

# Teóricos - Redes y Sistemas Distribuidos 2021

## ÍNDICE:

- ❖ [Introducción](#)
- ❖ [Capa de Aplicación](#)
- ❖ [Capa de Transporte](#)
- ❖ [Capa de Red](#)
- ❖ [Capa de Enlace de Datos](#)
- ❖ [Capa Física](#)

Elaboración: El siguiente resumen se hizo en colaboración de los siguientes boluditos:

- ★ Pepi
- ★ Valen
- ★ Sofi
- ★ Vene
- ★ Law
- ★ Marquidios
- ★ Migue
- ★ Tomi, a veces

Se recomienda discreción al leerlo, ningún miembro del equipo se hace responsable de los efectos secundarios que puede traer el aprendizaje de esta materia. Y sobre todo, tengan en cuenta que pueden experimentar momentos muy intensos aprendiendo dichos temas. Para más información, contactar con su profesor de preferencia, Juan 1 o Juan 2.



Seguinos en el Zulip para más consejos. (PEGI 18)

# Introducción

## Redes de computadoras

Conjunto de sistemas finales **interconectados**.

Dos hosts están **interconectados** si pueden intercambiar información entre ellos.

La **interconexión** se hace por medios de transmisión como cables, ondas, fibra óptica, etc.

Tipos de máquinas para **interconectar** por medio de redes:

- Hosts o sistemas finales (PCs, notebooks, smartphones, etc)
- Dispositivos IoT, que pueden:
  - Intercambiar datos con otros dispositivos interconectados.
  - Recolectar datos de otros dispositivos y procesarlos localmente.
  - Realizar tareas localmente y otras tareas dentro de la infraestructura de la red.

El intercambio de información entre hosts se hace por medio de señales que viajan en los medios de transmisión.

## Tipos de redes

The diagram illustrates the hierarchy of network types based on interprocessor distance. It features a vertical double-headed arrow on the left labeled "difusión" (diffusion) pointing upwards, and a horizontal arrow pointing right below it, labeled "Punto a punto" (point-to-point). To the right is a table with three columns: "Interprocessor distance", "Processors located in same", and "Example". The table rows show the following data:

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	
100 m	Building	Local area network
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	
1000 km	Continent	Wide area network
10,000 km	Planet	The Internet

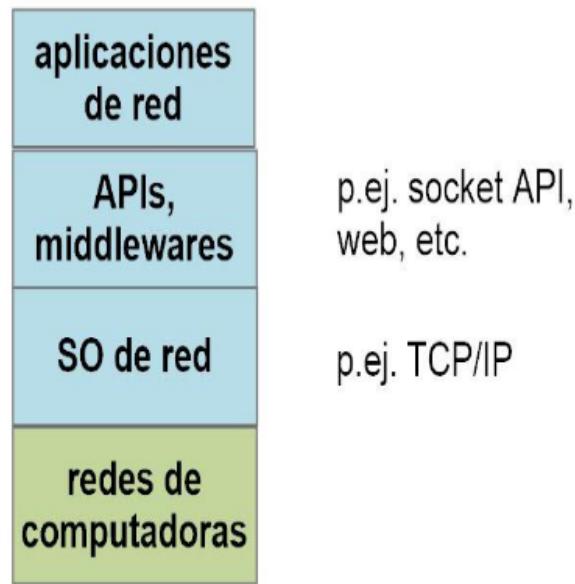
Para comunicar redes entre sí, se utilizan interredes (por ejemplo internet). Una **interred** es un conjunto de redes **interconectadas**. Las **puertas de enlace** conectan redes de distintas tecnologías.

Las redes de computadoras se usan para proveer **servicios**. Para ello, se crean **aplicaciones de red** (programadas mediante **APIs** o **Middlewares**), que se ejecutan en la internet. Para envío y recepción de mensajes se utilizan **protocolos**.

Los hosts acceden a la internet a través de proveedores de servicios de internet, para que dos hosts que están conectados a diferentes ISP se comuniquen entre sí, estas 2 ISP deben estar **interconectadas**.

**Problema:** Dados miles de ISP de acceso, ¿cómo conectarlos entre sí?

**Solución:** tener ISPs globales de tránsito que conectan los ISP de acceso.



Las ISP de acceso son interconectadas a través de redes ISP nacionales e internacionales de más alto nivel llamados ISPs de capa superior o globales de tránsito

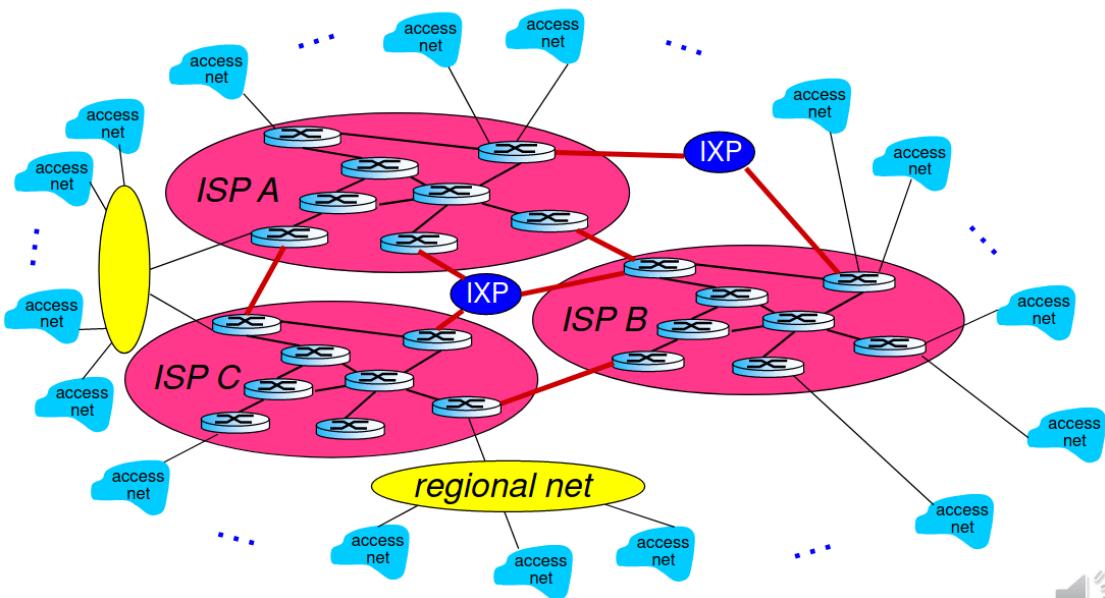
- Las ISP globales de tránsito deben estar interconectadas entre sí.
- Cada red ISP, ya sea de acceso o de capa superior, es manejada independientemente

**Problema:** Los ISP globales no tienen presencia en cada ciudad o región del mundo, esto implica que hay ISPs de acceso que no se pueden conectar a ISP globales.

**Solución,** en una región puede haber un ISP regional al cual se conectan los ISP de acceso en la región.

#### ¿Cuáles son las consecuencias de la solución anterior?

- Cada ISP regional se conecta con ISPs globales de tránsito
- Los ISP de acceso pagan al ISP regional al cual se conectan, y cada ISP regional paga al ISP global de tránsito al cual se conecta.
- En algunos lugares un ISP regional; puede cubrir un país entero y a ese ISP regional se conectan otros ISP regionales.



▪“tier-1” ISPs comerciales(p.ej. redes globales de tránsito) cobertura nacional e internacional.

- Redes proveedoras de contenido

• En el medio ISP regionales.

• Finalmente ISPs de acceso.

**IXP = internet exchange point**

### Redes de área amplia (WAN)

Una WAN cubre un área geográfica grande, como país o continente.

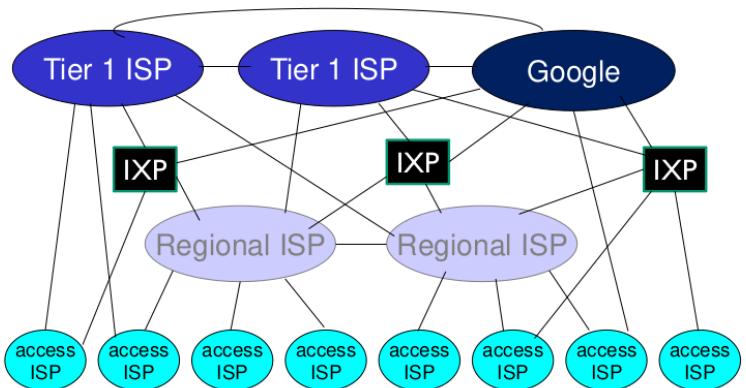
Una WAN se organiza en: Subred: varios enrutadores conectados entre sí forman un grafo

- Un arco representa cable que une 2 enrutadores.

- A una subred pueden estar conectadas computadoras o LAN enteras.

- Para ir de una máquina a otra hay distintas rutas alternativas.

## Estructura de la Internet



Para enviar mensajes en una WAN se utilizan algoritmos de **almacenamiento y reenvío**:

- El paquete sigue una ruta de enrutadores
- Se almacena en cada enrutador de la ruta
- Espera allí hasta que la línea de salida requerida esté libre y se reenvía al siguiente enrutador
- Los paquetes se pueden perder si se llena el buffer

Toma  $L/R$  segundos transmitir un paquete de  $L$ -bit en un enlace de  $R$  bps.

Demora de almacenamiento y reenvío:

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

$d_{queue}$  es la demora por encolado y depende de la congestión.

$d_{proc}$  es el procesamiento del nodo (chequeo de errores y determinar línea de salida)

$d_{trans}$  es la demora por transmisión.

$d_{prop}$  es la demora por propagación.

### Redes de área metropolitana (MAN)

- Redes de cable: red TV por cable
  - Cable coaxial para unir varias casas.
  - Elementos de conmutación unidos por fibra óptica para comunicar viviendas con distintos coaxiales.
  - Asimétricas (más bajada que subida de datos).
- Redes móviles: Redes inalámbricas de alta velocidad.

### Redes de área Local (LAN)

Una red de área local (LAN) es una red operada privadamente dentro de un edificio o casa.

Las LAN usadas por compañías se llaman redes empresariales.

- **Inalámbricas:** Máquinas comunicadas sin cable por medio de un access point
- **Ethernet:** Las máquinas se comunican entre sí por cables a un conmutador (switch).

Si una máquina envía un mensaje, todas las demás lo reciben.

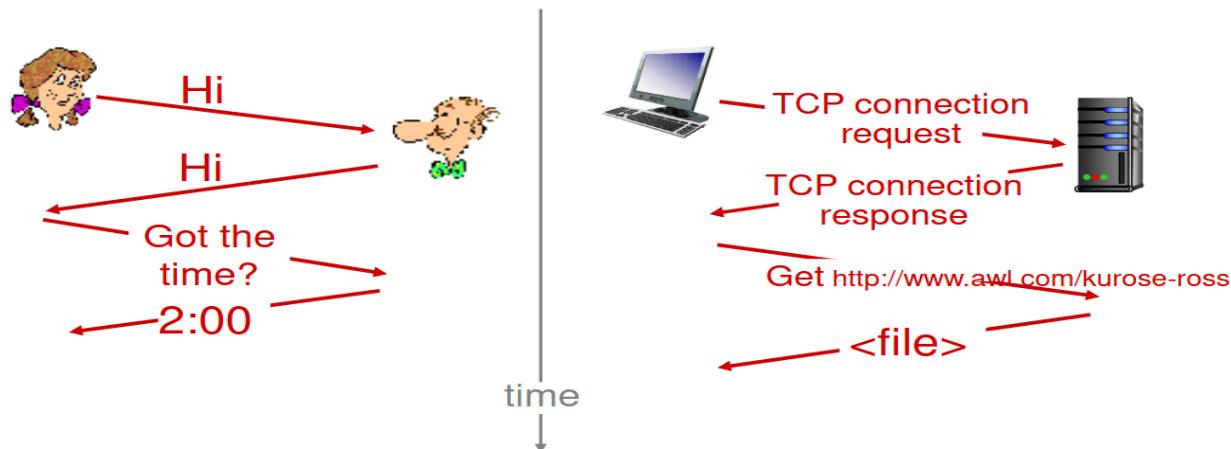
**Situación indeseable:** Se envían mensajes en una red de difusión y se pierden.

**Causa colisión**, más de una máquina manda simultáneamente un mensaje.

- Los mensajes colisionan y se dañan.

Protocolos:

### Un protocolo humano y un protocolo de redes de computadoras



Los sistemas operativos de red consisten de varios protocolos de comunicación. Estos definen: formato, orden de mensajes enviados y recibidos entre máquinas de la red, y acciones tomadas en la transmisión y recepción de mensajes.

**Jerarquías de protocolos:**

Los **sistemas operativos** de redes (SOR) están organizadas como una pila de capas o niveles, cada una construida arriba de la que está debajo de ella.

- La cantidad de capas, los nombres de las capas, sus contenidos y su función, difieren de un tipo de red a otro.

Las pilas de capas se usan para reducir la complejidad del diseño de los SOR.

**¿Cuál es el propósito de una capa en una arquitectura multicapa?**

1. Ofrecer ciertos servicios a las capas superiores
2. Ocultar la implementación a las capas superiores

Interfaces entre capas = operaciones y servicios primitivos ofrecidos por una capa inferior a una capa superior.

Una capa n se piensa como una conversación entre la capa n de una máquina con la capa n de otra máquina, sin tener que preocuparnos de ciertos problemas que resuelven las capas inferiores a la capa n. Para especificar cómo es esta conversación se definen protocolos

**arquitectura de red = conjunto de capas y protocolos = pila de protocolos.**

Comprender problemas de diseño a resolver en distintas capas:

- Problema: Hace falta un mecanismo para identificar a las máquinas de una red.
- Solución: Se usan direcciones para las máquinas.

**Control de flujo:** Sigue cuando un emisor muy rápido quiere saturar a un receptor muy lento

Situación indeseable: mensajes que llegan al receptor se pierden

**Causa:** un emisor rápido satura de datos al receptor hasta que este ya no puede almacenar más datos que le llegan y comienza a perder datos.

**Solución:** Uso de retroalimentación al emisor. Es decir, indicarle cuándo y cuánto puede enviar.

**Fragmentación de mensajes:** Cuando entre capas se quieren enviar mensajes estos deben dividirse en pequeños paquetes para no sobrepasar su **tamaño máximo** que normalmente imponen las capas, estos paquetes luego se vuelven a ensamblar cuando llegan a su destino.

**Situación indeseable:** mensajes que llegan no pueden ser aceptados en una capa.

**Causa:** los procesos son incapaces de aceptar mensajes que superan una cierta longitud

**Idea de solución:** fragmentar mensajes, transmitir fragmentos y re-ensamblar mensajes.

**Congestión:** Cuando un host quiere sobrecargar la red de paquetes y esto puede causar que se pierdan u ocasione demora en la llegada al host destino

**Situación indeseable:** los mensajes enviados de host de origen a destino se pierden antes de llegar o demoran demasiado en llegar.

**Idea de solución:** que máquinas emisoras se enteren de la congestión y reduzcan el tráfico de salida.

Distintos tipos de capas:

## Capa de aplicación

En la capa de aplicación tenemos las aplicaciones de red.

Hay dos opciones para desarrollar aplicaciones de red:

1. El programador para especificar la comunicación usa una interfaz para programas de aplicación(API).
2. El programador se apoya en middlewares para construir la aplicación red.

## Capa de transporte

La CT se ejecuta por completo en los hosts.

¿Qué cosas se debería solucionar la CT?

- Uso de temporizadores y las retransmisiones de paquetes.
- Uso de búferes y control de flujo.
- Evitar congestionar la red poniendo demasiados paquetes en ella. Control de congestión.

La capa de transporte tiene dos protocolos:

- **TCP (Transfer Control Protocol)**
- **UDP (User Datagram Protocol)**

## Capa de Red

- Algoritmos de almacenamiento y reenvío
- Control De Congestión.
- Resolver problemas que surgen cuando un mensaje tiene que viajar por redes de distinta tecnología para llegar a destino.

**Situación indeseable:** un mensaje demora demasiado en llegar

**Causa:** en determinadas redes (p.ej.WAN,internet,etc.) hay múltiples rutas entre el origen y el destino y justo se toma una ruta demasiado lenta o larga entre origen destino

De solucionar esto se encargan los algoritmos de enrutamiento.

Los mensajes viajan a su destino de forma independiente esto significa que pueden llegar en un orden diferente al que fueron enviados.

¿Cómo se distingue entre diferentes máquinas (que tienen una conexión a internet)?

**Direcciones IP:** 4 números entre 0 y 255 separados por '.'.

Para crear la ruta de los paquetes se usan protocolos de enrutamiento: se usan OSPF y BGP para enrutamiento de paquetes.

Comunicación de procesos:

**Procesos**: programas ejecutándose dentro de un host.

**Proceso cliente**: proceso que inicia la comunicación.

**Proceso servidor**: proceso que espera ser contactado

Los procesos en diferentes hosts se comunican intercambiando mensajes.

**¿Cómo se identifican los procesos?** Mediante direcciones IP y número de puerto.

## Capa de enlace de datos

Su propósito es transformar un medio de transmisión puro en una línea de comunicación que aparezca libre de errores de transmisión.

**Situación indeseable**: mensajes llegan con errores

**Causa**: medio físico de comunicaciones es imperfecto y ocasiona errores

## Capa Física

Transportar un stream de datos de una máquina a otra usando medios físicos.

**Enlace físico**: lo que yace entre el transmisor y receptor.

**Medios guiados**: Las señales se propagan en medios sólidos: copper, fiber, coaxial.

**Medios no guiados** : Las señales se propagan libremente, e.j., radio.

## Modelo Híbrido:

Función	Asuntos/problemas considerados
aplicaciones de red middleware	capa de aplicación
comunicación entre procesos	capa de transporte
envío de paquetes entre 2 hosts usando rutas entre ellos	capa de red
comunicación entre máquinas conectadas directamente entre sí	capa de enlace de datos
transporte usando medios físicos de un stream de datos	capa física

## Capa de Aplicación

En esta capa se encuentran las aplicaciones de red, cada una ofrece un servicio específico, con su propia forma de interfaz con el usuario.

La capa de aplicación se encuentra sobre la capa de transporte por lo cual usa los servicios de dicha capa.

## Opciones de desarrollo

- Interfaz para programas de aplicación (API): Conjunto básico de funciones. Por ejemplo socket API para software que se comunica sobre la internet.
- Middleware: Provee servicios al software de la aplicación. Es una función mucho más amplia, está autocontenido y es un software que corre de manera autónoma a parte de la aplicación que se esté usando.

## Arquitecturas

- **Cliente – Servidor:** un proceso cliente y un proceso servidor en distintas máquinas se comunican
  - Cliente manda solicitud al servidor
  - Cliente espera respuesta
  - Servidor recibe y procesa solicitud
  - Servidor envía respuesta al cliente
- **Peer-to-peer (P2P):** Por ejemplo bittorrent para distribución de archivos.
  - Hosts arbitrarios (compañeros/peers) se comunican directamente entre sí.
  - Mínimo o ningún apoyo en servidores.
  - Peers piden servicio de otros peers y proveen servicios en retorno
  - Nuevos peers traen nueva capacidad al servicio (+peers = +capacidad)
  - Los peers se conectan intermitentemente y cambian las direcciones IP

Características\Arquitecturas	Cliente - Servidor	Peer - To – Peer (P2P)
<b>Servidor</b>	Siempre presente (IP fija)	Mínimo o sin servidor
<b>Comunicación con clientes</b>	Nula	Constantemente
<b>Conexión clientes</b>	Intermitente	Intermitente
<b>Bajada de datos</b>	Pedido al servidor	Pedido a los demás clientes
<b>Tiempo de Distribución</b>	$D = \text{Max}\{\text{NF}/\text{Us}, \text{F}/\text{Dmin}\}$	$D = \text{Max}\{\text{F}/\text{Us}, \text{F}/\text{Dmin}, \text{NF}/(\text{Us} + \text{SumUi})\}$

N = Número de clientes; Us = Subida servidor; Ui = Subida compañero i; F = Tamaño archivo; Di = Descarga del compañero i; Dmin = descarga más rápida de un cliente (Min {D1,D2,...,Dn})

## Cliente-Servidor

UDP	TCP
-----	-----

Cliente crea datagrama con IP y puerto del servidor y los envía al servidor	Se ejecuta el servidor
Si llega, servidor lee datagrama	Servidor espera pedidos
Servidor envía respuesta especificando dirección y puerto cliente	Cliente requiere pedido
Si, llega cliente lee datagrama	Servidor acepta conexión
FIN DE LA CONEXIÓN	Cliente envía pedido (*)
	Servidor lee pedido y envía respuesta
	Cliente lee respuesta
	Cliente o vuelve a (*) o cierra conexión.
	Servidor cierra conexión

## BitTorrent

- Trozos de 256 Kb
- Compañeros envían y reciben trozos
- Tracker: lleva pista de los compañeros participando en torrent
- Torrent: grupo de compañeros intercambian trozos de archivos

### Cuando un compañero se une a torrent:

- No tiene trozos pero va a acumularlos a lo largo del tiempo
- Se registra con tracker para obtener la lista de compañeros
- Se conecta con un subconjunto de compañeros llamado vecinos
- Un compañero avisa periódicamente a tracker que está en BitTorrent

### Pedir trozos:

- Diferentes compañeros tienen diferentes subconjunto de trozos de archivos
- Periódicamente pedir a cada compañero la lista de trozos disponibles → conviene pedir primero los trozos menos comunes

### Enviar trozos:

- Se envían trozos a los top 4 mejores subidores del compañero (los que tienen mayor velocidad de envío) → el top 4 se reevalúa cada 10 segundos
- Cada 30 segundos se elige al azar otro compañero y comienza a enviarle trozos

## Protocolos

Cosas a definir en un protocolo de Aplicación	
Tipo de mensaje	Pedido o respuesta

<b>Sintaxis del mensaje</b>	Qué campos y como están delineados
<b>Semántica del mensaje</b>	Significado de los campos
<b>Reglas</b>	Cuándo y cómo los procesos envían y responden
<b>Estados de mensajes</b>	En qué consiste y cómo se mantiene
<b>Tipos de protocolos</b>	Abiertos o Propietarios

## Protocolo FTP

- Usado en transferencia de archivos hacia/desde host remoto.
- Servidor FTP: Puerto 21.
- Permite mensajes de control textuales e inspeccionar carpetas.
- Tipos de mensajes: Comando, respuesta y datos.

### Reglas de FTP:

1. Cliente FTP contacta con Servidor FTP en puerto 21, usando TCP
2. Cliente es autorizado en la conexión de control
3. Cliente inspecciona el directorio remoto, envía comandos sobre la conexión de control → comienza con identificación de usuario y contraseña.
4. Servidor recibe comando y comienza con transferencia de archivo → abre una segunda conexión de datos TCP (puerto 20).
5. Cuando termina la transferencia, el servidor cierra la conexión de datos

El servidor FTP mantiene el “estado”: directorio corriente o autenticación previa.

## Web

### Páginas Web

Pueden contener vínculos a otras páginas. Suelen contener texto. Suelen referenciar a varios objetos (HTML, imágenes, etc).

Las páginas/objetos se nombran usando URLs (localizadores uniformes de recursos).

Partes de una URL:

- Nombre del protocolo
- Nombre DNS de host que contiene la página
- Camino al archivo → nombre del archivo que contiene a la página

Si cambia el nombre de una máquina, la IP del dominio no va a cambiar. Solo habría que cambiar la asociación IP-dominio.

## Browsers

Sirven para ver la página web.

Cómo funcionan:

- A través del protocolo HTTP, pide una página u objetos al servidor web.
- Servidor web retorna la página/objetos solicitados.
- Browser interpreta el texto y los comandos de formateo que contiene y despliega la página adecuadamente formateada en pantalla.

Las páginas web se escriben en HTML para que los navegadores las entiendan.

Sitio web: Conjunto de páginas web relacionadas, localizadas bajo un único nombre de dominio, publicadas por al menos un servidor web. Pueden ser producidos por personas u organizaciones.

Home Page: Página de entrada al sitio web que sirve de guía hacia las páginas que contienen la información necesaria. Es la página que se carga por default.

**¿Cómo se sabe con qué máquina se va a comenzar la conexión TCP?** Se traduce el URL → Servidores DNS convierten nombres de dominio a direcciones IP.

#### Orden de comunicación entre Browser y Servidor web:

1. Cliente inicia una conexión TCP con el servidor web en puerto 80
2. Servidor web acepta la conexión del cliente
3. Mensajes HTTP(del protocolo) intercambiados entre browser y servidor web
4. La conexión TCP se cierra.

Con HTTP el servidor web no mantiene información acerca de pedidos pasados del cliente.

Los protocolos que mantienen el estado son complejos. El estado de la historia pasada debe ser mantenido. Si el servidor/cliente se caen sus visiones del estado pueden ser inconsistentes y deben reconciliarse.

No todas las páginas contienen solamente HTML. Para ello el servidor regresa con la página el tipo MIME de ella. Las páginas de tipo text/HTML se despliegan de manera directa.

Tipo MIME (Multipurpose Internet Mail Extension): Si no es de los integrados, el navegador consulta una tabla de tipos MIME que asocia un tipo MIME con un visor.

Para desarrollar visores se pueden utilizar: Plug-ins o aplicaciones auxiliares.

#### Plug-ins:

Es un modulo de código. El navegador los obtiene de un directorio especial del disco. Navegador instala un plug-in como una extensión del sistema. Los plug-in se ejecutan dentro del proceso del navegador, por lo tanto pueden acceder a la página actual y modificar su apariencia.

Interfaz del plug-in: Conjunto de procedimientos que todos los plug-in tienen que implementar para que el navegador pueda llamarlos.

Interfaz del navegador: Conjunto de procedimientos del navegador que el plug-in puede llamar.

Cuando un plug-in termina su trabajo se elimina de la memoria del navegador.

#### Aplicaciones auxiliares:

Se ejecutan en procesos separados del browser. No ofrecen interfaz al navegador ni usan servicios de este. Suelen aceptar el nombre de un archivo y lo abren y lo despliegan.

#### Servidores web:

Tiene páginas web y objetos que permiten construir páginas web.

Se le proporciona el nombre de un archivo correspondiente a una página a buscar y regresar.

Problema: Cada solicitud requiere un acceso al disco para obtener al archivo → Ineficiente → la página puede pedirse innumerables veces.

Solución: caché en la memoria.

**Arquitectura de un módulo front end y k módulos de procesamiento(hilos)** → hacer al servidor web más rápido.

Todos los MP(módulos de procesamiento) tienen acceso al caché.

Pasos de un servidor web con múltiples hilos:

1. Cuando llega una solicitud el front-end lo acepta y construye un registro corto que lo describe.
2. Entrega el registro a uno de los MP
3. MP verifica la caché, si el archivo se encuentra ahí.
4. Si está, actualiza el registro para incluir un apuntador al archivo
5. No está, MP inicia una operación de disco, cuando el archivo llega del disco se incluye en el caché y se regresa al cliente.
6. Mientras los MP están bloqueados haciendo operaciones de disco, otros MP pueden estar trabajando en otras solicitudes
7. Conviene tener múltiples discos, más de un disco puede ser ocupado al mismo tiempo.

#### **Funcionamiento del MP:**

1. Resuelve el nombre de la página web solicitada. No siempre se solicita un URL completo. Manejo de solicitud entrante sin el nombre real del archivo.
2. Control de acceso en la página web → páginas no disponibles para el público general o si la solicitud se puede satisfacer a partir de la identidad y ubicación del cliente → servidores listan el tipo de acceso. Se puede prohibir que dominios particulares accedan a la web.
3. Verifica el cache
4. Obtiene del disco la página solicitada o ejecuta un programa para construirla
5. Determina el tipo MIME de la página a través de un algoritmo, se incluye en la respuesta
6. Regresa la respuesta al cliente.
7. Realiza una entrada en el registro del servidor.

## Cookies

En los pedidos y respuestas HTTP se envía información del estado de sesión → se usan cookies. Son pequeños archivos o cadenas de a lo sumo 4 KB. Y su contenido es de la forma nombre = valor.

### Campos de una cookie:

1. Dominio: Se pueden almacenar hasta 20 cookies por cliente
2. Ruta en la estructura del directorio del servidor → Identifica qué partes del árbol de archivos del servidor podrían usar el cookie
3. Contenido: nombre = valor.
4. Expira → Si este campo está ausente → navegador descarta el cookie cuando sale. Sino proporciona una fecha y una hora → mantiene la cookie hasta que expira ese horario.
5. Seguro → Indica que el navegador solo puede retornar la cookie a un servidor usando transporte seguro → aplicaciones seguras. (bancarias por ej).
6. Eliminación de una cookie del disco duro del cliente → el servidor la envía nuevamente con una fecha caducada.

### Directorio de cookies:

El navegador puede almacenar cookies en el disco duro de la máquina del cliente. La información de las cookies (del lado del servidor) se almacenan en una base de datos.

Comunicación de las cookies al cliente → cuando un cliente solicita una página web → el servidor envía una cookie junto a la página.

Comunicación de las cookies al servidor web:

- Antes que un navegador solicite una página a un sitio web, verifica su directorio de cookies para ver si el dominio al que está solicitando la página ya colocó alguna cookie
- Si lo hizo → todas las cookies para ese dominio se incluyen en el mensaje de solicitud
- Cuando el servidor web las obtiene las interpreta de la forma que deseé.

### Necesidades para un protocolo para la web:

- pedido de páginas/objetos/ejecución de programas que generan páginas
- manejos del estado de sesión
- mantener el sistema de archivos del servidor web
- recepción de páginas por un browser
- seguridad (encriptación del mensaje)
- feedback adecuado cuando no se puede responder a los pedidos
- comunicación confiable.

## HTTP

Hipertexto porque las páginas tienen enlaces a otras páginas.

- Transfiere páginas de servidores web a navegadores y manda pedidos de navegadores a servidores web
- Tipos de mensajes: HTTP-Request y HTTP-Response.

<u>HTTP no persistente</u>	<u>HTTP persistente</u>
Un solo objeto se manda por TCP luego se cierra la conexión	Múltiples objetos pueden ser enviados a través de una única conexión TCP entre el cliente y el servidor
Descargar múltiples objetos requiere múltiples conexiones	Soportado por HTTP 1.1
HTTP 1.0 → establece conexión por cada solicitud y se libera al recibir respuesta	

RTT: tiempo necesario para que un paquete pequeño viaje del cliente al servidor y de regreso.

#### **Tiempo de respuesta de HTTP no persistente:**

- RTT para iniciar la conexión TCP
- RTT para el pedido HTTP y regreso de los primeros bytes de la respuesta HTTP
- Tiempo de transmisión
- Tiempo de respuesta = **2RTT + tiempo de transmisión**

#### Pedidos HTTP :

Información:

- Si se quiere recibir una página: URL del documento y especificación del programa que genera una página web.
- Tipo de acción que se quiere hacer en el sistema de archivos del servidor web
- Enviar información sobre la máquina/software del cliente → Servidor retorna páginas adecuadas
- Mandar información del estado de sesión para que el servidor se entere
- Restricciones sobre el tipo de páginas que el cliente puede aceptar.
- Indicar el tipo de acción que se quiere hacer → usar un campo de acción con el tipo de acción requerido → primera palabra de la primera línea es el nombre del método.

#### HTTP Métodos:

	<b>Method</b>	<b>Description</b>
Fetch a page →	<b>GET</b>	Read a Web page
	<b>HEAD</b>	Read a Web page's header
Used to send input data to a server program →	<b>POST</b>	Append to a Web page
	<b>PUT</b>	Store a Web page
	<b>DELETE</b>	Remove the Web page
	<b>TRACE</b>	Echo the incoming request
	<b>CONNECT</b>	Connect through a proxy
	<b>OPTIONS</b>	Query options for a page

#### **Para subir el input de un formulario hay dos opciones:**

- POST: página web a menudo contiene input de formulario → input se encuentra en un campo llamado cuerpo de la entidad
- GET: usa URL → input se sube en el campo URL de la línea de pedido

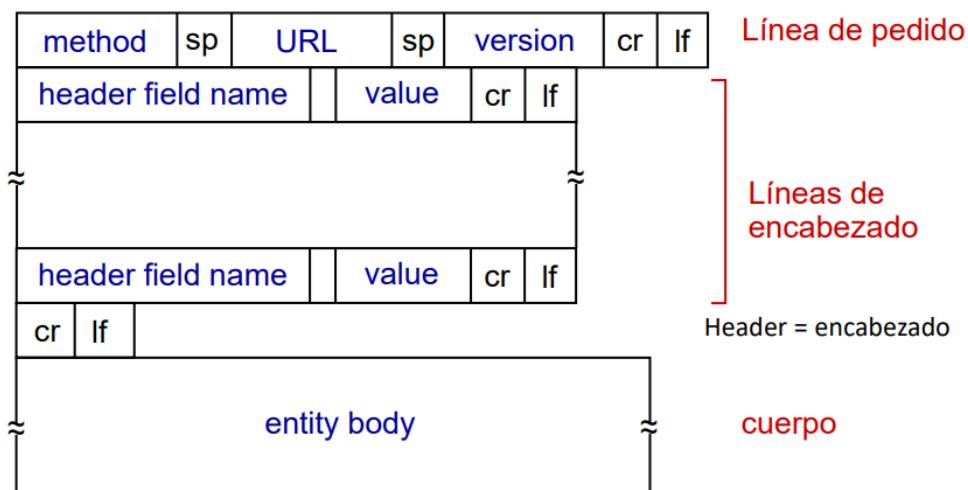
### Método para actualizar páginas del servidor web:

- PUT: Sube un archivo en el cuerpo de la entidad en el campo especificado por el campo URL
- DELETE: Borra el archivo especificado en el campo URL

### Otros métodos:

- HEAD: solicita el encabezado del mensaje sin la página real.
- OPTIONS: cliente consulta con el servidor y obtiene los métodos y encabezados que pueden ser usados con esa página.

Para tratar información diversa, se indica el tipo de información que se trata y luego la información en si → encabezados de pedido → pares encabezado, valor.



- A la línea de solicitud le pueden seguir líneas adicionales llamadas **encabezados de solicitud**.

## Respuestas HTTP

Información:

- Feedback adecuado sobre el pedido realizado
- Página o documento solicitado
- Información sobre el tipo de documento enviado → Tipo MIME
- Información de estado de sesión para mantener actualizado al cliente.
- Feedback → Línea de estado: Contiene un código de estado de 3 dígitos que indica si la solicitud fue atendida y si no, porqué.

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

Los encabezados de respuesta son pares (nombre de encabezado, valor).

Partes de una respuesta HTTP:

1. Línea de estado

2. (Opcional) encabezado de respuesta

3. Datos.

## Encabezados HTTP:

- Mensajes HTTP suelen tener encabezados
- Son de pedido, respuesta o ambos
- Proveen información a ser procesada por el receptor del mensaje
- Fijan restricciones que deben cumplir los mensajes futuros
- Proveen información sobre eventos importantes
- Datos de los estados de sesión

Header	Type	Contents
User-Agent	Request	Information about the browser and its platform
Accept	Request	The type of pages the client can handle
Accept-Charset	Request	The character sets that are acceptable to the client
Accept-Encoding	Request	The page encodings the client can handle
Accept-Language	Request	The natural languages the client can handle
Host	Request	The server's DNS name
Authorization	Request	A list of the client's credentials
Cookie	Request	Sends a previously set cookie back to the server
Date	Both	Date and time the message was sent
Upgrade	Both	The protocol the sender wants to switch to
Server	Response	Information about the server
Content-Encoding	Response	How the content is encoded (e.g., gzip)
Content-Language	Response	The natural language used in the page
Content-Length	Response	The page's length in bytes
Content-Type	Response	The page's MIME type
Last-Modified	Response	Time and date the page was last changed
Location	Response	A command to the client to send its request elsewhere
Accept-Ranges	Response	The server will accept byte range requests
Set-Cookie	Response	The server wants the client to save a cookie

## Páginas estáticas:

Documentos en algún formato → usualmente HTML → actualmente se usa HTML5

## HTML

- Lenguaje estándar para crear páginas web
- estructura de una página web
- indica al navegador cómo mostrar el contenido
- sintaxis similar a XML
- permite crear páginas que incluyan texto, gráfico, hipervínculos, etc
- tablas y formularios.
- Un documento HTML es una serie de elementos. Un elemento es contenido encerrado entre etiquetas. Las etiquetas tienen la forma <nombre>. Las etiquetas a su vez pueden tener atributos, un atributo es de la forma nombre = "valor".

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h $n$ > ... </h $n$ >	Delimits a level $n$ heading
<b> ... </b>	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
<ul> ... </ul>	Brackets an unordered (bulleted) list
<ol> ... </ol>	Brackets a numbered list
<li>	Starts a list item (there is no </li>)
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a Horizontal rule
	Displays an image here
<a href="..."> ... </a>	Defines a hyperlink

### Páginas dinámicas:

Las páginas estáticas son muy ineficientes cuando la información cambia frecuentemente o la información de la página varía de acuerdo a parámetros.

La solución es usar páginas dinámicas:

- páginas HTML generadas por un programa del lado del servidor
- tienen parámetros de entrada
- estos parámetros suelen ser ingresados por formularios HTML

Pasos para crear una página dinámica:

1. El usuario llena un formulario y lo envía
2. se envía mensaje al servidor web con el contenido del formulario → se lo envía a un programa o secuencia de comandos → el programa lo procesa
3. El programa solicita información a un servidor de base de datos
4. El servidor de base de datos responde con la información requerida
5. El programa genera una página HTML personalizada y la envía al cliente
6. El browser muestra la página

Se pueden omitir los pasos 3 y 4 si el servidor no está conectado a una base de datos.

Con un URL no basta para especificar la página dinámica deseada.

- necesita parámetros para crear la página dinámica
- hace falta poder ingresar los parámetros en el pedido HTTP

Solución 1: el URL contiene nombre de programa y parámetros. Los parámetros son ingresados por medio de formulario HTML.

- parámetros: nombre = valor
- se separan del nombre del programa con ‘?’
- se separan parámetros con ‘&’

Solución 2: Los parámetros se ingresan separados por ‘&’ en el cuerpo de la entidad.

La solución 2 se utiliza cuando los parámetros ocupan mucho espacio → límite de una URL 2048 caracteres.

### Formulario HTML:

Tag	Description
<form action="" method =""> ... <\form>	Declara un formulario. Action es URL de la página ejecutable que procesa formulario. Method especifica cómo los datos se mandan al servidor (p.ej. GET, POST)
<select> ... <\select>	Para especificar una lista de la que usuario elige un elemento.
<option value = ""> ... <\option>	Para indicar opción de <select>
<textarea rows = "" cols=""> ... </textarea>	Control de ingreso de texto de varias líneas
<input name="" type="" value="">	Permite definir campo de input donde type puede ser: button, radio, password, text, submit, checkbox, hidden, etc.

Dentro de select se usa option.

## PHP

Define páginas dinámicas mediante la inserción de comandos especiales (scripts) dentro de páginas HTML. (permite definir clases y objetos).

Para utilizar PHP, el servidor web debe entender PHP:

- Código PHP interpretado por el servidor web
- Diseñado para trabajar en el servidor web Apache
- puede usarse en la mayoría de los servidores web

Utilidad de PHP:

- puede generar contenido de página dinámica
- operar con archivos del servidor
- recolectar datos de formulario
- enviar y recibir cookies
- acceder a encabezados de pedido HTTP
- definir encabezados de respuesta HTTP
- acceder a base de datos

Construcción	Description
<?php ... ?>	PHP script Puede ir en cualquier lugar del documento
'\$' NAME	Declaración de variable Case sensitive
echo EXPR	Para mostrar datos en pantalla
//	comentarios
Define(name, value)	Definición de constantes
VARIABLE = EXPR	Igual que en C (+=, -=, *=, /=)
Include 'filename'	Toma el texto/código/markup en un archivo y lo copia en el archivo que usa la sentencia <i>include</i>

## Acceder a campos de formulario

- `$_POST` → recolectar datos de un formulario con método POST
- `$_GET` → recolectar datos de un formulario con método GET

Ejemplo: `$_POST['fname']` → datos del campo fname

## Operadores

- De asignación y comparación para los distintos tipos de datos.

## Acceder a información de encabezados HTTP:

`$_SERVER` contiene la información de los encabezados, caminos y localización de scripts.

Para acceder a los encabezados, necesitamos agregar como argumentos:

- `HTTP_USER_AGENT`
- `SERVER_ADDR`
- `SERVER_NAME`
- `SERVER_SOFTWARE`
- `SERVER_PROTOCOL`
- `REQUEST_METHOD`
- `REQUEST_TIME`
- `QUERY_STRING`
- `HTTP_ACCEPT`
- `HTTP_ACCEPT_CHARSET`
- `HTTP_HOST`

entre otros.

## Definir encabezados de respuesta HTTP:

Se utiliza la función `header()` → se debe fijar encabezados antes de la etiqueta `<html>`.

## Definición de cookies:

Se utiliza la función `setcookie()` → define una cookie que va a ser enviada junto con los encabezados HTTP. → se debe fijar antes de la etiqueta `<html>` → antes de generar cualquier salida.

se utiliza de la siguiente forma: `setcookie(name, value, expire, path, domain, httponly)`

## Acceder al valor de una cookie:

- `$_COOKIE` → retorna el valor de una cookie
- `htmlspecialchars()` → convierte caracteres especiales en entidades html

Forma de uso: `$_COOKIE['fname']` → valor de la cookie fname

## Críticas al modelo actual:

Que el servidor web tenga que construir páginas dinámicas puede ser ineficiente por los siguientes motivos:

- la página nueva a generar dinámicamente puede tener muchas partes en común con la que tiene el browser → estas partes se generarían de vuelta y serán nuevamente enviadas → van a tener que ser reinterpretadas por el browser.
- El cliente se queda bloqueado esperando luego de hacer un pedido HTTP y recién puede continuar con la ejecución cuando recibe una página → pedidos sincrónicos.

## Liberando al servidor web y aprovechando el poder del cliente:

El servidor web en respuesta a un pedido HTTP solo enviará datos para actualizar parte de una página web en el navegador y el browser se ocupa de actualizar la interfaz del usuario.

- Pedidos asincrónicos → el cliente puede seguir ejecutando tras realizar un pedido
- Cuando llegan datos del servidor web se genera e inserta la parte nueva de la IU y se deja igual el resto a preservar.

Implementación:

- Se pide el código de la IU de la aplicación al servidor web → escrito en HTML + JavaScript + Pedidos HTTP asincronicos → IU enriquecida similar a la IU de las aplicaciones de escritorio

- Los datos recibidos de un pedido asíncrono son procesados por el cliente para actualizar la IU

## JavaScript:

Construcción	Description
<script> ... </script>	Para insertar un script en JavaScript
function NAME (PARAMETERS) {BODY}	Declaración de funciones
Var NAME = EXPR;	Declaración e inicialización de variables
Objeto document	Representa la página siendo visualizada
Document.write(t)	Escribe texto <i>t</i> en el stream de salida de la página siendo visualizada.
Método document.getElementById(X)	Retorna elemento de identificador <i>X</i> . <i>Es el elemento con atributo id y valor X.</i>
Propiedad <i>e.innnerHTML</i>	Contenido del elemento <i>e</i>

Elementos HTML pueden tener atributo id → este debe ser único en todo el documento, sirve para referirse a ese único elemento.

## Procesamiento de eventos en JavaScript:

La reacción a un evento puede significar hacer cambios en la IU o hacer cierta tarea de procesamiento en el cliente o hacer pedidos al servidor web y mostrar respuesta en pantalla.

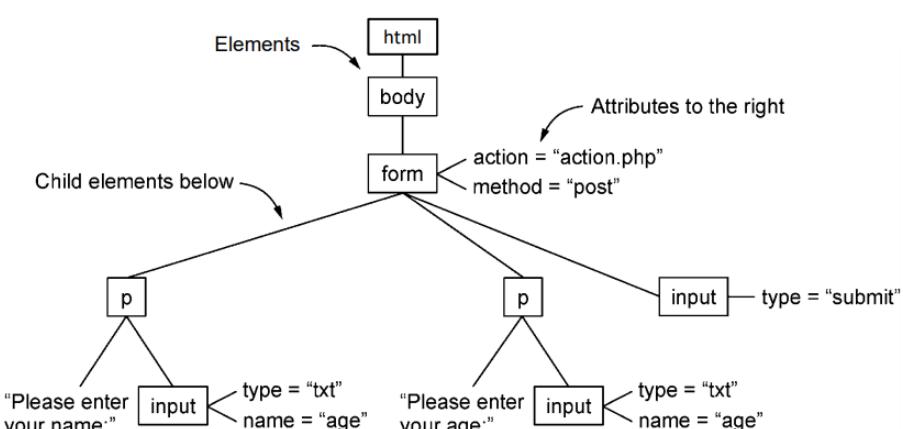
Definir reglas ECA (evento-condición-acciones):

- Si ocurre un evento y se da una condición entonces ejecutar una secuencia de acciones
- nombre del evento: abort, click, focus, load, select, submit, onload.
- Los eventos se asocian a etiquetas de páginas HTML → insertar atributo on[nombre-evento]
- La parte de acciones es el procesamiento es el procesamiento de un evento en sí → cuando ocurre un evento, ejecutar una serie de acciones o función JavaScript → encerradas en <script> dentro de <head>

## DOM (Document Object Model):

Representación de una página HTML que es accesible a programas. La página está estructurada como un árbol DOM → estructura jerárquica de los elementos HTML.

Además al lado de un elemento puede uno dibujar sus atributos. Permite a los programas cambiar partes de una página sin necesidad de sobreescibir toda la página. JavaScript puede acceder al modelo DOM. Se pueden tener scripts que modifican el documento HTML siendo visualizado.



JavaScript usa '.' para navegar a través de las propiedades y referencias asociadas con el documento.

- Jerarquía del árbol DOM es navegada comenzando con el objeto window → representa el navegador
- Window tiene una propiedad llamada document → representa a la página HTML
- Objeto document → tiene una colección de formularios llamada forms → indexada por el número de aparición en el documento, comienza en 0.
- Un formulario tiene una colección de elementos llamada elements → indexada por el número de aparición en el formulario → comienza en 0.

### Contenedores:

<div> → Define una división o sección en un documento HTML. Contenedor para otros elementos HTML. Pueden anidarse unos dentro de otros y permite definir una jerarquía de contenedores y organizar una página.

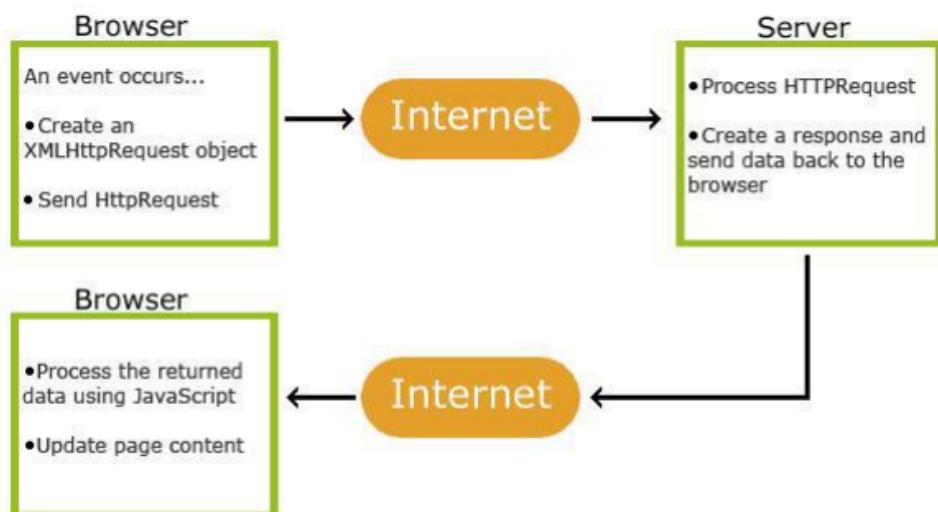
- Algunas sentencias y métodos que permiten modificar una página HTML siendo visualizada.

Construcción	Description
e.innerHTML = c;	Cambia contenido HTML de e por c
e.a = v;	Cambia valor de atributo a de elemento e por v.
e.setAttribute(a,v)	Crea en elemento e el atributo a con valor v. Si a ya estaba , solo cambia su valor.
document.createElement(N)	Crea elemento HTML de etiqueta N
document.createTextNode(T)	Crea nodo de texto T
e.appendChild(n)	Se agrega elemento n como último hijo del elemento e
e.removeChild(n)	De elemento e se remueve subelemento n.
e.replaceChild(n1,n2)	De elemento e se reemplaza subelemento n1 por elemento n2.

## Primitivas para hacer y procesar pedidos/respuestas HTTP

Construcción	Description
Objeto XMLHttpRequest	Objeto usado para intercambiar datos con servidor web.
Método Open(method, url, async, user, psw)	Para especificar el pedido HTTP
Método send()	Para enviar el pedido HTTP
Método setRequestHeader()	Para fijar encabezados del pedido HTTP
Propiedad responseText de objeto XMLHttpRequest	Se refiere a la respuesta del servidor como un string
Propiedad readyState	Estatus del XMLHttpRequest
getAllResponseHeaders()	Retorna toda la información de encabezados de respuesta

## Pasos para hacer pedido HTTP y contestarlo:



1. Especificar pedido HTTP con open()
2. Un pedido HTTP se envía con send() → pedido de datos o texto
3. Servidor obtiene los datos pedidos → crea respuesta HTTP y la envía al navegador
4. El cliente recibe la respuesta → procesa los datos → modifica la página actual → primitivas de IU

### Manejo de pedidos HTTP al servidor:

El objeto XMLHttpRequest puede usarse para intercambiar datos con un servidor web.

```
` var = new XMLHttpRequest();`
```

Se especifica el pedido HTTP con método open().

```
` open(method, url, async, usr, psw)`  
  - method → GET o POST  
  - url → localización del archivo  
  - async → true si es asíncrono  
  - user y password
```

Tipos de pedidos:

- Pedidos asíncronos → cliente hace pedido y sigue trabajando
- Pedidos sincrónicos → cliente hace pedido y se bloquea esperando una respuesta.

Para enviar pedido GET → send()

Para enviar pedido POST → send(string)

Para fijar encabezados → setRequestHeader()

### Manejo de respuestas HTTP del servidor:

La propiedad `responseText` retorna la respuesta del servidor como un string.

La propiedad `readyState` mantiene el estatus del XMLHttpRequest

La propiedad `onreadystatechange` → define una función a ser ejecutada cuando el readyState cambia

- conexión con el servidor establecida
- pedido recibido
- procesando pedido
- pedido terminó y respuesta lista

`getAllRequestHeaders()` → retorna toda la información de encabezados de respuesta del servidor

`getResponseHeader()` → retorna un encabezado específico de la respuesta.

### XML:

Extensive Markup Language. Se utiliza para definir datos:

- Nombre de los elementos de datos son expresados mediante etiquetas (`<nombre>`)

- Un elemento de datos de nombre n consiste en información cerrada entre etiquetas. `<n>y</n>`
- Pueden ser construidos por otros elementos anidados o solo texto

La propiedad `responseXML` retorna la respuesta como un documento XML.

`d.getElementsByTagName(N)` retorna los elementos XML dentro del documento d que tiene nombre de etiqueta N

Como recorrer un elemento XML:

- pensar el documento XML como un árbol → recorrerlo usando expresiones de camino → cada sección separada por '.'.
- `e.childNodes[i]` → retorna el nodo hijo i de e
- `e.nodeValue` → si es un elemento de solo texto retorna el texto.
- HTML respeta XML → se puede generalizar el árbol DOM para XML.

## AJAX:

Conjunto de tecnologías que trabajan juntas para permitir que las aplicaciones web sean tan eficientes y poderosas como las aplicaciones de escritorio tradicionales

- HTML y CSS → presentar información como páginas
- DOM → cambiar partes de las páginas mientras son vistas
- XML → permitir que los programas intercambien datos de aplicación con el servidor
- comunicación asincrónica → del cliente con el servidor web para enviar y retornar datos
- JavaScript → lenguaje para ligar todas estas facilidades.

## Capa de Transporte

Propósito de la capa de transporte

La **capa de transporte (CT)** provee comunicación lógica entre procesos de aplicación que se ejecutan en diferentes sistemas finales.

- esto no lo puede hacer la capa de red CR.
- La CT se implementa (salvo alguna excepción) solo en los sistemas finales.

**Comunicación lógica** como si los hosts ejecutando los procesos estuvieran directamente conectados.

Para **mejorar la calidad** los servicios de la CR:

- P.ej retransmisiones de paquetes perdidos en redes no orientadas a la conexión.
- P.ej cuando hay congestión en la red, regulando de manera fina la variación de la tasa de transmisión de paquetes de los hosts.

La CT se ejecuta por completo en los hosts/sistemas finales.

La CT confía en los servicios de la CR.

**Entidad de transporte (ET)** = software/hardware de la CT.

**Problemas que soluciona la capa de transporte:**

- Uso de temporizadores y las retransmisiones de paquetes.
- El direccionamiento explícito de los destinos.
- Uso de búferes y control de flujo.
- Evitar congestionar la red poniendo demasiados paquetes en ella.
- Cuando la CR pierde paquetes, la CT puede solucionarlo.
- No se preocupa de si un paquete se pierde por congestión.
- O si una máquina se quiere conectar con un servidor que no se está ejecutando.

**Segmento** unidad de datos del protocolo de transporte.

**Confirmaciones de recepción** de paquetes enviados.

**Tipos de paquetes que deben ser confirmados**

- paquete de datos
- paquetes con información de control

Si pasa demasiado tiempo sin una confirmación se asume que se perdió un paquete(suele suceder por congestión) → se usan temporizadores.

Si altero el flujo de datos estoy entregando un mensaje que el emisor quiere enviar que NO es y la capa de aplicación va a hacer algo que no tiene que hacer.

#### Para la entrega ordenada de segmentos el host puede:

- usar números de secuencia → números respetando el flujo de segmentos de la capa de aplicación.
- usar temporizadores de retransmisión por cada número de segmento enviado.
- confirmación de recepción (**ACK**).
- si caduca el temporizador sin recibir un **ACK** → retransmitir el segmento.
- reensamblar y enviar a la capa de aplicación los segmentos recibidos (en orden).

TCP proporciona un flujo de bytes confiables de extremo a extremo a través de una interred no confiable. Se adapta dinámicamente a las propiedades de la interred y se sobrepone a muchos tipos de fallas.

#### Problemas que resuelve TCP:

- retransmisión de paquetes → uso de números de secuencia, ACKs y temporizadores
- fijar la duración de temporizadores retransmisión
- manejo de conexiones entre pares de procesos
- direccionamiento
- control de congestión
- control de flujo

Una ETCP acepta flujo de datos a transmitir de procesos locales.

El servicio TCP se obtiene al hacer que tanto el servidor como el cliente creen sockets. **Dirección de un socket = IP + puertos**. Servicio TCP a través de una conexión entre el socket emisor y el socket receptor.

Un socket puede usarse para múltiples conexiones:

- dos o más conexiones pueden terminar en el mismo socket
- conexiones se identifican mediante los identificadores de sockets de los dos extremos.

Cada byte de un flujo de datos a enviar en una conexión TCP tiene su propio número de secuencia (32 bits). → límite en el tamaño de los datos.

#### Segmentos → encabezado TCP ++ Datos

Cada red tiene una unidad máxima de referencia (MTU) → cada segmento debe caber en el MTU (1500 B).

Los datagramas que llegan podrían llegar en orden incorrecto. Esto sucede en redes de datagramas.

1. Cuando un transmisor envía un segmento → inicia un temporizador.
2. Cuando llega a destino → ETCP receptora → devuelve ACK, es el siguiente número de secuencia.
3. Si el temporizador expira antes del ACK → emisor envía nuevamente el segmento

#### Campos de un segmento TCP:

1. encabezado fijo → 20 bytes
2. Opciones de encabezado en palabras de 32 bits
3. Datos opcionales

Los segmentos sin datos se usan como ACKs o paquetes de control.

#### Direccionamiento:

## **¿Cómo hacer para que un procesador adecuado atienda las necesidades de una máquina cliente?**

- El cliente conoce cual es el procedimiento adecuado para un servicio en particular(Servidor activo).
- El cliente podría no saber cuál es el procedimiento adecuado para un servicio particular.(Servidor activo).
- El cliente si lo sabe, pero el servidor no está activo .

### El cliente conoce el servidor (Servidor activo)

Para conectarse al servidor existe un proceso especial llamado **servidor de directorio** que para cada tipo de servicio sabe cuáles son los puertos de los servidores que prestan ese tipo de servicio

**servidor de directorio** siempre está escuchando esperando una señal de CONNECT.

#### Procedimiento:

1. El usuario establece una conexión con el servidor de directorio (que escucha en un puerto bien conocido).
2. El usuario envía un mensaje especificando el nombre del servicio.
3. El servidor de directorio le devuelve la dirección puerto.
4. El usuario libera la conexión con el servidor de directorio y establece una nueva con el servicio deseado.

#### **Creación de un servicio nuevo:**

- El servicio nuevo debe registrarse en el servidor de directorio, dando su nombre de servicio como la dirección de su puerto.
- El servidor de directorio registra esta información en su base de datos.

### El cliente conoce el servidor (Servidor inactivo):

Usar un servidor que ejecuta los servidores inactivos: **protocolo inicial de conexión**

#### Procedimiento:

- Un usuario emite un CONNECT más el puerto de servicio activo
- Como el Servidor de Procesos tiene los servidores inactivos para esa solicitud, se conecta al protocolo inicial de conexión
- el protocolo inicial de conexión genera(activa) algun servidor con capacidad de atender la solicitud y le hereda la conexión

#### **Control de flujo:**

La ET emisora debe manejar búferes para los mensajes de salida porque puede hacer falta retransmitirlos. El emisor almacena en búfer todos los segmentos hasta que se confirma su recepción.

Hay que evitar que un host emisor rápido desborde a un host receptor lento.

#### **¿Por qué control de flujo en la CT si la CR también lo tiene?**

El receptor puede demorarse en procesar mensajes debido a problemas en la red

- pérdida de segmentos
- no se puede procesar segmentos porque faltan anteriores

#### **Protocolo parada y espera:**

- Una vez que un emisor manda un segmento y para de enviar (parada).
- El emisor espera por un ACK del segmento (espera).
- Si llega el ACK a tiempo se envía el siguiente paquete.

- Si se pierde un pkt o un ACK, expira un temporizador de retransmisiones y se va a tener que mandar de nuevo.

No existe el problema de desbordamiento ya que siempre tiene la confirmación de recepción.

### Procedimiento:

- **Suposición:** el canal de comunicaciones subyacente puede perder paquetes (de datos, de ACKs)
  - Los paquetes tienen N° de secuencias,
  - Se trabaja con Ack
    - El receptor debe especificar N° de secuencia del paquete siendo confirmado.
  - Se usan retransmisiones de paquetes.
    - Para esto se requiere de uso de temporizadores.

### Comportamiento del emisor:

1. El emisor envía paquete P y para de enviar.
2. **Espera:** El emisor espera una cantidad "razonable" de tiempo para el ACK
3. Si llega el ACK a tiempo, se envía siguiente paquete. Goto 2.
4. Sino se retransmite paquete P. Goto 2.
- Si hay paquete o ACK demorado pero no perdido:
  - La retransmisión va a ser un duplicado con igual N° de secuencia ; luego se descarta.

- Desempeño pobre.
- el protocolo limita el uso de recursos físicos
- RTT es tiempo de ida y vuelta de un bit: RTT.

Si el tiempo de transmisión de un segmento es mucho menor que el RTT, la utilización del canal en parada-espera va a ser muy baja.

Se nota cuando las máquinas están muy alejadas entre sí, ya que se agranda el problema → termina siendo ineficiente.

### Protocolos de Tubería:

**Tubería:** el emisor puede enviar múltiples paquetes al vuelo a ser confirmados.

- El rango de números de secuencia debe ser incrementado
- Uso de búferes en el emisor y/o el receptor.

Hay dos formas genéricas de protocolos de tubería:

- Repetición Selectiva
- Retroceso N

**Situación:** Se perdió una confirmación de recepción y se envió el paquete de nuevo. El mismo paquete llega dos o más veces al receptor y la capa de transporte la pasa a la capa de aplicación más de una vez. Para evitar esto se asignan números de secuencia a los paquetes que salen.

### Retroceso-N:

- Receptor envía *ack acumulativo*
  - No confirma paquetes si hay un agujero.
- El emisor tiene un timer para el paquete más viejo no confirmado
  - Cuando expira el timer retransmite todos los paquetes no confirmados.

### Repetición selectiva:

- El receptor envía *confirmaciones individuales* para cada paquete
- El emisor mantiene un timer para cada paquete no confirmado
  - Cuando el timer expira, retransmite solo ese paquete no confirmado.

## Retroceso N:

Se envía un ack acumulativo →

Ejemplo: Enviamos una rafaga de 10 paquetes con número de secuencia entre 0 y 9, el receptor solo devuelve un ack de respuesta con número de secuencia 9 para confirmar la llegada de los 10 paquetes.

Si se pierde alguno de esos paquetes el receptor descarta todos los paquetes de la rafaga y no devuelve ningún ack para que el emisor envíe toda la rafaga de nuevo.

- Receptor envía ack acumulativo (todos los segmentos anteriores se recibieron bien).
- Al expirar el timer del segmento más viejo no confirmado, retransmite todos los segmentos no confirmados → si pierdo un segmento debe enviar todos los que le siguen otra vez.

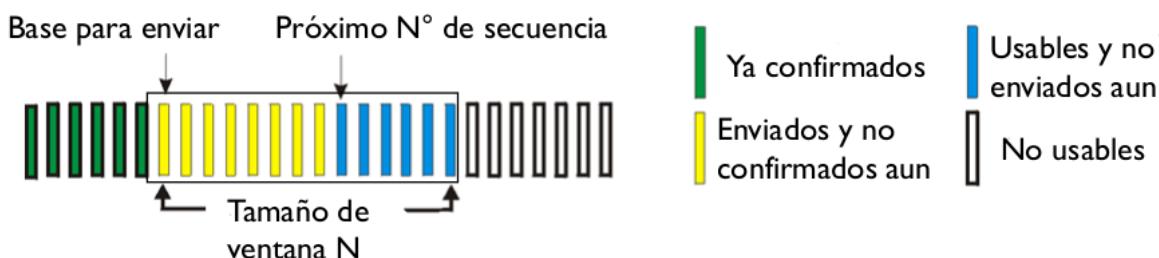
¿Cómo representamos cuántos paquetes seguidos SIN confirmación puede enviar el emisor?

A través de **intervalos de números de secuencia** en el **espacio de números de secuencia**.

ejemplo: con el ejemplo de rafagas de arriba, supongamos que enviamos 15 rafagas entonces el **espacio de números de secuencia** sería = {0,1,...,149} y los **intervalos** serían de [0,...,9][10,...,19], etc. Estos **intervalos** reciben el nombre de **ventana corrediza**.

Ventana emisora: tramas(paquetes) enviadas sin ack positivo o tramas listas para ser enviadas

- El tamaño de la ventana emisora no puede superar MAX\_SEQ cuando hay MAX\_SEQ + 1 números de secuencia.
  - La “ventana” permite hasta  $N$  paquetes consecutivos sin confirmar
  - **ventana emisora** = tramas enviadas sin ack positivo o tramas listas para ser enviadas.



- $timeout(n)$ : retransmite paquete  $n$  y todos los paquetes de mayor N° de secuencia en la ventana.

## Problema Retroceso N:

- Uso ineficiente del canal cuando se pierden paquetes

Protocolo de Repetición Selectiva: soluciona el problema de retroceso N y algunas mejoras más

- Cuando se pierde un paquete de una rafaga de paquetes en lugar de descartar los que llegaron correctamente, los guardamos en un buffer y esperamos a que llegue el paquete perdido(reenviado) para entregarlos en orden a la capa de aplicación, existe un límite para la cantidad de paquetes que podemos almacenar en un buffer del receptor esperando el perdido.
- Confirmamos individualmente todos los paquetes, no hay ack acumulativo.
- tiene un ack Negativo(NAK): es decir, que cuando detecta que llegaron varios paquetes “mayores” a digamos N, genera un NAK para que el emisor reenvíe el paquete N.

Para ser más rápido, ya no tiene que esperar a que expire el temporizador del paquete u otros métodos.

### Ventana del emisor:

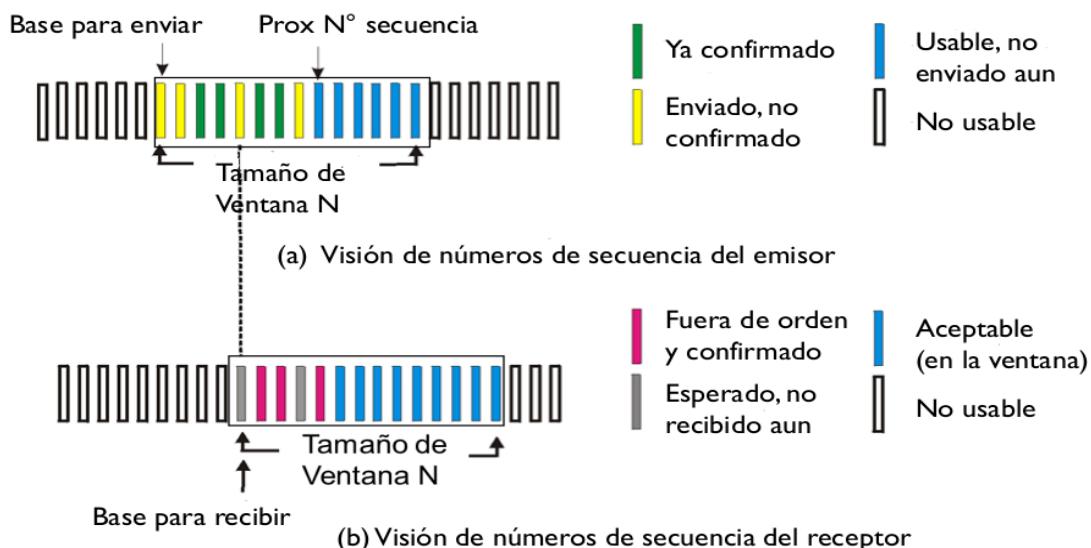
- Paquetes que contiene:
  - enviados y no confirmados
  - enviados y confirmados pero alguno enviado antes todavía no se confirmó
  - listos para enviar

### Ventana del receptor:

- Tipos de paquetes:
  - paquetes esperados y no recibidos
  - paquetes recibidos fuera de orden
  - paquetes aceptables en la ventana que aún no han llegado

Entonces, se mantiene en buffer un paquete aceptado por la ventana receptora hasta que todos los que le preceden hayan sido entregados a la capa de aplicación.

## Repetición Selectiva: ventanas del emisor y del receptor



### Procedimiento:

#### Emisor

##### Datos vienen de arriba:

- Si el próximo N° secuencia a enviar de la ventana está disponible, almacenar y enviar paquete

##### timeout( $n$ ):

- Reenviar paquete  $n$ , reiniciar timer

##### ACK( $n$ ) en [sendbase, sendbase+N]:

- marcar paquete  $n$  como recibido
- Si  $n$  es paquete más pequeño no confirmado, **avanzar base de ventana** al siguiente N° secuencia no confirmado.

#### Receptor

##### pkt $n$ en [base rcv, base rcv +N-1]

- Enviar ACK( $n$ )
- Fuera de orden: almacenarlo
- En orden: entregar (también entregar paquetes en bufer en orden), avanzar ventana al siguiente paquete que no ha sido recibido aun.

##### pkt $n$ en [base rcv-N, base rcv-1]

- Enviar ACK( $n$ )

##### Sino:

- ignorar

**Súposiciones:**  $N$  es tamaño de ventana

### Regla para el tamaño de la ventana receptora:

- Tamaño de ventana receptora =  $(\text{MAX\_SEQ} + 1)/2$ .
- Con tamaños mayores de ventana receptora no funciona.

### **¿Cómo transmitir datos entre dos máquinas y en ambas direcciones eficientemente?**

Solución: llevar a caballito (piggybacking).

- Cuando llega un segmento S con datos, el receptor se aguanta y espera hasta que la capa de aplicación le pasa el siguiente paquete P.
- La confirmación de recepción de S se anexa a P en un segmento de salida (usando el campo ack en el encabezado del segmento de salida).

### **4. Control de flujo cuando la cantidad de datos que quiere recibir y procesar el receptor varía.**

Cuando cambia el patrón de tráfico de la red; se abren y cierran varias conexiones en el receptor. El receptor y el emisor deben ajustar dinámicamente sus alojamientos de búferes, esto significa ventanas de tamaños variables. Ahora el emisor no sabe cuántos datos puede mandar en un momento dado, pero sí sabe cuántos datos le gustaría mandar.

### **¿Qué reglas cumpliría un protocolo entonces?**

El host emisor solicita espacio en búfer en el otro extremo. Para estar seguro de no enviar de más y sobrecargar al receptor, y porque sabe cuánto necesita.

### **¿Qué pasa con el receptor al recibir ese pedido?**

Sabe cuál es su situación y cuánto espacio puede otorgar, y aquí el receptor reserva una cierta cantidad de búferes al emisor.

Comunicación entre host emisor y host receptor cuando el host emisor solicita espacio en búfer en el otro extremo:

1. Inicialmente el emisor solicita una cierta cantidad de búferes, con base en sus necesidades percibidas.
2. El receptor otorga entonces tantos búferes como puede.
3. El receptor, sabiendo su capacidad de manejo de búferes podría indicar al emisor "te he reservado X búferes".
4. ¿Cómo hace el receptor con las confirmaciones de recepción? El receptor puede incorporar tanto las ack como las reservas de búfer en el mismo segmento.
5. Aclaración: TCP no lo hace, tiene un buffer único, solo otorga espacio.
6. El emisor lleva la cuenta de su asignación de búferes con el receptor.

Información de reserva de búferes viaja en segmento que no contiene datos y ese segmento se pierde. Esto termina ocasionando deadlock.

¿Cómo evitar esta situación de deadlock? Cada host puede enviar periódicamente un segmento de control con el ack y estado de búferes de cada conexión. Así el estancamiento se romperá tarde o temprano.

### **5. Control de flujo en TCP:**

No se requiere que los emisores envíen datos tan pronto como llegan de la aplicación, que los receptores envíen confirmaciones de recepción tan pronto como sea posible, y que los receptores entreguen datos a la aplicación apenas los reciben, esta libertad puede explotarse para mejorar el desempeño.

Campo Tamaño de ventana en el encabezado TCP:

- N° de bytes que pueden enviarse comenzando por el byte cuya recepción se ha confirmado.
- 0: indica que se han recibido los bytes hasta n° de confirmación de recepción – 1, inclusive, pero el receptor quisiera no recibir más datos por el momento.
- El permiso para enviar puede otorgarse enviando un segmento con el mismo n° de confirmación de recepción y un campo tamaño de ventana distinto de 0

¿Qué se hace si la ventana anunciada por el receptor es de 0?

El emisor debe detenerse hasta que el proceso de aplicación del host receptor retire algunos datos del búfer y en cuyo momento el TCP puede anunciar una ventana más grande.

En TCP los hosts en cada lado de una conexión tienen un buffer de recepción circular para la conexión. Cuando la conexión TCP recibe bytes en el orden correcto y en secuencia, coloca los datos en el buffer de recepción.

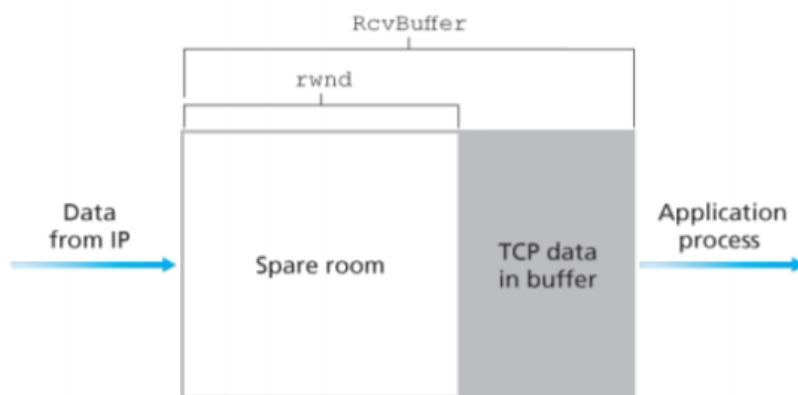


Figure 3.38 The receive window (*rwnd*) and the receive buffer (*RcvBuffer*)

El valor rwnd va en el campo tamaño de ventana.

Cálculo de tamaño de ventana:

- Tamaño de ventana = RcvBuffer – [LastByteRcvd – LastByteRead]
- Inicialmente se puede mandar: Tamaño de ventana = tamaño RcvBuffer
- Propiedad a respetar en el emisor: LastByteSent – LastByteAcked ≤ tamaño de ventana  
(Cantidad de bytes enviados pero no confirmados)

### ¿Cómo manejar pérdidas de segmentos en TCP?

La primera solución es que el receptor solicite segmento/s específico/s mediante segmento especial llamado NAK. Tras recibir segmento/s faltante/s, el receptor puede enviar una confirmación de recepción de todos los datos que tiene en búfer. Cuando el receptor nota una brecha entre el número de secuencia esperado y el número de secuencia del paquete recibido, el receptor envía un NAK en un campo de opciones.

La segunda solución es que (acks selectivos) el receptor le diga al emisor que piezas recibió. El emisor puede así reenviar los datos no confirmados que ya envió. Se usan dos campos de opciones:

- Sack permitted option: se envía en segmento SYN para indicar que se usarán acks selectivos.
- Sack option: Con lista de rangos de números de secuencia recibidos.

## Control de Congestión:

Si un emisor manda a un receptor más información que la capacidad de carga de la subred:

- la subred se congestionará pues será incapaz de entregar los segmentos a la velocidad con que llegan.
- Aca vemos directamente los algoritmos de TCP para el control de congestión
- TCP controla la congestión haciendo que los hosts reduzcan la tasa de datos
- Cualquier expiración de temporizador para TCP implica congestión en la red

## Para llevar la cuenta de cuántos datos un host puede enviar por la red:

- TCP maneja una ventana para la congestión (VC) - cuyo tamaño es el número de bytes que el emisor puede tener en la red en todo momento.

## **ACKs duplicados:**

El emisor recibe ack por cada paquete entregado con éxito, entonces para un paquete de número de secuencia N si recibe 3 ack positivos con números de secuencia > N va a reenviar el paquete

## **Para calcular un tamaño para la ventana de congestión (VC):**

- arranque lento
- TCP Tahoe
- TCP Reno

## **Arranque lento:**

- probar con un mínimo de datos e ir duplicando gradualmente hasta que no se pueda más
- la ventana crece hasta el tamaño del espacio de números de secuencia o hasta que se detecta que se perdió un paquete, en cuyo caso se reduce a la mitad de su tamaño

## **Deficiencias:**

- Recortar la ventana de congestión a la mitad porque hubo una expiración de temporizador y quedarse ahí, puede ser demasiado
- con la retransmisión disparada por expiración de temporizador el tiempo de espera puede ser relativamente grande

## **TCP Tahoe:**

- Usa un umbral además de las ventanas de recepción y congestión
- Al ocurrir una expiración del temporizador o detectarse 3 acks duplicados, se fija 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 arranque lento para determinar lo que puede manejar la red, excepto que el crecimiento exponencial termina al alcanzar el umbral.
- A partir del punto en el que se alcanza el umbral las transmisiones exitosas aumentan linealmente la ventana de congestión (en un segmento máximo por ráfaga).
- Recomenzar con una ventana de congestión de un paquete toma un RTT (para todos los datos previamente transmitidos que dejen la red y para ser confirmados, incluyendo el paquete retransmitido).
- Si no ocurren más expiraciones de temporizador/3 acks duplicados, la ventana de congestión continuará creciendo hasta el tamaño de la ventana del receptor.
  - En ese punto dejará de crecer y permanecerá constante mientras no ocurran más expiraciones de temporizador y la ventana del receptor no cambie de tamaño.

### **Crítica a Tahoe:**

- Comenzar con arranque lento cada vez que se pierde un paquete puede ser demasiado.

### **TCP Reno:**

Evitar arranque lento (excepto cuando la conexión es comenzada) cuando expira el temporizador de reenvíos.

### **Funcionamiento:**

1. Luego de iniciada la conexión se comienza con arranque lento.
2. A continuación la ventana de congestión crece linealmente hasta que se detecta una pérdida de paquete.
  - Se cuentan acks duplicados
  - Se considera pérdida de paquete 3 acks duplicados
3. El paquete perdido es re-transmitido usando **retransmisión rápida**
4. Retransmisión rápida:
  - Se manda un paquete por cada ack duplicado recibido.
  - Un RTT luego de la retransmisión rápida, el paquete perdido es confirmado.
  - La recuperación rápida termina con esa confirmación de recepción.
5. Luego de recibir el nuevo ack:
  - la ventana de congestión de una conexión se achica a la mitad de lo que era cuando se encontraron 3 duplicados (decremento multiplicativo).
  - El conteo de ack duplicados se pone en 0.
6. Luego la ventana de congestión va incrementando de a un segmento por cada RTT (crecimiento aditivo).
7. Este comportamiento continúa indefinidamente.

### **Problemas de tener segmentos duplicados retrasados y su resolución:**

¿Pueden viajar segmentos duplicados de un host emisor a uno receptor?(OBVIO XD)

- Sí, por ejemplo, cuando se pierde un ack y un segmento se retransmite.
- También cuando por congestión un segmento se demora, expira su temporizador y el segmento se retransmite.

Exigencia: No se pueden entregar segmentos duplicados a la capa de aplicación.

Consecuencia: Por lo tanto es necesario saber si un segmento que llega a un host es duplicado o no.

### Soluciones inefficientes:

- Comparar los segmentos bit a bit → sumado a los costos de almacenar los segmentos que llegaron previamente
- Enumerar los segmentos con números de secuencia → funcionaría bien si tuviéramos números de secuencia arbitrarios

No pueden ser arbitrarios porque queremos que los segmentos tengan longitud máxima → el espacio de números de secuencia es finito.

Números de secuencia alcanzan el máximo y vuelven a reiniciarse y aumentan hasta alcanzar el máximo de vuelta → espacio de secuencia no es suficientemente grande → duplicado retrasado permanece demasiado tiempo en la red.

Sucede la siguiente situación: un segmento S con n° de secuencia x queda demorado debido a que la red está congestionada.

- El temporizador de retransmisiones asociado a S expira y se retransmite S.
- El protocolo de enrutamiento cambia las rutas y la retransmisión de S llega rápido a destino.
- Pero aun quedó en la red un duplicado retrasado de S (de n° de secuencia x).
- Ese duplicado retrasado de S más adelante llega a destino generando problemas.

## Este tipo de problemas es tan serio que debe ser evitado.

Para evitarlo:

- Asegurar que ningún paquete viva más allá de T sec
- Esto se refiere a paquetes de datos, retransmisiones de ellos y a confirmaciones de recepción.
- Eliminar los paquetes viejos que dan vueltas por la red.

Duplicados retrasados en una sola conexión

- El origen etiqueta los segmentos con número de secuencia que no va a ser reutilizados en T sec.
- El espacio de secuencia debe ser lo suficientemente grande para garantizar esto → logra que los segmentos viejos con el mismo número de secuencia hayan desaparecido después de que se regrese al principio de los números de secuencia.

Evitar que duplicado retrasado que pasa de una conexión a otra genere problemas:

- Una conexión debe tener un número de secuencia inicial de secuencia con el que comienza a operar.
- Elegir como un numero de secuencia inicial un numero de secuencia que haga imposible o improbable que el duplicado retrasado numero de secuencia x genere problemas. Además se mantiene dentro de una conexión que el origen etiqueta los segmentos número de secuencia que no van a utilizarse dentro de los T sec.

Implementaciones:

1. al crear una nueva conexión cada extremo genera un número de secuencia de 32 bits aleatorio que pasa a ser el número inicial para los datos enviados. → Alguna implementación de TCP usa esta solución. → La probabilidad de que un paquete duplicado retrasado genere problemas en una conexión es baja debido a la elección aleatoria del número inicial de secuencia de la siguiente conexión.
2. vincular número de secuencia de algún modo al tiempo y para medir el tiempo usar un reloj. Cada host tiene un reloj de hora del día. Los relojes de los hosts no necesitan ser sincronizados. Cada reloj es un contador binario que se incrementa a sí mismo en intervalos uniformes. El reloj continúa funcionando ante la caída de un host. → Cuando se establece una conexión los k bits de orden de mayor del reloj = número inicial de secuencia.

Si el reloj se mueve más rápido que la asignación de números de secuencia a los paquetes que se envían → el número inicial de secuencia de una nueva conexión va a ser mayor al número de secuencia de cualquier duplicado retrasado de una conexión previa.

## Establecimiento y liberación de conexiones:

Como al establecer una conexión se usan segmentos, una conexión debería tener un N° inicial de secuencia con el que comienza a operar.

Idea: vincular N° inicial de secuencia de algún modo al tiempo, y para medir el tiempo, usar un reloj.

## Implementación de la idea (de Tomlinson):

- Cada host tiene un reloj de hora del día.
  - Los relojes de los hosts no necesitan ser sincronizados, se supone que cada reloj es un contador binario que se incrementa a sí mismo en intervalos uniformes.
  - El reloj continúa operando aún ante la caída del host
  - Cuando se establece una conexión los k bits de orden mayor del reloj = número inicial de secuencia.

Para lograr que al regresar al principio de los n° de secuencia, los segmentos viejos con el mismo n° de secuencia hayan desaparecido hace mucho tiempo el espacio de secuencia debe ser lo suficientemente grande.

Problema: Cuando un host se cae, al reactivarse sus ET no saben dónde estaban en el espacio de secuencia.

- Este es un problema porque para el siguiente segmento a enviar no se sabe qué números de secuencia generar.
  - si se genera mal, entonces el nuevo segmento podría tener el mismo número de secuencia que otro segmento distinto circulando por la red.
- Solución: requerir que las ET estén inactivas durante T segundos tras una recuperación para permitir que todos los segmentos viejos expiren.

### Establecimiento de Conexión: Acuerdo a tres bandas

Para establecer conexión el host de origen envía un segmento CONNECTION REQUEST(N,P) al destino y espera una respuesta CONNECTION ACCEPTED.

Donde N es el número de secuencia y P el puerto.

¿Qué pasa si hay un duplicado de **CONNECTION REQUEST**?

No tenemos forma de saber si un segmento **CR** contenido un n° de secuencia inicial es un duplicado de una conexión reciente o una conexión nueva.

¿Cómo lo solucionamos ?

Host 1 (emisor) sabe las conexiones que creó, entonces si recibe un CONNECTION ACCEPTED duplicado(mismo número de secuencia N) va a rechazar la conexión. Luego host 2 (receptor) se da cuenta que fue engañado por un duplicado y abandona la conexión.

### Establecimiento de una conexión TCP:

- El n° de secuencia inicial de una conexión no es 0
- Se usa un esquema basado en reloj con un pulso de reloj cada  $4 \mu$  sec.
- Al caerse un host, no podrá reiniciarse durante el tiempo máximo de paquete 120 seg, para asegurar que no haya paquetes de conexiones previas vagando por Internet.

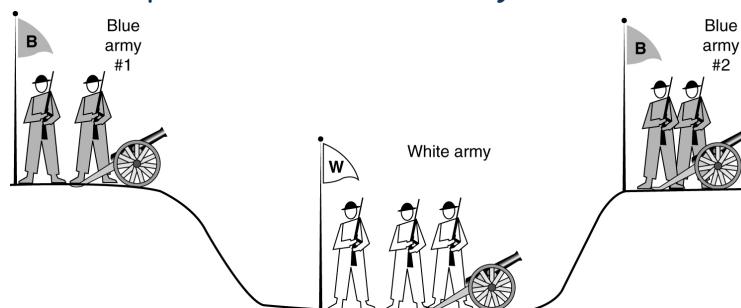
### Campos del encabezado TCP para el establecimiento de conexiones:

SYN se usa para establecer conexiones:

Solicitud de conexión: SYN = 1 y ACK = 0.

Respuesta de conexión: SYN = 1 y ACK = 1.

### Liberación de Conexiones: problemas de los dos ejércitos



Si los dos ejércitos azules atacan simultáneamente van a ganar. Por eso quieren sincronizar su ataque. Es muy difícil que ocurra esto porque se necesita saber si el ejército recibió el mensaje...

No existe un protocolo que resuelva el problema de los dos ejércitos.

Para el caso de liberación de conexión “atacar” equivale a “desconectar”. Si ninguna de las partes está preparada para desconectarse hasta estar convencida que la otra está preparada para desconectarse también, nunca ocurrirá la desconexión.

**Idea 2:** Vamos a permitir que cada parte decida cuando la conexión está terminada.

La liberación de conexión en un host significa que la ET remueve la información sobre la conexión de su tabla de conexiones abiertas y avisa de alguna manera al dueño de la conexión.

Se estudian en 4 casos:

1. Caso Normal → Host 1 envía un disconnection request e inicia temporizador para el caso que no llegue DR a host 2 → llega DR a host 2 → host 2 emite un DR e inicia un temporizador → host 1 recibe DR → envía ACK a host 2 y libera conexión → host 2 recibe el ack y libera la conexión
2. Si se pierde el último ACK → expira el temporizador y la conexión se libera
3. Si se pierde el segundo DR → host 1 no recibe la respuesta esperada → exipran los temporizadores → comienza todo de vuelta
4. Respuesta perdida y DRs subsiguientes perdidos → Tras N intentos el emisor se da por vencido y libera la conexión → Expira el temporizador del receptor y se desconecta.

Este protocolo falla si se pierde el DR inicial y N retransmisiones → El emisor se da por vencido y libera la conexión → el otro lado no se entera de los intentos de desconexión y quedará siempre activo. Conexión abierta a medias.

Soluciones de conexiones abiertas a medias:

1. Obligar al emisor seguir insistiendo hasta recibir una respuesta → si se permite que expire el temporizador del otro lado → el emisor insistirá eternamente, nunca va a aparecer una respuesta.
2. Si no ha llegado ningún segmento a host 2 durante una cierta cantidad de segundos → host 2 libera la conexión. → host 1 detecta falta de actividad y se desconecta. Resuelve el caso que la red se rompió y los hosts ya no pueden conectarse.

Implementación de la solución 2:

- Es necesario que cada ET tenga un temporizador que se detenga y se reinicie con cada envío de un segmento.
- No se puede garantizar absolutamente que cuando se libera la conexión, no ocurra pérdida de datos → se puede limitar esta situación.

Liberación simétrica:

- Cada parte se cierra por separado, independientemente de las otras.
- Una de las partes emite un DISCONNECT porque ya no tiene más datos para enviar y aun está dispuesta a recibir datos de la otra parte.
- Una conexión se libera cuando ambas partes emitieron un DISCONNECT

Es ideal cuando cada proceso tiene una cantidad fija de datos por enviar y sabe con certidumbre cuando los ha enviado. En otras situaciones la determinación de si se ha efectuado o no todo el trabajo o si debe determinarse o la conexión no es tan obvia.

### Liberación de una conexión TCP:

Campo usado por TCP para liberación de conexiones. FIN especifica que el emisor no tiene más datos que transmitir (FIN prendido en 1). Tras cerrar una conexión, un proceso puede continuar recibiendo datos indefinidamente. Cuando ambos sentidos de la conexión se han apagado, se libera la conexión.

Para liberar una conexión cualquiera de las partes puede enviar un segmento TCP con el bit FIN establecido lo que significa que no tiene más datos por transmitir, pero todavía puede recibir datos del otro lado. Al confirmarse la recepción del FIN ese sentido se apaga (el receptor del ack no va a enviar más).

## ##FIJENSE SI HAY ALGO MÁS DE ESTO QUE QUIERAN AGREGAR

### Administración del temporizador del TCP:

¿Qué tan grande debe ser el intervalo de expiración del temporizador de retransmisión?

- Si se hace demasiado corto entonces ocurrirán retransmisiones innecesarias
- Si se hace demasiado largo sufrirá el desempeño por el gran retardo de retransmisión de cada paquete perdido

Idea: Ajustar constantemente el intervalo de expiración del temporizador, con base en mediciones continuas del desempeño de la red. Se soluciona con →

### Algoritmo de Jacobson(1988) usado por TCP:

Por cada conexión el TCP mantiene una variable, **RTT** (round trip time) → significa estimación actual del tiempo de ida y vuelta al destino.

Al enviarse un segmento se inicia un temporizador, para saber el tiempo que tarda el ack y para habilitar una retransmisión si se tarda demasiado.

Si llega el ack antes de expirar el temporizador: TCP mide el tiempo que tardó el ack digamos  $M$ , entonces actualiza el RTT así:

➤  $RTT = \alpha RTT + (1-\alpha) M$ ,  $\alpha$  es el peso que se le da al valor anterior Por lo común  $\alpha = \frac{1}{8}$ .

### Dado un RTT, hay que elegir una expiración adecuada del temporizador de retransmisión.

**Solución 1:** En las implementaciones iniciales.

Expiración del temporizador =  $2 \times RTT$ .

**Solución 2:** hacer que el valor de timeout sea sensible tanto a la variación de RTT como a la varianza de la función de densidad de probabilidad del tiempo de llegada de los ack.

- Se mantiene una variable amortiguada  $D$  la desviación media
- Al llegar un ack se calcula  $|RTT - M|$
- Se mantiene en  $D$  mediante  $D = \beta D + (1 - \beta) |RTT - M|$ , donde  $\beta$  típicamente es  $\frac{3}{4}$ .
- $D$  es una aproximación bastante cercana a la desviación estándar.

La mayoría de las implementaciones TCP usan ahora este algoritmo y establecen expiración del temporizador =  $RTT + 4 \times D$ .

Para estimar el temporizador de retransmisiones en ese caso → **algoritmo de Karn (lo usan la mayoría de las implementaciones TCP)**

- No actualizar el RT de ninguno de los segmentos retransmitidos
- Cuando ocurre un timeout se duplica la expiración del temporizador.
- Tan pronto se recibe un ack de segmento no retransmitido, el RTT estimado es actualizado y la expiración del temporizador se computa nuevamente usando la fórmula anterior.

### UDP (protocolo de datagramas de usuario):

Es no orientado a la conexión.

- segmentos = encabezado de 8 B + carga útil.

- 2 puertos de 16 b.
- El campo **longitud UDP** incluye el encabezado de 8 bytes y los datos.

Especialmente útil en las situaciones cliente servidor.

El cliente envía una solicitud corta al servidor y espera una respuesta corta.

Si se pierde la solicitud o la respuesta, el cliente puede probar nuevamente.

## Capa de Red

### La Capa de Red – Generalidades:

El propósito de la capa de red es llevar paquetes de un host de origen a uno de destino siguiendo una ruta conveniente.

### **Asuntos de los que se encarga la capa de red:**

- Almacenamiento y reenvío
- Enrutamiento
- Control de congestión
- Conectar redes de distintas tecnologías (Interredes)
- Fragmentación

Aclaración de Duran (y lo escribió Sofi. Sofi no me pegues :c):

El control de congestión usando solo la capa de transporte no es del todo eficiente, lo más que puede hacer el emisor es asumir que perdió un paquete. Pero si hay una capa de red, el host emisor se podría enterar antes de la congestión, para que se reduzca la tasa de transferencia.

### Cómo es el hardware subyacente a la CR:

Hardware subyacente de la capa de red es:

- Subred: formada por enrutadores interconectados
- Hosts o LANs conectadas a subred
- Varias subredes de distinta tecnología unidas entre sí usando puertas de enlace

No puede pasar un paquete tal cual de una red a otra. ¿Por qué?

- Formatos de paquetes y tamaños máximos difieren de una red a otra.

### Enfoques usados para envío de paquetes entre dos hosts:

- Para entender dos maneras existentes de hacer el enrutamiento en las subredes.

Enfoques para mandar un conjunto de paquetes desde un host de origen a un host de destino:

- Usar una ruta fija para mandar todos los paquetes (servicio orientado a la conexión).
- La ruta puede cambiar, por lo que distintos paquetes pueden seguir distintos caminos (servicio no orientado a la conexión).

### Servicio no orientado a la conexión:

- Alentado por la comunidad de internet.
- Los paquetes se enrutan de manera independiente.
  - La ruta a usar entre los hosts va a cambiar cada cierto tiempo.
  - Cada paquete debe llevar una dirección de destino completa.
- Nomenclatura usada:
  - Paquetes = datagramas

- Subredes = subredes de datagramas

### ¿Cómo diseñar la tabla de un enrutador?

Suponemos que existe un procedimiento que dada la dirección del host de destino me retorna la dirección del enrutador de destino (al cual está conectado host de destino) el cual sabe cómo entregar paquete al host de destino (por más que el host de destino esté en una LAN).

Entonces, para armar la tabla de enrutamiento solo necesita entradas para los enrutadores de la subred. La entrada de tabla de enrutador está formada por filas: <enrutador de destino, línea de salida> (La línea de salida es la dirección de un enrutador).

Dirección de red: sirve para identificar una red.

Número de máquina: sirve para identificar una máquina dentro de la red.

### ¿Qué se hace cuando llega un paquete a un enrutador?

1. Se lo almacena y se comprueba que llegó bien.
2. Se determina el enrutador de destino asociado al host de destino.
3. Se usa la fila de ese enrutador de destino para reenviar el paquete por la línea de salida de esa fila.

### Servicio orientado a la conexión:

- Alentado por las compañías telefónicas (p.ej. ATM)
- Todos los paquetes se mandan por la misma ruta.
- Trabajo a realizar antes de mandar paquetes:
  - Hay que configurar una ruta del host de origen al de destino (crear una conexión).
  - Circuito virtual (CV) = conexión.
- Cada paquete lleva un identificador que indica a qué CV pertenece.
- Cuando no se necesita enviar más paquetes se libera la conexión. Al hacer eso también se termina el CV.

Se elige una ruta de la máquina de origen a la de destino:

- Esta ruta se almacena en tablas dentro de los enrutadores.

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

# Enrutamiento Jerárquico

- Para entender cómo organizar tablas de enrutamiento en redes muy grandes.
- Cuando crece mucho el tamaño de las subredes, también lo hacen las tablas de enrutamiento.
- Consecuencias de tener tablas de enrutamiento grandes:
  - Estas tablas consumen memoria del enrutador, necesitan más tiempo de CPU para examinarlas.

¿Cómo hacer para que las tablas de enrutamiento no crezcan demasiado cuando crece mucho el tamaño de la subred?

- Los enrutadores se dividen en regiones.
- Un enrutador sabe cómo enrutar paquetes a destinos en su región.
- También sabe cómo enrutar a otras regiones.
- Pero no sabe nada de la estructura interna de las regiones en las que no está.

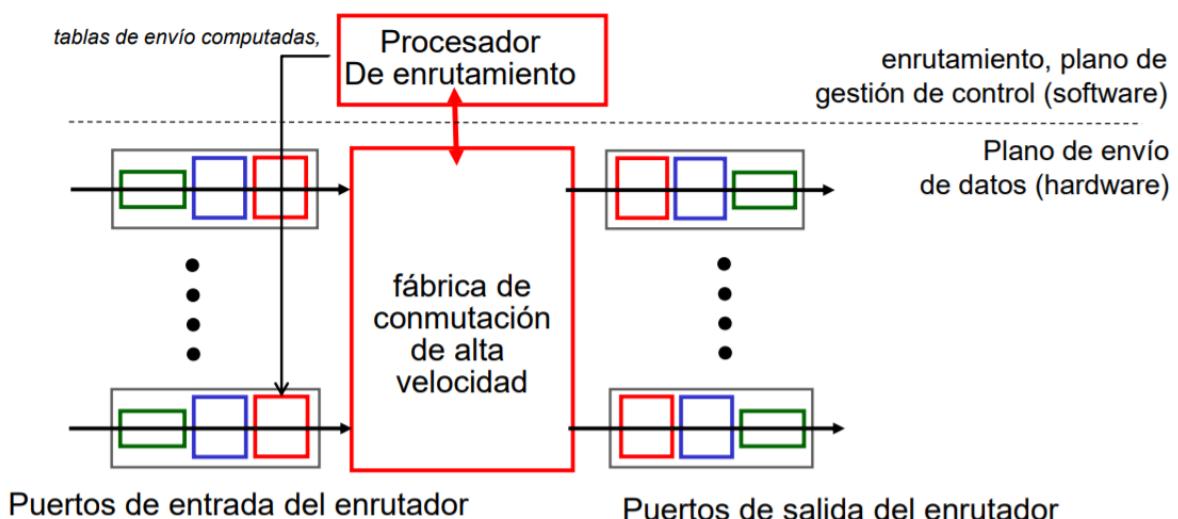
Precio a pagar con enrutamiento jerárquico: una longitud de ruta mayor (no se puede aspirar a encontrar la mejor ruta).

En las redes enormes, una jerarquía de dos niveles es insuficiente; tenemos que agrupar las regiones en clusters, los clusters en zonas, las zonas en grupos, etc.

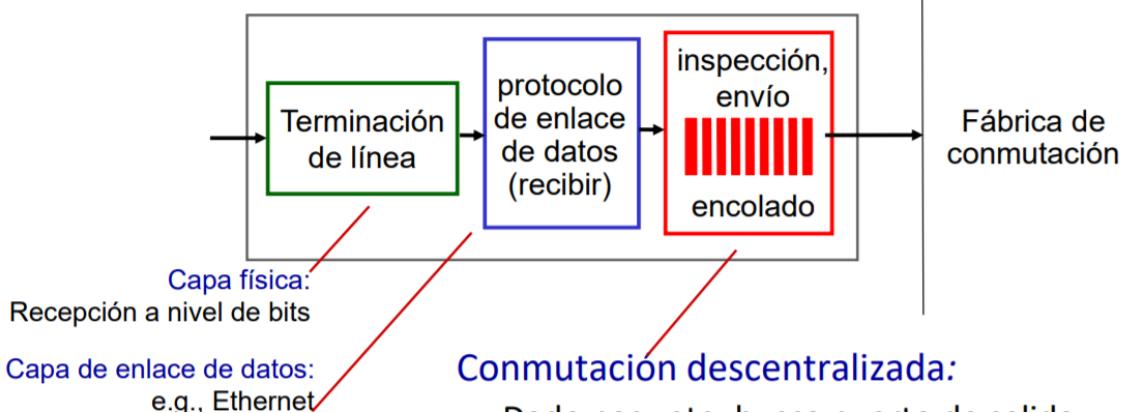
## Cómo es la arquitectura de un enrutador:

Funciones clave de un enrutador:

- Ejecutar algoritmos de enrutamiento/protocolos (RIP, OSPF, BGP)
- Enviar paquetes de enlaces de ingreso a enlaces de salida

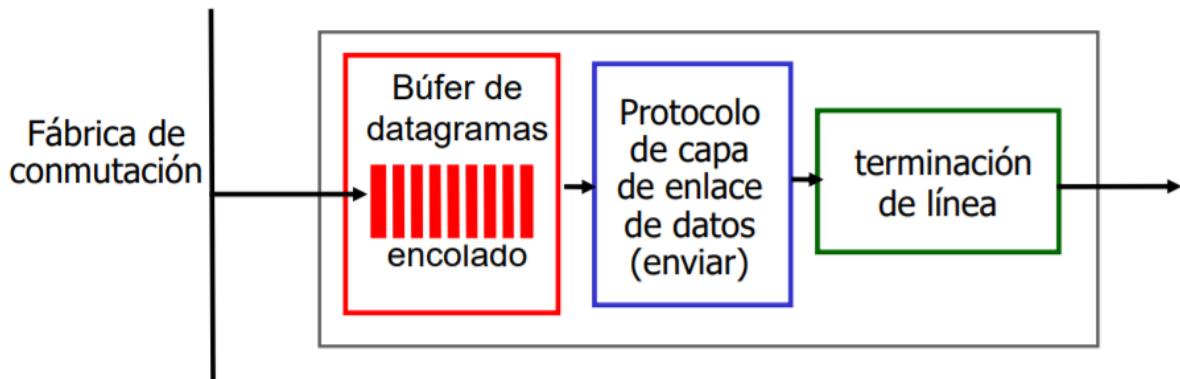


# Puertos de Entrada



Comutación descentralizada:

- Dado un paquete, éste busca un puerto de salida usando la tabla del enrutador.
- meta: procesamiento completo del input a la “velocidad de la línea”.
- encolado: si los paquetes arriban más rápido que la tasa de envío en la fábrica de conmutación.



- Encolado: requerido cuando los paquetes llegan de la fábrica de conmutación más rápido que la tasa de transmisión. Los paquetes pueden perderse debido a congestión, carencia de búferes.
- Disciplina de planificación: elige entre los paquetes encolados para transmisión.

Conceptos básicos para el algoritmos de enrutamiento:

Tenemos que evitar los siguientes efectos indeseados:

- Algunos enrutadores pueden quedar inactivos.
- Los caminos pueden ser innecesariamente largos.
- Se pueden sobrecargar algunas de las líneas de comunicación y los enrutadores asociados a ellas.

La causa es que **la capa de red elige mal las rutas para enviar paquetes.**

¿Cómo escoger bien las rutas para enviar paquetes?

Usar *algoritmos de enrutamiento* eficientes.

- Un algoritmo de enrutamiento se ejecuta en los enrutadores de la subred;
- es responsable de llenar y actualizar las tablas de enrutamiento.

Algoritmo de enrutamiento de caminos más cortos:

Algoritmo de cálculo de la ruta más corta entre dos nodos: Dijkstra (1959).

Procedimiento para calcular tablas de reenvío en redes de datagramas usando algoritmo de Dijkstra.

1. Construir grafo de la subred con costos.
2. Ingresar grafo de la subred con costos en los enrutadores.
3. En cada enrutador construir tabla de enrutamiento; para eso:
  - a. Ejecutar algoritmo de Dijkstra en el enrutador
  - b. A partir de un árbol de caminos más cortos con la raíz en el enrutador obtenido, generar la tabla de reenvío del enrutador.

Al aplicar Dijkstra la raíz del árbol va ser el enrutador. Al enrutar Dijkstra va a dar un árbol diferente, porque tiene diferentes raíces en cada enrutador.

## Inundación usando registro de paquetes difundidos

Algoritmos de enrutamiento: buscan determinar las rutas a seguir para enviar paquetes de un origen a un destino. (El conjunto de rutas puede no ser unitarios)

- Ese conjunto de rutas se puede definir por medio de un conjunto de reglas a respetar.
- Los algoritmos de enrutamiento se preocupan de actualizar las tablas de reenvío de mensajes en los enrutadores.

Idea de inundación: para enviar un paquete de un origen “u” a un destino “v” los caminos usados son aquellos que respetan las siguientes reglas:

- “u” manda el mensaje por todas las líneas de salida.
- Cada paquete llega a un enrutador distinto de “v”, y se reenvía por cada una de las líneas excepto aquella por la que llegó.

El problema de la inundación es que genera grandes cantidades de paquetes duplicados; a menos que se tomen algunas medidas para limitar el proceso.

Aclaración(desofí): cuando llega un paquete a un enrutador, éste se envía por todas las líneas menos por la que llegó.

Hace falta limitar un poco el proceso de inundación, entonces cada enrutador recuerda los paquetes difundidos previamente por él para decidir si acepta un paquete.

Refinamiento de la idea anterior:

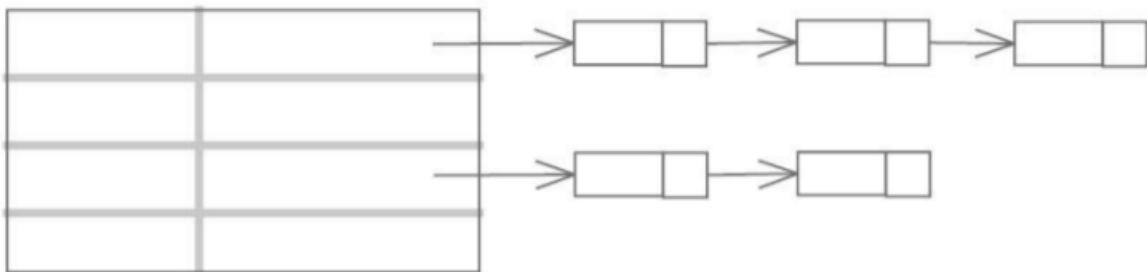
- El enrutador de origen pone un número de secuencia en cada paquete que recibe de sus hosts (así se distingue entre paquetes distintos del mismo enrutador de origen).
- Un enrutador recuerda para cada enrutador de origen los números de secuencia recibidos
- Si llega un paquete a un enrutador con par recibido antes, no se lo reenvía

Implementación: para cada enrutador usar tabla de registro de paquetes difundidos.

¿Cómo se puede evitar que las listas enlazadas crezcan sin límites?

Agregar una columna *contador* que indica el mayor número de secuencia tal que llegaron paquetes con todos los números de secuencia anteriores desde ese enrutador de origen.

## Nº enrutador Lista de Nº de de origen secuencia vistos



Aclaración:

- ¿Qué tiene que pasar para que tenga que actualizar la tabla de registros/paquetes difundidos?
  - Que llegue un paquete para re-transmitir con el número de secuencia mayor que el contador y que no esté en la lista, o que llegue un paquete para un enrutador de origen que nunca mandó nada, y me obligue a crear una fila nueva.
- ¿Por qué es una buena idea? Si
  - Porque el contador evita tener muchos nodos en las listas enlazadas, lo cual hace que sean más cortas.

## Enrutamiento de vector de distancia

Cada enrutador mantiene una tabla de enrutamiento (o de reenvío) indexada por cada enrutador en la subred.

- Cada entrada (Cada fila) comprende la línea preferida de salida hacia ese destino y una estimación del tiempo o distancia a ese destino.

A partir de su tabla de enrutamiento un enrutador E puede obtener un vector de distancia que contiene una lista de pares <destino, retardo estimado>

El retardo de un enrutador a un vecino suyo, puede medirlo con paquetes de ECO que el receptor simplemente marca con la hora y los regresa tan rápido como puede.

Aclaración: la información del selector de distancia es la información de la tabla de enrutamiento de la cual se extrae porque no se necesita información de las líneas de salida.

## Enrutamiento de Estado de Enlace

Aclaración: Los algoritmos anteriores no se adaptan al cambio de la topología de la red. En general no se adaptan al tráfico en la red.

Enrutamiento de estado de enlace (Link state routing-LSR):

- En cada enrutador usar algoritmo de Dijkstra para encontrar la ruta más corta de un enrutador a los demás enrutadores.
- La topología y retardos en las líneas se distribuyen a cada enrutador.

- Este algoritmo es valioso porque responde rápido frente a cambios en la topología de la red, y es ampliamente usado en Internet (como parte del protocolo OSPF).

¿Qué tareas debe hacer un enrutador LSR?

1. Descubrir sus vecinos
2. Medir el costo a cada uno de sus vecinos
3. Construir un paquete diciendo lo que ha aprendido
4. Enviar este paquete a todos los demás enrutadores
5. Computar el camino más corto a cada uno de los otros enrutadores

¿Cómo se puede averiguar quiénes son los vecinos de un enrutador?

- Se envía paquete Hello a cada línea punto a punto
- Se espera que el enrutador del otro extremo regrese una respuesta indicando quién es.

¿Cómo se puede hacer para que el enrutador conozca retardo a sus vecinos?

- Enviar un paquete ECHO especial a través de la línea
- Una vez que llegue al otro extremo, éste debe regresar inmediatamente
- Uso de temporizadores para medir el tiempo.
- Método: Se mide el tiempo de ida y vuelta y se divide por 2. (Retardo por vecino)

Cada enrutador construye un paquete de estado de enlace (LSP)

- ¿Qué datos poner en el LSP?
  - Identidad del emisor
  - Número de secuencia
  - Edad
  - Lista de <vecino, retardo al vecino>
- ¿Cuándo se pueden construir los LSP?
  - Construirlos a intervalos regulares.
  - Construirlos cuando ocurra un evento significativo, como la caída o la reactivación de la línea o de un vecino, o el cambio apreciable de sus propiedades.

Distribución confiable de los LSP.

- usar inundación para distribuir los LSP.
- se lleva registro de los paquetes difundidos.
  - Cada paquete contiene un número de secuencia que se incrementa con cada paquete nuevo enviado (desde su enrutador de origen).
  - ¿Cómo es este registro?
  - Los enrutadores llevan el registro de todos los pares <enrutador de origen, secuencia> que ven.

Cuando llega un LSP a un enrutador, ¿Qué se hace con él?

- Ayuda: comparar el valor de su número de secuencia con el que figura en la tabla (de paquetes difundidos) para el enrutador que lo mandó.
- Si es nuevo (número de secuencia mayor que los anteriores),
  - se reenvía a través de todas las líneas, excepto aquella por la que llegó.
- Si es un duplicado (número de secuencia mayor visto, pero repetido),
  - se descarta.
- Si llega un paquete con número de secuencia menor que el mayor visto hasta el momento,
  - se rechaza como obsoleto debido a que el enrutador tiene datos más recientes.

¿Cuándo se puede construir la tabla de enrutamiento de un enrutador?

- Una vez que el enrutador ha acumulado un grupo completo de paquetes de estado del enlace

Construir el grafo de la subred completa.

- Cada enlace se representa dos veces, una para cada dirección.
- Los dos valores pueden promediarse o usarse por separado.

Se ejecuta el algoritmo de Dijkstra para construir la ruta más corta a todos los destinos posibles.

- Con los resultados del mismo se actualiza la tabla de enrutamiento.

## Complementos de protocolos de enrutamiento

### Inundación:

Inundación con contador de saltos: integrar un contador de saltos en el encabezado de cada paquete, que disminuye con cada salto y el paquete se descarta cuando el contador llega a 0.

¿Cómo se determina el contador de saltos?

- Lo ideal es inicializar el contador de saltos a la longitud de la ruta entre el origen y el destino.
- Si el emisor desconoce el tamaño de la ruta, puede inicializar el contador al peor caso, es decir, al diámetro total de la subred.

Inundación Selectiva: una idea para la inundación bastante práctica, es la inundación selectiva:

- Los enrutadores no envían cada paquete de entrada por todas las líneas, sino sólo por aquellas que van aproximadamente en la dirección correcta.
- ¿Qué tipo de información necesita almacenar un enrutador para poder aplicar inundación selectiva?
  - Se necesita saber en qué dirección va cada línea y en qué dirección está el destino.

El algoritmo de inundación de paquetes de estado de enlace tiene algunos problemas que mencionamos a continuación.

- Si los números de secuencia vuelven a comenzar, reinará la confusión:

Tenemos que usar un número de secuencia de longitud suficiente para que el problema anterior no suceda. Por ej. de 32 bits.

- Si llega a corromperse un número de secuencia y se escribe 65540 en lugar de 4 (un error de un bit), los paquetes 5 a 65540 serán rechazados como obsoletos, dado que se piensa que el número de secuencia actual es 65540:

Como protección contra los errores en las líneas enrutador-enrutador, se confirma la recepción de todos los paquetes de estado del enlace. Es decir haría falta que antes de actualizarse el número de secuencia más grande, el router mande una confirmación de recepción al transmisor y luego espera una respuesta afirmativa o negativa del transmisor.

- En el primer caso se actualiza el número de secuencia más grande.
- En el segundo caso se descarta el paquete que se recibió por estar errado.

- Si llega a caerse un enrutador (de origen), perderá el registro de su número de secuencia. Si comienza nuevamente en 0, se rechazarán el siguiente paquete:

La información de los enrutadores sólo expira (a lo largo de la red) cuando el enrutador está caído.

¿Cuándo se puede detectar que un enrutador está caído?

- Cuando se actualicen las tablas de enrutamiento y se mandan los paquetes Hello, se puede detectar que el enrutador está caído.

¿Una vez identificado que un enrutador está caído cómo proceder?

- Se propaga la información de este hecho por toda la red.
- Se hace que la información asociada al enrutador caído expire (paquetes pendientes a enviar, número de secuencia más grande recibido, etc.).
- Así que cuando ese enrutador vuelva a la vida, puede comenzar con número de secuencia 0.
- ¿Cómo hacer para asegurar que no pueda perderse ningún paquete y sobrevivir durante un período indefinido?
 

Incluir un campo de edad en cada paquete:

  - Disminuir la edad una vez cada segundo.
  - Los enrutadores también decrementan el campo de edad durante el proceso inicial de inundación.
  - Se descarta el paquete cuya edad sea 0.

Algoritmo de inundación de paquetes de estado de enlace más eficiente:

- Una vez que un paquete de estado del enlace llega a un enrutador para ser inundado, no se encola para transmisión inmediata. En vez de ello, entra en un búfer de almacenamiento donde espera un tiempo breve.
- Si antes de transmitirlo, llega otro paquete de estado del enlace proveniente del mismo origen, se comparan sus números de secuencia.
  - Si son iguales, se descarta el duplicado.
  - Si son diferentes, se desecha el más viejo.

El buffer de paquetes para un enrutador contiene una celda por cada paquete de estado de enlace recién llegado, pero aún no procesado por completo.

Una fila de la tabla del búfer de paquetes de un enrutador contiene:

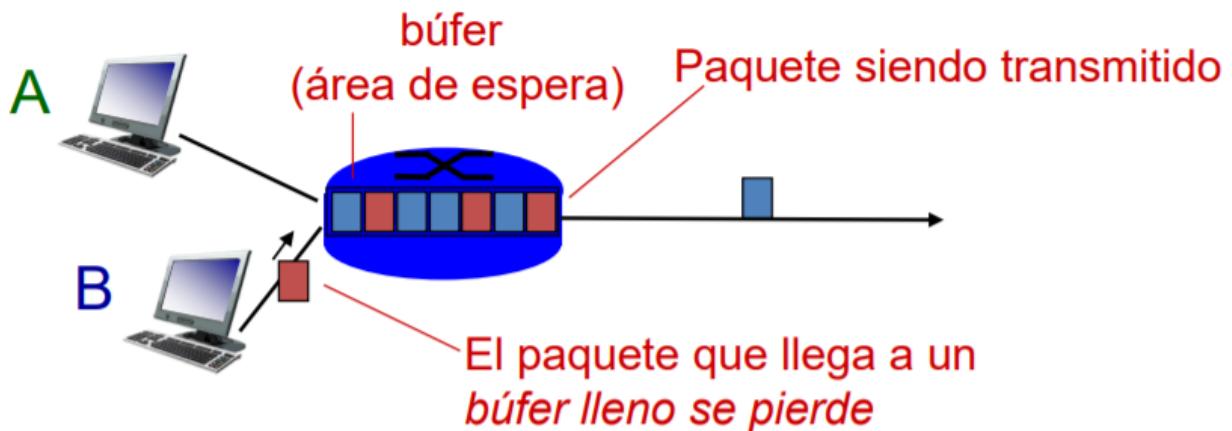
- Origen del paquete, número de secuencia, edad, datos de los estados de enlaces.
- Banderas que pueden ser:
  - Banderas de confirmación de recepción: indica a dónde tiene que enviarse la confirmación de recepción del paquete.
  - Banderas de envío: significan que el paquete debe enviarse a través de las líneas indicadas.
  - Si llega un duplicado mientras el original aún está en el búfer, los bits de las banderas tienen que cambiar.

## Control de congestión

La cola en un búfer que precede a un enlace tiene capacidad finita.

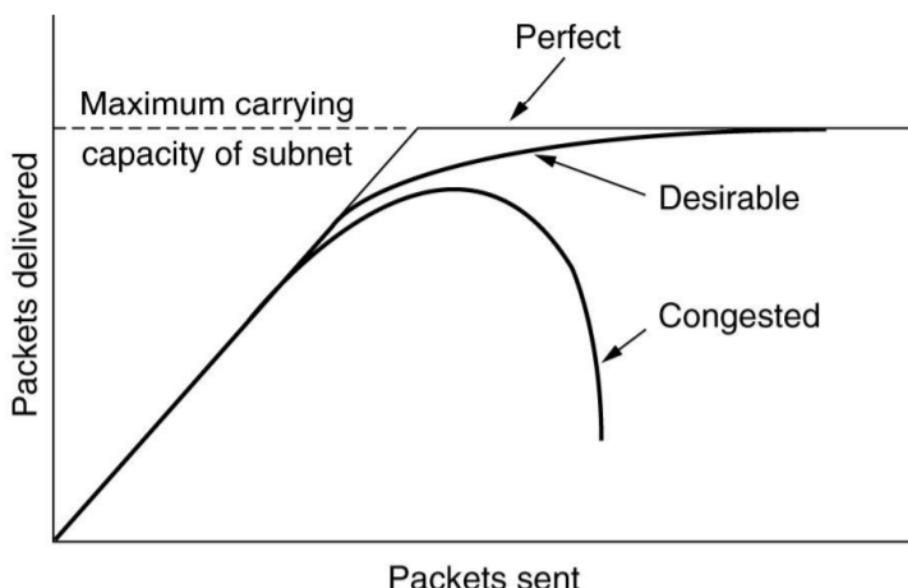
¿Qué pasa con un paquete cuando llega a una línea de salida con buffer lleno?

- El paquete que llega a un búfer lleno se pierde.
- Los paquetes perdidos deben ser retransmitidos por el enrutador previo o el host emisor.



Si comienzan a llegar muchos paquetes por algunas líneas de entrada y todas necesitan la misma línea de salida,

- Se irán acumulando los paquetes en una cola.
- Si no hay suficiente memoria para almacenar todos los paquetes, muchos de ellos se perderán.



Cuando hay demasiado tráfico, surge la congestión y el desempeño se degrada rápidamente. La meta del control de congestión es asegurar que la subred sea capaz de transportar el tráfico ofrecido.

Problemas de los algoritmos de control de congestión de TCP estudiados:

- El host de destino demora demasiado en enterarse de la congestión (solo por expiración de temporizador de retransmisiones o 3 acks duplicados).
- Los hosts solo se enteran de pérdidas de paquetes, no pueden controlar qué paquetes perder y cuáles no.

Formas de disminuir la carga en la subred:

- Regulación del tráfico: hacer que hosts responsables de la congestión se enteren más rápido (que con protocolos de TCP) de la congestión y reduzcan su tasa de transferencia.

- Desprendimiento de carga: los enrutadores descartan paquetes inteligentemente antes que se saturen búferes.

## **Cómo identificar la congestión**

¿Cómo puede hacer un enrutador para darse cuenta si tiene algún puerto de salida congestionado? Cada enrutador monitorea la demora de la cola de la línea de salida. Siempre que la demora reciente de cola de esta línea rebasa un umbral, la línea de salida entra en un estado de advertencia. Cada paquete nuevo que llega se revisa para ver si su línea de salida está en estado de advertencia. Si es así, se realiza alguna acción.

Regulación de tráfico es cuando los emisores ajustan sus transmisiones para enviar un tráfico que la red pueda soportar. La congestión se da en los enrutadores (y no en los hosts).

¿Cómo se puede enterar un host de que hay congestión? Se le avisa de la congestión.

Una vez que un enrutador tiene una línea de salida en estado de advertencia puede avisar a los hosts responsables de los paquetes que llegan a esa línea congestionada.

### **Método de paquetes reguladores:**

1. Usar paquetes reguladores si la línea de salida está en estado de advertencia, el enrutador regresa un paquete regulador (PR) al host de origen, proporcionándole el destino encontrado en el paquete.
2. Para que el paquete original no genere más PR más adelante en la ruta en el paquete original se activa un bit del encabezado y después se reenvía.
3. El PR le pide al host de origen que reduzca en un porcentaje X el tráfico enviado al destino especificado.
4. El host ignora los PR que se refieran a ese destino por un intervalo fijo.
5. Una vez que haya expirado ese tiempo, el host escucha más PR durante un intervalo.
  - a. Si llega alguno el host reduce el flujo aún más y comienza a ignorar nuevamente los PR.
  - b. Si no llega ningún PR durante el host incrementa el flujo.

En el esquema anterior cuando se satura una línea de salida de un enrutador, se pierden paquetes indiscriminadamente. Para evitar la pérdida descontrolada de paquetes. Conviene preocuparse de esto porque no todos los paquetes tienen la misma importancia y por eso es mejor controlar qué paquetes se descartan.

La primera solución es descartar paquetes inteligentemente antes de que se ocupe todo el espacio de búfer cuando hay estado de advertencia en una línea de salida. Algunos criterios para escoger qué paquetes descartar según el tipo de aplicación que se está usando.

#### **Estrategia Vino:**

- Descartar primero los paquetes más nuevos.
- P.ej. en la transferencia de archivos.

#### **Estrategia Leche:**

- Descartar primero los paquetes más viejos.
- P.ej. en multimedia.

Según la importancia de los paquetes.

- Marcar los paquetes con clases de prioridades.
- Los enrutadores primero se desprenden de paquetes de la clase más baja, luego los de la siguiente clase, etc.
- Aclaración: si todos los hosts marcan los paquetes de máxima prioridad habría problemas

La segunda solución es usar desprendimiento de carga junto con reducción de tráfico.

- La respuesta a paquetes perdidos por desprendimiento de carga es que el origen disminuya su tasa de transferencia.
- Si expira el temporizador de retransmisiones, el emisor lo toma como pérdida de paquete.
- Vemos ahora una implementación de esta solución.

#### **Algoritmo de detección temprana aleatoria (RED):**

Para detectar cuándo comenzar a descartar paquetes, los enrutadores mantienen un promedio móvil de sus longitudes de cola. Cuando este promedio de una cola C sobrepasa el umbral una pequeña fracción de los paquetes son descartados al azar. Con cada uno de esos paquetes el enrutador elige un paquete al azar de C, se descarta el paquete seleccionado, y el origen notará falta de ACK y la capa de transporte disminuirá la velocidad de transmisión. Las consecuencias de elegir paquetes al azar hace más probable que los hosts emisores más rápidos pierdan un paquete, lo noten, y reduzcan su tasa de transferencia.

## **Complementos de Control de Congestión**

- La adición de memoria puede ayudar hasta cierto punto.
- Se demostró que si los enrutadores tienen infinita memoria, la congestión empeora en lugar de mejorar,
  - ya que para cuando los paquetes llegan al principio de la cola su temporizador ha terminado (repetidamente) y se han enviado duplicados.
  - Todos estos paquetes serán re-enviados al siguiente enrutador, aumentando la carga en todo el camino hasta su destino.
- Los procesadores lentos también pueden causar congestión.
  - Si las CPUs de los enrutadores son lentas para llevar a cabo las tareas requeridas, las colas pueden alargarse, aún cuando haya un exceso de capacidad de línea.
- Las líneas de poco ancho de banda también pueden causar congestión.
  - Probablemente la cola de una línea de salida de poco ancho de banda se va a agrandar si otras líneas tienen mayor ancho de banda y están recibiendo muchos paquetes destinados a la línea de salida.

#### **Subredes de Circuitos Virtuales:**

Idea 1: Usar una técnica de control de admisión para evitar que empeoren las congestiones que ya han comenzado y que consiste en que una vez que se ha detectado la congestión (usando la técnica estudiada), no se establecen CVs nuevos hasta que ha desaparecido el problema.

Idea 2: permitir el establecimiento de nuevos CV, pero enrutando cuidadosamente los circuitos nuevos por otras rutas que no tengan problemas.

Idea 3: negociar un acuerdo entre el host y la subred cuando se establece un CV.

- Este arreglo normalmente especifica el volumen y la forma del tráfico, la calidad de servicio requerido y otros parámetros.

- Para cumplir con su parte del acuerdo, la subred por lo general reservará recursos a lo largo de la ruta cuando se establezca el circuito.
- Estos recursos pueden incluir espacio en tablas y en búfer en los enrutadores y ancho de banda en las líneas.
  - De este modo es poco probable que ocurran congestiones en los CV nuevos.

**Método de bit de advertencia.** Señalar el estado de advertencia activando un bit especial en el encabezado del paquete.

- Cuando el paquete llega a su destino, la entidad transportadora copia el bit en la siguiente confirmación de recepción que se regresa al origen.
- A continuación el origen reduce el tráfico.
- Mientras el enrutador está en estado de advertencia, continua activando el bit de advertencia, lo que significa que el origen continúa obteniendo confirmaciones de recepción con dicho bit activado.

El origen monitorea la fracción de confirmaciones de recepción con el bit activado y ajusta su tasa de transmisión de manera acorde.

- En tanto los bits de advertencia continúan fluyendo, el origen continúa disminuyendo su tasa de transmisión.

Cuando la tasa de transmisión disminuye lo suficiente, el origen incrementa su tasa de transmisión.

- Debido a que cada enrutador a lo largo de la ruta puede activar el bit de advertencia, el tráfico se incrementa solo cuando no haya enrutadores con problemas.

Una implementación de bit de advertencia usada por TCP es ECN (Explicit Congestion Notification):

- Se usa en TCP/IP.
- Se marcan 2 bits en el encabezado IP con distintos fines:
  - 00: transporte no capaz de ECN
  - 10: transporte capaz de ECN, ECT(0)
  - 01: transporte capaz de ECN, ECT(1)
  - 11: congestión encontrada, CE
- Si ambos extremos soportan ECN mandan sus paquetes con ECT(0) y ECT(1) respectivamente.
- Si el paquete atraviesa la cola congestionada y el enrutador soporta ECN, se cambia código en el paquete a CE para avisar al receptor de la congestión.
- Se usan dos banderas en encabezado TCP para soportar ECN:
  - ECE (ECN echo): se usa para mandar indicación de congestión al emisor.
  - CWR (ventana de congestión reducida): es usada para confirmar que la indicación ECE fue recibida.

Secuencia de ejecución de ECN típica:

- Se negocia ECN en conexión TCP
- Emisor manda un paquete IP “P” con ECT(0)
- “P” llega a un enrutador congestionado que soporta ECN y enrutador marca “P” con CE.
- Receptor recibe “P” con CE y manda segmento “Q” (con ACK de “P”) de vuelta usando bandera ECE prendida.
- Emisor recibe “Q” con ECE prendido, entonces emisor reduce ventana de congestión.
- El emisor manda el siguiente segmento al otro extremo usando bandera CWR prendida para confirmar la recepción de aviso de congestión.
- Nota: Se continúa transmitiendo segmentos con ECE prendido hasta recibirse segmento con CWR prendido.

Un problema del método de paquetes reguladores es que a altas velocidades o distancias grandes, el envío de un paquete regulador a los hosts de origen no funciona bien porque la reacción es muy lenta. La solución es usar el método de paquetes reguladores de salto por salto, hacer que el paquete regulador ejerza su efecto en cada salto que da.

## Protocolo de capa de Red IP

Protocolo de CR IP (protocolo de internet).

– Su propósito:

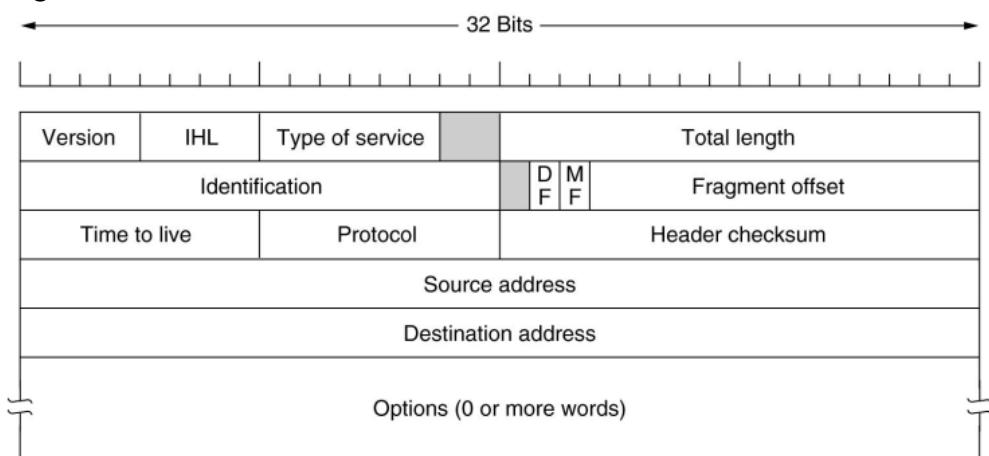
- Explicar formato de datagramas.
- Definición de direcciones IP.
- Definición de redes.
- Definición y uso de tablas de reenvío.
- Manejo de fragmentación de paquetes.

### ¿Por qué estudiar IP?

1. Para entender cómo se hacen asignaciones de direcciones de red a máquinas en una red local, a instituciones varias.
  - Para entender cómo designar o identificar a las redes.
2. Para comprender cómo se hace el reenvío en internet.
3. Para comprender cómo se hace la fragmentación y reensamblado de mensajes.
4. IP da la base conceptual para entender otros protocolos de capa de red en internet.

- Solo estudiamos IPV4, IPV6 en el complemento.
- secciones en la que se divide.

### 1. Datagramas IP



- IHL : Longitud del encabezado en palabras de 32 bits
- Total Length : 2B de encabezado + datos  $\leq 65535$  B
- Type of service : Los 6 primeros bits se usan para indicar clase de servicio
- Protocol : (8b) dice a cuál proceso de transporte (p.ej. TCP, UDP, etc.) entregar el paquete
- identificación : se usa para que el host de destino determine a qué paquete un fragmento pertenece
- tiempo de vida : tiempo de vida de un paquete
- Header checksum : se usa para detectar errores cuando el paquete viaja a lo largo de la red.

## 2. Direcciones IPv4:

En un datagrama IP los campos de direcciones de origen y de destino:

- Cada una tiene 32 b.
- indican el número de red y el número de máquina.
- Consecuencias:
  - uso números IP diferentes para distinguir las máquinas de una red.
  - Las direcciones IP son jerárquicas.
- Cada host y enrutador en la internet tiene una dirección IP.
  - Una máquina puede tener más de un IP
    - Una máquina tiene un IP por cada red a la que está conectada

## 3. Conceptos fundamentales en los que nos basamos:

Para entender cómo asignar nombre a redes y cómo describir ciertos parámetros de las mismas.

Una red corresponde a un bloque contiguo del espacio de direcciones IP llamado prefijo.

- Prefijos se escriben dando la dirección IP más baja en el bloque y la cantidad de bits usadas para la dirección de la red.
- Ejemplo: ¿Qué significa el prefijo 128.208.0.0/24?
  - La porción de la red es de 24 bits
  - Que tengo  $2^{24}$  máquinas en la red
  - La dirección IP más baja en el bloque es 128.208.0.0.

Máscara: está formada de 1's para identificar la red, seguido de 0's para identificar las máquinas

ejemplo:

¿Cuál es la máscara de 128.208.0.0/24?

- 11111111 11111111 11111111 00000000
- Otra forma de expresarla es: 255.255.255.0

Subredes: Conjunto de interfaces de dispositivos con la **misma parte de red** de la dirección IP

## 4. Asignaciones de redes a organizaciones

Para entender cómo se hace la asignación de redes a organizaciones teniendo en cuenta los conceptos y problemas mencionados.

Evitar tablas de reenvío demasiado grandes:

¿Cómo asignar a una organización una red sin que se desperdicien demasiadas direcciones y sin que las tablas de enrutamiento crezcan demasiado?

- red chica: si crece demasiado vas a tener que asignar otro prefijo
- red grande: asignar una red muy grande y que la terminen utilizando pocas máquinas es otra deficiencia
- Colocar todas las subredes del mundo en una tabla de reenvío hace que la tabla sea demasiado grande.

solución: Alojar las direcciones IP de una red en un bloque contiguo que permite  $2^k$  máquinas

CIDR: Implementación de la solución: CIDR (Classless Inter Domain Routing).

- En todas las máquinas de la red, la parte de la dirección IP para identificar la red es la misma.
- Se representa la red asignada con un único prefijo.

## **5. Tablas de enrutamiento**

**¿ Cómo definir las tablas de enrutamiento ?**

El enrutamiento es jerárquico y sólo se representan redes de organismos.

- Cada entrada de tabla de enrutamiento se extiende para darle una máscara de 32 bits.
- Tabla de enrutamiento para todas las redes tiene entradas:
  - (dirección IP inicio subred, máscara, línea de salida.)

**¿ Cómo se usa la tabla de enrutamiento cuando llega un paquete?**

1. Extraer dirección de destino IP.
2. Luego analizar la tabla entrada por entrada,
  - Hacer AND de la máscara de la entrada con la dirección de destino y comparar el resultado con la dirección IP de inicio de la subred de la entrada.
  - ¿Qué produce ese AND?
3. Si coinciden entradas múltiples se usa la máscara más larga (la red más pequeña).

## **6. Control de tamaño de tablas de enrutamiento**

- Uso de enfoque de agregación de prefijos
- Ejemplo: la misma dirección IP que un enrutador trata como parte de un /22 puede ser tratada por otro enrutador como parte de un /20 más grande.

Solución: CIDR (Classless Inter Domain Routing) Cont.

- Para evitar que las tablas de enrutamiento crezcan demasiado
- se combinan varios prefijos en un prefijo único más grande (conocido como superred).
  - A esto se le llama agregación de prefijos.
  - Cuándo se utiliza la agregación de prefijos, es un proceso automático.
  - La agregación de prefijos es fuertemente usada en la Internet y puede reducir el tamaño de las tablas de los enrutadores en alrededor de 200.000 prefijos.

## **7. Racionamiento de uso de direcciones IPv4**

- Para prolongar el uso de IPv4 a pesar de la escasez de direcciones IPv4.
- Uso del enfoque NAT.

**Situación:** Un ISP tiene una red de /c; esto quiere decir que se le dan  $2^{(32 - c)}$  números IP para máquinas.

- Con el esquema actual los clientes no pueden tener más de  $2^{(32 - c)}$  máquinas usando el servicio del ISP en un momento dado.

**¿Cómo aumentar la cantidad de máquinas?**

- traducción de dirección de red (NAT).
  - Asignar un solo N° de IP a cada organización para el tráfico de internet.

- a. Dentro de la organización cada computadora tiene una dirección IP única que se usa para el tráfico interno(o sea no se usan en internet si no dentro de las organizaciones y puede repetirse en distintas organizaciones)
- b. Cuando un paquete sale de la organización y va al ISP, se presenta una traducción de dirección (de la dirección de la computadora en la organización a la dirección IP usada por la organización en internet )

**ISP:** El proveedor de servicios de Internet, (**ISP**, por las siglas en de **Internet Service Provider**)

**implementación:** Para hacer posible este esquema los 3 rangos de direcciones IP se han declarado como privados

- las organizaciones pueden usarlo internamente cuando deseen
- La única regla es que ningún paquete que contiene estas direcciones puede aparecer en internet. Los 3 rangos reservados son :
  - 1.0.0.0 – 10.255.255.255/8 (16,777,216hosts)
  - 172.16.0.0 – 172.31.255.255/12 (1,048,576hosts)
  - 192.168.0.0 – 192.168.255.255/16 (65,536hosts)

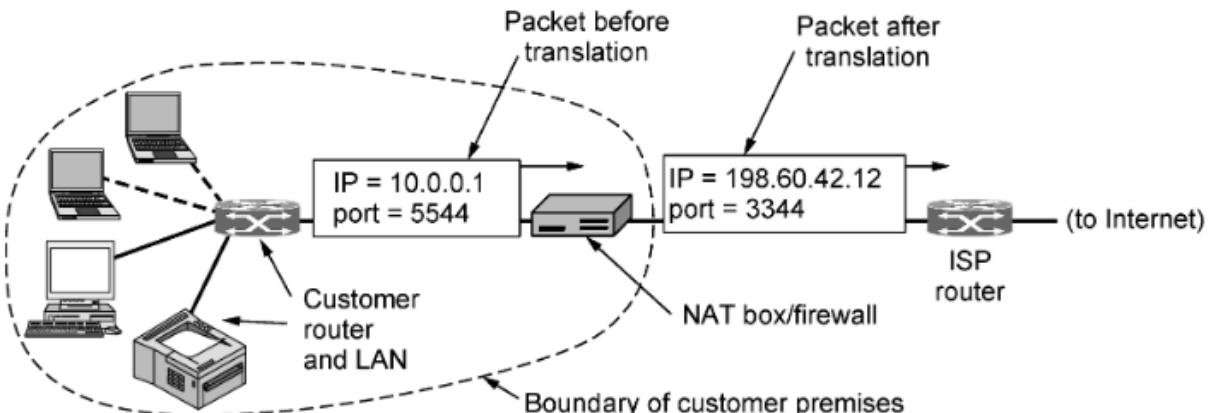


Fig. 60: Colocación y operación de la caja NAT

Supongamos que una organización cada máquina tiene dirección 10.x.y.z

¿ Cómo hacer cuando un paquete sale de las instalaciones de la organización ?

El paquete pasa a través de una caja NAT que convierte la dirección interna de origen de IP a la dirección IP de la organización(Vemos su implementación más adelante, primero veamos puertos que son necesarios para esta)

### **PUERTOS:**

Cada mensaje TCP saliente contiene puertos de origen y destino que sirven para identificar los procesos que usan la conexión en ambos extremos

¿ Qué pasa con el uso de los puertos cuando un proceso quiere establecer una conexión TCP con un proceso remoto ?

- se asocia a un puerto TCP sin usar en su máquina conocido como puerto de origen (Indica donde enviar mensajes entrantes de esta conexión )
- el proceso otorga también un puerto destino para indicar a quién dar los mensajes del lado remoto

**Problema:** Cuando la respuesta vuelve, por ejemplo de un servidor web, se dirige naturalmente a la dirección IP de la compañía

¿ Como sabe ahora la caja NAT con que dirección se reemplaza?

Solución 1: Guardar asociación en la caja NAT de número IP al puerto de origen que viene en el mensaje TCP/UDP dentro del paquete

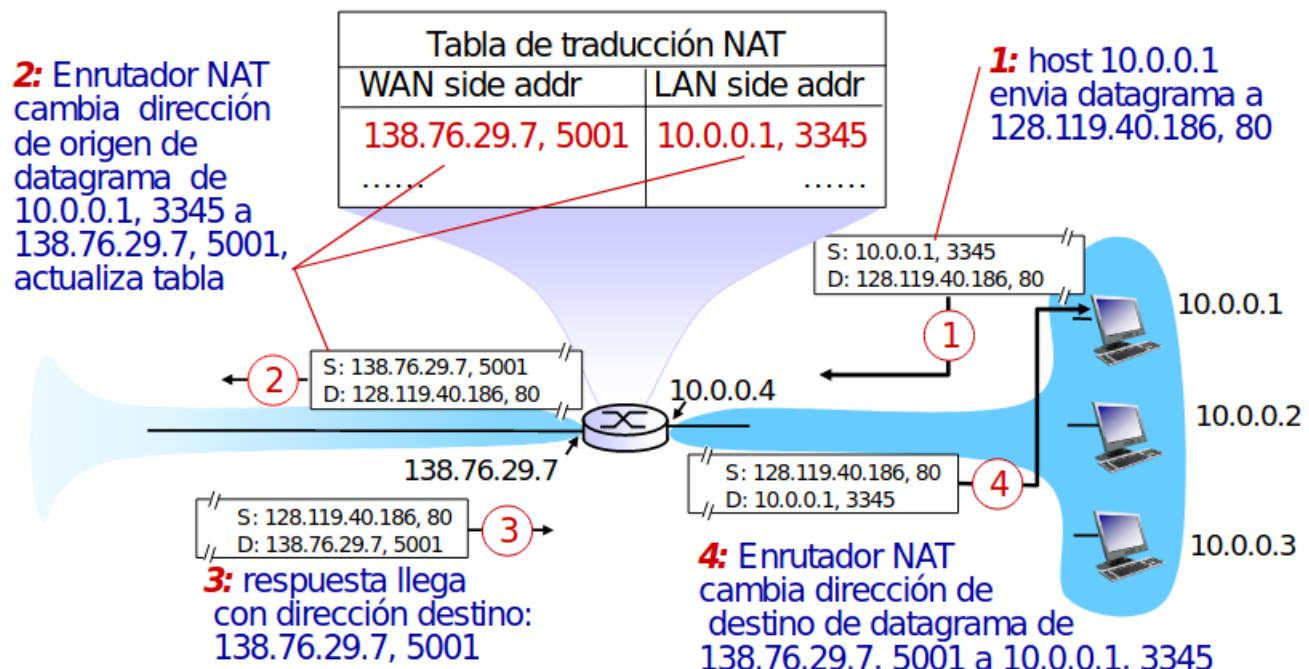
- Estas asociaciones pueden ser guardadas en una tabla en la caja NAT

Solución 2: Distinguir entre el número de puerto usado para identificar la máquina (osea IPs en la red interna) y el número de puerto usado por TCP/UDP para identificar la conexión

### Tabla de traducción de la caja NAT

- Los índices en la tabla son números de puertos para identificar la máquina
- Una entrada de la tabla contiene un par (número de puerto para identificar la conexión , dirección IP)

## NAT



¿ Cómo tratar un paquete que llega a la caja NAT desde el ISP?

El puerto de origen en el encabezado TCP se extrae y usa como un índice en la tabla de traducción de la caja NAT

- Desde la entrada localizada la dirección IP interna y el puerto TCP se extraen e insertan en el paquete
- Entonces el paquete se pasa al enrutador de la compañía para su entrega normal usando la dirección 10.x.y.z

¿Cómo tratar un paquete saliente de la caja NAT ?

La dirección de origen 10.x.y.z se reemplaza por la verdadera dirección IP de la compañía y el campo puerto de origen TCP se reemplaza por un índice en la tabla de traducción de la caja NAT

### **Críticas a NAT:**

- Viola el modelo de IP que dice que cada número IP pertenece solo a una máquina
- Si la caja Nat se cae, se pierde toda su traducción y por lo tanto todas las conexiones TCP se destruyen
- Retrasa la adopción de IPv6

## Protocolo IP

Campo tiempo de vida en los paquetes:

- Contador para limitar el tiempo de vida de un paquete
- Debe ser decrementado en cada salto
- Cuando llega a 0 el paquete es descartado y un paquete de advertencia se envía al host de origen
- Previene que un datagrama esté dando vueltas para siempre

## Subredes

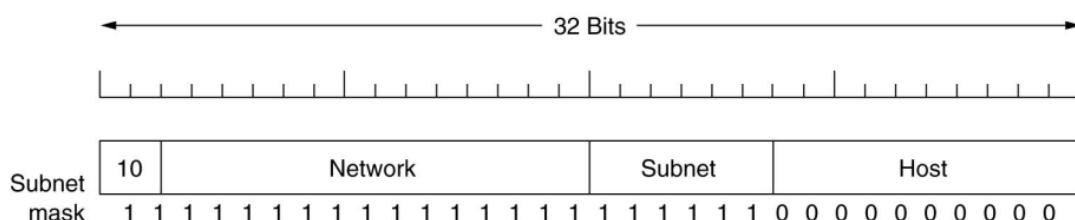
Permiten que una red sea dividida en varias partes para uso interno pero que actúe como una red simple para el mundo externo.

- Cada subred puede ser una LAN que tiene un enrutador
- Los enrutadores de una subred se conectan a un enrutador principal
- Subredes no visibles por fuera de la red

Cuando un paquete entra al enrutador principal, ¿cómo saber a qué subred pasarlo?

Soluciones:

- Tener una tabla en el enrutador principal que indique qué enrutador usar para cada host.
  - Se necesitaría una tabla demasiado grande.
- Eliminar algunos bits del N° de host para crear un número de subred
  - Ej: una red dividida en 64 subredes



### **¿Cómo expresar subredes?**

El enrutador principal usa una **máscara de subred** que indica la división entre el número de red + número de subred y el host.

La máscara es una información resumida de la red que me permite saber cuántos bits tiene la dirección de red y cuántos bits tenemos para las máquinas (cantidad de 0s para hosts y cantidad de 1s para bits de red). P.ej. si la universidad tiene 35 departamentos, se usa 6 bits para el número de subred y 10 bits para el número de host; lo que permite hasta 64 Ethernets, cada uno con a lo sumo 1022 hosts. La máscara se ve así:

255.255.252.0 o también IP/22 en CIDR

11111111 11111111 11111100 00000000

### Tabla de enrutamiento en un enrutador cuando hay subredes

Si se tienen entradas de la forma (dirección IP inicio subred, máscara) se hace AND booleano con la dirección de destino del paquete y cada máscara de subred para deshacerse del número de host y buscar la dirección resultante en sus tablas

La cantidad máxima de hosts se da por la cantidad de 0's a la derecha del último 1 en la dirección de origen.

## Complementos a IP

### IPv6

Con IPv4 algunos campos del encabezado hacen que el procesamiento de datagramas en los enrutadores lleve tiempo.

Formato de datagrama IPv6:

- Encabezado de longitud fija de 40 bytes para procesamiento más rápido de datagramas
- Capacidad de direccionamiento expandida: direcciones de 128 bits.
- Etiquetado de flujos: se etiquetan paquetes que pertenecen a un mismo flujo para los cuales el emisor requiere manejo especial. (por ejemplo transmisión de audio y video)

**Consecuencia de etiquetado de flujos:** Cuando un paquete con una etiqueta de flujo distinta de cero aparece, los enrutadores pueden ver en tablas internas para ver qué tipo de tratamiento especial requiere

**Etiqueta de flujo:** (20 b) para identificar datagramas en el mismo “flujo”

**Prioridad (pri)** tiene dos usos:

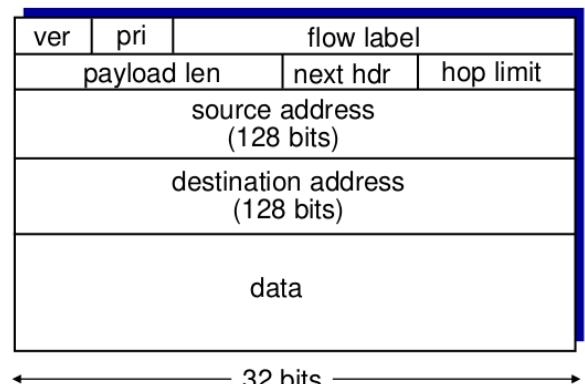
- para dar prioridad a ciertos datagramas dentro de un flujo.
- para dar prioridad a datagramas de ciertas aplicaciones sobre datagramas de otras aplicaciones.

**Longitud de carga útil:** (16 b) número de bytes en el datagrama IPv6 luego del encabezado (de 40 B).

**Límite de saltos:** (8 bits) el contenido de este campo se decrementa en 1 por cada enrutador que entrega el datagrama. Si el contador alcanza 0, el datagrama se descarta.

**Próximo encabezado:** (8 bits) significa:

- Cuál de los 6 encabezados de extensión de opciones actuales le sigue al encabezado.
- Si este encabezado es el último encabezado IP, el campo dice a cuál protocolo de transporte entregar el datagrama.
- Los encabezados de opciones también tienen este campo.



## DHCP: Protocolo de Configuración Dinámica de Host:

Meta: permitir a los hosts cuando se unen a una red obtener dinámicamente su dirección IP a partir de un servidor de red.

1. Host transmite “DHCP discover” msg [opcional]
2. Servidor DHCP responde con “DHCP offer” msg [opcional]
3. Host pide dirección IP : “DHCP request” msg
4. Servidor DHCP envía dirección: “DHCP ack” msg

DHCP puede retornar más que la dirección IP alojada en una subred:

- Dirección del enrutador del primer salto para el cliente
- Nombre y dirección IP del servidor DNS
- Máscara de red

DHCP es ampliamente usado en redes de acceso a internet residenciales y en redes LAN inalámbricas.

## UPnP: Universal Plug and Play

**Problema:** ¿Cómo puede un host detrás de NAT permitir pedidos de conexiones entrantes?

**Solución:** Usar protocolo Universal Plug and Play (UPnP)

Una aplicación ejecutada en un host puede pedir un mapeo NAT entre su (IP privado, Port privado) y su (IP público, Port público).

- ¿Si se acepta el pedido y se crea el mapeo entonces qué consecuencias tiene?
  - Nodos de afuera pueden iniciar conexiones TCP con el (IP público, Port público) asignado.
- ¿Cómo se pueden enterar máquinas de afuera de un servicio disponible por detrás de una NAT?
  - UPnP permite a la aplicación conocer el valor de (IP público, Port público) de modo que la aplicación lo puede avisar al mundo externo.

## ARP

- Problema: ¿cómo se convierten direcciones IP en direcciones de Ethernet?
- Solución: protocolo de resolución de direcciones (ARP): el host de origen da salida a un paquete de difusión hacia Ethernet preguntando: ¿quien posee una dirección IP w.x.y.z ?
  - Nota: para hacer una difusión la dirección de destino consiste solo de 1s.
- 1. La difusión llegará a cada máquina en Ethernet y cada una verificará su dirección IP.
- 2. Al host de destino le bastará con responder con su dirección de Ethernet E.
- 3. Así el host de origen aprende que la dirección IP de w.x.y.z está en el host con la dirección de Ethernet E.

### Optimizaciones para ARP

- 1) Una vez que una máquina ha ejecutado ARP, guarda el resultado en caso de que en poco tiempo tenga que ponerse de nuevo en contacto con la misma máquina.
  - La próxima vez encontrará la correspondencia en su propia caché, eliminando así la necesidad de una segunda difusión.
- 2) En muchos casos el host de destino necesitará devolver una respuesta, forzando también a que se ejecute el ARP para determinar la dirección Ethernet del emisor.
  - Puede evitarse teniendo el host de origen que incluir su correspondencia IP a Ethernet en el paquete ARP.
  - Cuando la difusión de ARP llega al host de destino, se introduce la dirección IP y de Ethernet del origen en el caché del host 2 para su uso futuro.
- 3) cada máquina difunde su correspondencia cuando arranca.

- Esto se hace mediante un ARP que busca su propia dirección IP.
- No debe haber una respuesta, pero un efecto lateral de la difusión es hacer una entrada en el caché ARP de todas las máquinas.
- Si llega inesperadamente una respuesta, es que la misma dirección IP se ha asignado a dos máquinas.
- La más reciente debe avisar al gerente de sistemas y no arrancar.

## OSPF

**Definición:** Un sistema autónomo (SA) consiste en un grupo de enrutadores bajo el mismo control administrativo.

- Generalmente distribuidos por los ISP
- A veces los ISP dividen su red en varios SA
- Los enrutadores dentro de un SA corren el mismo algoritmo de enrutamiento (protocolo de enrutamiento intra-SA)
- Internet es un conjunto de SAs

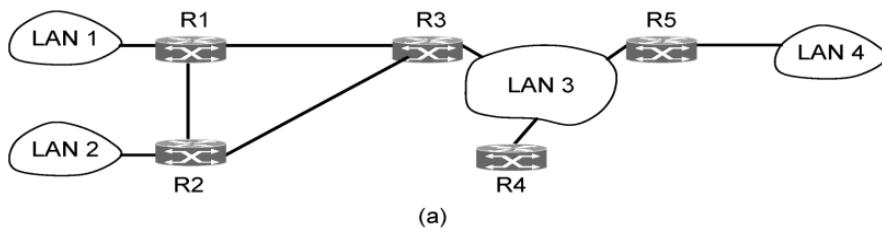
### ¿Por qué se necesita definir un protocolo intra-SA especial para internet?

- Los protocolos de enrutamiento estudiados no son compatibles con IP por la forma de las tablas de enrutamiento que se usaban.
- Los protocolos de enrutamiento anteriores estudiados no son adecuados cuando un SA es demasiado grande (se hace pesado consultar y actualizar las tablas de enrutamiento).
- El modelo de grafo para los protocolos de enrutamiento vistos no es adecuado cuando se trabaja con IP (los destinos son subredes con prefijo en lugar de enrutadores).
- A veces hay más de un camino más corto a un destino y no se saca provecho de esta situación para balancear la carga que tiene un enrutador.

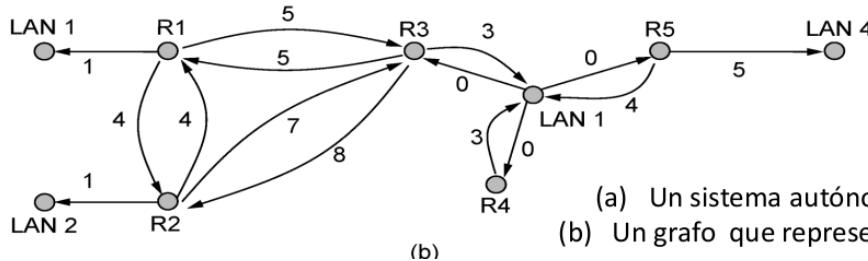
### OSPF (Open Shortest Path First)

- Es un protocolo de puerta de enlace interior (IGP, hace referencia a los protocolos usados dentro de un SA).
- Se considera una adaptación del método de enrutamiento de estado de enlace
- Es compatible con IP.
- En OSPF el modelo de grafo asociado a un SA es bastante más flexible que el usado para los protocolos de enrutamiento anteriores al considerar redes de distintos tipos.
- Para permitir los SA grandes, OSPF organiza un SA como una jerarquía de niveles.
- Con OSPF para un destino se puede considerar más de una línea de salida (cuando hay más de un camino óptimo) para balancear la carga en la red.

### Tipos de conexiones y redes soportadas por OSPF



(a)

(a) Un sistema autónomo.  
(b) Un grafo que representa (a).

- 1) Líneas punto a punto entre dos enrutadores (como R1 y R3)
- 2) Redes multiacceso con difusión (la mayoría de las LAN)
- 3) Redes de multiacceso con muchos enrutadores, cada uno de los cuales se puede comunicar directamente con los otros. (LAN 3 de (a))

### Representación de la red en un grafo

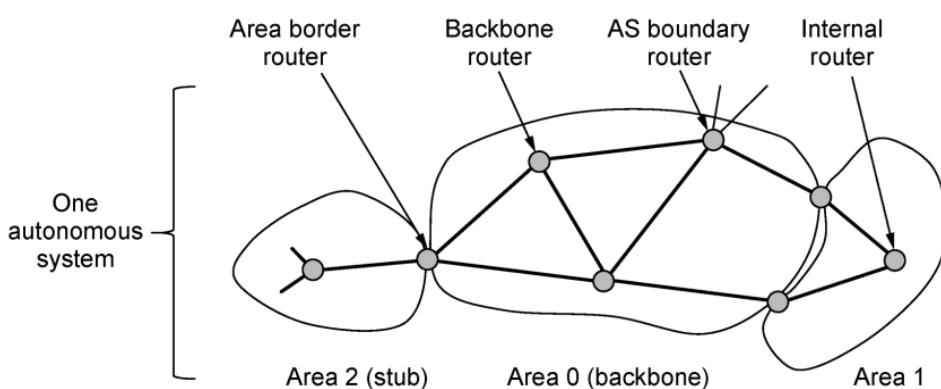
Los enrutadores se representan con nodos y a cada arco se le asigna un costo o retardo.

- Una **conexión punto-punto** entre dos enrutadores se representa por un par de arcos, uno en cada dirección.
  - Sus pesos pueden ser diferentes.
- Una **red de multiacceso** se representa con un nodo para la red en sí.
  - Los arcos desde el nodo de la red a los enrutadores tienen peso 0.

**¿Cómo conviene organizar un SA muy grande?** → Considerar un SA como una red jerárquica.

OSPF divide los SAs en áreas numeradas.

- Un área puede contener varias redes dentro de ella.
- Cada enrutador está configurado para conocer qué otros enrutadores están en su área.
- Las áreas no se traslapan



Tipos de áreas:

- Red dorsal (backbone) que tiene número 0, cuya topología no es visible desde fuera.
- Áreas conectadas a la red dorsal

Clasificación de los enrutadores de un SA:

- Enrutadores internos: yacen completamente dentro de un área.
- Enrutadores dorsales: enrutadores en un área dorsal

- Enrutador de borde de área (EBA). Es parte de una red dorsal y a la vez de una o más áreas.

Un tipo de Aviso de estado de enlace (AEE) contiene el costo de un enrutador a todos sus vecinos. Los EBA resumen información de enrutamiento aprendida de un área para hacerla disponible en sus AEE que envían a las otras áreas.

### **¿Cómo definir la información resumida de un área no dorsal?**

- Un EBA E recibe avisos de estado de enlace de todos los enrutadores de una de sus áreas A y con esa información determina el costo de alcanzar cada LAN de A.
- La información resumida de A contiene el costo de alcanzar cada LAN de A. Este paquete es puesto por el EBA E en la red dorsal para que llegue a las demás áreas.

### **¿Cómo definir la información resumida de un área dorsal?**

Por medio de un grafo donde:

- Todos los arcos unen pares de EBA
- El peso de cada uno de estos arcos es el costo de camino óptimo (en el área dorsal) que une el par de EBAs.
- Esto permite que todos los enrutadores del área dorsal aprendan el costo de alcanzar todas las redes de cada área.
- Todos los enrutadores aprenden a alcanzar todas las redes en el SA.
- Cada enrutador tiene una topología de su área detallada y solo conoce el costo del camino más corto a las redes en las otras áreas.

Al ejecutarse OSPF los enrutadores dentro de un área ejecutan una adaptación del protocolo de estado de enlace.

Cuando un enrutador se inicia, envía mensajes Hello a:

- Todas las líneas punto a punto
- El grupo de todos los otros enrutadores de una LAN (si está conectado a una LAN)
- De las respuestas cada enrutador aprende quiénes son sus vecinos.
- Los enrutadores en la misma LAN son todos vecinos.
- OSPF no fija una política de cómo los pesos de los enlaces son fijados.
  - Este es el trabajo del administrador de la red.
- OSPF trabaja intercambiando información entre enrutadores adyacentes.

Cada enrutador tiene base de datos de estado de enlace (BDEE).

- La BDEE contiene todos los AEE que el enrutador ha recibido.
- La BDEE debe ser creada, y luego mantenerse.
- Dentro de un área cada enrutador debe tener el mismo grafo (BDEE) para construir la tabla de reenvío.

Consecuencias de tener BDEE:

- En la BDEE se guarda información que un enrutador puede intercambiar con sus vecinos.
- La información de una BDEE puede ser actualizada luego que un enrutador recibe AEE de vecinos.

Tipos de paquetes usados para intercambio de información entre enrutadores adyacentes:

- Paquete de actualización de estado de enlace (PAEE): para mandar AEE asociado al enrutador emisor. Estos AEE tienen número de secuencia. Usando dicho número de secuencia el receptor puede ver si un AEE es más nuevo o más viejo que el que ya tiene.
- Paquete de confirmación de estado de enlace (PCEE): para confirmar los PAEE.
- Paquete de descripción de base de datos (PDBD): llevan resumen de la descripción de todos los AEE de la BDEE del enrutador emisor,
  - o sea, números de secuencia de los AEE del enrutador emisor.

- El receptor puede determinar cuáles AEE de ese grupo necesita, comparando número de secuencia de un AEE con número de secuencia de AEE (del mismo enrutador) que ya tiene.
- Paquete de pedido de estado de enlace (PPEE): se usan para solicitar AEEs.

### ¿Cómo actualizan sus BDEE los enrutadores?

Dos enrutadores vecinos deben sincronizar sus BDEE.

- Un vecino es el maestro y el otro es el esclavo. El maestro controla el intercambio de PDBD.
- Se intercambian PDBD, PPEE, PAEE, PCEE para asegurar que ambos vecinos tienen igual información en sus BDEE.

Como es ineficiente que cada enrutador en una LAN intercambie información con todos los otros, se elige uno de ellos como **enrutador designado**, quien intercambia mensajes con todos los enrutadores de la LAN mediante sincronización.

Usando **inundación** cada enrutador informa a todos los demás (de su área) sus enlaces con otros enrutadores y sus respectivos costos.

- Este intercambio se hace periódicamente, cuando una línea se cae o regresa y cuando cambia su costo.
- Con este proceso se construye un grafo de su área.

Para un enrutador R dentro de un área se puede ejecutar el algoritmo de Dijkstra.

- Para esto usar la BDEE de R.
- Dijkstra calcula el camino más corto desde R a cualquier otro enrutador de su área y red en el SA entero.

OSPF calcula todos los caminos más cortos entre dos nodos, y por ello puede dividirse el tráfico de envío entre ellos.

- Para esto se usa una cola de prioridades en el algoritmo de Dijkstra en un EBA (este Dijkstra modificado se llama ECMP, Multicamino de igual costo)

## Interredes

Varias redes pueden tener distintos protocolos. Los **enrutadores multiprotocolo** conectan estas redes. Para enviar paquetes a una red con distinto protocolo:

- 1) Las puertas de enlace traducen o convierten paquetes de un protocolo a otro.
- 2) Se construye una capa arriba de las diferentes redes que oculte las diferencias entre las distintas redes.

### Enrutamiento en interredes

Una vez construido el grafo de la interred pueden aplicarse algoritmos de enrutamiento al grupo de enrutadores multiprotocolo.

- En cada red se usa un **protocolo de puerta de enlace interior (IGP)**.
- Entre las redes se usa **protocolo de puerta de enlace exterior (EGP)** o **(PPEE)**.
- Cada red puede usar distintos IGP pero el EGP debe ser uno solo.

## Fragmentación

Ocurre cuando un paquete de gran tamaño quiere viajar a través de una red cuyo tamaño máximo de paquete es bastante más pequeño.

- Se fragmenta el paquete en la puerta de enlace
- Para recuperar el paquete original, se puede:

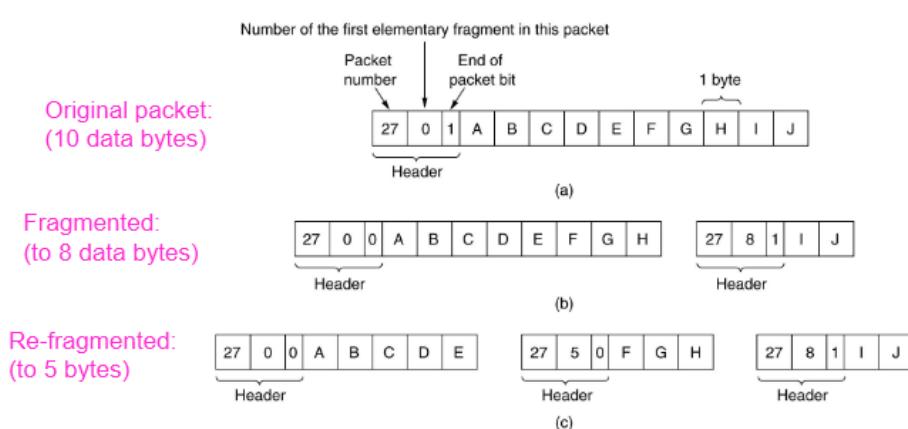
- 1) Dirigir todos los fragmentos a la misma puerta de salida y recombinar allí.
- 2) Tratar cada fragmento como el paquete original y recombinar en el host de destino.

### Desventajas de (1)

- La puerta de enlace de salida debe saber cuándo recibió todas las piezas, por lo que debe incluirse un campo de conteo o un bit de fin de paquete.
- Todos los paquetes deben salir por la misma puerta de enlace → menor desempeño.
- Sobrecarga de reensamblado y fragmentación repetidamente.

### Desventajas de (2)

- Todos los hosts deben ser capaces de re-ensamblar
- Al fragmentarse un paquete grande aumenta la sobrecarga total, ya que cada pieza debe tener un encabezado.



Fragmentación cuando el tamaño de datos elemental es de 1 B.

- (a) Paquete original contiene 10 B de datos.
- (b) Fragmentos luego de pasar por una red con tamaño de paquete máximo de 8 B de datos + encabezado.
- (c) Fragmentos luego de pasar por una red con tamaño máximo de paquete de 5 B de datos + encabezado..

## Entunelamiento

Tenemos una red distinta entre dos redes de la misma clase. Para mandar el paquete de una red a otra de la misma clase los paquetes son encapsulados en la red del medio usando un encabezado.

## EGP/PPEE(External Gateway Protocol/Protocolo de puerta de enlace exterior)

Para enrutamiento inter-SA es imposible encontrar un camino óptimo, ya que cada SA corre su propio protocolo interno, y es imposible calcular costos significativos en caminos que cruzan varios SA.

### Requisitos para un PPEE

- Encontrar algún camino de SAs sin ciclos.
- Respetar las políticas de los SAs (preferencias y limitaciones de enrutamiento)

Los PPEE suelen implementarse sobre Enrutadores de Borde de Sistema Autónomos (EBSA)

Un EBSA:

- Hace una elección de varias rutas a un destino
- Elige la mejor acorde a sus políticas (y ésta es la ruta que avisa)
- Informa a sus vecinos el camino usado para cada destino.

## Relaciones entre SA

El ISP cliente paga al ISP proveedor para entregar paquetes a otros destinos y recibir paquetes enviados de otros destinos.

- Proveedor-Consumidor: Un ISP proveedor entrega paquetes a destinos y recibe de otros. El proveedor debe dar publicidad de rutas al ISP consumidor sobre el enlace que los conecta. El consumidor pública rutas a los destinos en su red al proveedor.
- Compañerismo: los ISP no se cobran por mandarse mensajes. Los SA compañeros mandan publicidad el uno al otro para los destinos en sus redes.
- Multihoming: Un ISP está conectado a varios ISP, mejora la confiabilidad por si un ISP falla.

## BGP

Border Gateway Protocol: Es el PPEE usado por la internet.

- Provee a cada SA un medio para:
  - Obtener información de alcanzabilidad de las subredes desde SA vecinos
  - Propagar esta información dentro del SA
  - Determinar buenas rutas a las subredes
  - Que cada red publique su existencia al resto de la internet.

En BGP los destinos son prefijos. Cada prefijo representa una subred o colección de subredes.

Un SA es identificado por un número globalmente único ASN (Número de sistema autónomo)

- Cuando un enrutador avisa de un prefijo a lo largo de una sesión BGP, se incluye con el prefijo una ruta que pasa por varios SA, compuesta por un prefijo y **atributos BGP**.

### Atributos BGP:

- AS-PATH: contiene los ASN de los SA por los cuales el aviso de prefijo ha pasado
  - Se usa para detectar y prevenir ciclos.
  - También para elegir entre varios caminos al mismo prefijo.
- NEXT-HOP: IP de la interfaz del enrutador que comienza el AS-PATH hacia el destino.

### Propagar información de rutas en BGP

Pares de enrutadores intercambian información de rutas sobre conexiones TCP semipermanentes en el puerto **179**

### Sesiones BGP

- Es la conexión TCP con todos los mensajes BGP enviados
- Una sesión BGP entre enrutadores de dos SA se llama sesión externa BGP (eBGP)
- Una sesión BGP entre enrutadores del mismo SA se llama sesión interna BGP (iBGP)

### Aviso de rutas

Cuando una puerta de enlace P recibe rutas:

- P usa las sesiones iBGP para distribuir rutas a otros enrutadores dentro de su SA

La mejor ruta a un prefijo se guarda en la Base de información de Enrutamiento (BIE) siguiendo reglas:

- Rutas con mayor valor de preferencia local son elegidas
- De las restantes, la ruta con el camino AS-PATH más corto es elegida
- De las restantes, la que tiene el NEXT-HOP más cercano es elegida
- Si queda más de una se usan criterios adicionales

Mensajes de actualización son usados para:

- Información acerca de una ruta a través de internet
- Una lista de rutas previamente avisadas por el enrutador emisor, que no son más válidas

Un mensaje de actualización hace que la BIE se actualice y que se emitan mensajes de actualización hacia otros vecinos.

Un enrutador BGP **no tiene la obligación de avisar una ruta a destino**

### Política de Importación

Cuando una puerta de enlace recibe aviso de ruta usa su política de importación para aceptar o filtrar la ruta.

Si un enlace falla o cambia una política, los enrutadores BGP pueden cancelar caminos avisados previamente. (Avisa ruta removida)

## Capa de Enlace de Datos (CED)

→ Capítulo 5: Generalidades

- La CED toma de la CR paquetes y los encapsulan en tramas.
- Las tramas tienen una longitud máxima impuesta.
- Cada paquete de la CR se divide en tramas.
- En la CR de la máquina de origen hay un proceso que entrega bits a la CED para transmitirlos a la máquina de destino.
- El trabajo de la CED es transmitir los bits a la máquina de destino para que puedan ser entregados a su CR.

Limitaciones de los canales de comunicación

- Cometén errores ocasionales
- Tienen una tasa de datos finita
- Hay retardo de propagación

### Meta necesaria:

Lograr una comunicación confiable y eficiente entre dos máquinas adyacentes, o sea conectadas por un canal de comunicaciones.

**¿Cómo cumplir con este requisito?** Definir una capa debajo de la capa de red que se encargue de esto → CED

- Un protocolo de CED hace que las líneas de comunicación parezcan perfectas o al menos bastante buenas.

### Funciones de la CED:

#### Control de flujo

- Evitar que un emisor rápido sature a un receptor lento.
- Uso de protocolos de tubería.
- **Control de buffer receptor** → Parada y espera, Go-back-N, Selectiva.
- Se da entre un nodo receptor y uno que transmite un flujo de datos que fluye a lo largo de toda la red → se da salto a salto.

#### Entramado (flaming)

- En el canal de difusión solo hay un stream de bits.

- ¿Cómo detectar el inicio y fin de cada trama? Usualmente se usa un patrón especial de bits para ello (llamado bandera).
- **Inicio y fin de trama** → Preámbulo, Bandera de inicio.
- Se usa para que el nodo sepa cuando terminó el salto y donde inicia el próximo.

### Detección y corrección de errores

- Se agrega a los mensajes de CED (tramas) bits adicionales, ya sea para detectar errores en la transmisión o para saber cuál es el error.
- Checksum, CRC.
- Especie de “hash”, se chequea al final de la trama que los hash sean iguales si no hubo un error y la trama debe ser descartada o tirar un timeout.

### Manejo de colisiones

- Ocurren en canales de difusión usados por varias máquinas.
- Cuando dos máquinas intentan transmitir tramas al mismo tiempo ocurre una colisión.
- **Subcapa de Control de Acceso al Medio (SCAM o MAC)** (canales de difusión cableados/inalámbricos) → CSMA/CD (Acceso Múltiple con Detección de Portadora y Detección de Colisiones) (Ethernet).

### ¿Por qué estudiar la capa de enlace de datos?

Ayuda a comprender sobre el funcionamiento de las LAN (cableadas: Ethernet, inalámbricas: WiFi). Además, permite comprender los protocolos que resuelven los problemas de diseño de las LAN, como ser: control de flujo, control de colisiones y control de errores.

### Informaciones que debería contener una trama de CED:

- encabezado: suele contener direcciones del origen y del destino, a veces la longitud de la trama, etc.
- campo de carga útil (el contenido que se quiere enviar).
- un terminador final (para control de errores).
- alguna información adicional para el entramado.

Bytes	8	6	6	2	0-1500	0-46	4
(a)	Preamble	Destination address	Source address	Type	Data ↓↓	Pad	Check-sum
(b)	Preamble	S o F	Destination address	Source address	Length ↓↓	Data ↓↓	Pad

Formato de trama Ethernet

### Fundamentos de la comunicación en la CED:

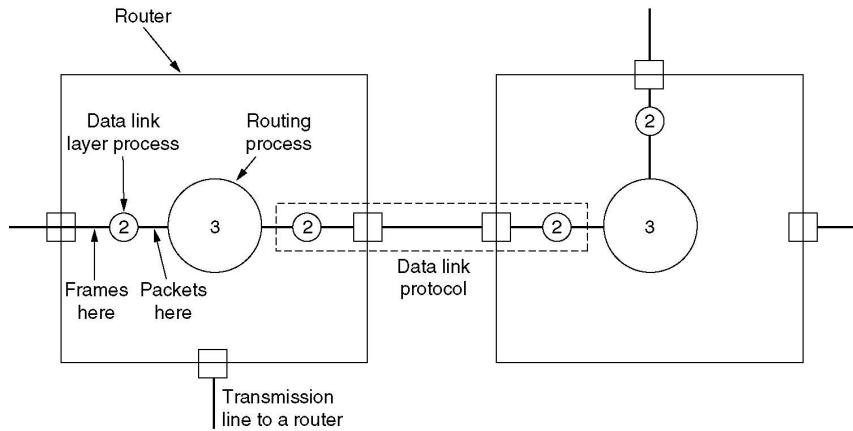
Se trabaja con:

- Confirmaciones de recepción de tramas
- Temporización de reenvío
- Retransmisiones de tramas (perdidas o dañadas)
- Uso de números de secuencia en las tramas (para identificar tramas duplicadas).
- Llevar a caballito (piggybacking) para aprovechar mejor el canal de comunicaciones.
- Control de flujo (para evitar que emisor sature a receptor más lento)

- go back N, repetición selectiva.

## Flujo entre enrutadores

1. Al llegar trama al enrutador el hardware verifica si está libre de errores.
2. La CED comprueba si esta es la trama esperada y de ser así, entrega el paquete dentro de la trama al software de enrutamiento.
3. El software de enrutamiento elige la línea de salida adecuada y entrega el paquete a la CED para enviarlo.



## Necesidad de Canales de Difusión:

Es costoso e incómodo hacer que todo par de máquinas de una organización estén conectadas directamente entre sí por dos canales (dedicados exclusivamente para ellas).

Si hay n máquinas daría n \*(n-1) conexiones.

Alternativa económica pero con nuevos problemas de diseño → [Canales de difusión](#)

- En un canal de difusión están conectadas varias máquinas que quieren transmitir tramas por el canal. Si una máquina envía un mensaje todas las demás lo reciben.

## Tipos de canales de difusión:

### Inalámbricos

- P.ej. por uso de señales de radio o de microondas.

### Cableados

- P.ej. De un cable coaxial salen cables a distintas máquinas.
- P.ej. de un concentrador salen cables a distintas máquinas (es la idea de triple o de zapatilla).

## Mediante el uso de conmutadores

### Necesidad de Control de Colisiones:

Si dos tramas se transmiten en forma simultánea en un canal de difusión se traslapan en el tiempo y la señal resultante se altera. Este evento se llama [colisión](#).

### ¿Cómo evitar colisiones o hacer que ocurran lo menos posible?

- Definir una subcapa de la CED que se encargue del control de colisiones.
- Esta subcapa se llama subcapa de control de acceso al medio SCAM.

- La subcapa MAC es una subcapa inferior de la CED.

La **SCAM** sirve para comprender cómo se organizan, diseñan, y funcionan las LAN cableadas e inalámbricas, y cómo los distintos tipos de LAN hacen control de colisiones. Para esto último se usan protocolos de control de colisiones.

- En una red de difusión el asunto clave es cómo determinar quién puede usar el canal cuando hay competencia por él. Los **protocolos de acceso múltiple (PAM)** determinan quién sigue en un canal de difusión.

## Soluciones al control de colisiones:

### Inalámbricas

- Estación base (access point) que coordina la comunicación entre hosts.
- Se usa protocolo 802.11 (WIFI).

### Cableadas

- Ethernet cuando varias máquinas se enchufan a un concentrador (Hub) o a un mismo cable (cable coaxial).
- Ethernet usa protocolo CSMA/CD para control de colisiones.

### → Capítulo 5: Control de colisiones en redes cableadas

Para comprender las capacidades de observación del canal que puede tener una máquina nos basaremos en supuestos:

#### 1. Modelo de Estaciones

- Hay N estaciones independientes que genera tramas para transmisión
- Una vez generada una trama, la estación se bloquea hasta que la trama se haya transmitido con éxito.

#### 2. Suposición de canal único:

Hay un solo canal disponible donde todas las estaciones pueden transmitir y recibir

## Propiedades de los canales de difusión modernos:

### Fenómenos sucediendo en un canal que una estación podría detectar

- Detectar que el canal está en uso (o sea, alguna estación está enviando una trama).
- Detectar que hay una colisión en el canal.

En las LAN actuales cada estación puede detectar si el canal está en uso (se fija si están llegando bits de alguna trama a la máquina que hace la detección). Los protocolos que pueden hacer esto se llaman **protocolos de detección de portadora (CSMA)**. La ventaja de poder hacer esta detección es evitar generar colisión poniendo tramas en el canal cuando están llegando bits de alguna trama. Además, las LAN pueden detectar si está ocurriendo una colisión cuando está transmitiendo una trama. Para esto se usa:

- El hardware de una estación escucha el cable mientras transmite.
- Si lo que lee es distinto de lo que puso en él, sabe que está ocurriendo una colisión.

**¿Qué hacer si se detecta una colisión?** Dejar de enviar la trama, abortar las transmisiones de la estación apenas se detecte una colisión.

## Ventajas de la detección de colisiones

- Ahorra tiempo y ancho de banda.
- Si no se detectan, la estación no va a recibir la confirmación de recepción (expira un temporizador) y va a tener que retransmitir la trama.

En conclusión, para definir PAMs conviene que una estación pueda detectar lo que está pasando en el canal.

### CSMA con Detección de Colisiones (PAM CSMA/CD):

#### En CSMA/CD el emisor:

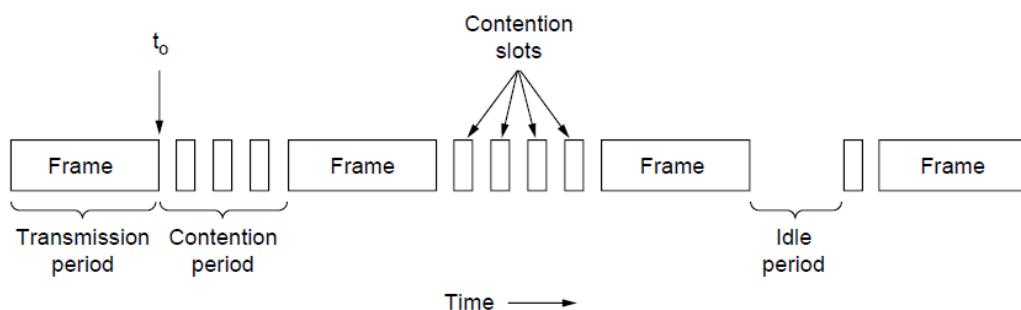
1. Antes de transmitir una trama detecta a la portadora (refiere a la frecuencia central donde se portan los datos).
2. Si el canal está libre transmite. Si no, espera hasta que el canal se desocupe para transmitir.
3. Si el emisor detecta una colisión, aborta la transmisión, espera un tiempo aleatorio, y una vez que pasó este tiempo: goto 1

#### En CSMA/CD el receptor:

1. Recibe una trama buena si no hubo colisión y el medio no cometió errores.
2. En caso contrario, recibirá una trama dañada la cual será descartada.
3. Al mandar una confirmación de recepción hace los pasos del emisor.

### Evaluación del uso del canal:

Períodos alternantes de contención y transmisión, ocurriendo períodos de inactividad cuando todas las estaciones no necesitan enviar tramas.



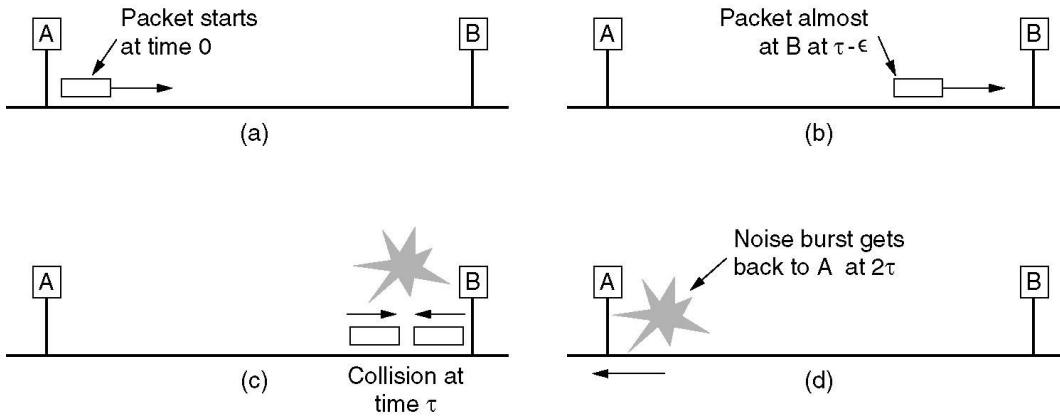
### Cuando se utiliza CSMA el canal puede estar en 3 estados.

1. Período donde el canal está ocioso, ninguna estación está enviando ninguna trama.
2. Período donde se está enviando una trama exitosamente.
3. Período de contención, donde no se logró transmitir ninguna trama con éxito. → hubo colisiones durante las ranuras de contención.

Se dice que una estación ha **tomado el canal** cuando todas las demás estaciones sabían que estaba transmitiendo y no interfirieron.

El **tiempo mínimo en detectar la colisión** es el tiempo que tarda la señal para propagarse de una estación a otra.

**El peor caso de demora de una estación en enterarse que ha habido una colisión es cuando el ruido de la colisión recorre la mitad de la distancia para llegar.**



$\tau$  el tiempo que tarda una señal en propagarse entre las dos estaciones más lejanas A y B

Cómo ocurre una colisión en CSMA/CD y cuándo se enteran las estaciones de ella::

1. A comienza a transmitir en  $t = 0$ .
2. En  $\tau - \epsilon$  un instante antes de que la señal llegue a B, B comienza a transmitir.
3. B detecta la colisión casi de inmediato y se detiene. En Ethernet se genera ráfaga de ruido de 48 bits.
4. La ráfaga de ruido causada por la colisión no regresa a A hasta pasados  $2\tau - \epsilon$ .

En el peor caso una estación no puede estar segura de que ha tomado el canal hasta que ha transmitido durante  $2\tau$  sin detectar una colisión.

Si una estación E intenta transmitir una trama demasiado corta y ocurre una colisión, la transmisión de E se completa antes de que la ráfaga de ruido llegue de regreso, en el momento  $2\tau$ . El emisor entonces supondrá incorrectamente que la trama se envió con éxito.

Para evitar esto las tramas deberán tardar más de  $2\tau$  para enviarse. Por lo tanto las tramas tienen un requisito de tamaño mínimo → 512 bits.

### Ethernet (IEEE 802.3):

Asuntos de la CED se pueden hacer por hardware: Entramado, control de errores, detección de portadora, detección de colisiones.

### Hardware que hay que tener para estos casos:

**Transceptor:** maneja detección de portadora y detección de colisiones.

**Tarjeta controladora:** Se encarga de ensamblar los datos en el formato de trama adecuado, calcular terminador de las tramas de salida, comprobar las tramas de entrada (p.ej detección de errores).

### Tipos de cableado en Ethernet:

Cada cableado de Ethernet tiene una longitud máxima de cable por segmento. Se utiliza el concepto de segmento, el segmento fija una limitación ya que su longitud determina la distancia máxima que puede haber entre dos máquinas. Una señal a medida que se va propagando por un cable se va debilitando. Llega un punto a partir del cual la señal es demasiado débil como para continuar su viaje.

Para que una señal pueda viajar más de ese punto se usan repetidores:

- Un repetidor es un dispositivo de capa física que recibe, amplifica (regenera) y retransmite señales en ambas direcciones
- Los repetidores introducen un retardo.

Para permitir redes mayores que un segmento en Ethernet conectar múltiples cables mediante repetidores.

**Restricción de Ethernet:** puede haber múltiples segmentos de cable y múltiples repetidores, pero ningún par de transceptores puede estar separado por más de 2,5 km y ninguna ruta entre dos transceptores puede atravesar más de 4 repetidores.

Para redes de más de un segmento → Conectar segmentos a repetidores, la distancia máxima entre dos máquinas se vuelve mayor a 1 segmento.

### Cómo diseñar una red de mayor velocidad

Supongamos que aumenta la velocidad de la red, y la longitud máxima del cable permanece igual. La longitud mínima de trama debe aumentar.

Supongamos que aumenta la velocidad de la red y la longitud de trama mínima no cambia. La longitud máxima del cable debe disminuir, de manera proporcional.

A medida que aumente la velocidad de la red, la longitud mínima de la trama debe aumentar o la longitud máxima del cable debe disminuir, de manera proporcional.

### Ethernet Conmutada:

A medida que se agregan mas y mas estaciones a Ethernet, aumenta el tráfico. En algún momento la LAN se saturará.

Para evitar este problema se usa → **Ethernet conmutada**.

Un conmutador (switch) contiene una matriz de conmutación de alta velocidad de 4 a 32 tarjetas de línea, cada tarjeta de línea contiene de 1 a 8 conectores. Hay matrices de conmutación que funcionan a más de 1 Gbps.

**Tarea realizada por un conmutador:** Almacenamiento y reenvío de tramas de Ethernet.

**Transparencia:** Los hosts no son conscientes de la presencia de conmutadores.

Los conmutadores aprenden por sí solos **no necesitan ser configurados**.

Si dos máquinas conectadas a la misma tarjeta de conexión transmiten tramas al mismo tiempo y si todos los puertos de la tarjeta forman una LAN local dentro de la tarjeta:

- Las colisiones en esta LAN en tarjeta se detectan y manejan igual que en una red CSMA/CD.
- Las tarjetas pueden estar transmitiendo en paralelo.

Si cada puerto de entrada se almacena en un búfer

- todos los puertos de entrada reciben y transmiten tramas al mismo tiempo, para una operación en paralelo duplex.
- Cada puerto es un dominio de colisión independiente.

**Cada conmutador tiene una tabla de conmutador:** <dirección MAC del host, interfaz para alcanzar el host, estampilla de tiempo>

- Un conmutador aprende cuáles hosts pueden ser alcanzados a través de cuáles interfaces.
- Cuando el conmutador recibe una trama registra el par emisor /localización en la tabla del conmutador.

Reenvío de una trama recibida por el conmutador:

Registrar enlace de ingreso, dirección MAC del host emisor de la trama.

- Identificación de la interfaz del destino:

Se busca en la tabla del conmutador la dirección MAC del destino:

**if** se encuentra la entrada para el destino

**then {**

**if** el destino está en el segmento por el cual vino la trama

**then** descartar trama

**else** enviar trama en la interfaz indicada por la entrada

**}**

- si no se encuentra una entrada para el destino:

**else** inundar /\* enviar en todas las interfaces excepto aquella por la que llegó la trama \*/

La ventaja de usar conmutadores es que se pueden enviar tantos datos por segundo como la capacidad de la matriz de conmutación de alta velocidad. A su vez, un conmutador tiene varios buffers por tarjeta o por cada puerto → hay muchas menos colisiones.

→ [Capítulo 5: Complementos sobre generalidades y control de colisiones en redes cableadas](#)

**Problemas al diseñar protocolo de CED:**

¿Cómo asegurar que una trama se entregue? → Si una trama no se entregó, entonces el emisor la reenvía.

**Para implementar esto:**

Regresar tramas de control con confirmaciones de recepción positivas o negativas de las tramas que llegan.

Método que usa temporizador de retransmisiones en la CED.

- Al enviarse una trama, se inicia un temporizador.
- Si la trama o la confirmación de recepción se pierden el temporizador expirará. Luego, se puede enviar la trama de nuevo.
- Si la confirmación de recepción llega antes que el temporizador expira, entonces el temporizador se cancela.

Se perdió una confirmación de recepción y se envió la trama de nuevo → La misma trama llega dos o más veces al receptor y la CED la pasa a la CR más de una vez.

¿**Cómo hacer para evitar entregar a la CR tramas repetidas?** → Método que asigna números de secuencia a las tramas que salen. El receptor tiene una función que dado un número de secuencia de la trama que llega decide si ella es duplicada.

**¿Qué hacer con un emisor que quiere transmitir tramas a mayor velocidad que aquellas con que puede aceptarlos el receptor?** Solución basada en retroalimentación el receptor autoriza al emisor a enviar más datos. (**control de flujo**).

**¿Cómo transmitir datos entre dos máquinas y en ambas direcciones eficientemente?** Solución llevar a caballito (**piggybacking**).

- Cuando llega una trama de datos, el receptor se aguanta y espera hasta que la CR le pasa el siguiente paquete P.

- La confirmación de recepción se anexa a la trama de datos de salida con P (usando el campo ack en el encabezado de la trama).

**¿Qué pasa si la CED espera demasiado por una trama a la cual superponer el ack?** El temporizador del emisor expirará y la trama será retransmitida.

**¿Cómo hacer para evitar que pase eso?** Si llega en menos de x mseds un paquete, el ack se superpone a él sino, la CED manda **trama de ack independiente**.

## PAM: ALOHA puro

### El emisor:

- Transmite cuando tiene datos para enviar.
- Escucha el canal por un tiempo igual a la demora de propagación de ida y vuelta máxima en la red más un incremento fijo de tiempo.
- Si se escucha un ack en ese tiempo, todo anduvo bien. Si no, se espera un tiempo aleatorio y la trama se manda de nuevo.
- Si se falla en recibir un ack luego de varias retransmisiones se tira la toalla.

### El receptor:

- Al recibir una trama chequea su validez y si lo es, inmediatamente manda un ack.
- Si la trama es inválida el receptor la ignora, la trama puede ser inválida por ruido o por colisión.

## Evaluación de ALOHA puro

- Este método bajo carga baja es eficiente y tiene una demora baja.
- En ALOHA puro una estación no escucha el canal antes de transmitir esto generará probablemente muchas colisiones.
- Como el número de colisiones crece rápidamente a medida que aumenta la carga, la máxima utilización del canal es alrededor del 18%.

## CSMA persistente 1

### Protocolo CSMA persistente 1 para el emisor:

- Si una estación tiene datos por enviar, primero escucha el canal para saber si otra está transmitiendo en ese momento.
- Si el canal está ocupado, entonces la estación espera hasta que se desocupe.
- Cuando la estación detecta un canal inactivo, transmite una trama.
- Si ocurre una colisión, la estación espera una cantidad aleatoria de tiempo y comienza de nuevo.

#### Comportamiento luego que emisor envió una trama →

- La estación espera un tiempo razonable por un ack, teniendo en cuenta el tiempo de propagación de ida y vuelta máximo en la red y el hecho que la estación receptora también debe competir por el canal para responder.
- Si no recibe ack en ese tiempo, la estación espera una cantidad aleatoria de tiempo y comienza de nuevo.

### Protocolo CSMA persistente 1 para el receptor:

- Al recibir una trama chequea su validez y si lo es, inmediatamente manda un ack.
- Si la trama es inválida el receptor la ignora, la trama puede ser inválida por ruido o por colisión.

El retardo de propagación tiene un efecto importante en el desempeño de CSMA persistente 1.

- Caso de que justo después de que una estación comienza a transmitir, otra estación está lista para enviar, si la señal de la primera estación no ha llegado aún a la segunda, esta última detectará un canal inactivo y comenzará a enviar también, eso producirá una colisión.
- Cuanto mayor sea el tiempo de propagación, más importante será este efecto.

Aun si el retardo de propagación es cero, habrá colisiones → dos estaciones quieren enviar y detectan que una tercera está transmitiendo. Luego que la tercera termine de transmitir las dos estaciones que quieren enviar detectarán un canal inactivo, por lo tanto enviarán y se producirá una colisión.

## Ethernet

	Bytes	8	6	6	2	0-1500	0-46	4
(a)	Preamble	Destination address	Source address	Type	Data	Pad	Check-sum	

Trama DIX Ethernet (Dec, Intel, Xerox)

Preámbulo de 8 bytes, cada uno es 10101010

### Direcciones

- Se usan direcciones de 6 bytes
- Se escriben como 6 pares de dígitos hexadecimales separados por '-'. → P.ej: **1A-23-F9-CD-06-9B**
- El bit de orden mayor de la dirección de destino es 0 para las direcciones ordinarias y de 1 para las direcciones de grupo.
- Una trama que consiste únicamente de bits 1 en el campo de destino
- se acepta en todas las estaciones de la red (broadcasting).

### Campo Tipo

- Uso de múltiples protocolos de CR a la vez en la misma máquina.
- El kernel debe saber a cual entregarle la info de la trama que llegó.
- El campo de tipo indica al receptor a qué proceso entregarle la trama.

### Longitud de trama mínima

- Las tramas deben tener al menos 64 bytes de largo, de la dirección de destino a la suma de verificación.
- **Cuando los datos más el encabezado ocupan menos de 64 bytes.**
- Cuando la porción de datos de una trama es menor a 46 bytes. Uso del campo de relleno (para alcanzar los 64 B)

### Suma de verificación

- Tiene 32 bits de largo.
- Se usa un método de detección de errores llamado código polinomial.

### Cuando IEEE estandarizó la Ethernet hizo los siguientes cambios al formato DIX

- Reducir el preámbulo a 7 bytes y usar el último byte para un delimitador de inicio de trama.
- Cambiar el campo de Tipo por un campo de Longitud.
- Poner un pequeño encabezado a los datos para dar información de tipo.

### Diferentes modos de cablear un edificio:

1. Un cable pasa entre cuarto y cuarto y cada estación se conecta a él en el punto más cercano.
2. Una **columna vertical** corre del sótano a la azotea y en cada piso se conectan cables horizontales a dicha columna.

- **¿hacen falta repetidores?** En cada piso conectar cable a columna con un repetidor entre ambos.

### 3. Topología de **árbol**

- El medio de transmisión es un cable que se divide en ramas.
- El árbol tiene puntos conocidos como **headends** donde uno o más cables comienzan (a su vez cada uno de estos podrá tener ramas).
- La transmisión desde una estación se propaga por el medio y puede ser recibida por todas las otras estaciones.

#### **Topologías de cables:**

- Lineal
- Columna (Spine)
- Árbol
- Segmentado

**Algoritmo de retroceso exponencial binario** → Algoritmo que determina en Ethernet el tiempo de espera del emisor cuando ocurre una colisión.

Tras una colisión el tiempo se divide en ranuras cuya longitud es igual al tiempo de propagación de ida y vuelta en el peor caso en el cable  $2\tau$ . El tiempo de ranura es 512 tiempos de bit o 5,12  $\mu$  seg. Cuando ocurre una colisión las estaciones afectadas por la colisión eligen cada una aleatoriamente una cierta cantidad de ranuras a esperar.

Supongamos  $S$  es un conjunto formado por estaciones que colisionaron entre sí. (Puede suceder que ocurran múltiples colisiones consecutivas de estaciones de  $S$ ).

Para manejo de colisiones consecutivas de estaciones de  $S$  hay dos opciones:

1. que el intervalo donde se elige aleatoriamente (una cantidad de ranuras a esperar) sea fijo ó
2. que el intervalo donde se elige aleatoriamente sea de tamaño variable (es decir, que el tamaño cambie con cada nueva colisión de estaciones de  $S$ ).

La ventaja de permitir que el intervalo sea de tamaño variable es que puede acelerar la resolución de la colisión inicial de las estaciones de  $S$ . Para esto, con cada nueva colisión de estaciones de  $S$  se puede agrandar el intervalo donde se elige aleatoriamente.

Para esto se usa el **algoritmo de retroceso exponencial binario** donde:

- Tras la primera colisión cada estación espera de 0 a 1 tiempos de ranura antes de intentarlo de nuevo.
- Si dos estaciones entran en colisión, y ambas escogen el mismo número aleatorio, habrá una nueva colisión.
- Después de la segunda colisión cada una escoge 0,1,2 o 3 al azar y espera ese número de tiempos de ranura.
- Si ocurre una tercera colisión, entonces para la siguiente vez el número de ranuras a esperar se escogerá al azar en el intervalo 0 a 7.
- Tras  $i$  colisiones se escoge un número aleatorio entre 0 y  $(2^i - 1)$  y se salta ese número de ranuras.
- Tras haberse alcanzado 10 colisiones el intervalo de aleatorización se congela en un máximo de 1023 ranuras.
- Tras 16 colisiones el controlador tira la toalla y avisa de un fracaso a la computadora La recuperación posterior es responsabilidad de las capas superiores.

## Evaluación

- El algoritmo asegura un retardo pequeño cuando unas cuantas estaciones entran en colisión.
- El algoritmo asegura que la colisión se resuelva en un intervalo razonable cuando hay colisiones entre muchas estaciones.

## Ethernet Rápida:

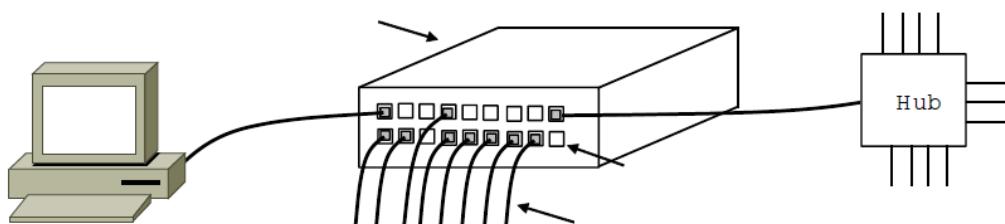
Debido al incremento de la capacidad de almacenamiento y en el poder de procesamiento, las PC actuales pueden manejar gráficos de gran calidad y aplicaciones multimedia complejas.

Las aplicaciones de red que envían ficheros con multimedia tienen el problema de que cuando estos ficheros son compartidos en una red, las transferencias de un cliente a otro producen un gran uso de los recursos de la red. A 10 Mbps, pueden ocurrir grandes demoras cuando se envían ficheros grandes a través de la red. Para evitar esas demoras → mayor velocidad en las redes.

**Fast Ethernet (o 802.3u)** es el nombre de una serie de estándares de IEEE de redes Ethernet de 100 Mbps.

## Cableado de Ethernet Rápida:

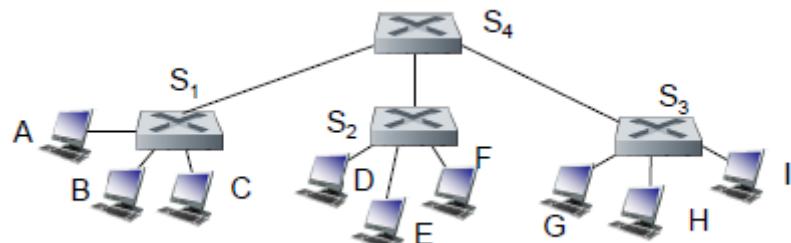
- Cable par trenzado de cobre con (Cat 5 UTP) domina el mercado.
- **100BASE-T (100BASE-TX y 100BASE-T4)**
  - Se usan pares de cobre trenzado
    - 2 tipos de dispositivos de interconexión concentradores y comutadores.
    - Se usan las reglas estándar el formato de las tramas, CSMA/CD y el algoritmo de retroceso exponencial binario.
    - En 100BASE-TX Se usan dos pares de cable trenzado de categoría 5 por estación, uno para enviar y otro para recibir.
- **100BASE-TX**
  - Se usan dos pares de cable trenzado de categoría 5 por estación, uno para enviar y otro para recibir.
  - Los cables pueden manejar velocidades de reloj de 125 MHz.
- **100BASE-FX**
  - 2 líneas de fibra óptica una para recepción (RX) y la otra para transmitir(TX).
  - La distancia entre una estación y el comutador es de hasta 2 km.
  - Los cables 100BaseFX deben conectarse a comutadores.
  - Los concentradores no están permitidos con 100Base-FX.



Los conmutadores pueden estar conectados a computadoras, concentradores y conmutadores.

### Interconectando conmutadores:

Conmutadores pueden conectarse entre sí:



**Q:** Enviando de A a G ¿Cómo hace  $S_1$  para saber cómo enviar una trama destinada a F vía  $S_4$  y  $S_3$ ?

**A:** Autoaprendizaje! (se trabaja exactamente de la misma manera que en el caso de un único conmutador).

→ [Capítulo 5: Redes Inalámbricas](#)

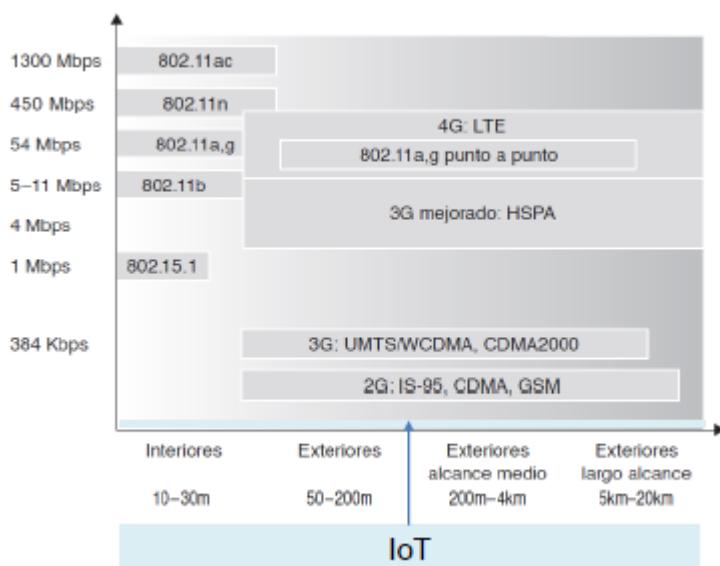
### Redes Inalámbricas y Móviles:

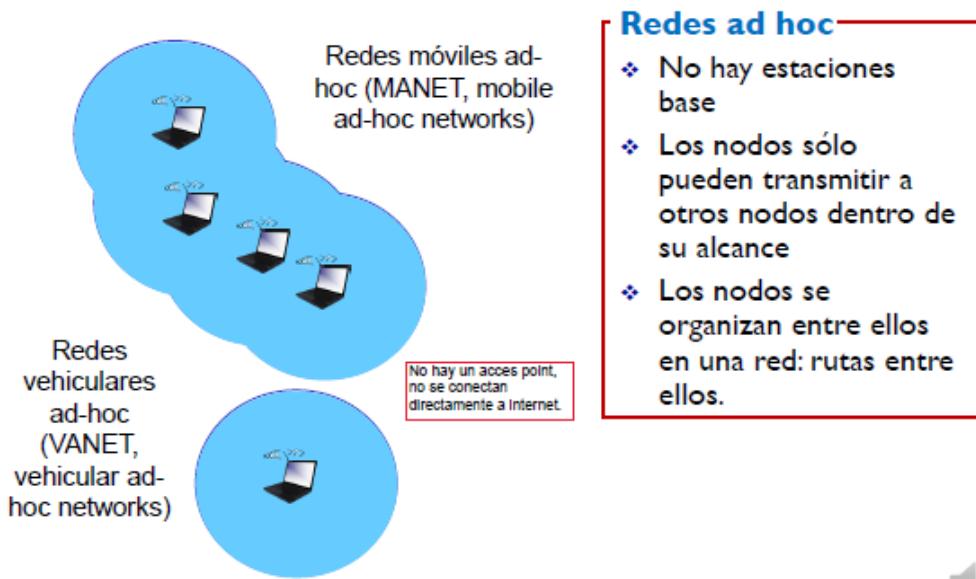
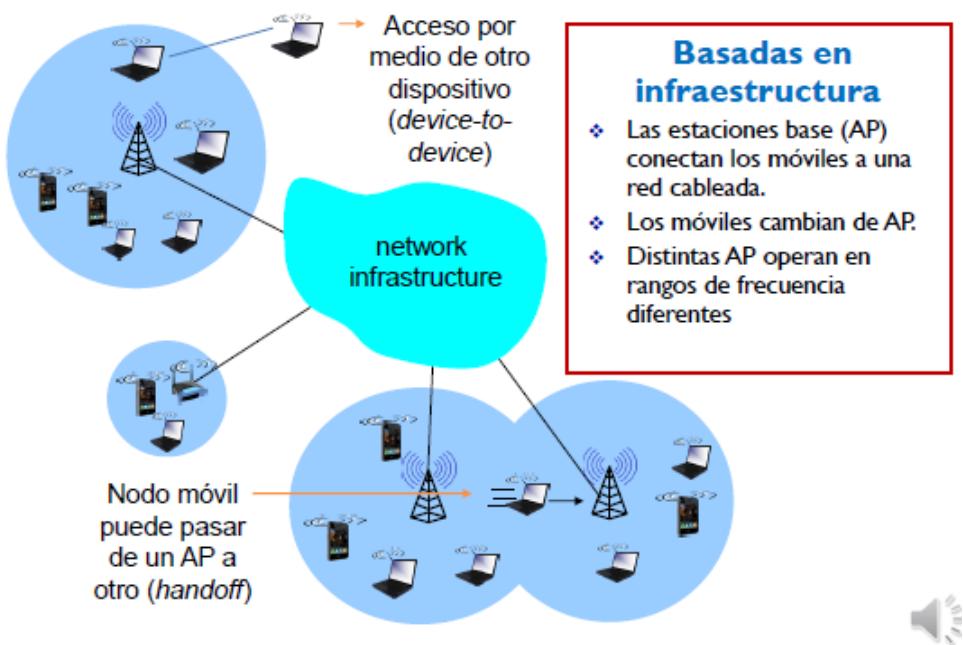
#### Desafíos:

La naturaleza **inalámbrica** del canal.

La **movilidad** de los nodos. (Contacto entre nodos vecinos).

Uso: **acceso** en la frontera de la red.





## Señales:

### Intensidad decreciente de la señal

Dispersión (A más lejos de la antena, menos intensidad en la señal de recepción), atenuación (energía que se pierde en el cable a lo largo del tiempo).

### Interferencias de otros orígenes

Ruido electromagnético, bandas abiertas ISM.

### Propagación multi-camino

Rebotes en objetos.

Mayor tendencia a errores en el bit que las redes cableadas por lo que usan técnicas de detección y recuperación de errores más robustas.

## SNR y BER:

Señal electromagnética (no eléctrica).

### Necesario 1) sensibilidad del receptor (RSSI)

- Que el receptor pueda detectar a señal.

## Necesario 2) relación señal a ruido (SNR)

- Que el receptor pueda entender la señal

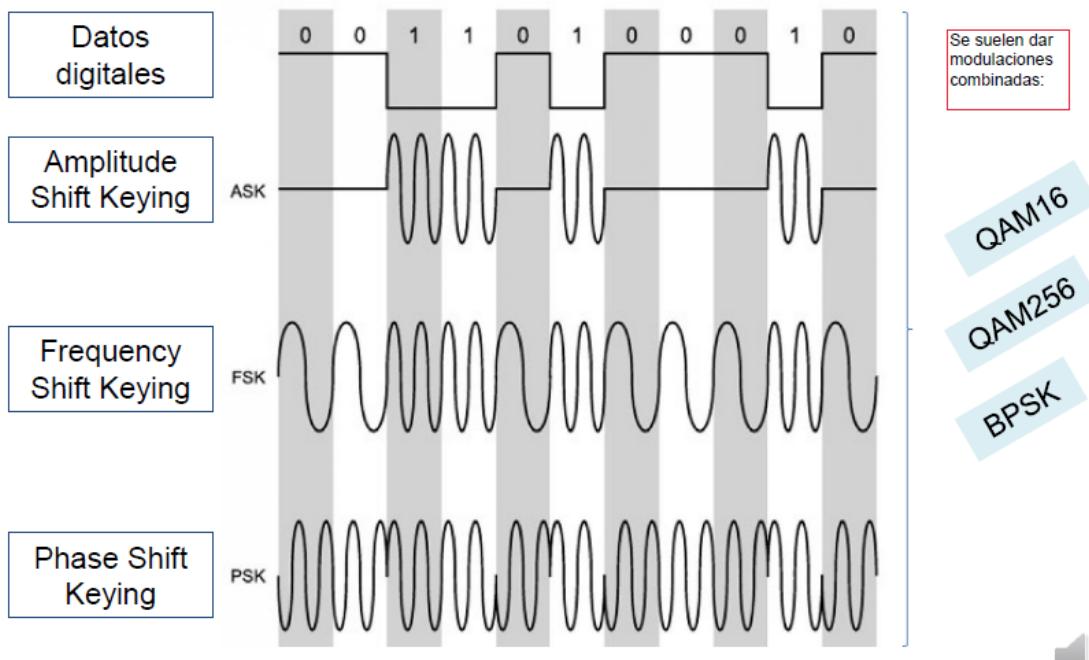
↑ Se miden en dB y dBm (log)

Esquema de Modulación (Propiedades de ondas, no se puede cargar 1 y 0 → se deben traducir mediante modulación).

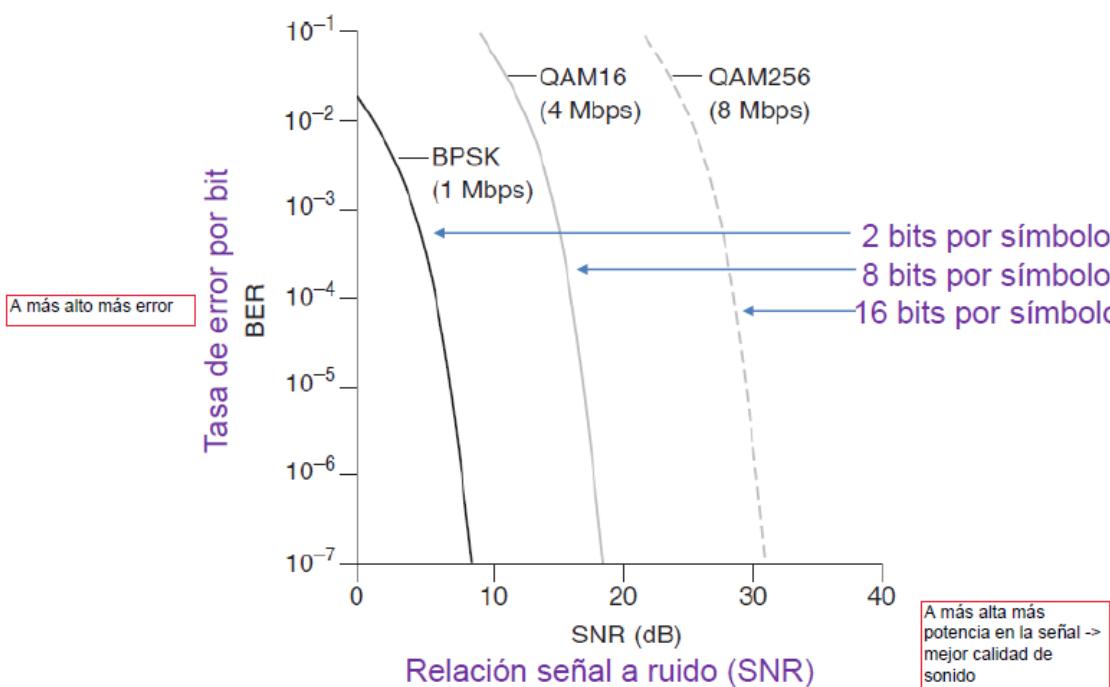
### Tasa de error (BER)

- Cada cuantos bits tengo un bit errado, depende el ruido y la potencia.

Modulación:



Modulación SNR y BER:



Para un esquema de modulación, cuanto mayor es la SNR, menor BER.

Para una SNR dada, una modulación con tasa de bit más alta tendrá un mayor BER.

Puede utilizarse una selección dinámica del esquema de modulación y de la potencia de transmisión adaptativa para cumplir con un BER objetivo ( WiFi , 4G,...).

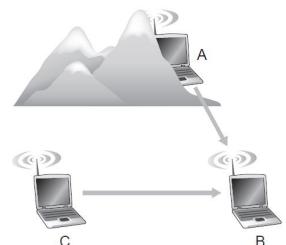
### Problemas de la comunicación inalámbrica:

Los nodos inalámbricos usualmente no pueden transmitir y recibir al mismo tiempo. (Redes cableadas como Ethernet sí pueden).

La potencia generada por el emisor es mucho más alta que lo que probablemente será una señal recibida y así se satura el circuito receptor.

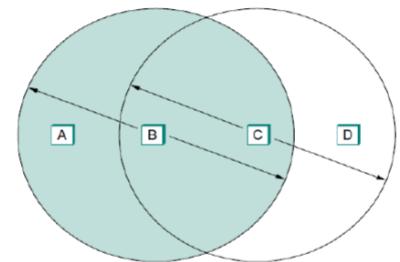
No se puede comparar lo que se transmite con lo que se escucha para detectar colisiones. Las redes inalámbricas no pueden escuchar colisiones, por lo que se busca evitarlas.

Se usa **CSMA/CA** (Acceso Múltiple con Detección de Portadora y Evitación de Colisiones) (WiFi 802.11)



### Problema de la estación expuesta

La estación C transmite a la estación B. Si A detecta el canal no escuchará nada y concluirá erróneamente que ahora puede comenzar a transmitir a B. Si lo hace → **colisión!**



### Problema de la estación expuesta

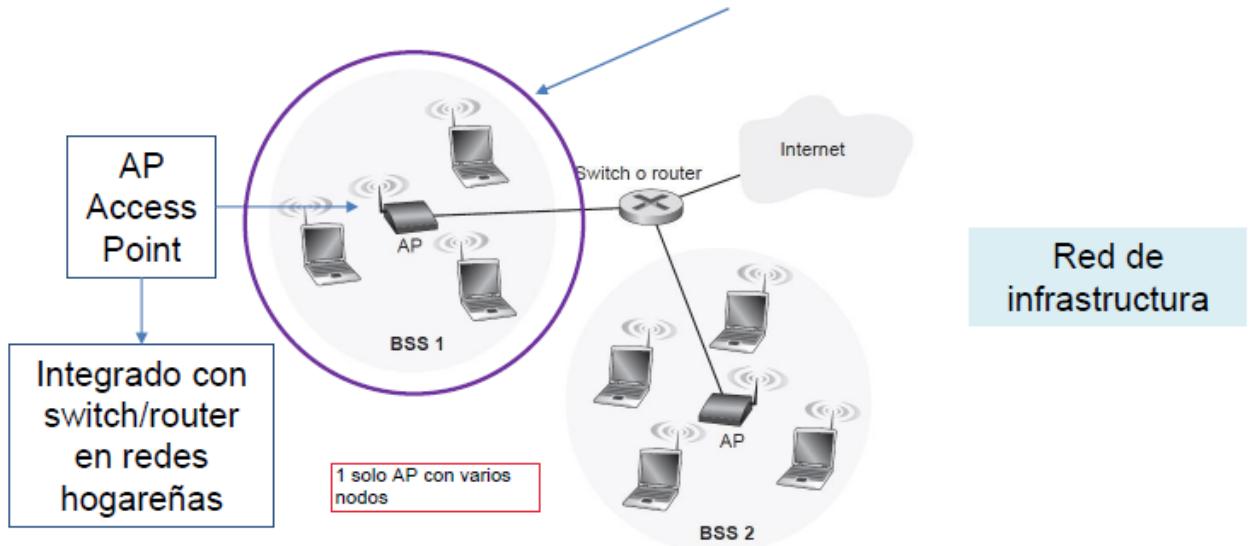
- Supongamos que B transmite a A, y que C desea enviar a D por lo que escucha el canal.
- Cuando C escucha una transmisión concluye erróneamente que no debería transmitir a nadie porque escucha la transmisión de B.
- Pero no hay problema si C transmite a D, porque no va a interferir con la habilidad de A de recibir de B (si puede interferir con A enviando a B, cosa que no pasa en nuestro ejemplo).

### Arquitectura IEEE 802.11: WiFi, varias familias

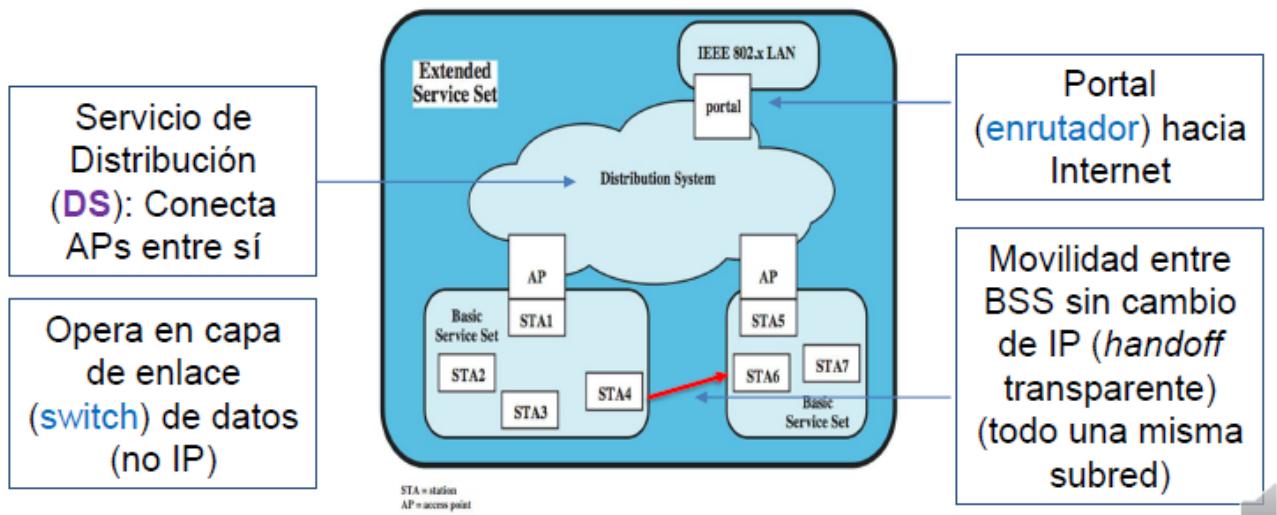
Estándar	Rango de frecuencias	Velocidad de datos
802.11b	2,4 GHz	hasta 11 Mbps
802.11a	5 GHz	hasta 54 Mbps
802.11g	2,4 GHz	hasta 54 Mbps
802.11n	2,5 GHz y 5 GHz	hasta 450 Mbps
802.11ac	5 GHz	hasta 1300 Mbps

- Frecuencias altas requieren mayor potencia de transmisión.
- Todos usan el mismo formato de trama y el control de acceso al medio (MAC o SCAM), cambia la capa física, y hay compatibilidad hacia atrás en todos los casos.

**Componente fundamental:** Conjunto de Servicio Básico (o Basic Service Set, BSS).



**Componente extendido:** Conjunto de Servicio Extendido (o Extended Service Set, ESS)  
Compuesto de 2 o más BSS y un DS.

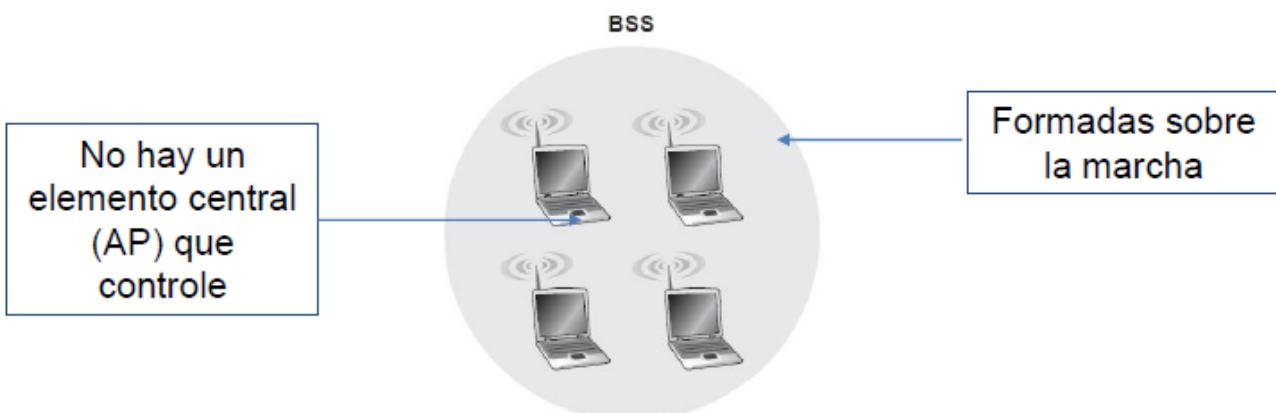


El WiFi integra las BSS dando como resultado las ESS.

**Ad Hoc:**

**IEEE 802.11 sin infraestructura**

Las conexiones se forman de manera dinámica, se puede entrar y salir sin ejecutar ninguna asociación en particular ni algún otro proceso de acceso.



## Canales y Asociación:

Cada host necesita asociarse con un AP antes de poder enviar o recibir datos de la capa de red.

- Puede haber más de 1 AP.
- Se crea un “cable virtual” entre el host y el AP.

AP → Identificador (SSID, Service Set Identifier) (como la red wifi de Juan 2, “FRAIRE”)

AP → 11 canales parcialmente solapados.

Depende del estándar: i.e., 85 Mhz dentro de la banda 2,4 GHz a 2,4835 GHz.

El estándar 802.11 no especifica un algoritmo para elegir con cual de las AP disponibles (re)asociarse.

**Situación 1:** Un nodo puede no estar asociado a ninguna AP y necesita asociarse a alguna

**Situación 2:** Un nodo puede pasar a estar insatisfecho con su AP y quiere cambiar

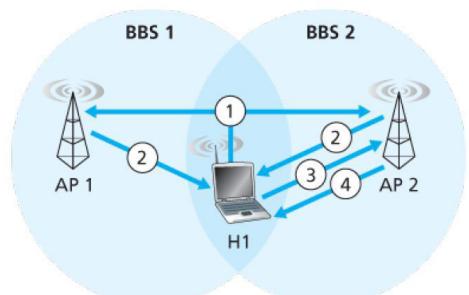
- Calidad de la señal del AP actual insuficiente (RSSI o SNR)
- Red muy cargada tráfico
- Otros

**Escaneo Activo:** iniciado por el host

1. El nodo manda una trama de prueba.
2. Los AP al alcance responden con una trama de respuesta.
3. El nodo elige el AP y envía trama de pedido de asociación.
4. El AP responde con una trama de respuesta de asociación.

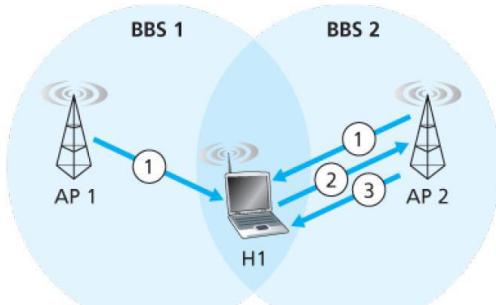
Re asociación : el nuevo AP notifica al AP anterior del cambio.

Si estos AP son partes del mismo sistema extendido hay un protocolo de comunicación entre APs que permite notificar que un nodo se ha dado de baja en uno y se ha dado de alta en otro.



**Escaneo Pasivo:** iniciado por el AP

1. AP difunde una trama guía periódicamente
  - Capacidades (i.e., tasas de transmisión) e identificador del AP, la hora, cuánto falta para la próxima trama guía, etc.
2. El nodo elige el AP y envía trama de pedido de asociación.
3. El AP responde con una trama de respuesta de asociación.



**Otras responsabilidades de AP:**

- Asociación : establece asociación inicial entre nodo y AP.
- Reasociación : para transferir una asociación a otro AP ( handoff)
- Desasociación : notificación por nodo o AP que una asociación existente terminó.

**Subcapa MAC 802.11:**

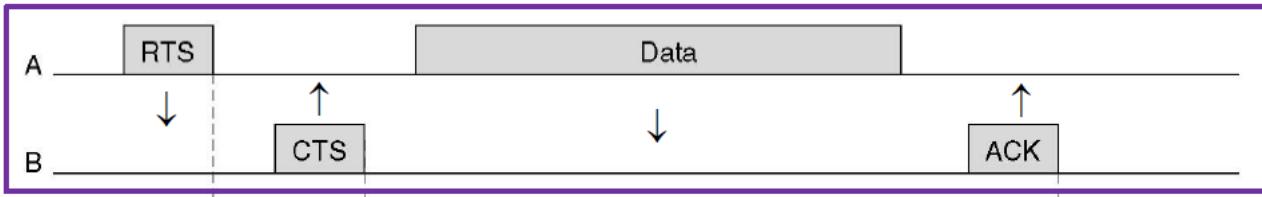
Escucha si hay alguien transmitiendo en el momento. Usa CSMA/CA.

**Soporta dos modos**

- DCF función de coordinación distribuida.→ para redes ad hoc
- PCF función de coordinación puntual → para redes basadas en infraestructura (AP)

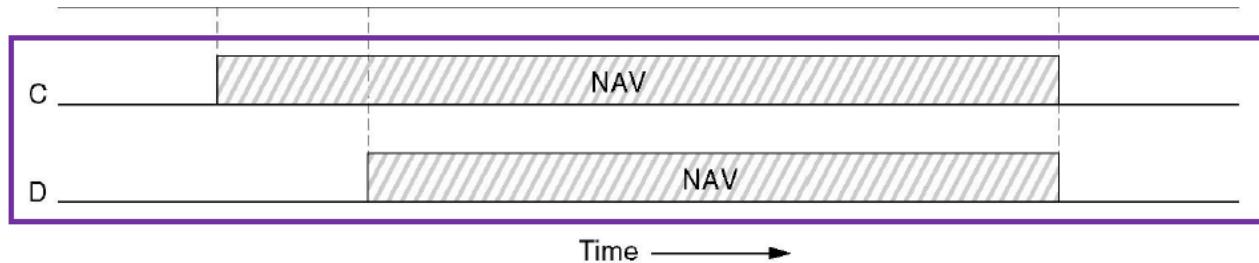
**DCF: función de coordinación distribuida (para redes ad-hoc)**

En ethernet no existe RTS ni CTS ya que no necesita evitar la colisión.



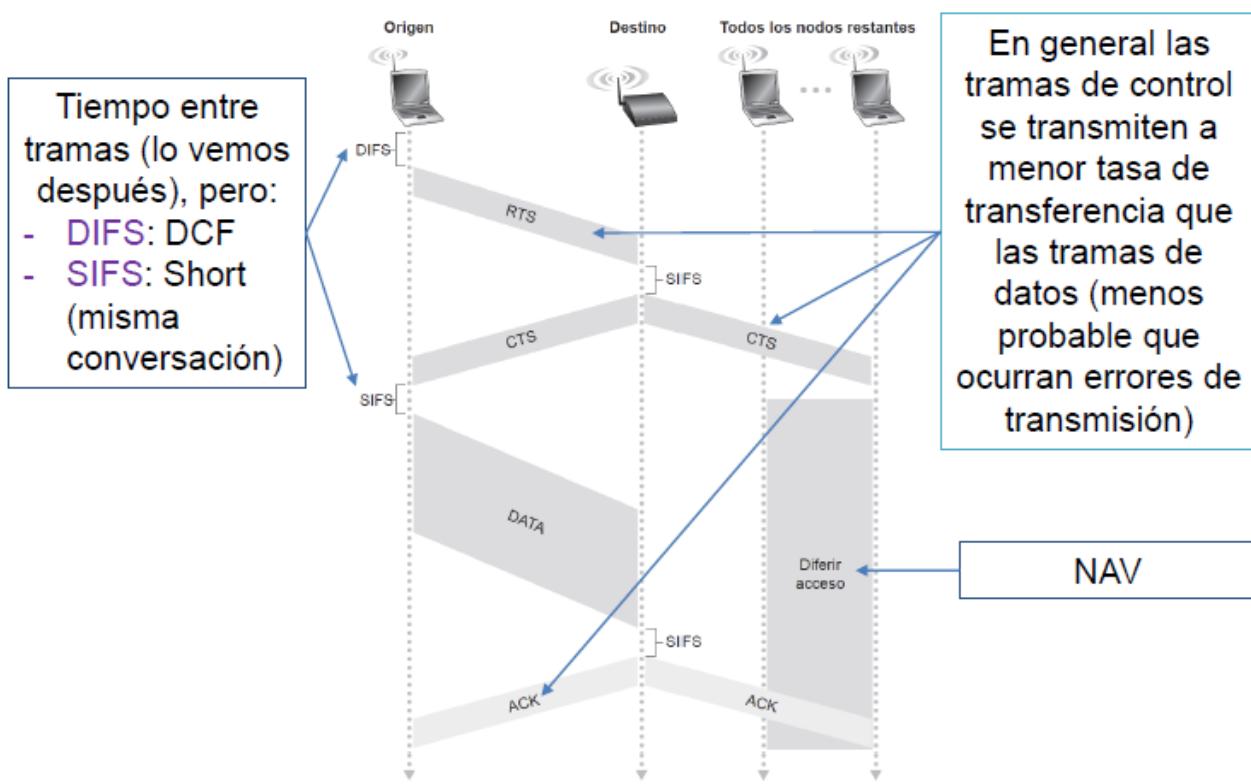
A desea enviar a B. C es una estación que está dentro del alcance de A y D está dentro del alcance de B pero no dentro del de A. Entonces:

1. **A:** envía una trama **RTS** a B (permiso para enviarle una trama).
2. **B:** recibe esta solicitud, y decide otorgarle el permiso, envía una trama **CTS**.
3. **A:** recibe **CTS** y envía su trama. Comienza su temporizador de **ACK**. Si termina antes de que el **ACK** regrese, todo el protocolo se ejecuta de nuevo
4. **B:** al recibir correctamente la trama, responde con una trama de **ACK**.



Comportamiento de los hosts C y D:

1. **C:** recibe la trama **RTS** y desiste de transmitir hasta que el intercambio esté completo. Con la información en **RTS**, C estima cuánto tardará la secuencia, incluyendo el ACK final, e inicia un NAV (vector de asignación de red).
2. **D:** escucha el **CTS** y también impone un canal NAV para sí misma.



**Colisiones:** Dos nodos pueden enviar RTS simultáneamente.

Emisores asumen una colisión porque no reciben el CTS luego de un cierto intervalo de tiempo.

Emisores esperan una cantidad de tiempo (algoritmo de retroceso exponencial binario) e intentan de nuevo.

→ En canales con mucha latencia pueden existir otro tipo de colisiones.

**Estación oculta:** CTS es escuchado por una estación oculta (establece el NAV)

No envía nada (tiempo incluido en el RTS y CTS).

Luego de ese tiempo más un pequeño intervalo el canal puede ser asumido disponible otra vez y otro nodo es libre de intentar enviar.

**PCF: Función de coordinación puntual (redes basadas en infraestructura)**

Es opcional usarlo, tiene que haber un momento donde los nodos puedan "hablar" DCF ya que no todos "hablan" PCF.

**Hay una relación uno a uno entre hosts y AP (asociación) → BSS**

**AP:** responsable de enviar y recibir datos (i.e. paquetes) desde y hacia hosts asociados con el AP

**AP:** responsable para coordinar la transmisión de varios hosts inalámbricos asociados.

**AP:** sondea los nodos preguntándoles si tienen tramas para enviar

**AP:** → no ocurren colisiones

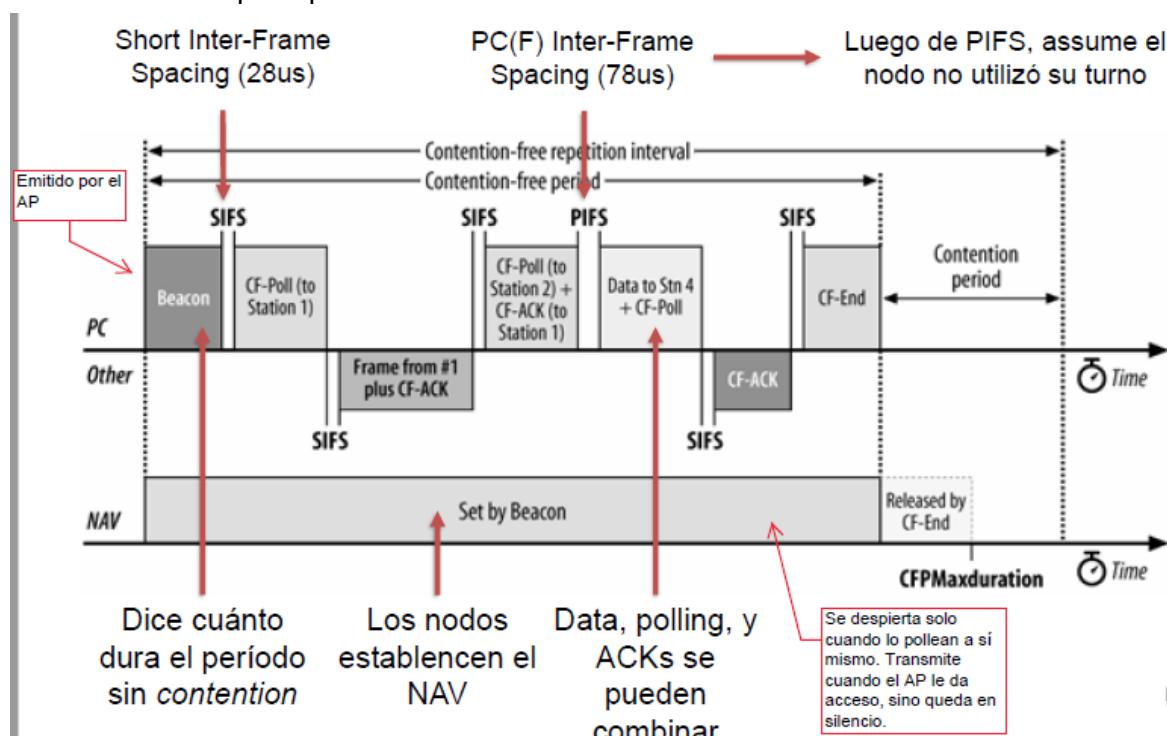
El tiempo en el medio se divide en:

**Período sin (contention) disputa (PCF)**

- Implementada en AP, quien coordina el acceso al medio.
- Nodos transmiten sólo si lo pide el AP.
- El AP tiene una lista de nodos "privilegiados"
- Los nodos se registran para estar en la lista.

**Período en (contention) disputa (DCF)**

- Implementado en los nodos
- Los nodos compiten por el medio



**Beacon (baliza):** el AP demarca el inicio de la trama.

- Contiene información de cuánto esperar para el próximo.
- Ese es el tiempo de un NAV, dentro del cual ocurren diálogos dentro del PCF.

**Poll (sondeo):** el AP pide a la estación que transmita.

- Cuando esa estación termina de transmitir, termina su turno y el derecho a transmitir pasa a la siguiente estación.
- Ni el orden ni la frecuencia son especificados en el estándar.

**Tiempos Entre Tramas:**

**SIFS (Short Inter Frame Space) de 28 us**

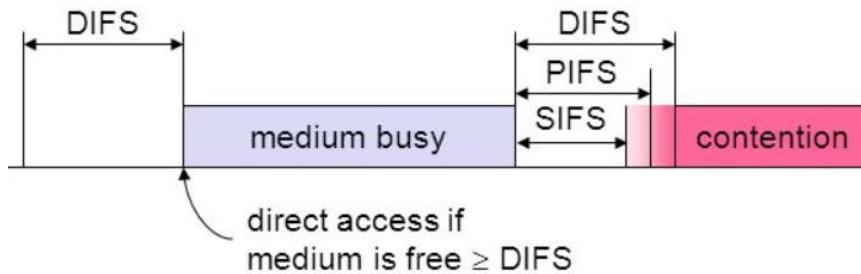
- Intervalo entre tramas en un mismo diálogo (ACK, CTS, datos).

**PIFS (Point Coordination IFS) de 78 us**

- Intervalo entre tramas asumido por el AP (PCF).

**DIFS ( Distributed IFS) de 128 us**

- Intervalo entre tramas asumido por nodos (DCF).



**SIFS:**

Dentro de un diálogo se usan intervalos SIFS

- Hacen falta los SIFS para cosas como calcular suma de verificación, entrampado, de la próxima trama.
- Una estación usando SIFS para determinar la oportunidad de transmisión tiene la prioridad más alta.
- Hay solo una estación que puede responder luego de un intervalo SIFS, nodo el cual está transmitiendo o recibiendo en ese momento.

**PIFS:**

Entre dos diálogos diferentes se usa un PIFS (dentro de PCF).

- Luego de un PIFS el AP puede mandar una trama beacon o poll.
- Dentro de un PIFS se impide el uso de DCF.

El AP puede hacer sondeo en forma *round-robin* a todas las estaciones:

- Cuando se emite un sondeo, la estación responde usando un SIFS.
- Si el AP recibe una respuesta a un poll, puede hacer otro poll usando PIFS. Si no se recibe respuesta al poll, el AP puede hacer otro poll.

**DIFS:**

- Luego de un período de PCF, viene un DCF (con CSMA/CA), cuyas conversaciones se rigen por un DIFS.
- Si el AP no tiene nada que decir y ocurre un tiempo DIFS, cualquier estación puede intentar adquirir el canal.

## Capa Física

Esta se encarga de transportar un stream de datos de una máquina a otra usando medios físicos como lo son:

- Bit, enlace físico, par trenzado (TP), cable coaxial, cable de fibra óptica, radio link types, wide-area, satellite.
  - En los medios físicos viajan señales.
- Los medios físicos **se conectan entre sí** usando dispositivos como codecs, modems, multiplexores, etc. Formándose así **redes complejas de distintos tipos**

## Señales Digitales y Analógicas

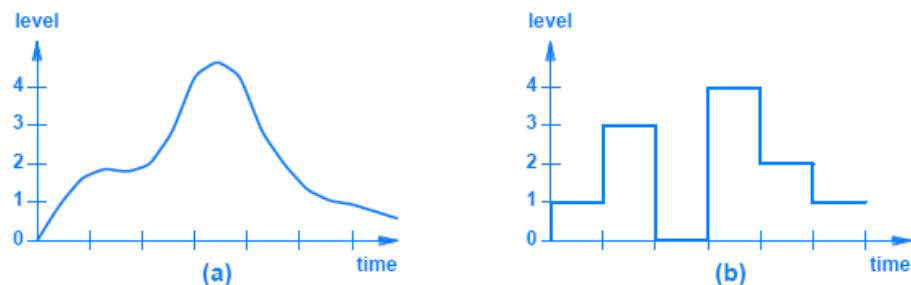


Figure 6.1 Illustration of (a) an analog signal, and (b) a digital signal.

Tipos de información de la comunicación de datos en la capa física

- **Señales analógicas:** Caracterizadas por función matemática continua
- **Señales digitales:** Con conjunto fijo de niveles válidos

Representación de señales como funciones del tiempo

### ★ Señales digitales vs señales analógicas

- Las señales digitales generalmente son más baratas que las señales analógicas y son menos susceptibles a interferencias de ruidos
- Las señales digitales sufren más de **atenuación** (reducción de fuerza de la señal) que las señales analógicas
  - A frecuencias mayores los pulsos se tornan más redondeados y pequeños
  - Esta attenuación puede llevar rápidamente a la pérdida de información contenida en la señal

## Ondas Sinusoidales

Son producidas por fenómenos naturales, por ejemplo, los tonos audibles suelen ser ondas sinusoidales

- *Onda sinusoidal*

- $s(t) = A \sin(\pi f t + \phi)$ ,  $t$  número real
- Propiedades
  - Frecuencia = número de oscilaciones por segundo  $f = 1/T$  (cantidad de ciclos por segundo)
  - Amplitud = diferencia entre las alturas máxima y mínima
  - Fase = cuánto es desplazado el comienzo de la onda sinusoidal a partir de un tiempo de referencia
- El período ( $T$ ) = tiempo requerido por un ciclo

Los sistemas de comunicación usan altas frecuencias (expresadas en millones de ciclos por segundo - megahertz (MHz))

### Señales Compuestas

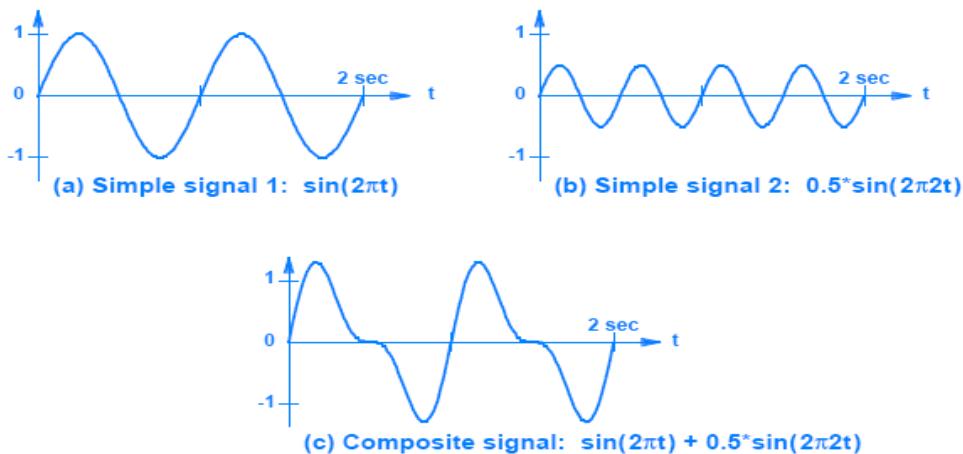


Figure 6.5 Illustration of a composite signal formed from two simple signals.

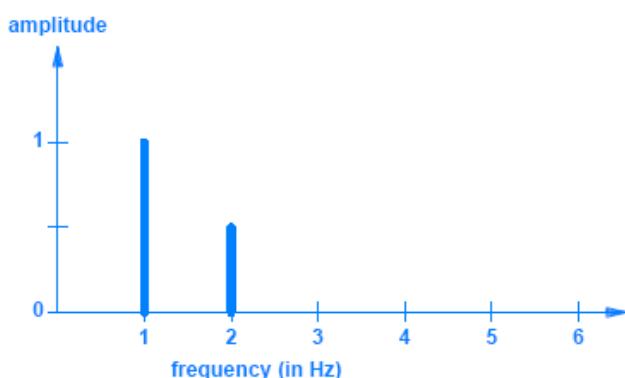
- ❖ Señales simples:
  - Una onda sinusoidal
- ❖ Señales compuestas:
  - Puede descomponerse en un conjunto de ondas sinusoidales simples
  - Una señal electromagnética va a ser compuesta; además va a ser hecha de varias frecuencias
- ❖ Descubrimiento de Fourier
  - Toda señal es hecha a partir de un conjunto de funciones sinusoidales (cada una con frecuencia, amplitud y fase)
- ❖ Señales periódicas:
  - $s(t+T) = s(t)$  para todo  $-\infty < t < \infty$ .
- ❖ Señales aperiódicas (también no periódicas)
- ❖ Si una señal compuesta es periódica, entonces las partes constitutivas son también periódicas
  - La mayoría de los sistemas usan señales compuestas para transportar información

- Una señal compuesta es creada en uno de los extremos y el receptor descompone la señal en sus componentes simples

### Representaciones Gráficas De Las Señales

- *Representación de dominio de tiempo*
  - Grafo de una señal como función del tiempo
- *Representación de dominio de frecuencia*
  - Grafo de dominio de frecuencia
    - Muestra conjunto de ondas sinusoidales simples que constituyen la función compuesta
    - $A \sin(2\pi ft)$  es representada por una línea simple de altura  $A$  que se posiciona en  $x = f$

### Representación De Dominio De Frecuencias



**Figure 6.6** Representation of  $\sin(2\pi t)$  and  $0.5\sin(2\pi 2t)$  in the frequency domain.

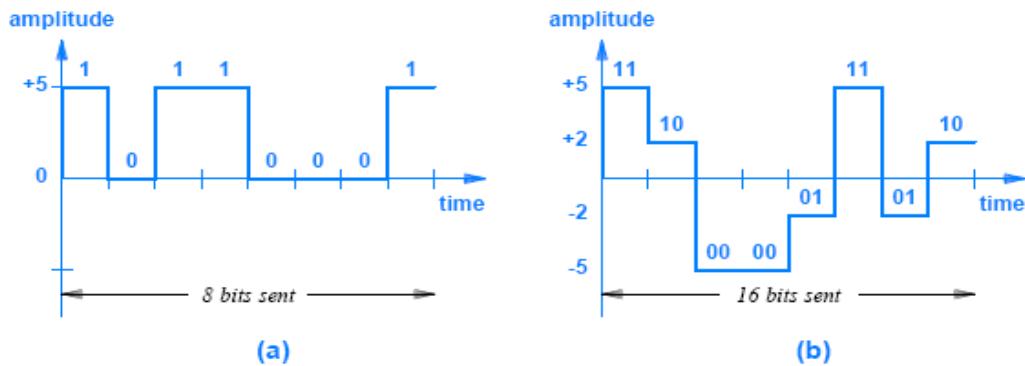
La representación de dominio de frecuencia es muy compacta

- ❖ El espectro de una señal = rango de frecuencias que contiene
  - Es el intervalo desde la frecuencia más chica a la frecuencia más grande
- ❖ El ancho de banda analógica = ancho del espectro
  - Diferencia entre las frecuencias más altas y la más bajas

### Señales Digitales

Las señales digitales usan voltajes para representar valores digitales

- Mecanismos de transmisión físicos usan dos o más niveles de voltaje para enviar señales digitales
  - Cada nivel representa un número binario
- Usar  $2^n$  niveles para representar número de  $n$  bits.



**Figure 6.8** (a) A digital signal using two levels, and (b) a digital signal using four levels.

- (a) Un voltaje positivo corresponde al **uno lógico** y un voltaje cero corresponde al **cero lógico**
- (b) 4 niveles de voltaje: -5 V, -2 V, +2 V, +5 V

### Baudios y Bits por Segundo

- La respuesta depende de los siguientes factores
  - Del número de niveles de señal
  - De la cantidad de tiempo que el sistema permanece en un nivel dado antes de moverse al siguiente
- El hardware coloca límites en cuán corto el tiempo en un nivel debe ser
  - Si la señal no permanece en un nivel por suficiente tiempo, el hardware receptor va a fallar en detectarlo
  - La cantidad de veces que una señal puede cambiar por segundo se mide en **baudios**
- baud y número de niveles de señal controlan la tasa de bits
- Relación entre baudios, niveles de señal y tasa de bits
  - **bits por segundo = N° baudios \* [log2(niveles)]**

### La Tasa De Datos Máxima De Un Canal

En algunos casos se introduce un **filtro** en el circuito para limitar la cantidad de ancho de banda disponible para cada cliente.

### **Métodos para estimar la tasa de datos máxima de un canal**

#### **Teorema de Nyquist**

Nyquist probó que si se pasa una señal a través de un filtro pasa-bajas de ancho de banda  $H$ , la señal filtrada se puede reconstruir por completo tomando solo  $2H$  muestras por sec.

- No tiene sentido muestrear la línea a una rapidez mayor porque las componentes de mayor frecuencia que tal muestreo puede recuperar se han filtrado
- Si la señal consiste de  $V$  niveles de voltaje, el **teorema de Nyquist** (1924) establece:
  - **Tasa de datos máxima =  $2H \log_2 V$  bps**

La **cantidad de ruido térmico** se mide por la relación entre la potencia del ruido, llamada **radiación señal a ruido**.

Si indicamos la potencia de una señal con S y la potencia del ruido con una N, la **relación señal a ruido** es S/N

La relación misma no se expresa; en su lugar se da la cantidad  $10 \log_{10} S/N$ . Estas unidades se conocen como **decibeles (dB)**.

### **Resultado de Shannon (1948)**

- La tasa de datos máxima de un canal ruidoso cuyo ancho de banda es H Hz y cuya relación señal a ruido es S/N, está dada por:
  - **Nº máximo de bps =  $H \log_2(1+S/N)$**
- La fórmula solo da un límite superior y los sistemas reales rara vez lo alcanzan

### **Cálculo de niveles distinguibles de voltaje que valen la pena**

- Suponemos que conocemos la relación señal a ruido S/N.
- La cantidad de niveles de voltaje permitidos depende de S/N.

◆ Igualando Nyquist y Shannon:  $V = (1+S/N)^{1/2}$

## **Modems, portadora, modulación**

### **Módem:**

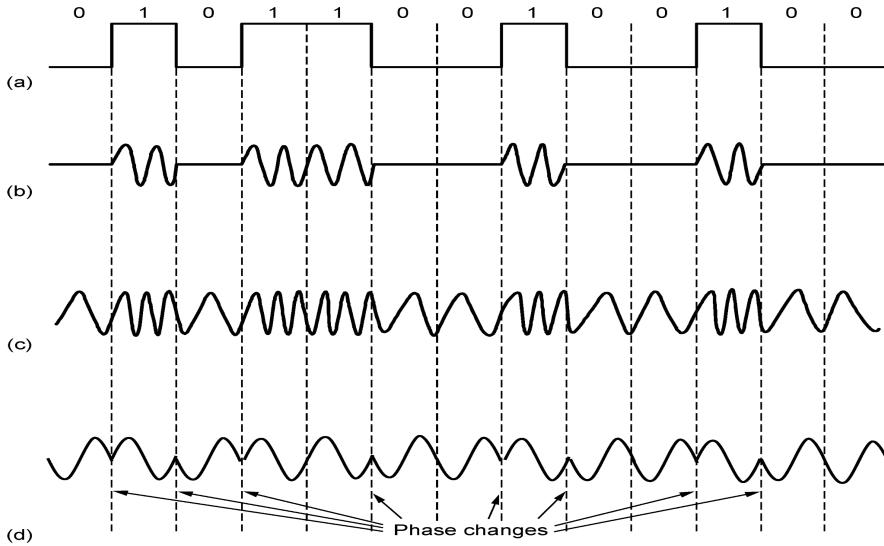
Un módem permite **convertir** señales digitales en analógicas y recíprocamente. Todos los módems modernos transmiten tráfico en ambas direcciones al mismo tiempo (mediante el uso de frecuencias distintas para las diferentes direcciones).

Muchos sistemas de comunicación de larga distancia usan una **portadora (carrier)** de orden sinusoidal. Los sistemas hacen pequeños cambios a la portadora para representar información siendo enviada.

### **Modulación:**

El envíador debe cambiar una de las características de la onda: amplitud, frecuencia, desplazamiento de fase.

### **Tipos de modulación**

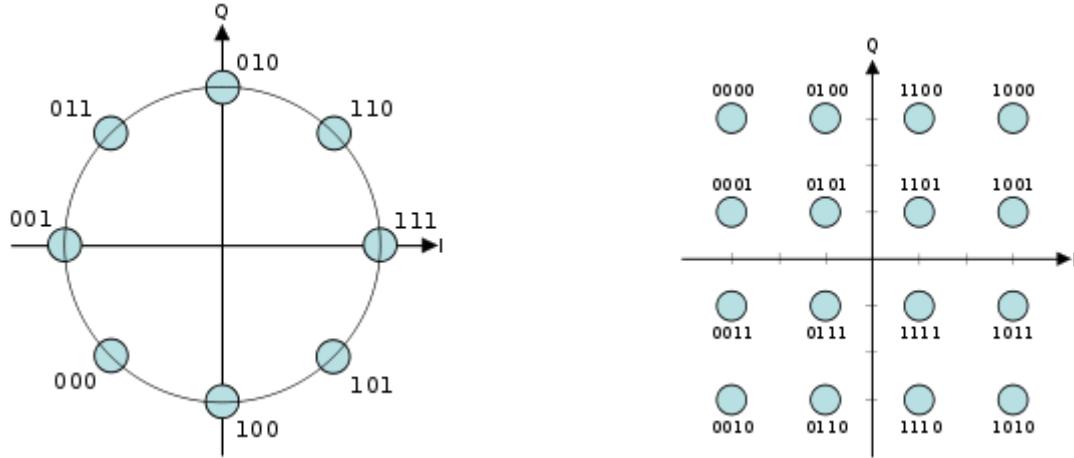


- *Portadora de onda senoidal* = tono continuo en el rango de 1000 a 2000 HZ
- *Modulación de amplitud (b)*
  - Se usan dos niveles diferentes de amplitud para representar 0 y 1
- *Modulación de frecuencia (c)*
  - Se usan dos o más tonos diferentes
  - Si la señal es más fuerte, la frecuencia del carrier aumenta y si la señal es más débil, la frecuencia del carrier disminuye
  - Es más difícil de visualizar
- *Desplazamiento de fase (DF)*
  - Es posible usar cambios en la fase para representar una señal
  - El DF se mide por el ángulo de cambio
- *Modulación de fase (d)*
  - Al requerir el DF al final de cada intervalo, se facilita que el receptor reconozca los límites de los intervalos

### Detección de cambio de fase

- Un **receptor** puede medir la cantidad portadora desplazada durante un *DF*
  - Sistema que reconoce un conjunto de *DF* y usa cada *DF* para representar valores de datos específicos.
- Usualmente los sistemas están diseñados para usar  $2^n$  *DF*, así un emisor puede usar bits de datos para elegir entre los *DF*.

## Diagramas De Constelación



Es un método de representación en el plano complejo de los estados de símbolo en términos de amplitud y fase en los esquemas de modulación digitales tales como QAM o PSK

El eje horizontal se refiere a los componentes de símbolos que están en fase con la señal portadora y el eje vertical a los componentes en cuadratura ( $90^\circ$ )

Los diagramas de constelación también pueden usarse para reconocer el tipo de interferencia y distorsión en una señal

Los ejes del plano del diagrama suelen ser llamados "I" (en fase) y "Q" (en cuadratura). En la constelación se representa la relación de amplitud y fase de una portadora modulada digitalmente y por lo tanto, el módulo y la fase de cada una de las posibles señales que conforman la modulación.

El módulo viene dado por la distancia entre el origen de las coordenadas y el punto. La fase es el ángulo que una línea que une al origen con el punto hace con el eje horizontal.

### **Interpretación del diagrama**

Al recibir la señal, el demodulador examina el símbolo recibido, que puede haber sido afectado por el canal o el receptor debido al ruido blanco aditivo gaussiano, distorsión, ruido de fase o de interferencia.

como estimación de lo que se transmitió realmente se selecciona el punto en el que el diagrama de constelación que está más cerca del símbolo recibido. Por lo tanto, demodulará incorrectamente si la corrupción de la señal ha hecho que el símbolo recibido se acerque a otro punto de la constelación diferente del símbolo emitido. Esta detección es la máxima probabilidad.

El diagrama de constelación permite una visualización directa de este proceso, ya que puede suponerse el símbolo recibido como un punto arbitrario en el plano IQ y luego decidirse cual es el punto de la constelación más cercano a él.

Con el propósito de analizar la calidad de la señal recibida, algunos tipos de corrupción son evidentes en los diagramas de constelación:

- **Ruido gaussiano** que muestra puntos de la constelación como difusos.

- **Interferencia de frecuencia única no coherente**, que muestra puntos de la constelación como círculos.
- **Ruido de fase** que muestra puntos de la constelación dispersos en forma rotacional.
- **Atenuación** que hace que los puntos de la esquina del diagrama se muevan hacia el centro.

## **Multiplexado**

Cuando **queremos** ver desde el punto de vista económico, es mucho más conveniente usar un solo cable para transportar varias señales que instalar un cable para cada señal, y queremos que los canales de comunicación puedan ser compartidos por múltiples señales...

**¿Cómo hacer para poner muchas señales en un mismo canal?** → Usar multiplexores y demultiplexores

- **Multiplexado:** Un canal transportar varias señales
- **Multiplexor:** Mecanismo que implementa el concepto anterior
- **Demultiplexado:** Separar la combinación de señales constitutivas
- **Demultiplexor:** Mecanismo que implementa el concepto anterior

Estos esquemas de multiplexado se pueden dividir en:

Tenemos varios circuitos analógicos, cada uno con su señal analógica y queremos colocar todas estas señales analógicas en un mismo canal.

**¿Cómo hacer para multiplexar y demultiplexar un conjunto de señales analógicas?**

- **FDM (multiplexado por división de frecuencia):** El espectro de frecuencias se divide en bandas de frecuencia y cada usuario posee exclusivamente alguna banda.
  - **Funcionamiento de un multiplexor en FDM**
    - Primero se eleva la frecuencia de los canales de voz, cada uno en una cantidad diferente
    - Después de lo cual se pueden combinar, porque en ese momento no hay dos canales que ocupen la misma porción del espectro.
  - **Funcionamiento de un demultiplexor**
    - Se usan filtros para recuperar las señales originales
  - **Utilidad de FDM**
    - FDM aún se usa sobre cables o canales de microondas, requiere circuitos analógicos

Tenemos un conjunto de señales digitales y queremos enviar todas esas señales por un mismo canal.

**¿Cómo hacer para multiplexar y demultiplexar un conjunto de señales digitales?**

**TDM (multiplexado por división de tiempo):** Los usuarios esperan su turno (en round-robin), y cada uno obtiene en forma periódica toda la banda durante un breve lapso de tiempo.

Los bits de cada una de las señales de entrada son tomados en una **ranura fija de tiempo** y enviados a la señal agregada de salida.

TDM puede manejarse por completo mediante dispositivos digitales y por ello es popular

**Aplicación de TDM**

- TDM es ampliamente usado como parte de las redes de teléfonos y redes de celulares.

## **Principio de Superposición de Ondas**

*Propiedades físicas de la interferencia:*

Si dos señales en un punto están en fase se agregan para sumar sus amplitudes, pero si están fuera de fase, se restan para dar una señal que es la diferencia de las amplitudes.

Tenemos varios circuitos analógicos, cada uno con su señal analógica, queremos colocar todas esas señales analógicas en un mismo canal

### **¿Cómo hacer para multiplexar y demultiplexar un conjunto de señales analógicas?**

**CDM (multiplexado por división de código):** Permite varias señales de diferentes usuarios, compartir la misma banda de frecuencias.

- ❖ Varios usuarios pueden coexistir y transmitir simultáneamente con interferencia mínima
- ❖ Un CDM a menudo se lo llama CDMA (Code Division Multiple Access)
- ❖ En CDMA las tramas que colisionan no son distorsionadas; en cambio, se agregan múltiples señales en forma lineal
  - Esto es debido al principio de superposición de ondas
- ❖ En CDMA cada tiempo de bit se subdivide en  $m$  intervalos cortos llamados **chips**
  - Hay 64 o 128 chips por bit
- ❖ A cada estación se le asigna un código único de  $m$  bits llamado secuencia de chips

*Notación bipolar:* el 0 en binario es -1 y el 1 en binario es +1

- Usamos la notación bipolar para la secuencia de chips y mostraremos la secuencia de chips entre paréntesis

*Transmisión en un tiempo de bit*

- Una estación puede transmitir un 1 enviando su secuencia de chips en bipolar,
- Puede transmitir un 0 enviando su negativo de su secuencia de chips
  - i.e. se cambia el signo de cada componente de su secuencia de chips en bipolar
- Puede quedarse en silencio y no transmitir nada

*Requisito*

- Todas las estaciones están **sincronizadas**
  - i.e. todas las secuencias de chips comienzan al mismo tiempo

*Notación*

- El símbolo **S** significa el vector de  $m$  chips para la estación S (en notación bipolar) y **S̄** para su negación (cambiar de signo cada componente de **S**)
- Dos secuencias de chips **S** y **T** son **ortogonales** si y sólo si cumplen:

$$\mathbf{S} \cdot \mathbf{T} \equiv \frac{1}{m} \sum_{i=1}^m S_i T_i = 0$$

o sea, el producto interno normalizado de **S** y **T** es 0

**¿Cómo hacer para que un receptor pueda recuperar la señal enviada por una estación de manera sencilla?** Todas las secuencias de chips deben ser **ortogonales dos a dos**.

*Para recuperar el flujo de bits de una estación, el receptor:*

- Calcula el producto interno normalizado de la secuencia de chips recibida y la secuencia de chips de la estación cuyo flujo de bits se está tratando de recuperar
- Si la secuencia de chips recibida es **S** y el receptor está tratando de escuchar una estación cuya secuencia de chips es **C**, simplemente calcula **S · C**
- Las secuencias ortogonales de chips para las estaciones se pueden generar utilizando un método conocido como **código de walsh**

*Algunas propiedades*

- Si  $\mathbf{S} \cdot \mathbf{T} = 0$ , entonces  $\mathbf{S} \cdot \mathbf{T} = 0$
- El producto normalizado de cualquier secuencia de chips por si mismo es 1.

$$\mathbf{S} \cdot \mathbf{S} = \frac{1}{m} \sum_{i=1}^m S_i S_i = \frac{1}{m} \sum_{i=1}^m S_i^2 = \frac{1}{m} \sum_{i=1}^m (\pm 1)^2 = 1$$

- Además  $\mathbf{S} \cdot \mathbf{S} = -1$

- Idealmente, en un sistema CDMA sin ruido, el número de estaciones que envían concurrentemente puede ser hecho arbitrariamente grande usando secuencias de chip más largas
  - Para  $2^n$  estaciones, códigos de Walsh pueden proveer  $2^n$  secuencias de chip ortogonales de longitud  $2^n$
- *Aplicación de CDM*
  - Además de que en las redes celulares, CDMA es usado por redes satelitales y de cable.
- El incremento de la cantidad de información que se va a enviar de  $b$  bits/seg a  $mb$  chips/seg solo puede realizarse si el ancho de banda disponible **se incrementa** por un factor de  $m$

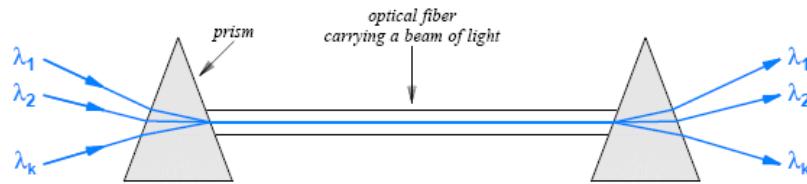
### **OFDM (Orthogonal Frequency Division Multiplexing)**

- El ancho de banda del canal es dividido en varias portadoras que independientemente envían datos (e.g., con QAM)
- Estas portadoras son empaquetadas juntas en el dominio de frecuencias, de modo que las señales de cada portadora se extienden a las adyacentes
- Las portadoras pueden ser muestradas en sus frecuencias del centro sin interferencia de sus vecinos
  - Para hacer este trabajo, un **tiempo guarda** es necesario para repetir una porción de los símbolos de señales en el tiempo de modo que tienen la respuesta de frecuencia deseada
  - Sin embargo, esta sobrecarga es mucho menos que la necesitada para varias bandas que guarda
- La idea de OFDM ha estado disponible por mucho tiempo, pero solo en la última década ha sido adoptada ampliamente
- OFDM es usada en 802.11 y redes de cable
- Usualmente un stream a tasa alta de información digital es dividido en varios streams de tasa baja que son transmitidos con las portadoras en paralelo
  - Esta división es útil porque es más fácil tratar al nivel de la portadora con degradaciones del canal
  - Algunas portadoras pueden ser muy degradadas y excluidas en favor de las portadoras que son recibidas bien

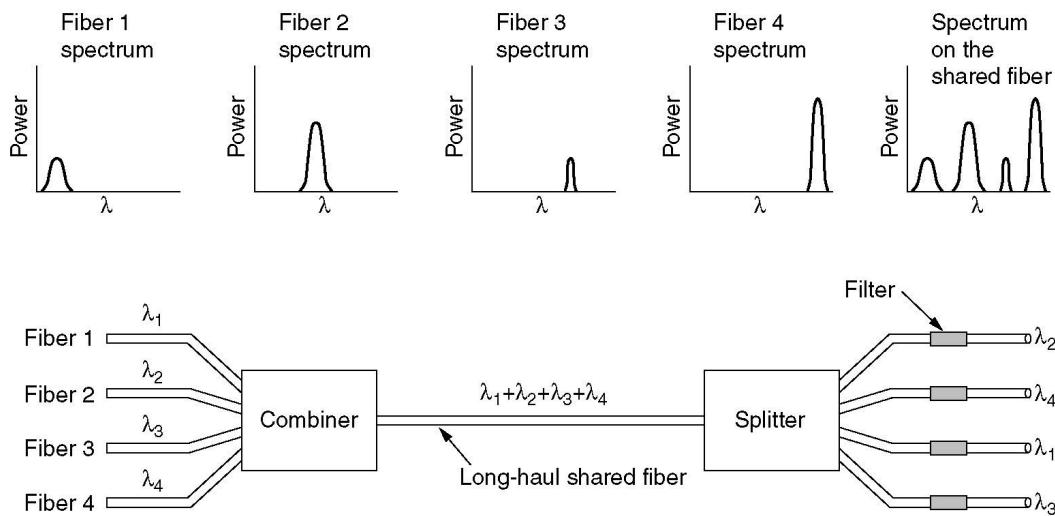
### **WDM (multiplexación por división de longitud de onda):**

- Se refiere a la aplicación de FDM a la fibra óptica
  - Algunas fuentes usan el término **WDM denso** (DWDM) para enfatizar que muchas longitudes de onda de luz pueden ser usadas
- Las entradas y salidas de WDM son **longitudes de onda de luz**
  - Denotadas por la letra griega  $\lambda$ , e informalmente llamados colores
- Cuando la luz pasa a través de un prisma
  - Los colores del espectro son separados
- Si el conjunto de rayos de colores son dirigidos a un prisma en el ángulo correcto
  - El prisma va a *combinar los rayos* para formar un rayo único de luz blanca

- Prismas forman la base del multiplexado y demultiplexado óptico
  - Un multiplexor acepta rayos de luz en varias longitudes de onda y usa un prisma para combinarlos en un rayo único
  - Un demultiplexor usa un prisma para separar las longitudes de onda



**Figure 11.7** Illustration of prisms used to combine and separate wavelengths of light in wavelength division multiplexing technologies.



- Esto se trata de multiplexado por división de frecuencia a frecuencias muy altas
  - Siempre y cuando cada canal tenga su propio rango de frecuencia (es decir, longitud de onda), y todos los intervalos estén separados, se pueden multiplexar juntos en la fibra de largo alcance
  - La única diferencia con respecto a la FDM eléctrica es que un sistema óptico que usa una rejilla de difracción es totalmente pasivo y por ello muy confiable

### Medios De Transmisión

Los medios físicos se clasifican en:

- Medios guiados
  - cable de cobre, fibra óptica
- Medios no guiados
  - radio
- Medios magnéticos
  - DVDs, Blu-ray, cintas magnéticas

**Cable de par trenzado de cobre:** Son dos alambres de aproximadamente 1 mm de grueso se trenzan, logran alcanzar algunos km sin amplificación, para distancias mayores, usar repetidores, pueden usarse para transmitir señales digitales o señales analógicas.

### Evaluación:

- El ancho de banda depende del ancho del cable y la distancia recorrida
  - Varios Mbps pueden ser alcanzados por unos pocos kilómetros
- Debido a su comportamiento adecuado y bajo costo, los cables de par trenzado son **ampliamente utilizados**.

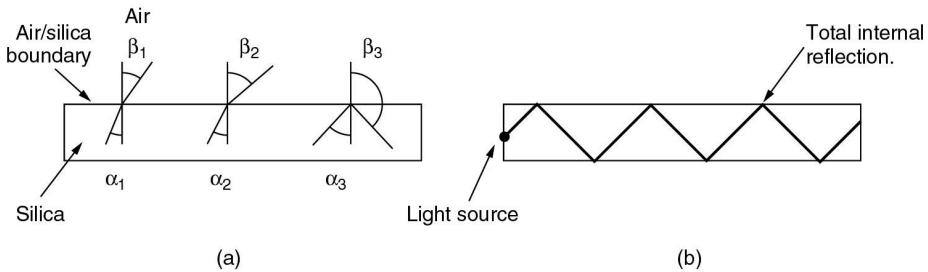
**Cable coaxial:** Con mejor blindaje que los pares trenzados, puede recorrer distancias más largas a mayor velocidad

### Evaluación:

- La construcción y el blindaje le confieren una buena combinación de alto ancho de banda y excelente inmunidad al ruido
- Velocidad de propagación entre 66% y 90% de la velocidad de la luz

**Fibra Óptica:** Con la tecnología de fibra óptica el ancho de banda alcanzable es de 50,000 Gbps (50 Tbps). El límite actual ronda a los 100 Gbps y se debe a nuestra inhabilidad de convertir entre señales eléctricas y ópticas más rápidamente

- Para construir enlaces de mayor capacidad, varios canales son transportados en paralelo sobre una sola fibra



### Física óptica:

- Refracción (a): 3 ejemplos de un rayo de luz desde el interior de una fibra de silicio refractándose en la frontera aire/silicio en ángulos diferentes. El grado de refracción depende de las propiedades de los medios
- (b): Un rayo de luz que incide en un ángulo **mayor o igual al crítico** queda atrapado dentro de la fibra y se puede propagar por varios km sin pérdida
- Varios rayos estarán rebotando con ángulos distintos
  - Se dice que cada rayo tiene un modo diferente, una fibra que tenga esta propiedad se llama **fibra multimodo**
  - En las fibras multimodo el diámetro del núcleo de vidrio es de 50 micras - el grosor de un cabello humano

**Fibra monomodo:** El diámetro de la fibra se reduce a unas cuantas longitudes de onda de luz, la luz se propaga sólo en **línea recta**.

- Son más caras

Se pueden usar en distancias mayores

El grosor del núcleo de vidrio es de 8 a 10 micras

Pueden transmitir datos a 100 Gbps por 100 km sin amplificación

**¿Cómo hacer para aprovechar la fibra óptica si los hosts no usan señales ópticas?**

Usar un sistema de **transmisión óptico** con las siguientes 3 componentes:

1. **Fuentes de luz**
  - a. Un pulso de luz indica un bit 1 y la ausencia de luz indica un bit 0
  - b. Se usan dos clases de fuente de luz para producir las señales
    - i. LED - diodos emisores de luz

- ii. láseres semiconductores
- 2. El medio de transmisión es una fibra de vidrio ultra delgada
- 3. El detector genera un pulso eléctrico cuando la luz incide en él

**Transmisión inalámbrica:** Para los usuarios móviles los medios cableados no son de utilidad, para ellos la comunicación inalámbrica es la respuesta

Ondas electromagnéticas:

- Cuando los electrones se mueven, crean ondas electromagnéticas que se pueden propagar por el espacio libre (aún en el vacío)

Frecuencia  $f$  de una onda electromagnética = cantidad de oscilaciones por segundo (se mide en Hz)

Longitud de onda  $\lambda$  = distancia entre dos puntos máximos o mínimos consecutivos

Principio:

- Al conectarse una antena del tamaño adecuado a un circuito eléctrico, las ondas electromagnéticas pueden ser difundidas de manera eficiente y ser captadas por un receptor a cierta distancia.
- En el vacío, todas las ondas electromagnéticas viajan a la **velocidad de la luz**  $c$  y es de 30cm por nanosegundo.
- En el cobre o la fibra óptica, la velocidad es aproximadamente  $\frac{2}{3}c$  y se vuelve ligeramente dependiente de la frecuencia.
  - La velocidad de la luz es el límite máximo de velocidad.
- **Ley:** La relación entre  $\lambda$ ,  $f$  y  $c$  en el vacío es de  $\lambda \cdot f = c$

### La red telefonía pública conmutada

Para enviar datos digitales de una PC sobre una línea analógica de acceso telefónico, es necesario convertir los datos a formato analógico. → **Módem telefónico:** hace conversación

Los datos se convierten a formato digital en la oficina central de la compañía telefónica para transmitirlos sobre las troncales

- **Codec:** es el dispositivo que hace esa conversación
- En el extremo receptor (del otro lado del troncal) el stream de bits es usado para reconstruir los datos analógicos

Para las troncales de largo alcance, la principal consideración es cómo reunir múltiples señales y enviarlas juntas por la misma fibra óptica → Se usa multiplexado

### **DSL (Digital Subscriber Line):**

Ahora estudiaremos en qué consiste la comunicación de banda ancha (también llamado xDSL)

El sistema telefónico está preparado desde un inicio para trabajar con canales de 4312 Hz cada uno donde se pueden mandar datos o llamadas

**¿Cómo hacer para aprovechar a full un cable de cobre para envío de datos y llamadas con un ancho de banda de 1,1 MHz?**

- DMT (multitonos discreto) divide el espectro de 1,1 MHz en el circuito local en 256 canales de 4312 Hz cada uno
- Multiplexado OFDM es usado
- El canal 0 se usa para el servicio telefónico convencional
- Los canales 1 a 5 no se emplean, para evitar que las señales de voz y de datos interfieran entre sí

- De los 250 canales restantes, 1 se usa para control del flujo ascendente y 1 para el control del flujo descendente
- El resto está disponible para datos del usuario

## **ADSL**

- El proveedor determina cuántos canales se utilizarán para el flujo ascendente y cuántos para el flujo descendente
- La mayoría de los proveedores asigna entre 80 y 90% del ancho de banda al canal descendente
- Esta situación dio lugar a la A (asimétrica) de ADSL
- Dentro de cada canal, **modulación QAM** es usada a una tasa de alrededor de 4000 symbols/sec
- Los datos actuales se envían con modulación QAM con un máximo de 15 bits por baudio
- Con 224 canales descendentes y 15 bits por baudio a 4000 baud, la velocidad del flujo descendente es de 13,44 Mbps
  - En la práctica la relación señal a ruido nunca es suficientemente buena para alcanzar esa tasa
  - Pero en trayectorias cortas sobre circuitos de alta calidad es posible lograr 8 Mbps
- La calidad de la línea en cada canal se monitorea de manera constante y la tasa de datos se ajusta cada vez que es necesario
- El módem ADSL funciona como 250 módems QAM operando en paralelo a diferentes frecuencias (implementa OFDM)

## **Internet Por Cable**

- Hay segmentos de cable coaxial, a cada uno de ellos se conectan varias casas
- **Sistema HFC(red híbrida de fibra óptica y cable coaxial)**
- Los convertidores electro-ópticos se llaman **nodos de fibra**
- Como el ancho de banda de la fibra es mucho mayor que el del cable coaxial, un nodo de fibra puede alimentar múltiples cables coaxiales
- **Modem-head end** antes de nodo de fibra
  - El ancho de banda que proporciona a cada nodo de fibra es grandísimo
  - Por lo tanto, siempre y cuando no haya demasiados suscriptores en cada segmento del cable, la cantidad de tráfico será manejable
  - Los cables típicos tienen entre 500 y 2000 casas

## ***¿Cómo aprovechar el cable de TV por cable para poder usarlo también para envío de datos?***

- Normalmente la TV y la internet coexisten en un mismo cable

## ***¿Cómo son los módems de internet por cable?***

- **Conexión Modem-computador:** Ethernet u ocasionalmente USB
- El otro extremo es más complicado y usa FDM, TDM Y CDMA para compartir el ancho de banda del cable entre los suscriptores
- En el cable se necesita modulación analógica
  - Para el flujo descendente cada canal descendente de 6 MHz se lo modula con:
    - QAM-64 (casi 36 Mbps de la cual se aprovecha 27 Mbps de carga útil), o
    - Si la calidad de cables es muy buena, QAM-256 (carga útil de 39 Mbps)
  - Para el flujo ascendente se varía de QPSK o QAM-128

## ***¿Cómo se hace multiplexado en un cable coaxial?***

- El uso de canales para flujo descendente de 6 Mhz o de 8 MHz es la parte FDM

- TDM se usa para compartir ancho de banda en el flujo ascendente entre varios suscriptores
  - El tiempo se divide en miniranuras y diferentes suscriptores envían en las diferentes miniranuras
  - Varios módem pueden compartir la misma miniranura, usando CDMA enviando simultáneamente, pero a una tasa reducida

## **Redes celulares:**

Generaciones: 1G, 2G, 3G, 4G y 5G.

(Todas las comunicaciones tienen que ser compatibles con las anteriores)

### **conceptos básicos:**

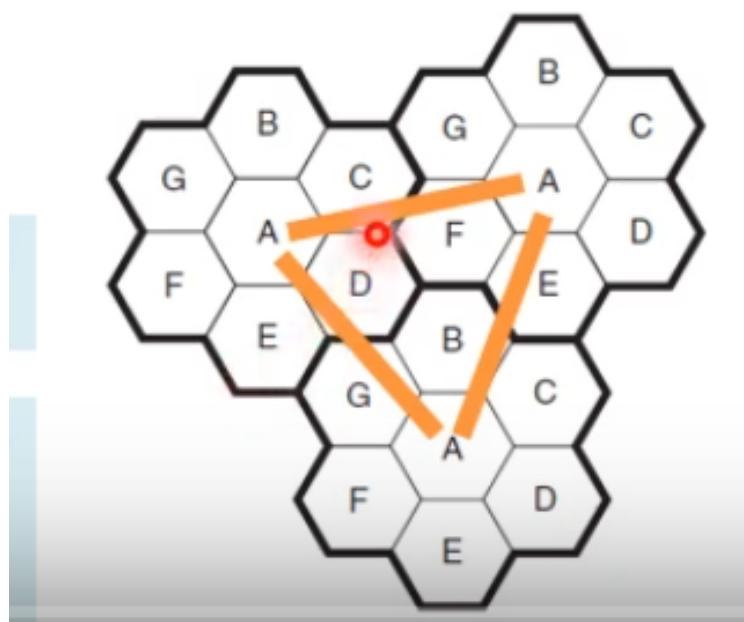
Nuestro dispositivo para acceder a la red, se tiene que conectar a una señal inalámbrica

Dividimos geográficamente la señal en celdas

celdas adyacentes != frecuencia de señal

Modelo de hexágono: este modelo es un ejemplo de la división geográfica de las celdas. Siendo las letras en el interior una frecuencia de señal, todas las celdas adyacentes tienen distintas frecuencias .

Este método se utiliza para evitar la interferencia de señales



una terminal(smartphone/dispositivo) se conecta a una única celda a la vez, pero puede ir variando en el tiempo dependiendo de cuál celda sea la más adecuada(señal más potente)

- las celdas se conectan a un MSC(centro de comutación móvil) multi-nivel que no se ve en detalle

### **Categorías en la que se dividen los canales de la red:**

La imagen es un ejemplo para el caso de 1G. Durante los distintos tipos de generaciones miramos algunos de estas categorías

## - Categorías

- Canales de **control** (base a móvil) → administración (21)
- Canales de **localización** (base a móvil) → avisos llamadas
- Canales de **acceso** (bidireccional) → llamadas (~45)
- Canales de **datos** (bidireccional) → fax, datos

## 1G

Tanto 1G como 2G están diseñados para el envío de paquetes de voz, no para el envío de datos de red(o sea internet)

Características:

- Analogica
- Advanced mobile phone system
  - usa FDM
- Frequency division multiplex (FDM)
  - Simple electrónicamente
  - Full duplex(comunicación que puede dar y recibir en el mismo momento)

## 2G

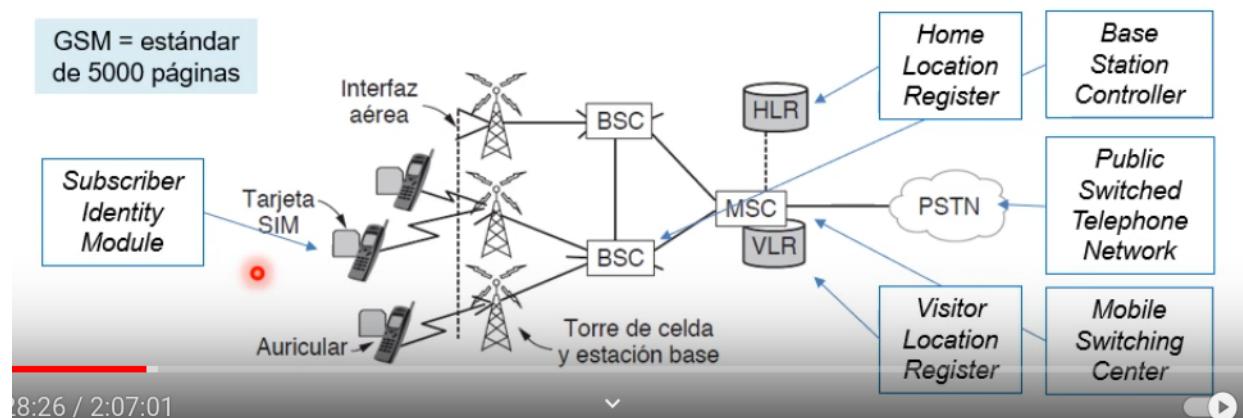
Incluye el envío de datos pero sigue siendo principal el de voz pero ahora digitalizada

Disputa de estas dos tecnologías:

CDMA: acceso múltiple por división de código(ete no)

GSM: Global system for mobile comm(Ete si)

- número de tarjeta SIM. Se utiliza hasta hoy en día. Se abstrae el conectarse a la red de la fabricación del artefacto
- El móvil no transmite la mayoría del tiempo para ahorrar datos



Datos : voz → 900 - 1800 - 1900 MHZ más espectros, más usuarios.

FDM → Principio de división de canales por frecuencia

- 2 canales grandes de 124 canales c/u → 124 de la base al móvil
- 124 del móvil a la base

TDM → dentro de cada canal tenemos una división por tiempo(tramas/T)

### Control :

Canal de control de difusión → todos los nodos escuchan

Canal de control dedicado → a una celula en específico

Canal de control común → localización

→ acceso aleatorio

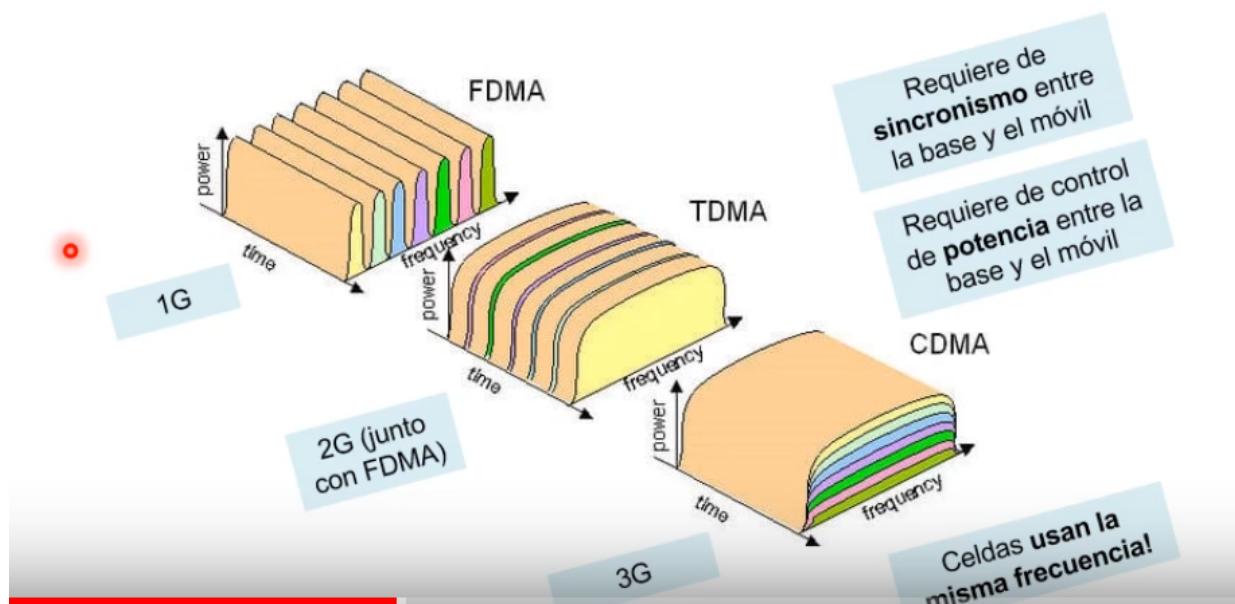
→ concesión

## 3G

El principal objetivo ahora es el envío de Datos. la voz pasa a un segundo plano

CDMA: Acceso por división de código

En TDM o FDM no es posible asignar ranuras → en CDMA un usuario puede reducir su interferencia



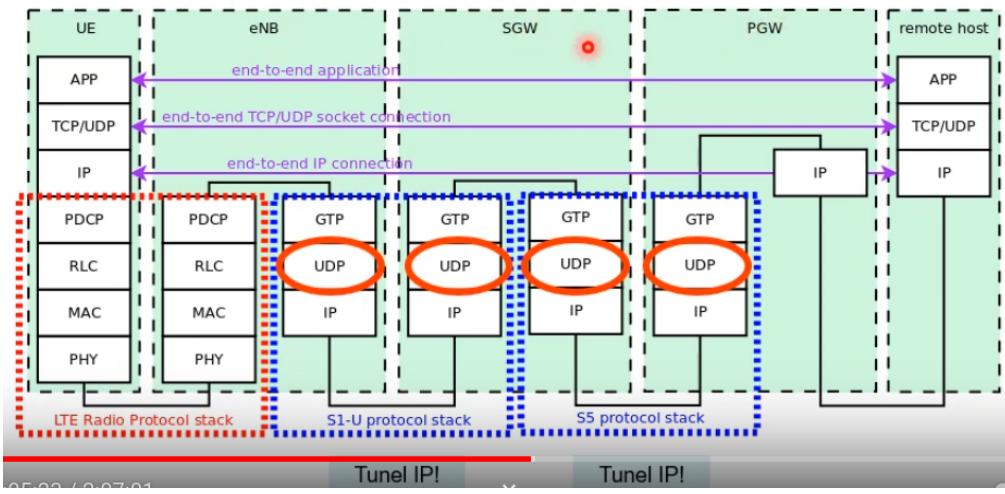
## 4G

Core y voz completamente sobre IP

Rompemos el modelo de capas: Capa de APP y TRANSPORTE siguen igual

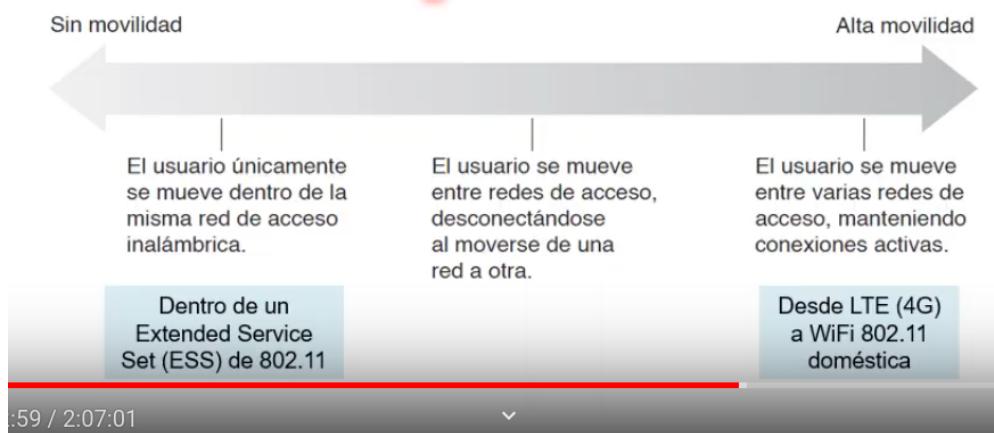
### Red(IP)“cambia”:

- ip usa radio base, con IP y UDP (una mezcla )
- túnel IP/UDP → arquitectura recursiva



## Gestión de movilidad:

Como cambia el punto de conexión con la red desde un smartphone/dispositivo a lo largo del tiempo



- trabajamos con movilidad media, Sin movilidad no interesa obviamente y Alta movilidad super tryhard

Movilidad media:

conceptos básicos:

HOME NETWORK:

- Red propia
- Home agent: Gestiona la movilidad

FOREIGN NETWORK:

- Red ajena o visitada
- Foreign agent : Gestiona la movilidad

CORRESPONSAL:

- El que se quiere comunicar con el móvil

## CORE OF ADDRESS(CoA):

- Parte de la red Foreign, asignada al dispositivo que la está visitando

¿Cómo funciona ?

El home agent controla en qué red visitada está el smartphone

situación: fuera de la red Home, en alguna red Foreign

### Enrutamiento indirecto: (más usado)

- paso 1: Servidor envía los datos a la dirección permanente(Home Network)
- paso 2: El Home Agent encapsula los datos dentro de un paquete más grande, y lo envía a CoA
- paso 3: El Foreign agent desencapsula y le entrega los datos a el dispositivo
- paso 4: El nodo móvil ahora envía directamente sus datos al Servidor

### Enrutamiento directo:

- Se necesita un foreign agent, este se comunica con home agent y le aviso donde está(el dispositivo) luego, como enrutamiento indirecto