orientado a conexión
- La CR se ejecuta en los enrutadores de los operadores
- Los usuarios finales no controlan la CR
- El proveedor a veces no cumple el compromiso de dar un servicio confiable:
  - pierden paquetes, dañan paquetes, o la red se cae
- Se tiene que poner otra capa sobre la CR para mejorar la calidad del servicio
- Si accidentalmente se pierde una conexión de red, la CT establece automáticamente otra conexión de red con la CT remota

- La CT facilita a los usuarios crear aplicaciones de red sin tener que lidiar con la subred
- Subred: medios de transmisión y dispositivos de los proveedores del servicio de transmisión
  - Empresas públicas de telecomunicaciones PSTN
  - ISPs

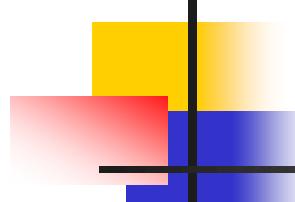




## 6.1.2 Primitivas de servicio de transporte

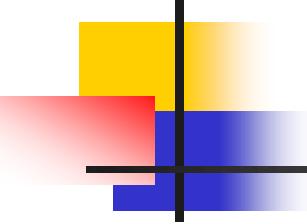
- Para que las aplicaciones puedan acceder al **servicio de transporte**, la CT facilita algunas primitivas de servicio
- Primitivas del servicio de orientado a conexión

Primitiva	Paquete enviado	Significado
LISTEN	Ninguno	Se bloquea
CONNECT	Req	Intenta establecer
SEND	Datos	
RECEIVE	Ninguno	Se bloquea
DISCONNECT	Req	Intenta liberar



# Diferencias entre los servicios de transporte y de red

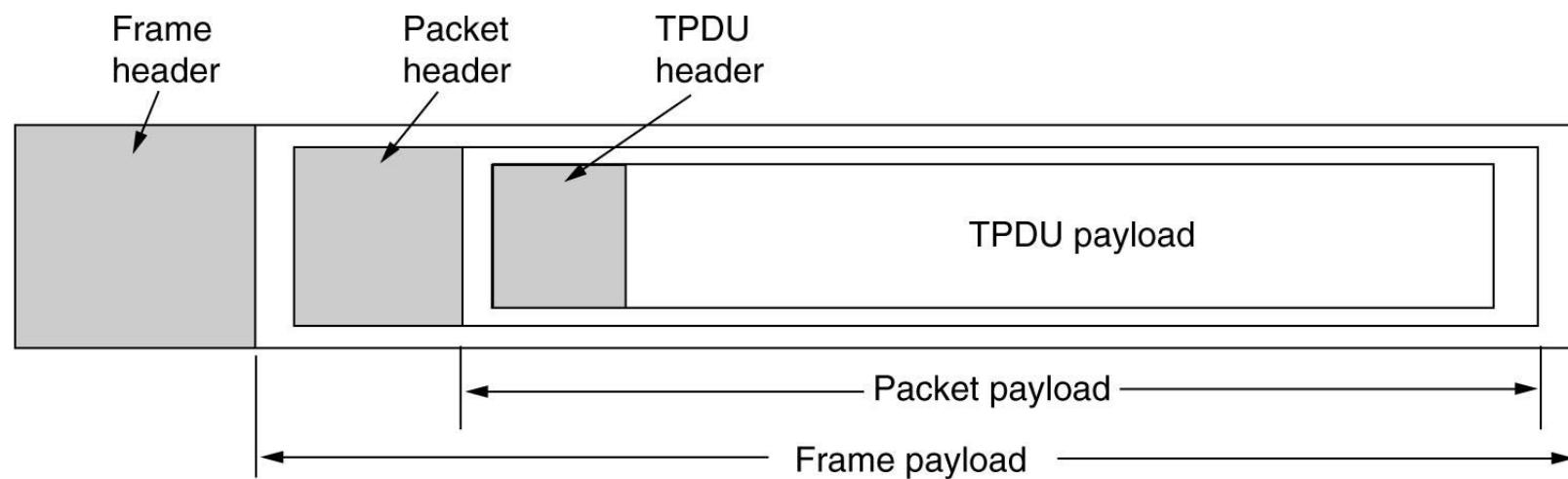
- **Primera diferencia**
- El servicio de la CR depende de las redes físicas. Estas pierden paquetes, producen errores, o se caen. Servicio no confiable
- En cambio, el servicio de CT orientado a conexión sí es confiable
- CT da un servicio confiable sobre una red no confiable
- CT oculta los defectos de la CR para que los procesos de usuario sientan un flujo de bits libre de errores y pérdidas
- **Segunda diferencia**
- Los programadores tratan con las primitivas de transporte, que son fáciles de usar. Casi nunca con las primitivas de red



# Funciones de las entidades de transporte

1. Confirmaciones ACKs
  2. Controles de errores y pérdida de paquetes
  3. Manejo de temporizadores
  4. Retransmisiones
- Son funciones transparentes para las aplicaciones

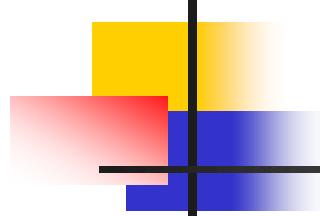
# Anidamiento





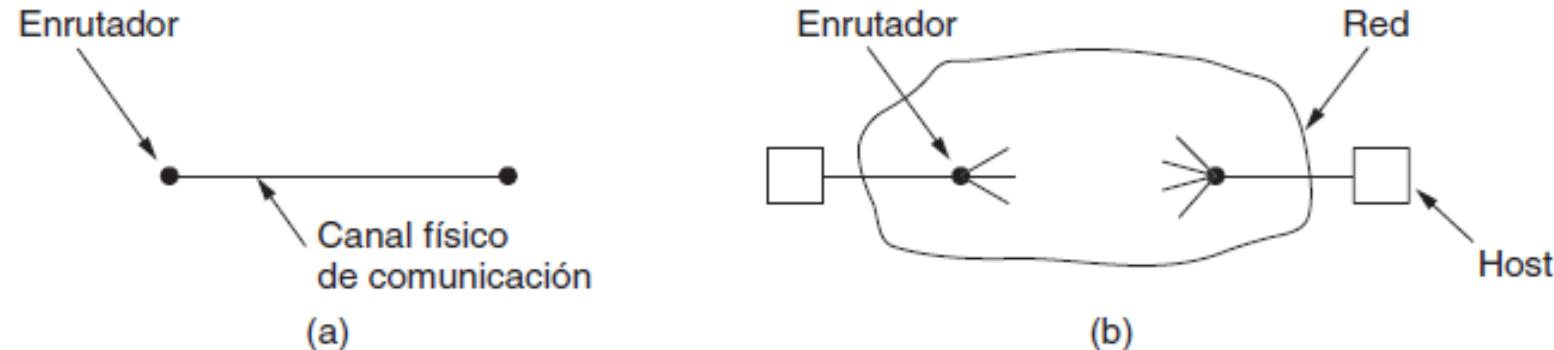
## 6.2 Elementos de los protocolos de transporte

- 
- El **servicio** de transporte se implementa mediante un **protocolo** de transporte
  - Los protocolos de capa 4 se parecen a los protocolos de capa 2 porque controlan:
    - Errores
    - Secuenciación
    - Flujo

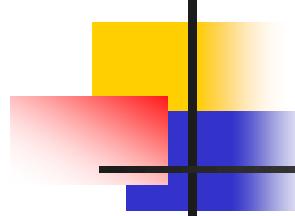


## Diferencias entre los dos protocolos

- En la capa 2, dos sistemas se conectan P2P a través de un enlace físico
- En la capa 4 los dos sistemas se conectan *end-to-end* a través de una subred
- En capa 2 el enrutador no necesita especificar la dirección del otro enrutador
- En capa 4 hay que indicar la dirección destino
- En capa 2 es más sencillo que en capa 4 establecer conexiones
- La subred almacena y reenvía paquetes. Esto produce complicaciones: retardos, pérdida, duplicación

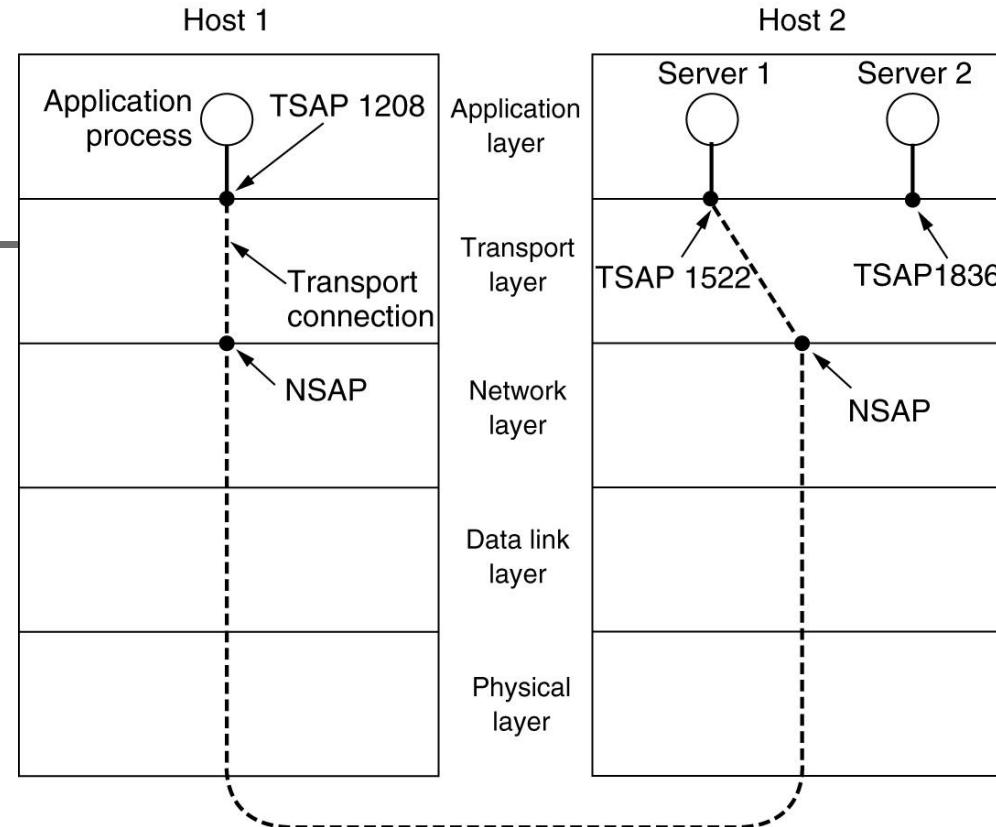


**Figura 6-7.** (a) Entorno de la capa de enlace de datos. (b) Entorno de la capa de transporte.



## 6.2.1 Direcccionamiento

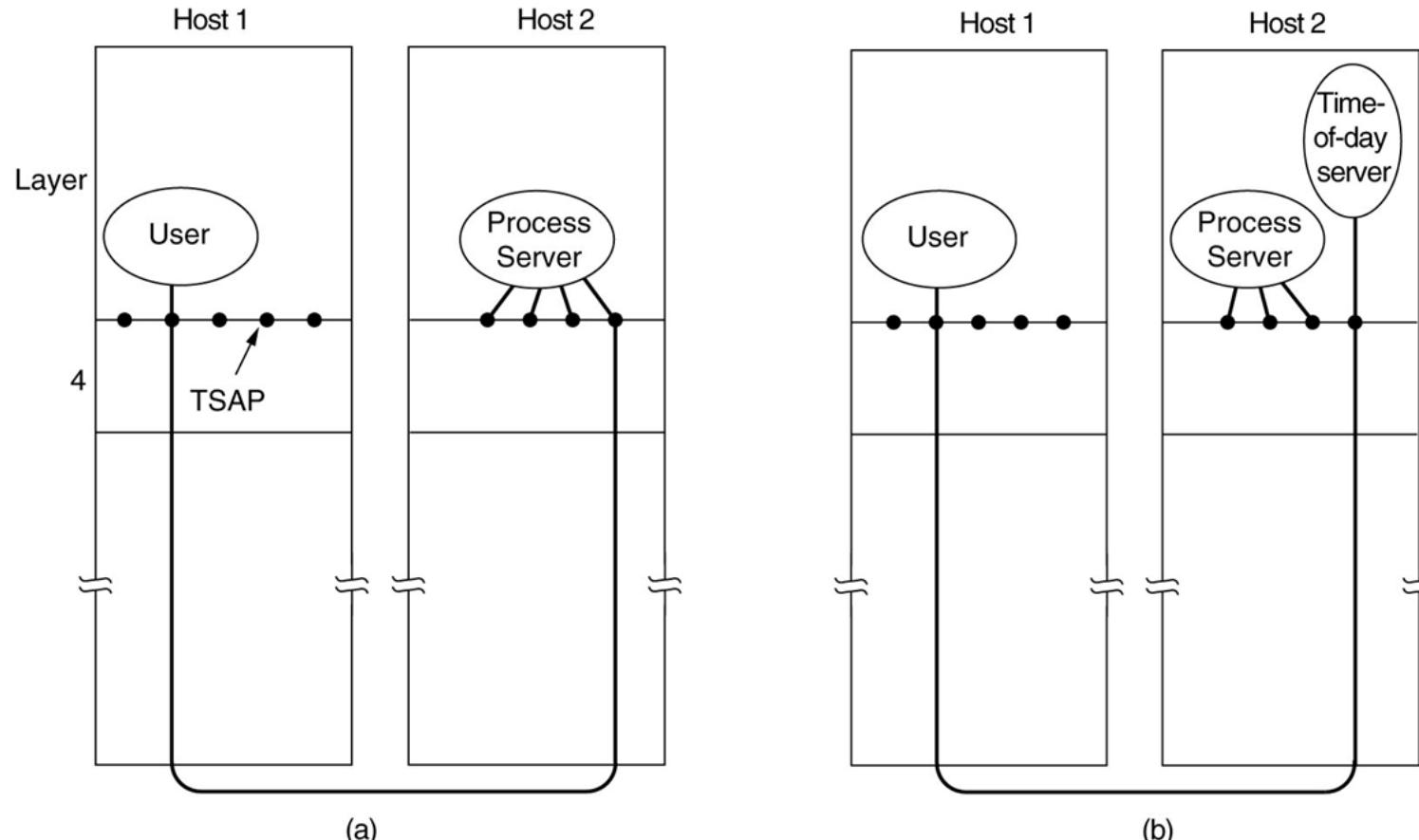
- A más de la dirección de red, los procesos, local y remoto que se comunican necesitan una dirección de transporte
- Los procesos pueden estar a la escucha de solicitudes de conexión
- En Internet estas direcciones se llaman puertos o TSAP

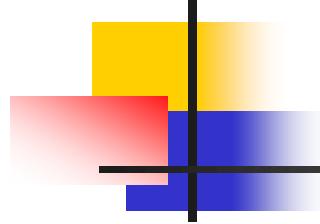


- El proceso en el host 1 sabe que el servidor de hora del día está conectado al TSAP 1522
- Hasta 1023 son servicios bien conocidos /etc/services

# Servidor de procesos *inetd*

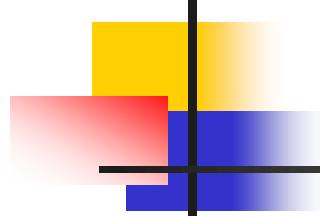
## Apoderado o proxy de servidores de menor uso





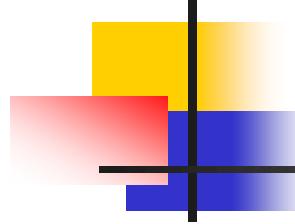
## 6.2.2 Establecimiento de una conexión

- La red puede perder o duplicar paquetes. Esto complica el manejo de las conexiones
- Un usuario establece una conexión con un banco para transferir dinero
- Si cada paquete de la transacción se duplica, se haría más de una transferencia de dinero



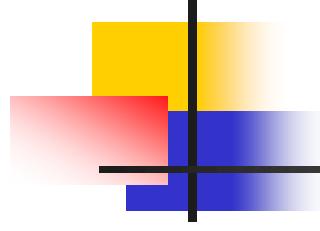
## Forma de solucionar duplicación de paquetes

- Uso de direcciones de transporte desecharables
  - Cada vez que se requiere una transacción se genera un nueva dirección de transporte
  - Al cerrar la conexión se desecha la dirección y no se vuelve a usar
  - Si aparece un paquete duplicado, con una dirección anterior, éste se descarta



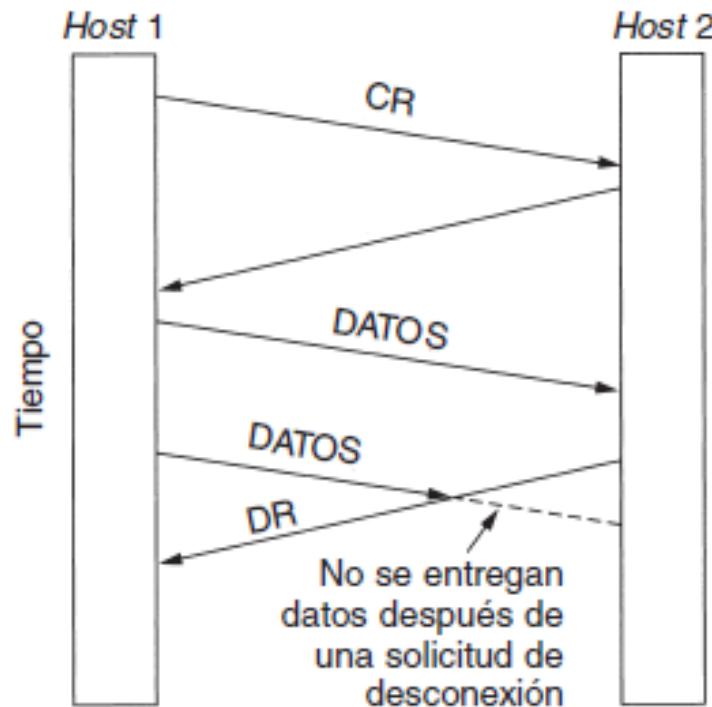
## 6.2.3 Liberación de una conexión

- Es más fácil liberar que establecer una conexión
- Hay 2 estilos de terminar una conexión:
  - Asimétrica
  - Simétrica



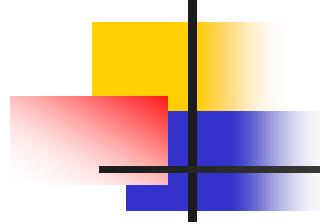
## Liberación asimétrica

- Análogo al sistema telefónico
- Si una parte cuelga se interrumpe la conexión
- Es abrupta y puede haber pérdida de datos



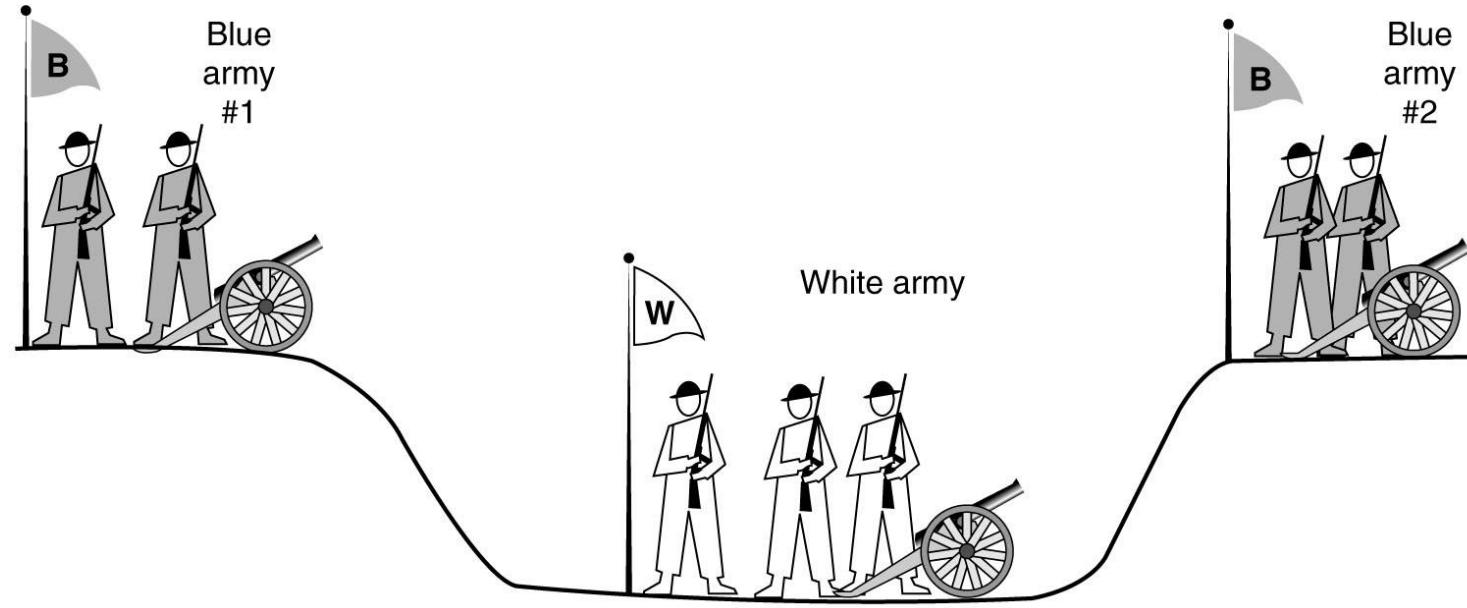
**Figura 6-12.** Desconexión abrupta con pérdida de datos.

CR connection request; DR disconnection request

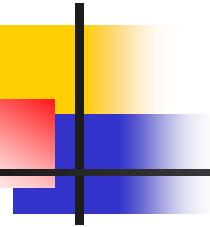


## Liberación simétrica

- Protocolo más refinado
- Evita pérdida de datos
- Consta de **2** conexiones unidireccionales
- **Cada dirección se libera por separado**
- Un host puede recibir datos luego de enviar un DR



Los ejércitos azules deben sincronizarse para atacar simultáneamente al ejército blanco para poder vencer



- Problema: el emisor del mensaje final nunca sabrá que su mensaje llegó
- Si ninguna de las partes sabe que la otra está lista para desconectares, nunca ocurrirá la desconexión
- El protocolo podría fallar
- Un acuerdo de tres vías **por lo general** funciona bien

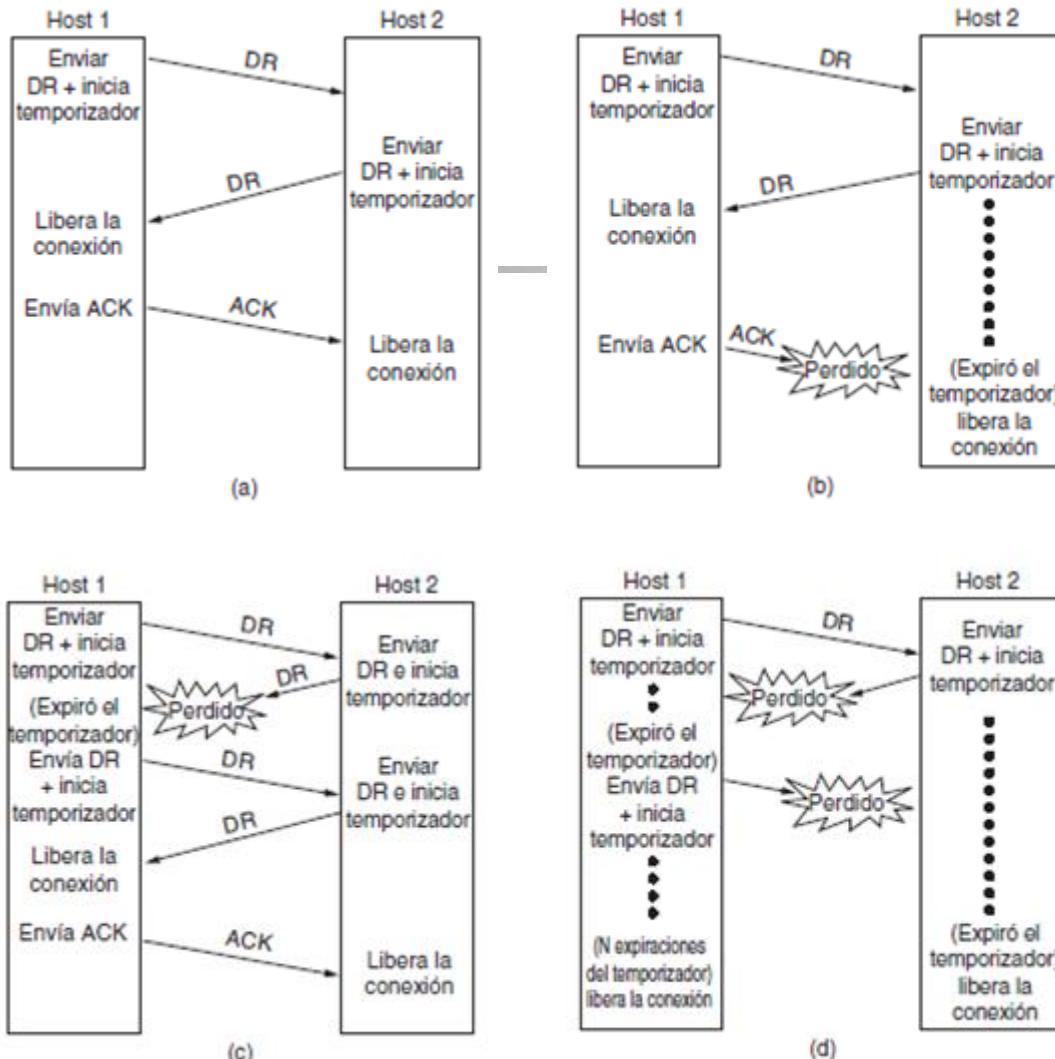
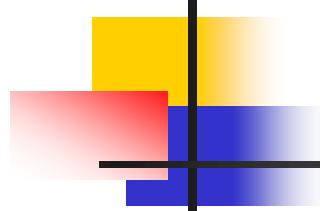


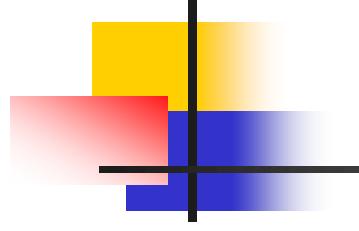
Figura 6-14. Cuatro escenarios de un protocolo para liberar una conexión. (a) Caso normal del acuerdo de tres vías. (b) Pérdida del último ACK. (c) Respuesta perdida. (d) Respuesta perdida y pérdida de los segmentos DR subsecuentes.

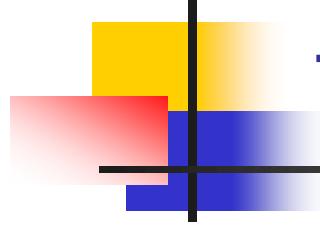


## 6.2.4 Control de flujo y almacenamiento en búfer en el manejo de las conexiones

- El control de flujo en la CT es similar al de la capa 2
- Las CTs tienen una mecanismo para no desbordar al receptor:
  - Parada espera: El emisor envía un paquete y espera un ACK
  - Ventana deslizante: El emisor envía varios paquetes y espera un ACK

- La diferencia en la CT está en que un router tiene **pocas líneas**, y un host puede tener **muchas conexiones**
- Un enrutador no tiene CT
- En la capa 2 existen buffers por línea. En la CT existen buffers por conexión
- El emisor tiene buffers de salida que almacenan las TPDU hasta recibir ACK del receptor
- El receptor tiene buffers de entrada para almacenar las TPDU mientras se procesa una TPDU

- 
- Se puede manejar:
    - Un grupo de buffers para todas las conexiones de transporte, o
    - Un grupo de buffers para cada conexión
  - La cuestión es el tamaño de los buffers



## Tamaño de los buffers

- Se asignan buffers por conexión
- Hay maneras según el caso:
  1. TPDUs del casi del mismo tamaño: Cadena de buffers de tamaño fijo
  2. TPDUs de tamaño variable: Cadena de buffers de tamaño variable
  3. Un solo buffer grande circular

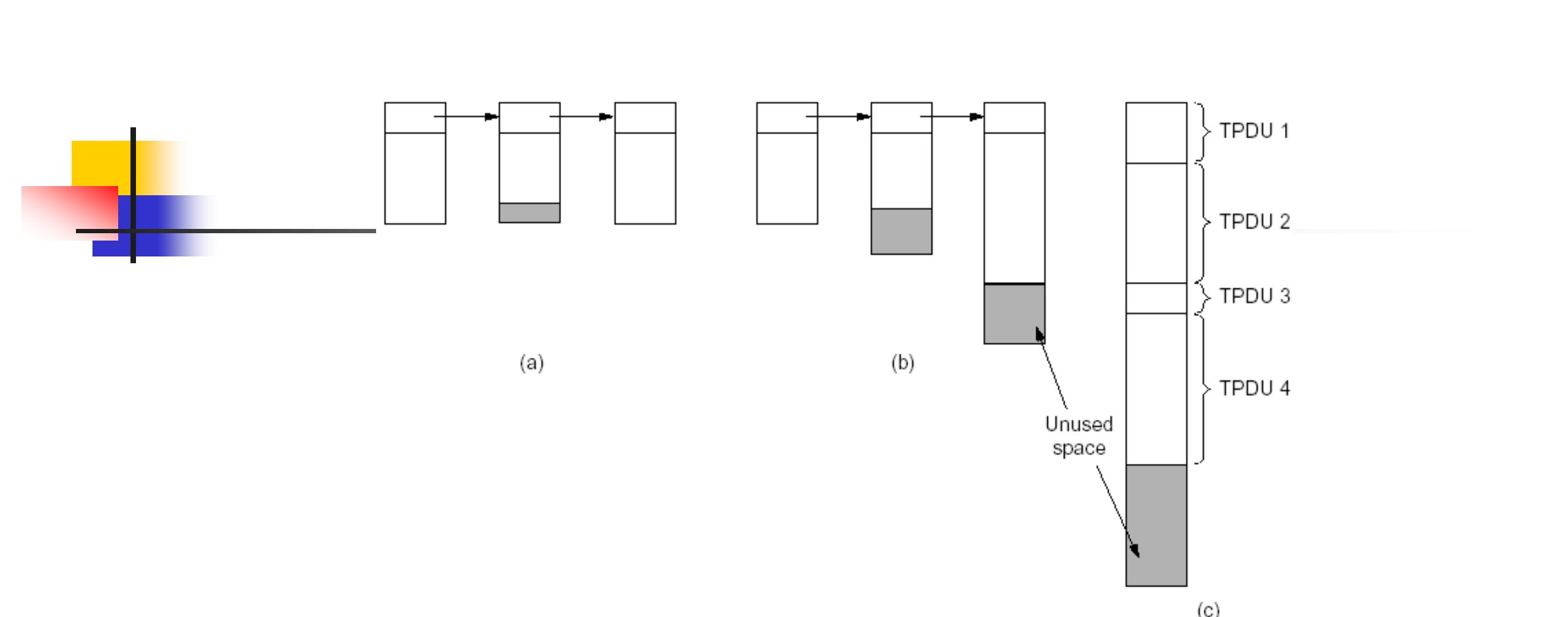
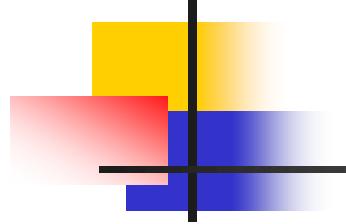
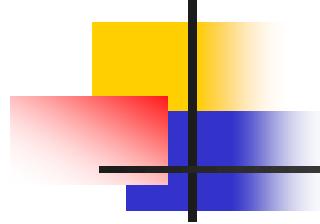


Fig. 6-15. (a) Chained fixed-size buffers. (b) Chained variable-sized buffers. (c) One large circular buffer per connection.

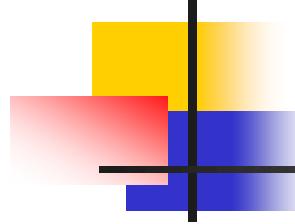
- La CT permite que el host emisor pida al receptor que se provea de suficientes buffers para que pueda recibir información
- Cuando se abren conexiones se asigna un número de tráfico
- El número de buffers en el origen y destino depende del tráfico que cambia dinámicamente
- Para tráfico con ráfagas **no se asigna un número fijo de buffers**, sino se lo hace según la magnitud de la ráfaga

- 
- La RAM es económica. Los hosts vienen con memoria suficiente. Los protocolos de CT no tienen este limitante
  - El cuello de botella está el número y capacidad de las líneas de la subred



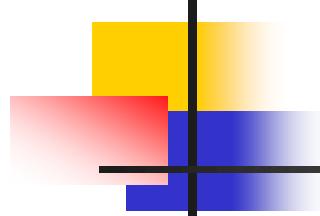
## 6.2.5 Multiplexación

- La multiplexión de varias conversaciones puede darse de tres maneras:
  1. Un enlace físico
  2. Un circuito virtual de capa 3 de OSI
  3. Una conexión de capa 4



## Multiplexión hacia arriba

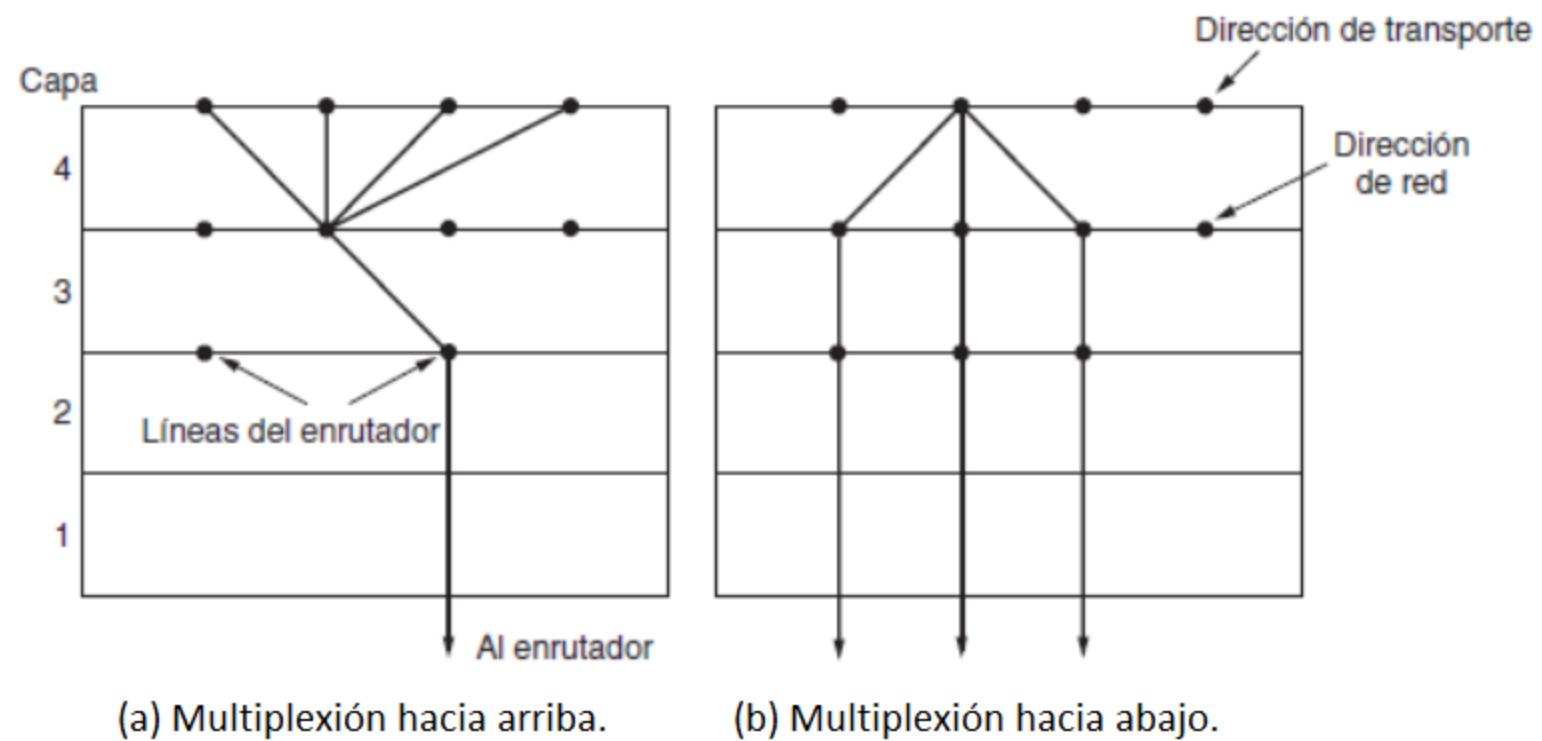
- En el Modelo TCP/IP, la capa de Internet no ofrece circuitos virtuales, solo datagramas (servicio sin conexión), por lo tanto no hay multiplexión
- En la CT surge la necesidad de multiplexar porque se tienen varios puertos y una sola dirección **de red** (IP en el modelo TCP/IP)
- Todas las conexiones de transporte tienen que compartir esta dirección

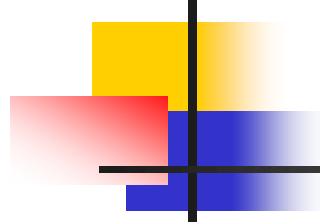


## Multiplexión hacia abajo

- En el Modelo OSI, La subred provee circuitos virtuales
- Si un usuario necesita más ancho de banda del dado en un CV, se puede abrir más CVs y distribuir el tráfico entre ellos
- Para ello, se necesitan varias direcciones de red
- En el Modelo TCP/IP los hosts solo tienen una dirección de red y no hay multiplexión hacia abajo

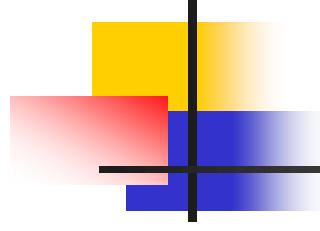
# Modelo OSI

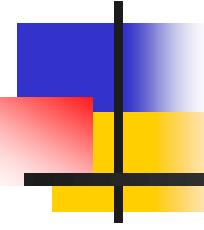




## 6.2.6 Recuperación de caídas

- Servidores, clientes y routers están sujetos a caídas
- Si la CR da servicio de datagramas, estos se pueden perder
- Si la CR da servicio orientado a conexión, se pueden perder los CVs
- Estos problemas lo maneja la CT
- Es deseable que los clientes sigan trabajando mientras un servidor que se recupera de una caída
- El problema es más complejo si se cae el host

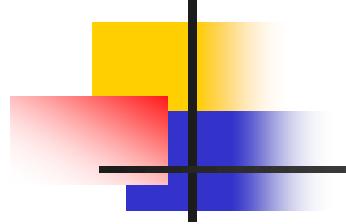
- 
- Para recuperar su estado previo, el servidor difunde una *TPDU* a todos los hosts con los que hubo conexiones para:
    - Anunciar que acaba de reiniciarse y
    - Solicitar información del estado de todas las conexiones abiertas

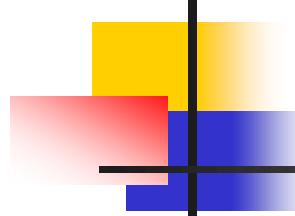


## 6.3 El protocolo de Internet UDP

---

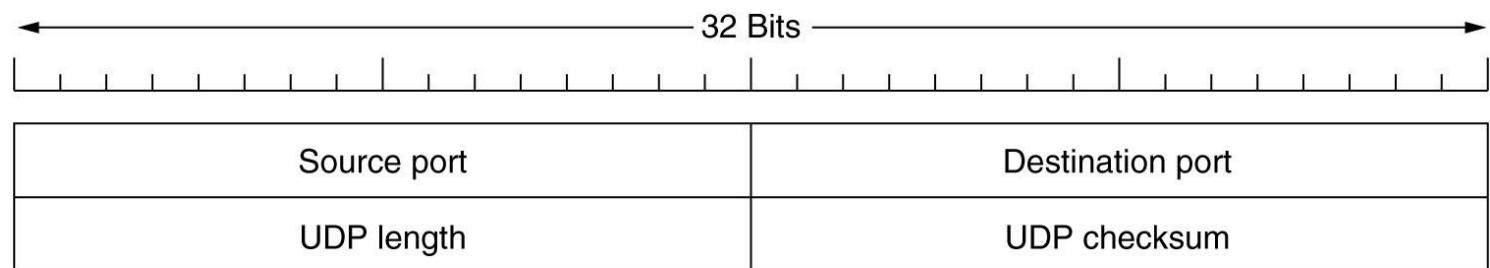
UDP User Datagram Protocol

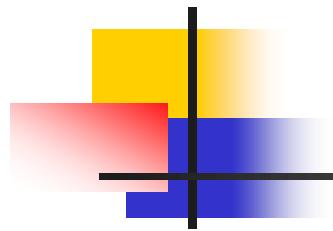
- 
- Internet tiene dos protocolos en la capa de transporte:
    - Uno orientado a la conexión TCP
    - Sin conexión UDP

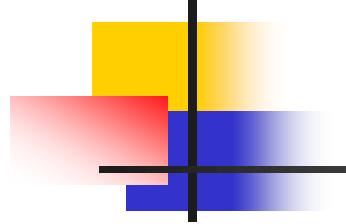


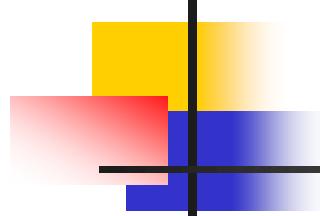
## 6.3.1 Introducción a UDP

- UDP permite que las aplicaciones funcionen sin conexión
- UDP divide los mensajes de las aplicaciones y los encapsula en segmentos
- Segmento: encabezado de 8 bytes + porción de mensaje (carga útil)
- ¿Por qué no usar solo datagramas IP?
- Se usa UDP en lugar de solo IP para indicar los puertos origen y destino
- Sin los puertos, la CT no sabría a qué aplicación enviar el segmento
- El puerto origen se necesita cuando se debe enviar una respuesta al emisor



- 
- UDP no realiza:
    - Control de flujo
    - Control de errores
    - Control de pérdidas
    - Retransmisión
  - Esto le corresponde hacer a los procesos de usuario, si lo necesita
  - UDP ofrece a las aplicaciones una interfaz al protocolo IP
  - Multiplexa varios procesos utilizando puertos

- 
- UDP es útil en ciertas aplicaciones cliente-servidor
  - El cliente envía una solicitud corta al servidor
  - El cliente espera una respuesta corta
  - Si se pierde la solicitud o respuesta, el cliente intenta de nuevo
  - Ejemplo: la aplicación DNS utiliza UDP: `Obt_direccion_IP(nombre_de_host)`



## 6.3.2 Llamada a procedimiento remoto RPC

- Es enviar un mensaje a un proceso que corre en otro host y obtener respuesta, como si se llamara a una *función* en un lenguaje de programación
- RPC hace que las aplicaciones de red sean más fáciles de programar y usar
- Así, los detalles de conexión se **ocultan** al programador de aplicaciones
- RPC *Remote Procedure Call* se ideó en 1984 y se usa en sistemas distribuidos

# Pasos para realizar un RPC

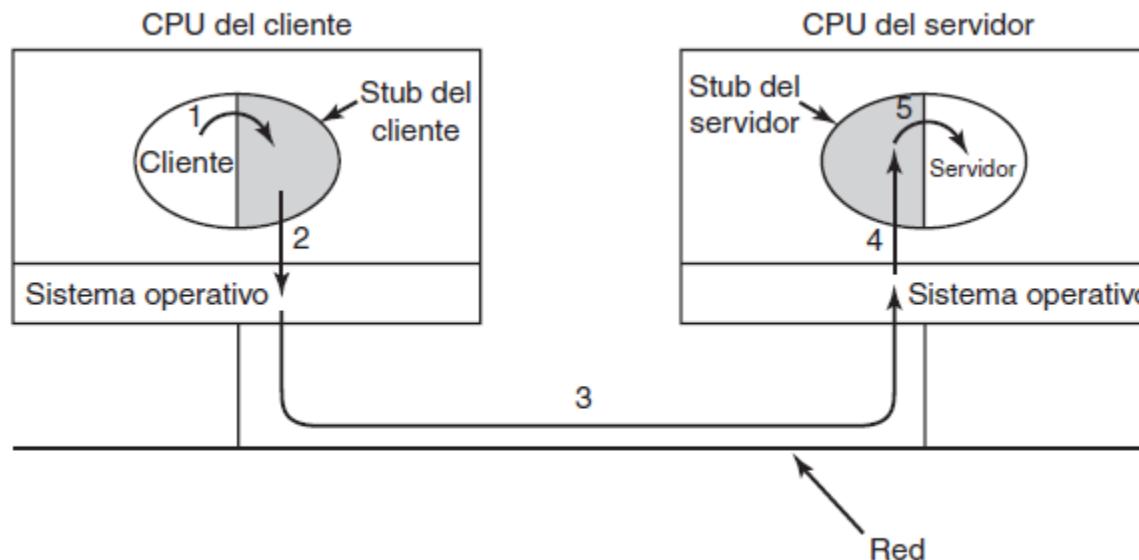
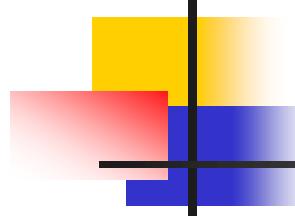
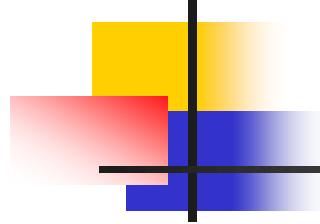


Figura 6-29. Pasos para realizar la llamada a un procedimiento remoto. Los stubs están sombreados.



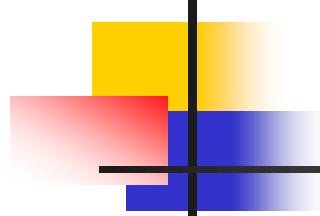
## Pasos para realizar un RPC

1. El cliente llama al *stub* del cliente
  2. El *stub del cliente* empaqueta los parámetros en un mensaje y llama al SO para enviarlo al servidor
  3. El SO envía el mensaje al servidor
  4. En el servidor, el SO pasa el paquete entrante al *stub del servidor*
  5. El *stub del servidor* llama al procedimiento servidor con los parámetros desempacados
- La respuesta sigue la misma ruta en forma opuesta
  - No se utilizan sockets



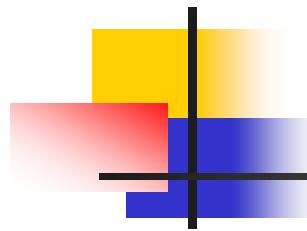
## Desventajas de RPC

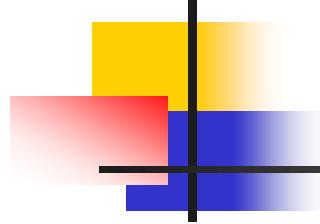
- Con RPC el paso de punteros es imposible porque los procesos residen en diferentes espacios de direcciones
- RPC se utiliza, con restricciones, muy ampliamente

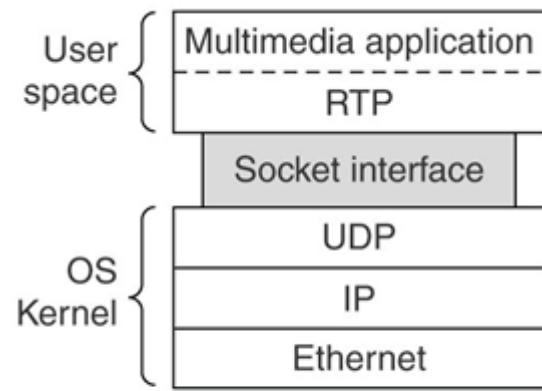


## 6.3.3 Protocolo de transporte en tiempo real RTP

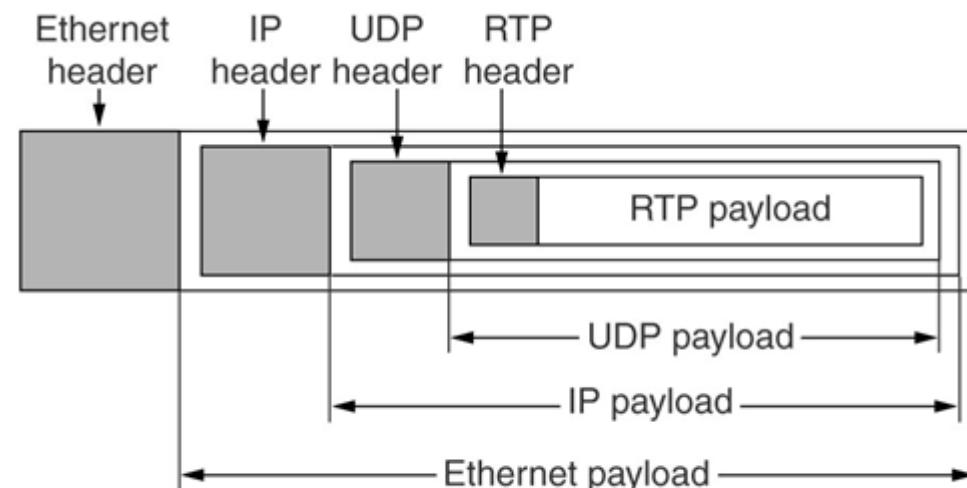
- *Tiempo real*: en vivo, momento actual, ahora
- *Información en tiempo real*: estar al tanto en forma inmediata del evento que ocurre ahora
- *Aplicación de tiempo real*: aplicación que informa de un evento con restricciones de retardo de transmisión y propagación
- *Tiempo de transmisión*: tiempo empleado por una estación para colocar todos los bits de una trama en el medio de transmisión
- *Tiempo de propagación*: Tiempo empleado por un bit en atravesar la red desde el origen al destino

- 
- UDP también se usa en aplicaciones en tiempo real
    - Sistemas de control de tráfico aéreo
    - Sistemas de monitoreo de signos vitales de un paciente
    - Aplicaciones para realizar operaciones financieras en la bolsa de valores
    - Sistemas de reserva de vuelos
    - Sistemas de control de procesos industriales:  $t^0$ , presión, rpm, . . .
    - Sistemas de control de generación distribución y consumo de energía eléctrica
    - Rastreo satelital para la industria del transporte
    - Videoconferencia

- 
- Aplicaciones en tiempo real requieren un protocolo de transporte de tiempo real
  - RTP Real-time Transport Protocol
  - RTP se ejecuta sobre UDP

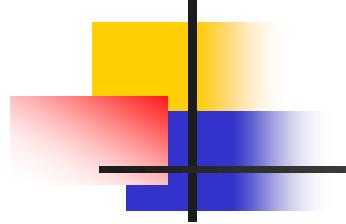


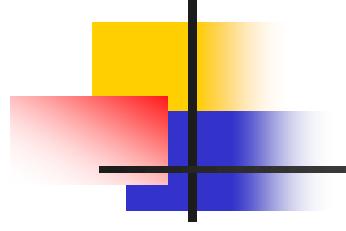
(a)

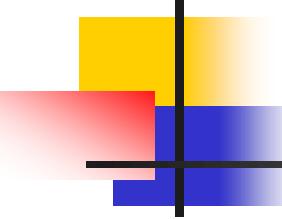


(b)

HDLC (High-Level Data Link Control) protocolo de capa de enlace punto a punto

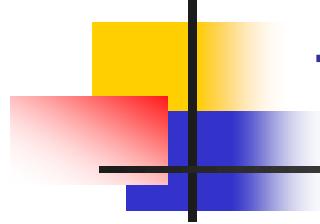
- 
- La aplicación de multimedia tiene **múltiples** flujos: audio, video y texto
  - RTP multiplexa los flujos de tiempo real, **función básica de RTP**
  - Luego se encapsula en paquetes RTP
  - RTP es un protocolo independiente de la aplicación
  - RTP se parece más a un protocolo de transporte
  - RTP es un protocolo de transporte implementado en la capa de aplicación

- 
- Cada paquete enviado en el flujo RTP tiene un número secuencial
  - Si falta uno, el receptor lo aproxima por interpolación
  - La retransmisión no es práctica, pues llegaría muy tarde para ser útil
  - No hay garantía de entrega



## RTP no hace:

- Control de flujo
- Control de errores
- Confirmación de recepción
- Retransmisiones



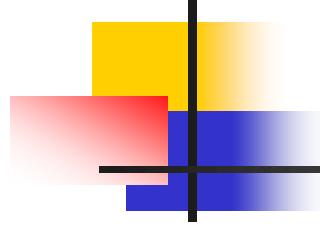
## Timestamping

- Marcación del tiempo
- Es una característica de las aplicaciones en tiempo real
- El destino almacena una pequeña cantidad de paquetes y lo reproduce unos milisegundos después del inicio del flujo
- Reduce el efecto de la fluctuación
- Permite que la sincronización de múltiples flujos: video, voz, música
- Ejemplo: TV digital estéreo y doblada en varios idiomas



## 6.4 El protocolo de Internet TCP

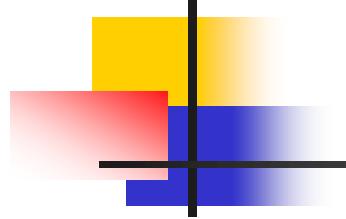
TCP Transport Control Protocol



## 6.4.1 Introducción a TCP

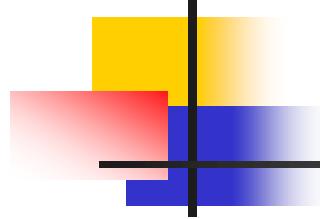
- TCP (1981) da un flujo de bytes confiable de extremo a extremo a través de la subred no confiable
- La entidad TCP se implementa:
  - en el kernel del SO, que es lo más común
  - en un procedimiento de biblioteca
  - en la aplicación de usuario

- El tamaño máximo de un fragmento TCP, al igual que un segmento UDP es de 64 Kbytes =  $64 \times 1024 - 1 = 2^{16} - 1 = 65.535$  bytes
- Los datos de usuario son 65535 bytes—el tamaño de la cabecera TCP o UDP
- IP soporta una carga de 65.535 bytes
- La aplicación de usuario funciona sobre un servicio orientado a la conexión de TCP que, a su vez recibe un servicio sin conexión de IP
- En la práctica, un segmento TCP lleva 1460 bytes de datos para que calce en una trama Ethernet, que puede llevar datagramas de hasta 1.500 bytes



## ■ Funciones de TCP:

1. Enviar los segmentos sin causar congestionamiento en la red
  2. Realizar retransmisiones de ser necesario
  3. Ordenar los fragmentos que llegan al destino en desorden
- Así, TCP proporciona confiabilidad en la comunicación, que no lo proporciona IP



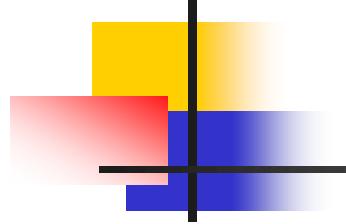
## 6.4.2 El modelo del servicio TCP

---

- Emisor y receptor crean cada uno un punto terminal llamado socket
- Socket = dirección IP del host + puerto de 16 bits
- Un puerto es un TSAP
- Hay que establecer explícitamente una conexión entre sockets origen-destino

Primitiva	Significado
SOCKET	Crea un nuevo punto terminal de comunicación.
BIND	Asocia una dirección local con un socket.
LISTEN	Anuncia la disposición de aceptar conexiones; indica el tamaño de la cola.
ACCEPT	Establece en forma pasiva una conexión entrante.
CONNECT	Intenta establecer activamente una conexión.
SEND	Envía datos a través de la conexión.
RECEIVE	Recibe datos de la conexión.
CLOSE	Libera la conexión.

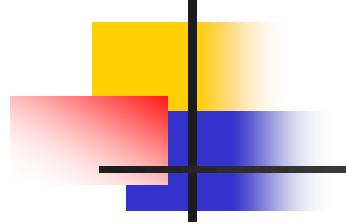
**Figura 6-5.** Las primitivas de socket para TCP.

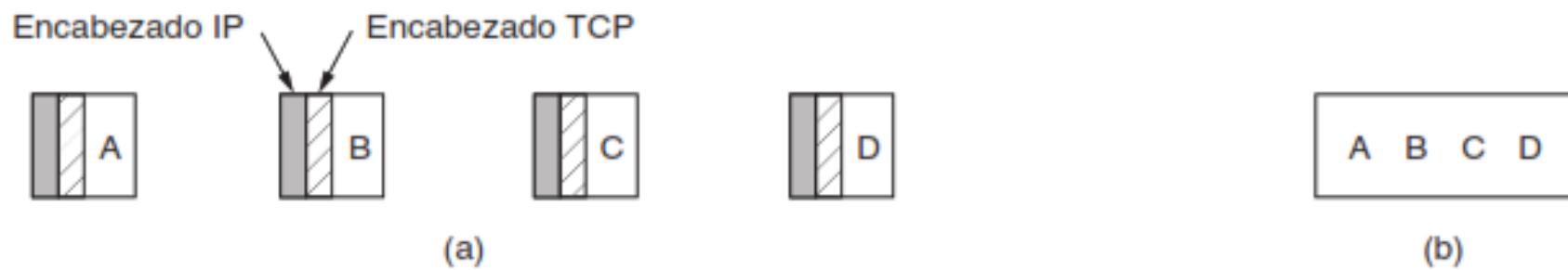
- 
- Un socket puede usarse para varias conexiones (multiplexión)
  - Dos o más conexiones pueden terminar en el mismo socket
  - Esto se hace en servidores que atienden a múltiples clientes a la vez
  - Las conexiones se identifican con los identificadores de los dos sockets
  - Los número de puerto menores a 1024 se llaman ***puertos bien conocidos*** y se reservan para servicios estándar
  - La lista de puertos bien conocidos se muestra a continuación

Puerto	Protocolo	Uso
20, 21	FTP	Transferencia de archivos.
22	SSH	Inicio de sesión remoto, reemplazo de Telnet.
25	SMTP	Correo electrónico.
80	HTTP	World Wide Web.
110	POP-3	Acceso remoto al correo electrónico.
143	IMAP	Acceso remoto al correo electrónico.
443	HTTPS	Acceso seguro a web ( <a href="#">HTTP sobre SSL/TLS</a> ).
543	RTSP	Control del reproductor de medios.
631	IPP	Compartición de impresoras.

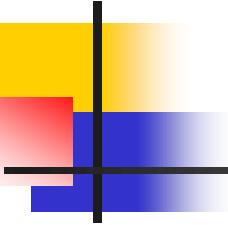
**Figura 6-34.** Algunos puertos asignados.

- El demonio de cada servicio se conecta a su respectivo puerto; por ejemplo, el servidor ftp se conecta al puerto 21
- Hacerlo así podría llenar la memoria con demonios inactivos
- Un super demonio de Internet (*inetd* en Unix) se conecta a múltiples puertos
- Se puede tener demonios permanentes en los puertos más ocupados (ej. 80) e *inetd* en los demás
- Conexiones TCP son full duplex
- TCP no soporta la multidifusión ni difusión, UDP sí

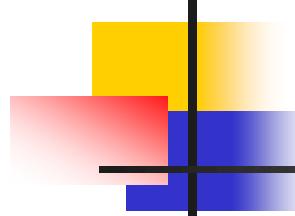
- 
- Una conexión TCP es un flujo de bytes, no un flujo de mensajes
  - TCP no reconoce mensajes de usuario
  - El flujo de bytes es agrupado en diferentes segmentos TCP
  - Si el usuario emite 4 mensajes de 512 bytes en un flujo TCP, estos mensajes se envían en cuatro, dos o un segmento de 512 bytes, 1024 bytes, o 2048 bytes
  - La CT en el receptor no distingue si llegaron uno, dos o cuatro mensajes. TCP solo detectó que llegó un flujo de bytes a través de diferentes segmentos
  - La CA se encarga de tomar el flujo de bytes y construir los mensajes



**Figura 6-35.** (a) Cuatro segmentos de 512 bytes que se envían como diagramas IP separados. (b) Los 2 048 bytes de datos que se entregan a la aplicación en una sola llamada READ.

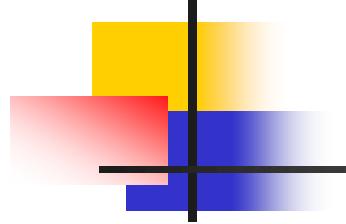


- TCP envía los datos inmediatamente, o los almacena en un buffer para recolectar de la aplicación más datos y luego enviarlos
- Si la aplicación necesita que los datos se envíen inmediatamente activa el bit de la cabecera del segmento PUSH



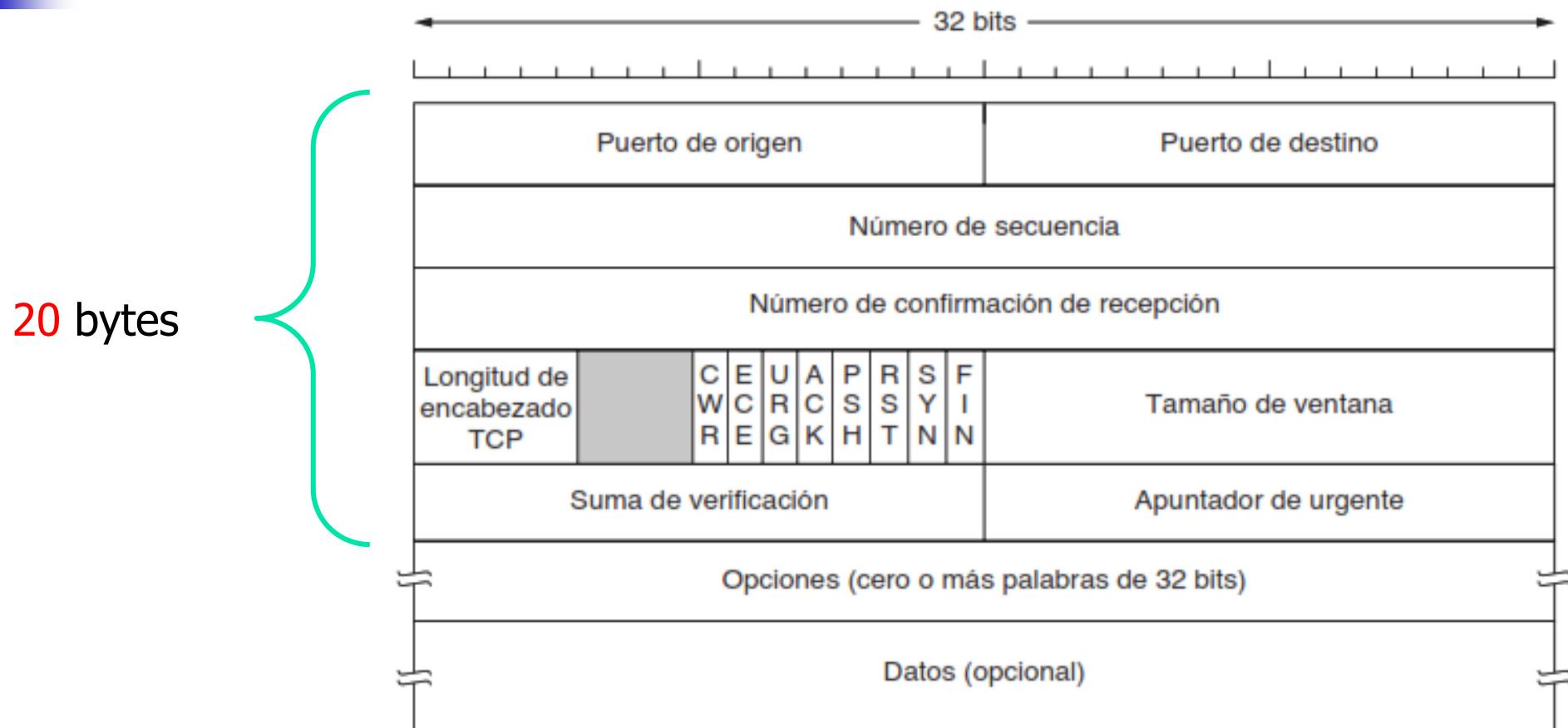
## 6.4.3 El protocolo TCP

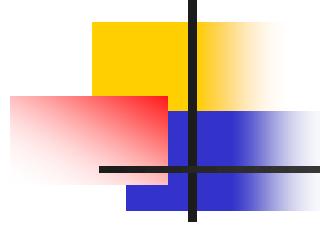
- A cada byte de una **conexión** le corresponde un número de secuencia de 32 bits
- Antes, los enlaces entre enrutadores eran líneas telefónicas de  $56Kbps$
- ¿En cuánto tiempo se agota la numeración?
- El número máximo es  $2^{32} - 1 = 4.294'967.295 \approx 4.2\text{ GB}$
- $56Kbps = 7.000\text{ bytes/s}$
- Al número máximo  $2^{32} - 1$  se llegaba en  $\frac{(2^{32}-1)\text{bytes}}{7.000\text{ bytes/s}} \approx 1\text{ semana}$
- Los números de secuencia se usan en el mecanismo de ventana deslizante

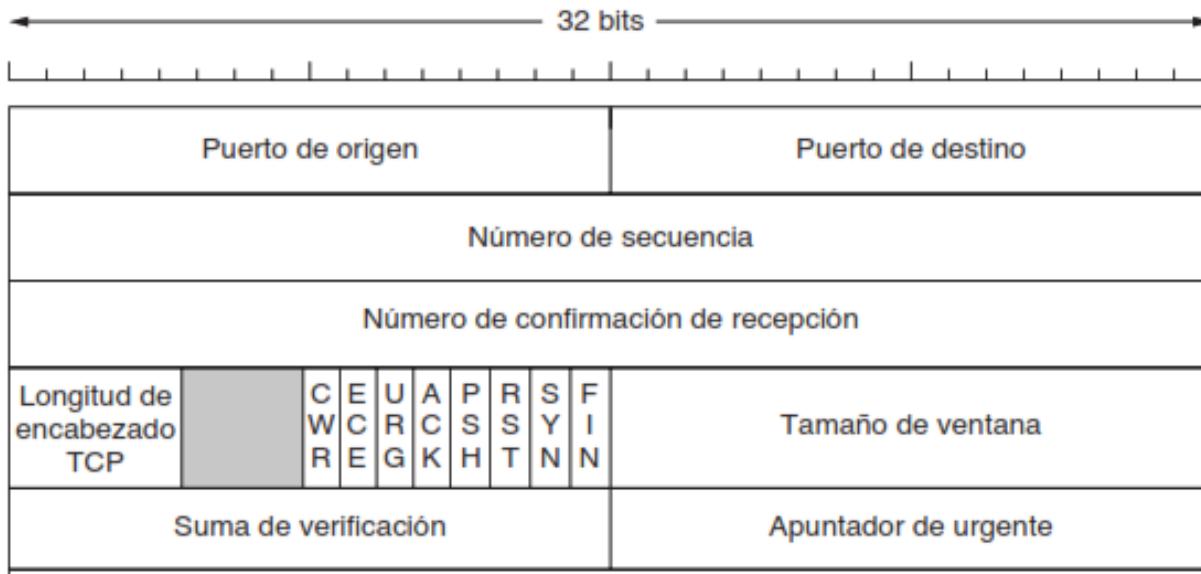
- 
- Un segmento TCP tiene una cabecera de 20 bytes (+ una parte opcional) seguido de 0 o más bytes de datos
  - TCP decide el tamaño de los segmentos en función de dos restricciones:
    1. Que el segmento quepa en la carga útil de IP
    2. Que quepa en la MTU de ruta (Path Maximum Transfer Unit) para evitar fragmentación. En la práctica  $MTU=1500$  bytes (carga útil de Ethernet)

- TCP usa el protocolo de ventana deslizante:
  - Cuando el emisor transmite un segmento inicia un temporizador de retransmisión
  - Cuando el segmento llega al destino, TCP envía un ACK con o sin datos, indicando el siguiente número de secuencia a recibir (byte)
  - Si el temporizador expira, el emisor retransmite el segmento

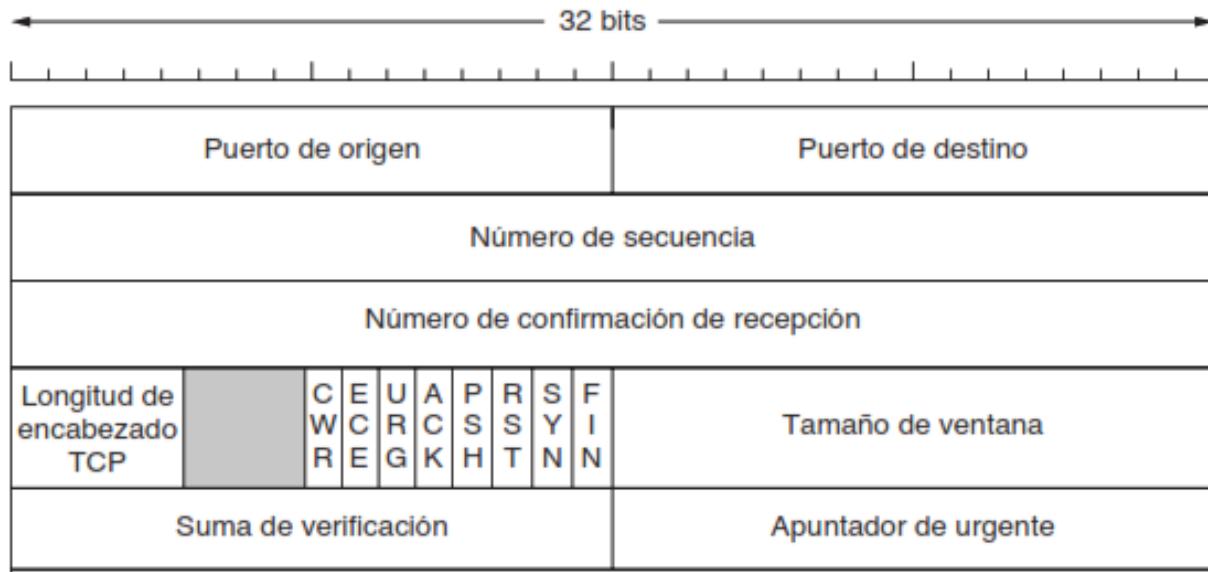
## 6.4.4 Encabezado del segmento TCP



- 
- Cada segmento comienza con un encabezado de formato fijo de **20** bytes
  - El encabezado puede ir seguido de opciones
  - Tras las opciones, si las hay, están los de datos de usuario
  - Los segmentos sin datos se usan para ACK y mensajes de control



- *Puerto origen y puerto destino.* Son los puntos terminales locales de conexión
- Dirección IP + puerto forman un punto terminal único de 48 bits -socket local
- Los puntos terminales identifican la conexión
- El identificador de conexión se llama 5-tupla. Tiene 5 piezas de información: protocolo IP origen, puerto origen, IP destino, protocolo TCP, puerto destino
- *Número de secuencia.* Indica el primer byte del segmento

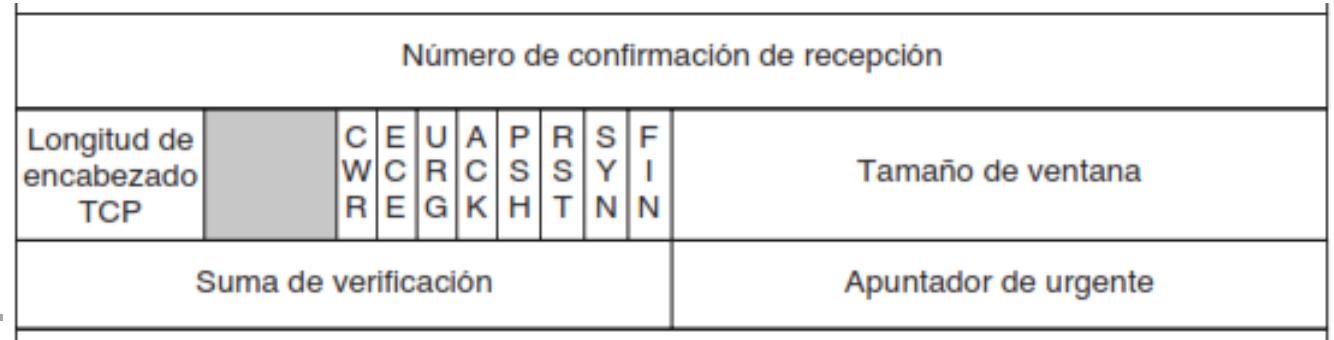


- *Número de confirmación de recepción* (ACK). Indica el número del primer byte del siguiente segmento esperado, no es el número del siguiente segmento
- *Longitud del encabezado*. Con 4 bits expresa en palabras de 32 bits. Máximo  $15 \times \frac{32}{8} = 60 \text{ bytes}$
- Luego hay 4 bits que no se utilizan

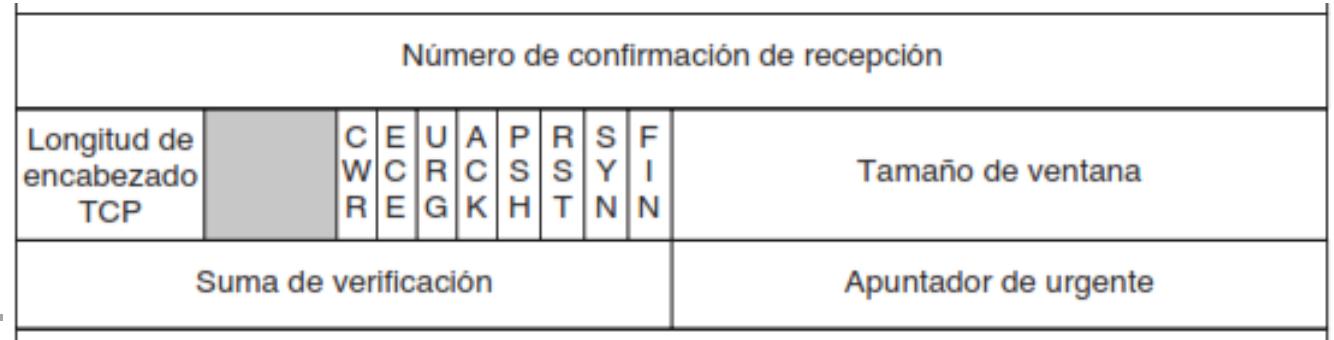
# Ocho Indicadores de 1 bit

Número de confirmación de recepción										
Longitud de encabezado TCP		C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Tamaño de ventana
Suma de verificación						Apuntador de urgente				

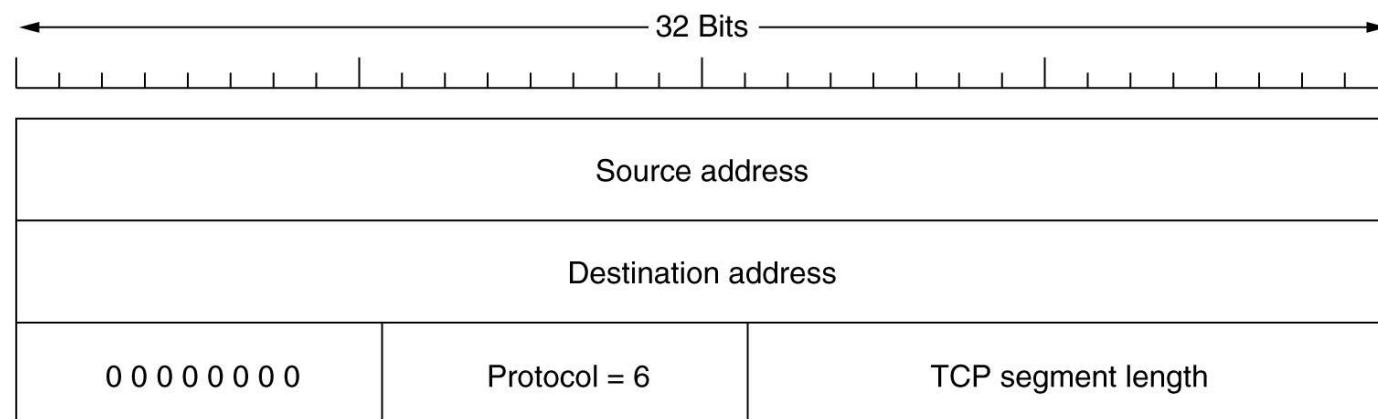
- **CWR** Congestion Window Reduced. La red avisa al receptor que hay congestión
- **ECE**. Explicit Congestion notification-Echo. Al mismo tiempo la red pide al emisor que baje la velocidad
- **URG**. Si el bit está en 1 indica que está activo el *Apuntador de urgente*
- **ACK**. Indica que el *Número de confirmación de recepción* es válido
- **PSH**. Push. La aplicación solicita a TCP que envíe el mensaje inmediatamente al destino y en el receptor que entregue inmediatamente a la aplicación



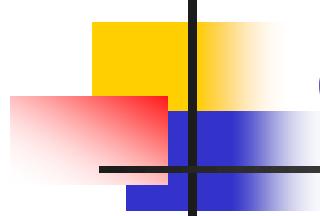
- **RST** Reset. Restablece la conexión debido a la caída del host
- **SYN**. Se usa para establecer una conexión
  - CONNECTION REQUEST: SYN = 1, ACK = 0
  - CONNECTION ACCEPTED: SYN = 1, ACK = 1
- **FIN** Finish. Se usa para liberar una conexión. Un emisor no tiene más datos que enviar pero puede seguir recibiendo datos
- **Tamaño de ventana**. (Capítulo 3. Canalización) El control de flujo se maneja usando una ventana de tamaño variable. A menor tamaño menor flujo
- El tamaño máximo de ventana es  $2^{16} = 64\ KB$  antes de recibir el primer ACK
- Si es 0, el receptor necesita momentáneamente un descanso



- **Suma de verificación.** Considera el encabezado TCP, los datos y un pseudo encabezado que incluyen las direcciones IP
- La inclusión de direcciones IP viola la jerarquía de protocolos ya que estas corresponden a la capa de Internet
- UDP también incluye el pseudo encabezado en el checksum
- *Apuntador de urgente.* Trabaja con el bit *URG*. Indica el desplazamiento en bytes en el que están los datos urgentes a partir del *Número de secuencia* del byte actual



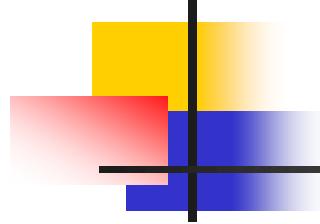
# Pseudo encabezado

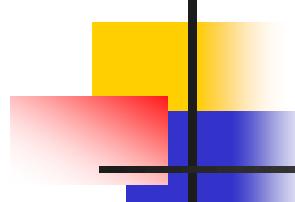


## Campo de opciones

- El host especifica la carga útil TCP máxima que está dispuesto a aceptar
- Esta opción se indica al momento de establecer la conexión
- Segmentos grandes son más eficientes porque hay menos encabezados para más datos
- La carga útil predeterminada es 536 bytes
- Se requiere que todos los hosts de Internet acepten segmentos TCP de 536 + 20 bytes de cabecera

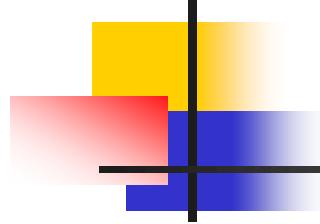
- En líneas con gran ancho de banda y/o alto retardo de propagación, la ventana de 16 bits es muy poco
- Si se tiene una línea de FO transoceánica de 600 Mbps, el tiempo de transmisión de una ventana de 16 bits ( $2^{16}$  bytes = 64KB) es
- $$\frac{64 \text{ KB}}{600 \text{ Mbps}/8} = 8 \times 10^{-4} \text{ s} < 1ms$$
- Si la distancia de una lugar a otro es unos 15.000 Km el retardo de propagación es de unos 50 ms
- El emisor estará ocioso  $\left(1 - \frac{1ms}{50ms}\right) \times 100 = 98\%$  del tiempo
- Un tamaño de ventana mayor a 16 bits permitirá al emisor enviar más datos
- La opción *escala de ventana* permite al emisor y receptor negociar al momento del establecimiento de la conexión, el aumento de tamaño de la ventana hasta 14 bits más, para transmisiones de hasta  $2^{30}$  bytes

- 
- *Confirmación de recepción selectiva*. Permite al receptor indicar al emisor los rangos de números de secuencia que ha recibido
  - Se utiliza después de haber perdido un paquete
  - Así, el emisor sabe qué segmentos se deben retransmitir en forma selectiva



## 6.5.5 Establecimiento de una conexión TCP

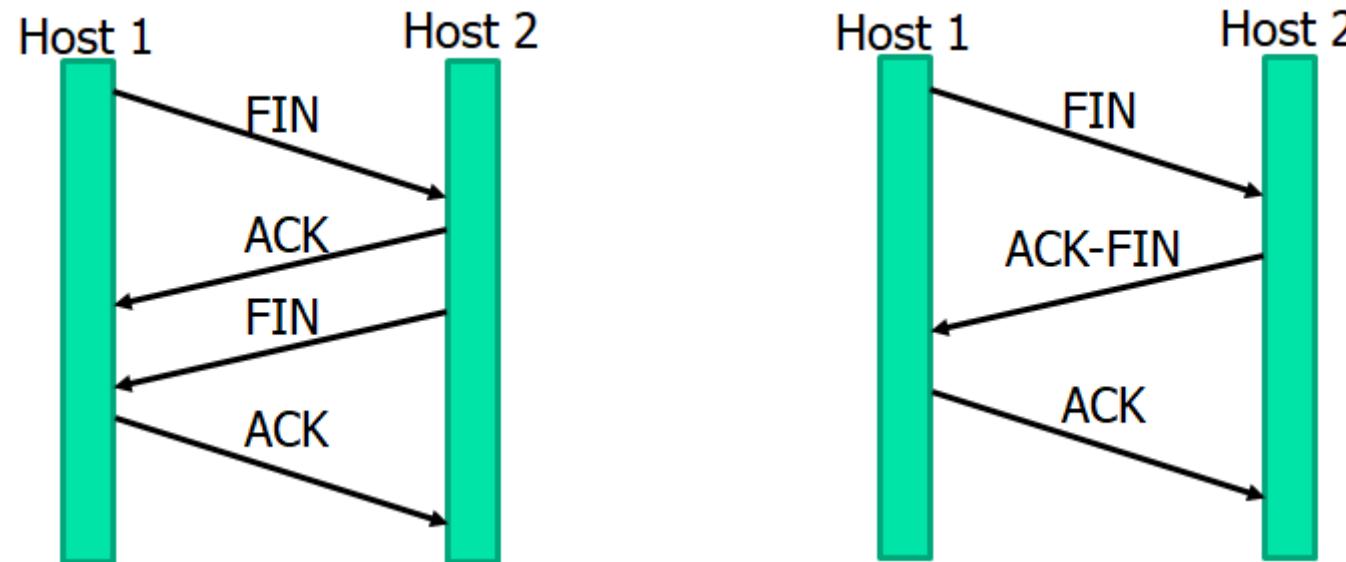
- Se usa un acuerdo de 3 vías
- El cliente solicita una conexión ( $SYN=1$ ,  $ACK=0$ ) indicando dirección IP y puerto destinos, y el tamaño máximo del segmento TCP a aceptar
- El cliente espera una respuesta del servidor
- El servidor espera una conexión entrante indicando el origen
- El servidor revisa si está esperando una conexión en el puerto destino especificado
- Si hay, acepta la conexión ( $SYN=1$ ,  $ACK=1$ ). Si no lo hay, el servidor envía una respuesta al cliente rechazando la conexión ( $RST=1$ )

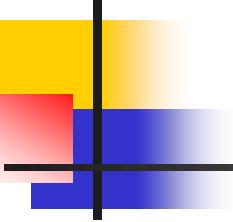


## 6.5.6 Liberación de una conexión TCP

- Las conexiones TCP son full dúplex, formadas con dos conexiones simplex
- Cada conexión simplex se libera independientemente
- Para liberar una conexión, una de las partes puede enviar un segmento TCP indicando que no tiene más datos por transmitir
- Al recibirse este segmento, ese sentido de la conexión se apaga
- Pero puede continuar un flujo indefinido de datos en el otro sentido
- Cuando ambos sentidos se apagan se libera la conexión

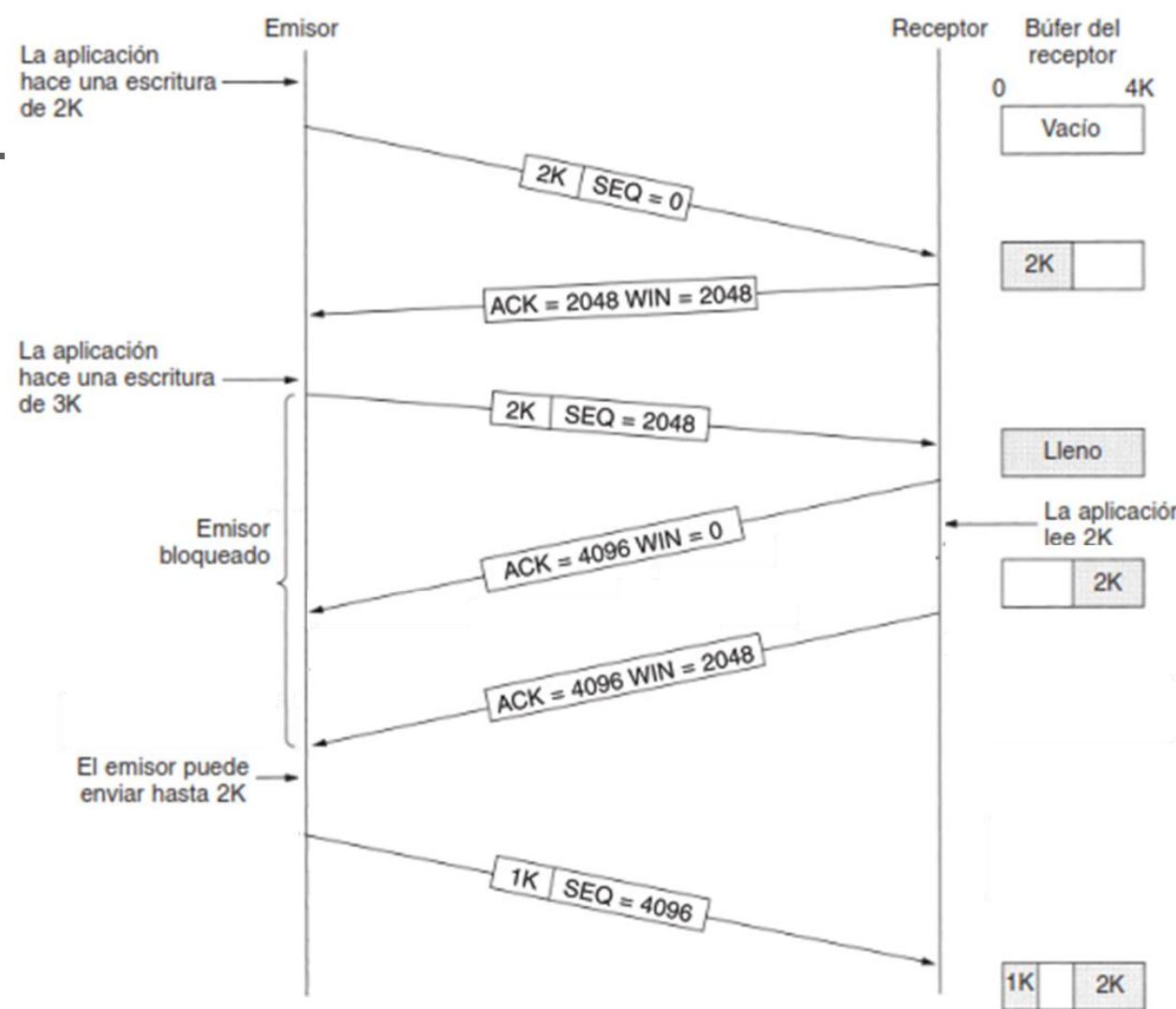
- Normalmente se necesitan 4 segmentos TCP para liberar una conexión: un FIN y un ACK para cada sentido
- Pero, es posible que el primer ACK y el segundo FIN estén contenidos en el mismo segmento, reduciéndose la cuenta total a 3.





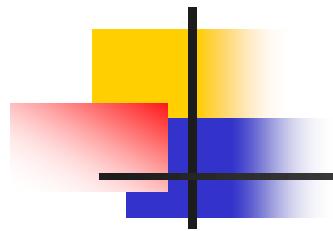
- Para evitar el problema de los 2 ejércitos, se usan temporizadores
- Si no llega una respuesta a un FIN en un máximo de dos tiempo de la vida de un paquete, el emisor de FIN libera la conexión
- Tarde o temprano el otro lado notará que nadie le está escuchando, también liberará la conexión
- Ésta no es una solución perfecta pero pocas veces ocurren problemas

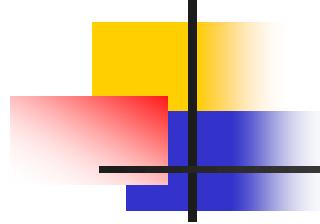
## 6.5.8 Política de transmisión de TCP



- Suponga que el receptor tiene un buffer de 4096 bytes (4K)
- La aplicación en el emisor hace una escritura de 2048 bytes (2K)
- El emisor envía un segmento de 2K, desde el byte 0 al byte 2047
- El receptor los recibe correctamente
- El receptor envía la confirmación de recepción del segmento
- La aplicación en el receptor aun no retira datos del buffer
- El receptor tiene solo 2K libres
- El receptor anuncia una ventana de 2K y el siguiente byte esperado
- La aplicación en el emisor hace una escritura de 3072 bytes (3K)

- El emisor envía un segmento de 2K desde el byte 2048 al byte 4095
- El buffer en el receptor se llena
- El receptor avisa recibo de los datos y que el siguiente segmento a recibir es desde el byte 4096, pero al momento el buffer está lleno
- Luego la aplicación en el receptor lee del buffer 2K de datos
- El receptor avisa al emisor que el siguiente segmento a recibir es desde el byte 4096, y el espacio libre en buffer es de 2K
- El emisor envía el 1K restante iniciando desde el byte 4096

- 
- No se requiere que el emisor envíe datos tan pronto como llegue de la aplicación
  - Al inicio de la transmisión, el emisor podía esperar otros 2KB para transmitir un segmento con una carga útil de 4 KB
  - Tampoco que se requiere que los receptores envíen confirmaciones de recepción tan pronto como sea posible
  - El receptor podía esperar procesar todos los segmentos del buffer, y luego anunciar que tiene una ventana de 4 KB, y no sólo de 2KB

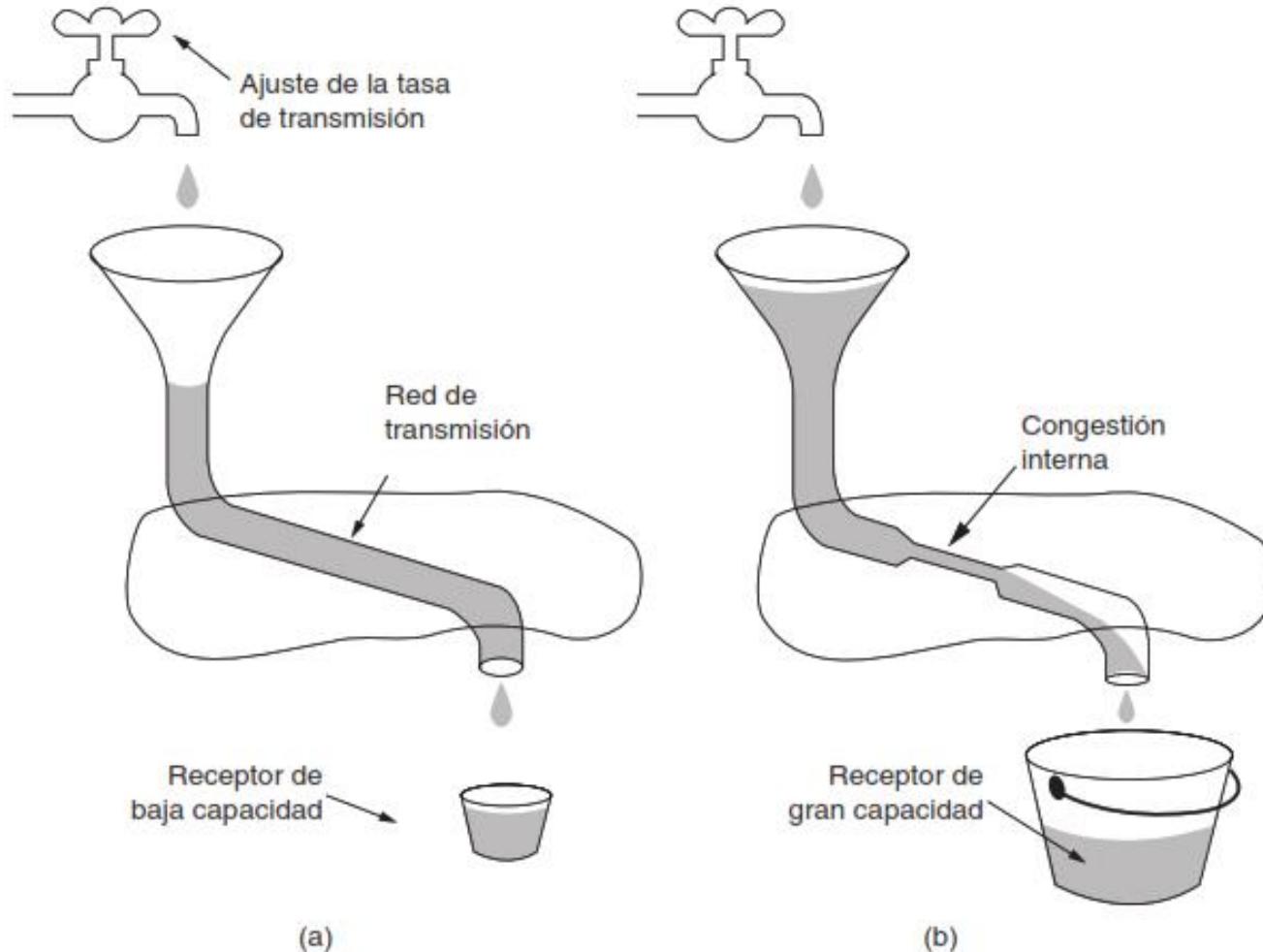


## 6.5.9 Control de congestión en TCP

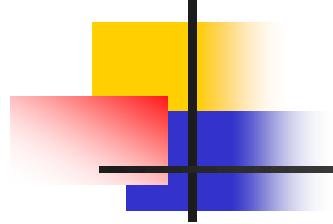
- Cuando la carga injectada a la red es mayor que la que puede manejar, se genera una congestión
- La capa IP intenta controlarla con los protocolos de enrutamiento
- Pero el trabajo pesado recae sobre TCP
- La solución a la congestión es disminuir la tasa de datos: No injectar un paquete nuevo a la red hasta que salga uno viejo
- TCP lo soluciona manipulando dinámicamente los tamaños de las ventanas

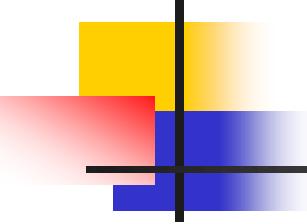
- El primer paso es detectar la congestión
- La congestión se puede detectar observando las expiraciones de temporizador
- Expiración de temporizador implica la retransmisión de paquetes
- La retransmisión implica aumento de carga de la red
- La expiración de un temporizador por un paquete perdido, puede deberse a:
  - Ruido en la línea
  - Desbordamiento en un enrutador **congestionado**
- No hay pérdida de paquetes por errores de transmisión en líneas de fibra óptica. En las redes inalámbricas es diferente
- La mayoría de esas expiraciones en Internet se deben a congestión

- 
- TCP toma acciones preventivas y acciones correctivas frente a la congestión
  - Evitación o prevención de congestiones
    - Al establecer una conexión, emisor y receptor acuerdan un tamaño de ventana tal que no haya derrame en el receptor
  - Pero aun puede ocurrir congestión interna en la red
    - El temporizador caduca y el emisor baja la velocidad



**Figura 6-22.** (a) Una red veloz que alimenta a un receptor de baja capacidad. (b) Una red lenta que alimenta a un receptor de alta capacidad.

- 
- En Internet hay dos problemas potenciales
    - Capacidad de la red y
    - Capacidad del receptor
  - La estrategia es manejarlos por separado
  - Para ello, cada emisor mantiene dos ventanas:
    - La ventana de recepción que ha otorgado el receptor, y
    - La ventana de congestión
  - Cada una indica la cantidad de bytes que puede enviar el emisor
  - La cantidad que puede enviar el emisor es la **menor** de las dos ventanas

- 
- Hay dos algoritmos que permiten determinar la cantidad de bytes que puede enviar el emisor sin producir congestión ni en el receptor ni en la red
    - Arranque lento
    - Algoritmo de Internet

# Algoritmo de arranque lento



Tamaño de la ventana de congestión del emisor



Segmento confirmado

Tamaño de la ventana luego de la



1ra. transmisión de la ventana

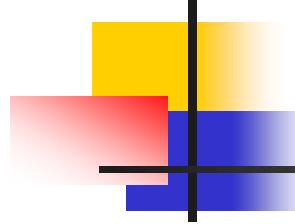


2da. transmisión de la ventana



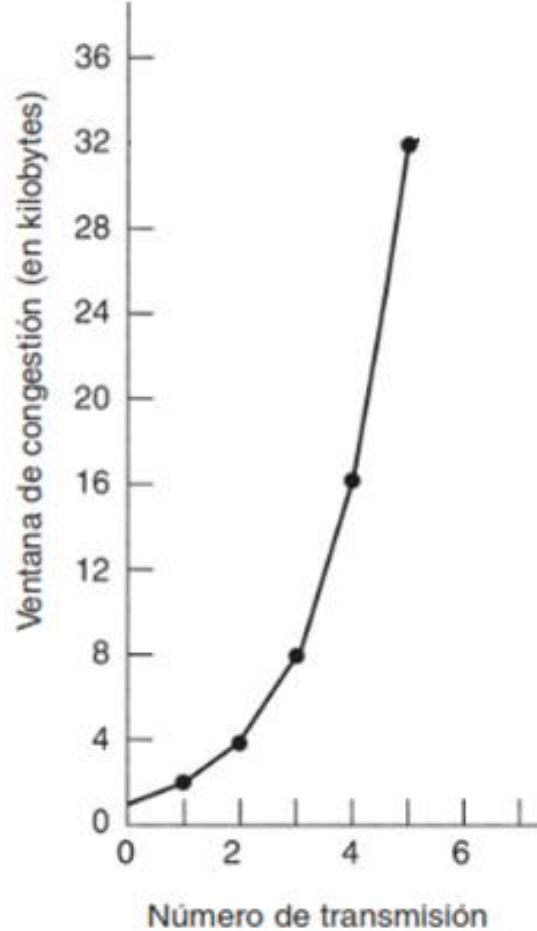
3ra. transmisión de la ventana



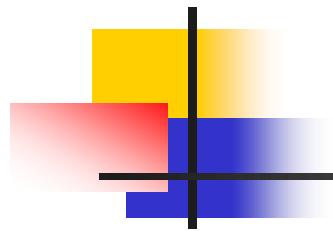


## Algoritmo de arranque lento

- Al establecer una conexión, el emisor asigna a la ventana de congestión el tamaño de **segmento máximo**, entonces envía un segmento máximo
- Si se recibe la confirmación de recepción de este segmento antes de que expire el temporizador, el emisor aumenta el tamaño de la ventana en un segmento máximo y envía dos segmentos de tamaño máximo
- A medida que se confirma cada uno de estos segmentos, se aumenta el tamaño de la ventana de congestión en un segmento máximo
- Cada **ráfaga** confirmada duplica el tamaño de la ventana de congestionamiento



### Ejemplo del algoritmo de Arranque Lento

- 
- La ventana de congestión sigue creciendo exponencialmente hasta:
    - Ocurrir una expiración del temporizador, o
    - Alcanzar el tamaño de la ventana receptora

# Algoritmo de Internet

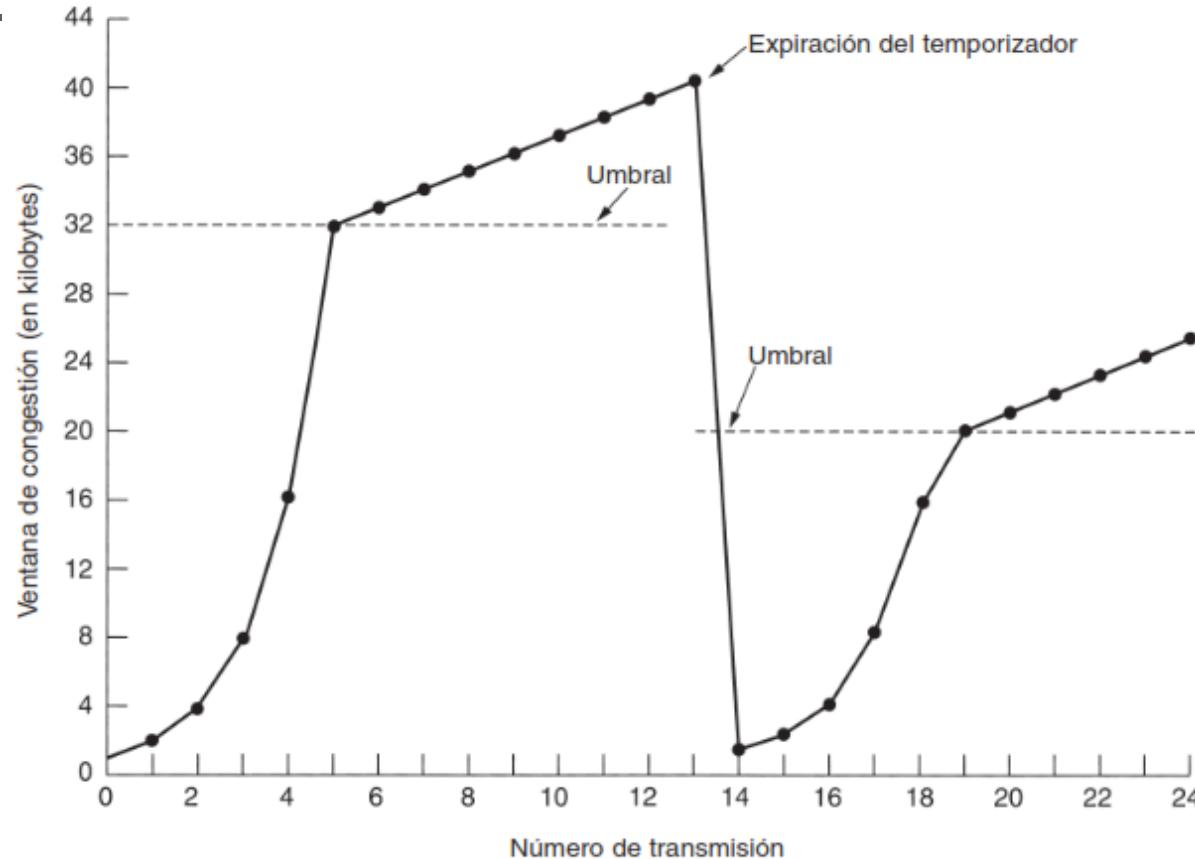
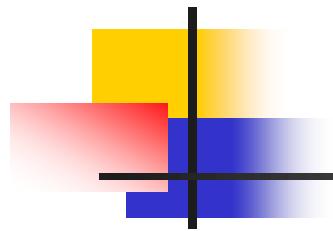


Figura 6-37. Ejemplo del algoritmo de congestión de Internet.

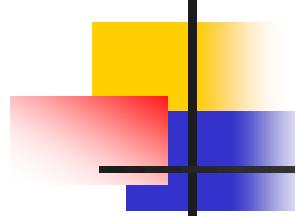


# Algoritmo de Internet

 Tamaño de la ventana de congestión del emisor

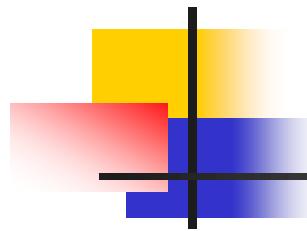
 Segmento confirmado

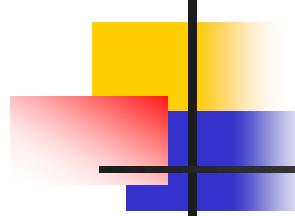




# Algoritmo de Internet

- Se usa el parámetro **umbral**, inicialmente de 64 KB, además de la ventana del receptor y de la ventana de congestión
- Al expirar el temporizador, se establece el umbral en la mitad de la ventana de congestión actual, y la ventana de congestión se restablece a un segmento máximo
- Luego se usa el *algoritmo de arranque lento*
- Pero, el crecimiento exponencial termina al alcanzar el umbral
- Luego, las transmisiones exitosas aumentan linealmente la ventana de congestión; o sea, aumenta en un segmento por ráfaga, no en un segmento por segmento

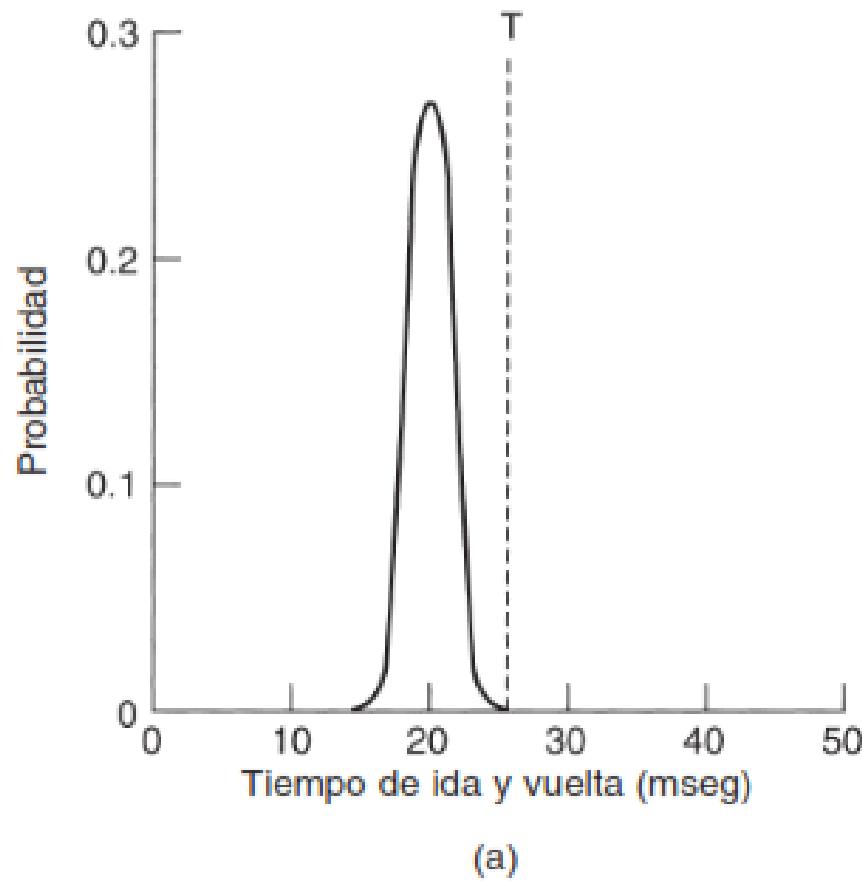
- 
- Si no ocurren más expiraciones del temporizador, la ventana de congestión continuará creciendo hasta el tamaño de la ventana del receptor
  - En este punto, dejará de crecer y permanecerá constante mientras no ocurran más expiraciones del temporizador



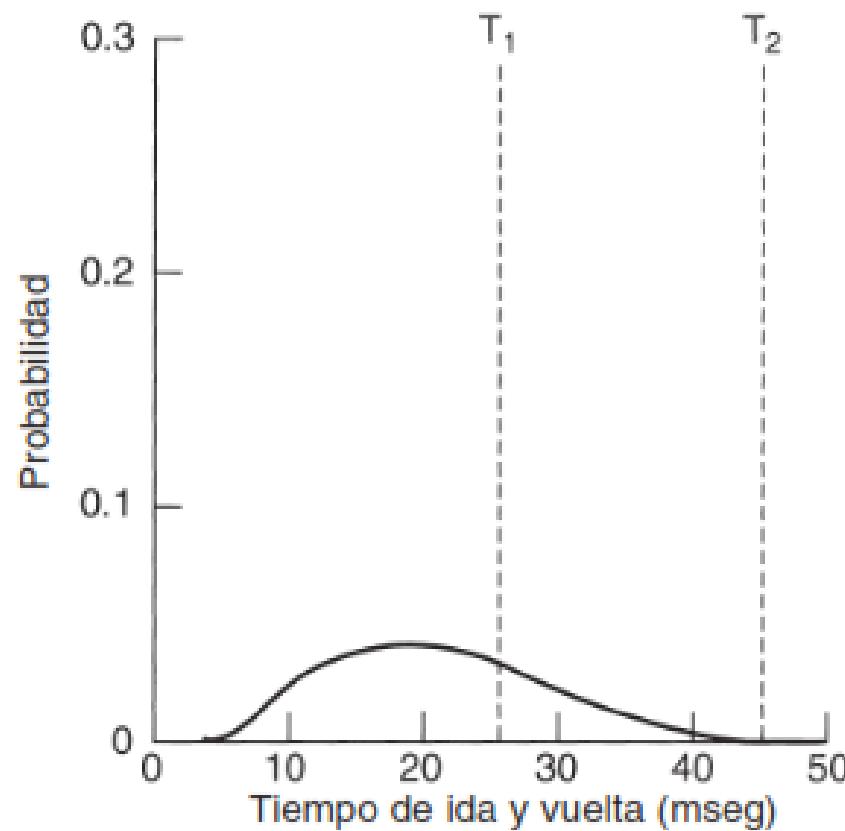
## 6.4.10 Administración de temporizadores del TCP

- Al enviarse un segmento se inicia el temporizador de retransmisiones
- Si la confirmación de recepción del segmento llega antes de expirar el temporizador, este se detiene
- Si el temporizador termina antes de llegar la confirmación de recepción, se retransmite el segmento y se reinicia el temporizador
- ¿Qué tanto debe durar el temporizador?
- Esta duración es mucho más predecible en el capa 2 que en la capa 4. Fig. 6.38(a)

## Probabilidad del tiempo de llegada de la confirmación en la capa 2 vs el temporizador T

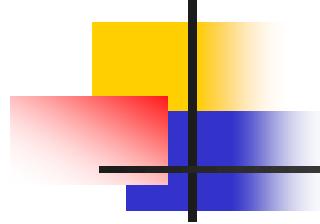


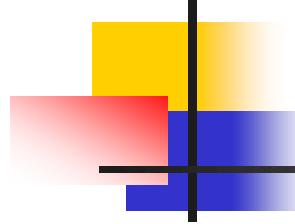
## Probabilidad del tiempo de llegada de la confirmación en la capa 4 vs el temporizador T1 o T2



(b)

- En la capa 2 las confirmaciones de recepción pocas veces se retrasan
- La ausencia de confirmación de recepción generalmente se debe a la pérdida de esta o de la trama
- En TCP, es difícil la determinación del tiempo de ida y vuelta
- Si el intervalo de expiración es corto ( $T_1$ ) ocurren retransmisiones innecesarias, sobrecargando la red innecesariamente
- Si es largo ( $T_2$ ), el desempeño disminuye debido al mayor retardo de retransmisión de cada paquete perdido
- Por otro lado el tiempo de ida y vuelta puede variar rápidamente

- 
- La solución es usar un algoritmo que ajuste dinámicamente el intervalo de expiración del temporizador
  - Esto se logra con base en mediciones continuas del desempeño de la red
  - El algoritmo que usa TCP es de Jacobson (1988)



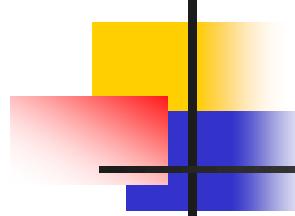
## Algoritmo de Jacobson

---

- TCP utiliza cuatro temporizadores diferentes:
- De retransmisión
- De persistencia
- De seguir con vida
- De espera cronometrada

## Temporizador de retransmisión

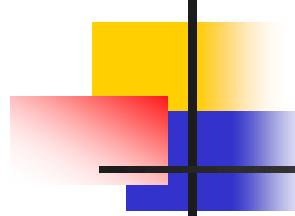
- Para cada conexión, TCP mantiene una variable  $RTT$  (Round-Trip Time), que es la mejor estimación actual del tiempo de ida y vuelta al origen
- Al enviar un segmento se inicia un temporizador para ver el tiempo de confirmación de recepción y para habilitar la retransmisión si se tarda demasiado
- Si llega la confirmación de recepción en un tiempo  $M$ , antes de expirar el temporizador, TCP actualiza  $RTT$ :
  - $RTT = \alpha RTT + (1 - \alpha)M$
  - $\alpha$  es el peso que se da al  $RTT$  anterior
  - Por lo general  $\alpha = 7/8$ . Así, se da más peso a  $RTT$  anterior que a  $M$
  - Algunas implementaciones de TCP usa el temporizador de retransmisión = $2RTT$



## Temporizador de persistencia

- Este es diseñado para evitar el bloqueo
- El receptor envía una confirmación de recepción, indicando al emisor que espere (tamaño de ventana 0). Esta indicación llega al emisor
- Después el receptor actualiza la ventana, pero se pierde el paquete con la actualización
- Ahora el emisor y receptor esperan que el otro haga algo
- Cuando termina el temporizador de persistencia el emisor envía un sondeo al receptor

- 
- La respuesta al sondeo da el tamaño de la ventana
  - Si la ventana sigue en 0, se inicia otra vez el temporizador de persistencia
  - Si la ventana es diferente de cero el emisor puede enviar datos

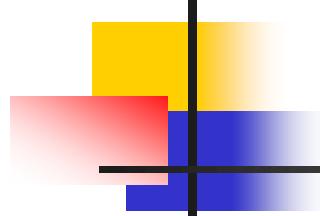


## Temporizador de seguir con vida

- Si una conexión está inactiva por mucho tiempo, este temporizador expira
- El un lado comprueba que el otro lado aun está allí
- Si no recibe respuesta, se termina la conexión

## Temporizador de espera cronometrada

- Este opera durante el cierre de una conexión
- Tiene una duración del doble del tiempo máximo de vida del paquete
- Así se asegura que todos los paquetes desaparezcan antes del cierre

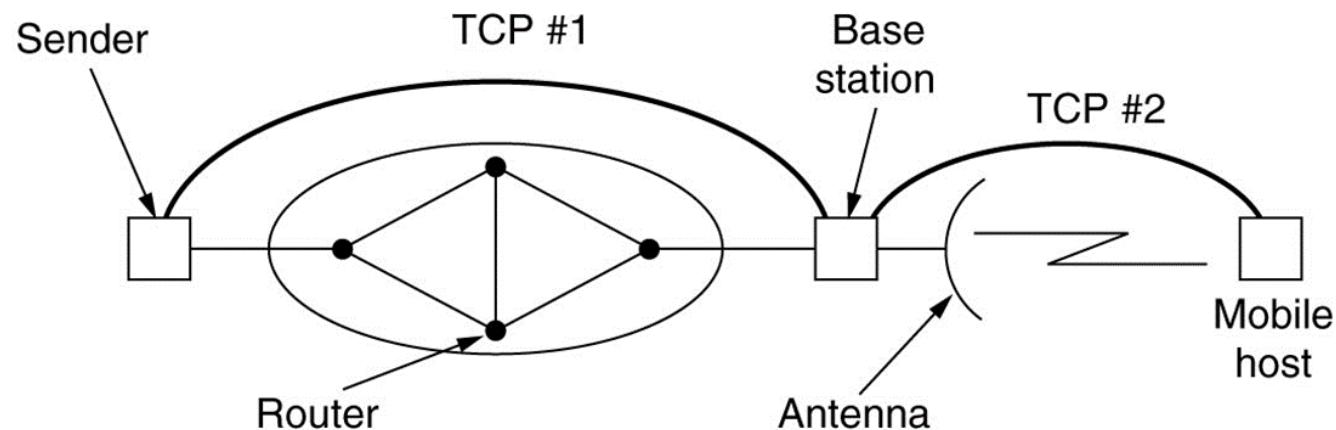


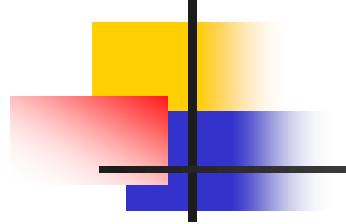
## 6.4.11 TCP y UDP inalámbricos

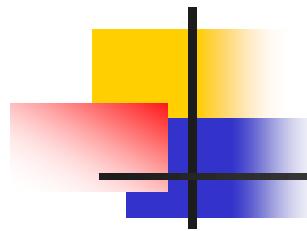
- Los protocolos de transporte deberían ser independientes de la tecnología de la capa de red subyacente
- TCP no debería preocuparse si IP opera sobre cobre, fibra o radio
- En la práctica sí importa, ya que las implementaciones TCP son optimizadas para redes cableadas, no para redes inalámbricas
- El problema principal es el algoritmo de control de congestión
- TCP supone que las expiraciones del temporizador ocurren por congestionamiento, no por paquetes **perdidos**
- Por tanto, TCP disminuye su velocidad (arranque lento) para reducir la carga a la red y aliviar la congestión

- Pero los enlaces inalámbricos pierden muchos paquetes, no por congestionamiento
- El enfoque adecuado para el manejo de paquetes perdidos es reenviarlos inmediatamente
- Al perderse un paquete por congestionamiento en una red cableada, el emisor debe reducir la velocidad
- En redes inalámbricas la reducción de la velocidad empeora las cosas, el emisor debe al menos mantener la velocidad, no reducirla
- Si el emisor no sabe si la red es cableada o inalámbrica, no se puede tomar la decisión correcta

- Con frecuencia, el camino desde el emisor al receptor no es homogéneo
- **TCP indirecto** es la solución propuesta
- La solución es dividir la conexión TCP en conexiones distintas



- 
- La ventaja de este esquema es que ahora ambas conexiones son homogéneas
  - Las expiraciones del temporizador en la red 1 reducen la velocidad del emisor
  - Las expiraciones del temporizador en la red 2 aceleran la velocidad
  - La desventaja: viola la semántica de TCP: conexión extremo a extremo

- 
- La siguiente solución no quebranta la semántica de TCP
  - Funciona haciendo modificaciones pequeñas en la estación base
  - Un cambio es incluir un agente que almacena en caché los segmentos TCP que van al host móvil, y las confirmaciones de recepción que regresan de él
  - Si el agente no ve la llegada de una confirmación de recepción antes de que el temporizador expire, retransmite este segmento sin indicar el origen de retransmisión