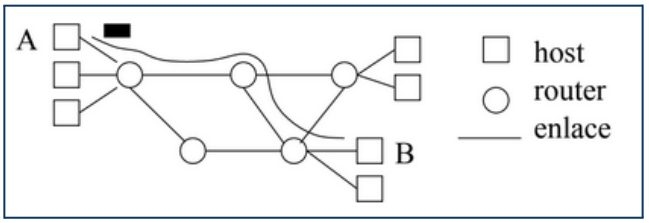
Introducción

Elementos de internet

- **Hardware:**
  - **Hosts:** Origen y destino de las transmisiones
  - **Enlaces:** medio físico por el que se realiza la transmisión
  - **Routers:** Interconectan enlaces



- **Software:** protocolos básicos (TCP, IP) o de aplicación (HTTP, SMTP)
- **ISP:** Proveedores de baja escala de Internet a usuarios o de alta escala a países/internacionales

Servicios

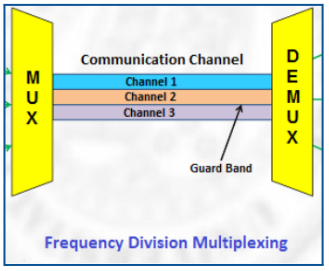
- **Con conexión (TCP, HTTP):**
  - Se solicita la conexión, se fijan parámetros y se preparan ambos extremos.
  - Se transmiten los datos y al terminar se liberan los recursos
  - La tasa de envío puede ser controlada por el emisor (control de flujo) y ser ajustada a las capacidades de la red (control de congestión)
  - Segmentación: El TCP recoge datos que la app escribe en el socket y los separa en paquetes según el MSS (maximum segment size)
  - El receptor envía confirmaciones: si el emisor no recibe el ACK de un paquete, se retransmite
- **Sin conexión (UDP):**
  - No hay fase de establecimiento ni confirmaciones ni control de flujo
  - Más rápida pero menos fiable

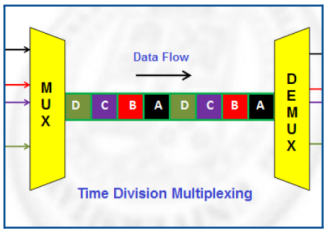
Tipos de redes según hardware

- **De conmutación:** circuitos o paquetes
- **De difusión:** Ethernet, inalámbricas, etc.
  - Todos los hosts reciben las transmisiones, solo el destinatario las procesa

Redes de conmutación de circuitos

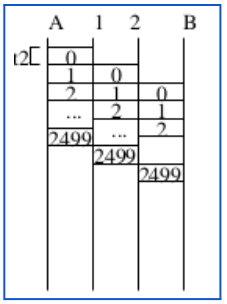
- Fase de conexión: se reservan recursos hardware y se establecen las rutas
- Transmisión de datos
- Tras la desconexión se liberan los recursos
- **Sin multiplexación:** Sin FDM ni TDM. Se envía solo un paquete de cada vez.
- **Con multiplexación:** se reparte la capacidad del enlace entre:
  - División en frecuencia (FDM): bandas de frecuencia
  - División en tiempo (TDM): ranuras temporales
- Mantienen los recursos aunque no haya datos que enviar



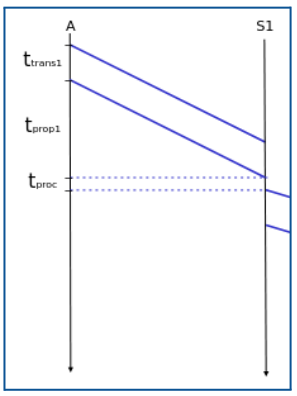


Redes de conmutación de paquetes

- No se reservan recursos, se comparten y asignan bajo demanda
- Segmentación de datos (paquetes) con información de control (ACK)



- Los routers funcionan como conmutadores de paquetes: reciben el paquete y si hay sitio lo almacenan en la cola (si no se descarta) [store and forward]
- **Retardos:** Se deben sumar:
  - De transmisión(t_{trans}): Tamaño paquete / ancho de banda[1].
  - De propagación(t_{prop}): Longitud enlace / v propagación
  - De cola(t_{cola}): Nº routers*Tiempo de espera en cola de router
  - De procesamiento(t_{proc}): Nº routers * T de proc. por router



- **Capacidad** del enlace: retardo * ancho de banda
  - Máximo nº de bits que puede estar en tránsito a la vez
  - Aprovechamiento: nº de bits que el emisor emite antes de que el receptor reciba el primero
    - Si el emisor espera ACKs, el retardo será doble.

[1] tamén denominado 'velocidad de transmisión del enlace'

- Pueden ser:
    - **De datagramas:** Cada paquete incluye en la cabecera la IP de destino
        - Reenvío: el router examina la cabecera y lo coloca en la sálida más apropiada
        - No mantienen información de estado
        - Más flexibles y tolerantes a fallos, mejores en redes grandes y heterogéneas como internet.
    - **De circuitos virtuales:** Se establece la conexión planificando una ruta: el número de circuito virtual.
        - A cada paquete se le escribe el CV que usan los routers para el reenvío
        - Los routers mantienen información de estado
        - Más fiables y rápidas en redes pequeñas y homogéneas

Ejemplo - Retardo de propagación

- **Suponer un enlace de microondas a 10 Mbps entre un satélite geoestacionario y su estación base en la Tierra, a una distancia de 36.000 Km. El satélite toma (una) fotografía digital por minuto y la envía a la estación base. La velocidad de propagación es de $2.4*10^8$m/s**
    - **¿Cuál es el retardo de propagación del enlace?**
    - **Calcular el producto retardo por ancho de banda**
    - **Sea x el tamaño de la fotografía en bytes. Calcular el valor mínimo de x para que el enlace esté transmitiendo continuamente.**
- El retardo de propagación es el tiempo que tarda la señal en viajar por el enlace.
    - $3.6*10^6$m / $2.4*10^8$m/s = 0.15 s
- Se asume que el retardo es sólo por propagación.
    - 10 Mbps * 0.15s = 1.5.
- Para que esté transmitiendo continuamente, debe cumplirse que $t_{trans} >= 2*t_{prop}$
    - t_{trans} = x bits / 10_7 bps >= 0.3 s
    - La foto debe ocupar al menos $3*10^6$ bits, o 375000 bytes (375 kb)

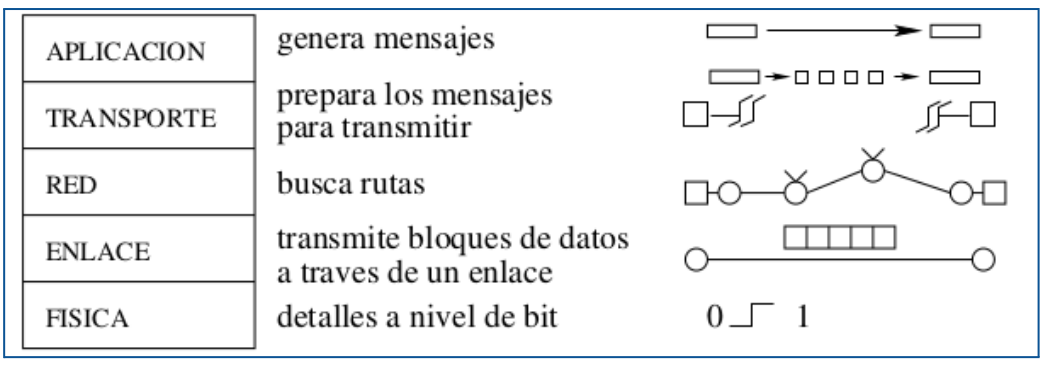
Acceso a internet

- Acceso residencial (FTTH -> fibra hasta el domicilio, ADSL)
    - Módem: usa línea telefónica
    - DSL: aprovecha todo el ancho de banda de frecuencias del cable telefónico. Existen frecuencias distintas para voz, subida a internet y bajada de internet. Velocidades de hasta 30mbps
- **Acceso empresarial/doméstico:** WiFi, Ethernet
    - Una empresa generalmente realiza el acceso mediante una red local LAN de tipo internet.
- **Móvil:** WiFI, 3G,4G,5G, LTE

Non vou facer apuntes dos tipos de cables.

Arquitectura TCP/IP

- La arquitectura en capas consiste en dividir la comunicación en tareas independientes, cada una en una capa.
- Además de los datos, cada una de las capas debe transmitir instrucciones, añadidas a los mensajes como cabeceras.
- **Capa de aplicación:** Contiene los procesos que se comunican entre sí. HTTP, FTP, SMTP, DNS…
- **Capa de transporte:** Prepara los mensajes para poder transmitirse fuera del computador. TCP, UDP.
- **Capa de red:** Busca las rutas y transporta los paquetes. IPv4, IPv6, (DHCP, NAT)
- **Capa de enlace:** Maneja los detalles de bajo nivel del transporte de paquetes. Datagramas, circuitos virtuales, difusión etc.
    - El enlace peude ser de cable (Ethernet) o por ondas (WLAN)
    - Encargado de comenzar la transmisión, almacenar los datos en memoria al recibirse etc.
- **Capa física:** Convierte los bits en señales eléctricas y define las características físicas del medio de transmisión



Capa de aplicación

Introducción

- La **capa de aplicación** se encarga de la comunicación entre la aplicación y el servidor.
- El agente de usuario es la interfaz entre el usuario y la aplicación (navegador)
- Se utilizan distintos protocolos de comunicación, que facilitan las funciones de envío y recepción.
    - Las funciones especifican el tipo de mensajes a intercambiar, las reglas de cuándo y cómo se envían, la sintaxis y semántica de los mensajes.
    - Existen:
        - **Protocolos TCP:** HTTP (web) , STMP/POP3/IMAP (correo) y FTP (ficheros): servicio fiable, orientado a conexión
        - **Protocolos UDP:** DNS (traducciones), no fiable pero más rápido

HTTP (Protocolo de transferencia de hipertexto)

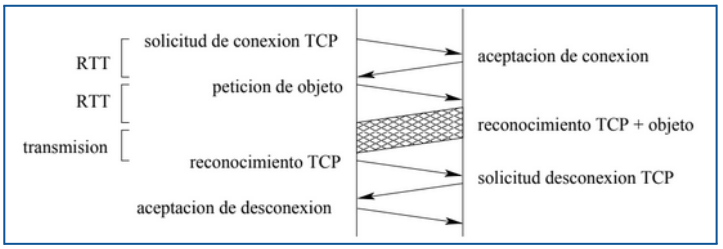
- Define la comunicación entre un servidor web y un cliente web.
- Protocolo sin estado, a veces utiliza mecanismos como cookies
    - Puede utilizar DNS, pero no lo necesita si el equipo ya tiene asociada la IP a un nombre de host.
- **Página web:** Consta de un archivo HTML base y objetos (archivos direccionables por un URL)
- **Servidor web:** Alberga objetos

Cookies

- Archivo de texto enviado por el servidor web al navegador, que lo almacena localmente y le vuelve a enviar al servidor con cada solicitud.
- Permite que el servidor reconozca al usuario entre solicitudes, guardando sus preferencias e identidad.
- La cookie contiene fecha de expiración, dominio, ruta y la seguridad donde debe usarse (https). Limitada a 4kb.
- **Alternativa**: almacenamiento del lado del servidor
    - En la cookie sólo se guarda un identificador del usuario
    - Con este identificador se accede a la base de datos del servidor, que guarda las preferencias.

Conexiones HTTP

- **No persistentes:** Se usa una conexión TCP distinta para cada objeto (HTTP/1.0)
  - En serie: se espera por la conexión previa.
  - En paralelo: Se inician varias conexiones a la vez
- **Persistentes:** Se transfieren varios objetos o páginas con la misma conexión TCP
  - Sin entubamiento: El cliente pide un nuevo objeto tras recibir el previo
  - Con entubamiento: Se pueden pedir varios objetos (por defecto en HTTP/1.1)
- **Tiempo de transferencia:**
  - RTT: Tiempo de ida y vuelta de un paquete pequeño entre servidor y cliente
  - Tiempo de transmisión: Depende del tamaño del archivo.
  - En todos los tipos de conexiones el primer objeto tarda 2*RTT + t_{trans}, el resto dependerá del tipo de conexión.
    - En las no persistentes (imagen), cada nuevo objeto implica un tiempo de 2RTT+t_{trans}, ya que se debe resolicitar la conexión.
    - En las persistentes sin entubamiento cada objeto requiere un tiempo de RTT+t_{trans}, mientras que con entubamiento no es necesario pedir cada objeto por lo que es sólo t_{trans}.



Mensajes HTTP

- Pueden ser de petición o de respuesta (contienen objetos)
- Ambos tienen **cabecera** en ASCII de 7 bits y **cuerpo** en binario (contiene los datos)

FTP (Protocolo de transferencia de ficheros)

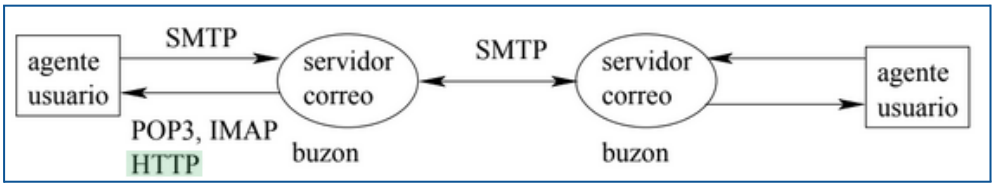
- Define la comunicación con un servidor de archivos.
- Mantiene información durante toda la sesión
- Usa dos conexiones TCP paralelas, ambas de ASCII de 7 bits (también permite datos binarios):
  - Conexión de control: puerto 21, persistente, transmite comandos
    - Comandos: USER, PASS (usuario y clave), LIST (lista de archivos) RETR (traer archivo), STOR(guardar archivos)
  - Conexión de datos: puerto 20, transmite datos en respuesta a comandos, no persistente (nueva conexión para cada archivo)

Protocolos de correo electrónico

- Para enviar correo a un servidor de correo o entre servidores, SMTP o HTTP
- Para acceder al correo por agente de usuario, POP3, IMAP o HTTP

SMTP (Protocolo de transferencia de correo sencillo**)**

- Comunicación entre agente de usuario y servidor o entre servidores.
- Usa el puerto 25.



- Conexiones TCP persistentes durante la sesión (varios mensajes), ASCII de 7 bits
- Tipos de mensajes:
  - Comandos:Palabra en mayúscula seguide de parámetros. HELO (nombre servidor), MAIL FROM (remitente), RCPT TO (dest), DATA (contenido), QUIT
  - Respuestas: código numérico que informa sobre el resultado de un comando
  - Datos: Contenido del correo
- Diferencias con HTTP:
  - SMTP es inseguro, no pide clave ni nombre de usuario
  - HTTP es un protocolo de demanda y SMTP de oferta
  - SMTP requiere cabecera y cuerpo de 7 bit, HTTP permite contener binario
  - HTTP envía objetos en archivos diferentes, SMTP los encapsula en el texto
  - HTTP no tiene estado, SMTP debe recordar la fase en la que se encuentra

POP3

- Protocolo de oficina postal. Permite descargar correos del servidor y borrarlos o mantenerlos en él.
- Usa el puerto 110.
- Fases: Autorización, transacción y actualización.
- Conexiones persistentes, no mantiene información entre sesiones.
- Mensajes:
  - Comandos: user, pass, list, retr, dele, quit
  - Respuestas: OK o ERR, con una explicación
  - Datos: Contenido del correo

IMAP

- Más complejo que POP3: mantiene información entre sesiones y permite carpetas en el servidor. (primero van a inbox, se pueden mover)
- Usa el puerto 993.

DNS (Servicio de nombres de dominio)

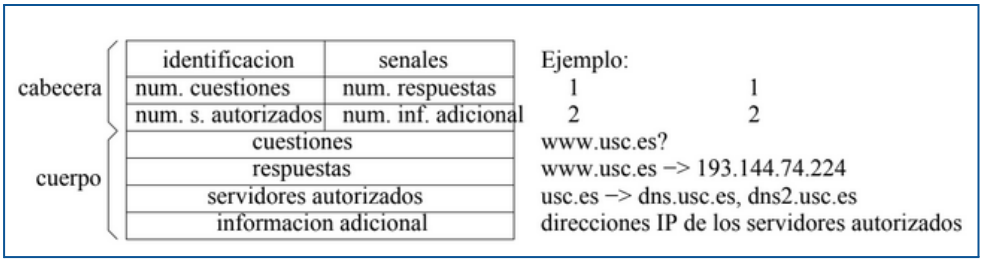
- Permite traducir entre nombres de hosts y direcciones IP.
- Suele usar UDP.
- El puerto de DNS (destino de la solicitud) es **53**. El cliente en su solicitud usa uno aleatorio.
- Formado por una base de datos jerárquica de nombres y un protocolo que permite a los hosts pedir traducciones a los servidores de DNS, que intercambian datos entre ellos.
    - Servidores locales: Atienden a las consultas de los hosts. Se identifican con una red local ('dns.empresa.com')
    - Servidores autorizados: En ellos están registrados todos los hosts accesibles en internet. Cada host debe estar al menos en dos, por fiabilidad.
        - Normalmente pertenecen al ISP. Muchos se comportan como locales
    - Servidores raíz: Existen sobre 400, gestionados por 13 organizaciones
    - Servidores intermedios (TLD): Información sobre los niveles intermedios. Se identifican con una IP.
- Informan de los servidores autorizados para un dominio, simplifica direcciones de correo y permite distribuír carga asignando varias IPs a un mismo nombre de host.

Consultas DNS

- Consultas recursivas: Cada servidor DNS interroga al siguiente. (host → local→raíz→intermedio→autorizado)
- Consultas iterativas: Un servidor DNS local contacta con todos.
- Todos los niveles almacenan caché de las correspondencias obtenidas durante unos dos días.

Mensajes DNS

- **Cabecera:** Información de control
  - Identificación: 16 bits que identifican consulta y respuesta correspondiente
  - Señales: 4 bits. Indican si es una consulta o respuesta, tipo de consulta.
  - Tamaño de los campos del cuerpo
- Cuerpo: 4 campos de tamaño variable.
  - Cuestiones: Una o varias preguntas. Ej: nombre o dirección a traducir
  - Respuestas: Una o varias. Pareja host/IP pedida.
  - Servidores autorizados: Permite hacer una cadena de consultas



Distribución de contenidos

- Se distribuyen los contenidos de un mensaje en distintas zonas, dirigiendo las peticiones al servidor de menor tiempo de respuesta.
- Proxy: Conecta clientes con servidor web
  - Mantiene copias de los contenidos durante un tiempo
  - Servidio proporcionado por el ISP
- CDN: Redes de distribución de contenidos.
  - Replica los contenidos de los clientes en los servidores CND.
  - El contenido es entregado por el servidor CND que pueda hacerlo más rápido
- P2P: Los usuarios son simultáneamente servidores y clientes
  - Se puede construir un directorio o almacenar la información en un archivo (torrent)

Capa de transporte

Introducción

- La **capa de transporte** prepara los mensajes de las aplicaciones para ser transmitidos, recupera los mensajes y los entrega a las aplicaciones.
- La capa de transporte sólo está implementada en origen y destino, no en routers intermedios.
- Proporciona una comunicación lógica entre procesos,.
- Protocolos de transporte TCP y UDP:
    - **TCP:** segmenta los mensajes y añade a cada uno la cabecera de la capa de transporte. Se garantiza la entrega y orden de todos los segmentos.
    - **UDP:** añade la cabecera de la capa de transporte, utilizada para la detección de errores básicos.
        - No segmenta los mensajes. Envía los datos en bloques llamados datagramas. Si un datagrama supera el tamaño del MTU, la fragmentación deberá ocurrir en la capa de red.
        - No garantiza la entrega ordenada ni transmisión de paquetes perdidos. Se usa cuando se prefiere velocidad sobre fiabilidad.
        - Puede manejar más clientes simultáneamente, útil en DNS

Multiplexación / Demultiplexación

- Los mensajes pasan de capa de aplicación a transporte a través de un **socket**.
    - Los procesos escriben y leen del socket
    - La capa de transporte recoge los mensajes del socket y los lleva al socket destino
- **Multiplexación:** Recorrer todos los socket abiertos, procesar los mensajes y enviarlos a la capa de red.
- **Demultiplexación:** Recoger los segmentos que llegan de la capa de red, reconstruir los mensajes y colocarlos en el socket destino.
    - El socket destino se identifica a través de los números de puerto de la cabecera del segmento. Son enteros de 16 bits.
    - Los puertos del 0 al 1023 están reservados para servicios estáFndar.

- Sin conexión (UDP): El socket se identifica por la pareja IP/puerto de destino

    - Los segmentos de distintos hosts entregados al mismo puerto son recogidos por el mismo proceso

- Orientado a conexión (TCP): El socket se identifica por la tupla de direcciones IP (origen y destino) y la tupla de puertos origen y destino.
    - Segmentos de distintos hosts/puertos origen que vayan al mismo puerto van a sockets distintos.
        - Existe el socket de servidor, que espera conexiones, y varios sockets de conexión que se encargan de la transmisión.
    - Varias conexiones al mismo puerto pueden ser atendidas por distintos procesos/hilos.

UDP (protocolo de datagrama de usuario)

- Protocolo de transporte simple y poco sofisticado. No orientado a conexión.
- En origen se añade una cabecera de 4 campos de 8 bits cada uno (total 4 bytes):
    - Puertos origen/destino
    - Longitud total del segmento
    - Suma de comprobación
- En destino sólo se comprueba si el paquete llegó sin errores.
    - Si llega mal se descarta (común) o se pasa al socket pero con un aviso
    - Funcionamiento de la suma de comprobación: en el origen:
        - Se suman todas las palabras del segmento, incluida la cabecera
        - Se toman 4 bits menos significativos del resultado, sumando los acarreos
        - Se hace el complemento a 1. El resultado se guarda en 'suma comprob.'
    - En el destino se realizan las mismas cuentas, y se comprueba si el resultado es igual al que viene en 'suma comprob'.
- **Características:**
    - Sin conexión, no hay retardo de establecimiento de conexión
    - Sin estado → Más rápido y menos recursos, pero menor fiabilidad
    - Cabeceras más pequeñas.
    - Se usa en DNS, RIP(encaminamiento), SNMP (admin. de red)

TCP (protocolo de control de transmisión)

ARQ de parar y esperar

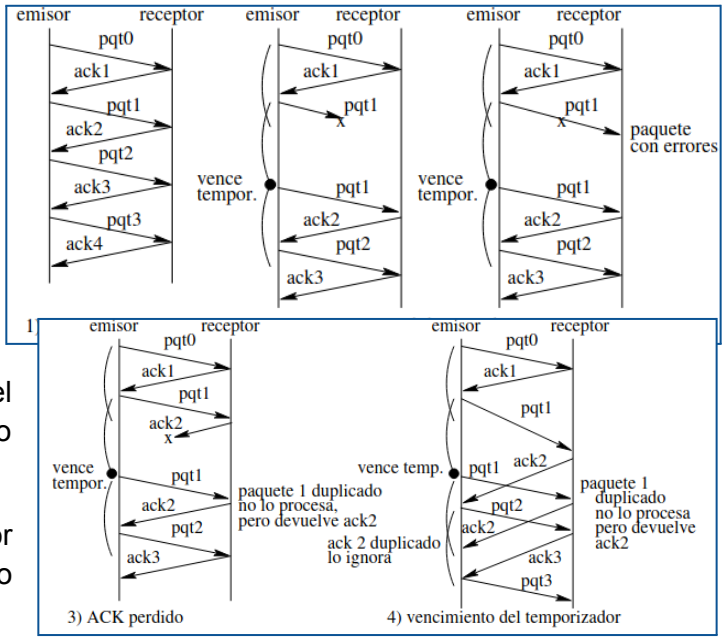
- **Automatic Repeat reQuest:** solicitud automática de repetición. Se retransmiten los paquetes con errores.

- Caso sin errores: Devuelve un ACK al emisor. Por convenio es el número del siguiente paquete a enviar. Cuando el emisor obtiene un ACK, pasa a enviar el siguiente paquete.

- Caso de pérdida: El receptor no recibe nada. Cuando el emisor envía un paquete pone a funcionar un temporizador y si se acaba el tiempo sin recibir reconocimiento, lo retransmite.

    - Si el paquete se recibe con errores es el mismo caso: el receptor lo descarta y no devuelve reconocimiento

- ACK perdido: El emisor retransmite el paquete, por lo que el receptor lo repite duplicado. No lo procesa, devuelve el ACK



- Vencimiento del temporizador: Cuando el temp. es muy corto o cuando hay congestión. Habrá paquetes y ACKs duplicados. El paquete duplicado se procesa como en el previo caso, el ACK duplicado se ignora.

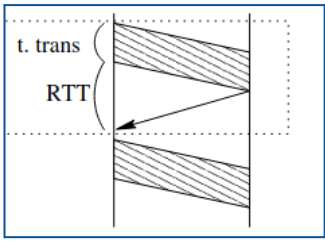
    - Variantes:

        - **NAK:** Indica la recepción de un paquete con errores, no espera al timeout.

        - Dos ACKs iguales equivalen a un NAK, la recepción de un paquete con errores devuelve el ACK del último paquete correcto, pero si vence el timer se envían continuamente paquetes duplicados.

        - Tres ACKs iguales equivalen a un NAK, lo que resuelve el previo problema. Es lo que usa TCP con variaciones.

    - Inconveniente: Poca utilización del enlace. $U = t_{trans} / (RTT+t_{trans})$.[2] Se resuelve con ARQ de ventana deslizante.



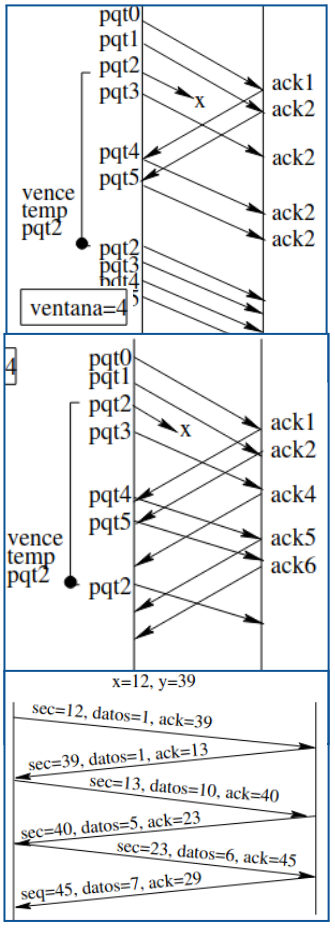
ttrans = longitud de paquete / velocidad en bps (bits)

ARQ de ventana deslizante

- **Es el ARQ usado en TCP.**

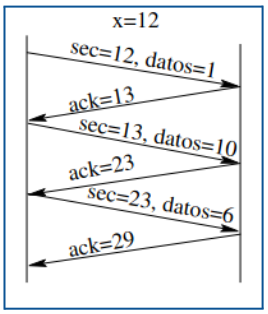
[2] U: cociente entre tiempo útil y tiempo total. Tiempo útil: aquel en el que el emisor transmite.

- En lugar de enviar un paquete y esperar a recibir un ACK para enviar el siguiente, envía un número N de paquetes antes de recibir las confirmaciones (Entubamiento)
- Tiempo útil = $n*t_{trans}$, tiempo total = t_{trans} + RTT
- Ventana emisora: Conjunto de N paquetes que el emisor puede enviar o están pendientes de confirmación
- Ventana receptora: Conjunto de N paquetes que el receptor puede aceptar o está procesando
- Requisitos: el rango de números de secuencia debe abarcar al menos el doble del tamaño de ventana emisora, y emisor/receptor deben almacenar más de un paquete.
- **Tipos:**
    - Retroceder N: El receptor sólo acepta paquetes ordenados
        - Si un paquete no llega o llega mal se descartan todos los siguientes hasta que el emisor retransmita el paquete perdido.
        - Si expira el temporizador de un paquete se retransmiten los siguientes
    - Repetición selectiva: El receptor acepta paquetes fuera de orden
        - Solo se retransmiten los paquetes erróneos o que no llegan
        - Hay que enviar los ACKs de todos los paquetes recibidos.
- **Piggybacking / superposición:** Sirve en situaciones donde se transmiten datos en ambas direcciones.
    - En lugar de intercalar datos y reconocimientos, cuando llega un segmento de datos, no se envia inmediatamente el reconocimiento, sino que se espera a tener que enviar datos y se envían los datos del nuevo mensaje junto con el reconocimiento del previo. Es más eficiente.
- **Ventajas:** Mucho más eficiente, sobre todo en redes de alta latencia y ancho de banda.
- **Desventajas:** Más complejo de implementar

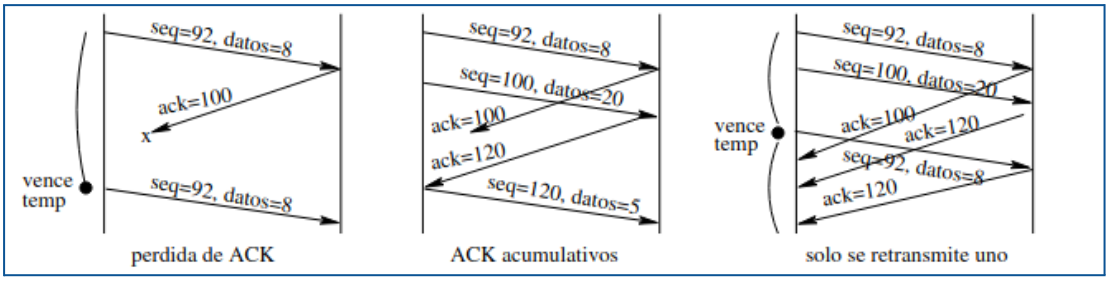


Protocolo TCP

- Emplea los principios de transmisión fiable explicados previamente.
- **Números de secuencia:** 32 bits[3], identifican **bytes**.
    - Se empieza por un x aleatorio con cada mensaje incrementa por el nº de bytes enviados.
    - Los ACKs indican el siguiente byte a recibir, o lo que es lo mismo, el próximo nº de secuencia que se espera.



- El emisor usa un temporizador único para la retransmisión, que se reinicia cuando llega un ACK, para ahorrar recursos.
- El ARQ



utilizado en TCP es de **ventana deslizante** y combina elementos de ambos tipos, pero es más próximo a **repetición selectiva**.

- Retransmite sólo los segmentos perdidos.
- Utiliza **ACKs acumulativos** para indicar hasta qué nº de secuencia se han recibido correctamente los datos.
- Si los segmentos llegan desordenados:
    - Cuando el receptor recibe un segmento con sec mayor del esperado, envía un ACK con el número del paquete que esperaba recibir.
    - Una vez llega este paquete, el receptor los reordena y envía el ACK del próximo paquete que aún no ha sido enviado. El emisor no necesita retransmitir paquetes.

- **Tiempos:**
    - DevRTT: desviación de RTT, en general a=0.125 y B=0.25
    - Cuando se produce la expiración del temporizador el emisor duplica el tiempo de temporización.

$$\begin{aligned} \text{Temporizador} &= \text{EstimacionRTT} + 4\text{DevRTT} \\ \text{EstimacionRTT} &= (1-\alpha)\text{EstimacionRTT} + \alpha\text{MuestraRTT} \\ \text{DevRTT} &= (1-\beta)\text{DevRTT} + \beta|\text{MuestraRTT} - \text{EstimacionRTT}| \end{aligned}$$

[3] Debido a esto, el tamaño máximo de archivo que enviar es el nº de bytes representables con 32 bits: 2^{32}=4GiB≈4.29 GB

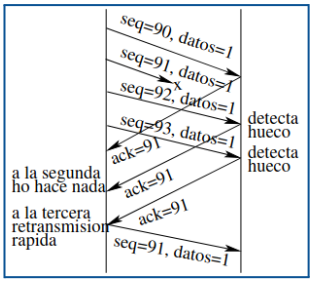
Ejemplo - Segmentación TCP

- **Suponer que la MTU de los enlaces entre el host A y el host B está limitado a 1500 bytes. Indicar cuántos datagramas IPv4 se necesitarían para enviar un archivo de 4000 bytes si la aplicación utiliza TCP con un MSS de 1460 bytes.**
- Al ser TCP, es necesario enviar 2 datagramas primero, uno con SYN y otro con ACK tras recibir el SYN del host B. Estos datagramas no pueden ser fragmentados (NF=1) y no llevan datos (tamaño=40).
- Dado que se desean enviar 4000 bytes y el MSS es 1460, serán necesarios 3 segmentos. Los dos primeros llevan 1460 bytes de datos y ocupan 1500 bytes en total. El último lleva los 1080 bytes restantes y ocupa 1120 bytes. El valor del campo NF puede ser 1 o 0 dependiendo de la implementación.
- El valor de los campos 'MF' y 'desplazamiento' será 0 para todos pues cada datagrama es su propio fragmento (no es necesaria fragmentación a nivel IP)

Tamaño	Identificación	MF	NF	Desplazamiento
40	356	0	1	0
40	357	0	1	0
1500	358	0	X	0
1500	359	0	X	0
1120	360	0	X	0

Retransmisión rápida

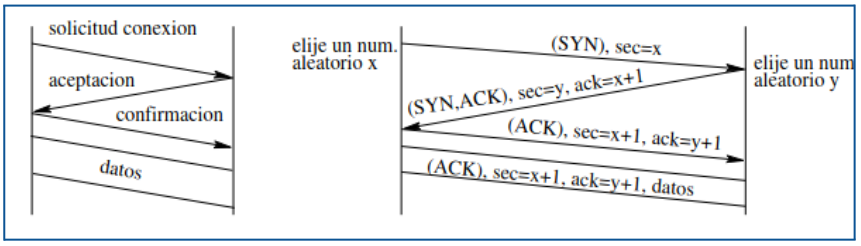
- Detectar pérdidas con el temporizador puede ser lento. TCP tiene una alternativa:
- **Retransmisión rápida:** Envío de un paquete que había sido perdido antes de que expire el temporizador.



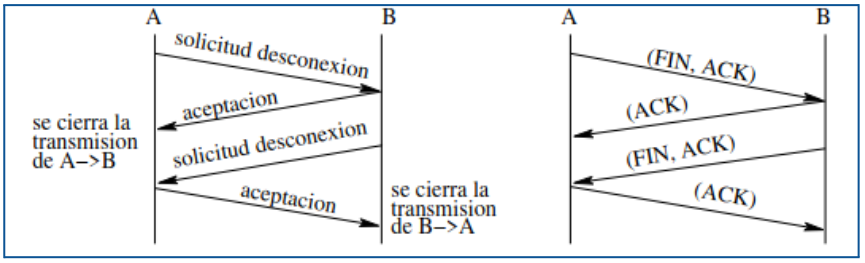
    - El receptor sabe que se ha perdido un paquete porque se ha saltado un número de secuencia.
    - El receptor responde con un ACK duplicado: el mismo ACK del último paquete recibido correctamente.
        - Continúa enviando este ACK cada vez que el emisor le envíe un paquete.
    - Cuando el emisor recibe **3 ACKs duplicados** asume que el paquete siguiente al nº de secuencia indicado se ha perdido y reenvía solo este paquete.
- Mezcla elementos de retroceder N (ACKs acumulativos) y repetición selectiva (acepta segmentos fuera de orden, retransmite sólo los necesarios).

Conexión / desconexión TCP

- **Conexión** en tres fases.
- SYN=1 cuando se envía por primera vez x o y, ACK=1 cuando se confirma un segmento. En la tercera fase se pueden enviar datos.



- **Desconexión** en dos fases, una para cada sentido. Se usa FIN para solicitarla y ACK para aceptarla.



Transmisión de datos

- La aplicación va escribiendo los datos en el socket.
  - La transmisión de un segmento se dispara cuando se alcanza el MSS (maximum segment size) o cuando la aplicación fuerza el envío con **push**.
- TCP genera el segmento y se lo pasa a IP.
- **Generación de ACKs:**En la transmisión A→B se debe incluir el ACK (piggybacking)
  - Si el receptor no tiene datos para enviar, recibe un segmento en orden y el anterior fue confirmado, retrasa el ACK hasta que reciba otro segmento o transcurran 0.5s

Control de flujo en TCP

- Mecanismo que permite al receptor indicar al emisor el ritmo al que puede recibir datos.
- En el momento de la conexión el receptor indica el tamaño de su ventana de recepción, que puede modificar en cada transmisión
- **Congestión:** Demasiados paquetes en la red, que producen retardos en las transmisiones y pérdidas de paquetes.
- Alternativas: prereservar recursos para evitar congestión o dejar que ocurra congestión y resolverla. (internet fai esto ultimo)
- El tamaño **máximo** de la ventana viene limitado por la velocidad del enlace (t_{max}/RTT=v_{enlace})
  - La ventana varía de t_{max}/2+3MSS a t_{max}, por la fase de AIMD.

- Posibles orígenes de congestión:
    - Dos emisores, tasa de transmisión mayor que (velocidad transmisión/2): El enlace no puede proporcionar paquetes a esa velocidad, por lo que quedan en la cola del router.
    - El mismo caso, pero router con memoria finita: la tasa entregada disminuye, porque algunos paquetes serán duplicados.
    - Varios emisores y varios enlaces, con tasa de transmisión elevada: Los buffers de los routers se llenan.

Control de congestión en TCP

- **Mecanismos para actualizar ventana de congestión:**
    - Inicio lento: Determina capacidad inicial de la red.
        - Inicialmente, el tamaño de la ventana de congestión (**VS)** es pequeño (1 o 2 veces el máximo tamaño de segmento)
        - El tamaño se duplica con cada ciclo de envío de datos.
        - Este proceso se continúa hasta que ocurre una pérdida de paquetes por congestión o hasta alcanzar **umbralAL**.
    - Evitación de la congestión (AIMD): Incremento aditivo, decremento multiplicativo. Comienza tras acabar el inicio lento.
        - El tamaño de la ventana de congestión aumenta más lentamente, aumentando en 1 MSS cada ciclo.
        - Cuando se produce una pérdida, se asume que ha sido por congestión y se detiene el crecimiento.
            - Si es una pérdida por temporizador se pasa a AL.
            - Si es una pérdida por 3 ACKs se pasa a recuperación rápida.
    - Recuperación rápida: Evita volver a la fase de inicio lento cuando hay congestión.
        - Cuando se detecta una pérdida, **umbralAL = max(VS/2, 2*MSS)**
        - Es decir, se reduce a la mitad umbralAL (y luego se fija VS al mismo valor para evitar volver a inicio lento).
        - Luego de fijar VS a umbralAL, por cada ACK duplicado que se reciba antes de recuperar la pérdida, VS += MSS. Entonces, en este tiempo VS puede superar umbralAL.
        - Cuando se recupera la pérdida, se vuelve a AIMD. Si VS había sindo incrementado por MSS, se vuelve a fijar al valor de umbralAL.

- **Notificación explícita de congestión (ECN) :** Un router congestionado activa unos bits de la cabecera IP, en un campo denominado ECN.
    - Cuando el receptor recibe este paquete, notifica al emisor mediante un flag en el encabezado del paquete TCP.
    - Cuando el emisor recibe esta señal, reduce la tasa de transmisión para evitar perder paquetes.
- Si dos aplicaciones comparten un enlace de capacidad limtiada,
    - Dos conexiones TCP se repartirán la capacidad del enlace.
    - Si es una TCP y una UDP, la UDP acaparará la mayor parte de la capacidad.
    - Si una aplicación usa n veces más conexiones TCP paralelas, utilizará n veces más capacidad.

Ejemplo - Recuperación rápida

- **Recuperación rápida. Cuando se retransmite el paquete, umbral=2048 y ventana de congestión=2816. Si MSS=256, ¿qué ha sucedido?**
- En este caso, al retransmitir el paquete la ventana de congestión es de 2816 bytes. Conocemos, entonces, que no se ha pasado a inicio lento sino a recuperación rápida, por lo que sabemos que se ha producido una pérdida de un paquete y se ha detectado después de que el receptor enviase 3 ACKs duplicados.
- Además, se comprueba que VS = umbralAL+ 3*MSS (2816=2048+3*256). Esto nos confirma que se acaba de usar el mecanismo de recuperación rápida para reducir a la mitad la ventana de congestión (a partir del tamaño del umbral, conocemos que antes de la pérdida era de 2048*2=4096 bytes).

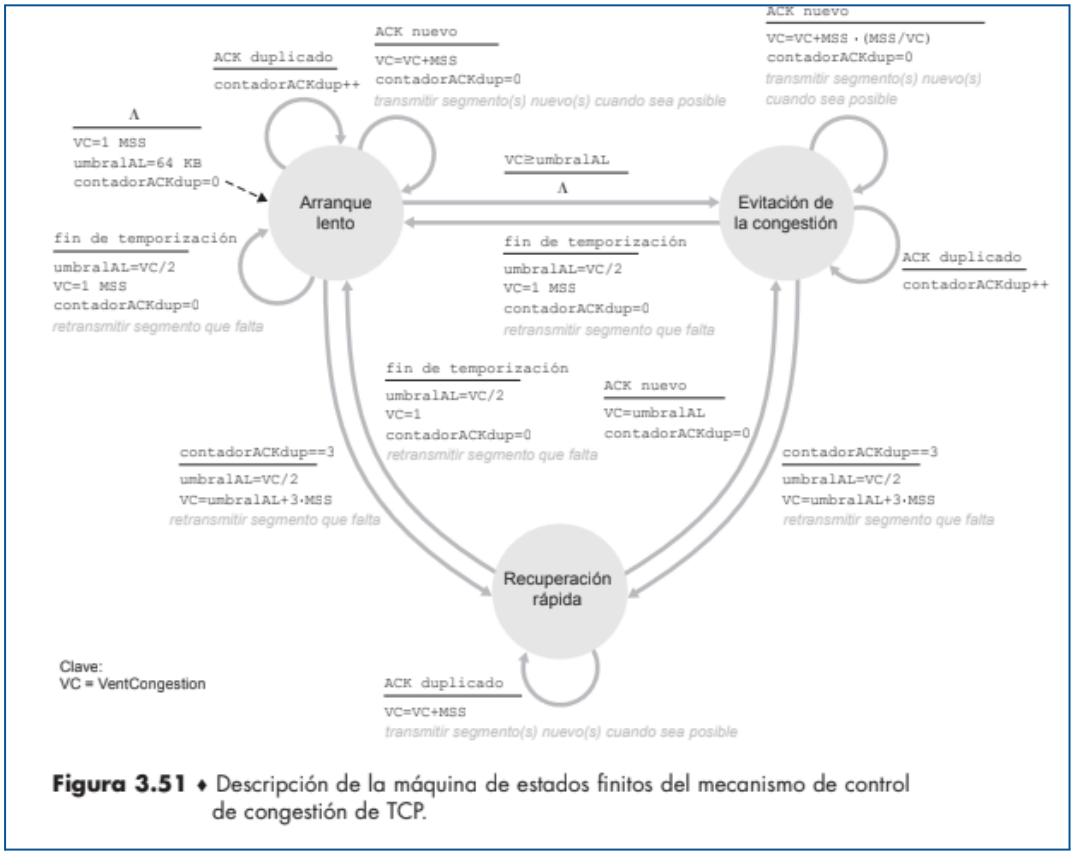
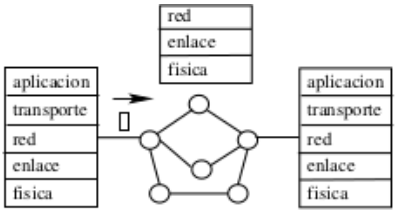


Figura 3.51 ♦ Descripción de la máquina de estados finitos del mecanismo de control de congestión de TCP.

Capa de red

Introducción



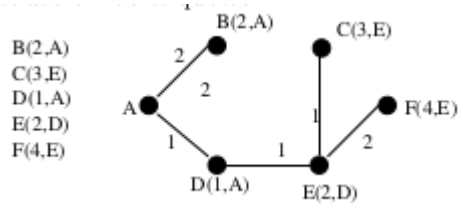
- La **capa de red** lleva los paquetes del host origen al host destino.
- En internet se denomina **IP**:
  - Es de tipo datagrama → encamina los paquetes en función de su dirección de destino, que consulta con la tabla de rutas
  - Sin estado: Cada paquete es tratado independientemente de los previos
  - No fiable: No garantiza que los paquetes lleguen al destino
    - Los paquetes pueden llegar desordenados o no llegar
  - Facilidad de interconexión: permite facilmente conectar tecnologías diferentes (Ethernet, inalámbricas, telefónicas…)

Algoritmos de rutado

- Encargado de encontrar el camino mínimo entre origen y destino.
- Problema equivalente a encontrar camino mínimo en un grafo, donde los routers son los nodos.
  - A los enlaces se les asigna un coste que puede depender de su velocidad o carga actual.
- **Tipos:**
  - Globales (EE, de estado de los enlaces): Se dispone previamente de toda la información de los enlaces.
    - Cada nodo puede computar por sí solo su tabla de rutas
  - Descentralizados (VD, de vector de distancias): Todos los nodos colaboran para calcular los caminos mínimos.
    - Cada nodo sólo conoce la distancia a sus vecinos, la cual comparten con los demás.
  - Estáticos si sólo se puede cambiar manualmente la topología de red, dinámicos si se ejecutan periódicamente de forma automática (internet)
  - Sensibles a la carga si el coste de los enlaces varía dinámicamente para reflejar su nivel de tráfico. Internet es insensible.

        - En los sensibles a la carga se producen oscilaciones: alta congestión → no se dirigen ahí paquetes → se reduce la congestión → pasan paquetes → …

Algoritmo de estado de enlaces (EE)



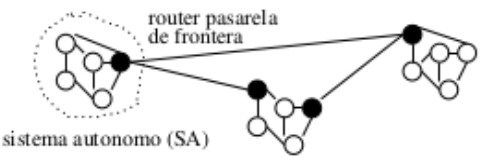
- Cada nodo calcula los caminos, conociendo toda la red. Ejemplo: Algoritmo de Dijkstra.
- En un momento dado los nodos son provisionales (conoce un camino, pero no necesariamente el más corto) o permanente (se conoce su camino mínimo)
- En cada iteración, los provisionales pasan a ser permanentes si se concoce su camino definitivo.
- En el ejemplo, tabla de rutas del nodo A.

destino	salida
B	B
D	D
C	D
E	D
F	D

Algoritmo de vector de distancias (VD)

- Todos los nodos colaboran en la obtención de caminos mínimos.
- En una serie de iteraciones, los nodos comunican a sus vecinos la información de distancias que han recopilado. Con esa información, computan distancias a nuevos nodos o actualizan distancias previamente calculadas.
- Características:
    - Si se cambia el coste de un enlace, la propagación de este cambio a los demás nodos se realizará en varias iteraciones.
        - Si disminuye el coste es muy rápido, si aumenta el coste tardan más en darse cuenta.
    - Más complejo que EE y menos robusto, pero más adecuado para redes con cambios frecuentes.

Rutado jerárquico



- Las redes grandes como internet se dividen en sistemas autónomos, operados por sistemas u organismos.
- Los routers conocen los detalles del encaminamiento sólo dentro de su región. El tráfico entre regiones se realiza entre routers pasarela de frontera.
- Dentro de cada región (intra-SA) se elige el protocolo de rutado que se quiera, pero el interautónomo debe ser común entre todos.

RIP (Protocolo de información de rutado)

- Rutado intra-SA de tipo VD.
- Considera que el costo de todos los enlaces es 1 y que la distanci máxima es 15 → sólo puede encaminar paquetes a través de hasta 15 routers intermedios.
- Los mensajes RIP se envían sólo a los vecinos.
    - Cuando un router quiere información envía a los vecinos un mensaje de petición RIP, y los routers responden con mensajes de respuesta RIP con una lista de hasta 25 redes destino y la distancia a ellas.
    - Los routers envían cada 30 segundos mensajes, y si uno no habla en 180 segundos se considera muerto.

OSPF (open Shortest Path First)

- Protocolo intra-SA de tipo EE, pensado como sucesor a RIP.
- Los mensajes se envían a todos los routers y cada uno tiene la información completa, por lo que ejecuta el algoritmo de Dijkstra.
    - Además, se envían HELLO para comprobar enlaces y solicitudes de info
- El coste de los enlace puede ser fijado por el administrador
- OSPF implementa un protocolo de transporte propio, ni TCP ni UDP, por lo que los mensajes van autenticados y sólo se envían a routers fiables.
- Las tablas de ruta pueden tener varios caminos alternativos.
- Permite que las SA se dividan a su vez en más áreas

destino	salidas
A	B, C
B	B, D

BGP (Border Gateway Protocol)

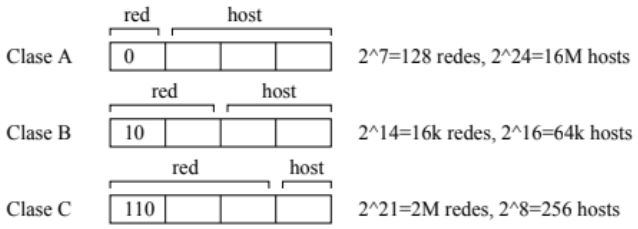
- Protocolo inter-SA estándar en internet. Comunica routers pasarela de frontera.
- Los routers se comunican entre si para intercambiar rutas. Pueden ser:
    - Entre dos routers BGP vecinos: pares BGP. Utilizan E-BGP.
    - Entre routers del mismo SA: vecinos lógicos. Utilizan I-EGP.
        - Aunque no estén conectados por un enlace se consideran vecinos.
- Similar a VD, pero intercambian rutas completas en lugar de distancias.
- Cada SA se identifica por su Autonomous System Number (ASN). Los facilita un organismo de Internet, y tipicamente cada ISP tiene su propio.
- Los administradores de la SA pueden seleccionar las políticas de rutado.

red destino	ruta
x	SA1/SA2/SA3/S4

IPv4

Direccionamiento IPv4

- Dirección IPv4: Número de 4 bytes en formato de número y puntos.
- Cada host y router tiene una Ip por interface[4]
- Parte de los bits identifican la red y son comunes dentro de ella, los demás identifican al host/router dentro de la red.
- Inicialmente se clasificaban en clases, A, B,C,D, comenzando por 0, 10, 110 y 1111 respectivamente (las 1111 están reservadas)
    - El byte 0 se reserva para referenciar a la red (172.168.0.0 = 172.168). El byte 255 también está reservado.



- Las clases se suprimieron en 1993, reemplazándose por las direcciones CIDR. Ahora se puede elegir manualmente cuantos bits forman las partes de red y hosts. Ej: x=20
    - Se indica mediante una **máscara** en la forma a.b.c.d/x, indicando que x bits son para identificar la red. Por ejemplo, para las clases previas, A→/8, B→/16, C→24.

193.144.48.0/20				
193	144	48		0
193	144	0011	0000	0
red (20 bits)			host	

Subredes

- Una red se puede dividir en **subredes** si se aumenta el tamaño del campo de red.
- Ejemplo: para dividir 193.144.48.0/20 en 8 subredes:
    - Queremos 8 subredes: necesitaremos aumentar el campo en 3 bits (2^3=8)
    - 193.144.48.0 → 193.144.[0011-**000**-0].0 Estos serán los 3 bits que diferencian cada subred.
        - 1ª subred: 193.144.[0011-**000**-0].0 → 193.144.48.0/23
        - 2ª subred: 193.144.[0011-**001**-0].0 → 193.144.50.0/23
        - 8ª subred: 193.144.[0011-**111**-0].0 → 193.144.62.0/23

[4] Interface: Unión de host o router con enlace. Ej: tarjeta ethernet, cable serie, cable USB, wireless...

Tabla de direccionamiento

- Un router guarda una tabla donde se asocia un conjunto de IPs de destino con la dirección del router al que enviar los paquetes destinados a esa IP. Ejemplo:

Destino	Máscara	Router
192.168.1.0	255.255.255.0	192.168.2.1
192.168.2.0	255.255.255.0	192.168.2.3
10.0,0,0	255.0.0.0	192.168.2.2
0.0.0.0	0.0.0.0	192.168.2.254

- Los paquetes para 192.168.1.x irán al router con IP 192.168.2.1
- Los paquetes para la red 10.x.x.x irán al router 192.168.2.2
- Los demás paquetes irán al router con 192.168.2.254
- **Agregación:** Combinar varias rutas en una sola entrada que las englobe todas para simplificar la tabla de enrutamiento.
    - Ej: en la tabla previa, reemplazar las dos primeras entradas por una que sea '192.168.0.0/22'.
- Si un paquete es válido para varias entradas escogerá la de mayor prefijo.

Ejemplo - Tabla de direccionamiento

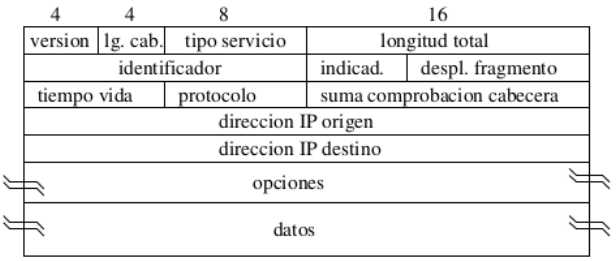
- **Suponer que un router ha construido la tabla de encaminamiento que se muestra a continuación. El router puede entregar paquetes directamente por las interfaces 0 y 1 o puede reenviar paquetes a los routers R2, R3 o R4. Asumir que el router busca la correspondencia con el prefijo más largo. Describir qué hace el router con un paquete dirigido a cada uno de los destinos siguientes:**

Subred	SiguienteSalto
128.96.164.0/22	Interfaz 0
128.96.170.0/23	Interfaz 1
128.96.168.0/23	R2
128.96.166.0/23	R3
⟨por defecto⟩	R4

- **a) 128.96.171.92**
    - 128.96.171.92 → 128.96.[1010 1011].92
    - 128.96.164.0 → 128.96.[1010 0100].0
        - Por la máscara /22, deben coincidir los 22 primeros bits. No coinciden. Probamos con la siguiente subred de la tabla.
    - 128.96.170.0 → 128.96.[1010 1010].0
        - Por la máscara /23, deben coincidir los 23 primeros bits, y coinciden.
        - Ningún prefijo de la tabla es mayor que 23 → no es necesario seguir comprobando.
        - El paquete dirigido a '128.96.171.92' se reenviará por la interfaz 1.
- **b) 128.96.167.151**
    - 128.96.167.151 → 128.96.[1010 0111].151
    - 128.96.164.0 → 128.96.[1010 0100].0
        - Por la máscara /22, deben coincidir los 22 primeros bits.
        - Coinciden, pero debemos probar si la dirección también coincide con una red con prefijo mayor.
    - 128.96.166.0 → 128.96.[1010 0110].0
        - Coincide, y el prefijo es mayor. El paquete se reenviará al router R3.

Cabeceras IPv4

- **Fragmentación:** Sirve si el tamaño de paquete es mayor que MTU. Cada fragmento llevará una cabecera IPv4.
    - Se vuelven a unir al llegar al destino.
    - Introduce complicaciones adicionales, se intenta evitar.
    - Tres campos se usan para hacer posible la fragmentación:
        - Identificador: Especifica a qué datagrama pertenece el fragmento
        - Indicador: 3 bits. NF a 1 indica 'no fragmentar', MF a 1 significa 'más fragmentos' (0 sólo en el último fragmento del datagrama)
        - Desplazamiento: indica pos. del fragmento dentro del datagrama original. **Se mide en bloques de 8 bytes**, y al dividir se debe redondear hacia abajo. (nota: es 0 en caso de segmentación TCP)
- **Numeración**: Dos niveles, uno para datagramas y otro para fragmentos.
    - Dado que se proporcionan 13 bits, hay un máximo de 2^{12} fragmentos por datagrama, dando una longitud máxima de datagrama de 65536 bytes.



- **Versión:** IPv4 o IPv6
- **Longitud de cabecera:** Mínimo de 5
- **Tipo de servicio:** Gran capacidad o alta fiabilidad
- **Tiempo de vida:** Inicialmente 255, all llegar a 0 se destruye
- **Protocolo:** TCP, UDP u otros.
- **Suma de comprobación de la cabecera:** Si hay errores el datagrama se desecha
- **Direcciones IP:** de origen y de destino
- **Opciones:** Ej: que se grabe la lista de nodos por lo que va pasando, incluir los nodos por los que queremos que pase.
    - El tamaño de la cabecera es variable por las opciones

Ejemplo - fragmentación IPv4, cálculo de offset

- **Un mensaje de 2048 bytes se envía con UDP de la subred 3 a la 1, pasando por la subred 2. MTU1=1700 bytes, MTU2=796 bytes, MTU3=1500 bytes. Indicar tamaños de los fragmentos y nº de fragmentos en subredes 3 y 1. Indicar el offset del último segmento.**

- Una cabecera UDP ocupa 8 bytes, que se suman al mensaje. Cada segmento llevará una cabecera IPv4 de 20 bytes.
- Subred 3: MTU=1500 bytes.
  - **Segmento 1:** 1480 bytes de mensaje (la cabecera UDP se considera parte del mensaje)
    - Tamaño: 1500 bytes
    - Offset: 0
  - **Segmento 2:** 2056-1480 = 576 bytes de mensaje
    - Tamaño: 576
    - Offset: 1480/8 = 185
- Subred 2: MTU=796 bytes.
  - **Segmento 1a**: 776 bytes de mensaje.
    - Tamaño: 796 bytes
    - Offset: 0
  - **Segmento 1b:** 1480-776=704 bytes de mensaje
    - Tamaño: 724 bytes
    - Offset: 756/8 (truncado) = 94
  - **Segmento 2:** no supera MTU, sigue igual
- Subred 1: MTU = 1700 bytes.
  - Ningún segmento supera MTU → se mantienen los mismos (total: 3 segmentos)

IPv6

Direccionamiento IPv6

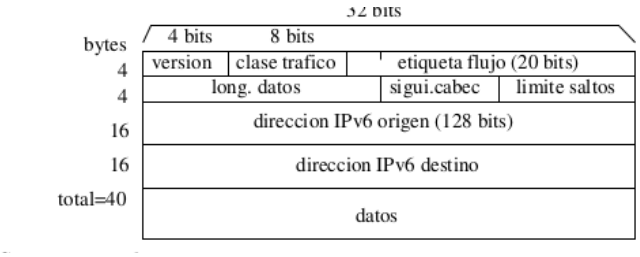
- Se amplia el nº de bits de 32 a 128, permitiendo más direcciones.
  - 128 bits → 16 bytes → 32 dígitos hexadecimales → 8 bloques de 4 dígitos
  - 1 0 más grupos de 4 ceros se abrevian como :: Ejemplos válidos:
    - `::0f53:6382:ab00:67db:bb27:7332`
      `[0:0:0f53:6382:ab00:67db:bb27:7332]`
    - `74dc::02ba [74dc:0:0:0:0:0:0:02ba]`
    - `::ffff:128.112.92.116` [IPv4 128.112.92.116, formato IPv6]
    - `::1` [dir. de loopback]

- Si hay varios grupos de 0s abreviados en posiciones distintas se debe indicar correctamente.
  - `::4ba8:95cc::db97:4eab` → incorrecto (ambiguo)
  - `::4ba8:95cc:0:0:db97:4eab`, `::4ba8:95cc:0:db97:4eab`, `::4ba8:95cc:0:0:0:db97:4eab`, `0:4ba8:95cc::db97:4eab`, etc → correcto

- Además de unicast (un host) y multicast (conjunto de hosts) se incluye anycast (el router entrega el paquete a un host cualquiera de la red), que reemplaza a broadcast (todos los hosts)
  - Las direcciones que comienzan por ff son multicast y el resto son unicast.

Cabeceras IPv6

- La cabecera está **simplificada:** Se elimina opciones → todas tienen el mismo tamaño
  - Se elimina tambien tam. cabecera por ser innecesario, fragmentación/resamblado (si es muy grande devuelve error al emisor), numeración de paquetes (no se numeran) y suma de comprobación (queda en la de transporte)



  - No existe fragmentación en capa de red: Si el mensaje supera la MTU, el router devolverá un ICMPv6 de error al emisor, que deberá fragmentar el mensaje correctamente.
  - Nuevo campo de etiquetado de flujo (misma etiqueta a los paquetes de un flujo de datos, audio o vídeo

- Campos que se mantienen: Versión (6), clase de tráfico (eq. a tipo de servicio), longitud de datos (no cuenta la cabecera), siguiente cabecera (protocolo de la siguiente cabecera), Límite de saltos (eq. a tiempo de vida)

IPv4	**IPv6**
32 bits (192.168.1.1)	128 bits (2001::1)
Fragmentación en origen y routers	Fragmentación solo en origen
Config. manual (DHCP/estática)	DHCP automática
Broadcast	Multicast/anycast
Cabecera variable	Cabecera simplificada y fija
Seguridad opcional	Seguridad obligatoria por defecto

Tunelización

- Permite la coexistencia de IPv4 y IPv6.
- Cuando un paquete IPv6 debe pasar por un router que no soporta IPv6, se encapsula dentro de un paquete IPv4, añadiéndole de forma temporal la cabecera.

VPN con túneles IP

- **VPN:** Red privada que utiliza la red pública para comunicarse de forma segura.
- Se pueden implementar mediante con túneles IP:
    - En la encapsulación, al paquete se le asigna un nuevo encabezado cuyas direcciones IP serán las de los extremos del túnel.
    - Además, los routers VPN en los extremos del tunel cifran el contenido usando un protocolo determinado (y el de destino lo descifra).
- De esta forma, todo el tráfico que viaje por internet pero dentro del túnel es privado y seguro.

ICMP (Internet control message protocol)

- Se emplea para que hosts y routers puedan informarse sobre errores o el estado de la red, o en comandos como *ping*.
- Funciona en la capa de transporte sobre el protocolo de la red Ip.
- Contienen:
    - Dos campos para tipo y código de mensaje
    - Primeros 8 bytes del datagrama que causó el envío del mensaje.
- Ejemplos típicos: destination host unreachable, expired TTL

Tipo ICMP	Código	Descripción
0	0	respuesta de eco
3	0	red destino inaccesible
3	1	host destino inaccesible
3	2	protocolo destino inaccesible
3	3	puerto destino inaccesible
3	6	red destino desconocida
3	7	host destino desconocido
4	0	apaciguar fuente
8	0	petición de eco
9	0	anuncio de router
10	0	descubrimiento de router
11	0	TTL espirado
12	0	mala cabecera

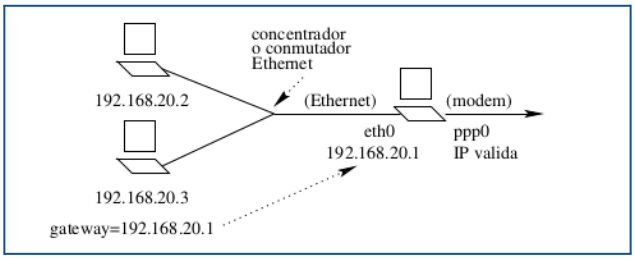
Distribución de IPs

- Tras asignarse un bloque de direcciones IP en la red, se pueden distribuir:
    - Estáticamente: Al instalar el SO, se indica la IP de cada host.
    - Dinámicamente: Utiliza DHCP.
        - Cada vez que el host arranca, el servidor le da una IP temporal. Se suele utilizar cuando los ISP no tienen IP para todos los abonados, o en conexiones inalámbricas.

DHCP (Protocolo de configuración dinámica de host)

- Descubrimiento de servidor: El cliente DHCP busca a un servidor DHCP en su red,
    - Se envía un mensaje de descubrimiento, con dirección de destino 255.255.255.255 (atienden todos los hosts) y origen 0.0.0.0
- Ofrecimiento de servicio: El servidor responde con su IP y máscara de red, junto con un tiempo de concesión de la IP (varias horas o días)
- Petición DHCP: Si hay varias ofertas, el cliente solicita una
- ACK DHCP: El servidor responde con un reconocimiento de la petición
- DHCP se encarga también de proporcionar información sobre cómo comunicarse con servicios de la red, como el **servidor DNS**.
    - Si un host quiere obtener la dirección del servidor DNS (y esta no se ha asignado manualmente en el host), es el protocolo DHCP el que la proporciona.

NAT (Traducción de direcciones de red)



- Traducción de direcciones de red: Si tenemos una única IP pero varios ordenadores en una casa, el servidor NAT actúa como router y traductor, cambiando de direcciones privadas a IP válidas.
- El servidor NAT tiene la dirección IP válida, y utiliza algun tipo de conexión (modem, ADSL…) para conectarse con el exterior.
- El resto de ordenadores tienen asignadas direcciones IP privadas. Una serie de conjuntos de direcciones IP vestán reservadas sólo para redes internas. Si algún paquete saliese de la red, sería ignorado por ser de una dirección reservada como interna.

Clase A	10.0.0.0/8
Clase B (varias)	172.16.0.0/12
Clase C (varias)	192.168.0.0/16

    - La dirección privada del NAT se les indica como gateway.
- El servidor NAT requiere dos interfaces y dos direcciones IP: una para el exterior (en el ejemplo, ppp0) y otra con una dirección privada para la red interna (eth0)
- Permite que varios hosts realicen consultas al DNS usando la misma IP pública.
- También toma nota del puerto origen para que, cuando llega una respuesta, se devuelve al ordenador adecuado.
    - Si varios ordenadores repiten puerto, el NAT los cambia para que sean únicos. Cuando recibe la respuesta, lo vuelve a cambiar para obtener el puerto original antes de cambiarse y le devuelve el mensaje.

Ejemplo - NAT

- **Desde un navegador en un PC de casa voy a la página whatismyip.com y me indica que la IP que estoy usando es 81.36.100.42. Desde un terminal ejecuto ifconfig o ip address show y me devuelve 192.168.1.109. También desde el terminal ejecuto dig www.cesga.es y me devuelve en la answer section 193.144.34.236. Desde el navegador accedo a www.cesga.es.**

- **Cuáles son la IP origen, puerto origen, IP destino y puerto destino en el datagrama una vez que ha dejado router de casa?**
- Una vez ha dejado el router utilizará la IP pública asignada por el NAT, que es la misma que sale al ir a 'whatismyip.com'.
    - IP origen: 81.36.100.42
    - Puerto origen: Un puerto libre del router de casa
    - IP destino: 193.144.34.236
    - Puerto destino: 80 (http) o 443 (https)

- **Cuáles son la IP origen, puerto origen, IP destino y puerto destino en el datagrama de respuesta que viaja por la red de casa?**
- Dentro de la red local, se usa la ip privada del host, que es la que sale al utilizar ifconfig.
    - IP origen: 193.144.34.236
    - Puerto origen: 80 (http) o 443 (https)
    - IP destino: 192.168.1.109
    - Puerto destino: Un puerto asignado por el SO

Capa de enlace

Introducción

- La **capa de enlace** transmite bloques de un nodo a otro de un enlace físico
    - Nodo**:** hosts, routers, switches… cualquier dispositivo que ejecute un protocolo de la capa de enlace.
    - Enlace**:** Canal de comunicación que conecta nodos adyacentes.
- El PDU (protocol data unit) de la capa de enlace se denomina trama o marco.
    - Una trama es un paquete de datos que almacena los datos útiles, direcciones MAC de origen/destino, control de flujo y errores y delimitadores.
- Implementada mayoritariamente en un **adaptador de red** o **tarjeta de interfaz de red** (hardware)

Servicios

- **Entramado:** Proceso de encapsular un datagrama de la capa de red en una trama.
- **Acceso al enlace**: Mediante un protocolo de acceso al medio (MAC, medium access control), se especifican las reglas para transmitir una trama por el enlace.
    - Único emisor y receptor → MAC simple o inexistente
    - Varios nodos comparten un enlace → MAC coordina las tramas entre nodos
- **Entrega fiable:** Similar a los servicios de la capa de transporte (como TCP).
    - Se usa en enlaces con muchos errores (inalámbricos)
- **Detección/corrección errores:** Usando los bits de control de errores de la trama

IEEE 802

- Conjunto de estándares que define las características y especificaciones técnicas para las redes LAN en cuanto a las funciones de la capa de enlace.
- Establece un enfoque para gestionar el acceso de los dispositivos a LANs de difusión (ethernet, wifi) para evitar colisiones.
- IEEE 802 divide la capa de enlace en dos subcapas:
    - Logical Link Control (LLC): Gestiona enlace de datos dentro de una misma red
        - Control de errores básico y multiplexación de varios protocolos

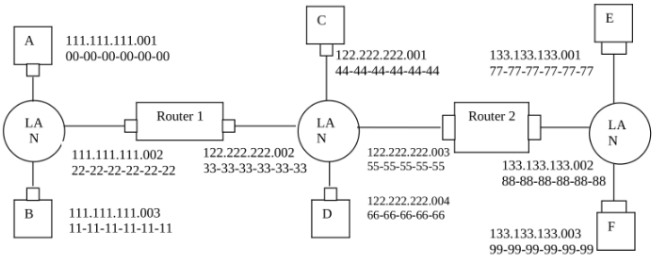
        - Puede ser sin conexión ni confirmaciones, sin conexión con confirmaciones o con ambas.
    - Medium Access Control (MAC): Gestiona acceso de los dispositivos al medio compartido
        - Determina cuándo los dispositivos pueden transmitir datos para evitar colisiones.
        - Ensambla los datos en tramas y los desensambla para leer las direcciones.

- Para un mismo LLC, están disponibles varios MAC

Direccionamiento de la capa de enlace

- Los hosts y routers, además de tener direcciones de la capa de red (ej: IP), mediante sus adaptadores de red tienen direcciones de la capa de enlace. Se suelen denominar **dirección MAC.**
    - Dentro de la LAN los adaptadores usan las MAC. Fuera, el paquete viaja usando la IP.
- Suelen tener 6 bytes de longitud, en hexadecimal: `00:08:74:4A:BA:4B`
- No se pueden repetir → cada fabricante tiene un prefijo único
- Direcciones especiales:
    - Todo 1s: broadcast, todos los nodos de la red
    - Bit menos significativo del primer byte a 1: multicast, grupo de nodos
- Si un nodo está en modo *promiscuo*, acepta cualquier trama.

Ejemplo: direccionamiento MAC



- **Supongamos que el host A envía un datagrama al host F. Indica las direcciones MAC origen y destino de la trama que contiene el datagrama a medida que se va transmitiendo:**
- **a) desde el host A al router R1,**
  - Dirección MAC origen: 00-00-00-00-00-00 (A)
  - Dirección MAC destino: 22-22-22-22-22-22 (R1)
  - IP origen: 111.111.111.001 (A)
  - IP destino: 133.133.133.003 (F)
- **b) desde el router R1 al router R2,**
  - Dirección MAC origen: 33-33-33-33-33-33 (R1)
  - Dirección MAC destino: 55-55-55-55-55-55 (R2)
  - IP origen: 111.111.111.001 (A)
  - IP destino: 133.133.133.003 (F)
- **c) desde el router R2 al host F.**
  - Dirección MAC origen: 88-88-88-88-88-88 (R2)
  - Dirección MAC destino: 99-99-99-99-99-99 (F)
  - IP origen: 111.111.111.001 (A)
  - IP destino: 133.133.133.003 (F)
- La dirección MAC cambia en cada router para indicar la próxima interfaz, la IP es constante para todo el envío.

ARP

- **Address Resolution Protocol:** Permite traducción entre direcciones de capa de red y direcciones MAC. (en concreto obtener MAC a partir de IP)
- Cuando ARP recibe una IP (y no está en la tabla caché):
  - ARP emite una trama.
    - IP y MAC de origen las del cliente.
    - IP destino la que se quiere traducir, MAC destino FF:FF…(broadcast)
  - El adaptador con esa IP responde con su MAC.
- El protocolo ARP mantiene una tabla caché con correspondencias, suelen durar 15 min.

Ethernet

- Red de tipo LAN más sencilla y común.
- No fiable, de difusión (topología de estrella → todos los adaptadores conectados con el centro).

Trama MAC Ethernet

- **Cabecera (7 bytes):** 10101010 7 veces. Seguido por **SFD(1):** 10101011, indicando el comienzo real de la trama.
- **Dirección de destino (6) y origen (6)**: Direcciones MAC, 6 bytes cada una
- **Longitud del campo de datos (2):** En Ethernet DIX, indica protocolo de red.
- **Datos LLC(46-1500)**: Si no se llega al mínimo (46) se completa con relleno.
- **FCS(4):** Frame Check Sequence.
    - Código CRC (comprobación de redundancia cíclica) de 4 bytes que permite que el receptor detecte los errores de bit-

Protocolo MAC en Ethernet

- Se usa **CSMA/CD**: Acceso múltiple por detección de portadora con detección de colisión.
- Todos los adaptadores escuchan continuamente el cable.
- Para transmitir, el adaptador escucha y espera a que esté libre.
    - Aun así, se pueden producir colisiones si coinciden dos señales de datos.

Resolución de colisiones

- Para garantizar que las colisiones se pueden detectar se debe cumplir que $\mathbf{t_{trama} >= 2*t_{prop}}$
    - Es decir, el tiempo que tarda en transmitirse una trama competa debe ser mayor que el que tarda una señal en viajar de un extremo a otro del enlace.
    - De esta forma, se puede detectar una colisión antes de que se termine de transmitir la trama colisionada.
- Otra forma de expresar la relación: $\mathbf{L_{min} >= R*RTT}$
    - $\mathbf{L_{min}}$**:** Tamaño mínimo[5] de una trama Ethernet, definido por el estándar como 64 bytes.
    - **R:** Velocidad de transmisión en bits por segundo.
    - **RTT:** Round trip time, tiempo que tarda una señal por la distancia máxima de la red y volver. $2*d / v_{prop}$

[5] El tamaño máximo de una trama Ethernet estándar es fijo y no depende de la distancia entre nodos. En Ethernet tradicional (10 Mbps), este tamaño es de **1,518 bytes (12,144 bits)**, incluidos los encabezados y datos.

Ejemplo - tamaño mínimo de trama

- **Suponer que hay cuatro nodos conectados a un concentrador mediante enlaces Ethernet a 10 Mbps. Las distancia entre el concentrador y estos cuatro nodos son 300, 400, 500 y 700 metros respectivamente. La velocidad de propagación de la señal es de 2 x 108 m/s. ¿Cuál es el tamaño mínimo de trama requerido? ¿Cuál es el tamaño máximo de trama requerido?**

- Al ser un concentrador, cualquier transmisión a un nodo se transmite por todos los enlaces, por lo que el tamaño de trama mínimo debe ser suficiente para todos los enlaces. El tamaño mínimo de trama requerido dependerá de la distancia máxima que debe recorrer una trama entre todos los potenciales casos. En este caso, consideramos la distancia más grande de las 4 (700 metros).
- Para asegurar la detección de colisiones se debe cumplir que el tiempo que tarda en transmitirse una trama competa debe ser mayor que el que tarda una señal en viajar de un extremo a otro del enlace. Entonces, Lmin >= r * RTT, siendo:
  - Lmin: tamaño mínimo de trama
  - r: 10^7bps
  - RTT: 2 * 700 / (2*108) = $7*10^{-6}$ s
- Obtenemos que el tamaño mínimo de trama requerido es de 70 bits, o de 9 bytes. El tamaño máximo de trama en Ethernet es de 1518 bytes y no depende del enlace.

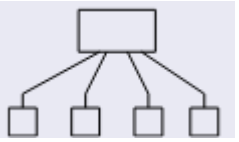
Resolución de colisiones

- Cuando un nodo detecta una colisión:
- Acaba de transmitir la cabecera de la trama
- Emite una jamming sequence de 32 bits y detiene la transmisión
- Usa el **Algoritmo de espera exponencial binaria** o **exponential backoff:**
  - Divide el tiempo en ranuras discretas de longitud T=$2t_{prop\_max}$
  - Las estaciones responsables esperan cada una un tiempo aleatorio dentro de una lista de valores posibles para repetir el envío, realizando hasta 16 intentos.
    - El intervalo de tiempos posibles es entre 0 y $(2^n-1)T$, n=num colisiones, hasta llegar a un máximo de 1023T con 10 col.
  - A los 16 intentos, desiste e informa a capas superiores del fallo.

Tecnologías Ethernet

- **Topología bus (obsoleta):** Bus de cable coaxial con terminadores en los extremos
  - Hasta 4 repetidores
  - Todos los nodos conectados entre si, en serie

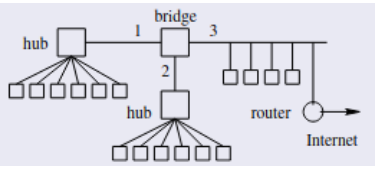
- **Topología estrella:** Par trenzado o fibra óptica conecta centro con nodos
    - Cada nodo tiene un par trenzado/fibra de entrada y otro de salida, a partir de 1000mpbs se usan 4 pares trenzados.
    - T→ par trenzado, F,S,L,E→fibra óptica
    - **Ejemplos:** 10base-T (10 Mbps, banda de base, 100m), 100base-FX (100 mbps, banda de base, 400m), 10Gbase-E (10Gigabit de fibra óptica)



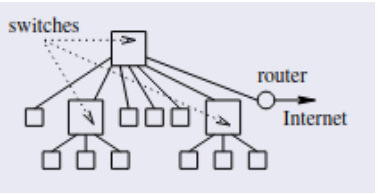
Dispositivos de capa física

- **Repetidores:** Dispositivo de capa 1 física que trabaja sobre bits individuales
    - 2 o más interfaces. Copia los bits que llegan por una interfaz al resto de interfaces
    - Reconstruye el pulso de tensión 
- **Concentradores (hubs):** Obsoletos. Misma función que repetidor de varias interfaces.
    - Puede informar de colisiones si recibe de varias interfaces a la vez.

Dispositivos de capa de enlace



- **Puentes / conmutadores (bridges/switches):** Trabajan a nivel de tramas Ethernet, no sobre bits.
- Procesan los campos, extraen dirección, revisan errores etc.
- Disponen de colas para la salida.
- Los switches tienen más interfaces y han dejado obsoletos a los puentes.



- Ambos poseen **autoaprendizaje:** Aprenden la localización de los adaptadores y crean una tabla de reenvío.
    - Inicialmente vacía. Al recibir una trama, obtienen la localización del adaptador por la interfaz de llegada y su identidad por la dirección origen.
    - En la tabla se guardan parejas de dirección MAC origen con su respectivo puerto.
    - Si se quiere enviar a una dirección que aún no está guardada, se enviará por todos los puertos excepto el de origen.
    - Se eliminan las entradas de más de unos minutos.

Routers

- Dispositivos de almacenamiento y reenvío, igual que los conmutadores
- Trabajan conc cabeceras IP en lugar de MAC
- Mantienen tablas de rutas e implementan algoritmos de encaminamiento.

VLAN

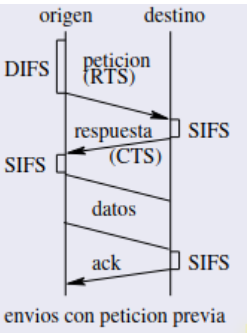
- **Virtual Local Area Network:** Permite segmentar una red física en varias redes lógicas independientes dentro de un mismo dispositivo.
    - Distintos grupos de dispositivos comparten el mismo medio físico, pero operan como si estuviesen en redes separadas.
- El switch mantiene una tabla de asignación de puertos a VLANs.
- El tráfico dentro de una VLAN no puede cruzar las fronteras sin pasar por un router de capa 3 (red). Este proceso se denomina encaminamiento inter-VLAN.
- El tráfico de broadcast o multicast se produce sólo dentro de la misma VLAN.
- Cualquier cambio en la configuración requiere reasignar los puertos a VLANs.

MPLS (Conmutación de etiquetas multiprotocolo)

- **T**ecnología de conmutación de paquetes utilizada para mejorar la eficiencia del tráfico de datos.
- En lugar de realizar el enrutamiento tradicional basado en direcciones IPs, MPLS asigna etiquetas a los paquetes.
- En el contexto de Internet se considera una tecnología de capa de enlace, como Ethernet, que facilita la interconexión de dispositivos IP.
- La **etiqueta** contiene información sobre cómo encaminar el paquete en la red.
    - Los paquetes de un mismo flujo de datos llevan la misma.
    - Cada router en el camino del paquete reemplaza su etiqueta.
    - Los algoritmos usados para calcular las rutas a partir de la etiqueta son extensiones de algoritmos como OSPF, y son específicos de cada fabricante.
- Los routers MPLs son más rápidos, son multiprotocolo y se usan comúnmente en VPNs.

WLAN (Red inalámbrica local)

- Basadas en el estándar **IEEE 802.11.** En cuanto a velocidad, 11b<11a<11g<11n<11ac
- Tipos:
  - Redes simples / ad-hoc: Conexión de igual a igual, comunican 2 estaciones
  - Redes distribuidas: LAN troncal cableada que conecta servidores con varios **puntos de acceso**, cada punto de acceso da servicio a varias estaciones móviles.
- Para acceso al medio se utiliza el protocolo **MACA** o CSMA/CA.



  - Para transmitir se sondea el medio y espera un intervalo de seguridad grande (DIFS).
    - Utiliza algoritmo de espera exponencial binaria.
  - Para asegurar la transmisión, se envía primero una trama de petición de envío (RTS) y el destino responde con una trama de reserva del canal (CTS).
    - En la RTS se incluye el tiempo necesario de ocupación. Las demás estaciones crean un timer NAV al que esperan antes de comprobar el canal.
  - Si dos estaciones envían RTS a la vez colisión, la detectan al no recibir CTS.

ATM (Modo de transferencia asíncrona)

- Tecnología de conmutación de paquetes diseñada para redes de alta velocidad.
- Características:
  - Celdas de tamaño fijo (53 bytes, 5 de header y 48 de carga útil)
    - Dos formatos distintos: interfaz usuario-red y red-red.
    - Cabecera: Control de flujo genérico para QoS (4 bits, sólo en usuario-red), identificador de canal virtual (VPI 8, VCI 16), Tipo de carga útil (3), Bit de prioridad de la celda, byte de control de error (8)
  - Optimizado para tráfico multimedia con alta calidad de servicio.
  - Orientado a conexiones
  - Escalabilidad eficiente, adecuado para transporte de datos a gran escala.
- Tipos de servicio:
  - CBR: Flujo de datos constante (ej: video/voz en tiempo real)
  - VBR: Flujo de datos variable
  - ABR: Flujo adaptado a ancho de banda disponible (ej: transferencia de archivos)
  - UBR: Tráfico de baja prioridad (ej: correo)
- Puede funcionar sobre cualquier capa física.
  - La capa de adaptación a ATM (AAL) permite que otros protocolos usen ATM.
- Dos **niveles de conexión**: VCC (circuito virtual) y VPC (camino virtual, conjunto de varios VCCs con los mismos extremos).