

# ME QUIERO MORIR: FR EDITION

## TEMA 1: INTRODUCCIÓN

### 1. Sistemas de comunicación y redes

- **Red:** es un conjunto de equipos informáticos y software conectados entre sí por medio de dispositivos físicos que envían y reciben impulsos eléctricos, ondas electromagnéticas o cualquier otro medio para el transporte de datos, con la finalidad de compartir información, recursos y ofrecer servicios.
- **Comunicación:** es la transferencia de información con sentido desde un lugar (remitente, fuente, originador, emisor) a otro lugar (destino, receptor).
  - **Fuente:** dispositivo que genera los datos a transmitir. (Ej. un teléfono o un PC)
  - **Transmisor:** por lo general, los datos los genera la fuente, pero no los transmite en el formato que los genera. El transmisor transforma y codifica esta información, normalmente en forma de señales electromagnéticas (EM) susceptibles de ser transmitidas a través de algún sistema de transmisión o medio.
  - **Canal de comunicación:** medio a través del cual se produce el envío de información (las señales EM por ejemplo). Puede ser una simple línea de transmisión, o una red compleja compuesta por diferentes tecnologías.
  - **Receptor:** elemento que recibe la información en forma de señal EM a través del canal de comunicación. El receptor transforma esta señal de manera que el destino pueda interpretar de manera correcta el contenido de dicha información.
  - **Destino:** Último elemento que interviene en el proceso de comunicación. Es el encargado de tomar los datos procesados por el receptor (e interpretarlos internamente).

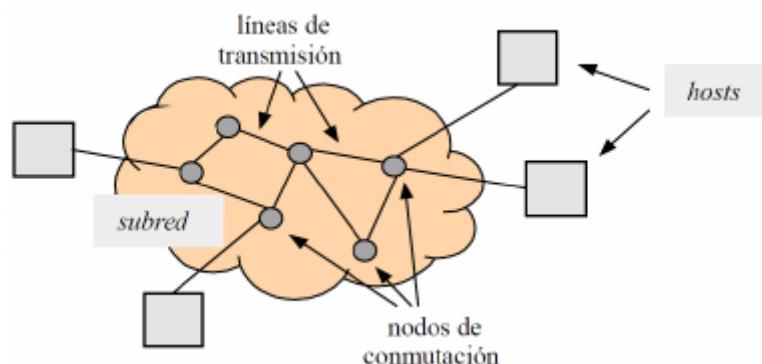


- **Información:** es un patrón físico al cual se le ha asignado un significado comúnmente acordado. El patrón debe ser único (separado y distinto), capaz de ser enviado por el transmisor, y capaz de ser detectado y entendido por el receptor.
- **Tareas de un sistema de comunicación:**
  - Uso eficiente del sistema de transmisión
  - Implementar (una interfaz con el canal)
  - Generación de la señal (compatible con el canal)
  - Formato de mensajes (estructura conocida)
  - Sincronización de emisor y receptor
  - Gestión del intercambio (colaborar para inicializar/finalizar la comunicación)
  - Detección y corrección de errores
  - Control de flujo (mecanismos para evitar la saturación)
  - Direcciónamiento (identidad del destino)
  - Encaminamiento (ruta hacia el destino)

- Recuperación ante pérdida de conexión
- Seguridad (evitar captura o alteración de datos)
- **De una red esperamos:**
  - Autonomía (capaz de procesar información)
  - Interconexión (mediante un sistema de comunicación)
  - Intercambio de información (con eficacia y transparencia)
- **Razones para su uso:**
  - Compartir recursos
  - Escalabilidad
  - Fiabilidad, robustez
  - Ahorro de costes

## Estructura general y elementos

- **Una red consta de:**
  - Hosts (máquinas finales)
    - Servidor, estación de trabajo, teléfono...
    - Ejecutan aplicaciones de red
    - **Son el borde/edge de la red**
    - Conectados a la red mediante enlaces de comunicaciones (cables, fibra, radio, satélite...)
  - Subredes/Comutadores (nodos de conmutación + líneas de transmisión)
    - Reenvían la info a través de rutas/paths dentro de la red
    - Son transparentes a los datos
    - Interconectados mediante enlaces de comunicaciones
    - **Forman el núcleo/core de la red**



- **Componentes:** elementos con funciones y características físicas específicas. Para elegirlos se debe tener en cuenta su función, contexto y economía:
  - **Servidor:** son computadoras que controlan las redes y se encargan de permitir o no el acceso de los usuarios a los recursos, también controlan los permisos que determinan si un nodo puede o no pertenecer a una red. La finalidad de los servidores es controlar el funcionamiento de una red. Los servicios que realice cada servidor dependerán del diseño de la red.
  - **Estación de trabajo:** computadoras conectadas a una red, pero que no pueden controlarla, así como a ninguno de los nodos o recursos de la misma. Cualquier computadora puede ser estación de trabajo, siempre que esté conectada y se comunique en la red.
  - **Nodo de red:** cualquier elemento que se encuentre conectado y comunicado a una red. Incluso los periféricos que se conectan a una estación de trabajo se convierten en nodo si están conectados a la red y pueden compartir sus servicios para ser utilizados por los demás usuarios, ejemplos: impresoras, discos.

- **Tarjeta de red:** tarjetas de circuito integrados que se insertan en módulos de expansión de la placa madre de un computador. Su función es recibir el cable que conecta a la computadora con una red informática.
- **Medios de transmisión:** cables que conectan a las computadoras, a través de los cuales viaja la información. Los cables son un componente básico en la comunicación entre computadoras.
  - **Cable coaxial:** está constituido por un hilo principal de cobre cubierto por una capa plástica y rodeada por una película reflectante que reduce las interferencias; alrededor de ella existe una malla de hilos metálicos y todo esto esta cubierto por una capa de plástico/goma que protege a los conductores de la intemperie.
  - **Cable par trenzado:** twisted pair. Cables de cobre, utilizados para la conexión de redes o entre nodos de la red. Tienen 4 pares de cables, y existen tres variantes:
    - **UTP:** unshielded TP. Es la variante más utilizada para la conexión de redes por su bajo costo, porque permite maniobrar sin problemas y porque no requiere herramientas especiales ni complicadas para la conexión de nodos en una red.
    - **STP:** shielded TP. Tiene una malla metálica que cubre cada uno de los pares de cables, que además están cubiertos por una película reflectante que evita/reduce las interferencias.
    - **FTP:** foiled. Los pares no tienen un aislamiento propio, como en STP, pero cuenta con una malla reflectante que cubre todo el conjunto. Es menos costoso que el STP, pero también menos efectivo, aunque da mejor rendimiento que UTP.
  - **Cable de fibra óptica:** es resistente a la corrosión y a las altas temperaturas y, gracias a la protección de su envoltura, es capaz de soportar mucha tensión en la instalación. Además tiene un índice de refracción específico que protege ante las interferencias (por así decirlo, si una interferencia fuera a chocar con el cable desde fuera esta no entraría dentro del cable corrompiendo la info a no ser que llegara justito con el ángulo de refracción necesario)
    - La desventaja de este cable es que su costo es elevado, ya que para su elaboración se requiere vidrio de alta calidad, además de ser sumamente frágil de manipular durante su fabricación.

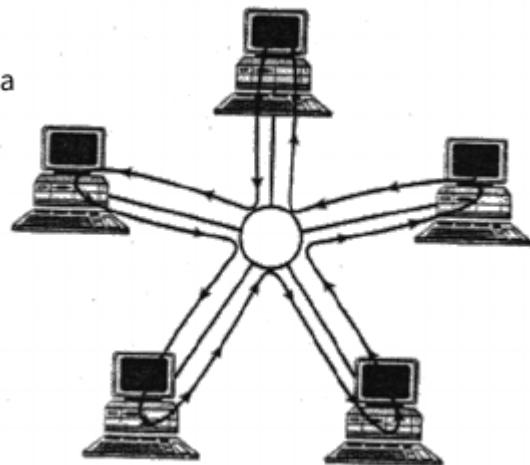
## Topología

Se llama topología de una red al patrón de conexión entre sus nodos, es decir, a la forma en que están interconectados los distintos dispositivos que la forman.

Puede ser física o lógica: la topología física no tiene por qué equivaler a la topología lógica. Por ejemplo, tres nodos dispuestos en bus y tres nodos enganchados a un switch son, a nivel lógico, lo mismo. Estas topologías que mencionamos, lo hacemos a nivel físico.

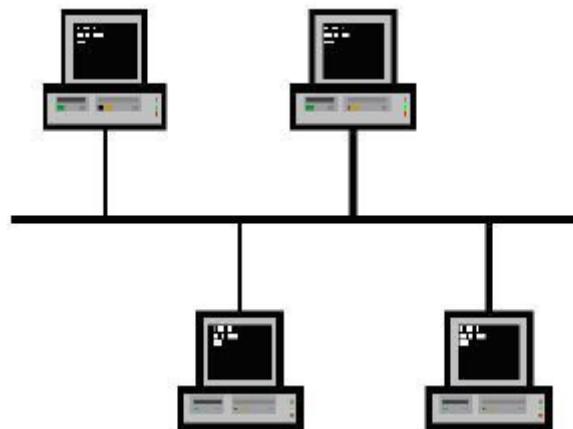
- Ejemplo:

- Topología física de los hosts en estrella
- Topología lógica en anillo



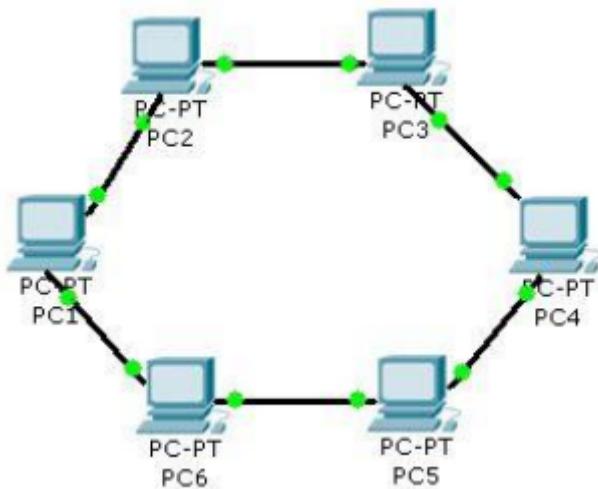
- **Topología en bus:** una red en forma de Bus o Canal de difusión es un camino de comunicación bidireccional con puntos de terminación bien definidos.

- Cuando un host (o estación) trasmite, la señal se propaga a ambos lados del emisor hacia todas las estaciones conectadas al Bus hasta llegar a las terminaciones del mismo.
- Cuando una estación trasmite su mensaje alcanza a todas las estaciones, por esto el Bus recibe el nombre de canal de difusión.
- Debe haber mecanismos de control de acceso al medio para que no haya colisiones en los datos.
- Muy barata pero si se rompe uno, se rompen todos los de detrás, si se cae el terminador se cae todo. También al transmitir se puede colisionar, y el ancho de banda es compartido

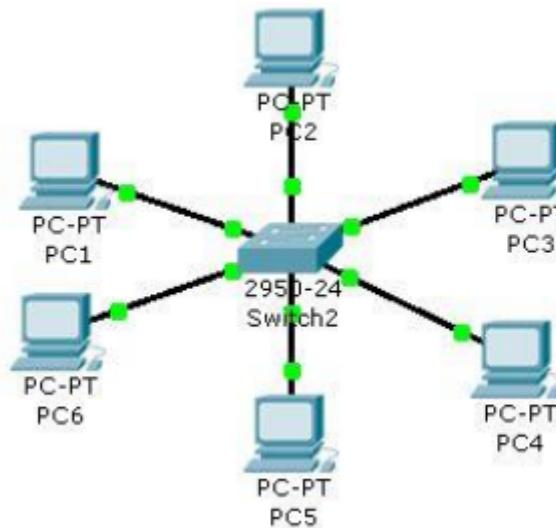


- **Topología en anillo:** esta topología se caracteriza por un definir camino unidireccional cerrado que conecta todos los nodos.

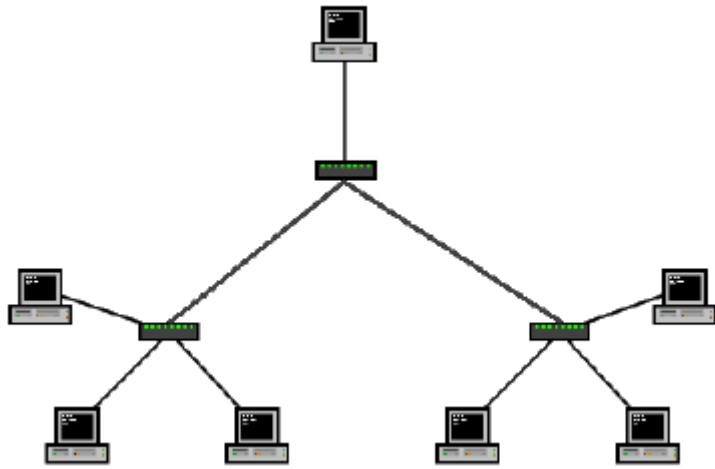
- Dependiendo del control de acceso al medio, se dan nombres distintos a esta topología. Por ejemplo: Bucle, se utiliza para designar aquellos anillos en los que el control de acceso está centralizado (una de las estaciones se encarga de controlar el acceso a la red).
- Poco eficiente, hay que recorrer todo el anillo para llegar a donde queremos.



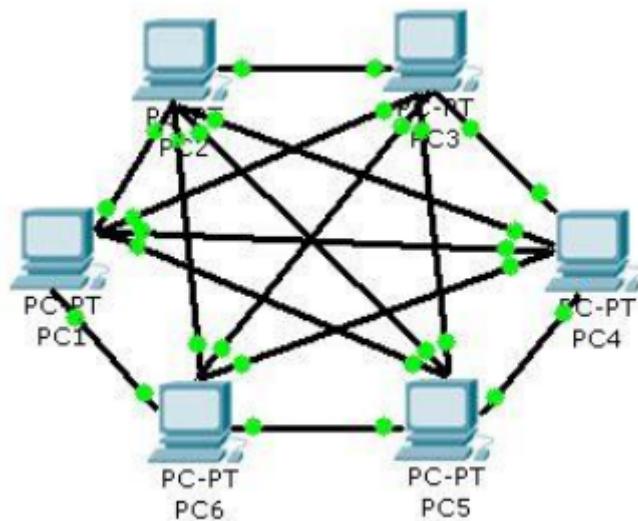
- **Topología en estrella:** esta topología se caracteriza por tener todos sus nodos conectados a un nodo central (controlador).
  - Todas las transmisiones pasan a través del nodo central, siendo éste el encargado de gestionar y controlar todas las comunicaciones.
  - Por este motivo, el fallo de un nodo cualquiera es fácil de detectar y no afecta al resto de la red, pero un fallo en el nodo central inutiliza la red completa.
  - Lo bueno es que es punto a punto y fácilmente escalable.



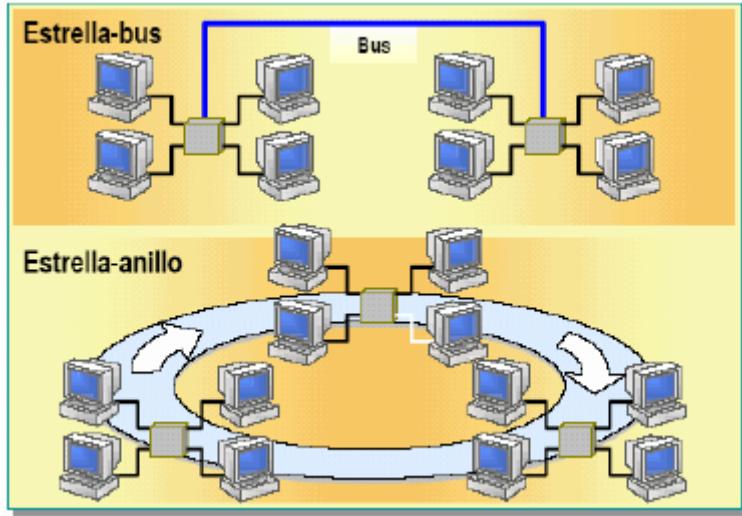
- **Topología en árbol:** esta topología es una variante de la topología en estrella. Como en la estrella, los nodos del árbol están conectados a un nodo central que controla el tráfico de la red. Sin embargo, no todos los dispositivos se conectan directamente al nodo central.
  - La mayoría de los dispositivos se conectan a un nodo secundario que, a su vez, se conecta al nodo central.
  - El acceso al nodo central es más lento, pero el funcionamiento de la red es más eficiente y además es más robusto ante errores.



- **Topología en malla:** en esta red cada nodo está conectado a todos los demás nodos de la red.
  - Esta configuración provee redundancia porque si un cable falla hay otros que permiten mantener la comunicación.
  - Es muy costosa por el gran despliegue de cables que hay que hacer.
  - Se suele combinar con otras topologías formando topologías híbridas.



- **Topología híbrida:** es una de las topologías más frecuentes y se deriva de la unión de varios tipos de topologías de red, de aquí el nombre de híbridas.
  - En una topología híbrida, se combinan dos o más topologías para formar un diseño de red completo que aproveche las ventajas de cada una de ellas.
  - Raras veces se diseñan redes considerando un solo tipo de topología.
  - Es importante asegurar que, si un nodo falla, no afecte al resto de la red.



## Clasificación de redes

- **Según tamaño y extensión:**
  - PAN - Personal Area Network (smartwatch, bluetooth)
  - LAN - Local Area Network
    - Redes de ordenadores cuya extensión es del orden de entre 10 metros a 1 kilómetro. Son redes pequeñas, habituales en oficinas, colegios y empresas pequeñas, que generalmente usan la tecnología de broadcast, es decir, aquella en que a un sólo cable se conectan todas las máquinas.
    - Como su tamaño es restringido, el peor tiempo de transmisión de datos es conocido, siendo velocidades de transmisión típicas de LAN las que van de 10 a 100 Mbps (Megabits por segundo).
  - MAN - Metropolitan Area Network (del orden de una ciudad, ej. UGR, muchas sedes)
    - Redes de ordenadores de tamaño superior a una LAN, soliendo abarcar el tamaño de una ciudad. Son típicas de empresas y organizaciones que poseen distintas oficinas repartidas en un mismo área metropolitana, por lo que, en su tamaño máximo, comprenden un área de unos 10 kilómetros
  - WAN - Wide Area Network (operadora)
    - Tienen un tamaño superior a una MAN y consisten en una colección de host o de redes LAN conectadas por una subred. Esta subred está formada por una serie de líneas de transmisión interconectadas por medio de routers, aparatos de red encargados de rutear o dirigir los paquetes hacia la LAN o host adecuado, enviándose éstos de un router a otro. Su tamaño puede oscilar entre 100 y 1000 kilómetros.
    - MAN y WAN son el mismo concepto aplicado a distintas escalas
- **Según tecnología de transmisión:**
  - **Broadcast/difusión:** la transmisión de datos se realiza por un sólo canal de comunicación, compartido entonces por todas las máquinas de la red. Cualquier paquete de datos enviado por cualquier máquina es recibido por todas las de la red.
  - **Punto a punto:** aquellas en las que existen muchas conexiones entre parejas individuales de máquinas. Para poder transmitir los paquetes desde una máquina a otra a veces es necesario que éstos pasen por máquinas intermedias, siendo obligado en tales casos un trazado de rutas mediante dispositivos routers.
- **Según tipo de transferencia de datos:**
  - **Simple:** los datos solo pueden viajar en un sentido

- **Half-duplex:** bidireccional asíncrono, los datos pueden viajar en ambos sentidos, pero sólo uno de ellos en un momento dado. Es decir, solo pueden haber transferencias en el mismo sentido a la vez.
- **Full-duplex:** bidireccional, pueden viajar en ambos sentidos a la vez. Más complejo porque hay que tener en cuenta colisiones (se destruye y nadie recibe el mensaje)

## 2. Diseño y estandarización de redes

- **Problemas a resolver por la red:**

- ¿Cómo enviar físicamente la información?
- Compartición del medio
- Segmentación de la información
- Control de flujo y de errores, en el enlace y también extremo a extremo
- Control del encaminamiento (enrutamiento) de los mensajes
- Control de congestión
- Entrega ordenada de los mensajes
- Gestión del diálogo o turno de palabra
- Representación (sintaxis) de los datos
- Significado (semántica) de los datos

### Modelo de referencia OSI

El modelo OSI (Open System Interconnection) es utilizado por prácticamente la totalidad de las redes del mundo. Este modelo fue creado por el ISO (Organización Internacional de Normalización), y consiste en siete niveles o capas donde cada una de ellas define las funciones que deben proporcionar los protocolos con el propósito de intercambiar información entre varios sistemas.

Esta clasificación permite que cada protocolo se desarrolle con una finalidad determinada, lo cual simplifica el proceso de desarrollo e implementación. Cada nivel depende de los que están por debajo de él, y a su vez, proporciona alguna funcionalidad a los niveles superiores. **Cada capa maneja un tipo de datos o PDU (Protocol Data Unit).**

Este modelo de referencia...

- Define capas y funcionalidades
- Funciones distintas deben estar en capas distintas
- Minimizar el flujo de información entre capas



- **Capa física:** se encarga de las conexiones físicas hacia la red en lo que se refiere al medio físico; características del medio y la forma en la que se transmite la información. **Se encarga de transformar una trama de datos proveniente del nivel de enlace en una señal adecuada al medio físico utilizado en la transmisión.** Dicha señal podrán ser impulsos eléctricos (transmisión por cable) o electromagnéticos (transmisión sin cables).

Sus principales funciones:

- Definir el medio físico por el que va a viajar la comunicación: cable de cobre, coaxial, guías de onda, aire, fibra óptica.
- Definir las características materiales (componentes y conectores) y eléctricas (niveles de tensión) que se van a usar en la transmisión de los datos por los medios físicos.
- Definir las características de la interfaz (alimentación, mantenimiento y liberación del enlace físico).
- Transmitir el flujo de bits a través del medio.
- Manejar las señales eléctricas/electromagnéticas.
- Garantizar la conexión (aunque no la fiabilidad de esta).

■ Su PDU es flujo de bits

- **Capa enlace:** se encarga proporcionar una transmisión sin errores, es decir, un **tránsito de datos fiable a través de un enlace físico.** Debe crear y **reconocer los límites de las tramas y resolver los problemas** derivados del deterioro, pérdida o duplicidad de las mismas.

La capa de enlace de datos se ocupa de:

- El direccionamiento físico.
- Topología de la red.
- Acceso a la red.
- Notificación de errores.
- Distribución ordenada de tramas.
- Control del flujo.

■ Su PDU son tramas

- **Capa red:** tiene como objetivo **hacer que los datos lleguen desde el origen al destino**, aún cuando ambos no estén conectados directamente. Los dispositivos que facilitan tal tarea se denominan routers o enrutadores.

La capa de red lleva un **control de la congestión de red**, la cual se produce cuando uno o varios nodos se saturan (al recibir demasiados paquetes), pudiendo quedar inutilizados ellos e incluso una parte de (o toda) la red. En este nivel se realiza el direccionamiento lógico y la **determinación de la ruta** de los datos hasta su receptor final.

■ Su PDU son paquetes

- **Capa transporte:** tiene como función básica **aceptar los datos enviados por las capas superiores, dividirlos en pequeñas partes si es necesario, y pasarlo a la capa de red.** En el caso del modelo OSI, también se asegura que **lleguen correctamente** al destino de la comunicación. Se encarga del transporte de los datos al destino independientemente de la red subyacente.

Es la primera capa que lleva a cabo la comunicación extremo a extremo (que se mantendrá en las capas superiores).

■ Dependiendo del protocolo la PDU se denominará de una forma:

- Segmento (TCP)

- Datagrama (UDP)
- **Capa sesión:** establece, gestiona y finaliza las conexiones entre usuarios (procesos o aplicaciones) finales. **Mantiene y controla el enlace establecido entre dos computadoras** que están transmitiendo datos.

Ofrece varios servicios muy importantes para la comunicación, como:

- Control de la sesión a establecer entre el emisor y el receptor (quién transmite, quién escucha y el seguimiento de ésta).
- Control de la concurrencia (que dos comunicaciones sobre la misma operación crítica **no se efectúen al mismo tiempo**).
- Mantener puntos de verificación que sirven para que, ante una interrupción de transmisión por cualquier causa, la misma **se pueda reanudar desde el último punto de verificación** en lugar de repetirla desde el principio.

Su PDU es SPDU

- **Capa presentación:** se encarga de la representación de la información, de manera que **aunque distintos equipos puedan tener diferentes representaciones internas** de caracteres (ASCII, Unicode, EBCDIC), números, sonido o imágenes, **los datos lleguen de manera reconocible a otros equipos**.

Esta capa es la primera en trabajar sobre el contenido de la comunicación. En ella se tratan aspectos tales como la semántica y la sintaxis de los datos transmitidos, ya que distintas computadoras pueden tener diferentes formas de manejarlos. Esta capa también permite cifrar los datos y comprimirlos, por ejemplo.

Su PDU es PPDU

- **Capa aplicación:** ofrece a las aplicaciones la posibilidad de **acceder a los servicios del resto de capas**. Define los protocolos que utilizan aplicaciones para intercambiar datos como por ejemplo:

- Correo electrónico (POP y SMTP).
- Gestores de bases de datos y servidor de ficheros (FTP).
- Muchos más...

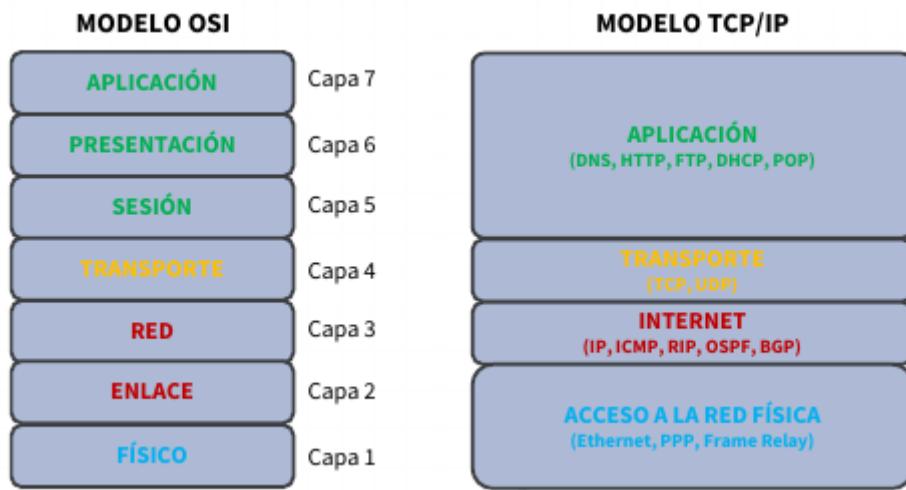
Hay casi tantos protocolos como aplicaciones distintas y, debido a que las redes están en continuo crecimiento y mejora de prestaciones, se desarrollan nuevas aplicaciones y, con ellas, nuevos protocolos. Debemos tener en cuenta que **el usuario normalmente no interactúa directamente con el nivel de aplicación**, sino que utiliza programas que, a su vez, interactúan con el nivel de aplicación pero haciéndolo transparente.

Su PDU es APDU

## Modelo TCP/IP

TCP/IP es el protocolo común utilizado por las computadoras conectadas a Internet, de manera que estas puedan comunicarse entre sí. En Internet se encuentran conectadas computadoras de clases muy diferentes y con hardware y software incompatibles en muchos casos. TCP/IP se encargará de que la comunicación entre ellas sea posible. TCP/IP es compatible con cualquier S.O. y con cualquier tipo de hardware.

TCP/IP no es un único protocolo, sino que en realidad lo que se conoce con este nombre es un conjunto de protocolos que cubren los distintos niveles del modelo OSI. Los dos protocolos más importantes son el TCP (Transmission Control Protocol) y el IP (Internet Protocol), que son los que dan nombre al conjunto de este modelo.



En Internet se diferencian cuatro niveles o capas en las que se agrupan los protocolos, y que se relacionan con los niveles OSI de la siguiente manera:

- **Aplicación:** se corresponde con los niveles de Aplicación, Presentación y Sesión. Se incluyen protocolos destinados a proporcionar servicios, tales como correo electrónico (SMTP), transferencia de archivos (FTP), conexión remota (TELNET) o páginas web (HTTP).
- **Transporte:** coincide con el nivel de transporte del modelo OSI. Los protocolos de este nivel, tales como TCP y UDP, se encargan de manejar los datos y proporcionar la fiabilidad necesaria en el transporte de los mismos.
- **Internet:** es el nivel de red del modelo OSI. Incluye al protocolo IP, que se encarga de enviar los paquetes de información a sus destinos correspondientes. Es utilizado con esta finalidad por los protocolos del nivel de transporte.
- **Acceso al medio:** los niveles OSI correspondientes son el de enlace y el nivel físico. Los protocolos que pertenecen a este nivel son los encargados de la transmisión a través del medio físico al que se encuentra conectado cada host, como puede ser una línea punto a punto o una red Ethernet.

## Organismos de estandarización de redes

- **ISO (Organización Internacional para la estandarización):** es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción, de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.
- **IEEE (The Institute of Electrical and Electronics Engineers):** es una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías.
- **IETF (Internet Engineering Task Force):** organización internacional abierta de normalización, que pretende contribuir a la ingeniería de Internet, actuando en áreas como transporte, encaminamiento o seguridad.

La IETF es mundialmente conocida por ser la entidad que regula las propuestas y los estándares de Internet, conocidos como RFC (Request For Comments). Sin ánimo de lucro y abierta a la participación de cualquier persona cuyo objetivo es velar porque la arquitectura de Internet y los protocolos que la conforman funcionen correctamente. Se la considera como la organización con más autoridad para establecer modificaciones de los parámetros técnicos bajo los que funciona la red.

### 3. Terminología y servicios

#### Comunicación OSI

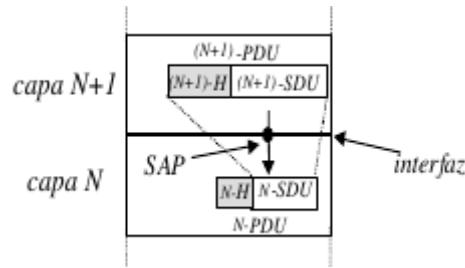
- Cada capa tiene tareas bien definidas.
- La comunicación se realiza entre dos capas adyacentes ( $N$ ) y ( $N+1$ )
  - Capa inferior ( $N$ ) → Proveedora de servicios
  - Capa superior ( $N+1$ ) → Usuaria de servicios

La capa  $N$  ofrece una serie de funciones o prestaciones (servicios) transparentes para la capa  $N+1$ . Ejemplo: La capa física es proveedora del servicio de transmisión eléctrica sobre el canal respecto a la de enlace, siendo esta la usuaria de dicho servicio.

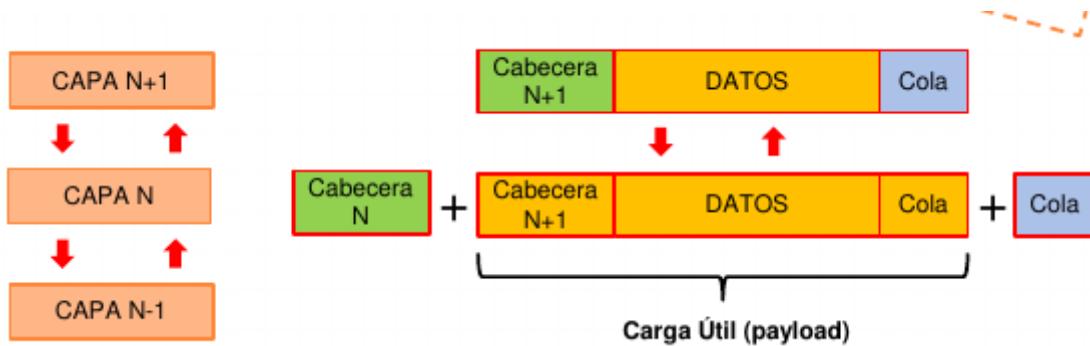


La interfaz son los mecanismos que permiten interaccionar a dos capas adyacentes

- Los elementos activos (hardware y software) de la capa  $N$ , reciben el nombre de entidades de nivel  $N$ .
  - Las entidades de nivel  $N$  en el emisor y receptor reciben el nombre de entidades pares o paritarias.
- Dos tipos de comunicación:
  - **Comunicación Real o Vertical:** intercambio de datos entre capas adyacentes en sentido descendente en el emisor y ascendente en el receptor.
  - **Comunicación Virtual u Horizontal:** comunicación observada desde el punto de vista de las entidades paritarias.
- **Protocolo:** conjunto de reglas a utilizar en una comunicación entre 2 entidades paritarias para llevar a cabo un servicio.
  - Se basan en el paso de mensajes que generan ciertas acciones por parte de las entidades sobre los datos.
  - Presentan una estructura concreta y bien definida.
- **Arquitectura de red:** Conjunto de capas + Protocolos asociados.
  - OSI no puede considerarse una arquitectura de red (no define protocolos asociados).
  - TCP/IP es una arquitectura de red → Pila de Protocolos
- La comunicación producida entre capas adyacentes se realiza a través de una interfaz de separación → **Punto de acceso al servicio (Service Access Point, SAP)**.
  - Información transmitida sobre los SAP entre 2 entidades:
    - **Unidad de datos de servicio (Service Data Unit, SDU)** → Datos manejados por la entidad y que proceden de la capa superior.
    - **Unidad de datos del Protocolo (Protocol Data Unit, PDU)** → SDU recibida de la capa superior más la cabecera.

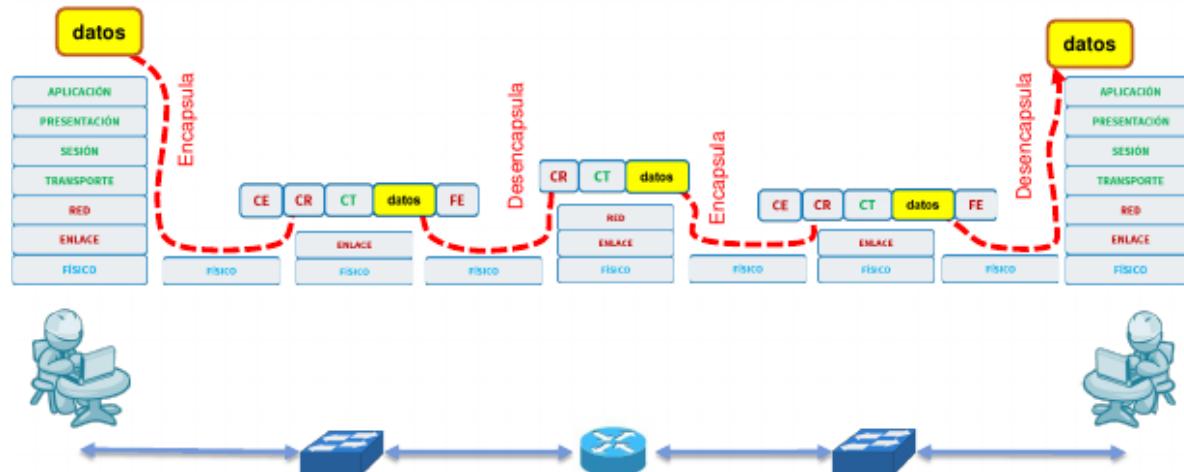
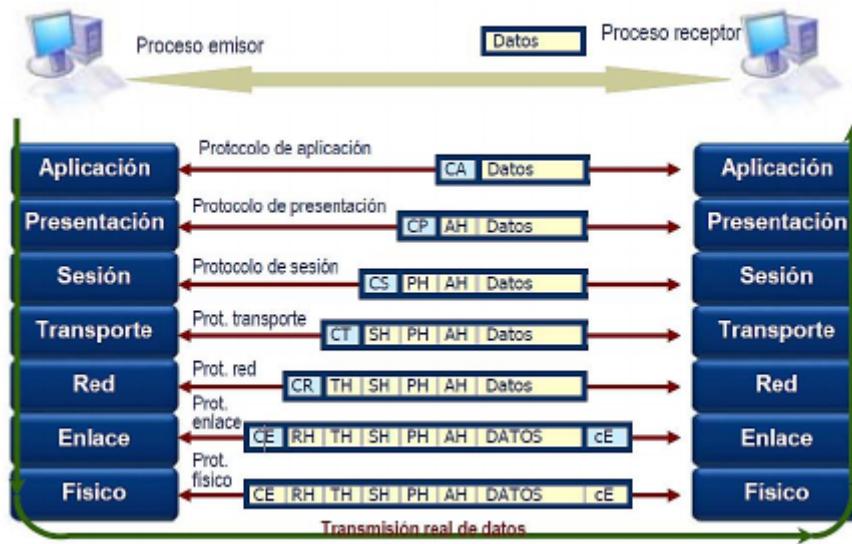


- A excepción de la capa física, el resto de capas añaden/eliminan información suplementaria (cabeceras + colas) para permitir la comunicación coherente entre entidades paritarias . Esto se conoce como **encapsulado** (o encapsulamiento) de datos.



El PDU de una capa (incluyendo su cabecera/cola) se convierte en los datos de la inferior

La transmisión real de datos se hace de arriba a abajo en el emisor y de abajo a arriba en el receptor:



## MTU (Maximum Transfer Unit)

¿Qué pasa cuando tenemos un paquete demasiado grande para ser enviado a través de la red?

Cada tecnología tiene un tamaño máximo de tramas que puede transmitir. En un router, host, conmutador, etc, cada interfaz tiene un valor de MTU concreto, que depende del tipo de interfaz por la que se vayan a transmitir los datos.

Protocolo a nivel de enlace	MTU (bytes)
PPP (valor por defecto)	1500
PPP (bajo retraso)	296
SLIP	1006 (límite original)
X.25	1600 (RFC 1356)
Frame relay	1600 normalmente (depende de la red)
SMDS	9235
Ethernet DIX	1500
Ethernet LLC-SNAP	1492
IEEE 802.4/802.2	8165
Token Ring 16 Mb/s	17940 (token holding time 8 ms)
Token Ring 4 Mb/s	4440 (token holding time 8 ms)
FDDI	4312
Hyperchannel	65535
Classical IP over ATM	9180

- **MTU grande:**

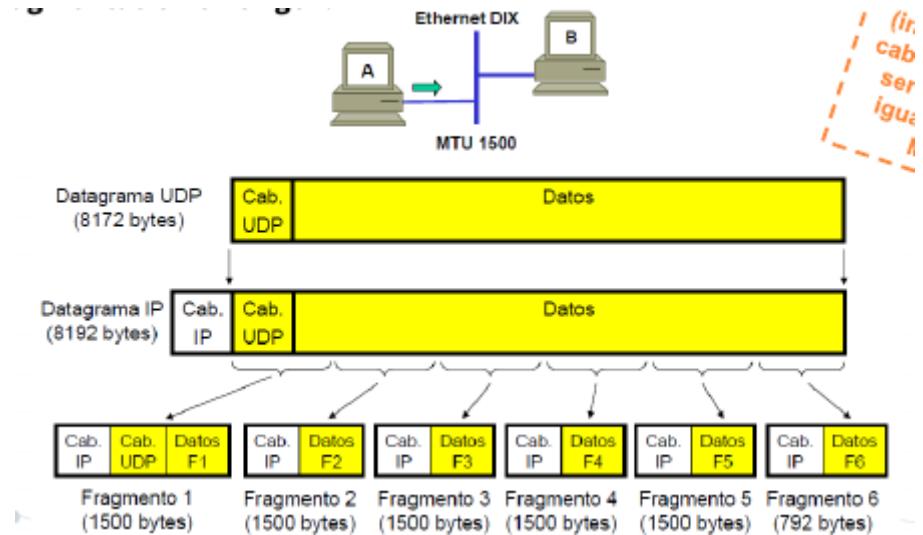
- Pros
  - Mejora en la eficiencia de comunicación y reduce la sobrecarga en la red (menor ancho de banda (BW) en el envío de cabeceras).
  - Reduce la carga de CPUs de los dispositivos, porque procesan menos paquetes.
- Contras
  - Mayores buffers (para almacenar los paquetes recibidos antes de procesarlos).
  - Si se pierden paquetes por error o congestión, la perdida de información es mayor.
  - En líneas de baja capacidad, el envío de un paquete grande, bloquea una interfaz y puede generar problemas en el envío de tráfico prioritario.

## Fragmentación

Cuando enviamos un datagrama IP a través de una red (capa 3), esta información es "envuelta" en una trama del nivel de enlace (capa 2). Si el datagrama es demasiado grande (mayor de la MTU que se puede transmitir), se deberá dividir en trozos más pequeños para que "quepan" en la MTU disponible.

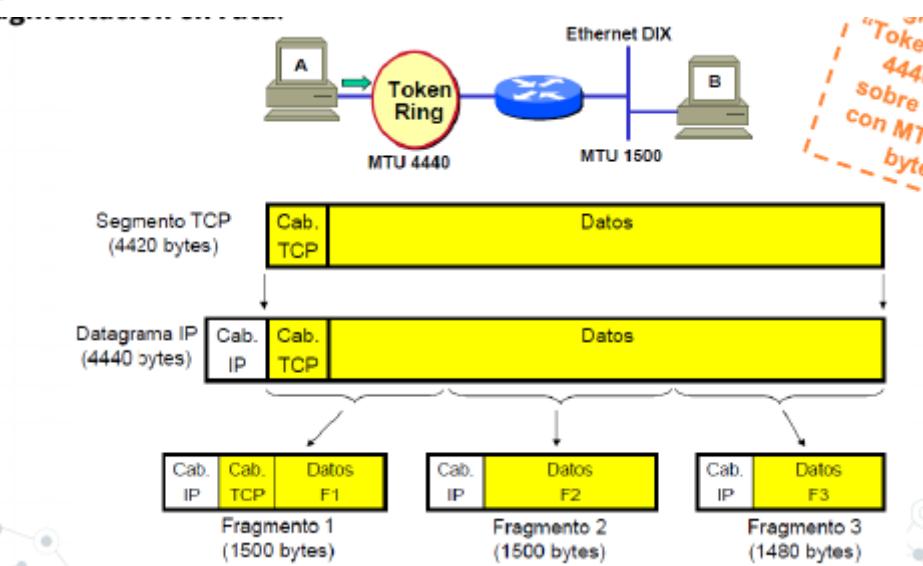
2 tipos de fragmentación:

- **Fragmentación en origen:** realizada por los hosts cuando pretenden enviar paquetes superiores a la MTU de la interfaz.



El tamaño de los fragmentos (cabecera incluída) debe ser menor o igual que el MTU

- **Fragmentación en ruta:** realizada por los routers cuando reciben un paquete más grande del que puede enviar a través de la MTU de la interfaz de salida.



Envío de un segmento TCP "Token ring" de 4440B sobre una red con MTU 1500B

## Retardos en la comunicación

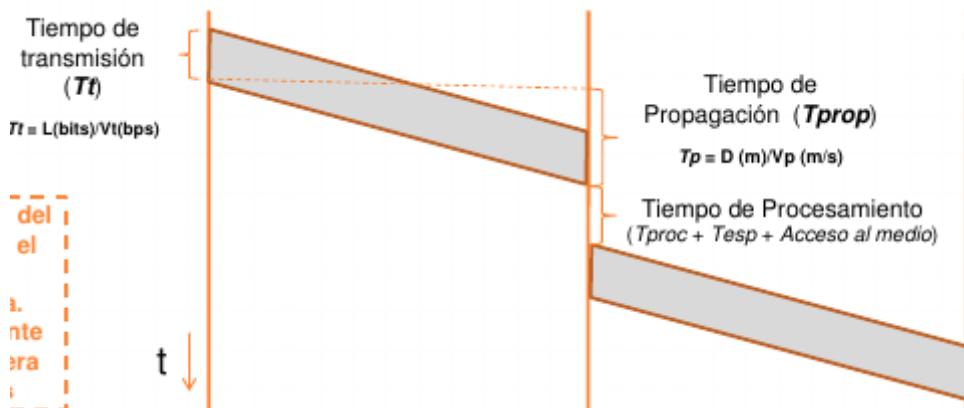
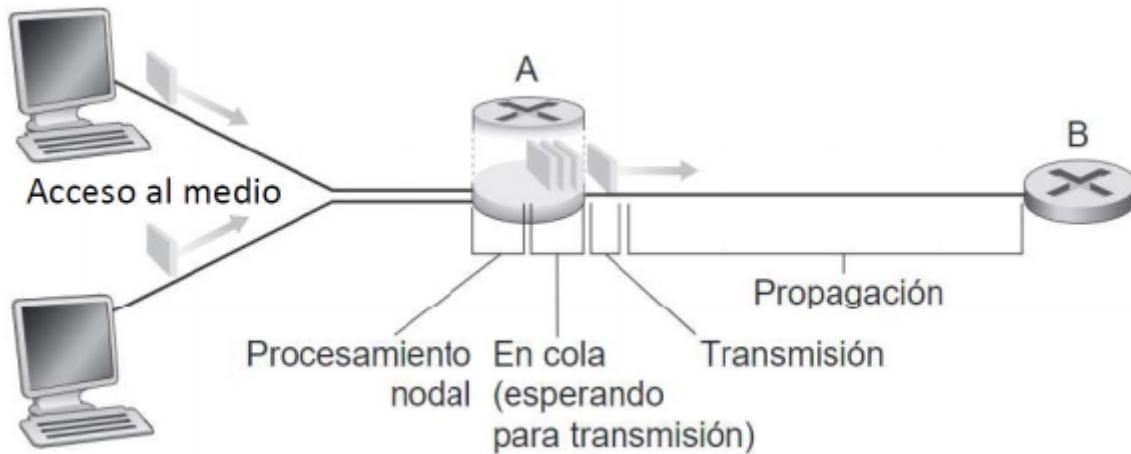
- **Tiempo de Propagación al siguiente nodo/host (Tprop):** depende de la distancia y del medio de transmisión (la velocidad que pueda ofrecer).

$$T_{prop} (\text{segundos}) = \text{Distancia (metros)} / \text{Velocidad prop (bps)}$$

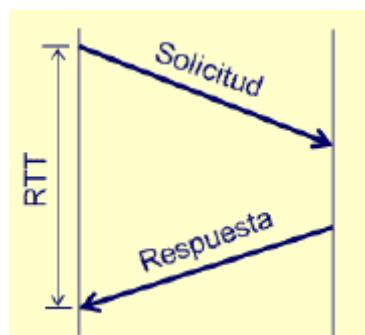
$$T_{prop} (\text{microsegundos}) = \text{Distancia (metros)} / \text{Velocidad prop (Mbps)}$$

- **Tiempo de procesamiento en los nodos(Tproc):** tiempo que se tarda en decidir qué hacer con el paquete (desencapsular e interpretar). Depende del tipo de nodo/router y de su carga. Normalmente despreciable
- **Tiempo de espera en la cola salida (Tesp):** depende del tráfico en la red.
- **Tiempo de transmisión (Tt):** depende de la velocidad del enlace y tamaño del paquete.

$$T_t (\text{segundos}) = \text{Longitud paquete (Bytes)} / \text{Velocidad transmisión cable (2*10}^8 \text{ m/s)}$$



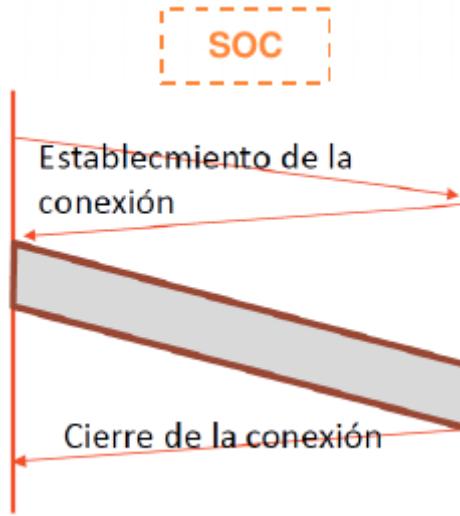
- **Round Trip Time (RTT):** tiempo para enviar un paquete y recibir su respuesta asociada. Está constituido por la suma de los retardos de cada uno de los enlaces utilizados (ida y vuelta) y el tiempo de proceso en el servidor



## Tipos de servicios

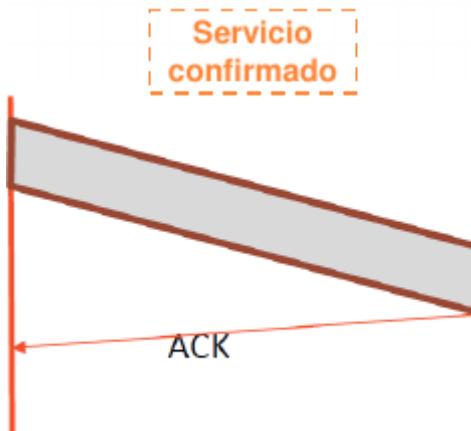
Conexión:

- **Orientado a conexión (SOC):** se caracteriza porque antes de transmitir los datos o establecer una comunicación, se debe establecer una conexión. (Ej: Servicio de telefonía)
- **No Orientado a conexión (SNOC):** no precisa la existencia de una conexión previa a la transmisión de la información. (Ej: Envío Postal)



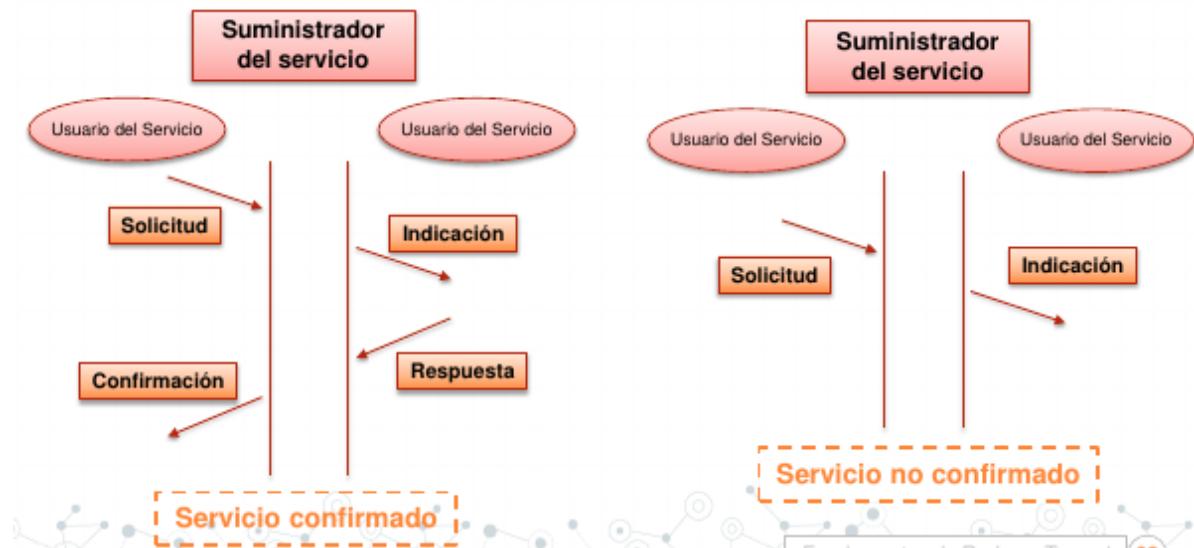
Confirmación:

- **Confirmado (fiable):** cuando el emisor tiene constancia de la recepción en el destino. (Ej: Envío postal certificado)
- **No confirmado (no fiable):** no se produce dicha confirmación. (Ej: Envío postal normal)



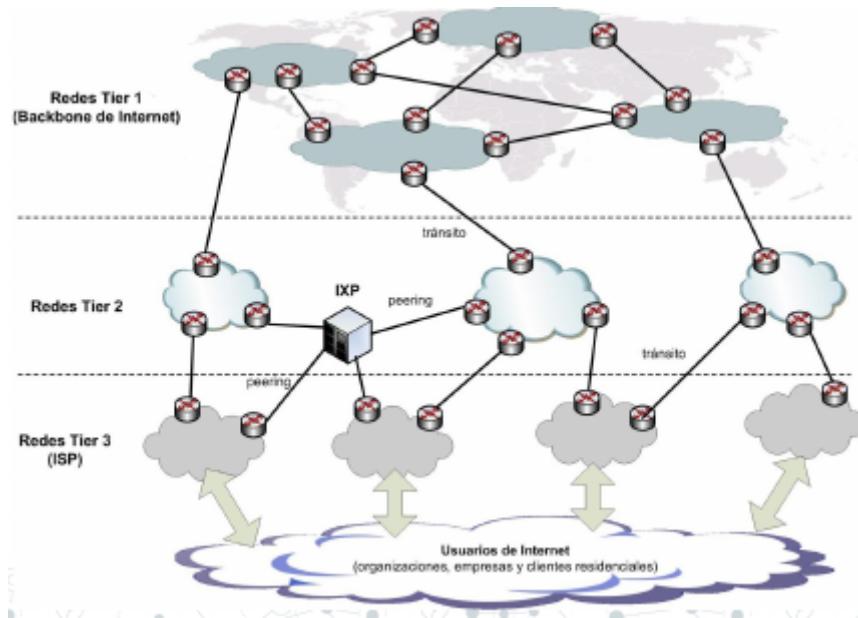
## Especificación de un servicio

- **Primitivas de servicio:** un servicio se especifica de manera formal con un conjunto de primitivas disponibles para que un usuario u otra entidad acceda al servicio. Estas primitivas ordenan al servicio que ejecute alguna acción o que informe de una acción que haya realizado una entidad paritaria.
  - **Request:** Petición o solicitud para realizar una acción.
  - **Indication:** Notificación de que ha ocurrido un suceso.
  - **Response:** Solicitud de respuesta a un suceso.
  - **Confirm:** Confirmación de que ha llegado la respuesta de una acción anterior.



## 4. Internet: arquitectura y direccionamiento

La estructura actual de Internet está basada en la interconexión de redes de forma jerárquica, con varios niveles conocidos como tiers. De forma general existen tres niveles conocidos como Tier 1, Tier 2 y Tier 3.



- **Tier 1:** son las redes de los grandes operadores globales (Global Carriers) que tienen tendidos de fibra óptica por al menos dos continentes.
  - Desde una red Tier 1 se puede acceder a cualquier punto de Internet, dado que todas las redes Tier 1 tienen que estar conectadas entre sí (requisito a los operadores).
  - Se puede decir que las redes Tier 1 forman el núcleo (backbone) de Internet.
- **Tier 2:** son operadores de ámbito más reducido que no pueden alcanzar todos los puntos de Internet y que necesitan conectarse a una red Tier 1 para ello.
  - Ofrecen servicios de conectividad a los operadores Tier 3.
- **Tier 3:** pertenecen a los operadores que dan servicio de conexión a Internet a los usuarios residenciales y a muchas empresas.
  - Son los llamados ISP (Internet Service Provider) o Proveedores de acceso a Internet.

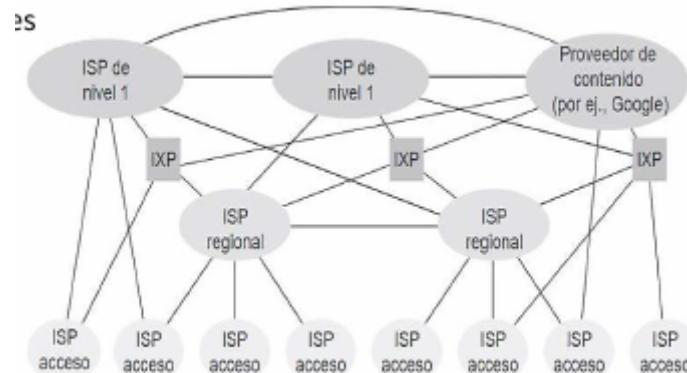
La conexión entre las redes de diferentes operadores se puede hacer de dos formas. Las conexiones son acuerdos entre las operadoras.

- **Conexión de tránsito:**

- Conexión entre operadores de diferente jerarquía.
- El operador de mayor jerarquía (proveedor) vende una conexión de tránsito al operador de menor jerarquía (cliente).
- El proveedor le da acceso al cliente a todas sus rutas (conexiones), es decir, el cliente recibirá tanto las rutas de la red del proveedor como las rutas con destino a otras redes.
- El cliente publica al proveedor sólo sus rutas y no otras que pueda tener con otros proveedores.
- Las redes Tier 1 son las únicas que no utilizan conexiones de tránsito.

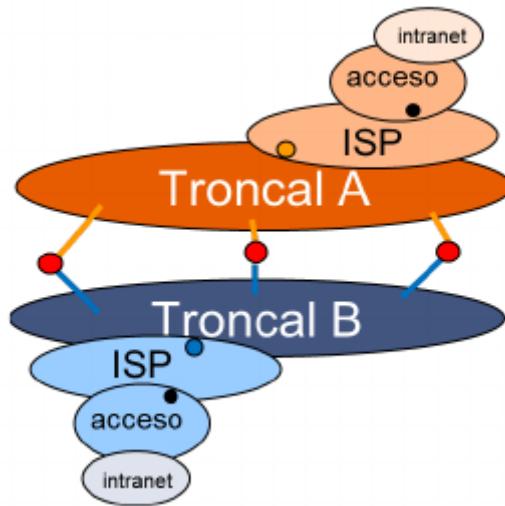
- **Conexión de peering:**

- Conexión utilizada para el intercambio de tráfico sin coste entre dos operadores.
- Cada operador publica sólo sus rutas y no otras rutas que tenga con otros proveedores u otras rutas de peering.
- El peering sirve para acceder desde un operador al rango de direcciones IP del otro operador, no sirve para llegar a otros rangos de direcciones.
- Puede ser de dos tipos:
  - Públicos: utilizando un IXP (Internet eXchange Point)
    - Se trata de una infraestructura física que permite a diferentes ISP intercambiar tráfico de Internet entre sus redes.
    - Este intercambio se lleva a cabo mediante conexiones peering.
    - En Europa existe una asociación de IXP llamada Euro-IX que agrupa a todos los IXP europeos y algunos IXP de Japón y Estados Unidos.
  - Privados: conexión directa entre los dos proveedores.



Topología jerárquica:

- Redes troncales (ATM, SDH, SONET, etc) de grandes operadores de telecomunicaciones (ISP de nivel 1).
- Redes de acceso (xDSL, RDSI, FTTH, etc) del ISP
- Intranets (Ethernet) del usuario: zona pública + zona privada



## Direccionamiento

Para que dos sistemas (hosts/nodos) conectados a Internet se puedan comunicar entre sí, es necesario que puedan ser identificados, para que los nodos intermedios (routers) sean capaces de transmitir los paquetes de datos desde el origen al destino.

En Internet la identificación se realiza mediante direcciones IP (Direccionamiento IP). Una dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz de un sistema dentro de una red que utilice el protocolo IP.

**Las direcciones IP están asociadas a una interfaz, no a un sistema final** (un sistema final tendrá una dirección IP diferente para cada una de sus interfaces).

- **Las direcciones IPv4** son números binarios de 32 bits, representadas normalmente mediante notación decimal separada por puntos.

Los 32 bits se dividen en 4 grupos de 8 bits cada uno, y los valores decimales de cada grupo de 8 bits (que son números comprendidos entre 0 y 255) se concatenan con puntos.

- **Las direcciones IPv6** son números binarios de 128 bits, que se dividen en 8 grupos de 16 bits cada uno. A su vez, cada uno de estos 16 bits se divide en 4 subgrupos de 4 bits. Los valores hexadecimales de cada subgrupo de 4 bits (comprendidos entre 0 y F) se concatenan.

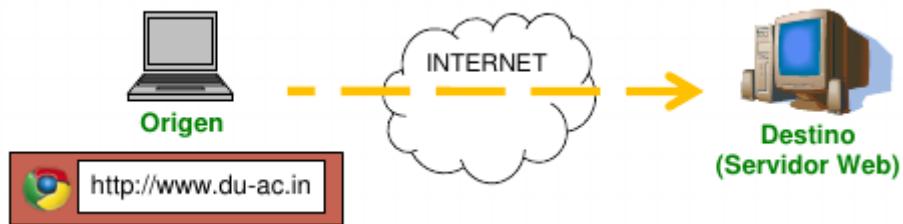
IPv6 se ha diseñado con el objetivo de reemplazar a IPv4. El proceso de migración de IPv4 a IPv6 no se completará hasta dentro de muchos años.

Dependiendo del tipo de red a la que pertenezca, una dirección IP puede ser:

- Pública: dirección que tiene cualquier sistema conectado de forma directa a Internet. Las IP públicas no pueden repetirse.
- Privada: Las direcciones IP privadas se utilizan para identificar sistemas dentro de redes domésticas o privadas.

Dependiendo del modo en que se asigna una dirección IP puede ser:

- Fija: Las direcciones IP fijas son aquellas que no cambian. Es decir, una vez que se asigne la dirección IP al dispositivo, este tendrá siempre la misma, ya sea en Internet (IP fija pública) o en una red privada (IP fija privada). Las direcciones IP fijas son comúnmente utilizadas en servidores (por ejemplo, google.com siempre tiene la misma IP).
- Dinámica: Las direcciones IP dinámicas son direcciones variables. Un mismo equipo puede tener una dirección IP en un cierto momento y otra distinta en otro (la IP que te proporciona tu proveedor de internet suele cambiar con el tiempo porque eres un Don Nadie).



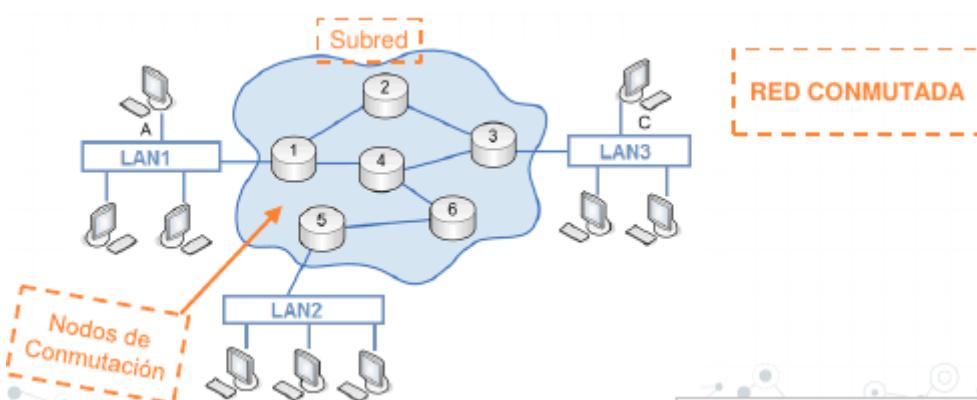
- URL: <http://www.du-ac.in/index.html> (nombre de dominio: du-ac.in) → Capa de aplicación
- Puertos: identifican al proceso en origen y destino → Capa de transporte
- Dirección IP: identifica a los hosts) → Capa de red/internet
  - Origen: 192.168.1.10
  - Destino: 70.185.33.15

## TEMA 2: CAPA DE RED

### 1. Funcionalidades

El objetivo de la capa de red en Internet es la interconexión de redes, con independencia de la tecnología subyacente. En el modelo OSI el control de congestión se realiza en esta capa.

- **Comutación:** acción de cursar tráfico entre los nodos de la red.
- **Encaminamiento (routing):** encontrar la mejor ruta desde un origen a un destino.



Funciones TCP en el emisor	Funciones TCP en el receptor
<ul style="list-style-type: none"> <li>- Divide la información en paquetes</li> <li>- Agrega un código detector de errores para comprobar si el paquete llega correctamente a su destino</li> <li>- Pasa el paquete al protocolo IP para que gestione su envío</li> </ul>	<ul style="list-style-type: none"> <li>- Recibir los paquetes que le pasa IP</li> <li>- Ordena los paquetes, comprueba que están todos y correctos</li> <li>- Extrae la información útil de los paquetes</li> <li>- Si detecta un paquete incorrecto o faltante, genera un paquete para enviar al emisor que indica que se debe enviar de nuevo</li> </ul>

## 2. Comutación

Proceso donde se pone en comunicación un host con otro, a través de una infraestructura de comunicaciones común, para la transferencia de información.

Se necesita establecer un sistema de comunicación entre dos puntos, un emisor (Tx) y un receptor (Rx) a través de equipos/nodos de transmisión. Se determinará y establecerá un camino que permita transmitir información extremo a extremo.

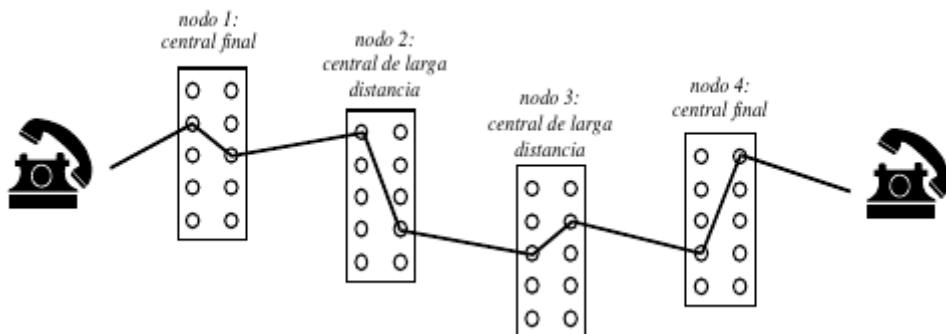
La comutación para conectar redes entre sí funciona en la Capa 3 del modelo OSI (Capa de Red).

Los servicios fundamentales que emplean técnicas de comutación son:

- Servicio telefónico
- Servicio telegráfico
- Servicio de datos

Tecnologías de comutación:

- **De circuitos:** consiste en el establecimiento de un circuito físico previo al envío de información, que se mantiene abierto durante todo el tiempo que dura la trasmisión. El camino físico se elige entre los disponibles, empleando diversas técnicas de señalización: "por canal asociado" (si viaja en el mismo canal) o "por canal común" (si lo hace por otro distinto), encargadas de establecer, mantener y liberar dicho circuito.



Pasos:

1. **Establecimiento del circuito:** el host emisor solicita a un cierto nodo de comutación el establecimiento de conexión hacia un host receptor (servicio orientado a conexión). El receptor es el encargado de dedicar uno de sus canales lógicos al emisor. También será el encargado de encontrar los nodos intermedios para llegar al receptor, teniendo en cuenta ciertos criterios de encaminamiento, coste, etc...
2. **Transmisión de datos:** una vez establecido el circuito exclusivo para esta transmisión, se transmite desde el emisor hasta el receptor comutando sin demoras de nodo en nodo (los nodos tienen reservado un canal lógico para ello).
3. **Desconexión del circuito:** terminada la transferencia, el emisor o el receptor indican a su nodo de comutación más inmediato que ha finalizado la conexión. Este nodo informa al siguiente de este hecho y luego libera el canal dedicado, así hasta liberar el canal dedicado completo en el otro extremo.

Las ventajas son:

- Recursos dedicados (circuito en exclusiva).
- Facilita comunicaciones tiempo-real (voz y vídeo).
- No hay colisiones (no hay contienda por acceder al medio).
- No hay contención (el medio está disponible completamente quiere decir que se transmite a la máxima velocidad posible).

- No hay encaminamiento (una vez establecido el circuito), lo que equivale a una transmisión más rápida.
- Simplicidad de gestión en nodos (se recibe siempre por la misma entrada y se transmite siempre por la misma salida).

Las desventajas son:

- Retraso para establecimiento de la conexión (hay que resolver toda la ruta).
- Bloqueo y posible infrautilización de recursos (la línea está reservada aunque no se aproveche).
- Poca flexibilidad para adaptarse a cambios (no se reajusta la ruta si surgen posibles rutas alternativas mejores).
- Poco tolerante a fallos (si falla un nodo del camino, se cae todo el circuito).

- **De paquetes:** no es necesario establecer una conexión previa.

Los paquetes permanecen muy poco tiempo en memoria, por lo que resulta muy rápida. Un paquete se compone de datos útiles y información de control (que determina la ruta a seguir a lo largo de la red hasta el destino).

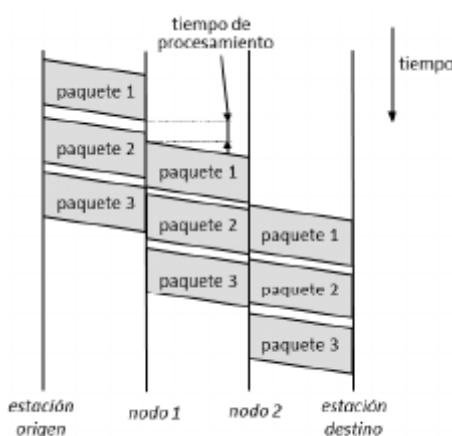
Pasos:

1. Un host que quiere enviar información a otro la divide en paquetes
2. Pasa los paquetes a un nodo intermedio que es el encargado de pasarlo hacia el siguiente destino. Cada nodo intermedio realiza las siguientes funciones:
  - **Almacenamiento y retransmisión (store and forward):** el paquete se detiene (almacena) el tiempo necesario para procesarlo
  - **Control de ruta (routing):** selección de un nodo del camino por que deben retransmitirse los paquetes para hacerlos llegar a su destino

Los paquetes toman diversos caminos pero nadie puede garantizar que todos los paquetes vayan a llegar en un momento determinado ni en ningún orden.

Se admiten dos variantes distintas:

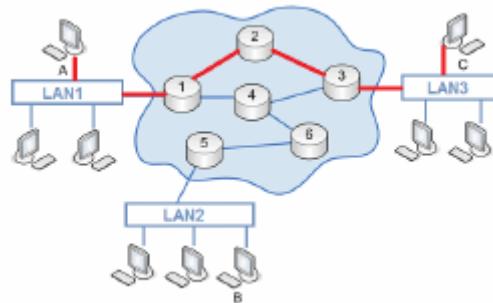
- **Comutación de datagramas:**
  - No hay conexión
  - Envío en unidades de datos (paquetes) independientes
    - En cada salto: almacenamiento y re-envío
  - Cada paquete debe contener las direcciones origen y destino
  - Los paquetes pueden seguir rutas diferentes y pueden llegar desordenados
  - Ejemplo: IP



- **Comutación con circuitos virtuales:**

- Orientado a conexión.

- Antes de la transmisión se establece una ruta entre el origen y el destino (puede ser diferente en cada sentido).
  - Se envían unidades de datos (paquetes) independientes.
  - No se acaparan los recursos (se comparten).
  - En cada salto: almacenamiento y re-envío (se debe comprobar si los recursos están libres).
  - Los paquetes llegarán ordenados.
  - Ejemplo: ATM (Asynchronous Transfer Mode)



Ventajas de circuitos virtuales frente a datagramas:

- El encaminamiento en cada nodo sólo se hace una vez para todo el grupo de paquetes. Por lo que los paquetes llegan antes a su destino.
  - Todos los paquetes llegan en el mismo orden del de partida ya que siguen el mismo camino.
  - En cada nodo se realiza detección de errores, por lo que si un paquete llega erróneo a un nodo, éste lo solicita otra vez al nodo anterior antes de seguir transmitiendo los siguientes.

Desventajas de circuitos virtuales ante datagramas:

- En datagramas no hay que establecer la conexión → para pocos paquetes, es más rápida la conmutación de datagramas.
  - Los datagramas son más flexibles → si hay congestión en la red, una vez que ya ha partido algún paquete, los siguientes pueden tomar caminos diferentes. En circuitos virtuales, esto no se puede hacer.
  - El envío mediante datagramas es más fiable → si un nodo falla, se perderá sólo un paquete. En circuitos virtuales se perderán todos (si no hay un mecanismo de recálculo de la ruta).

### **3. El protocolo IP**

Es un protocolo para la interconexión de redes (también llamadas subredes). Resuelve el encaminamiento en Internet: encontrar la ruta para llegar al destino.

- Es un protocolo salto a salto. Involucra a hosts y routers.
  - Ofrece un servicio no orientado a conexión y no fiable:
    - No hay negociación o “handshake” → no hay una conexión lógica entre las entidades.
    - No existe control de errores, ni control de flujo, ni control de congestión.
  - La unidad de datos (paquete) de IP se denomina datagrama.
  - IP es un protocolo de máximo esfuerzo (“best-effort”) o buena voluntad: los datagramas se pueden perder, duplicar, retrasar o llegar desordenados. No hay garantías de nada, pero lo hará lo mejor que pueda.
  - IP gestiona la fragmentación: adaptar el tamaño del datagrama a las diferentes Maximum Transfer Units (MTUs) de las subredes hasta llegar al destino.
  - Cada entidad en Internet se identifica por su dirección IP. Cada dirección IP es única en internet

- Una dirección IP es una etiqueta numérica que identifica, de manera lógica a una interfaz de un sistema dentro de una red que utilice el protocolo IP.

Internet adopta un direccionamiento jerárquico que simplifica las tablas de routing. Las direcciones IPv4 tienen 32 bits, agrupados en 4 bloques de 8 bits cada uno. Se representan mediante notación decimal (entre 0 y 255) separada por puntos. Ej: 200.110.23.77

- Cada dirección IP tiene dos partes bien diferenciadas:

- **Un identificador de la subred** o prefijo (parte izquierda de la IP)
  - La máscara de red es un patrón de '1s' que determina qué bits de la IP completa corresponden al identificador de subred.

Dada una IP, para obtener la dirección o identificador de la subred, se realiza una operación lógica "&" (AND) con la máscara de red:

Dirección IP: 200.27.4.112 → 11001000.00011011.00000100.01110000

Máscara: 255.255.255.0 → 11111111.11111111.11111111.00000000

200.27.4.112 → 11001000.00011011.00000100.01110000

& &

255.255.255.0 → 11111111.11111111.11111111.00000000

-----

Subred → 200.27.4.0 ⇔ 11001000.00011011.00000100.00000000

La máscara se puede representar de forma compacta, indicando el número de '1s' que tiene. Por ejemplo, la máscara anterior sería /24

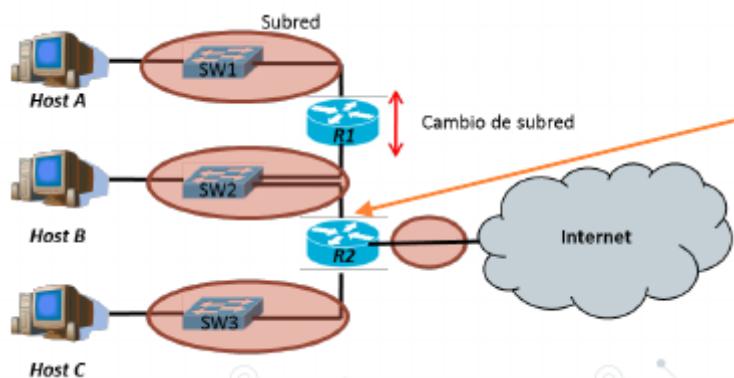
- **Un identificador del dispositivo** dentro de esa subred (parte derecha de la IP).

Cada subred tiene un identificador (o prefijo) único en la intranet (red privada). Cada dispositivo (interfaz) tiene un identificador único en la subred.

## Subredes

Podemos considerar internet como un conjunto de subredes conectadas

- **Subred**: líneas de transmisión e infraestructura de red que permite conexión directa de dispositivos IP sin intermediarios (un switch se considera transparente).
- **Switch**: se usa para crear redes de computadoras. Son transparentes. Trabaja a nivel de enlace (capa 2 OSI).
- **Router**: se usa para conectar redes entre sí. Es un punto de separación, ya que limita el tráfico entre las redes. Redirige los paquetes hacia el destino de una transmisión. Trabaja a nivel de red (capa 3 OSI).



¿Cómo determinar las subredes en un esquema de red?

"Para determinar las subredes, separe cada interfaz de los hosts y routers, creando redes aisladas. Dichas redes aisladas se corresponden con las subredes."

Tendrán dirección IP cada una de las interfaces de los hosts y de los routers. Los switches no tienen dirección IP.

### ¿Cómo se elige la máscara?

Según el número de dispositivos que necesitemos direccionar en la subred, tal que se ajusta para no desaprovechar direcciones. Recordar que cada subred tiene un identificador único en nuestra intranet.

Número de dispositivos =  $(2^n - 2)$

Ej: 8 ceros (/24, 11111111.11111111.11111111.00000000) permite 254 dispositivos

El -2 viene de que la primera IP y última son reservadas: la 0.0.0.0 se reserva para la dirección de red y la FF..FF se usa para broadcast.

- 200.27.4.0 = 11001000.00011011.00000100.00000000 → Reservada (subred)
- 200.27.4.1 = 11001000.00011011.00000100.00000001 → Dispositivo #1
- ...
- 200.27.4.254 = 11001000.00011011.00000100.11111110 → Dispositivo #254
- 200.27.4.255 = 11001000.00011011.00000100.11111111 → Reservada (difusión)

### Tipos de direcciones IP

- **Públicas:** cada dirección se asigna a sólo 1 dispositivo (una interfaz) en toda la Internet global. Se asignan centralizadamente.
  - Las IPs públicas son únicas en todo internet
- **Privadas:** sólo sirven para tráfico dentro de las intranets. Se pueden repetir en distintas intranets. Las asigna el usuario según su criterio. Rangos de IPs privadas:
  - 10.0.0.0/8 → de 10.0.0.0 a 10.255.255.255
  - 172.16.0.0/16 → de 172.16.0.0 a 172.31.255.255
  - 192.168.0.0/24 → de 192.168.0.0 a 192.168.255.255

Originalmente se definieron 5 clases de direcciones IP:

- **Clases A, B, C:** jerárquicas a dos niveles (identificador de red+identificador dispositivo)
- **Clase D:** multicast
- **Clase E:** como previsión para algún uso futuro, aún sin determinar

Clase A	0	red (7 bits)	host (24 bits)
Clase B	1 0	red (14 bits)	host (16 bits)
Clase C	1 1 0	red (21 bits)	host (8 bits)
Clase D	1 1 1 0	dirección grupo <i>multicast</i> (28 bits)	
Clase E	1 1 1 1 0	uso futuro	

## Rangos de direcciones IP

Según su clase:

A → 0.0.0.0 – 127.255.255.255	→ 128 redes	x 16.777.216 hosts
B → 128.0.0.0 – 191.255.255.255	→ 16.384 redes	x 65.536 hosts
C → 192.0.0.0 – 223.255.255.255	→ 2.097.152 redes	x 256 hosts
D → 224.0.0.0 – 239.255.255.255	→ para multicast	
E → 240.0.0.0 – 255.255.255.255	→ usos futuros	

Recordar reglas especiales:

- 00..00 para identificar una red, nunca un dispositivo
- FF..FF para difusión en la red especificada
- 127.0.0.0 autobucle/loopback

Reserva direcciones privadas:

A → 10.0.0.0	→ 1 Red privada de Clase A
B → 172.16.0.0 – 172.31.0.0	→ 16 redes privadas de Clase B
C → 192.168.0.0 – 192.168.255.0	→ 256 redes privadas de Clase C

## Agotamiento de IPs

Los bloques de direcciones IPv4 se “agotaron” ya (Nov. 2019). Sólo quedan disponibles bloques /24 (256 direcciones) a /32 (1 dirección). Se van recopilando direcciones de sitios obsoletos, empresas que hayan desaparecido, proyectos terminados, hosting que ya no está en uso...

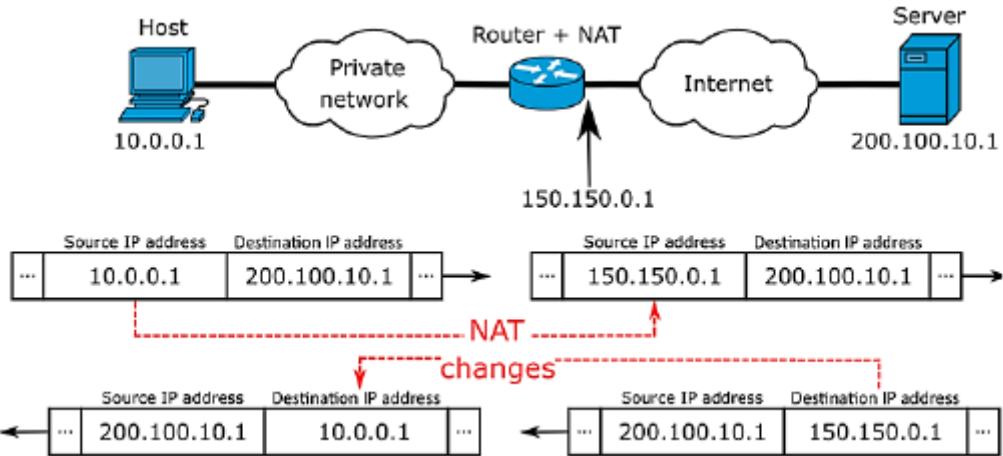
Se creó IPv6 para solucionar esto y reemplazar IPv4, pero eso tardará mucho en suceder porque hay muchos servicios que seguirán usando IPv4.

## IPv6

- IPv6 usa un esquema de direccionamiento de 128 bits.
- Notación hexadecimal, 8 grupos de 4 dígitos, separados por ":"
  - Cada dígito hexadecimal corresponde a 4 dígitos en binario (4 bits).
  - Rango: 0000:0000:0000:0000:0000:0000:0000:0000 a FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
  - 340 sextillones de direcciones diferentes.
- Compatible con IPv4.

## NAT (Network Address Translation)

- Consiste en traducir un conjunto de direcciones IPv4 en otras.
- Permite que una red con direccionamiento privado se pueda conectar a Internet (direcciónamiento público).
  - Cambia la dirección IP privada por una dirección pública al reenviar un paquete hacia el exterior de la red (hacia Internet).
  - Cambia la dirección IP pública por la correspondiente privada al reenviar un paquete hacia el interior.
- Utiliza una tabla de traducciones, que contiene direcciones IP y puertos.
  - Los puertos se asocian a los equipos de la red privada (para dirigir el tráfico entrante)

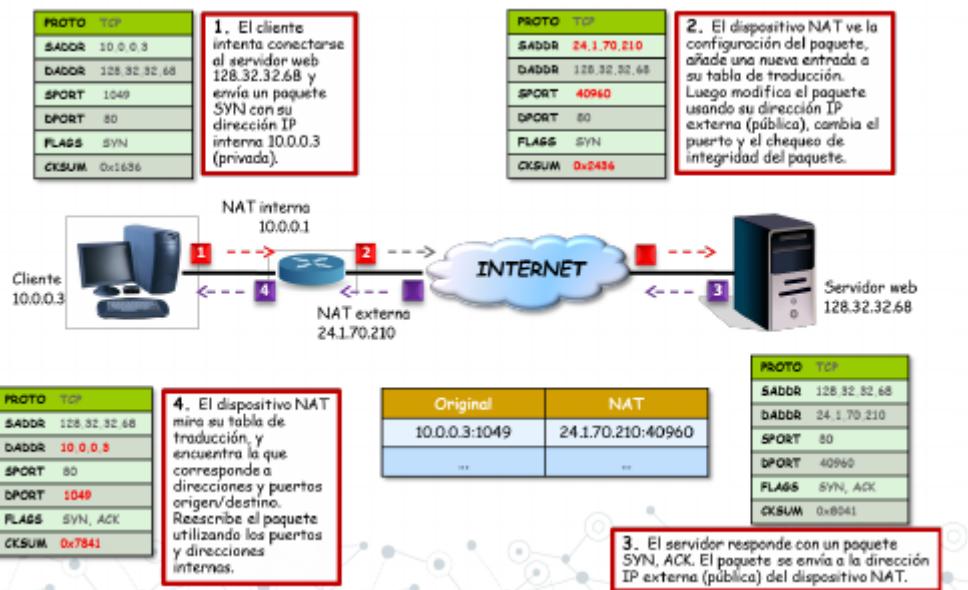


El problema de la escasez de direcciones IP:

- Se necesitan m direcciones pero se dispone de n, siendo n < m.
- Si n = 1 se denomina enmascaramiento (masquerading).
- Se usa en ISPs, para así poder dar acceso a más usuarios que direcciones IP tenga el ISP. Se supone que no todos los usuarios acceden simultáneamente. Las direcciones se asignan a los usuarios de forma dinámica.

Tipos de NAT:

- SNAT (Source NAT) → el origen de los datos está en la red privada; cambia la dirección IP de origen.
- DNAT (Destination NAT) → el origen de los datos está en la red pública; cambia la dirección IP de destino; requiere configurar en el router qué puerto irá dirigido a qué máquina.



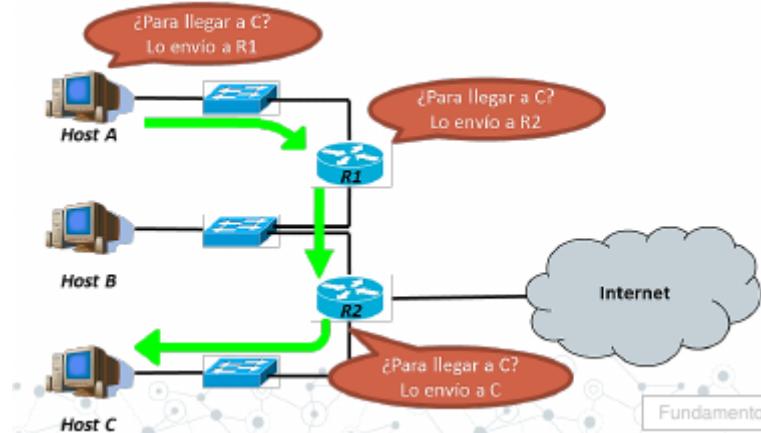
1. El cliente intenta conectarse al servidor web (que tiene una IP pública) y envía un paquete SYN con su dirección IP privada/interna
2. El dispositivo NAT ve la configuración del paquete, **añade una nueva entrada a la tabla de traducción**. Luego, modifica el paquete usando su dirección pública/externa, cambia el puerto y el chequeo de integridad del paquete
3. El server responde con un paquete SYN, ACK. El paquete se envía a la dirección IP pública/externa del dispositivo NAT
4. El dispositivo NAT **mira su tabla de traducción** y encuentra la dirección privada correspondiente y sus puertos de origen y destino. Reescribe el paquete usando los puertos y direcciones privadas/internas

## Enrutamiento

Encontrar el mejor camino para llevar la información (paquete) de un origen a un destino dado.

Se realiza paquete a paquete y salto a salto, en función de la IP destino del paquete y de las Tablas de Encaminamiento residentes en cada una de las entidades IP (host origen y routers).

En cada salto (router) se hace almacenamiento y retransmisión.



El encaminamiento se realiza salto a salto y datagrama a datagrama (IP es no orientado a conexión).

Modos de encaminamiento:

- **Directo** cuando lo resuelve el propio router
- **Indirecto** cuando lo resuelve el router siguiente en la ruta

Cada dispositivo (host o router) tiene una Tabla de encaminamiento.

Un router suele estar en varias redes distintas, un host suele estar en solo una.

## Encaminamiento (Enrutamiento)

### • Tabla de encaminamiento de R1

Destino ( $D_i$ )	Salto siguiente ( $S_i$ )	Máscara ( $M_i$ )
127.0.0.1	* Conexión directa	255.255.255.255
192.100.12.0	*	255.255.255.0
192.100.13.0	*	255.255.255.0
192.100.15.0	192.100.12.1	255.255.255.0
Default	150.100.0.222	0.0.0.0

- ¿Faltaría alguna entrada?

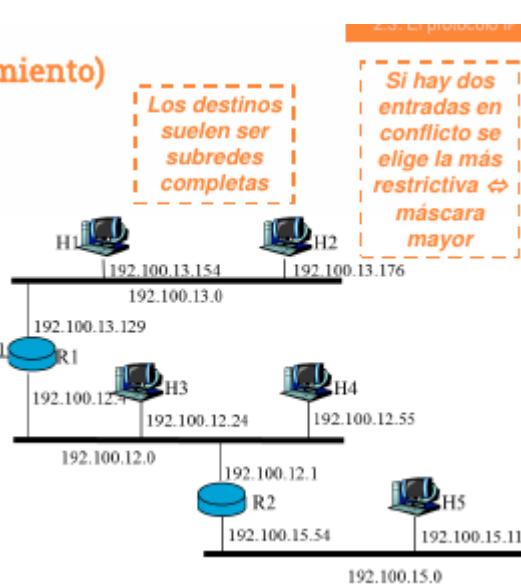
Una específica a la red 150.100.0.0/30

- La máscara se puede indicar en formato compacto:

255.255.255.0  $\Leftrightarrow$  /24

255.255.255.192  $\Leftrightarrow$  /26

255.255.255.252  $\Leftrightarrow$  /30



Proceso de encaminamiento:

1. Se extrae la dirección de destino (IP\_DESTINO) del datagrama

2. Por cada entrada i con  $i=1..N$  de la tabla de encaminamiento se calcula:

```
Ip_i = IP_DESTINO && MASCARA_i
```

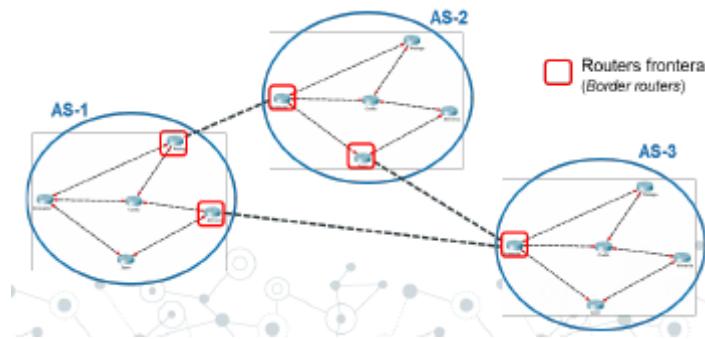
3. Si  $Ip_i = D_i$  y...

- ... es routing directo, entonces se reenvia el datagrama al destino final por la interfaz i
  - ... es routing indirecto, entonces se reenvia el datagrama al salto siguiente por la interfaz i
- Si hay varias coincidencias se elige el destino con la máscara más larga (con más 1s, la más restrictiva)
  - Si se ha barrido toda la tabla y no hay coincidencia con ninguna fila → error (possible mensaje ICMP)

## Sistemas autónomos

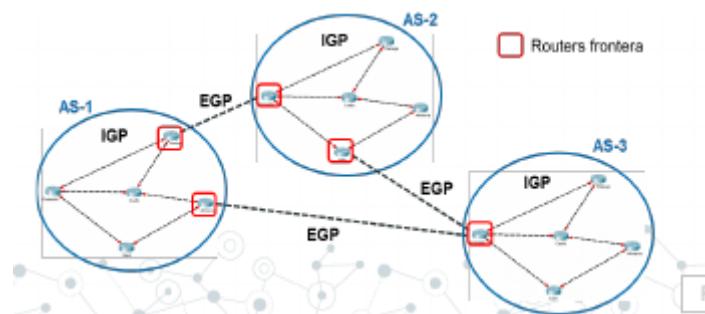
Para facilitar la administración y aumentar la escalabilidad Internet se jerarquiza en Sistemas Autónomos (SA). Un SA es un conjunto de redes y routers administrados por una autoridad.

Cada SA informa a los otros SA de las redes accesibles. Existe un router responsable de esto, denominado router exterior (o router frontera). Cada SA se identifica por un entero de 16 bits (DESDE 2007 ES 32-BITS). Ej: Rediris → AS766



Intercambio de tablas:

- Internet se jerarquiza en Sistemas Autónomos.
- Existe encaminamiento dinámico (mediante algoritmos automáticos).
- Se definen 2 niveles de encaminamiento (intercambio de tablas):
  - Algoritmos IGP → los que se usan dentro de un SA (el administrador tiene libertad de elección): RIP, OSPF, HELLO, IGRP, EIGRP
  - Algoritmos EGP → los que se usan entre SAs (norma única en Internet): BGP



## Algoritmos de encaminamiento

- Vector distancia:** los routers construyen su tabla de rutas con el único conocimiento de la distancia (métrica) y el siguiente salto (next hop) para llegar a la red de destino.

Esta distancia puede ser un número que indica: longitud del enlace, número de saltos, latencia (tiempo medio) u otros valores.

Requiere intercambiar información periódicamente con los routers vecinos para recalcular la distancia. Cada router envía su tabla de encaminamiento a los demás.

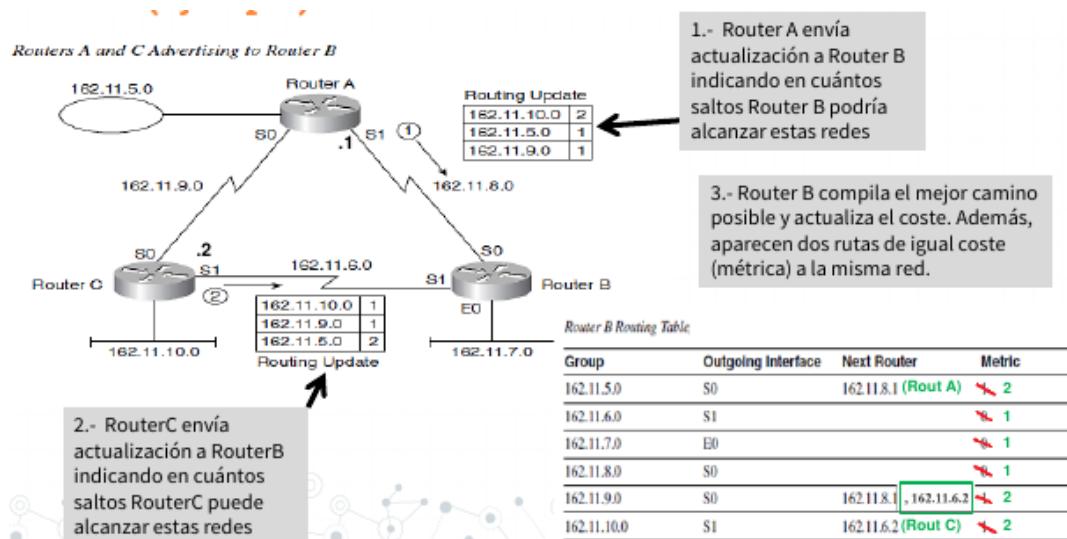
- **RIP (Routing Information Protocol)**: protocolo de la capa de aplicación (opera sobre UDP puerto 520).

Adopta un algoritmo vector-distancia (métrica basada en **número de saltos**). No considera la congestión de la red ni la velocidad de los enlaces.

- Una red directamente conectada a un router tiene coste 1. Máximo de 15 saltos (16 sería considerada distancia infinita o no alcanzable).

Periódicamente (por defecto cada 30 segundos) cada router RIP recibe de todos sus vecinos y envía a todos sus vecinos (dirección multicast 224.0.0.9) los vectores-distancia para todos los posibles destinos. De entre ellos, para un destino dado, se selecciona como salto siguiente el vecino que anuncie el menor coste, actualizando la métrica para ese destino sumando uno al coste anunciado (coste para alcanzar ese vecino desde el router actual).

- Problema convergencia lenta → las malas noticias tardan en propagarse.



- **Estado del enlace**: los routers necesitan conocer previamente toda la topología de la red (conexiones existentes entre los nodos) para calcular el camino al destino y generar su tabla de enrutamiento.

- **OSPF (Open Shortest Path First)**: basado en estado del enlace. Se publican los estados por difusión/inundación.

El coste por defecto que se considera en OSPF para cada enlace es:  $10^8/\text{Ancho de banda (BW)}$ .

- Ej: para un enlace con BW = 1 Mbps, coste=  $10^8/10^6 = 100$

El coste de los enlaces se podrá determinar en tiempo real mediante un administrador o un algoritmo automático.

Permite calcular rutas alternativas y hacer balanceo de carga. Se pueden considerar distintas métricas. Así se conseguirá dar prioridad a unos enlaces sobre otros y conseguir un balanceo de carga.

- Al conocer toda la red, las rutas se calculan usando un algoritmo de Dijkstra. A partir de las rutas se construyen las tablas de encaminamiento de cada router.
- Gestión en base a áreas independientes de la red.

Se minimiza la difusión mediante routers designados (son los que envían y reciben el estado de la red).

Mejor convergencia, ya que no hay que hacer cálculos sobre las rutas a difundir. Las actualizaciones se hacen sólo cuando hay cambios en la red.

Maneja distintas tablas (BD): vecinos, topología, rutas.

Mensajes: hello, database description, link status request/update/ack

## Formato datagrama IP

0	16	31			
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total		
Identificador		Flags	Posición de Fragmento		
Tiempo de Vida	Protocolo		Suma de Control de Cabecera		
Dirección IP de Origen					
Dirección IP de Destino					
Opciones		Relleno			

**Versión:**  
0100 ⇔ 4

**Tamaño cabecera:**  
En palabras de 32 bits (entre 5 y 15) ⇔ entre 20 y 60 bytes.

**Tipo servicio:**  
Preferencia de envío (mínimo retardo, máximo rendimiento, mínimo coste).

**Longitud total:**  
Tamaño en bytes del datagrama completo (incluyendo datos).

0	16	31			
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total		
Identificador		Flags	Posición de Fragmento		
Tiempo de Vida	Protocolo		Suma de Control de Cabecera		
Dirección IP de Origen					
Dirección IP de Destino					
Opciones		Relleno			

**Identificador:**  
Número de orden del paquete en un mensaje.

**Flags:**  
Indican si hay fragmentación.

**Posición fragmento:**  
Desplazamiento del fragmento respecto del paquete original (para reconstruirlo).

0	16	31			
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total		
Identificador		Flags	Posición de Fragmento		
Tiempo de Vida	Protocolo		Suma de Control de Cabecera		
Dirección IP de Origen					
Dirección IP de Destino					
Opciones		Relleno			

**Tiempo de vida (TTL):**  
Tiempo que puede estar el paquete en una red.

**Protocolo:** (RFC 3232)  
TCP, UDP, ICMP, etc

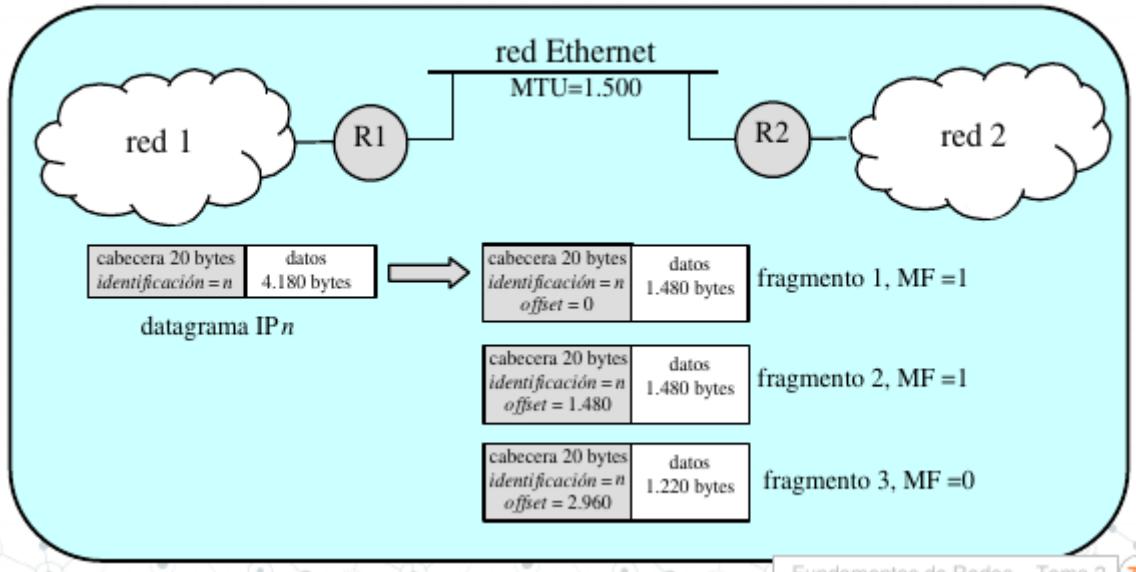
**Suma de control:**  
Número para comprobar la corrección de la cabecera.

Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total		
Identificador		Flags	Posición de Fragmento		
Tiempo de Vida	Protocolo	Suma de Control de Cabecera			
Dirección IP de Origen				<b>Opciones:</b> Hasta 40 bytes. Permite hacer funciones de test y depuración sobre la red (sello de tiempo, registro de ruta, etc).	
Dirección IP de Destino					
Opciones		Relleno			

## Fragmentación IP

- Tamaño máximo datagrama (incluyendo datos):  $(2^{16}) - 1 = 65.535$  bytes
- Adaptarse a la MTU
- Ensamblado en destino final:
  - desplazamiento: offset respecto del comienzo del paquete.
  - indicadores (I): "Don't Fragment", "More Fragments".

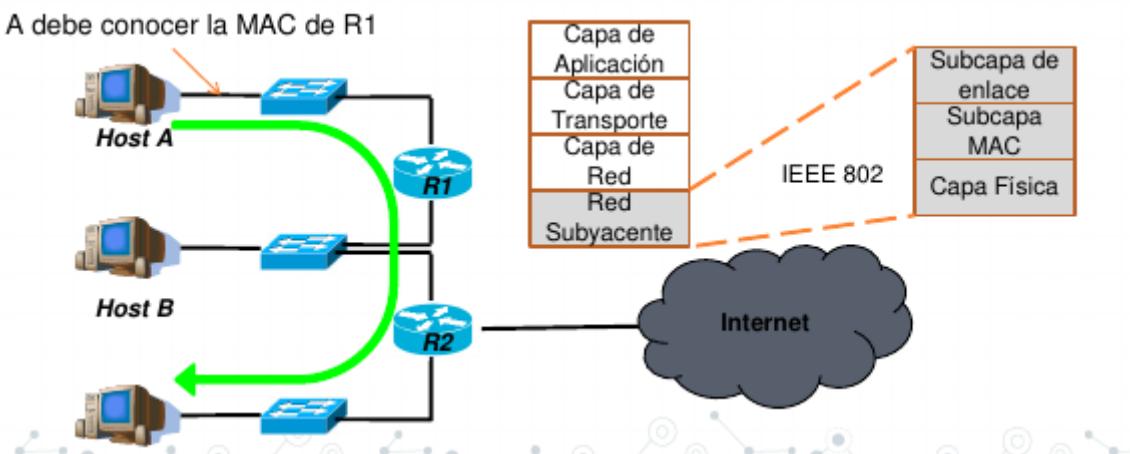
Nivel de enlace	MTU (bytes)
PPP normal	1500
PPP bajo retardo	296
X.25	1600 (RFC 1356)
Frame Relay	1600 (normalmente)
Ethernet DIX	1500
Ethernet LLC-SNAP	1492
Token Ring 4 Mb/s	4440 (THT 8ms)
Classical IP over ATM	9180



## 4. Asociación con la capa de enlace: protocolo ARP

### Direcciones MAC

Para transmisiones a nivel de enlace (físicas). Tras la redirección IP → Enviar a la MAC del siguiente nodo



### ARP

- Tras la redirección IP → Enviar a la dirección MAC (Medium Access Control) del siguiente nodo. Se usan en redes Ethernet (cableadas) y Wifi.
- Formato (6 bytes en hexadecimal): HH-HH-HH-HH-HH-HH
- Son únicas, asignadas por IEEE en lotes de 224 para cada fabricante
- Dirección de difusión está reservada en FF-FF-FF-FF-FF-FF

ARP:

- Address Resolution Protocol
- Obtener MAC a partir de IP

RARP:

- Reverse ARP
- Obtener IP a partir de MAC

## 5. El protocolo ICMP (Internet Control Message Protocol)

Informa sobre situaciones de error en IP → es un protocolo de señalización. Suelen ir (excepto eco y solicitudes) hacia el origen del datagrama IP original.

ICMP se encapsula en IP.

Tiene una cabecera de 32bits:

- Tipo (8 bits): tipo de mensaje

o

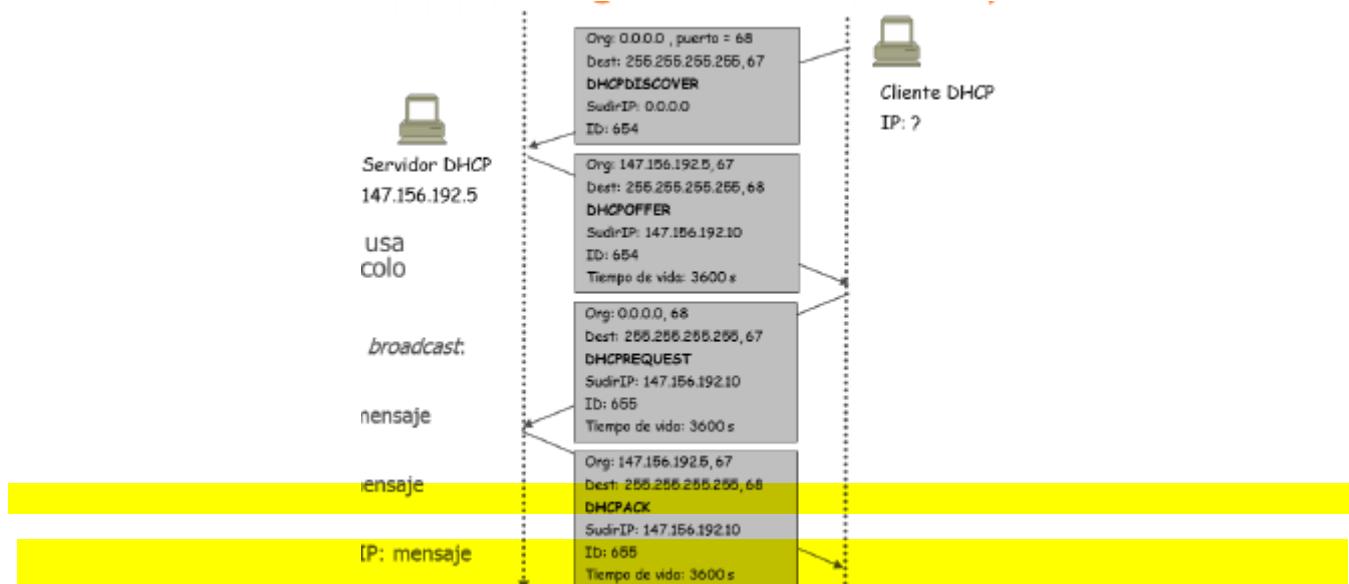
Campo tipo	Mensaje ICMP
8/0	Solicitud/respuesta de eco (ping)
3	Destino inalcanzable
4	Ralentización del origen
5	Redirecciónamiento
11	Tiempo de vida excedido
12	Problema de parámetros
13/14	Solicitud/respuesta de sello de tiempo
17/18	Solicitud/respuesta de máscara de red

- Código (8 bits): subtipo de mensaje
- Comprobación (16 bits)

## 6. Autoconfiguración de red: protocolo DHCP (Dynamic Host Configuration Protocol)

Asignación de IPs de forma dinámica en una red privada. Para asignar las IPs se usa DHCP, protocolo usuario de UDP (puerto 67):

1. El host cliente envía un mensaje broadcast "DHCP DISCOVER"
2. El server DHCP responde con un mensaje "DHCP OFFER"
3. El host solicita una dirección IP, mensaje "DHCP REQUEST"
4. El server DHCP envía la dirección IP, mensaje "DHCP ACK"



## TEMA 3: CAPA DE TRANSPORTE

# 1. Introducción a los protocolos de Capa de Transporte

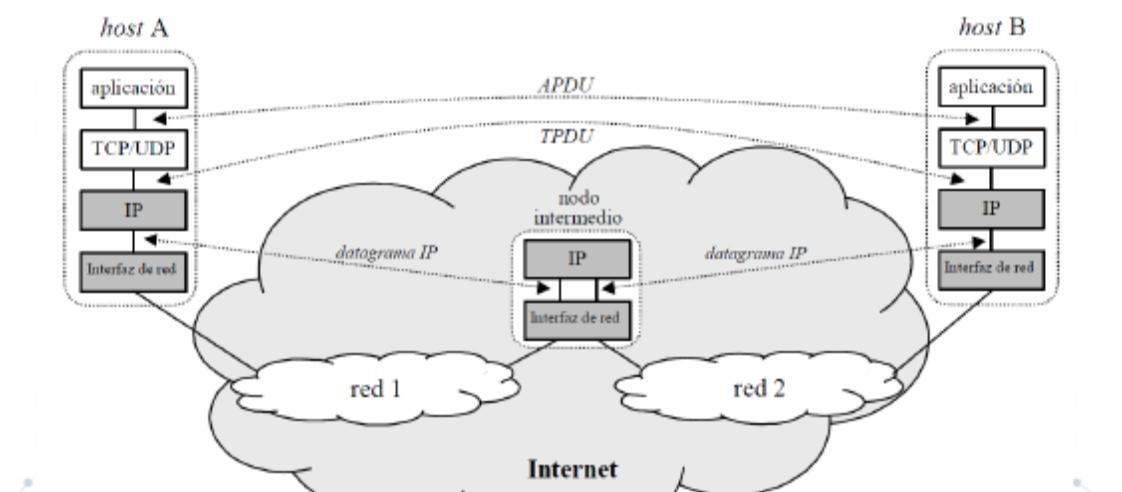
Las redes de datos e Internet nos dan soporte para establecer una comunicación continua y confiable entre los equipos.

En un único dispositivo, se pueden utilizar varios servicios (correo electrónico, acceso Web, mensajería instantánea, etc). Los datos de cada una de estas aplicaciones se empaquetan, se transportan y se entregan al servidor adecuado o aplicación en el dispositivo de destino.

La función principal de la capa de transporte es **aceptar los datos de las capas superiores, dividirlos en unidades más pequeñas si es necesario, y pasarlo a la capa de red** garantizando que lleguen a su destino independientemente la red o redes físicas que utilicen.

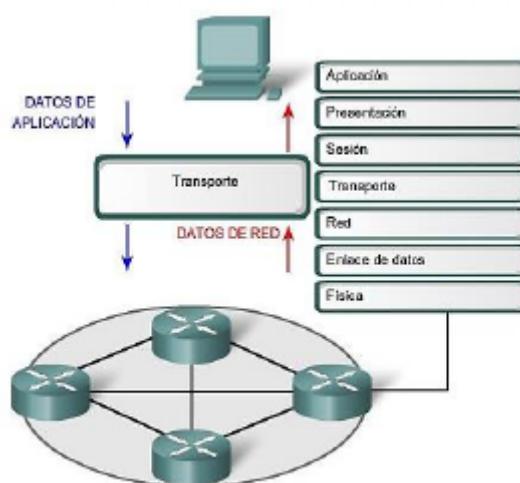
- **Entidades de transporte:** hardware y/o software que se encargan de realizar este trabajo.

La comunicación entre entidades de transporte es extremo a extremo (end-to-end). Es decir, se produce entre el emisor/receptor finales, no teniendo en cuenta a ningún otro dispositivo intermedio de las subredes.



El nivel de transporte mejora la calidad del servicio ofrecida por el nivel de red mediante:

- La multiplexación/demultiplexación de comunicaciones
- La introducción de redundancia en la información

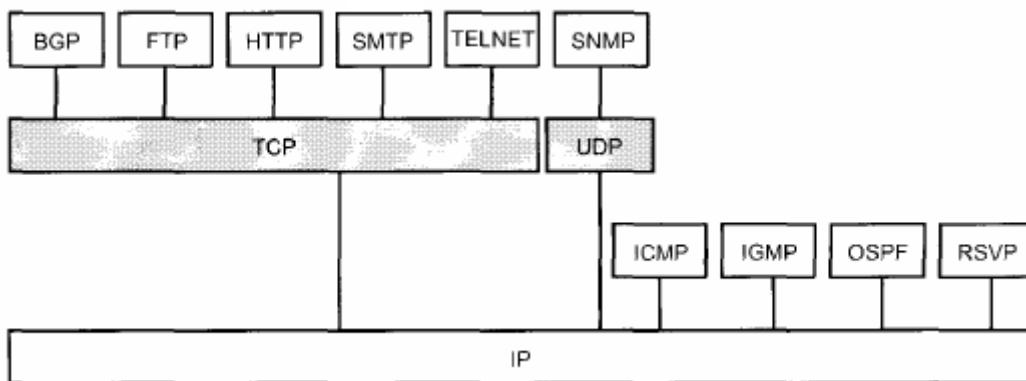


La capa de transporte es el enlace entre la capa de aplicación y la capa responsable de la transmisión en la red:

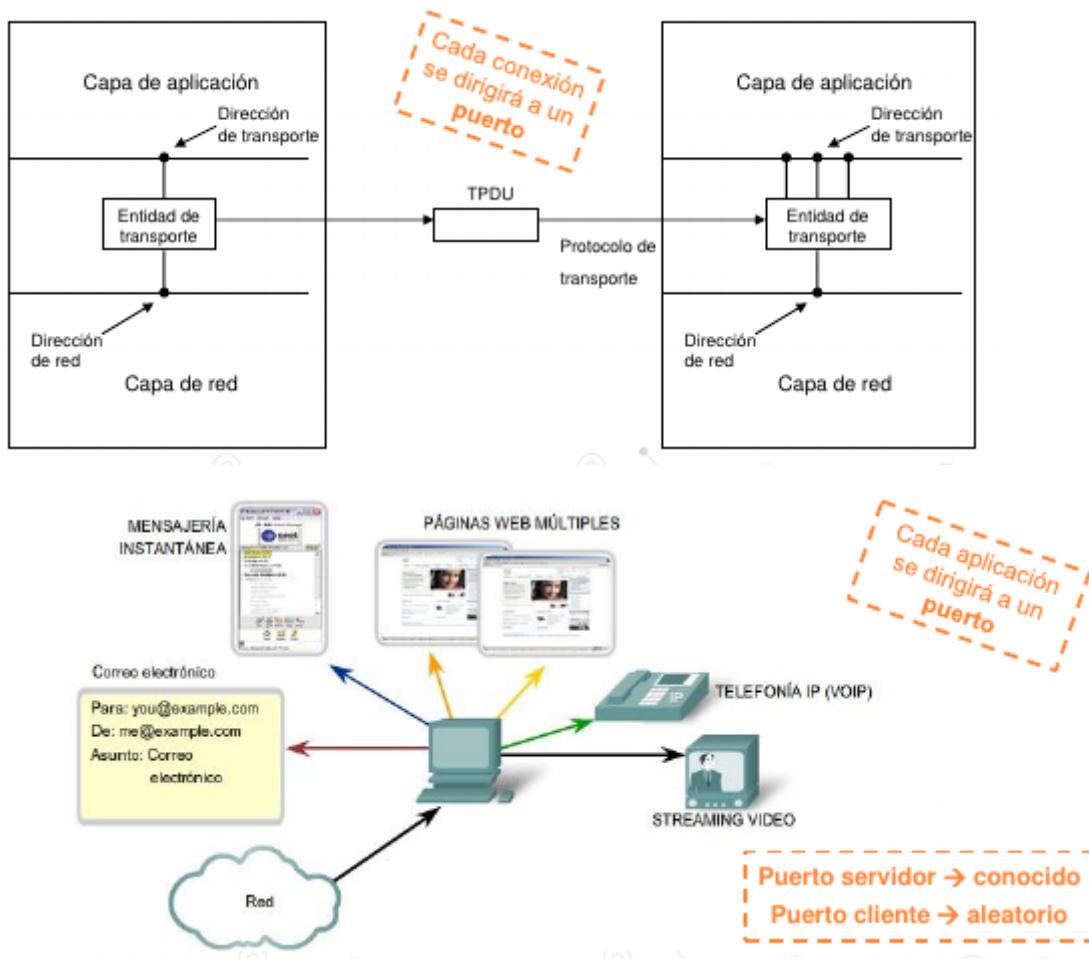
- Red en modelo OSI
- Internet en modelo TCP/IP

**Prepara los datos de la aplicación para su transporte en la red** y procesa los datos recibidos por la red para su uso en las aplicaciones.

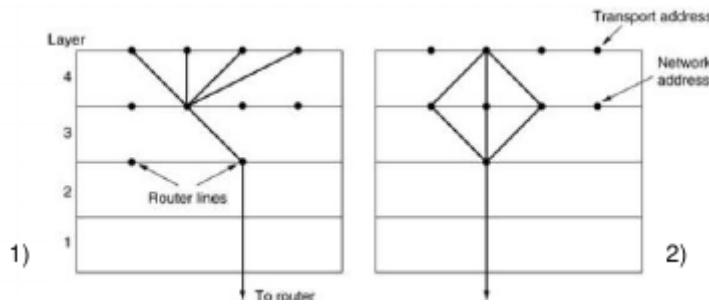
Protocolos de las capas de aplicación y transporte:



## Multiplexación de comunicaciones (de aplicaciones)



1. Varias conexiones de transporte en una conexión de red. Se utiliza cuando el coste por conexión del servicio de red es elevado
2. Una conexión de transporte en varias conexiones de red. Se utiliza cuando se quiere aumentar el caudal o reducirse el retardo en una conexión de transporte



## Segmentación de datos

La capa de transporte permite la segmentación de datos y brinda el control necesario para reensamblar las partes dentro de los distintos flujos de datos.

Las responsabilidades principales que debe cumplir son:

- Seguimiento de la comunicación individual entre origen y destino.
- **Segmentación de datos (de aplicación) y manejo de cada parte:**
  - Cada aplicación crea datos para enviarse a una aplicación remota.
  - Estos datos se deben preparar para ser enviados a través de los medios.
  - Los protocolos de la capa de transporte describen los servicios que segmentan estos datos.
  - Se requiere que se agreguen encabezados en la capa de transporte para indicar la comunicación a la cual está asociada e identificar las partes de los segmentos.
    - La segmentación facilita la multiplexación porque los segmentos son más fáciles de administrar y controlar (errores, flujo, etc.)
- **Reensamblaje de segmentos:**
  - Al recibirlos, cada segmento de datos se traslada a la aplicación adecuada.
  - Los segmentos de datos individuales deben unirse para reconstruir una trama completa de datos que sea útil para la capa de aplicación.
  - Los protocolos en la capa de transporte describen cómo se utiliza la información del encabezado de la capa para reensamblar las partes de los diferentes segmentos recibidos y pasarlo a la capa de aplicación.
    - Los segmentos/datos deberán llegar en una secuencia específica para ser usados por la aplicación.
    - Se espera recibir todos los datos, aunque algunas aplicaciones toleran pérdidas.
- **Identificación de diferentes aplicaciones en origen y destino:**
  - Para pasar la trama de datos a las aplicaciones adecuadas, la capa de transporte debe identificar cada aplicación final.
  - La capa de transporte asigna un identificador a la aplicación.
  - Los protocolos TCP/IP denominan a este identificador **número de puerto**.
- Multiplexación y Demultiplexación del tráfico de las aplicaciones.

## Puertos

### Asignación de puertos:

- **Estática:** existe una autoridad central que asigna los números de puerto conforme se necesitan y publica la lista de todas las asignaciones. Este enfoque se conoce como enfoque universal y las asignaciones de puerto especificadas se conocen como asignaciones bien conocidas.

- **Dinámica:** siempre que un proceso necesita un puerto el software de red le asignará uno. Se asigna de forma aleatoria dentro de un rango y evitando los puertos bien conocidos.

Los diseñadores de TCP/IP adoptaron una solución híbrida, que preasigna muchos números de puerto pero que también deja muchos de ellos disponibles.

### Rangos:

- El campo de puerto tiene una longitud de 16 bits, lo que permite un rango que va desde 0 a 65535, pero no todos estos puertos son de libre uso.
- El puerto 0 es un puerto reservado, pero utilizado si el emisor no permite respuestas del receptor.
- Los puertos 1 a 1023 reciben el nombre de Puertos bien conocidos.
  - En sistemas Unix, para enlazar con ellos ('abrirlos'), es necesario tener acceso como superusuario
- **Los puertos 1024 a 49151 son los llamados Puertos registrados, y son los de libre utilización.**
- Los puertos del 49152 al 65535 son Puertos efímeros, de tipo temporal, y se utilizan sobre todo por los clientes al conectar con el servidor.

### PUERTOS BIEN CONOCIDOS (I) (RFC 6335)

20 (TCP), utilizado por FTP (File Transfer Protocol) para datos  
 21 (TCP), utilizado por FTP (File Transfer Protocol) para control  
 22 (TCP), utilizado por SSH (Secure Shell)  
 23 (TCP), utilizado por TELNET (Teletype Network)  
 25 (TCP), utilizado por SMTP (Simple Mail Transfer Protocol)  
 53 (TCP), utilizado por DNS (Domain Name System)  
 53 (UDP), utilizado por DNS (Domain Name System)  
 67 (UDP), utilizado por BOOTP BootStrap Protocol (Server) y por DHCP  
 68 (UDP), utilizado por BOOTP BootStrap Protocol (Client) y por DHCP  
 69 (UDP), utilizado por TFTP (Trivial File Transfer Protocol)  
 80 (TCP), utilizado por HTTP (HyperText Transfer Protocol)  
 88 (TCP), utilizado por Kerberos (agente de autenticación)  
 110 (TCP), utilizado por POP3 (Post Office Protocol)  
 137 (TCP), utilizado por NetBIOS (servicio de nombres)  
 137 (UDP), utilizado por NetBIOS (servicio de nombres)  
 138 (TCP), utilizado por NetBIOS (servicio de envío de datagramas)  
 138 (UDP), utilizado por NetBIOS (servicio de envío de datagramas)

### PUERTOS BIEN CONOCIDOS (II) (RFC 6335)

139 (TCP), utilizado por NetBIOS (servicio de sesiones)  
 139 (UDP), utilizado por NetBIOS (servicio de sesiones)  
 143 (TCP), utilizado por IMAP4 (Internet Message Access Protocol)  
 443 (TCP), utilizado por HTTPS/SSL (transferencia segura de páginas web)  
 631 (TCP), utilizado por CUPS (sistema de impresión de Unix)  
 993 (TCP), utilizado por IMAP4 sobre SSL  
 995 (TCP), utilizado por POP3 sobre SSL  
 1080 (TCP), utilizado por SOCKS Proxy  
 1433 (TCP), utilizado por Microsoft-SQL-Server  
 1434 (TCP), utilizado por Microsoft-SQL-Monitor  
 1434 (UDP), utilizado por Microsoft-SQL-Monitor  
 1701 (UDP), utilizado para Enrutamiento y Acceso Remoto para VPN con L2TP.  
 1723 (TCP), utilizado para Enrutamiento y Acceso Remoto para VPN con PPTP.  
 1761 (TCP), utilizado por Novell Zenworks Remote Control utility  
 1863 (TCP), utilizado por MSN Messenger

## Resumen

- **Funciones y servicios de la capa de transporte:**
  - Comunicación extremo a extremo (end-to-end).
  - Multiplexación/demultiplexación de aplicaciones → puerto.
- **Protocolo UDP:**
  - Multiplexación/demultiplexación de aplicaciones (puertos).
  - Servicio no orientado a conexión, no fiable.
- **Protocolo TCP:**
  - Multiplexación/demultiplexación de aplicaciones (puertos).

- Servicio orientado a conexión, fiable:
  - Control de errores
  - Control de flujo
  - Control de congestión

## 2. Protocolo de datagrama de usuario (UDP)

Proporciona un servicio de entrega de datagramas:

- No orientado a conexión:
  - sin conexión previa (no hand-shaking).
  - no hay retardo de establecimiento de la conexión.
  - cada TPDU (datagrama) es independiente.
- No confiable:
  - no se comprueban errores.
  - puede haber pérdidas de paquetes.
- No hay garantía de entrega ordenada.
- No hay control de congestión (se entrega tan rápido como se pueda).
- Multiplexación/demultiplexación (transportar el TPDU al proceso/aplicación correcto).

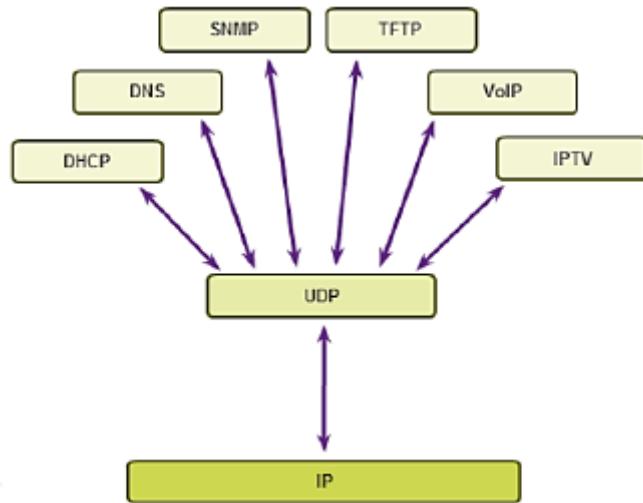
UDP utiliza IP (capa Red), pero agrega la capacidad para distinguir entre varias aplicaciones de destino dentro de un mismo sistema destino.

Una aplicación que utiliza UDP, asume la responsabilidad por los problemas de confiabilidad, incluyendo la pérdida, duplicación y retraso de los paquetes, así como la entrega desordenada de los mismos o las posibles pérdidas de conectividad.

UDP proporciona puertos de protocolo para distinguir entre muchos programas que se ejecutan dentro de una misma máquina.

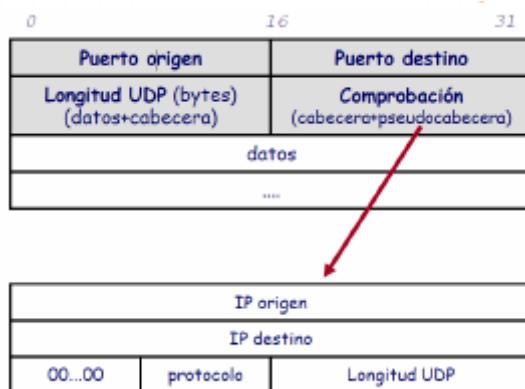
UDP suele utilizarse para:

- Aplicaciones de streaming multimedia (tolerantes a pérdidas, pero sensibles a retardos).
- Intercambio de mensajes (escaso).
  - Ej: Consultas DNS (< 512 bytes).
- Aplicaciones en tiempo real (no pueden esperar confirmaciones).
  - Ej: videoconferencia, voz sobre IP.
- Mensajes producidos periódicamente, ya que no importa si se pierde alguno.
  - Ej: SNMP (Simple Network Management Protocol)
- Para el envío de tráfico broadcast/multicast.



## Formato datagrama UDP

Cabecera:



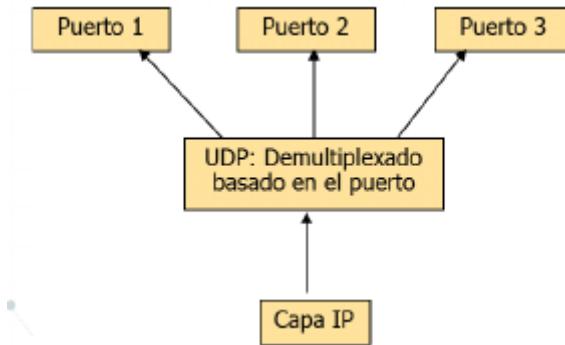
- Los números de puerto identifican los procesos emisor y receptor (ojo, puertos UDP distintos a TCP)
- **Longitud UDP** → longitud de la cabecera UDP + longitud de datos (valor mínimo 8 bytes)
- **Datos** → PDU de la capa superior
- **Comprobación** (Checksum):
  - Se calcula sobre la cabecera UDP y los datos UDP
  - Complemento a 1 de la suma de todo el datagrama.
  - El datagrama UDP puede contener un número impar de bytes → Se añade un byte de relleno (todo ceros).
  - Se considera una pseudo-cabecera de 12 bytes para el cálculo del checksum, que contiene algunos campos de la cabecera IP. (Doble comprobación de estos campos)

## Puertos UDP

UDP acepta datagramas de muchos programas de aplicación. Pasa los datagramas al nivel de red IP para su transmisión y los recibe de ese nivel en el otro extremo.

El multiplexado y demultiplexado entre el software UDP y los programas de aplicación ocurre a través del mecanismo de puerto (identificación).

Los puertos UDP son distintos a los puertos TCP.



### 3. Protocolo de control de transmisión (TCP)

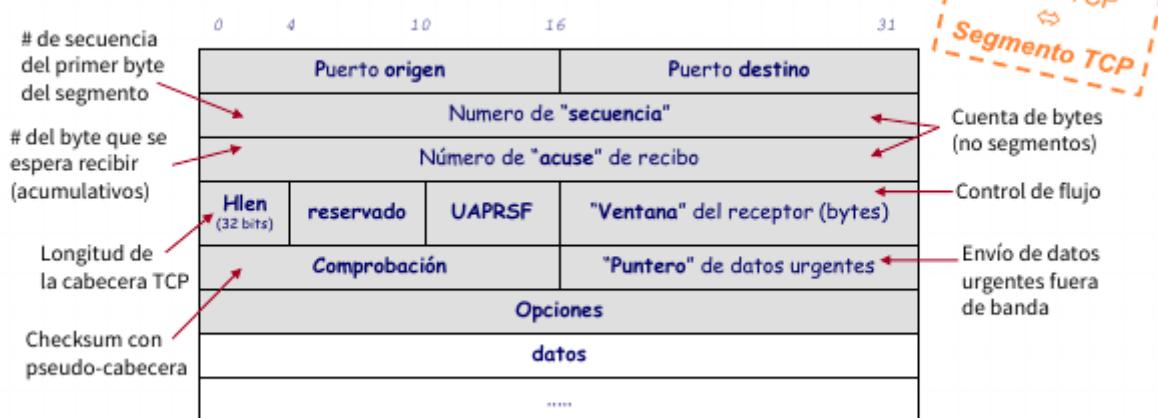
- **Servicio orientado a conexión:** exige un acuerdo entre emisor y receptor (hand shaking).
- **Entrega ordenada:** de las secuencias de bytes generadas por las aplicaciones (stream oriented).
- Transmisión **full duplex:** se pueden enviar datos en ambos sentidos al mismo tiempo.
- **Mecanismo de detección y recuperación de errores** (ARQ – Automatic Repeat reQuest):
  - Con confirmaciones positivas (ACKs) acumulativas.
  - Timeouts adaptables.
  - Incorporación de confirmaciones con los datos (piggybacking)
- **Servicio fiable:** control de congestión y control de flujo con ventanas deslizantes con tamaño máximo adaptable.
- **Servicio punto a punto:** no puede usarse para multicast.

#### Servicios que ofrece TCP

- **Establecimiento y cierre de la conexión:**
  - Al ser un protocolo orientado a conexión, dispone de mecanismos para establecer la conexión (antes de la transmisión de datos) y para cerrarla (al final de la transmisión).
- **Control de errores y de flujo:**
  - Se garantiza la recepción correcta y ordenada de los datos en la aplicación destino tal y como los generó la aplicación origen.
  - Es capaz de ajustar las diferencias que haya entre la tasa de generación de datos (en el origen) y la de consumo de los mismos (destino).
- **Control de congestión:**
  - Gestiona los recursos de la red (ancho de banda, almacenamiento temporal en los routers) para evitar su agotamiento, adaptando el tráfico a generar.
- **Multiplexación de aplicaciones:**
  - Al igual que UDP, utiliza puertos para dirigir los datos a las aplicaciones pertinentes en el destino.

#### Cabecera TCP

## CABECERA TCP



- Cada **segmento TCP** se encapsula en un **datagrama IP**.

## Multiplexación/demultiplexación

Consiste en transportar los segmentos a la aplicación correcta. Se realiza (al igual que en UDP) utilizando puertos asociados a cada aplicación.

Existen puertos preasignados (ojo, puertos TCP distintos de puertos UDP):

Puerto	Aplicación/Servicio	Descripción
20	<b>FTP-DATA</b>	Transferencia de ficheros: datos
21	<b>FTP</b>	Transferencia de ficheros: control
22	<b>SSH</b>	Terminal Seguro
23	<b>TELNET</b>	Acceso remoto
25	<b>SMTP</b>	Correo electrónico
53	<b>DNS</b>	Servicio de nombres de dominio
80	<b>HTTP</b>	Acceso hipertexto (web)
110	<b>POP3</b>	Descarga de correo

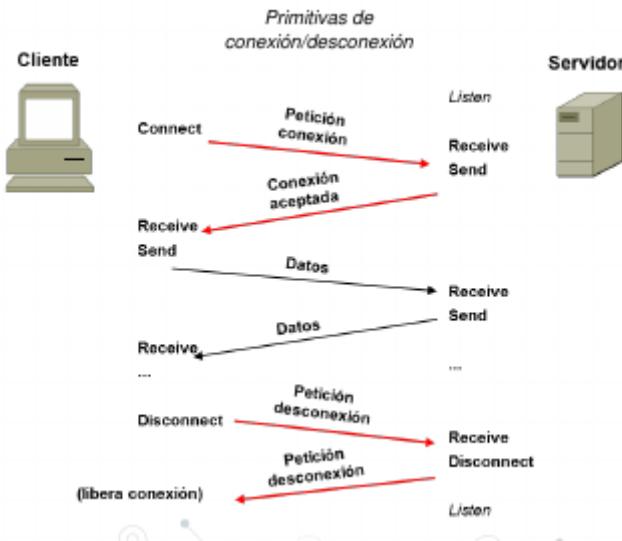
Una conexión TCP se identifica por IP y puerto origen, IP y puerto destino

## Control de conexión

El intercambio de información tiene tres fases:

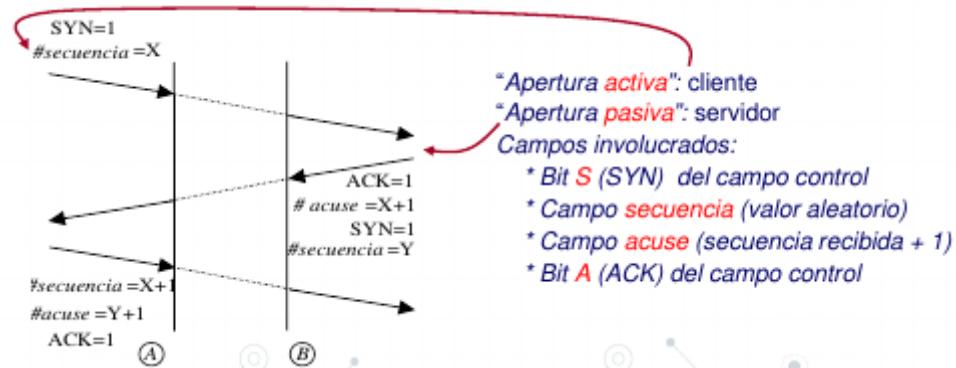
- Establecimiento de la conexión (sincronizar # de secuencia y reservar recursos).
- Intercambio de datos (full-duplex).
- Cierre de la conexión (liberar recursos).

Es un mecanismo de sincronización entre emisor y receptor para **garantizar una comunicación ordenada y sin errores**.



### Establecimiento de la conexión

¿Se podría garantizar un establecimiento/cierre fiable de la conexión sobre un servicio no fiable (IP)? NO, por ello se establece la conexión con three-way handshaking.



### Números de secuencia

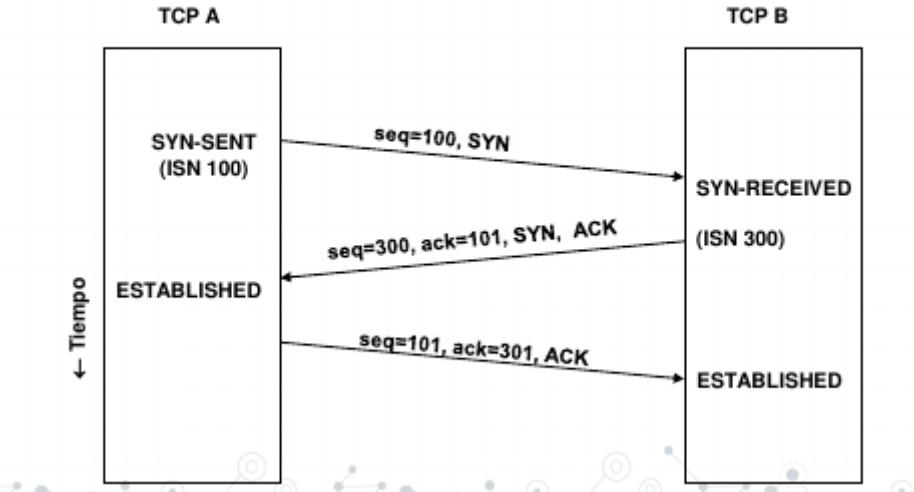
El número de secuencia es un campo de 32 bits que cuenta bytes en módulo  $2^{32}$  (el contador se da la vuelta cuando llega al valor máximo).

El número de secuencia no empieza normalmente en 0, sino en un valor denominado ISN (Initial Sequence Number) elegido “teóricamente” al azar; para evitar confusiones con transmisiones anteriores. El ISN es elegido por el sistema (cliente o servidor).

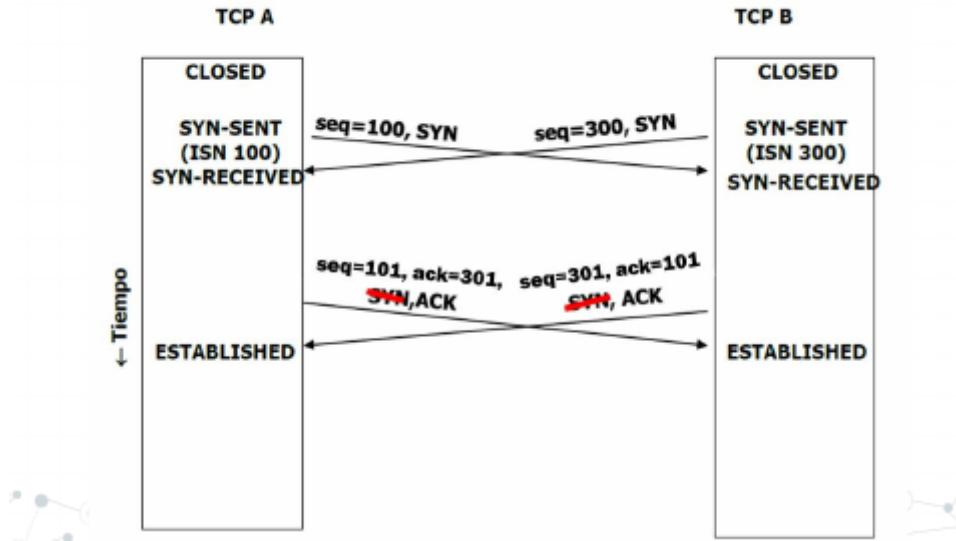
El estándar sugiere utilizar un contador entero incrementado en 1 cada 4 µs aproximadamente. En este caso el contador se da la vuelta (y el ISN reaparece) al cabo de 4 horas 46 min.

- El mecanismo de selección de los ISN es suficientemente fiable para proteger de coincidencias, pero no es un mecanismo de protección frente a sabotajes.
  - Es muy fácil averiguar el ISN de una conexión e interceptarla suplantando a alguno de los dos participantes.
- TCP incrementa el número de secuencia de cada segmento según los bytes que tenía el segmento anterior, con una excepción:
  - Cuando los flags SYN y FIN están activos, se incrementa en 1 el número de secuencia.
- La presencia además del flag ACK activo implica que no se incrementa el número de secuencia (no hay datos).

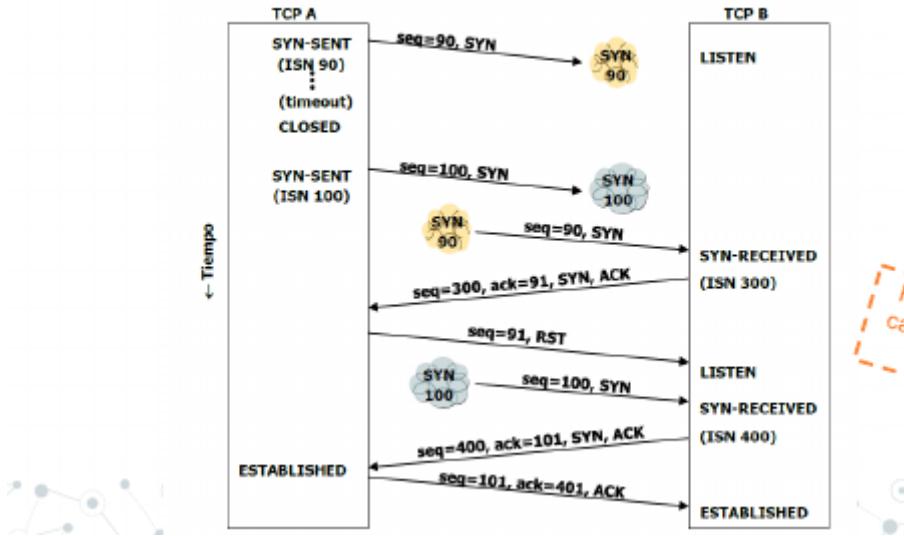
### Ejemplo: *three-way handshaking*



### Ejemplo: *three-way handshaking (Conexión simultánea)*



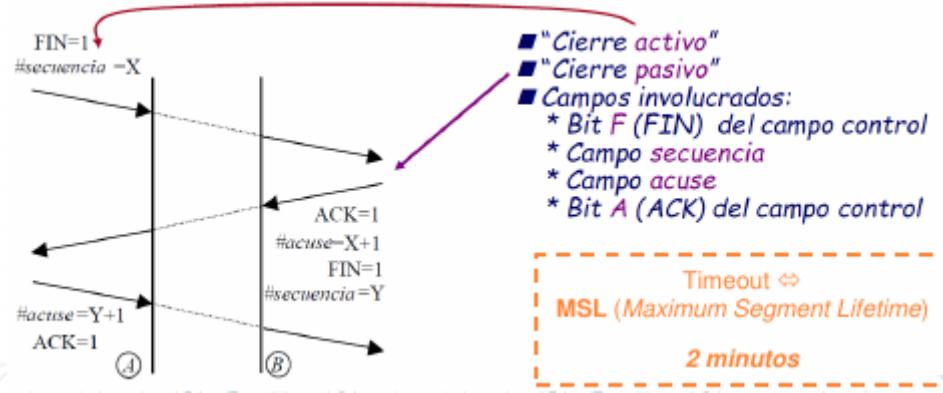
### Ejemplo: *three-way handshaking (SYN retrasados y duplicados)*



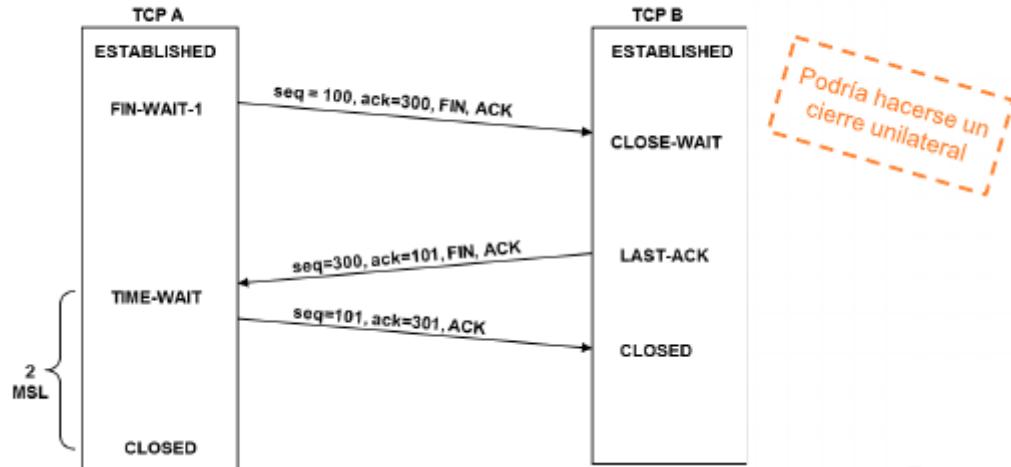
### Cierre

Sincronización para el cierre de la conexión y liberación de recursos asociados a la misma.

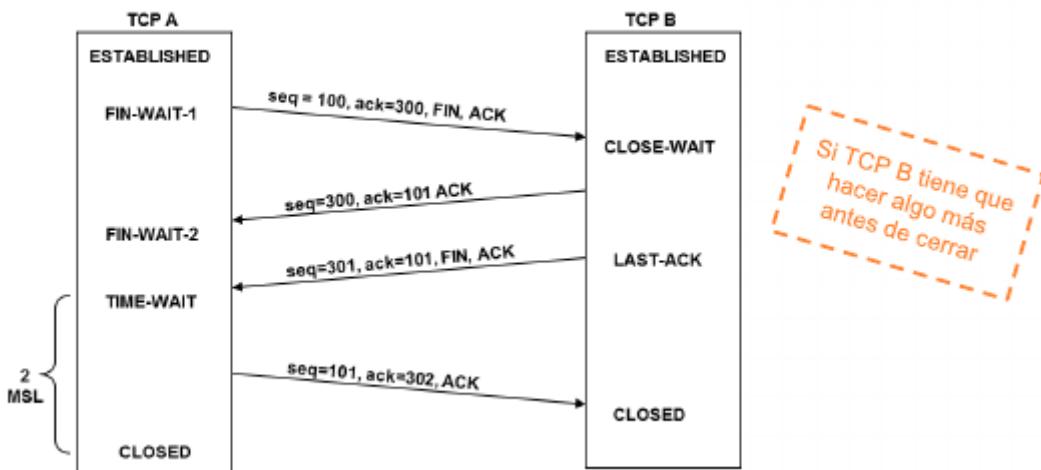
Una vez comenzado el procedimiento de cierre no se cierra inmediatamente por si hay paquetes en tránsito, sino que se usan timeouts.



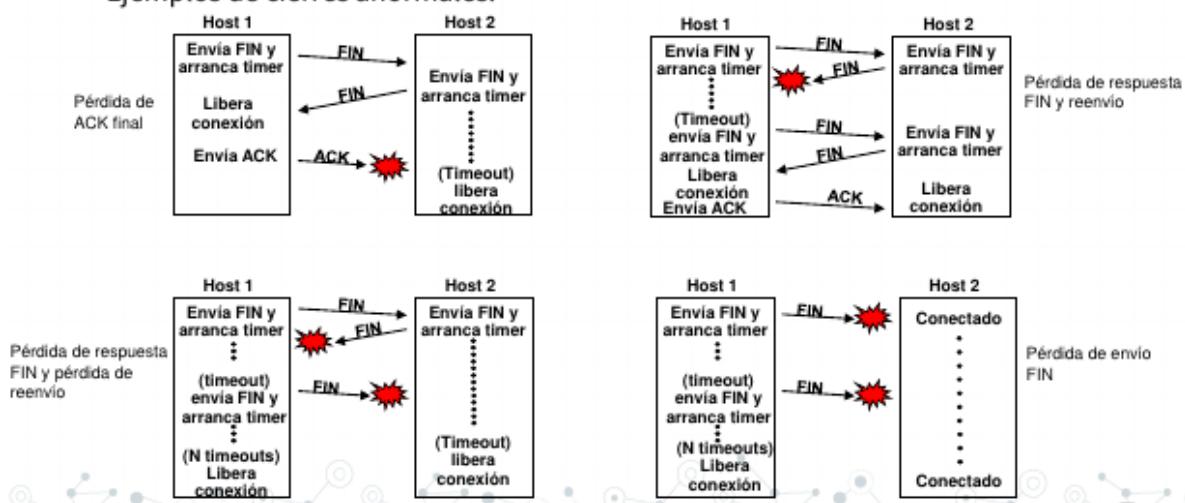
Ejemplo: cierre habitual en tres pasos (con *timeout*)



Ejemplo: cierre en cuatro pasos (con *timeout*)



#### Ejemplos de cierres anormales:



## Intercambio de datos

TCP a APLICACIÓN:

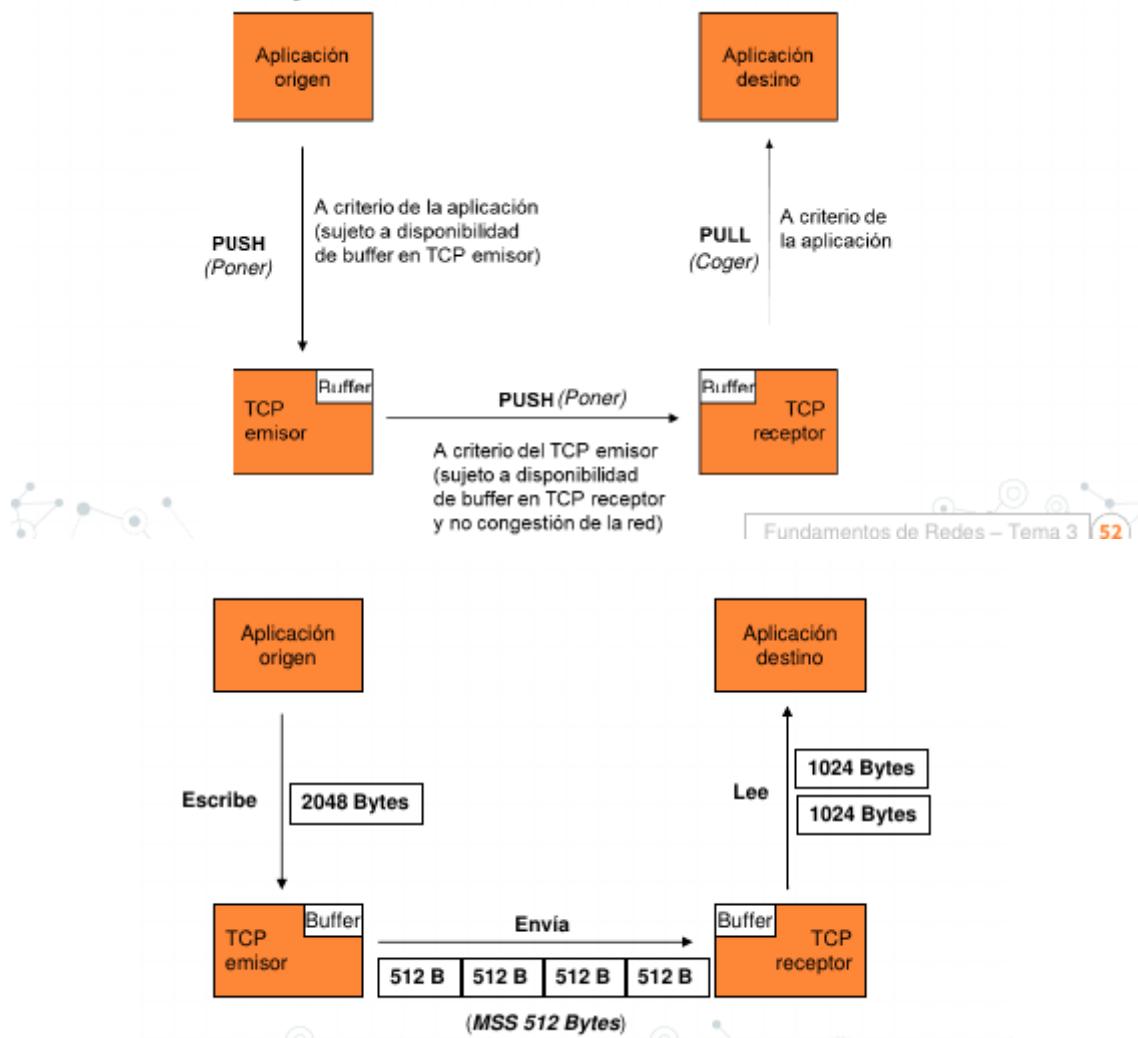
- El intercambio de datos lo realizan una aplicación en el origen y otra aplicación en el destino.
  - **Aplicación → TCP:** la aplicación envía los datos a TCP cuando quiere (siempre y cuando TCP tenga espacio libre en el buffer de emisión).
  - **TCP → Aplicación:** la aplicación lee del buffer de recepción de TCP cuando quiere y cuanto quiere. **Excepción: datos urgentes.**

Para TCP, los datos de la aplicación son un flujo continuo de bytes, independientemente de la separación que pueda tener la aplicación (registros, etc.). Es responsabilidad de la aplicación asegurarse de que esa separación (si existe) se mantenga después de transmitir los datos.

TCP a TCP:

- El TCP emisor manda los datos cuando quiere. **Excepción: datos “pushed”.**
- El TCP emisor decide el tamaño de segmento según sus preferencias. Al inicio de la conexión se negocia el MSS (Maximum Segment Size).
- Normalmente TCP intenta agrupar los datos para que los segmentos tengan la longitud máxima, reduciendo así el overhead (sobrecarga) debido a cabeceras y proceso de segmentos.

### TCP ↔ APLICACIÓN y TCP ↔ TCP



## Casos excepcionales

- **Datos "Pushed" (bit PSH):** La aplicación pide al TCP emisor que envíe esos datos lo antes posible (sin esperar a tener un segmento de datos completo). El TCP receptor los pondrá a disposición de la aplicación de inmediato, para cuando ésta le pida datos.
  - Ejemplo: telnet.
- **Datos Urgentes (bit URG y Urgent Offset):** Los datos se quieren entregar a la aplicación remota sin esperar a que esta los pida.
  - Ejemplo: abortar un programa con CTRL-C en una sesión telnet

## Control de errores

Para el control de errores se sigue un esquema ARQ (Automatic Repeat-reQuest) con confirmaciones positivas y acumulativas. No hay "confirmaciones negativas", y se pueden confirmar varios segmentos a la vez.

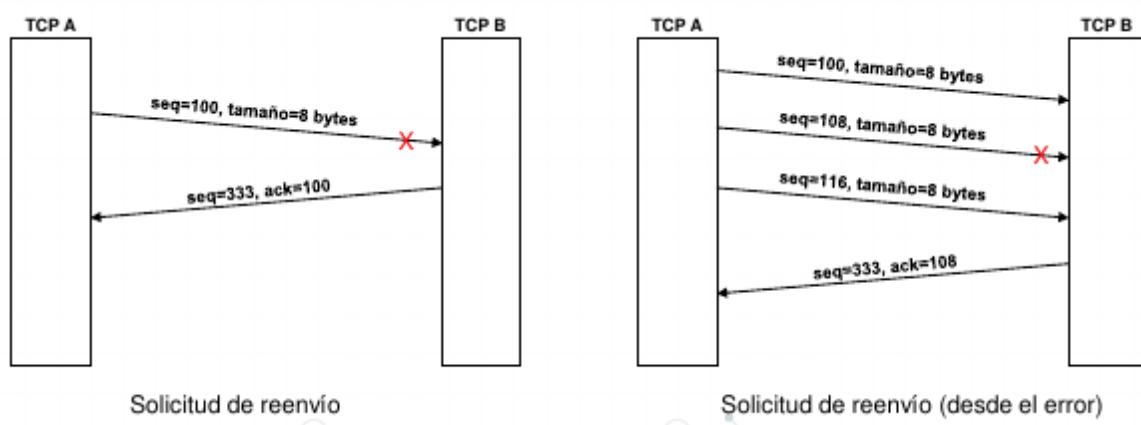
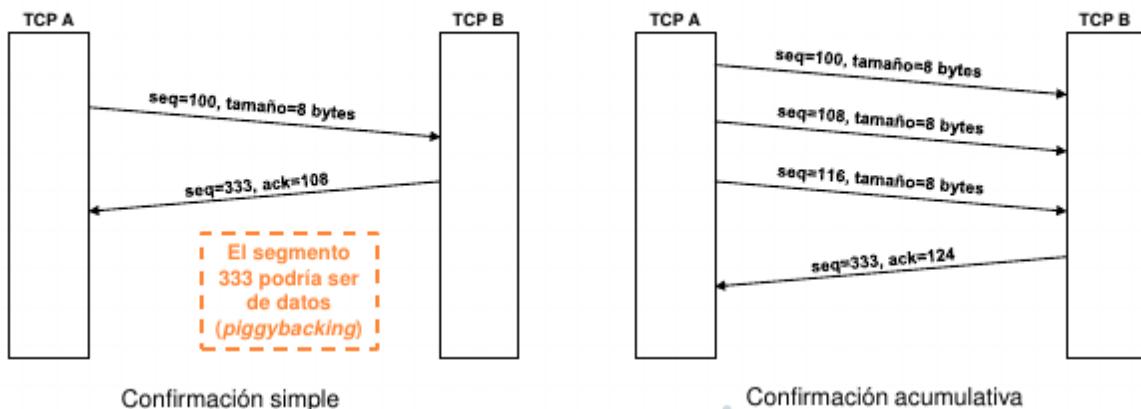
- **Campos involucrados:**

- Campo secuencia: offset (en bytes) dentro del mensaje.
- Campo acuse: número de byte esperado en el receptor.
- Bit A (ACK) del campo de control.
- Campo comprobación: checksum de todo el segmento y uso de pseudo-cabecera.

- **Confirmaciones mediante piggybacking:**

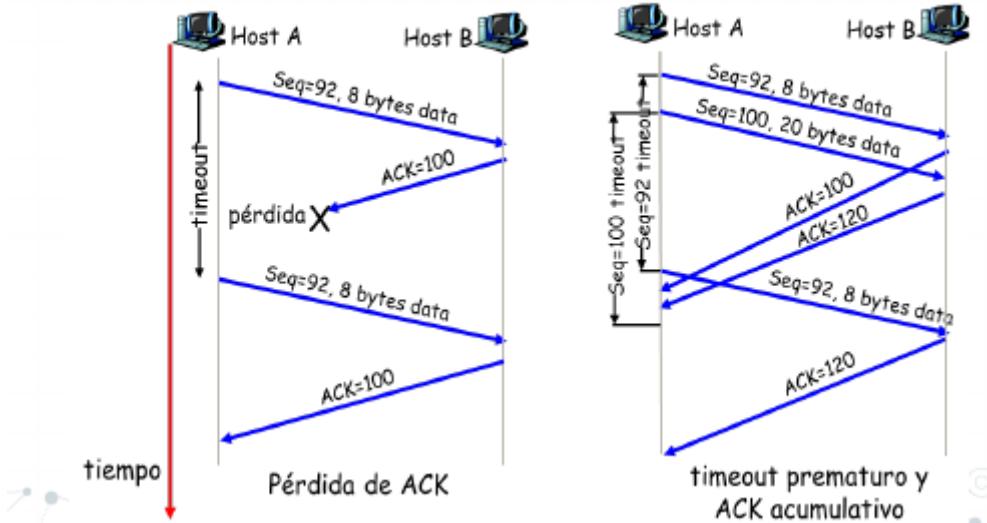
- Se envía la confirmación en un segmento con datos enviado en el otro sentido (los campos acuse y A se usan en un segmento de datos)

- **ARQ: Funcionamiento habitual**



## ARQ: Retransmisión

(Gráfico J.F. Kurose)



Protocolo de generación de ACKS:

EVENTO	ACCIÓN DEL TCP RECEPTOR
I llegada ordenada del segmento, sin discontinuidad, todo lo anterior ya confirmado	Retrasar ACK. Esperar a recibir el siguiente segmento hasta 500ms, si no llega enviar ACK
I llegada ordenada del segmento, sin discontinuidad, hay pendiente un ACK retrasado	Inmediatamente enviar un único ACK acumulativo
I llegada desordenada del segmento con número de secuencia mayor de lo esperado, discontinuidad detectada	Enviar un ACK duplicado, indicando el número de secuencia del siguiente bit esperado
I llegada de un segmento que completa una discontinuidad parcial o totalmente	Confirmar ACK inmediatamente si el segmento comienza en el extremo inferior de la discontinuidad

### ¿Cómo estimamos los timeouts?

- Debe ser mayor que el tiempo de ida-vuelta (RTT), pero ¿cuánto?
  - Si nos quedamos cortos, tendremos timeouts prematuros, con retransmisiones innecesarias
  - Si nos pasamos, la reacción será demasiado lenta, poco eficaz
- Para situaciones cambiantes, soluciones dinámicas:

- **RTTmedido:** tiempo desde la emisión de un segmento hasta la recepción del ACK.

$$RTT_{nuevo} = \alpha \cdot RTT_{viejo} + (1-\alpha) \cdot RTT_{medido}, \quad \alpha \in [0,1]$$

$$Desviacion_{nueva} = (1-x) * Desviacion_{vieja} + x * |RTT_{medido} - RTT_{nuevo}|$$

$$\text{Timeout} = RTT_{nuevo} + 4 * Desviacion$$

- Problema con ACKs repetidos: ambigüedad en la interpretación.

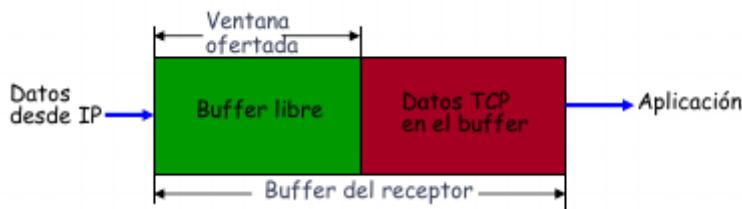
- Solución: Algoritmo de Karn, actualizar el RTT sólo para los no ambiguos, pero si hay que repetir un segmento duplicar el timeout:
- $tout_{nuevo} = \gamma \cdot tout_{viejo}, \gamma = 2$

## Control de flujo

Procedimiento para evitar que el emisor sature al receptor con el envío de demasiada información y/o demasiado rápido. Es un esquema crediticio: el receptor informa al emisor sobre los bytes autorizados a emitir sin esperar respuesta.

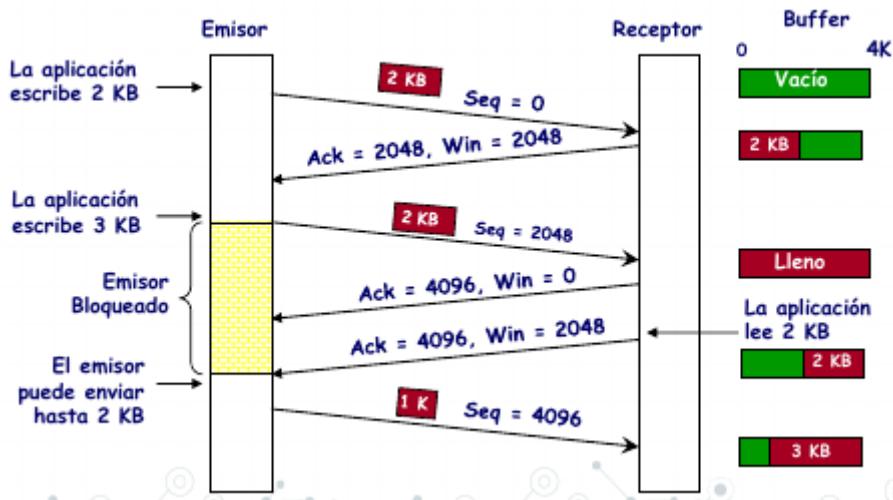
Se utiliza el campo ventana, un esquema de ventana deslizante:

```
ventana util emisor = ventana ofertada receptor - bytes en tránsito
```



- El TCP receptor informa en cada segmento al emisor del espacio que le queda libre en el buffer para esa comunicación. Para ello usa el campo tamaño de ventana (WIN).
- Anunciando una ventana cero el receptor puede bloquear al emisor, y ejercer así control de flujo.
- La ventana anunciada (u ofertada) es un espacio que el TCP receptor reserva para esa comunicación en su buffer.

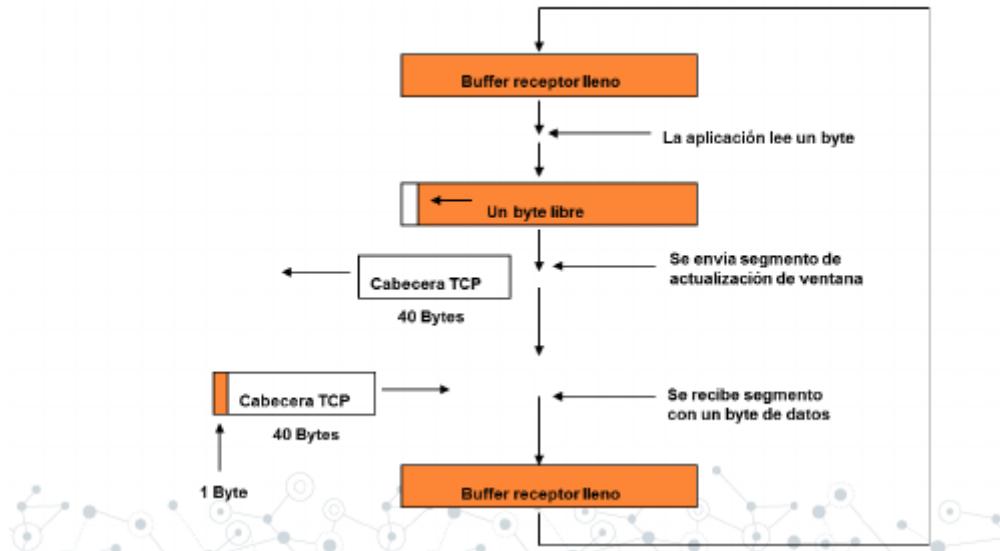
Tanto los números de secuencia como los tamaños de ventana se indican en bytes.



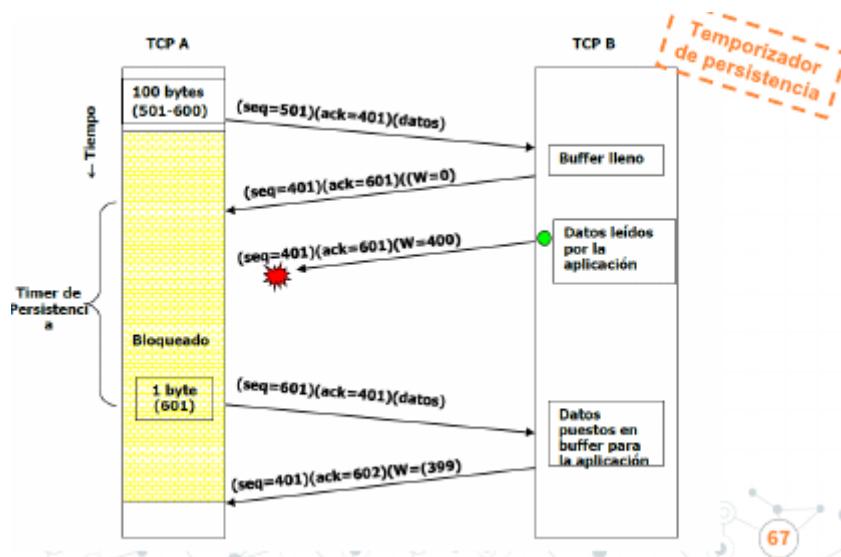
Si se perdiera el anuncio de la ventana disponible en el receptor, el emisor podría quedar bloqueado.

- Possible problema: síndrome de la ventana tonta (RFC 813) si se utilizan segmentos muy pequeños.

### Síndrome de la ventana tonta (RFC 813)



- Posible mejora: la ventana optimista (RFC 813) o solución de Clark:
  - El TCP receptor solo debe notificar una nueva ventana cuando tenga una cantidad razonable de espacio libre. Razonable significa:
    - Un MSS (segmento del tamaño máximo), o bien
    - La mitad del espacio disponible en el buffer.



## Control de congestión

Adaptación a las características o rendimiento de la red (RFC 2001). Es un problema debido a la insuficiencia de recursos (la capacidad o velocidad de transmisión de las líneas y el buffer en routers y hosts no son infinitos).

Es un problema diferente al control del flujo: el control de congestión es para proteger a la red debido a sus limitaciones.

Los episodios de congestión se manifiestan en retrasos en las ACKs y/o pérdidas de segmentos, dependiendo del nivel de severidad del episodio.

- Solución extremo a extremo: en el emisor limitar de forma adaptable el tráfico generado para evitar pérdidas, pero siendo eficaz.
- La limitación se hace mediante una aproximación conservadora: limitando el tamaño de la ventana de emisión.

- Cuando hay congestión TCP debe de reducir el flujo de datos.
  - El mecanismo para detectarla es implícito, por la pérdida de segmentos. Cuando ocurre TCP baja el ritmo.
- Además de la ventana de control de flujo (dictada por el receptor y transmitida en la cabecera TCP) el emisor tiene una ventana de control de congestión, que se ajusta a partir de los segmentos perdidos.
  - En cada momento se usa la más pequeña de ambas.
- El mecanismo de control de congestión de TCP se denomina arranque lento (slow-start) y fue diseñado por Van Jacobson en los años 80.

### **Slow start**

El emisor utiliza dos ventanas y un umbral.

```
bytes que se pueden enviar = mínimo(ventana congestión, ventana receptor)
```

```
ventana receptor: usada para control de flujo (tamaño variable) según el campo "ventana" recibido  
ventana congestión: inicialmente, 1*MSS
```

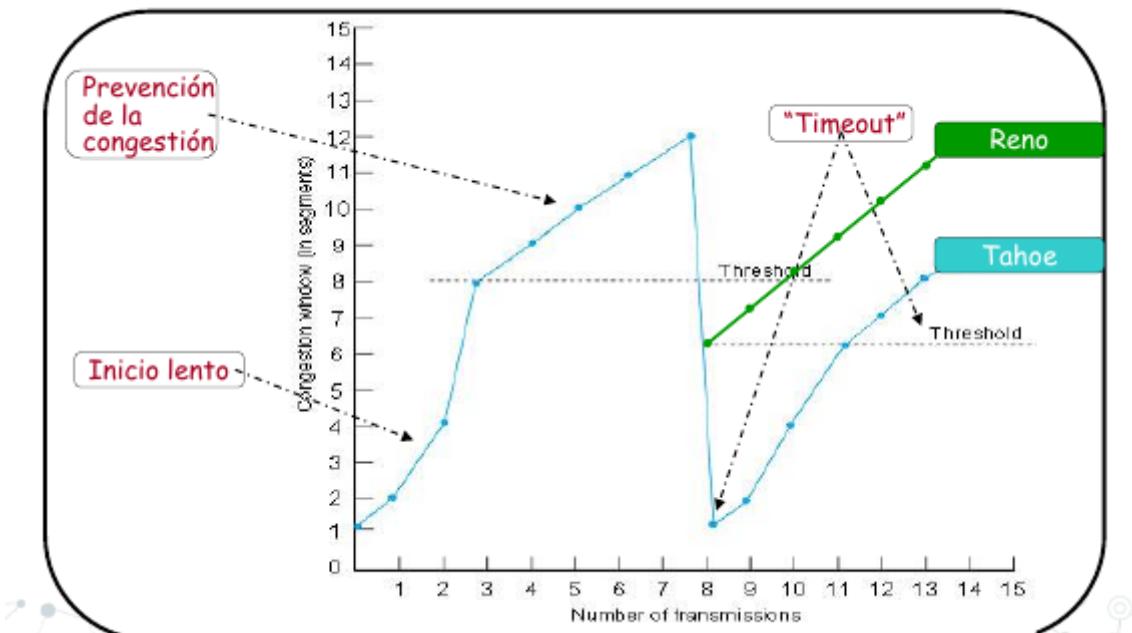
```
Si ventana congestión < umbral, por cada ACK recibido...  
ventana congestión += MSS (CRECIMIENTO EXPONENCIAL)
```

### **Prevención de la congestión**

```
Si ventana congestión > umbral, cada vez que se reciben todos ACKs pendientes...  
ventana congestión += MSS (CRECIMIENTO LINEAL)
```

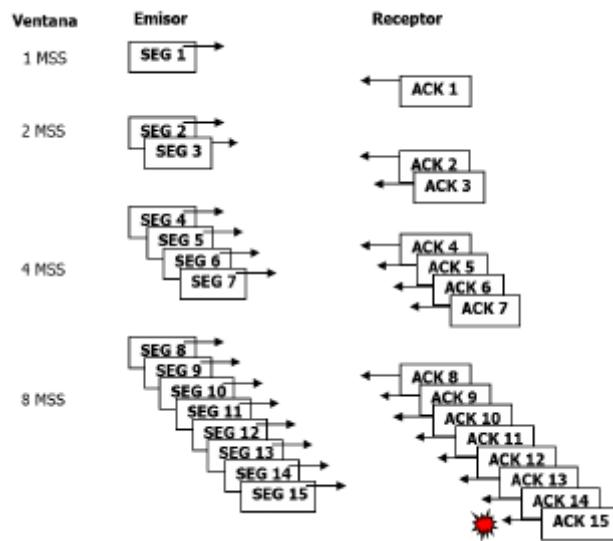
### **Timeout**

```
Si timeout:  
umbral = ventana congestión / 2  
ventana congestión = MMS
```



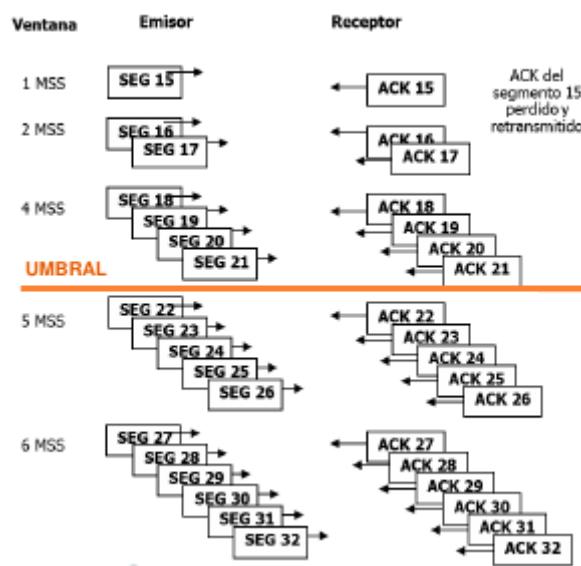
## Primera fase: inicio lento

1. Inicialmente la ventana de congestión tiene el tamaño de un MSS
2. Por cada segmento enviado con éxito la ventana se amplía en un MSS
  - En la práctica esto supone un crecimiento exponencial (en potencias de dos en cada ACK).
3. Si la ventana de congestión supera a la de control de flujo se aplica la restricción de ésta última con lo cual aquella deja de crecer.



## Segunda fase: prevención de congestión, cuando se pierde un segmento:

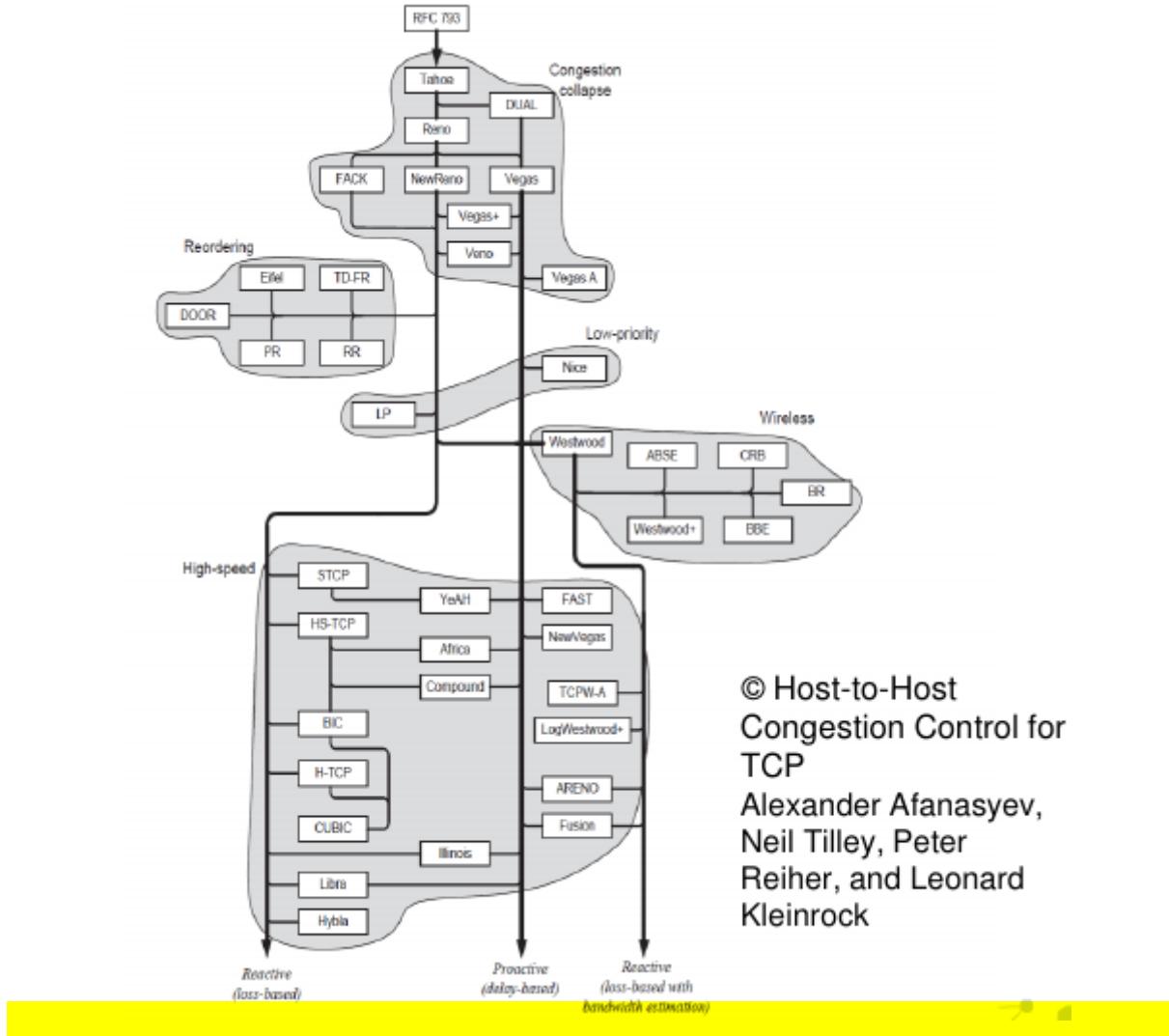
1. La ventana de congestión vuelve a su valor inicial.
2. Se fija un 'umbral de peligro' en un valor igual a la mitad de la ventana que había cuando se produjo la pérdida.
3. La ventana de congestión crece como antes hasta el umbral de peligro; a partir de ahí crece en sólo un segmento cada vez que se recibe un ACK.



## 4. Extensiones TCP

TCP se define con múltiples "Sabores". Los diferentes sabores no afectan a la interoperabilidad entre los extremos.

Desde cualquier versión de Linux con kernel mayor que la 2.6.19 se usa por defecto TCP CuBIC



## TEMA 4: SEGURIDAD EN REDES

### 1. Introducción a la seguridad en redes

Una red de comunicaciones es segura cuando se garantizan todos los aspectos de la misma. No hay protocolos ni redes 100% seguros, pero podemos intentarlo. ¿Qué es la seguridad? múltiples aspectos:

- **Confidencialidad/privacidad:** el contenido de la información es comprensible sólo para entidades autorizadas.
- **Autenticación:** las entidades son quienes dicen ser.
- **Control de accesos:** los servicios son accesibles sólo para entidades autorizadas.
- **No repudio o irrenunciabilidad:** el sistema impide la renuncia de la autoría de una determinada acción.
- **Integridad:** el sistema detecta todas las alteraciones (intencionadas o no) de la información.
- 
- **Disponibilidad:** el sistema mantiene las prestaciones de los servicios con independencia de la demanda.

¿En qué nivel/capa se debe situar la seguridad? en TODOS... el grado de seguridad lo determina el punto más débil.

- **Ataque de seguridad:** cualquier acción intencionada o no que menoscaba cualquiera de los aspectos de la seguridad.



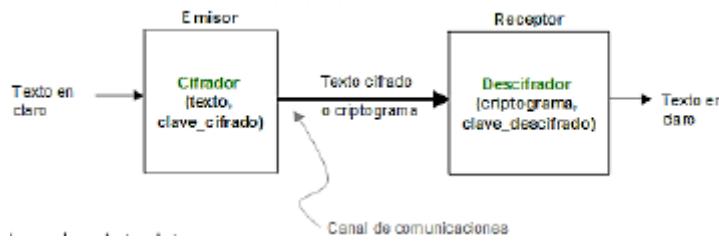
- **Sniffing - Intercepción:** vulneración a la confidencialidad, escuchar (husmear).
- **Poofing/Phising - Fabricación:** suplantación de la identidad de entidades.
- **Man in the middle - Intercepción/Modificación:** hombre/máquina en medio.
- **Distributed denial of service - Interrupción:** denegación de servicio distribuido,
- **Malware:**
  - Troyano: software oculto con la apariencia de otro programa)
  - Gusano: virus que se replica
  - Spyware: captura de datos privados
  - Backdoor: punto oculto de acceso a nuestra máquina
  - Rootkit: proporciona acceso remoto
  - Ransomware: captura o modificación de datos
  - Keylogger: captura de pulsaciones de teclas
- **Mecanismos de seguridad:**
  - **De prevención:**
    - mecanismos de autenticación e identificación.
    - mecanismos de control de acceso.
    - mecanismos de separación (física, temporal, lógica, criptográfica y fragmentación).
    - mecanismos de seguridad en las comunicaciones (cifrado de la información).
  - **De detección:**
    - IDS (Intrusion Detection System)
  - **De recuperación:**
    - copias de seguridad (backup).
    - mecanismos de análisis forense: averiguar alcance, las actividades del intruso en el sistema y cómo entró.

## 2. Mecanismos de seguridad:

### Cifrado

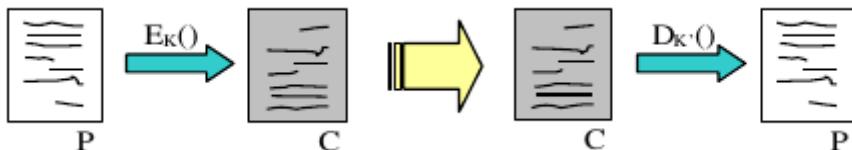
Se basa en la criptografía y en la definición de un criptosistema:

- Alfabeto de partida
- Espacio de claves
- Conjunto de transformaciones de cifrado
- Conjunto de transformaciones de descifrado



El cifrado es un procedimiento para garantizar la confidencialidad:

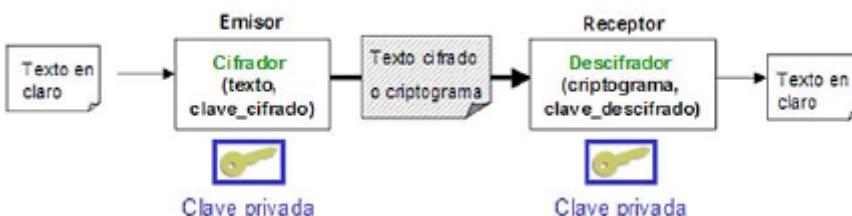
1. Se parte de un Texto llano/claro (plain text)
2. Se aplica un algoritmo de cifrado conocido como  $EK()$
3. Y un algoritmo de descifrado llamado  $DK'$  ()
4. Ambos dependen respectivamente de una clave de cifrado K y de una clave de descifrado K'.



El texto plano P se cifra y se convierte en C, se transmite y posteriormente se descifra C para obtener P de nuevo.

### Simétrico (de clave privada)

1. Emisor y receptor comparten la misma clave.
2. La clave sólo es conocida por ellos (privada/secreta).
3. Emisor encripta con ella y receptor desencripta con ella.
4. La clave deben compartirla por un canal seguro.



Algoritmos de clave secreta:

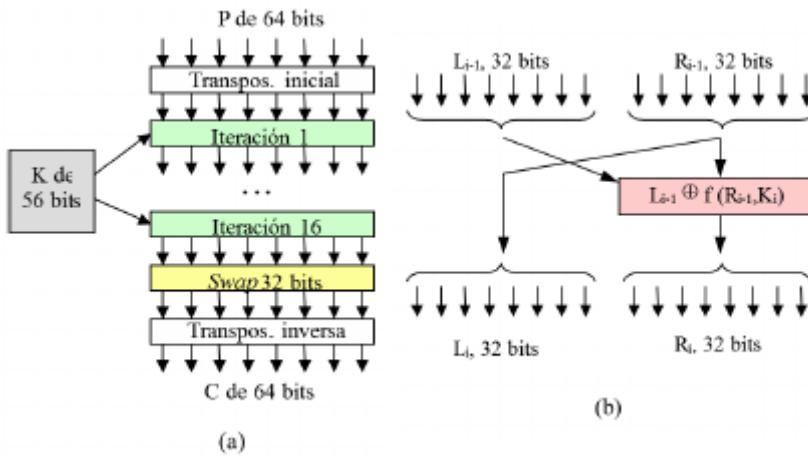
- **Algoritmo DES (Data Encryption Standard, IBM 1975):**

1. Se hace una transposición al bloque inicial de bits P
2. 16 iteraciones aplicando la clave K de 56 bits
  - 32 bits de la derecha pasan a ser los de la izquierda para la iteración siguiente
  - 32 bits de la derecha se obtienen haciendo XOR con los de la izquierda, junto con la aplicación de una función de transposición y duplicación de bits sobre R y K de la iteración actual, i.

En dicha función también se utilizan módulos de sustitución para cada grupo de 6 bits (8 grupos) y se obtienen 4 bits por cada bloque.

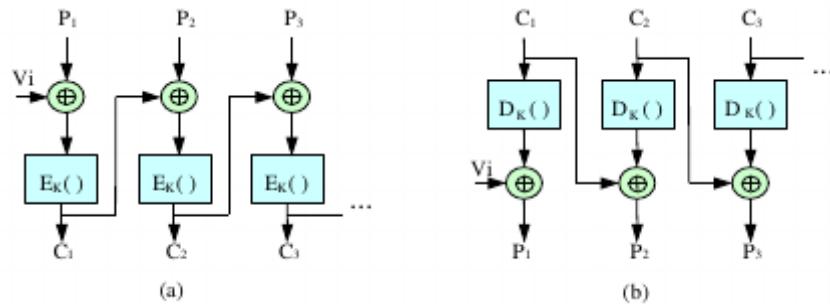
Por último se hace una nueva transposición del resultado.

3. Intercambio de 32 bits de orden más alto por los más bajos
4. Transposición inversa de 1



- **Encadenamiento DES:**

- Se realizan varios encriptamientos consecutivos y se combinan los resultados. -
- Con cada encriptamiento se aumenta en  $2^{56}$  la dificultad para descubrir la clave.



- **3DES:**

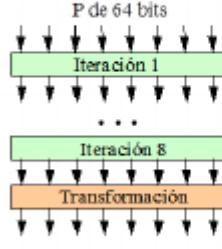
- Se hacen dos fases de encriptado y una de desencriptado entre ellas, usando cada vez una clave diferente.



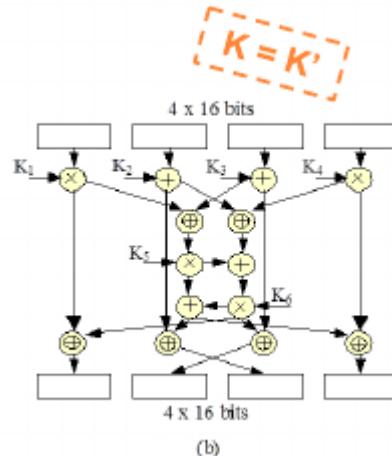
- **IDEA (International Data Encryption Algorithm):**

- Utiliza claves de 128 bits.
- Puede operar en tiempo real.
- Fácil de implementar en hardware.
- 8 iteraciones
- Aplica operaciones:
  - XOR
  - Suma módulo  $2^{16}$
  - Multiplicaciones módulo  $2^{16}+1$

Algorithm):



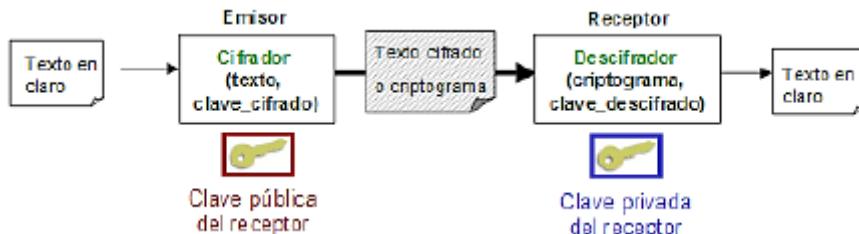
(a)



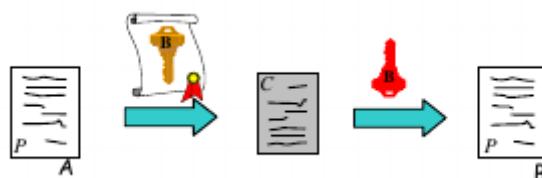
(b)

### Asimétrico (de clave pública)

1. El receptor tiene una clave pública y una clave privada (de la que deriva la pública).
2. Envía la clave pública a los emisores potenciales (por cualquier medio).
3. Emisor encripta con la clave pública del receptor.
4. Receptor desencripta con su clave privada.



- Dos claves por usuario (B): una pública  $KPUBB$  y otra privada  $KPRI_B$  distintas
- Conocida  $KPUBB$  es imposible conocer  $KPRI_B$
- Claves diferentes para cifrar y descifrar (P texto plano):
  - Cifrar:  $C = EKpubB(P)$
  - Descifrar:  $P = DKpriB(C)$
  - Autenticar:  $C = EKprivA(P)$



Algoritmos de clave pública:

- RSA (Rivest, Shamir, Adleman)
  - Elegimos p y q primos grandes ( $> 10^{100}$ )
  - $n = (p \cdot q)$  y  $z = (p-1) \cdot (q-1)$  (función de Euler)
  - Elegimos d coprimo con z (no tienen factores primos en común)
  - Calculamos e tal que  $e \cdot d \bmod z = 1$  (algoritmo de Euclides)
  - $Kpub = (e, n)$  y  $Kpri = (d, n)$ , de modo que:
    - $C = P^e \bmod n$
    - $P = C^d \bmod n$

## ALGORITMOS DE CLAVE PÚBLICA

- EJEMPLO RSA

$p = 3, q = 11$   
 $n = p \cdot q = 33, z = (p-1)(q-1) = 20 (= 5 \cdot 2 \cdot 2 \text{ en factores primos})$   
 $d = 7, \text{ coprimo respecto a } z$   
 $e = 3, e \cdot d \text{ mod } z = 1$   
 $K_{pub} = (3, 33) \text{ y } K_{pri} = (7, 33)$

$$\begin{cases} C = P^e \text{ mod } n \\ P = C^d \text{ mod } n \end{cases}$$

Simbólico	Numérico	$P^3$	$P^3 \text{ mod } 33$	$C^7$	$C^7 \text{ mod } 33$	Simbólico
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

## Autenticación

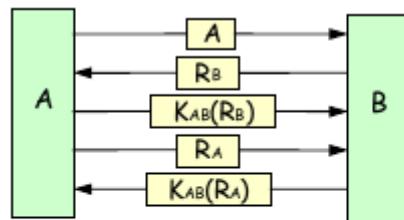
Autenticación y cifrado de la clave secreta:

- Esquema reto-respuesta (criptográfica):

- A desea autenticarse en B
- B le plantea un "reto" a A
- A responde al reto encriptándolo con la clave privada/secreta compartida entre A y B
- B comprueba si la respuesta es correcta y si lo es A se autentica
- El proceso se puede repetir para autenticar a B.

- Variante no criptográfica:

- La respuesta es la contraseña → ataque replay
- Contraseña con identificador → ataque replay con id
- Contraseña de un solo uso

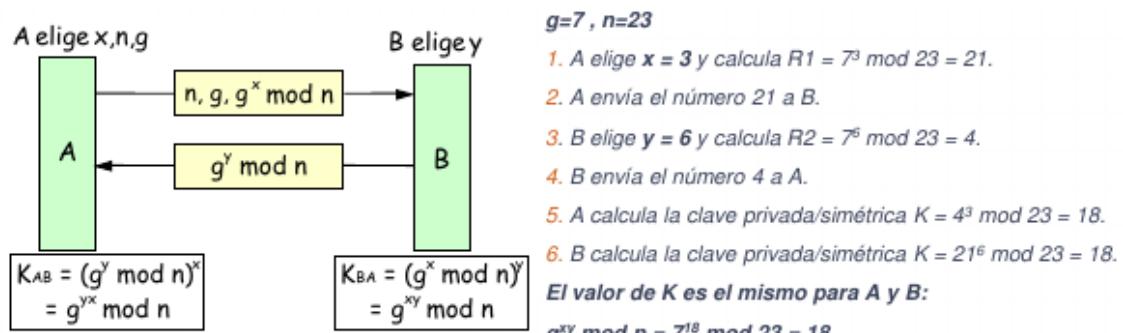


Mensaje nonce  
(sólo se genera una vez)

## Clave secreta

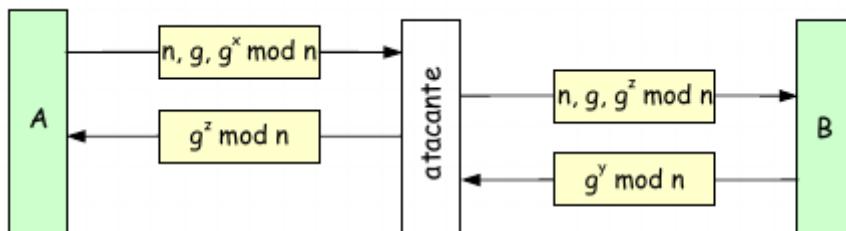
Establecimiento de la clave secreta:

- Intercambio de Diffie-Hellman: permite establecer una clave secreta entre dos entidades a través de un canal no seguro.



Usando números grandes no es vulnerable a escucha del canal

Fundamentos de Redes – Tema 4 26



Vulnerable a ataque MitM

## Funciones Hash

### FUNCIONES COMPENDIO (RESUMEN O DIGEST)

- Funciones unidireccionales (irreversibles) de cálculo sencillo.
- Texto de entrada (M) de longitud variable.
- $M \rightarrow H(M)$  siendo  $H(M)$  de longitud fija (256 ó 512 bits)
- Imposible obtener M a partir de su resumen H(M).
  - Invulnerables a ataques de colisión, dado M es imposible encontrar  $M'$  ya que  $M \neq M'$  y  $H(M) = H(M')$
- Las funciones Hash se pueden usar para garantizar integridad + autenticación (clave K): Hash Message Authentication Code (HMAC):  $M + H(K | M)$  pero para evitar ataques de extensión se usa  $M + H(K | H(K | M))$ 
  - Integridad: datos no alterados
  - Autenticación: generar hash correcto

Funciones compendio/resumen/digest:

- **MD5 (Message Digest 5, RFC 1321):**
  - Relleno bits "100..0" por la derecha, de longitud máxima 448 bits
  - Adición de campo de longitud de 64 bits
  - División del mensaje en bloques de 512 bits
  - Procesamiento secuencial por bloques.
  - De cada bloque se obtiene un digest de 128 bits. Cada bloque se procesa:
    - Se usan varias funciones (F, G, H, I) de operadores binarios (XOR, AND, OR, NOT) combinadas.
    - Se aplican los valores de los registros A, B, C, D (son registros ctes de 32b con valores hexadecimales)
    - Se hacen desplazamientos de bits.

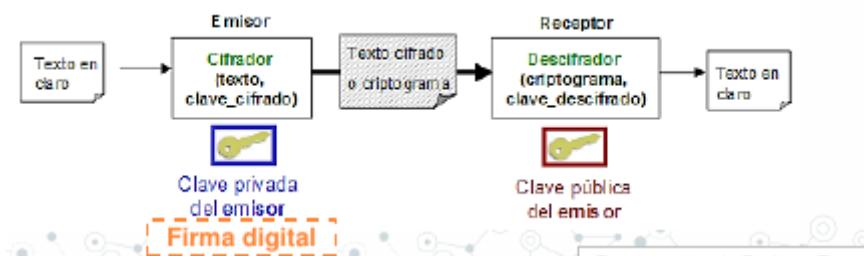
- Se hacen varias pasadas.
  - Se hace una suma final módulo 232.
  - La salida de un bloque será la entrada del siguiente.
- **SHA-1 (Secure Hash Algorithm 1, RFC 3174):**
    - Relleno bits “100..0” por la derecha, de longitud máxima 448 bits • Adición de campo de longitud de 64 bits
    - División del mensaje en bloques de 512 bits
    - Procesamiento secuencial por bloques.
    - De cada bloque se obtiene un digest de 160 bits. Cada bloque se procesa:
      - Se divide el bloque en palabras de 32 bits.
      - Se extienden las palabras combinándolas hasta tener 80.
      - Se agrupan de 20 en 20 y se combinan usando funciones.
      - Se usan varias funciones de operadores binarios (XOR, AND, OR, NOT) combinadas entre sí.
      - Se aplican los valores de los registros A, B, C, D, E.
      - Se hacen 4 pasadas de este proceso.
      - Se hace una suma final módulo 232.
      - La salida de un bloque será la entrada del siguiente.

## Firma digital

Una firma digital es un conjunto de datos que, consignados junto a otros o asociados con ellos, pueden ser utilizados como medio de identificación del firmante.

Objetivos:

- Que el receptor pueda autenticar al emisor.
- Que no haya repudio (que el emisor no pueda alegar que él no envió el mensaje).
- Que el emisor tenga garantías de no falsificación de su mensaje (integridad).



- **Firma digital big brother:**

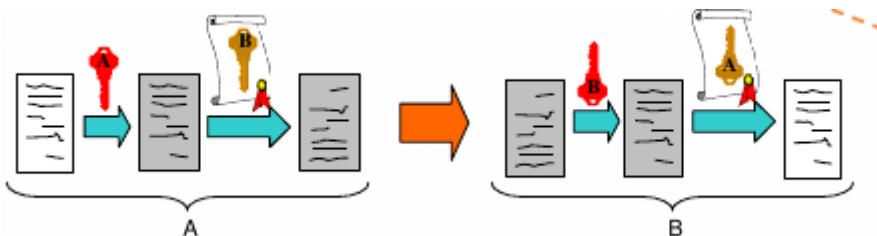
- Entidad central (BB) que interviene en el proceso de firma digital para la transmisión de un mensaje P entre A y B.
- A envía el mensaje cifrado con una clave que comparte con BB, KA , incluyendo además el propio destino del mensaje, B, y una marca de tiempo t.
- BB envía a B el mensaje cifrado con la clave que comparte con él, KB , la identidad de A, el mensaje P, su propia marca de tiempo t y su firma digital. La firma serán estos mismos valores encriptados con su propia clave KBB.



- **Firma digital con clave asimétrica y doble cifrado:**

- Un cifrado para autenticación, con KpriA
- Otro, para proporcionar privacidad, con KpubB
- Para firmar, enviar KpubB(KpriA(P))
- En el receptor se desencripta: KpubA(KpriB(KpubB(KpriA(P))))=P

¿El problema? garantizar el no repudio, una asociación fehaciente e indisoluble de A con su clave pública.



## Certificados digitales

Un certificado digital sirve para garantizar la asociación identidad-clave. Para que un usuario no pueda corromper una clave pública (de otro) y decir que es suya.

Autoridades de certificación:

- Entidad para garantizar la asociación entre identidad y claves.
- Funcionamiento:
  - El usuario obtiene sus claves pública y privada
  - Éste envía una solicitud, firmada digitalmente, a la AC indicando su identidad y su clave pública
  - AC comprueba la firma y emite el certificado solicitado:
    - Identidad de AC, identidad del usuario, clave pública del usuario y otros datos como, por ejemplo, el período de validez del certificado.
    - Todo ello se firma digitalmente con la clave privada de AC con objeto de que el certificado no pueda falsificarse .

**Las AC son responsables de:**

- emitir los certificados
- asignarles una fecha de validez
- revocarlos antes de esta fecha (en casos determinados)

**AC reconocidas:**

- ACE ([www.ace.es](http://www.ace.es))
- VeriSign ([www.verisign.com](http://www.verisign.com))
- CAMERFIRMA ([www.camerfirma.es](http://www.camerfirma.es))
- CERES ([www.cert.fnmt.es](http://www.cert.fnmt.es))

No es lo mismo firma digital (un uso en una transmisión) que certificado digital (muchos usos, acredita nuestra identidad)

**Tipos de certificados:**

- Certificados firmados localmente:
  - Firmados por un servidor local.
  - De uso interno en una red privada (intranet).
  - Para garantizar los intercambios confidenciales y para autenticar usuarios.
- Certificados firmados por una autoridad de certificación:

- Válidos en todo Internet.
- Para garantizar los intercambios seguros con usuarios anónimos.
- Para acreditar la identidad de un usuario.

#### **Formato certificado X.509:**

<i>Field</i>	<i>Explanation</i>
Version	Version number of X.509
Serial number	The unique identifier used by the CA
Signature	The certificate signature
Issuer	The name of the CA defined by X.509
Validity period	Start and end period that certificate is valid
Subject name	The entity whose public key is being certified
Public key	The subject public key and the algorithms that use it

## **Resumen**

Relación entre los mecanismos de seguridad y los servicios de seguridad:

- Confidencialidad: Se consigue mediante Cifrado (simétrico o asimétrico).
- Autenticación: Se consigue con los mecanismos de autenticación (reto-respuesta), y Firma digital (big brother, doble cifrado: cifrado en el emisor con clave privada y descifrado en receptor con clave pública).
- No repudio o irrenunciabilidad: Se consigue mediante firma digital (big brother, doble cifrado), certificado digital.
- Integridad: Se consigue añadiendo resúmenes generados con funciones hash/digest.
- Disponibilidad: Los mecanismos no proporcionan disponibilidad por sí mismos. Serían necesarios sistemas antiataque, redundancia en las líneas de acceso, en los servidores, etc.

## **3. Implementación de mecanismos de seguridad**

- Seguridad perimetral
  - Firewalls, IDS (sistemas detección intrusiones), IRS (sistemas respuesta intrusiones)
- Protocolos de seguridad:
  - Capa de Aplicación
    - Pretty Good Privacy (PGP)
    - Secure Shell (SSH)
  - Capa de Transporte
    - Secure Socket Layer (SSL) → HTTPS, IMAPS, SSL-POP
    - Transport Layer Security (TLS)
  - Capa de Red → IPSec (VPN)
  - Capas inferiores → PAP, CHAP, MS\_CHAP, EAP...

## **Firewall**

Es una combinación de técnicas, políticas de seguridad y tecnologías (hardware y software). Proporciona seguridad en la red, controlando el tráfico que entra y sale (normalmente entre una red privada e Internet). Debe combinarse con protocolos seguros (seguridad dentro de la red).

Funciones:

- Controlar (permitiendo o denegando) los accesos desde la red local hacia el exterior y los accesos desde el exterior hacia la red local.
- Filtrar los paquetes que circulan, de modo que sólo los servicios permitidos puedan pasar.
- Monitorizar el tráfico, supervisando destino, origen y cantidad de información recibida y/o enviada.
- Almacenar total o parcialmente los paquetes que circulan a través de él para analizarlos en caso de problemas.
- Establecer un punto de cifrado de la información si se pretende comunicar dos redes locales a través de Internet.

Técnicas aplicadas:

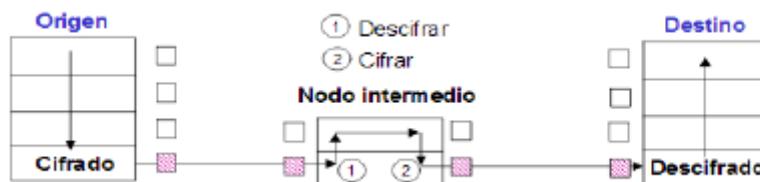
- Filtrado de paquetes:
  - Reglas que especifican qué tipos de paquetes pueden circular en cada sentido y cuáles se bloquearán.
  - Las reglas se basan en las cabeceras de los paquetes.
- Servicios de proxy:
  - Son aplicaciones especializadas que funcionan en un cortafuegos.
  - Hacen de intermediarios entre los servidores y los clientes reales.
  - Reciben las peticiones de servicios de los usuarios, las analizan y en su caso modifican, y las transmiten a los servidores reales .
  - Son transparentes al usuario.

## Cifrado en redes

### Cifrado de enlace

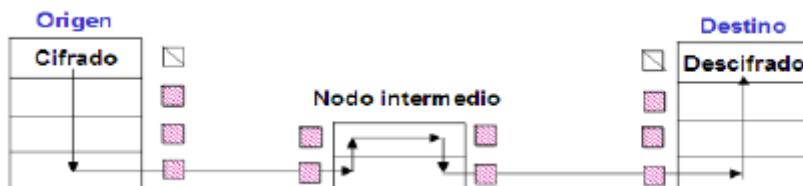
Capa 2 de OSI - Cifra todo el mensaje, incluidas las cabeceras de niveles superiores. Requiere nodos intermedios con capacidades de cifrado/descifrado.

La información está protegida entre cada par de nodos consecutivos (distintas claves para cada par) - Es necesario descifrarla, aunque sea parcialmente, para procesos de encaminamiento, control de errores, etc.



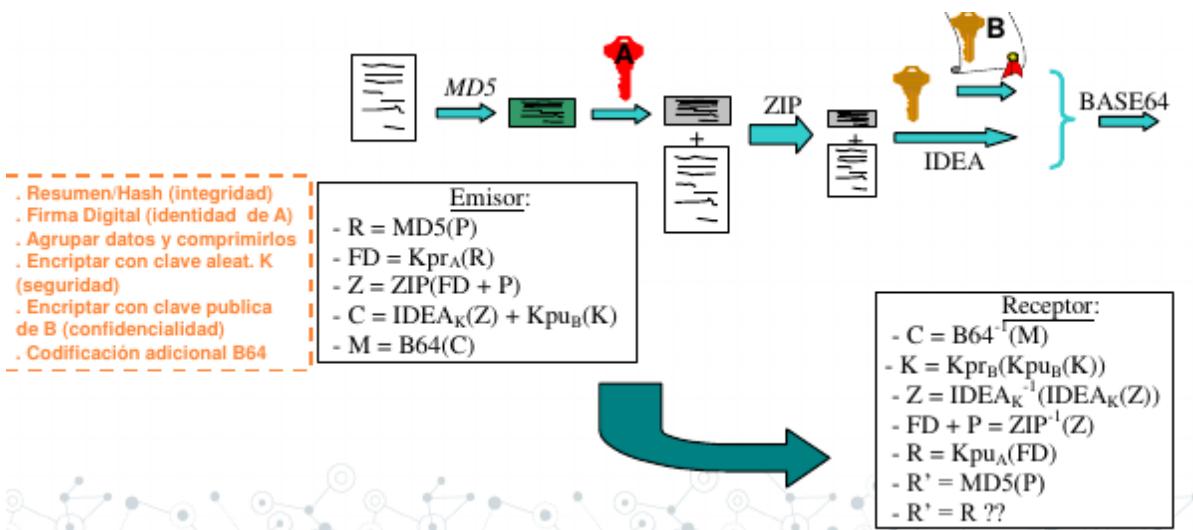
### Cifrado extremo a extremo

Capa 7 de OSI - Sólo se cifran los datos, las cabeceras se añaden y se transmiten sin cifrar.



## PGP (Pretty Good Privacy)

Usado para correo electrónico seguro y otros documentos en Internet.



## SSH (Secure SHell)

SSH es un protocolo de nivel de aplicación para crear conexiones seguras entre dos sistemas sobre redes no seguras. Alternativa a programas de acceso remoto no seguros, como telnet, ftp, rlogin, rsh y rcp (slogin, ssh y scp).

Proporciona un terminal de sesión cifrada con autenticación fuerte del servidor y el cliente, usando criptografía de clave pública. Incluye características como:

- Variedad de mecanismos de autenticación de usuarios (incluyendo autenticación externa Kerberos).
- Conexiones TCP arbitrarias de tunneling a través de la sesión SSH, protegiendo protocolos inseguros como IMAP y permitiendo el paso seguro a través de cortafuegos.
- Transferencias seguras de ficheros.
- Soporte para entorno gráfico.

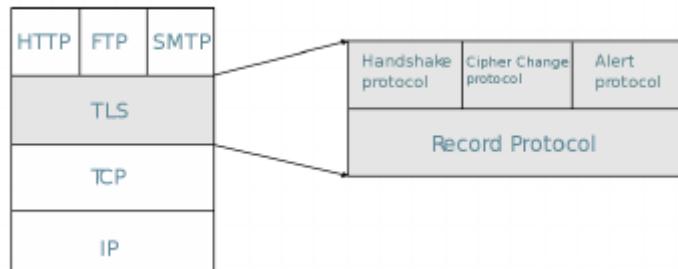
Secuencia de eventos de una conexión SSH

1. Se crea una capa de transporte segura para que el cliente sepa que está efectivamente comunicándose con el servidor correcto. Luego se cifra la comunicación entre el cliente y el servidor por medio de una clave simétrica/privada.
2. Una vez conectado de forma segura, el cliente se autentica ante el servidor sin preocuparse de que la información de autenticación pudiese exponerse.
3. Con el cliente autenticado ante el servidor, se pueden usar varios servicios diferentes con seguridad a través de la conexión, como una sesión de terminal interactivo, aplicaciones y túneles TCP/IP.

## SSL/TLS (Transport Layer Security)

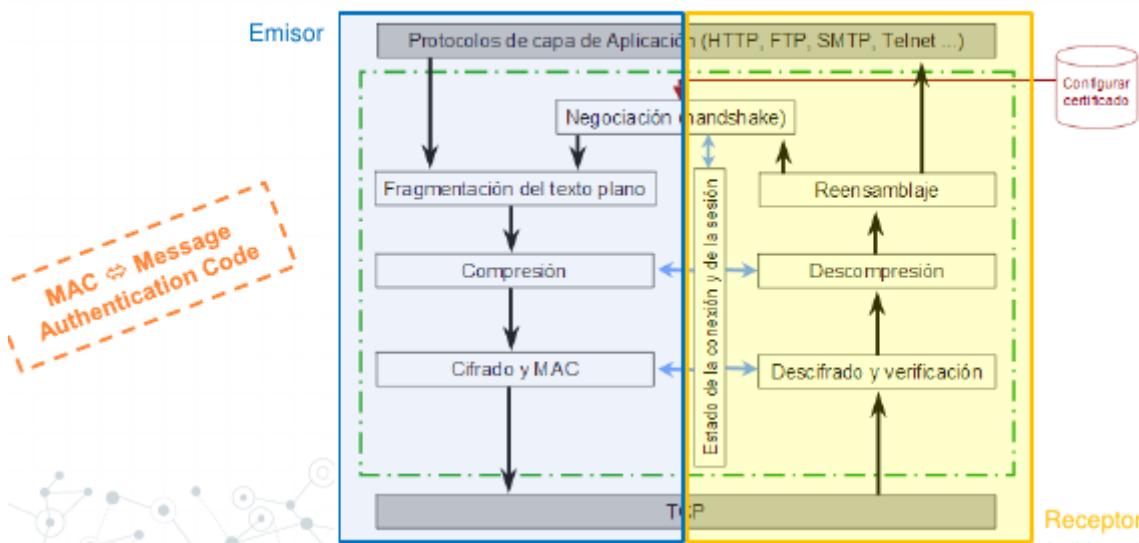
- SSL (Secure Socket Layer) → Desarrollado por Netscape en 1994 y puesto en dominio público para la definición de canales seguros sobre TCP.
  - SSL Record Protocol encapsula los protocolos y ofrece un canal seguro con privacidad, autenticación e integridad
  - SSL Handshake Protocol
    - Negocia el algoritmo de cifrado
    - Negocia la función Hash

- Autentica al servidor con X.509
- El cliente genera claves de sesión:
  - Aleatorias cifradas con KPUB\_SERVER ó Diffie-Hellman
- SSL Alert protocol
  - Informa sobre errores en la sesión
- Cipher Change Espec Protocol
  - Para notificar cambios en el cifrado



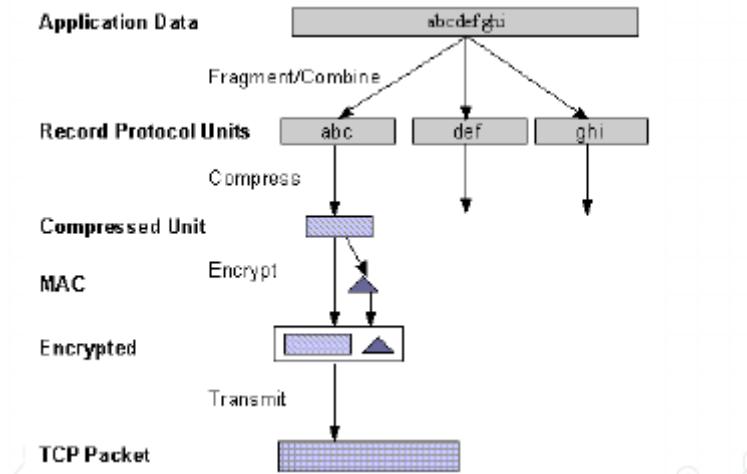
- TLS (Transport Layer Security) → Sucesor y mejora sobre SSL.
  - Corrige vulnerabilidades de SSL y permite la autenticación de emisor y receptor.
  - Se basa en el uso de certificados digitales para establecer la conexión.
  - Posteriormente emisor y receptor comparten una clave privada.
- Ambos son protocolos criptográficos que permiten realizar comunicaciones seguras sobre una red no segura.
- No funciona sobre UDP

## TRANSPORT LAYER SECURITY (SSL/TLS) - Arquitectura



1. El cliente al hacer la conexión informa sobre los sistemas criptográficos que tiene disponibles, y el servidor responde con un identificador de la conexión, su clave certificada e información sobre los sistemas criptográficos que soporta.
2. El cliente elegirá un sistema criptográfico y verificará la clave pública del servidor.
3. Entonces se generará una clave privada (de uso único) cifrada con la clave pública del servidor.
  - Si alguien pudiese descifrar la información, sólo conseguiría romper esa conexión/sesión, ya que una sesión posterior requeriría una clave privada diferente.
4. Una vez finalizado este proceso, los protocolos toman el control de nivel de aplicación, de modo que SSL/TLS nos asegura que:
  - Los mensajes que enviamos o recibimos no han sido modificados (integridad).

- Ninguna persona sin autorización puede leer la información transmitida (confidencialidad).
- Efectivamente envía/recibe la información quien debe enviarla/recibirla (autenticación).



Versión actual SSL 3.0. SSL es capaz de trabajar de forma transparente con todos los protocolos que trabajan sobre TCP. Para ello, el IANA tiene asignado un número de puerto por defecto a cada uno de ellos:

Identificador de protocolo	Puerto TCP	Descripción
https	443	HTTP sobre SSL
smtps	465	SMTP sobre SSL
nttps	563	NTTP sobre SSL
ladps	648	LDAP sobre SSL
telnets	992	TELNET sobre SSL
imaps	993	IMAP sobre SSL
ircs	994	IRC sobre SSL
pop3s	995	POP3 sobre SSL
fps-data	989	FTP-Datos sobre SSL
fps-control	990	FTP-Control sobre SSL

## IPSec (IP Security)

Proporciona seguridad en la capa de red y a las superiores que se apoyen en IP (RFC 2401). Su objetivo es garantizar autenticación, integridad y (opcionalmente) privacidad a nivel IP.

### IPSec consiste de 3 procedimientos:

1. Establecimiento de una "Asociación de seguridad": IKE (Internet Key Exchange, RFC 2409) -  
Objetivo: establecimiento de clave secreta (Diffie-Hellman).
  - Incluye previamente autenticación (con certificados) para evitar el ataque de MitM.
  - Es simplex: la asociación de seguridad tiene un único sentido.
  - Se identifica con la IP origen + Security Parameter Index (32 bits).
  - Vulnera el carácter NO orientado a conexión de IP.
2. Garantizar la autenticación e integridad de los datos: protocolo de "Cabeceras de autenticación" (RFC 2401)
3. (Opcional) Garantizar la autenticación e integridad y privacidad de los datos: protocolo de "Encapsulado de seguridad de la carga" (RFC 2411)

### IPSec tiene 2 modos de operación:

1. **Modo Transporte:** la asociación se hace extremo a extremo entre en host origen y host destino.

- Se protege la carga útil IP (payload) (capa de transporte)
- Comunicación segura extremo a extremo
- Requiere implementación de IPSec en ambos hosts

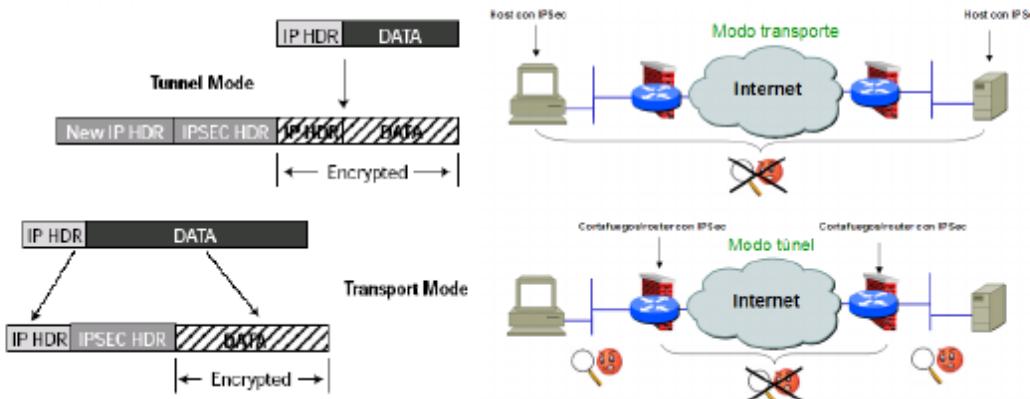
**2. Modo Túnel:** la asociación se hace entre dos routers intermediarios.

- Se protegen paquetes IP (capa de red)
- Para la comunicación segura entre routers/gateways de seguridad sólo se puede usar este modo
- Permite incorporar IPSec sin afectar a los hosts
- Se integra fácilmente con VPNs

### IPSec

➤ IPSec tiene 2 modos de operación:

- 1) **Modo Transporte:** la asociación se hace extremo a extremo entre en host origen y host destino
- 2) **Modo Túnel:** la asociación se hace entre dos routers intermediarios



## TEMA 5: CAPA DE APLICACIÓN

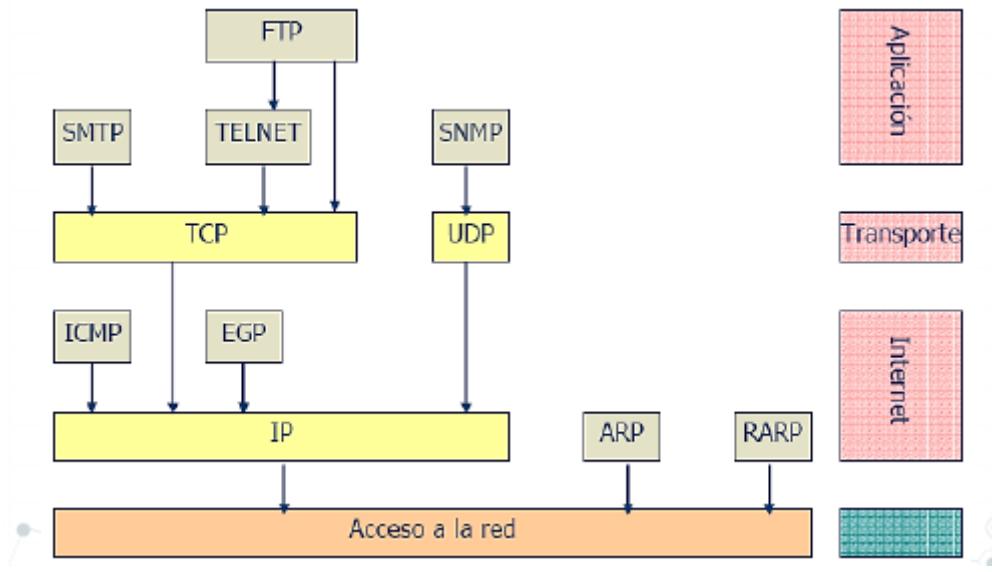
### 1. Introducción a las aplicaciones de red

#### Protocolos TCP/IP

El origen de esta familia de protocolos fue la red ARPANET (en ella se desarrollaron los conceptos fundamentales de diseño y gestión de redes), para la que se definió el precursor de TCP/IP: NCP (Network Control Program).

Los niveles más bajos (enlace y físico) no están implementados ya que TCP/IP se diseñó para no depender de una red física concreta:

- Los protocolos ARP (Address Resolution Protocol) y RARP (Reverse Address Resolution Protocol) se encargan de enlazar los sistemas de direccionamiento IP y el de la red física utilizada.



## Capa de red

La base de la familia de protocolos es el nivel de Red (IP, Internet Protocol).

Es un protocolo de comutación de paquetes muy sencillo, de tipo datagrama, de forma que se pueda implementar en cualquier tipo de máquina. Existen actualmente dos versiones IPv4, IPv6.

Protocolos de apoyo:

- ICMP (Internet Control Message Protocol): comunicación de mensajes entre nodos de la red
- IGMP (Internet Group Management Protocol): envío de mensajes a grupos de usuarios

## Capa de transporte

Implementa dos protocolos extremo a extremo (entre nodo origen y nodo destino).

- TCP (Transmission Control Protocol):
  - Es un protocolo orientado a la conexión.
  - Con control de errores.
  - Se encarga también del control de flujo.
  - Fragmentado y reensamblado de segmentos (garantiza el secuenciamiento).
- UDP (User Datagram Protocol):
  - Es un protocolo no orientado a conexión (datagrama).
  - No realiza control de errores.
  - No garantiza el secuenciamiento de la información.
  - Es muy rápido.
    - Útil para peticiones aisladas, o transmisión de audio/vídeo

## Capa de aplicación

- Protocolos basados en ICMP: - PING: solicitud de eco (comprobación de conectividad)
- Protocolos basados en TCP:
  - TELNET: terminal remoto
  - FTP(File Transfer Protocol): transmisión de ficheros
  - SMTP(Simple Mail Transfer Protocol): correo electrónico
  - HTTP(HyperText Transfer Protocol): páginas web
  - RPC (Remote Procedure Call): ejecución de procesos remotos
- Protocolos basados en UDP:

- SNMP(Simple Network Management Protocol): gestión de red
- BOOTP: arranque remoto
- DNS(Domain Name System)
- NFS (Network File System): gestión de ficheros en red

## Arquitectura cliente-servidor

La arquitectura (o modelo) cliente/servidor es una forma específica de diseño de aplicaciones, aunque también se conoce con este nombre a las computadoras en las que estas aplicaciones son ejecutadas.

- El **cliente** es la computadora que se encarga de efectuar una petición o solicitar un servicio y recibir una respuesta. El cliente no posee control sobre los recursos.
  - Funcionando intermitentemente
  - Pueden tener IP dinámica y privada
  - Se comunican con el servidor
  - No se comunican entre sí en relación a un servicio
- El **servidor** es una computadora (remota normalmente) que evalúa la petición del cliente y la acepta o la rechaza. Si es aceptada ejecuta el servicio y transmite la información resultante al cliente que efectuó la petición.
  - Siempre en funcionamiento
  - IP permanente y pública
  - Agrupados en granjas
  - Pueden comunicarse entre sí para optimizar el servicio

Cliente y servidor pueden residir en la misma máquina.



- **Ventajas:**

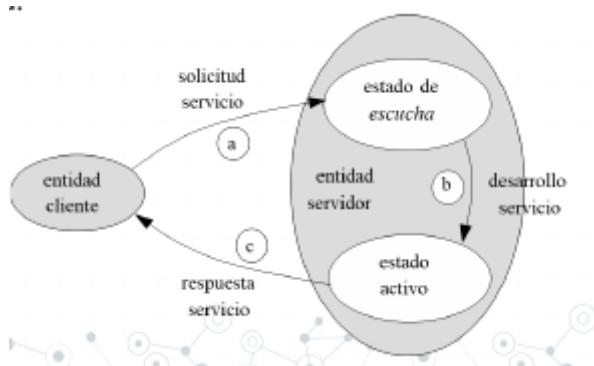
- Recursos centralizados: debido a que el servidor suele ser el centro de la red, puede administrar los recursos que son comunes a todos los usuarios (clientes).
- Seguridad mejorada: la cantidad de puntos de entrada que permite el acceso a los datos no es importante. El servidor garantizará la seguridad en dicho acceso.
- Administración al nivel del servidor: ya que los clientes no juegan un papel importante en este modelo, requieren menos administración.
- Red escalable: es posible quitar o agregar clientes (hasta un límite) sin que afecte demasiado al funcionamiento de la red y sin la necesidad de realizar mayores modificaciones.

- **Desventajas:**

- Costo elevado: debido a la complejidad técnica del servidor y a su gestión, seguridad y mantenimiento.
- Servidor es el eslabón débil: debido a que toda la red del servicio está construida en torno a él. Afortunadamente, el servidor suele ser altamente tolerante a los fallos (replicación, discos espejo, copia de seguridad, virtualización).

La mayoría de las transacciones entre aplicaciones se basan en el paradigma cliente- servidor:

- Servidor: programa que ofrece un servicio accesible a través de la red.
  - Normalmente el servidor usa puertos bien conocidos (reservados)
- Cliente: programa que envía peticiones y espera respuestas del servidor a través de la red.



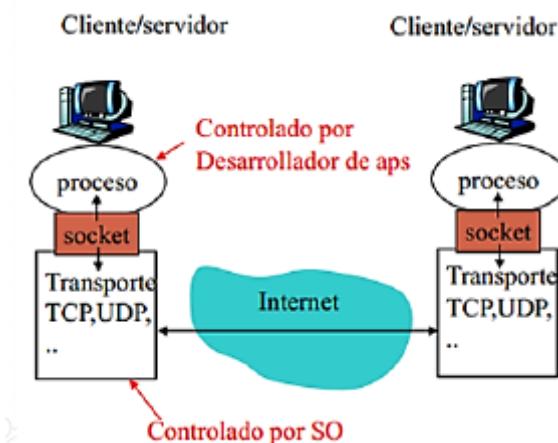
### Tipos de servidores:

- Servidor Iterativo:
  - Sólo puede atender a un cliente a la vez.
  - Se encolarán los clientes que realicen peticiones mientras el servidor está dando servicio a otro cliente.
- Servidor Concurrente:
  - Puede dar servicio a varios clientes a la vez.
  - Ejecuta un proceso (o hebra) por cada cliente, para procesar concurrentemente las peticiones (concurrentia real o simulada)

### Procesos cliente y servidor:

- Proceso Cliente: proceso que inicia la comunicación
- Proceso Servidor: proceso que espera ser contactado

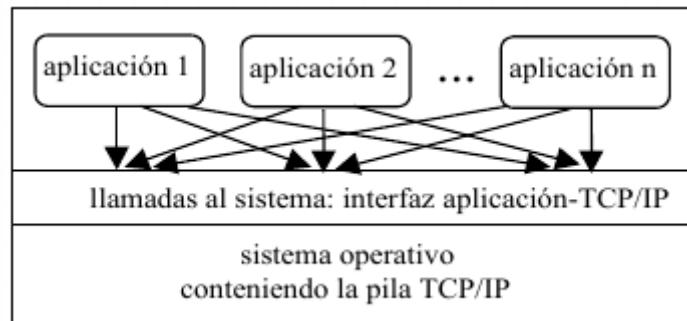
Un proceso envía/recibe mensajes a/desde un socket. Cada proceso debe tener un identificador compuesto por su dirección IP + número de puerto (ej: servidor web gaia.cs.umass.edu, Dirección IP: 128.119.245.12, Núm. Puerto: 80)



## Interfaz socket

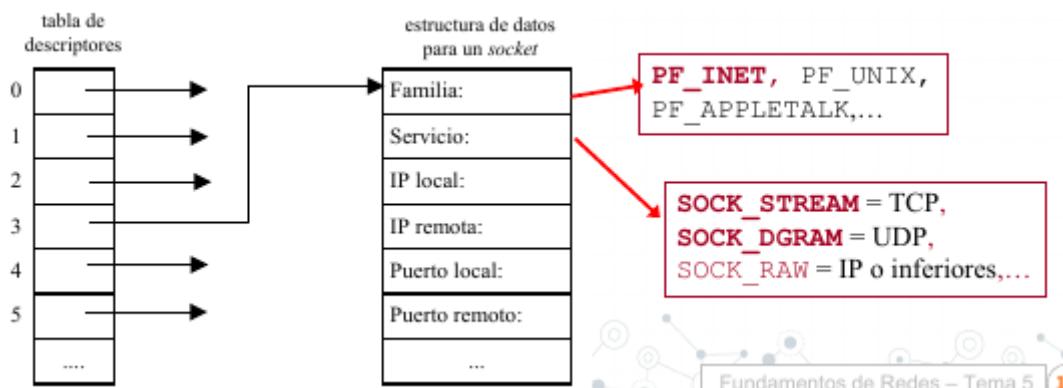
El software TCP/IP es parte del S.O. Las aplicaciones lo usan a través de una API consistente en llamadas al sistema. Es la llamada "Interfaz socket".

- La definición de la interfaz socket no es parte de ningún protocolo.
- Distintas implementaciones: Berkeley Socket Distribution, Winsock, Transport Layer Interface, etc.



Definimos socket como un descriptor de una transmisión a través del cual la aplicación puede enviar y/o recibir información hacia y/o desde otro proceso de aplicación.

Es una "puerta" de acceso entre la aplicación y los servicios de transporte. En la práctica un socket es un puntero a una estructura del tipo:



**Propiedades** (Según el tipo de socket):

- La fiabilidad de la transmisión: ningún dato transmitido se pierde.
- La conservación del orden de los datos: los datos llegan en el orden en el que han sido emitidos.
- La no duplicación de datos: sólo llega a destino un ejemplar de cada dato emitido.
- La comunicación en modo conectado: se establece una conexión entre dos puntos antes del principio de la comunicación (es decir, se establece un circuito virtual). A partir de entonces, una emisión desde un extremo está implícitamente destinada al otro extremo conectado.
- Delimitación de los mensajes: los límites de los mensajes emitidos se pueden encontrar en el destino.
- El envío de mensajes (urgentes): posibilidad de enviar datos fuera del flujo normal, accesibles inmediatamente.

**Tipos de sockets:**

- **SOCK\_STREAM:** Los sockets de este tipo permiten comunicaciones fiables en modo conectado (propiedades 1, 2, 3 y 4) y eventualmente autorizan, según el protocolo aplicado los mensajes fuera de flujo (propiedad 6). [SOCKET TCP]
- **SOCK\_DGRAM:** Corresponde a los sockets destinados a la comunicación en modo no conectado para el envío de datagramas de tamaño limitado. Los datagramas no trabajan con

conexiones permanentes (protocolo UDP). La transmisión por los datagramas se hace a nivel de paquetes, donde cada paquete puede seguir una ruta distinta, no garantizándose una recepción secuencial de la información. [SOCKET UDP]

- SOCK\_RAW: Permite el acceso a los protocolos de más bajo nivel (por ejemplo, el protocolo IP en el dominio Internet). Su uso está reservado al superusuario.
- SOCK\_SEQPACKET: Corresponde a comunicaciones que poseen las propiedades 1, 2, 3, 4 y 5.

## Protocolos

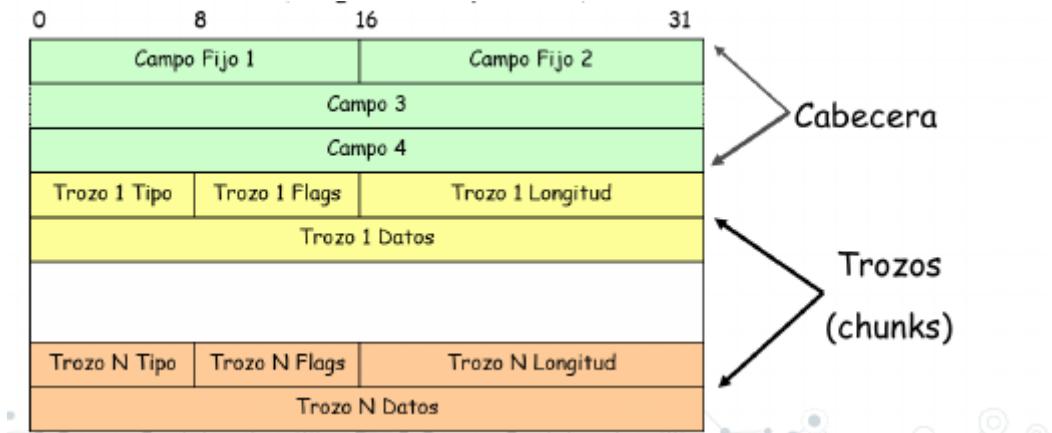
### ¿Qué define un protocolo?

- El tipo de servicio
  - Orientado o no orientado a conexión
  - Confirmado o no
- El tipo de mensaje
  - Request (solicitud), Response (respuesta), etc
- La sintaxis
  - Definición y estructura de campos en el mensaje
  - Hay protocolos orientados a texto (HTTP)
  - Y otros en binario (DNS)
  - Tendencia para otras capas: usar formato Type-Length-Value
- La semántica
  - Significado de los campos
- Las reglas
  - Cuándo los procesos envían mensajes/responden a mensajes

### Tipos de protocolos:

- Protocolos de dominio público (Definidos en RFCs (ej., HTTP, SMTP)) vs. propietarios (Ej: Skype, IGRP)
- Protocolos in-band vs. out-of-band
  - In-band: protocolos de red con la que se regula el control de datos.
  - Out-of-band: (urgent data en TPC) útil para separación de dos tipos diferentes de datos, no afecta a la velocidad o la prioridad.
- Protocolos stateless vs. stateful
  - stateless: protocolo que trata cada petición como una transacción independiente que no tiene relación con cualquier solicitud anterior, la comunicación se compone de pares independientes de solicitud/ respuesta. -
  - stateful: un protocolo que requiere el mantenimiento del estado interno en el servidor.
- Protocolos persistentes vs. no persistentes: En una conexión persistente solo se hará una conexión TCP, mientras que en una conexión no persistente se utilizarán múltiples conexiones TCP, una por cada objeto solicitado.

Tendencia a protocolos flexibles, con una cabecera fija y uno o varios trozos obligatorios u opcionales



Los trozos pueden incluir una cabecera específica más una serie de datos en forma de parámetros:

- Parámetros fijos: en orden
- Parámetros de longitud variable u opcionales. -
- Para los parámetros se usa Formato TLV (Type-Length-Variable)

#### Propiedades:

- Pérdida de datos (errores)
  - Algunas aplicaciones (Ej: audio) pueden tolerar algunas pérdida de datos;
  - Otras (Ej: FTP, telnet) requieren transferencia 100% fiable
- Requisitos temporales
  - Algunas aplicaciones denominadas inelásticas (ej., telefonía Internet, juegos interactivos) requieren retardo acotado (delay) para ser efectivas
- Ancho de banda (tasa de transmisión o throughput)
  - Algunas aplicaciones requieren envío de datos a una tasa determinada
- Seguridad
  - Encriptación, autenticación, no repudio...

Application	Data loss	Throughput	Time Sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5kbps-1Mbps video:10kbps-5Mbps	yes, 100's ms
stored audio/video	loss-tolerant	same as above	yes, few s
interactive games	loss-tolerant	few kbps up	yes, 100's ms
instant messaging	no loss	elastic	yes and no

<b>Application</b>	<b>Application layer protocol</b>	<b>Underlying transport protocol</b>
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (eg Youtube), RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP, proprietary (e.g., Skype)	typically UDP

## 2. Servicio de Nombres de Dominio (DNS)

La comunicación en Internet precisa de direcciones IP, pero las direcciones IP son difíciles de memorizar o recordar: necesitamos asignar a dichas direcciones nombres significativos conocidos como nombres de dominio.

Los nombres de dominio, al igual que las direcciones IP deben identificar de forma única a una máquina (una interfaz) en Internet.

### Historia

En los inicios de internet

- Se utilizaba una única tabla centralizada de traducción de nombres a direcciones.
- En los años 70 la red ARPANET estaba formada por unos cientos de máquinas y un sólo archivo, HOSTS.TXT que contenía toda la información que se necesitaba sobre esas máquinas.
- El centro de información de red del Departamento de defensa americano disponía de la versión maestra de la tabla y otros sistemas realizaban una copia regularmente.

Con el crecimiento de internet, el método "explotó" por varios motivos:

- El tráfico de red y la carga para la máquina que contenía las tablas que hacían posible el mapeo era desbordante.
- La consistencia del archivo era muy difícil de mantener, cuando el HOST.TXT llegaba a una máquina muy lejana estaba ya obsoleto.
- No se podía garantizar la no duplicidad de nombres (mantener una administración central en una red Internacional era muy complicado).
- El método no era escalable.

Conclusión:

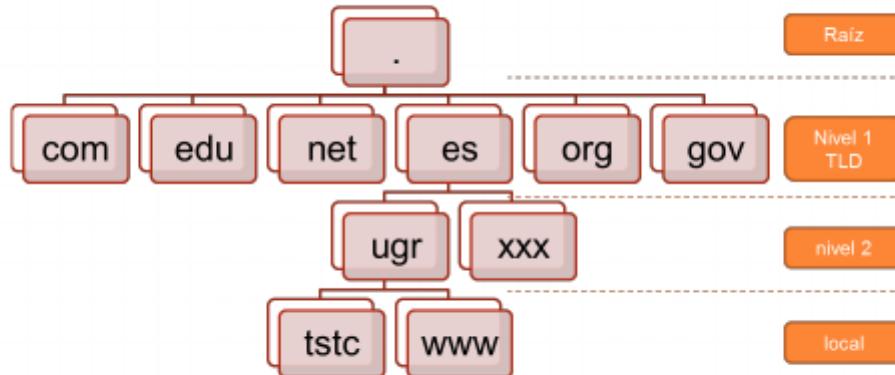
- Este enfoque quedó obsoleto debido a su ineficiencia para gestionar una gran cantidad de máquinas.
- Surgió un nuevo sistema de resolución de nombres, DNS (Domain Name System), para solventar los problemas anteriores.

### Sistema de nombres de dominio

- Sistema global: identifica únicamente en todo Internet.
  - ICANN (Internet Corporation for Assigned Names and Numbers). Es la autoridad central que controla la entrada de cualquier sistema nuevo. [www.icann.org](http://www.icann.org)

- NIC (Network Information Center). Organizaciones que permiten descentralizar esas tareas, gestionando parte de las numeraciones.
- Modelo jerárquico:
  - El dominio se divide en subdominios para facilitar su gestión por los NICs.
  - Se tiene una estructura en árbol.
  - Los dominios de la raíz se denominan Top Level Domain (TLD).
- Modelo lógico (no físico): hace falta una traducción.

## ESTRUCTURA JERÁRQUICA EN ÁRBOL



- Todos los dominios en Internet pueden representarse mediante un árbol.

- Las hojas del árbol serían los dominios que ya no contienen más subdominios.

Sintaxis nombre de dominio:

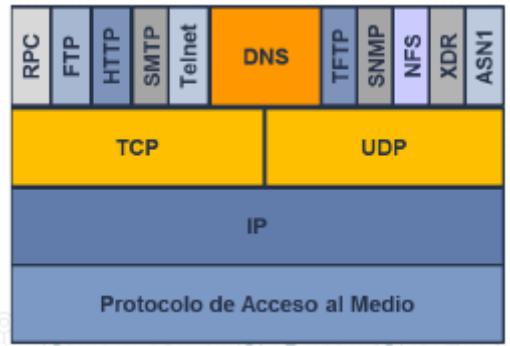
- Cadena de hasta 255 caracteres, formada por etiquetas separadas por puntos (long. etiqueta < 64 caracteres) que indican un nivel en la jerarquía.
- No se distinguen mayúsculas y minúsculas.
- Tipos de nombres de dominio:
  - Absolutos: terminados en "." (ugr.es.)
  - Relativos: no terminados en "."
- En el nivel raíz, los dominios se clasifican en:
  - Geográficos: división por países (o regiones).
  - Genéricos: en función del tipo de organización.

Inicialmente fueron definidos los siguientes 9 dominios genéricos (RFC 1591):

- .com → organizaciones comerciales
- .edu → instituciones educativas, como universidades, de EEUU.
- .gov → instituciones gubernamentales estadounidenses
- .mil → grupos militares de Estados Unidos
- .net → proveedores de Internet
- .org → organizaciones diversas diferentes de las anteriores
- .arpa → propósitos exclusivos de infraestructura de Internet
- .int → organizaciones establecidas por tratados internacionales entre gobiernos
- .'xy' → indicativos de la zona geográfica Ej: es (España); pt (Portugal)...

El servicio DNS es transversal (usado por otros servicios). Se sitúa en el esquema de capas de TCP/IP como protocolo de aplicación, tanto sobre UDP (paquetes de consultas pequeños) como TCP (paquetes de consultas grandes).

Puerto 53 de la capa de transporte.



## Servidores DNS

El servicio se basa en el uso de una base de datos descentralizada, distribuida entre diversos servidores (cada uno almacena una parte).

Cada servidor almacenará datos relativos a los dominios de los que es responsable. Ese grupo de dominios se conoce como zona.

El servidor que gestiona una zona se dice que tiene autoridad sobre ella. Hay dos tipos de servidores por zona:

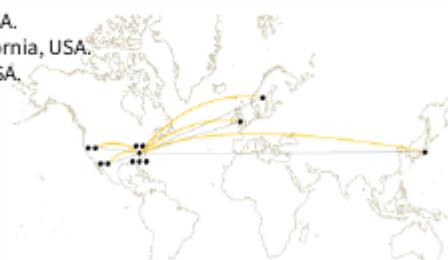
- Servidor Autoridad Primario (o master): mantiene una copia principal de la BD. Atiende las peticiones en primera instancia.
- Servidor Autoridad Secundario (o slave): almacena una copia de la BD que le transfiere el servidor primario cada cierto tiempo

Los servidores DNS también almacenan la información sobre los servidores a consultar en caso de que se les pregunte por un dominio sobre el que no tienen autoridad. Además, tienen una caché para almacenar las últimas peticiones resueltas en caso de que se les soliciten de nuevo.

Los **servidores raíz** contienen la información de localización de los servidores con autoridad sobre los TLDs. Son los primeros en ser consultados, por lo que deben estar bien dimensionados, ya que todas las peticiones DNS empiezan en ellos.

Existen 13 servidores raíz en el mundo, referenciados con las letras A-M. Aunque cada uno es un servidor distribuido en varias máquinas ubicadas en múltiples puntos geográficos.

- Servidor A: Network Solutions, Herndon, Virginia, USA.
- Servidor B: Instituto de Ciencias de la Información de la Universidad del Sur de California, USA.
- Servidor C: PSINet, Virginia, USA.
- Servidor D: Universidad de Maryland, USA.
- Servidor E: NASA, en Mountain View, California, USA.
- Servidor F: Internet Software Consortium, Palo Alto, California, USA.
- Servidor G: Agencia de Sistemas de Información de Defensa, California, USA.
- Servidor H: Laboratorio de Investigación del Ejército, Maryland, USA.
- Servidor I: NORDUnet, Estocolmo, Suecia.
- Servidor J: (TBD), Virginia, USA.
- Servidor K: RIPE-NCC, Londres, Inglaterra.
- Servidor L: (TBD), California, USA.
- Servidor M: Wide Project, Universidad de Tokio, Japón.



## Delegación de la autoridad

La organización que posee un nombre de dominio, es responsable del funcionamiento y mantenimiento de los servidores de nombres (zona de autoridad).

La solicitud de registro de un dominio se realiza a una autoridad competente, por ejemplo InterNIC (<http://www.internic.net/>) es una autoridad de registro.

Se puede solicitar un dominio a una empresa (Ej: [www.arsys.es](http://www.arsys.es)) y/o ISP. Cada país dispone de autoridades de registro.

- En una zona existe un administrador local que puede delegar en otros administradores.
  - Por ejemplo, ugr.es puede delegar en el departamento tsc.ugr.es para gestionar este dominio inferior
- Un mismo recurso puede tener asignados varios dominios o nombres registrados, formando servidores virtuales.
  - Ejemplo: <http://web1.ugr.es> y <http://www.universidades.org> son dos servidores de dos dominios diferentes pero que se pueden asociar a la misma IP.

## Funcionamiento del servicio DNS

Las entidades principales que intervienen en el servicio son:

- Clientes DNS:
  - Programas en los ordenadores de los usuarios que hacen peticiones de resolución de nombres (Ej: un navegador web).
- Servidores DNS:
  - Máquinas que responden a las consultas realizadas por los Clientes DNS. Respuesta CON autoridad (Respuesta CON autoridad)
  - Pueden dar la respuesta bien por tener autoridad sobre el dominio en cuestión o bien por tenerla en su caché (Respuesta SIN autoridad)
  - En caso de no tenerla pueden consultar a otros servidores DNS (No conoce respuesta)

### Formato de la DB de DNS

#### Formato de los mensajes DNS

## 3. Navegación web

La WWW (World Wide Web) es la aplicación más importante en Internet. WWW es un sistema de distribución de información basado en hipertexto o hipermedios enlazados y accesibles a través de Internet.

Con un navegador web, se accede a páginas web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y se navega a través de ellas usando hiperenlaces.

En los últimos años ha crecido enormemente gracias a:

- Presentación atractiva
- Fácil de usar
- Interface unificado para todos los servicios
- Permite de manera flexible e interactiva acceder a grandes cantidades de información

Por su flexibilidad, puede dar soporte a multitud de servicios diferentes (información, publicación de contenidos, interacción entre usuarios, servicios comerciales, publicidad, cursos, bases de datos, etc.).

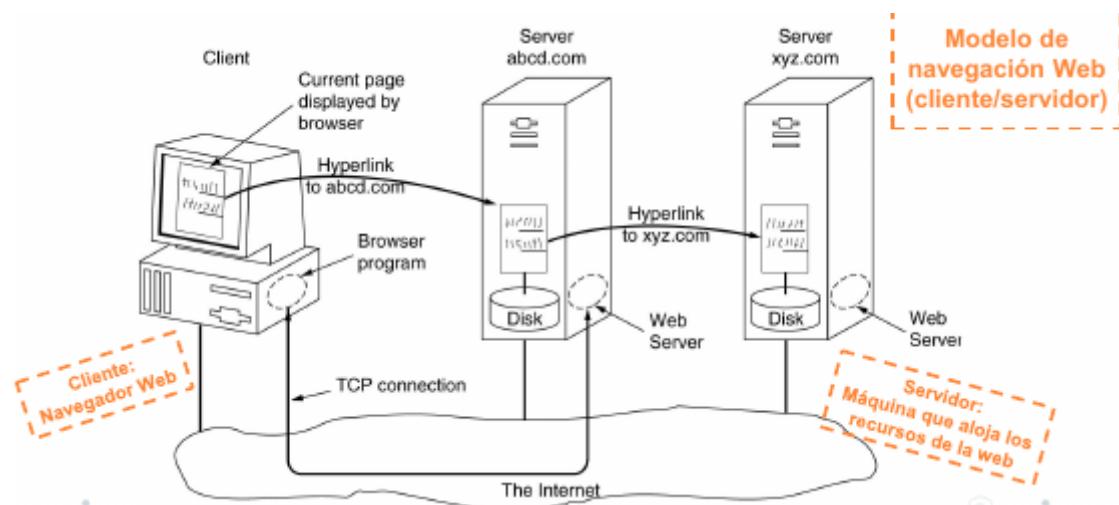
Es muy fácil publicar nueva información y hacerla accesible a todo el mundo. Está en continua evolución, y cada día sus capacidades de acceso y representación de información se vuelven más sofisticadas.

Ha sido la principal causa del espectacular crecimiento que Internet ha tenido en los últimos años, tanto en número de usuarios como en volumen de información disponible. También sirve como soporte para las denominadas "Intranets", redes privadas que usan tecnología de internet.

## Se destacan los siguientes estándares:

- El Identificador de Recurso Uniforme (URI), que es un sistema universal para referenciar recursos en la Web.
- El Protocolo de Transferencia de Hipertexto (HTTP), que especifica cómo se comunican el navegador y el servidor entre ellos.
- El Lenguaje de Marcado de Hipertexto (HTML), usado para definir la estructura y contenido de documentos de hipertexto (páginas web).
- El Lenguaje de Marcado Extensible (XML), usado para describir la estructura de los documentos

El World Wide Web Consortium (W3C) desarrolla y mantiene estos y otros estándares que permiten a los ordenadores de la Web almacenar y comunicar efectivamente diferentes formas de información.



## Cliente-Servidor Web

### • Cliente:

Un cliente web (también llamado navegador o browser) es esencialmente un programa que permite visualizar e interaccionar con páginas web. Sirve para acceder a la www y navegar por ella a través de enlaces.

El navegador hace peticiones al servidor para recibir los ficheros asociados a las páginas web (se resuelve previamente la correspondencia entre el nombre de dominio y la IP del servidor con DNS).

Tiene soporte para imágenes, sonidos y videos. No todas las páginas contienen HTML, las hay que pueden tener un documento PDF, un ícono GIF, un vídeo en MPEG...

El servidor indica el tipo MIME (Multipurpose Internet Mail Extensions). Si el tipo MIME no es de los integrados hay dos posibilidades:

- Plug-in
- Aplicaciones auxiliares

Puede utilizar otros protocolos, como ftp o file (ficheros locales).

### Procesamiento cliente:

1. El navegador determina la URL (de un enlace)
2. Accede al servicio DNS para averiguar la dirección IP de [www.epsg.upv.es](http://www.epsg.upv.es)
3. DNS contesta 158.42.144.1
4. El navegador se conecta al puerto TCP 80 de 158.42.144.1

5. Envía "GET /archivo.php?unaurlde loquesea"
6. El servidor manda el fichero archivo.php
7. Si existen imágenes u otro contenido asociado, el navegador las solicita y se envían
8. Se cierra la conexión
9. El navegador visualiza el contenido de archivo.php y los recursos asociados

- **Servidor:**

Máquina que aloja los recursos de las páginas web.

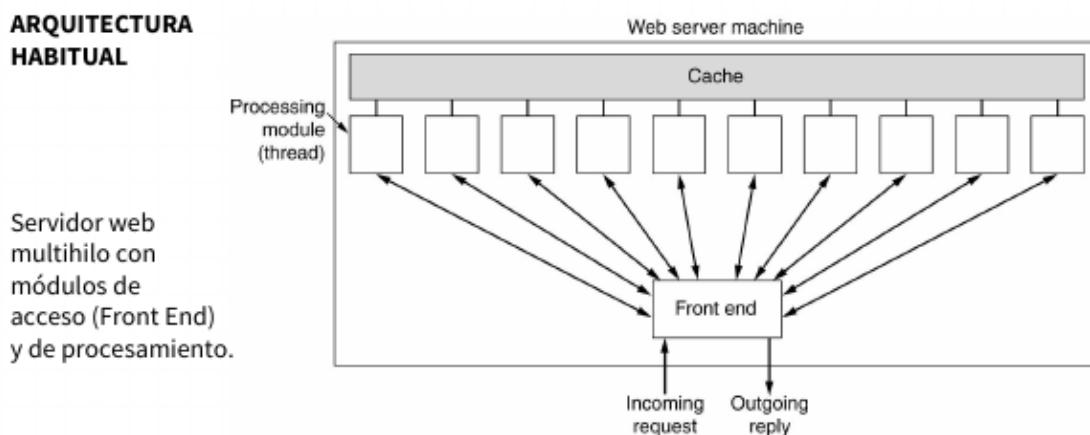
Escucha conexiones de tipo TCP en el puerto 80. Entrega los ficheros requeridos a través del protocolo HTTP.

**Procesamiento (básico):**

1. Acepta una conexión TCP de un cliente (navegador)
2. Obtiene el nombre del archivo solicitado por el cliente
3. Recupera el archivo (del disco), así como los dependientes
4. Envía el/los archivo/s al cliente
5. Libera la conexión TCP

**Procesamiento (avanzado):**

1. Resuelve el nombre de la página web solicitado
2. Autentica al cliente
3. Realiza control de acceso en el cliente
4. Realiza control de acceso en la página web
5. Verifica la caché
6. Obtiene del disco la página solicitada
7. Determina el tipo MIME que se incluirá en la respuesta
8. Devuelve la respuesta al cliente
9. Realiza una entrada en el registro del servidor



## Protocolo HTTP

El Protocolo de Transferencia de HiperTexto es un sencillo protocolo cliente/servidor que articula los intercambios de información entre los clientes y los servidores web.

Está soportado sobre los servicios que ofrecen los protocolos TCP e IP.

- Un proceso servidor escucha en un puerto de comunicaciones TCP (80) y espera solicitudes de los clientes web.
- Una vez establecida la conexión, HTTP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores. Se basa en sencillas operaciones de solicitud/respuesta.
  - cliente envía mensaje con los datos de la solicitud (request)

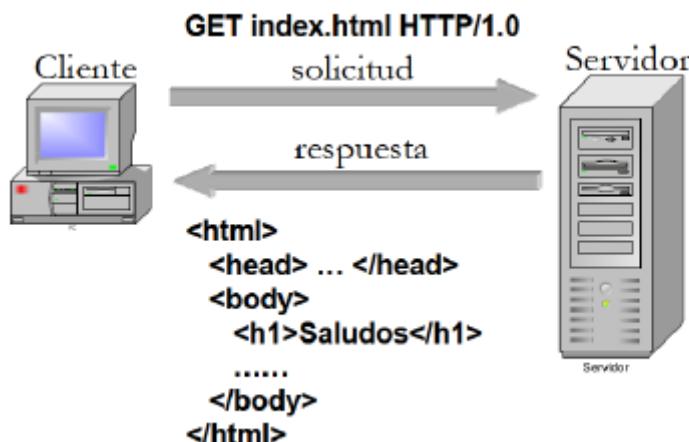
- o servidor envía mensaje con el estado de la operación y el resultado (response)

### Características:

- Protocolo basado en ASCII. De esta forma se puede transmitir cualquier tipo de documento: texto, binario, etc, respetando su formato original.
- Permite la transferencia de objetos multimedia. El contenido de cada objeto intercambiado está identificado por su clasificación MIME.
- Existen tres funciones básicas (otras no se utilizan) que un cliente puede utilizar en sus solicitudes:
  - o GET para recoger un objeto
  - o POST para enviar información al servidor -
  - o HEAD para solicitar las características de un objeto (Ejemplo: fecha de modificación de un documento HTML).
- Protocolo stateless, el servidor no mantiene información de las peticiones de los clientes.
  - o Cookie: información breve y estructurada enviada por un sitio web y almacenada en el navegador (cliente). La puede consultar para futuras visitas
- Dos tipos de servicio:
  - o No persistente: se envía únicamente un objeto en cada conexión TCP.
  - o Persistente: pueden enviarse múltiples objetos sobre una única conexión TCP entre cliente y servidor

### Funcionamiento:

1. El cliente HTTP inicia conexión TCP al servidor HTTP (proceso) en [www.ugr.es](http://www.ugr.es), en el puerto 80 (segmento SYNC de TCP)
  1. El Servidor HTTP acepta la conexión y solicita al cliente abrir la conexión (SYNC+ACK)
  2. El cliente confirma (ACK)
2. El cliente HTTP envía request message para el objeto
3. El servidor HTTP devuelve la respuesta (response message)
4. Si es persistente, envío de más objetos por la misma conexión TCP
5. Cierre conexión TCP (liberación de recursos)



## Protocolo HTTP 1.1

### CÓDIGOS DE RESPUESTA (para los *response messages del servidor*)

- **1xx** indican mensajes exclusivamente informativos
- **2xx** indican algún tipo de éxito
- **3xx** redirección al cliente a otra URL
- **4xx** indican un error
- **5xx** indican un error

## Servidor proxy

## Servidor caché

## Cookies

## 4. Correo electrónico

El correo electrónico (e-mail) es un servicio de red que permite a los usuarios enviar y recibir mensajes y archivos rápidamente mediante sistemas de comunicación electrónicos.

El correo electrónico nació a principios de los años 60. En este sistema inicial un usuario sólo era capaz de enviar mensajes a usuarios del mismo sistema. Una dirección de correo electrónico es un conjunto de palabras que identifican a una persona (únivamente) que puede enviar y recibir correo.

Dicha dirección tiene un acceso restringido mediante un nombre de usuario y una contraseña.

- El uso de la arroba (@) se incorporó en 1971, al ser un carácter que no existe en ningún nombre en el mundo.
- La arroba divide la dirección entre el usuario y el dominio del proveedor de correo (máquina en la que se aloja el correo).
- El nombre de usuario (remitente/destinatario) puede incluir letras, números y algunos signos.
- La dirección la tiene que proporcionar un proveedor de correo, que son quienes ofrecen el servicio de envío y recepción.
- Es indiferente que las letras que integran la dirección estén escritas en mayúscula o minúscula.

El correo electrónico se entrega usando una arquitectura cliente/servidor:

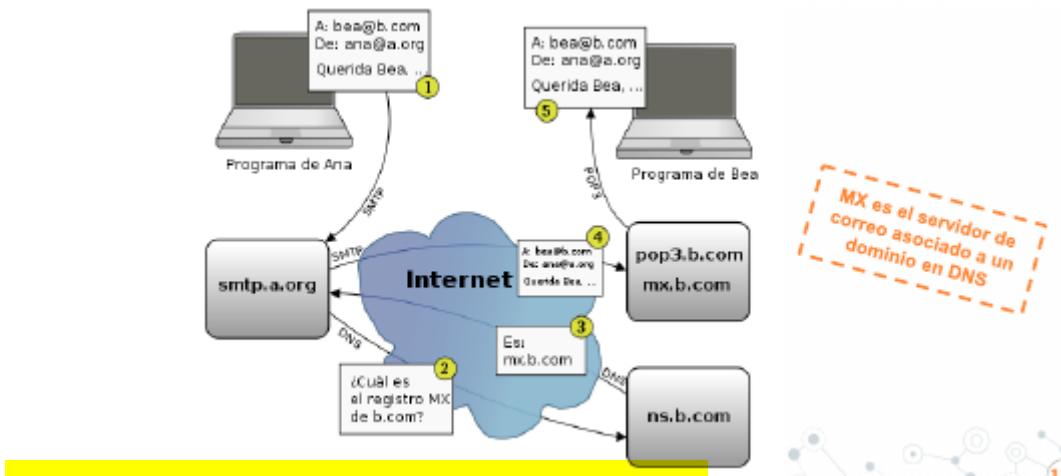
- - Un mensaje de correo electrónico se crea usando un programa de correo cliente.
  - Este programa envía el mensaje a un servidor.
  - El servidor lo redirige al servidor de correo del destinatario y allí se le suministra al cliente de correo del destinatario.
- Un e-mail consta de: destinatario, asunto y mensaje. Además puede tener ficheros adjuntos.
- Se pueden incluir además los campos:
  - CC (Copia de Carbón) para incluir destinatarios en copia .
  - CCO (Copia de Carbón Oculta) para incluir destinatarios no visibles por los demás.

### Elementos y protocolos principales:

- Cliente de correo (Mail User Agent)
- Servidor de correo (Mail Server o Mail Transfer Agent)
- Protocolo de envío: Simple Mail Transfer Protocol (SMTP)
- Protocolos de descarga (o lectura): POP3, IMAP, HTTP

- **Agente de usuario (MUA):** Compone, edita y lee mensajes de correo del buzón. Ej: Outlook, Thunderbird, etc.
- **Servidor de correo (MTA):** Reenvía mensajes salientes y almacena en buzones los mensajes entrantes de cada usuario. Permite desacoplar temporalmente a remitente y destinatario

Es un servicio que hace uso de DNS



## SMTP (Simple Mail Transfer Protocol)

Protocolo cliente/servidor sencillo. Funciona mediante TCP a través del puerto 25 (en el servidor):

- Handshaking
- Transferencia de msjs
- Cierre

Inicialmente los msjs se codificaban con ASCII 7bits. Posteriormente con MIME se pueden enviar ASCII de 8bits y formato enriquecido.

SMTP es un protocolo...

- Orientado a texto
- Orientado a conexión
- Stateful

La interacción entre cliente SMTP y servidor SMTP se realiza mediante comandos/respuesta:

- Comandos en texto ASCII
- Respuestas en código de estado y frases explicativas

### Pasos en el envío y recepción del correo

1. El usuario origen compone mediante su Agente de Usuario (MUA) un mensaje dirigido a la dirección de correo del usuario destino.
2. Se envía con SMTP (ó HTTP) el mensaje al servidor de correo (MTA) del usuario origen que lo sitúa en la cola de mensajes salientes.
3. El cliente SMTP abre una conexión TCP con el servidor de correo (MTA) (obtenido por DNS) del usuario destino.
4. El cliente SMTP envía el mensaje sobre la conexión TCP al servidor de destino.
5. El servidor de correo del usuario destino ubica el mensaje en el mailbox del usuario destino.
6. El usuario destino invoca su Agente de Usuario (MUA) para leer el mensaje utilizando POP3, IMAP ó HTTP.



### COMANDOS SMTP (Cliente)

Comando	Descripción
HELO (ahora EHLO)	Identifica el remitente al destinatario.
MAIL FROM	Identifica una transacción de correo e identifica al emisor.
RCPT TO	Se utiliza para <b>identificar un destinatario individual</b> . Si se necesita identificar múltiples destinatarios es necesario repetir el comando.
DATA	Permite enviar una serie de líneas de texto. El tamaño máximo de una línea es de 1.000 caracteres. Cada línea va seguida de un retorno de carro y avance de línea <CR><LF>. La <b>última línea debe llevar únicamente el carácter punto "."</b> seguido de <CR><LF>.
RSET	Aborta la transacción de correo actual.
NOOP	No operación. <b>Indica al extremo que envíe una respuesta positiva. Keepalives</b>
QUIT	Pide al otro extremo que envíe una respuesta positiva y cierre la conexión.
VRFY	Pide al receptor que confirme que un nombre identifica a un destinatario válido.
EXPN	Pide al receptor la <b>confirmación de una lista de correo</b> y que devuelva los nombres de los usuarios de dicha lista.
HELP	Pide al otro extremo información sobre los comandos disponibles.
TURN	El emisor pide que se <b>inviertan los papeles</b> , para poder actuar como receptor. El receptor puede negarse a dicha petición.
SOML	Si el destinatario está conectado, entrega el mensaje directamente al terminal, en caso contrario lo entrega como correo convencional.
SAML	Entrega del mensaje en el buzón del destinatario. En caso de estar conectado también lo hace al terminal.
SEND	Si el destinatario está conectado, entrega el mensaje directamente al terminal.

### RESPUESTAS SMTP (Servidor)

Código	Descripción
211	Estado del sistema.
214	Mensaje de ayuda.
220	Servicio preparado.
221	Servicio cerrando el canal de transmisión.
250	Solicitud completada con éxito.
251	Usuario no local, se enviará a <dirección de reenvío>
354	Introduzca el texto, finalice con <CR><LF>.<CR><LF>.
421	Servicio no disponible.
450	Solicitud de correo no ejecutada, servicio no disponible (buzón ocupado).
451	Acción no ejecutada, error local de procesamiento.
452	Acción no ejecutada, insuficiente espacio de almacenamiento en el sistema.
500	Error de sintaxis, comando no reconocido.
501	Error de sintaxis. P.ej contestación de SMTP a ESMTP
502	Comando no implementado.
503	Secuencia de comandos errónea.
504	Parámetro no implementado.
550	Solicitud no ejecutada, buzón no disponible.
551	Usuario no local, pruebe <dirección de reenvío>. Si no se tiene cuenta
552	Acción de correo solicitada abortada.
553	Solicitud no realizada (error de sintaxis).
554	Fallo en la transacción.

### EJEMPLO DE COMANDOS Y RESPUESTAS SMTP

```

S: 220 smtp1.ugr.es
C: EHLO ugr.es
S: 250 smtp1.ugr.es
C: MAIL FROM: uno@ugr.es
S: 250 Ok
C: RCPT TO: dos@ugr.es
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Subject: Correo estúpido
C: Tengo ganas de enviarte un correo...
C: ¿Te importa si lo hago?
C: .
S: 250 Ok: queued as KJSADHFFWDF
C: QUIT
S: 221 Bye

```

EHLO ⇔ HELLO

## MIME (Multipurpose Internet Mail Protocol Extensions)

MIME está especificado los RFCs: 2045, 2046, 2047, 4288, 4289 y 2077. Nada cambia respecto a la arquitectura de correo anterior.

Las extensiones de MIME van encaminadas a soportar:

- Texto en conjuntos de caracteres distintos de US-ASCII.
- Adjuntos que no son de tipo texto.
- Cuerpos de mensajes con múltiples partes (multi-part).
- Información de encabezados con conjuntos de caracteres distintos de ASCII.

### Cabeceras de mensaje MIME:

Cabecera	Descripción
MIME-Version:	Identifica la versión de MIME. Si no existe se considera que el mensaje es texto normal en inglés.
Content-Description:	Cadena de texto que describe el contenido. Esta cadena es necesaria para que el destinatario sepa si desea descodificar y leer el mensaje o no.
Content-Id:	Identificador único, usa el mismo formato que la cabecera estándar Message-Id.
Content-Transfer-Encoding:	Indica la manera en que está envuelto el cuerpo del mensaje.
Content-Type:	Especifica la naturaleza del cuerpo del mensaje.

- Content-Transfer Encoding:
  - Indica la manera en que está envuelto el cuerpo para su transmisión, ya que podría haber problemas con la mayoría de los caracteres distintos de letras, números y signos de puntuación.
  - Existen 5 tipos de codificación (RFC1521) : ASCII 7, ASCII 8, codificación binaria, base64 y entrecorbillada-imprimible.7.2.
- Content-Type: La lista inicial de tipos y subtipos especificada por el RFC 1521 es

Tipo	Subtipo	Descripción
Text	Plain	Texto sin formato.
	Richtext	Texto con comandos de formato sencillos.
Image	Gif	Imagen fija en formato GIF.
	Jpeg	Imagen fija en formato JPEG.
Audio	Basic	Sonido.
Video	Mpeg	Película en formato MPEG.
Application	Octet-stream	Secuencia de bytes no interpretada.
	Postscript	Documento imprimible PostScript.
Message	Rfc822	Mensaje MIME RFC 822.
	Partial	Mensaje dividido para su transmisión.
	External-body	El mensaje mismo debe obtenerse de la red.
Multipart	Mixed	Partes independientes en el orden especificado.
	Alternative	Mismo mensaje en diferentes formatos.
	Parallel	Las partes deben verse simultáneamente.
	Digest	Cada parte es un mensaje RFC 822 completo.

- Application:
  - El tipo **application** es un tipo general para los formatos que requieren procesamiento externo no cubierto por ninguno de los otros tipos.
  - El subtipo **octet-stream** simplemente es una secuencia de bytes no interpretados, tal que a su recepción, un agente de usuario debería *presentarla en la pantalla sugiriendo al usuario que se copie en un archivo y solicitarlo un nombre de archivo*.
  - El subtipo **postscript**, se refiere al lenguaje PostScript de Adobe Systems. Aunque un agente de usuario puede llamar a un intérprete PostScript externo para visualizarlo, hacerlo no está exento de riesgos al ser PostScript un lenguaje de programación completo.

- Message:

- El tipo **message** permite que un mensaje esté encapsulado por completo dentro de otro. Este esquema es útil para reenviar, correo electrónico.
- El subtipo **rfc822** se utiliza cuando se encapsula un mensaje RFC 822 completo en un mensaje exterior.
- El subtipo **partial** hace posible dividir un mensaje encapsulado en pedazos y enviarlos por separado. **Los parámetros hacen posible ensamblar correctamente todas las partes en el destino.** Ej: 1/3, 2/3, 3/3.
- El subtipo **external-body** puede usarse para mensajes muy grandes, por ejemplo películas de video. En lugar de incluir el archivo mpeg en el mensaje, se da una dirección de FTP y el agente de usuario del receptor puede obtenerlo a través de la red cuando se requiera.
- Multipart:
  - El tipo es **multipart**, que permite que un mensaje contenga más de una parte, con el comienzo y el fin de cada parte claramente delimitados.
  - El subtipo **mixed** permite que cada parte sea diferente.
  - El subtipo **alternative** indica que cada parte contiene el mismo mensaje, pero expresado en un medio o codificación diferente.
  - El subtipo **parallel** se usa cuando todas las partes deben “verse” simultáneamente, por ejemplo, en los canales de audio y video de las películas.
  - El subtipo **digest** se usa cuando se juntan muchos mensajes en un mensaje compuesto.

## POP3 (Post Office Protocol)

Es un protocolo utilizado para la entrega de mensajes al usuario final. Obtiene el correo electrónico del buzón remoto (en el servidor de correo) y lo almacena en la máquina local del usuario para su lectura posterior.

Por defecto, una vez transferido el mensaje, éste se borra automáticamente del servidor de correo. La versión 3 (la actual) utiliza TCP sobre el puerto 110.

Se basa en comandos y respuestas en texto ASCII, como SMTP. Tiene comandos para que un cliente establezca una sesión (USER y PASS), la termine (QUIT), obtenga mensajes (RETR) y los borre (DELE).

- POP3 se inicia cuando el usuario arranca el gestor de correo. Éste llama al servidor y establece una conexión TCP con el agente de transferencia de mensajes en el puerto 110.
- El protocolo POP3 administra la autenticación utilizando el nombre de usuario y la contraseña. Aunque no es seguro porque la información no va encriptada.
- Para añadir seguridad a POP3, es posible utilizar la encriptación Secure Socket Layer (SSL) para la autenticación del cliente y las sesiones de transferencias de datos.
- POP3 bloquea las bandejas de entrada durante el acceso, lo que significa que es imposible que dos usuarios accedan de manera simultánea a la misma bandeja de entrada.

## Comandos POP3:

Comando	Descripción
USER identification	Este comando permite la autenticación. Debe estar seguido del nombre de usuario, es decir, una cadena de caracteres que identifique al usuario en el servidor. El comando <i>USER</i> debe preceder al comando <i>PASS</i> .
PASS password	El comando <i>PASS</i> permite especificar la contraseña del usuario cuyo nombre ha sido especificado por un comando <i>USER</i> previo.
STAT	Información acerca de los mensajes del servidor
RETR	Número del mensaje que se va a recoger
DELE	Número del mensaje que se va a eliminar
LIST [msg]	Número del mensaje que se va a mostrar
NOOP	Permite mantener la conexión abierta en caso de inactividad
TOP <messageID> <n>	Comando que muestra <i>n</i> líneas del mensaje, cuyo número se da en el argumento. En el caso de una respuesta positiva del servidor, éste enviará de vuelta los encabezados del mensaje, después una línea en blanco y finalmente las primeras <i>n</i> líneas del mensaje.
UIDL [msg]	Solicitud al servidor para que envíe una línea que contenga información sobre el mensaje que eventualmente se dará en el argumento. Esta línea contiene una cadena de caracteres denominada <i>unique identifier listing</i> ( <i>lista de identificadores únicos</i> ) que permite identificar de manera única el mensaje en el servidor, independientemente de la sesión. El argumento opcional es un número relacionado con un mensaje existente en el servidor POP, es decir, un mensaje que no se ha borrado.
QUIT	El comando <i>QUIT</i> solicita la salida del servidor POP3. Lleva a la eliminación de todos los mensajes marcados como eliminados y envía el estado de esta acción.

### Ejemplo POP3:

#### Fase de autorización

Comandos del cliente:

**user:** nombre de usuario

**pass:** contraseña

Respuestas del servidor

**+OK**

**-ERR**

#### Fase de transacción, cliente:

**list:** lista mensajes por número

**retr:** obtiene mensajes por num.

**dele:** borra

**quit**

#### Fase de actualización del servidor

(tras desconexión)

```

S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

```

## IMAP (Internet Message Access Protocol)

Es un protocolo utilizado para la entrega de mensajes al usuario final, alternativo a POP3. La idea en que se basa IMAP es que el servidor de correo electrónico mantenga un depósito central al que puede accederse desde cualquier máquina.

No copia el correo electrónico en la máquina del usuario y no lo borra del servidor. IMAP supone que todo el correo electrónico permanecerá en el servidor de manera indefinida. La versión actual es IMAP4 (revisión en RFC 3501).

- Permite organización en carpetas (o buzones) en el lado del servidor (MTA).
  - Para ello, mantiene información entre sesiones (asociando flags a los mensajes).
- Permite la descarga de partes de los mensajes.
- Posible acceder con varios clientes (POP también, pero en modo descargar y guardar).
- Para seguridad adicional, es posible utilizar la encriptación SSL para la autenticación de clientes y para las sesiones de transferencia de datos.
- Un proceso cliente IMAP se comunica con el proceso servidor IMAP identificado a través del número de puerto 143 TCP (IMAP 4).

## POP3 vs. IMAP4

- IMAP es más rápido
- Operación en línea o fuera de línea.
  - Con POP3 los clientes se conectan brevemente al servidor de correo (sólo para descargar).
  - Con IMAP4, los clientes permanecen conectados el tiempo que su interfaz permanezca activa y descargan los mensajes bajo demanda.
- Conexión única o múltiple.
  - POP3 supone que el cliente conectado es el único dueño de una cuenta de correo.
  - IMAP4 permite accesos simultáneos de múltiples clientes y proporciona ciertos mecanismos a los mismos para que se detecten los cambios hechos a un buzón de correo por otro cliente concurrentemente conectado.
- Recuperación de mensajes completos o parciales.
  - Casi todo el correo electrónico en Internet es transmitido en formato MIME. IMAP4 permite a los clientes obtener separadamente cualquier parte MIME individual, así como obtener porciones de las partes individuales o los mensajes completos.
- Información de mensajes y estado del mensaje en el servidor.
  - POP3 elimina los mensajes del servidor.
  - IMAP4 mantiene los mensajes en el servidor. Además, mediante el uso de marcas/señales definidas en el protocolo IMAP4 de los clientes, se puede asignar y conocer el estado de un mensaje (si ha sido o no leído, respondido o eliminado).  
Estas señales se almacenan en el servidor, de manera que varios clientes conectados al mismo correo en diferente tiempo pueden detectar los cambios hechos por otros clientes.
- Búsqueda y recuperación selectiva de mensajes.
  - POP3 recupera todos los mensajes.
  - IMAP4 permite hacer búsquedas en el servidor para acceder sólo a mensajes determinados.
- Facilidad para incluir extensiones.
  - IMAP se diseñó para incorporar extensiones de manera relativamente sencilla. Por ejemplo IMAP IDLE permite que el servidor envíe una señal al cliente cuando haya nuevos correos, evitando que el cliente tenga que preguntar cada cierto tiempo.

Características	POP3	IMAP4
RFC	RFC 1939	RFC 2060
Puerto TCP utilizado	110	143
Dónde se almacena el correo electrónico	PC del usuario	Servidor
Tiempo de conexión requerido	Poco	Mucho
Uso de recursos del servidor	Mínimo	Amplio
Quién mantiene los buzones	Usuario	ISP
Bueno para usuarios móviles	No	Si
Descargas parciales de mensajes	No	Si
Sencillo de implementar Soporte amplio	Si	No

## Puertos

Los puertos utilizados para los servicios relacionados con Correo Electrónico son:

- POP3 - puerto 110
- IMAP - puerto 143
- SMTP - puerto 25
- HTTP - puerto 80
- Secure SMTP (SSMTP) - puerto 465
- Secure IMAP (IMAP4-SSL) - puerto 585
- IMAP4 over SSL (IMAPS) - puerto 993
- Secure POP3 (SSL-POP) - puerto 995

## 5. Aplicaciones multimedia

El término multimedia hace referencia al uso combinado de diferentes medios de comunicación: texto, imagen, sonido, animación y video.

Los programas informáticos que utilizan de forma combinada y coherente con sus objetivos diferentes medios, y permiten la interacción con el usuario son aplicaciones multimedia interactivas.

La evolución producida en los sistemas de comunicación ha dado lugar a este tipo heterogéneo de aplicaciones o programas que tienen dos características básicas:

- Multimedia: Uso de múltiples tipos de información (textos, gráficos, sonidos, animaciones, videos, etc.) integrados coherentemente.
- Hipertexto: Interactividad basada en los sistemas de hipertexto, que permiten decidir y seleccionar la tarea que deseamos realizar, rompiendo la estructura lineal de la información.

### Conceptos:

- Calidad de servicio (QoS): capacidad de ofrecer el rendimiento requerido para una aplicación IP ofrece
- Mejor esfuerzo (best effort): sin garantías de QoS

### TIPOS DE APLICACIONES

- Flujo de audio y vídeo (streaming) almacenado. Ej: YouTube
- Flujo de audio y vídeo en vivo. Ej: emisoras de radio o IPTV
- Audio y vídeo interactivo. Ej: Skype

### CARACTERÍSTICAS PRINCIPALES

- Ocupan un elevado ancho de banda.
- Tolerantes relativamente a la pérdida de datos.
- Exigen retardo (delay) acotado.
- Exigen fluctuaciones del retardo (jitter) acotado.
- Se pueden beneficiar de usar de multicast (direcciones destino de grupo).

### Network Delivery Issues

### Low Quality Source or Overly Aggressive Optimization

