



Fundamentos de Redes

Tema 1.

Introducción a los fundamentos de redes

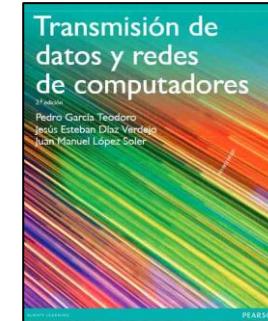
Antonio M. Mora García



Bibliografía

Básica

- P. García-Teodoro, J.E. Díaz-Verdejo, J.M. López-Soler.
Transmisión de datos y redes de computadores, 2^a Edición.
Editorial Pearson, 2014. **CAPÍTULO 1**



Complementaria

- James F. Kurose, Keith W. Ross. Redes de computadoras. Un enfoque descendente. 7^o Edición. Editorial Pearson S.A., 2017.
CAPÍTULO 1



Preguntas previas

- ◎ ¿Qué son las redes?
- ◎ ¿Qué son las comunicaciones?
- ◎ ¿Qué elementos son necesarios para poder establecer una comunicación?
- ◎ ¿Cómo están diseñadas las redes? ¿Quién define cómo deben diseñarse?
- ◎ ¿Qué es Internet?

Índice

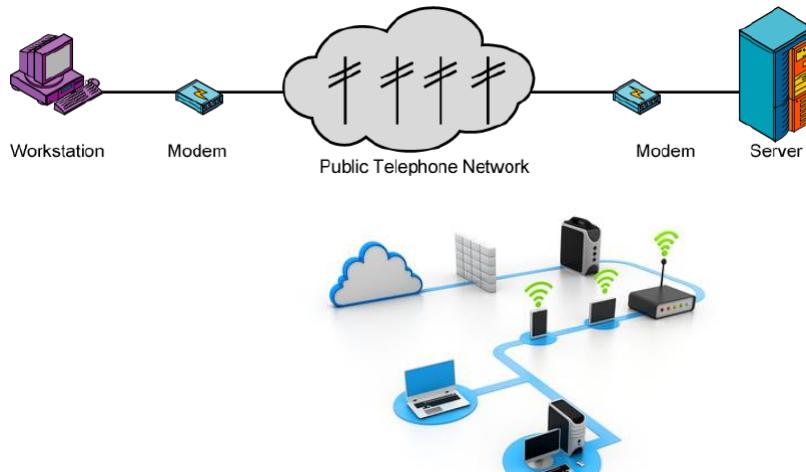
- ◎ **1.1.** Sistemas de comunicación y redes.
- ◎ **1.2.** Diseño y estandarización de redes.
- ◎ **1.3.** Terminología y servicios.
- ◎ **1.4.** Internet: Arquitectura y direccionamiento.
- ◎ **1.5.** Cuestiones y ejercicios.

TEMA 1. Introducción

- **1.1. Sistemas de comunicación y redes.**
- **1.2. Diseño y estandarización de redes.**
- **1.3. Terminología y servicios.**
- **1.4. Internet: Arquitectura y direccionamiento.**
- **1.5. Cuestiones y ejercicios.**

¿Qué es una 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**.



Sistema de comunicación

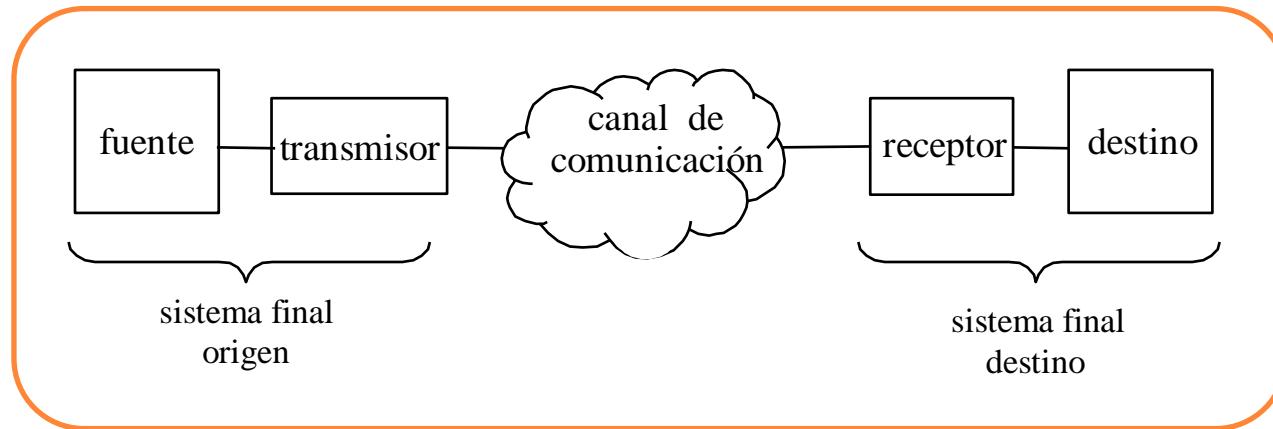
- **Comunicación:**

Es la **transferencia de información** con sentido **desde** un lugar (**remitente**, fuente, originador, emisor) a otro lugar (**destino**, receptor).

- **Información:**

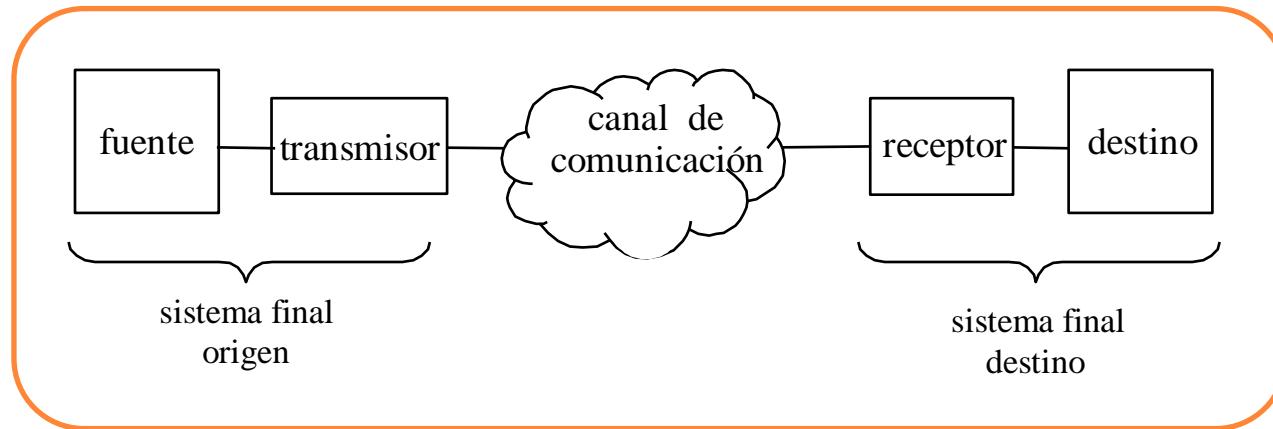
Es un **patrón** físico al cual se le ha asignado un **significado** comúnmente **acordado**. El patrón **debe ser único** (separado y distinto), capaz de ser **enviado por el transmisor**, y capaz de ser detectado y **entendido por el receptor**.

Sistema de comunicación



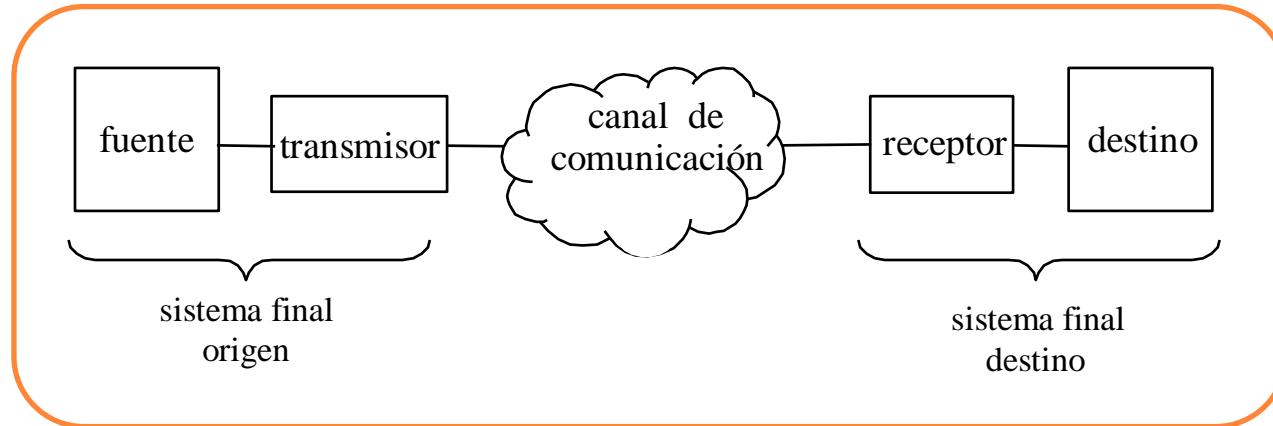
- **Fuente:** Dispositivo que **genera los datos a transmitir**. (Ej. un teléfono o un PC)
- **Transmisor:** Por lo general los datos los genera la fuente, pero no los transmite en el formato que los genera. El transmisor, **transforma y codifica esta información**, normalmente en forma de **señales electromagnéticas** (EM) susceptibles de **ser transmitidas a través de algún sistema de transmisión o medio**.

Sistema de comunicación



- **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.

Sistema de comunicación



- **Receptor:** Elemento que **recibe la información** en forma de señal EM a través del sistema de transmisión. 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).

Sistema de comunicación

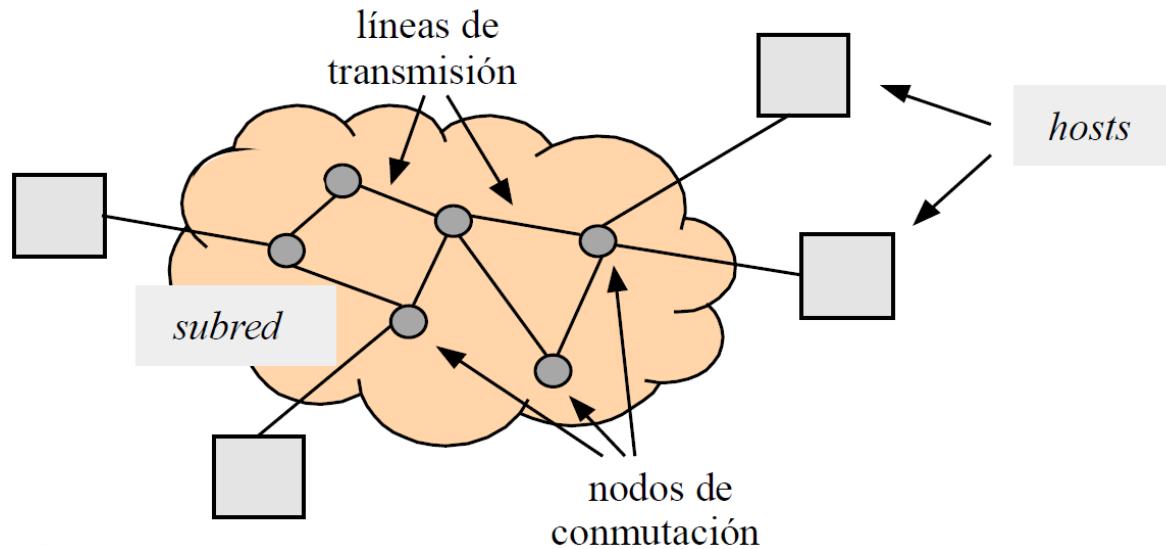
- **Tareas de un sistema de comunicación:**

- **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 iniciar/finalizar la comunicación)
- **Detección y corrección de errores** (si la señal se distorsiona)
- **Control del flujo** (mecanismos para evitar saturación)
- **Direccionamiento** (identidad del destino)
- **Encaminamiento** (elección de la ruta hasta el destino)
- **Recuperación** (ante pérdida de conexión)
- **Seguridad** (evitar captura o alteración de los datos)

Redes

- **Qué esperamos de una red (de computadores, de móviles, de dispositivos...):**
 - **Autonomía** → con capacidad de procesar información
 - **Interconexión** → mediante un sistema de comunicación
 - **Intercambio de Información** → con eficacia y transparencia
- **Razones (motivación) para su uso:**
 - Compartir recursos
 - Escalabilidad
 - Fiabilidad, robustez → duplicidad (redundancia)
 - Ahorro de costes

Redes – Estructura general y elementos



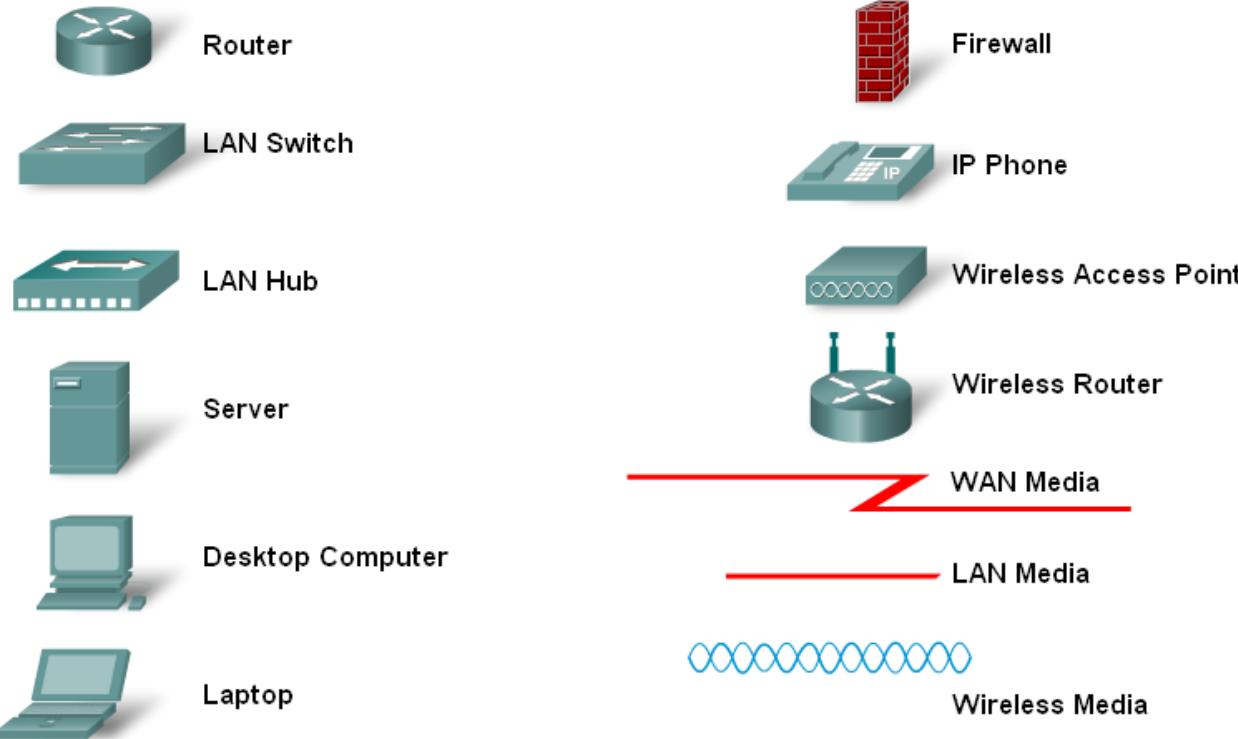
- Una red consta de:
 - Hosts → máquinas finales
 - Subred → nodos de conmutación + líneas de transmisión

Redes – Estructura general y elementos

- Hosts:
 - Servidores, estaciones de trabajo, teléfonos, PDAs, tostadoras, TVs, etc.
 - Ejecutan aplicaciones de red
 - Forman el **borde** (edge) de la red
 - Conectados con la red mediante enlaces de comunicaciones (cobre, fibra, radio, satélite)
- Comutadores:
 - Reenvían la información a través de rutas o caminos (*paths*) dentro de la red.
 - Son transparentes a los datos.
 - Comutadores telefónicos o routers en el caso de Internet.
 - Interconectados mediante enlaces de comunicaciones.
 - Forman el **núcleo** (core) de la red

Redes – Estructura general y elementos

- Simbología típica usada en el diseño de redes:



Redes - Componentes

- Los **componentes** de una red tienen **funciones específicas** y se utilizan dependiendo de las **características físicas (hardware)** que tienen.
- Para elegirlos se requiere **considerar las necesidades** y los **recursos económicos** de quien se desea conectar a la red, por eso deben conocerse las características técnicas de cada componente de red.

Redes – Componentes principales

- **Servidor (server):**

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 (workstation):**

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.

Redes – Componentes principales

- **Nodo de red (node):**

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.

- **Tarjetas de red (interface):**

Son 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.

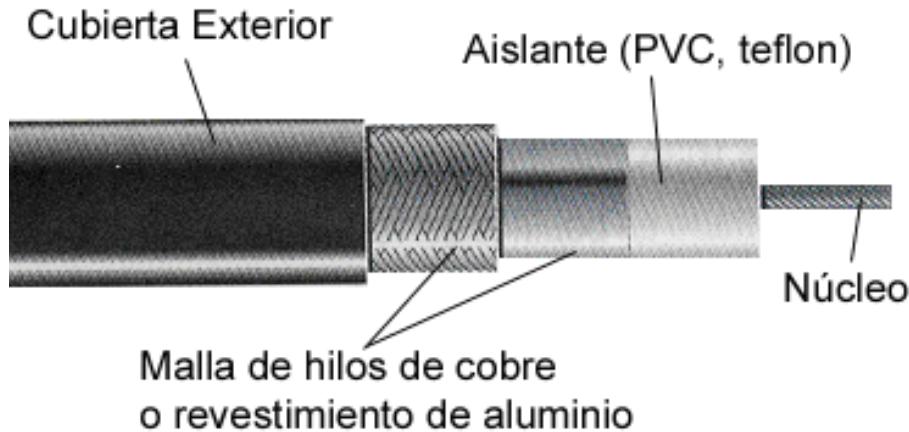
Redes – Medios de transmisión

- Estos elementos **hacen posible la comunicación** entre dos computadoras. Son 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.
- Existen **diferentes tipos de cable** y su elección depende de las necesidades de la comunicación de red.

Redes – Medios de transmisión

- **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.



Redes – Medios de transmisión

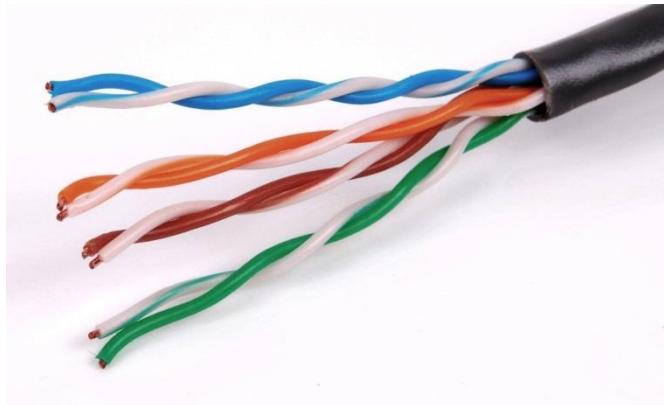
- **Cable par trenzado:**

- Son cable de cobre.
- Se utiliza para la conexión de redes o entre nodos de una red.
- Tiene **4 pares de cables**, pero existen **3 variantes**:
 - UTP (unshielded twisted pair)
 - STP (shielded twisted pair)
 - FTP (foiled twisted pair)

Redes – Medios de transmisión

- **Cable par trenzado UTP (par trenzado no apantallado):**

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.



Redes – Medios de transmisión

- **Cable par trenzado STP (par trenzado apantallado):**

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.



Redes – Medios de transmisión

- **Cable par trenzado FTP** (par trenzado con pantalla global):

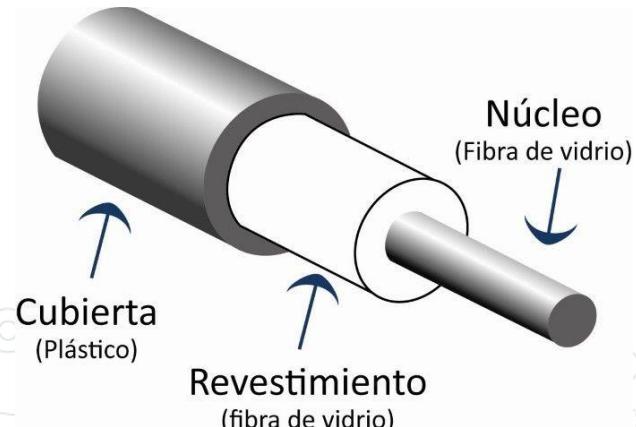
Los pares no tienen un aislamiento propio, como en STP, pero cuenta con una malla reflectante que cubre todo el conjunto. Es menos costoso que el STP, pero también menos efectivo, aunque da mejor rendimiento que UTP.



Redes – Medios de transmisión

- **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.
- 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.

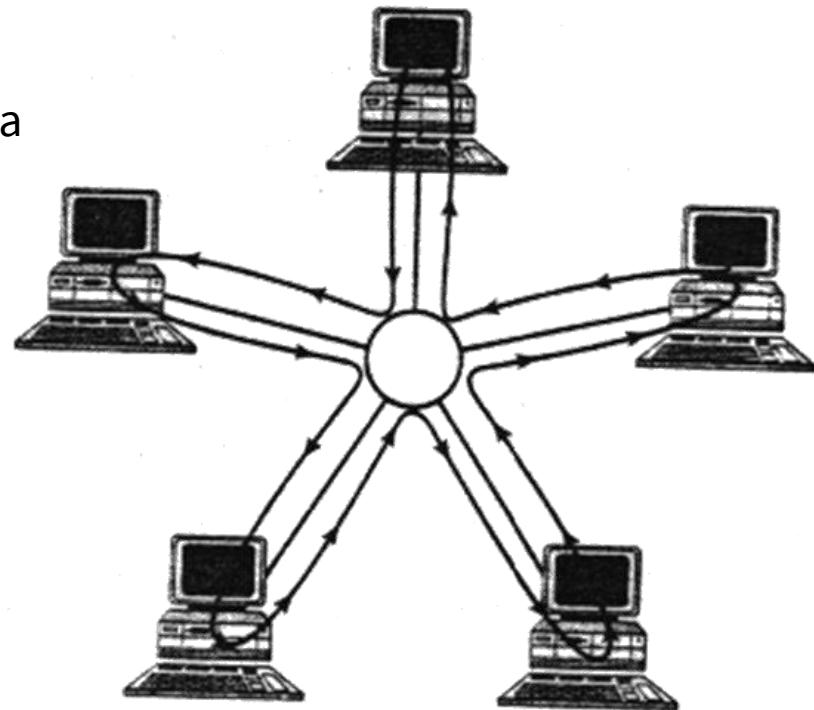


Redes – Topología

- 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:
 - **Topología física:** Se refiere al diseño actual del medio de transmisión de la red.
 - **Topología lógica:** Se refiere a la trayectoria lógica que los datos a su paso por los nodos de la red.

Redes – Topología

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



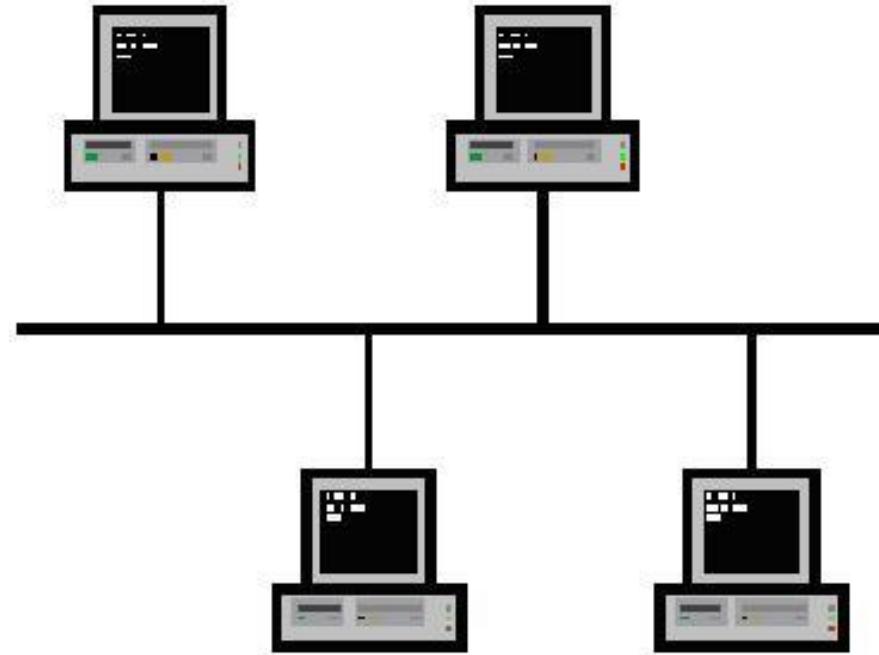
Redes – Topología

- **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.

Redes – Topología

- **Topología en bus:**



Redes – Topología

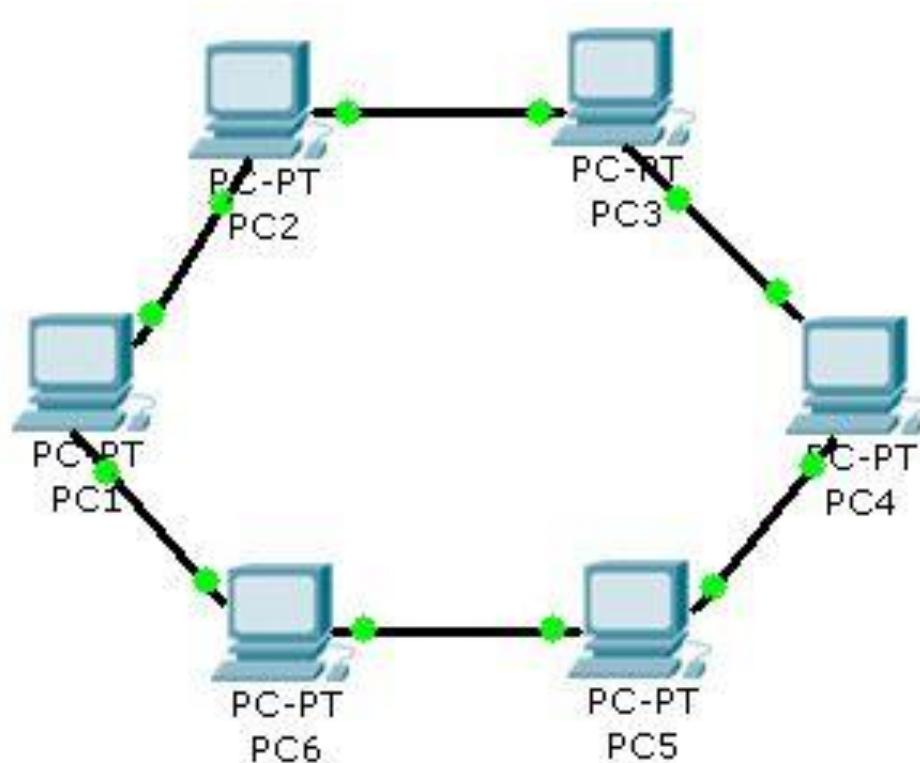
- **Topología en anillo:**

- 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).

Redes – Topología

- **Topología en anillo:**



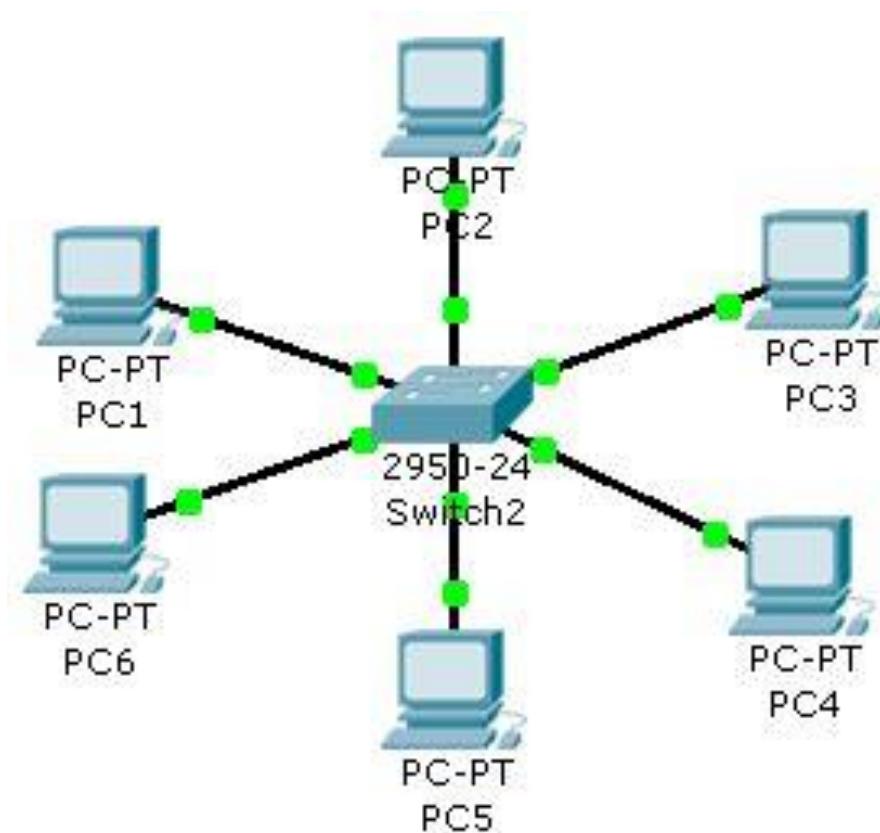
Redes – Topología

- **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**.

Redes – Topología

- **Topología en estrella:**



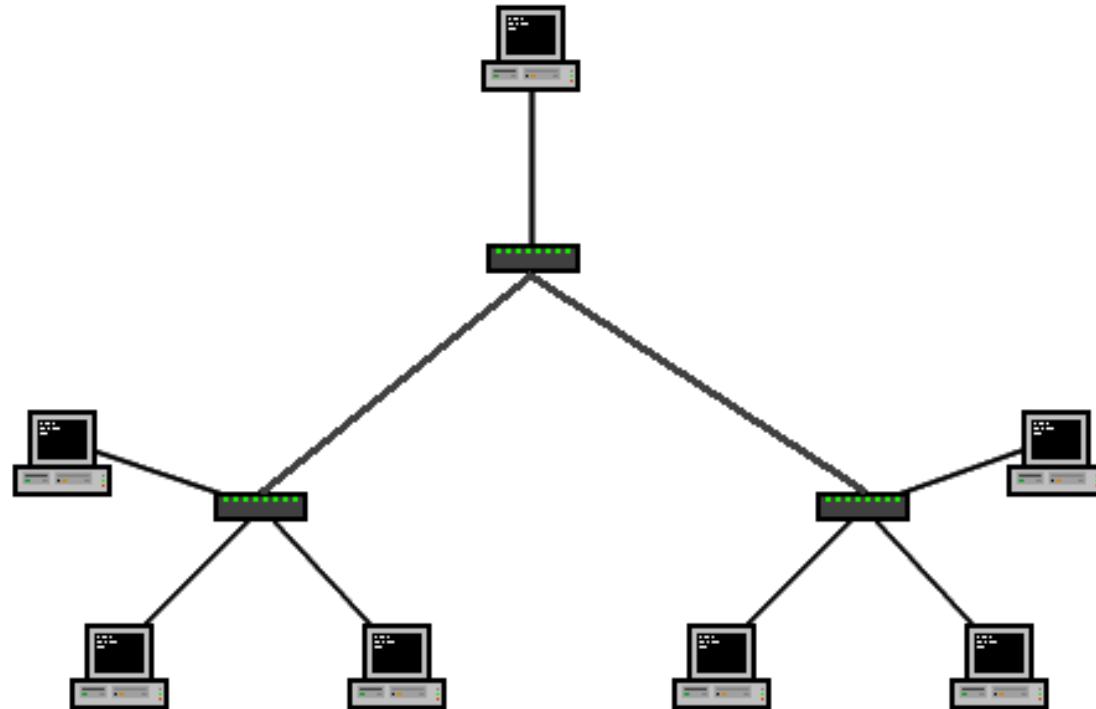
Redes – Topología

- **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 antes errores.**

Redes – Topología

- **Topología en árbol:**



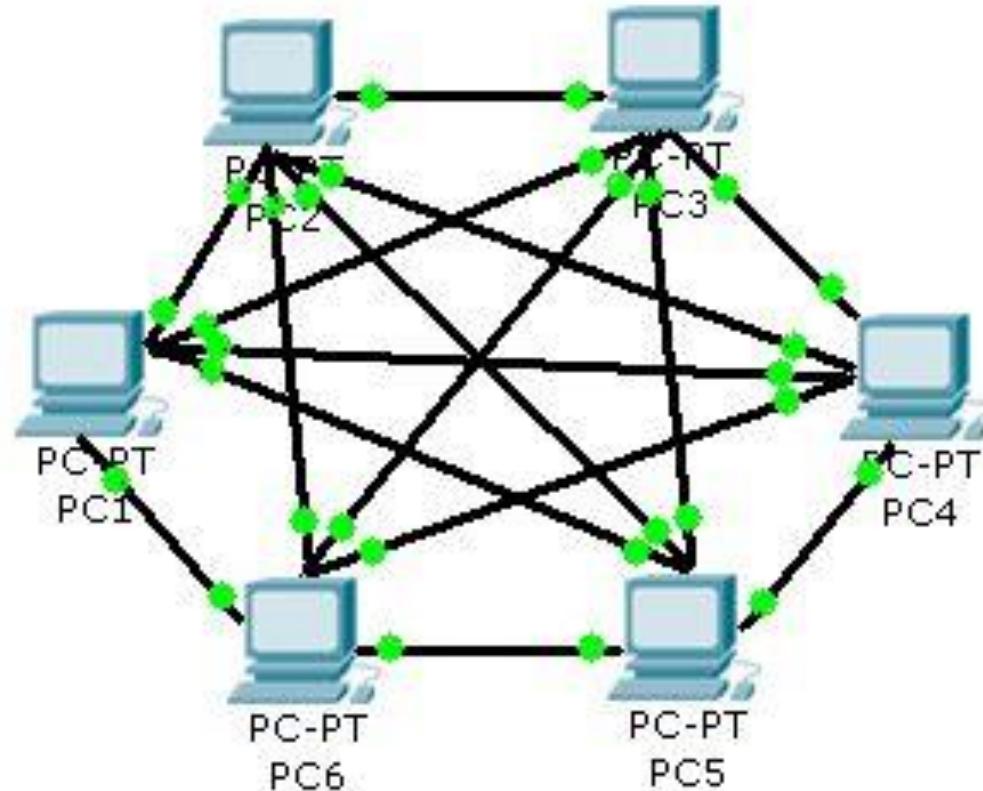
Redes – Topología

- **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.

Redes – Topología

- **Topología en malla:**



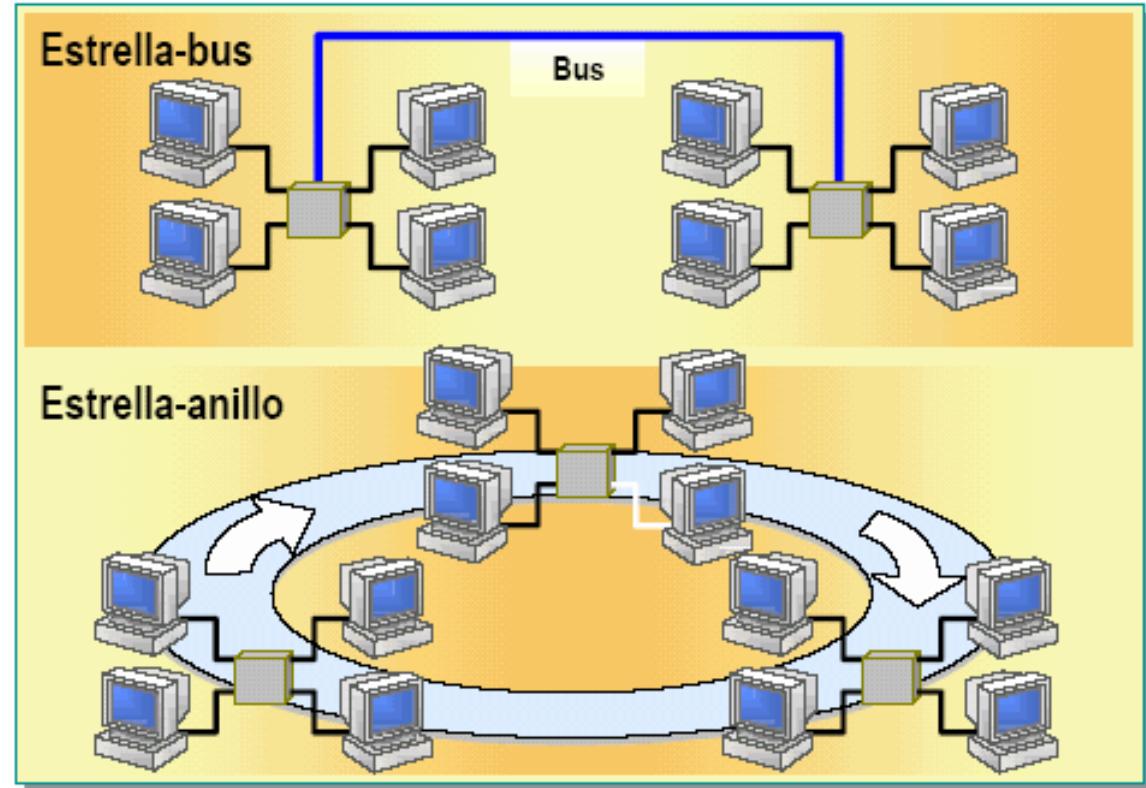
Redes – Topología

- **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.

Redes – Topología

- **Topología híbrida:**



Redes – Clasificación de las redes

Según su tamaño y extensión

- **LAN:**

Las redes de área local (Local Area Network) son 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).

Redes – Clasificación de las redes

Según su tamaño y extensión

- **MAN:**

Las redes de área metropolitana (Metropolitan Area Network) son 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:**

Las redes de área amplia (Wide Area Network) 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.

Redes – Clasificación de las redes

Según su tecnología de transmisión

- **Redes Broadcast:**

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.

- **Redes Point-to-Point:**

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.

Redes – Clasificación de las redes

según el tipo de transferencia de datos que soportan

- **Redes de transmisión simple:**

Son aquellas redes en las que los datos sólo pueden viajar en un sentido.

- **Redes Half-duplex:**

Aquellas en las que los datos pueden viajar en ambos sentidos, pero sólo en uno de ellos en un momento dado. Es decir, sólo puede haber transferencia en un sentido a la vez.

- **Redes Full-duplex:**

Aquellas en las que los datos pueden viajar en ambos sentidos a la vez.



TEMA 1. Introducción

- 1.1. Sistemas de comunicación y redes.
- **1.2. Diseño y estandarización de redes.**
- 1.3. Terminología y servicios.
- 1.4. Internet: Arquitectura y direccionamiento.
- 1.5. Cuestiones y ejercicios.

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



Modelo de referencia:

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

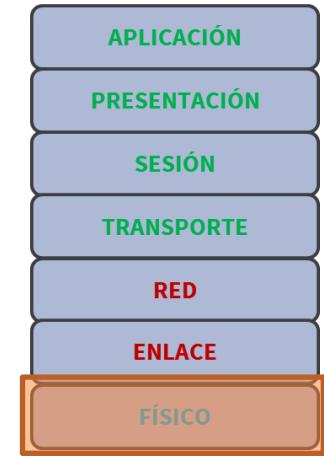
Modelo de referencia OSI

- 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).



Modelo de referencia OSI

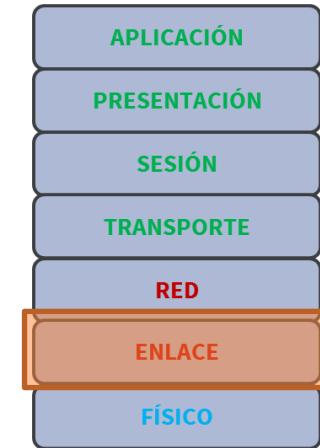
- La **CAPA FÍSICA** se encarga de las **conexiones físicas** hacia la red en lo que se refiere al **medio físico**; características del medio y la **forma** en la que se **transmite la información**.
- Se encarga de **transformar** una **trama de datos** proveniente **del nivel de enlace** en **una 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).



PDU
Flujo de bits

Modelo de referencia OSI

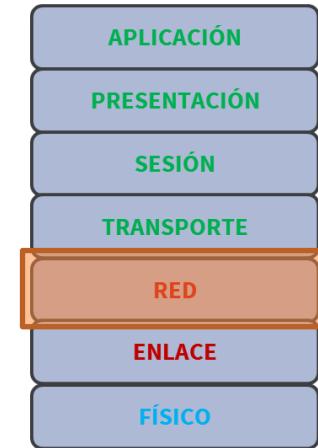
- La **CAPA DE ENLACE DE DATOS** se encarga **proporcionar** una transmisión sin errores, es decir, un **tránsito de datos fiable a través de un enlace físico**.
- Debe **crear y reconocer** los **límites de las tramas** y **resolver los problemas** derivados del **deterioro, pérdida o duplicidad** de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



PDU
Trama

Modelo de referencia OSI

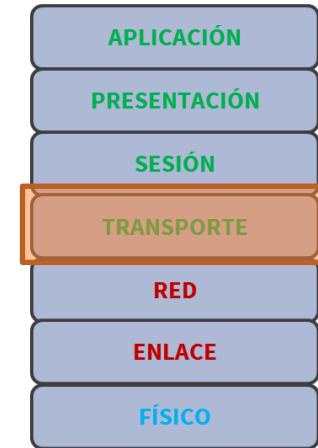
- La **CAPA DE RED** tiene como objetivo **hacer que los datos lleguen desde el origen al destino**, aun cuando ambos no estén conectados directamente.
- Los **dispositivos** que facilitan tal tarea se denominan **routers o enrutadores**.
- La capa de red lleva un **control de la congestión de red**, 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**.



PDU
Paquete

Modelo de referencia OSI

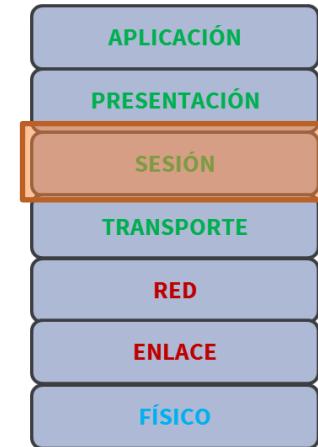
- La **CAPA DE TRANSPORTE** tiene como función básica aceptar los **datos** enviados por las **capas superiores**, **dividirlos** en pequeñas partes si es necesario, **y pasarlos a la capa de red**.
- En el caso del modelo OSI, también se **asegura que lleguen correctamente** al **destino** de la comunicación.
- 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)



PDU
Segmento /
Datagrama

Modelo de referencia OSI

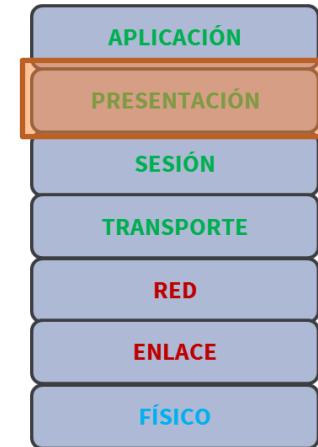
- La **CAPA DE SESIÓN** establece, gestiona y finaliza las **conexiones** entre usuarios (procesos o aplicaciones) finales.
- **Mantiene y controla el enlace establecido** entre dos computadoras que están transmitiendo datos.
- **Ofrece varios servicios** muy importantes para la comunicación, como:
 - 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.



PDU
SPDU

Modelo de referencia OSI

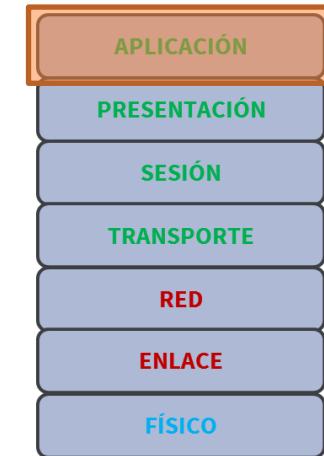
- La **CAPA DE PRESENTACIÓN** se encarga de la **representación de la información**, de manera que **aunque** distintos **equipos** puedan tener **diferentes representaciones internas** de caracteres (ASCII, Unicode, EBCDIC), números, sonido o imágenes, los **datos lleguen de manera reconocible** a otros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



PDU
PPDU

Modelo de referencia OSI

- La **CAPA DE APPLICACIÓN** 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.



PDU
APDU

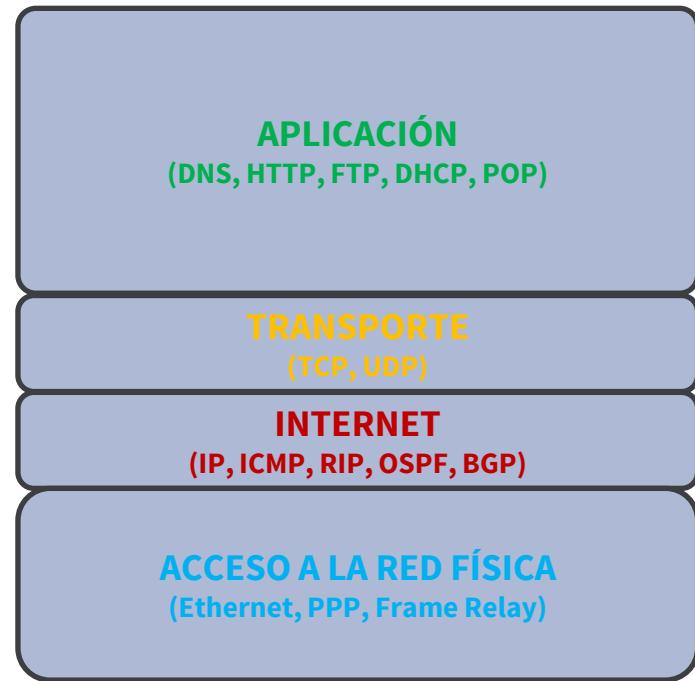
Modelo TCP/IP



MODELO OSI

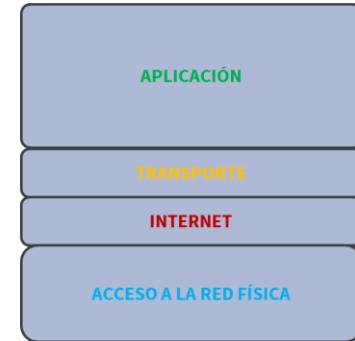


MODELO TCP/IP



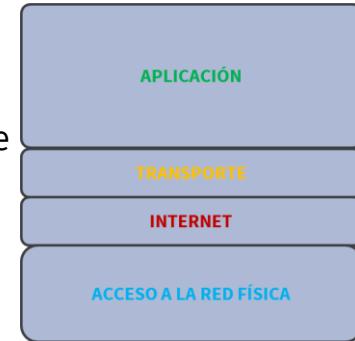
Modelo TCP/IP

- **TCP/IP** es el **protocolo** común **utilizado** por las **computadoras conectadas a Internet**, de manera que estas puedan comunicarse entre si.
- 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 es 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 mas importantes son el **TCP (Transmission Control Protocol)** y el **IP (Internet Protocol)**, que son los que dan nombre al conjunto de este modelo.



Modelo TCP/IP

- 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.



Organismos de estandarización de redes

- **IEEE (The Institute of Electrical and Electronics Engineers).**
- El Instituto de Ingenieros Eléctricos y Electrónicos 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**, como **ingenieros en telecomunicación, ingenieros en electrónica, ingenieros en informática e ingenieros en computación**.



Organismos de estandarización de redes

- **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**.



I E T F®

TEMA 1. Introducción

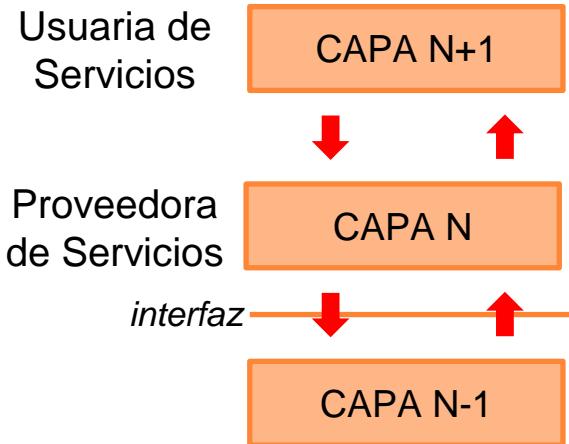
- 1.1. Sistemas de comunicación y redes.
- 1.2. Diseño y estandarización de redes.
- **1.3. Terminología y servicios.**
- **1.4. Internet: Arquitectura y direccionamiento.**
- **1.5. Cuestiones y ejercicios.**

Comunicación OSI

- Cada capa tiene **tareas bien definidas**.
- La **comunicación** se realiza **entre dos capas adyacentes**: (N) y (N+1)
- Capa **inferior** → **Proveedora** de servicios
- Capa **superior** → **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 la componen los mecanismos que permiten interaccionar a dos capas adyacentes



Comunicación OSI

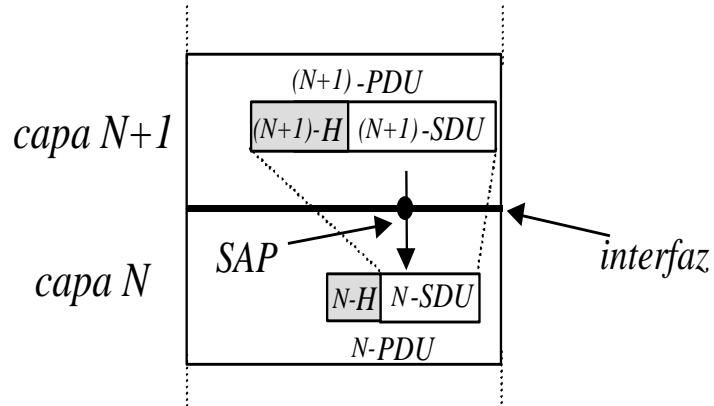
- Los **elementos** activos (**HW y SW**) de la **capa N**, reciben el nombre de **entidades de nivel N**.
- Las **entidades de nivel N en el emisor y receptor** reciben el nombre de **entidades pares o paritarias**.
- 2 Tipos de comunicación:
 - **Comunicación Real o Vertical**: intercambio de datos 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.

Comunicación OSI

- **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**

Comunicación OSI

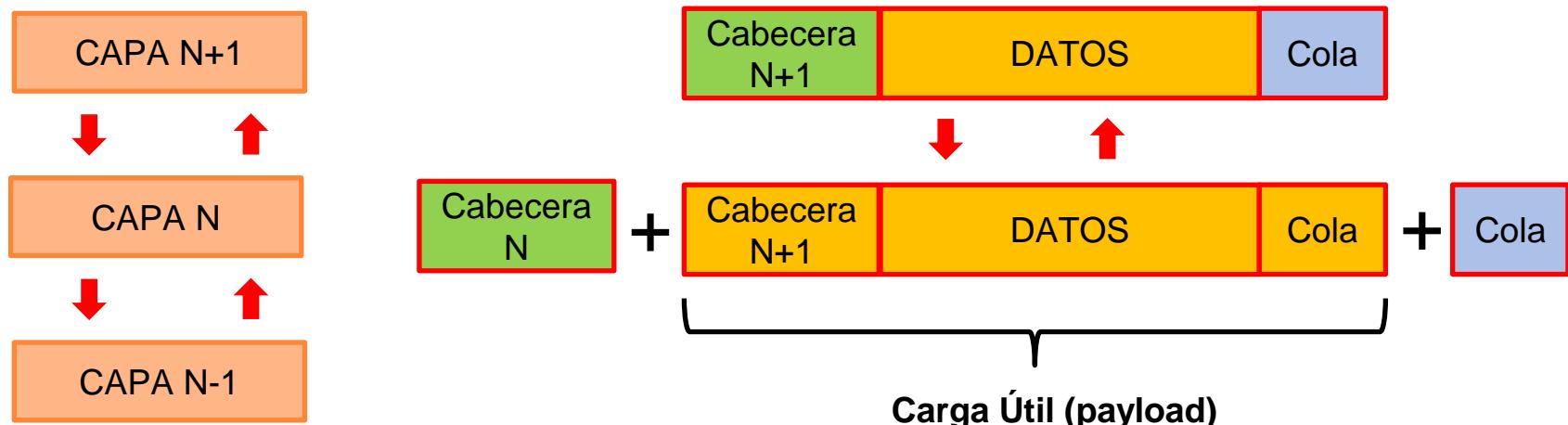
- La **comunicación** producida **entre capas adyacentes** se realiza **a través de una interfaz** de separación → **Punto de acceso al servicio** (*Service Access Point*, SAP).
- Información transmitida sobre los SAP entre 2 entidades:
- **Unidad de datos de servicio** (*Service Data Unit*, SDU) → **Datos** 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**.



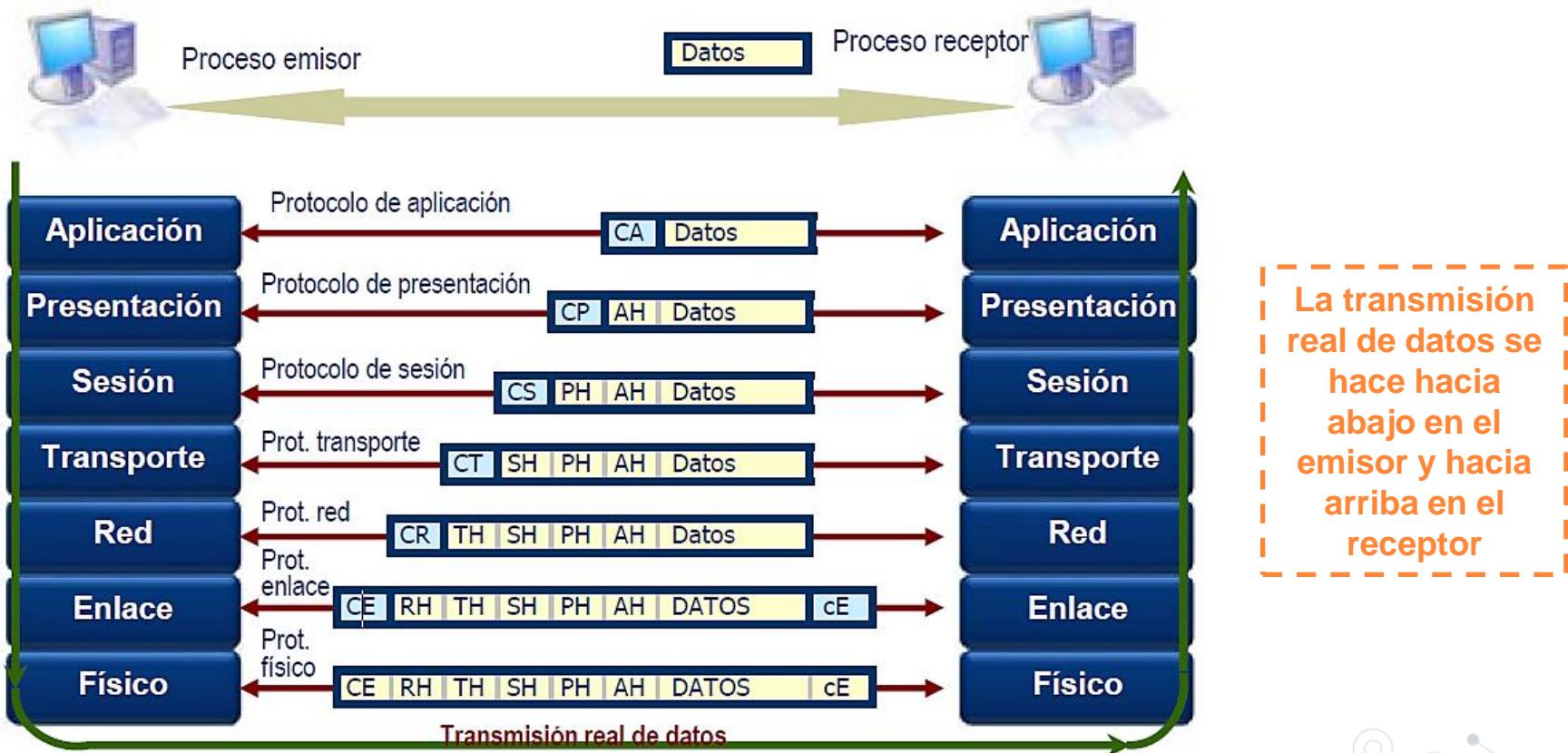
Comunicación OSI

- A excepción de la capa física, el resto de **capas añaden/eliminan información complementaria (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 cabecera/cola se convierte en los Datos de la inferior



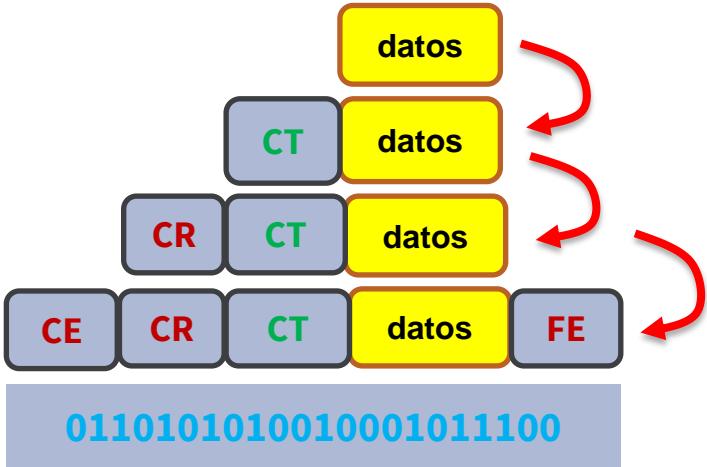
Comunicación OSI



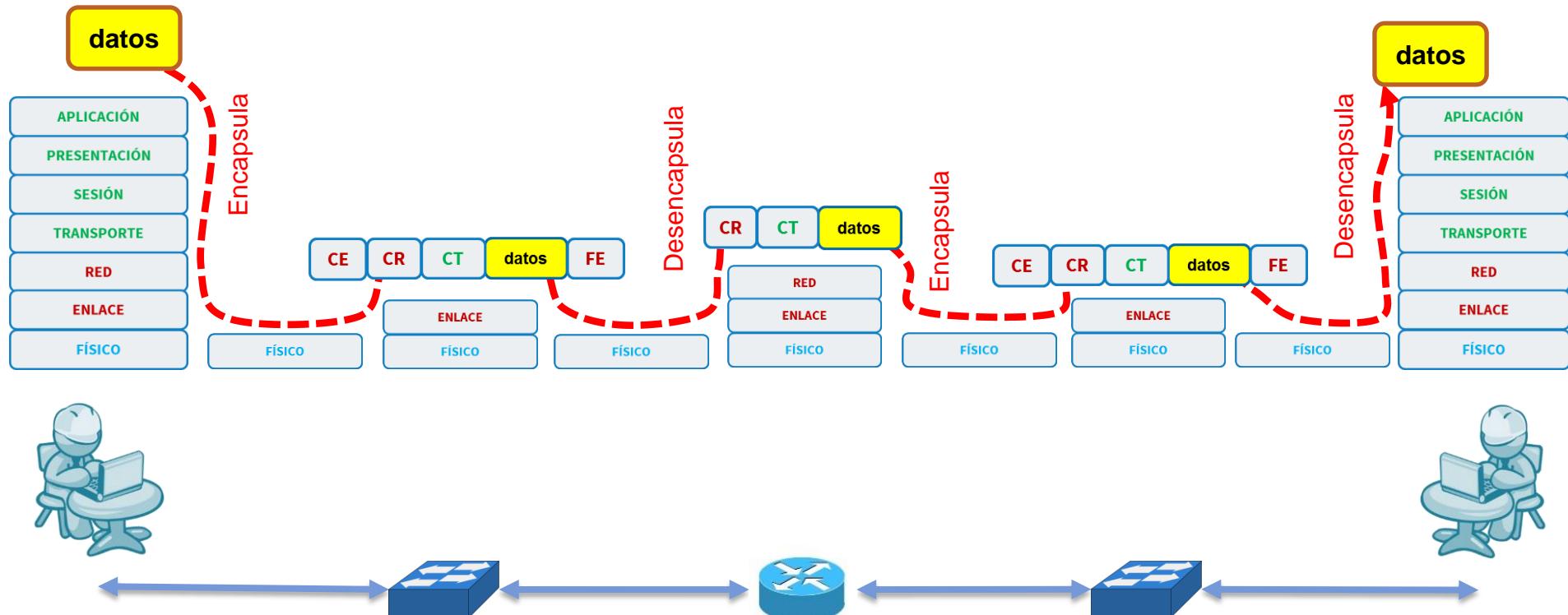
Comunicación OSI



Encapsulamiento



Comunicación OSI



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

MTU (Maximum Transfer Unit)

- Cada tecnología tiene un tamaño máximo de tramas que puede transmitir → MTU.
- 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 retardo)	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	8166
Token Ring 16 Mb/s	17940 (token holding time 8 ms)
Token Ring 4 Mb/s	4440 (token holding time 8 ms)
FDDI	4352
Hyperchannel	65535
Classical IP over ATM	9180

MTU (Maximum Transfer Unit)

MTU Grande

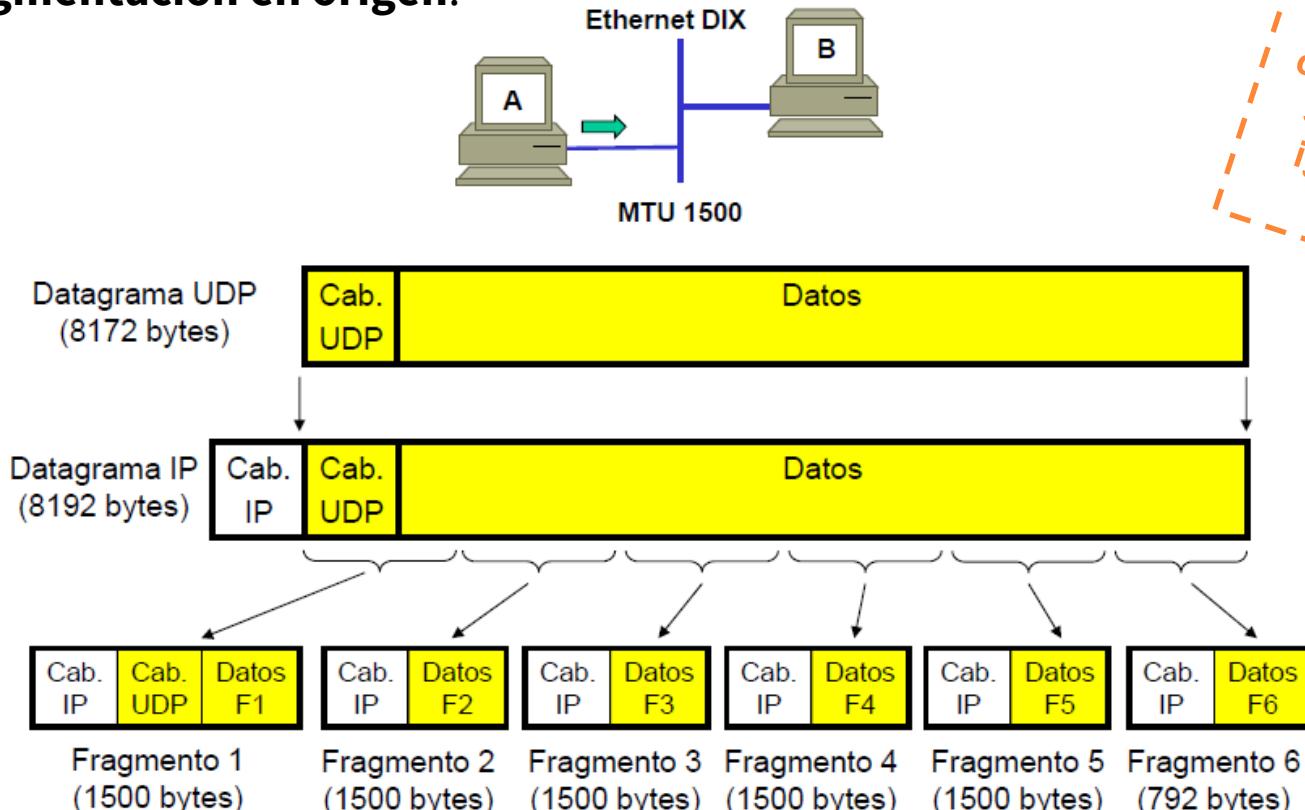
- Ventajas:
 - Mejora en la eficiencia de comunicación y reduce la sobrecarga en la red (menor ancho de banda (BW) en el envío de cabeceras).
 - Reduce la carga de CPUs de los dispositivos, porque procesan menos paquetes.
- Inconvenientes:
 - Mayores buffers (para almacenar los paquetes recibidos antes 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.
 - **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.

Fragmentación

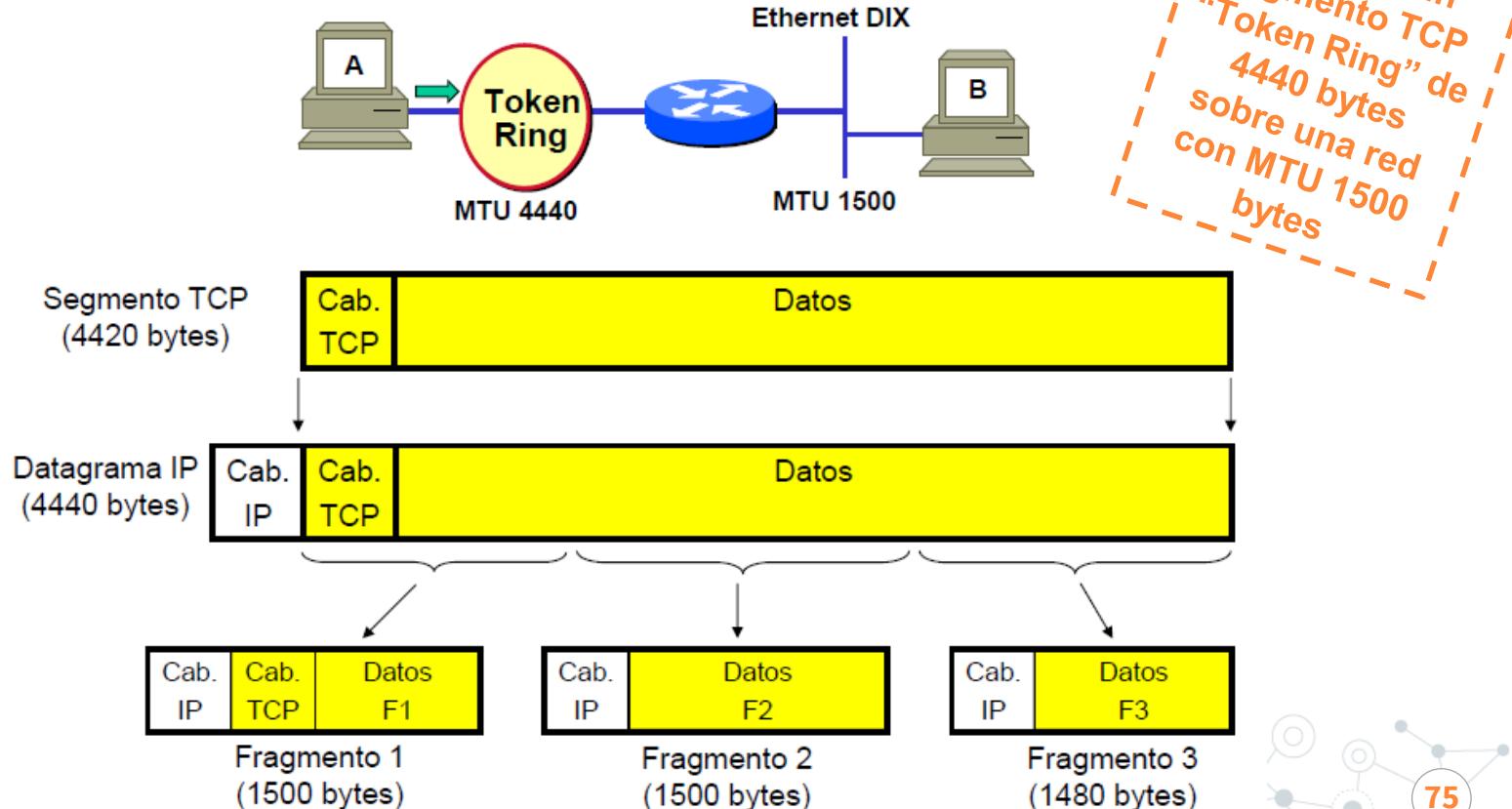
- Fragmentación en origen.



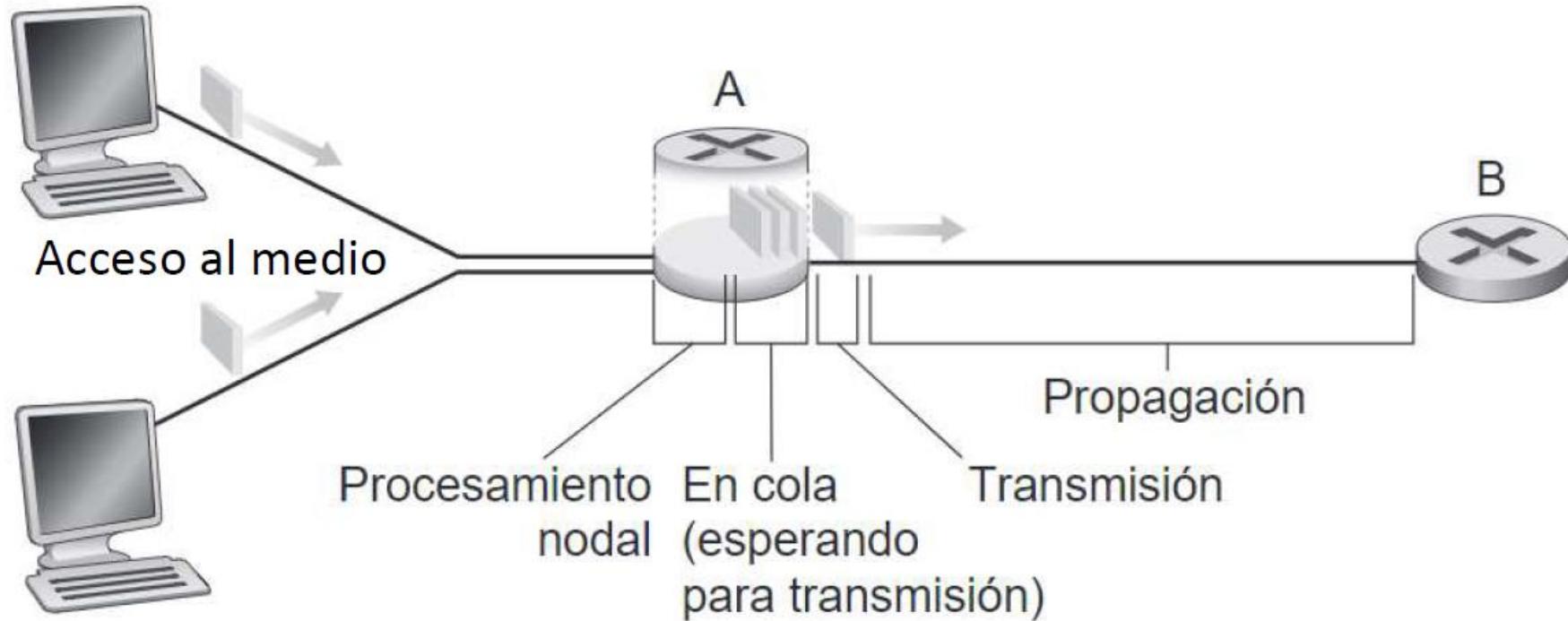
El tamaño de los fragmentos (incluyendo la cabecera) debe ser menor (o igual) que la MTU

Fragmentación

- Fragmentación en ruta.



Retardos en la comunicación



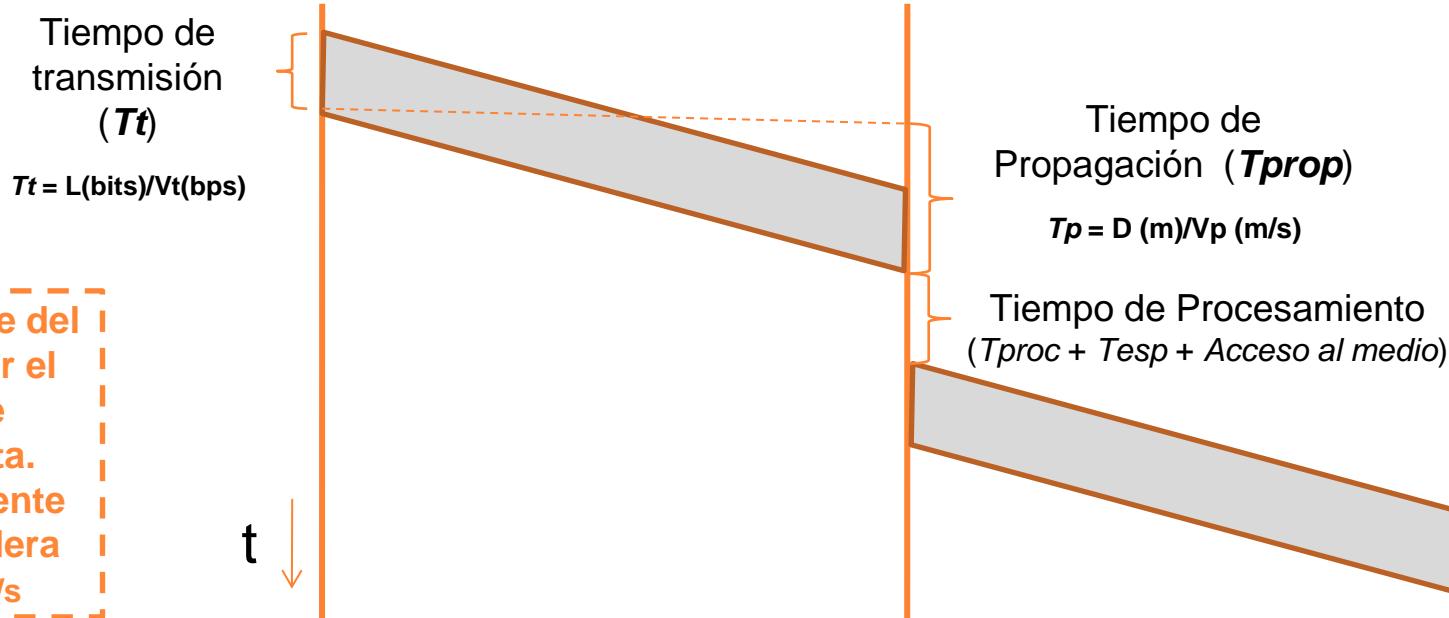
Retardos en la comunicación

- Tiempo de Propagación al siguiente nodo/host (**T_{prop}**).
Depende de la distancia y del medio de transmisión (la velocidad que pueda ofrecer).
- Tiempo de procesamiento en los nodos(**T_{proc}**).
Tiempo que se tarda en decidir qué hacer con el paquete (desencapsular e interpretar).
Depende del tipo de nodo/router y de su carga.
- Tiempo de espera en la cola salida (**T_{esp}**).
Depende del tráfico en la red.
- Tiempo de transmisión (**T_t**).
Depende de la velocidad del enlace y tamaño del paquete.

$$T_{\text{prop}} = \frac{D \text{ (Distancia a Recorrer)}}{V_p \text{ (Velocidad Propagación)}}$$
$$T_t = \frac{L \text{ (Longitud del Paquete)}}{V_t \text{ (Velocidad Transmisión)}}$$

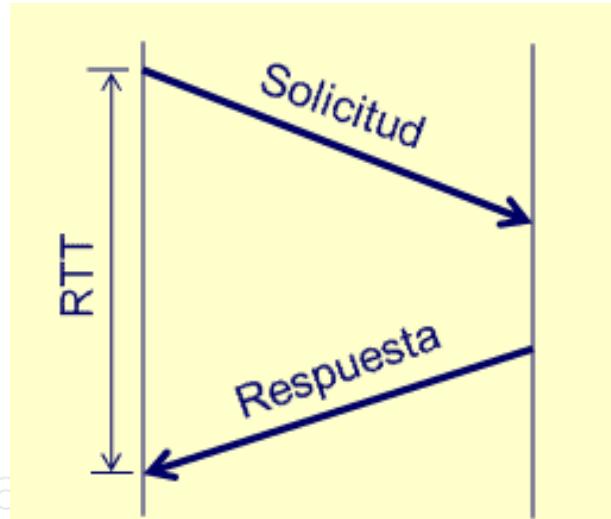


Retardos en la comunicación



Retardos en la comunicación

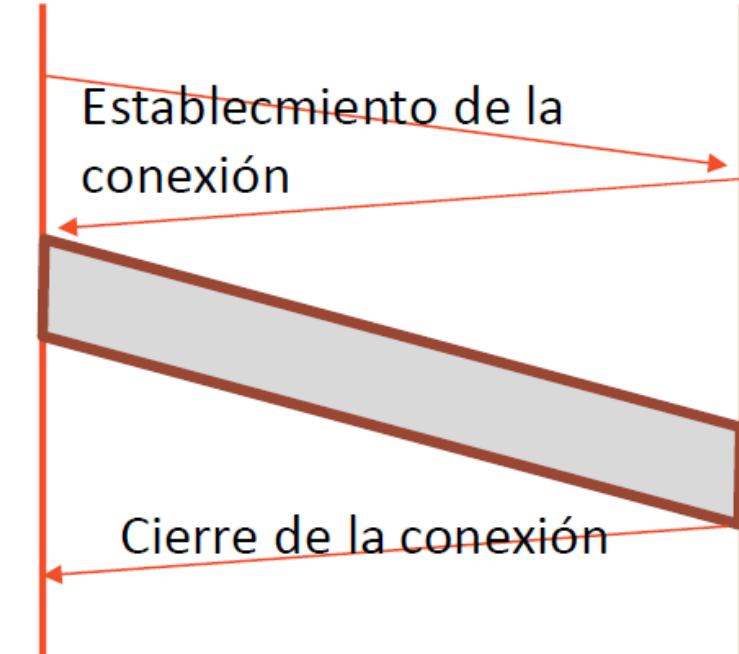
- **Round Trip Time (RTT)**: Tiempo para enviar un paquete y recibir su respuesta asociada.
- Está constituido por la **suma de los retardos de cada uno de los enlaces utilizados** (ida y vuelta) y el **tiempo de proceso en el servidor**.



Tipos de servicios

- En cada capa los servicios pueden ser de dos tipos:
 - 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)

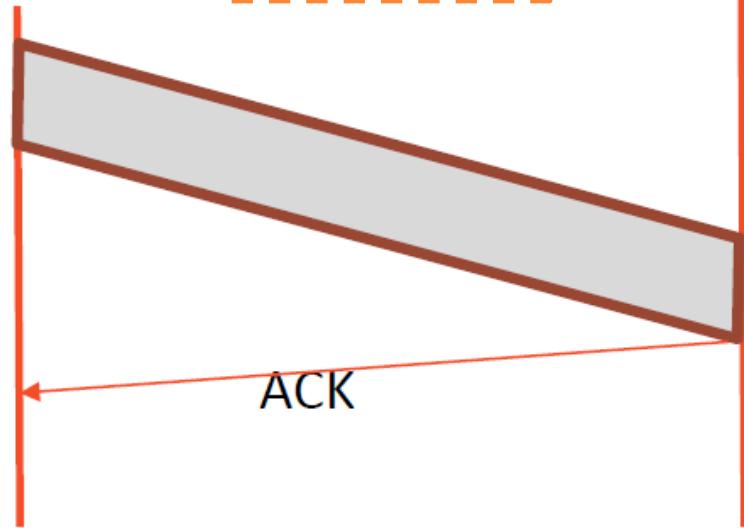
SOC



Tipos de servicios

- Además, los servicios pueden ser:
 - **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)

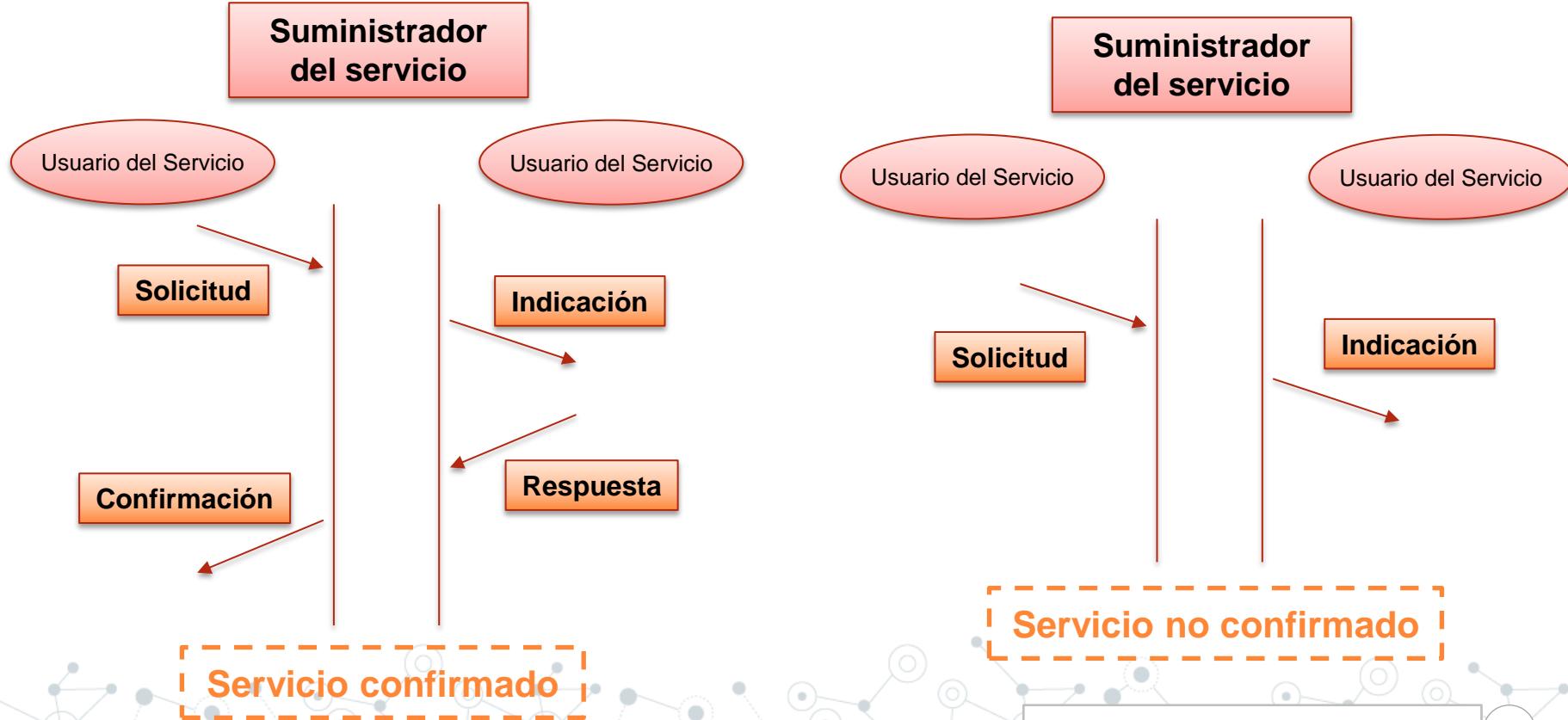
Servicio
confirmado



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.
- Primitivas:
 - 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.

Especificación de un servicio



TEMA 1. Introducción

- 1.1. Sistemas de comunicación y redes.
- 1.2. Diseño y estandarización de redes.
- 1.3. Terminología y servicios.
- **1.4. Internet: Arquitectura y direccionamiento.**
- 1.5. Cuestiones y ejercicios.

Internet

- Internet se puede considerar **la mayor red de comunicaciones del planeta**, a menudo denominada “**la red de redes**” y formada por la **interconexión** de miles o incluso **millones de redes** de todo el mundo.

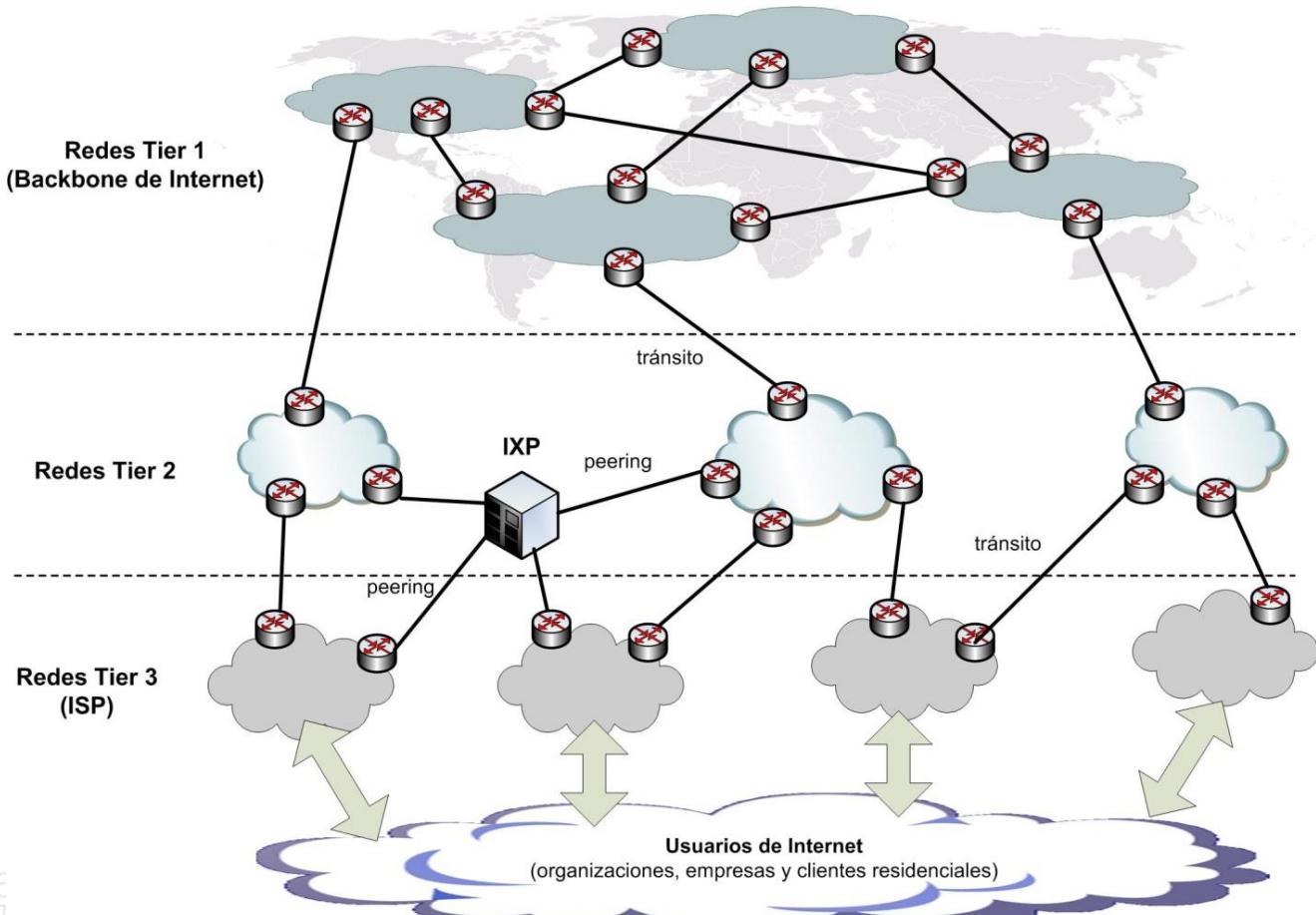


Internet

- Históricamente se reconoce que su **preursora fue ARPANET**, usada para conectar tres universidades en Estados Unidos y que después evolucionó a red de defensa.
- Se suele denominar la **WorldWide Web** (o la web), aunque es una confusión, puesto que “www” es un conjunto de protocolos que funcionan sobre Internet.
- Posteriormente **Internet creció en torno** a una red llamada **NSFNET**, que hacía las funciones de **red troncal**, es decir, una red que servía para unir al resto de las redes.
- Cuando un organismo o empresa quería conectarse a Internet, tenía que establecer un enlace con NSFNET.
- Esta red troncal pertenecía a una institución norteamericana llamada **NSF (National Science Foundation)**.
- En 1995, **NSF cedió la función de red troncal** a cuatro grandes operadoras comerciales norteamericanas y comenzó la **descentralización de Internet**.

Topología

- 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.



Topología

- Redes **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.
- Redes **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.
- Redes **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.

Conexiones entre operadores

La conexión entre las redes de diferentes operadores se puede hacer de dos formas:

- **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.

Las conexiones
son acuerdos
entre las
operadoras

Conexiones entre operadores

La conexión entre las redes de diferentes operadores se puede hacer de dos formas:

- **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 (ver el siguiente apartado)
 - Privados: conexión directa entre los dos proveedores.

Las conexiones son acuerdos entre las operadoras

Conexiones entre operadores

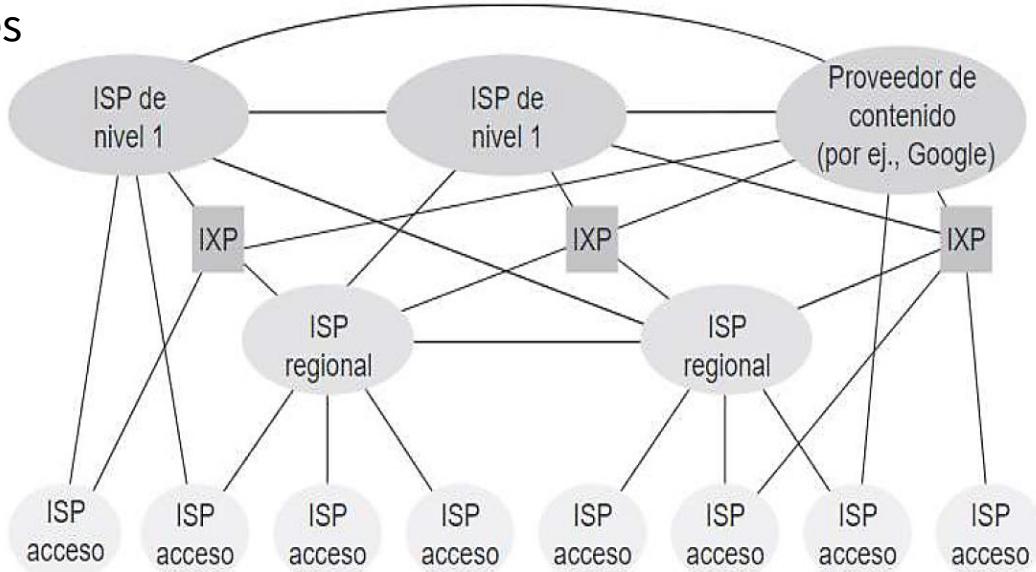
Existe un **tipo de conexión** entre operadores llamada **IXP** (*Internet eXchange Point*) o Punto de intercambio de tráfico de Internet.

- 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.

(<https://www.euro-ix.net/en/about-us/members/>)

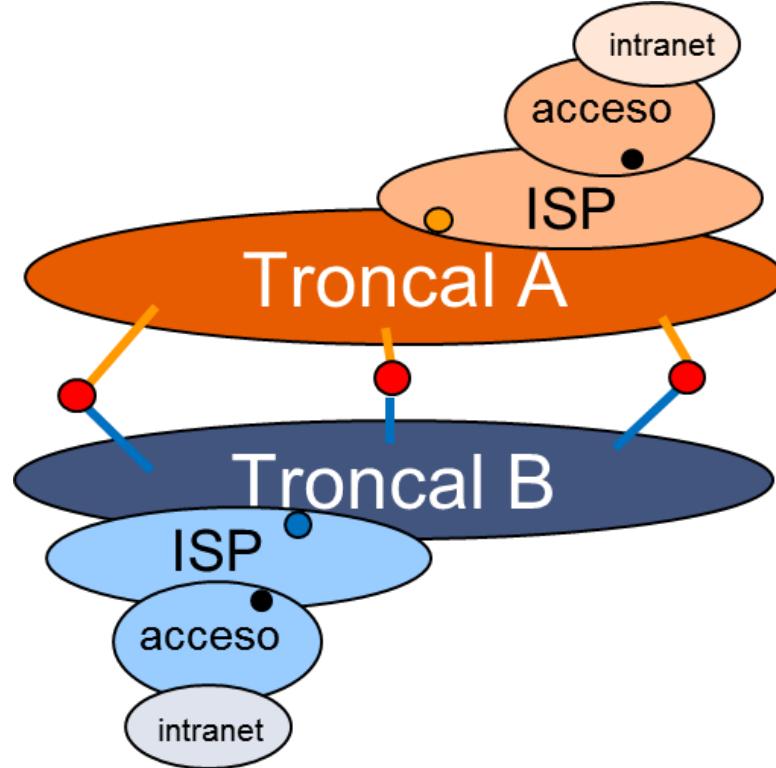
Topología jerárquica

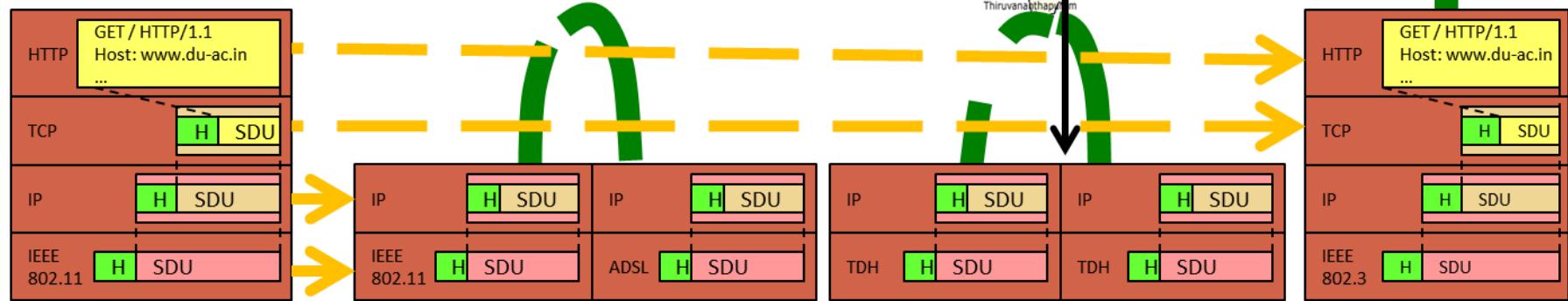
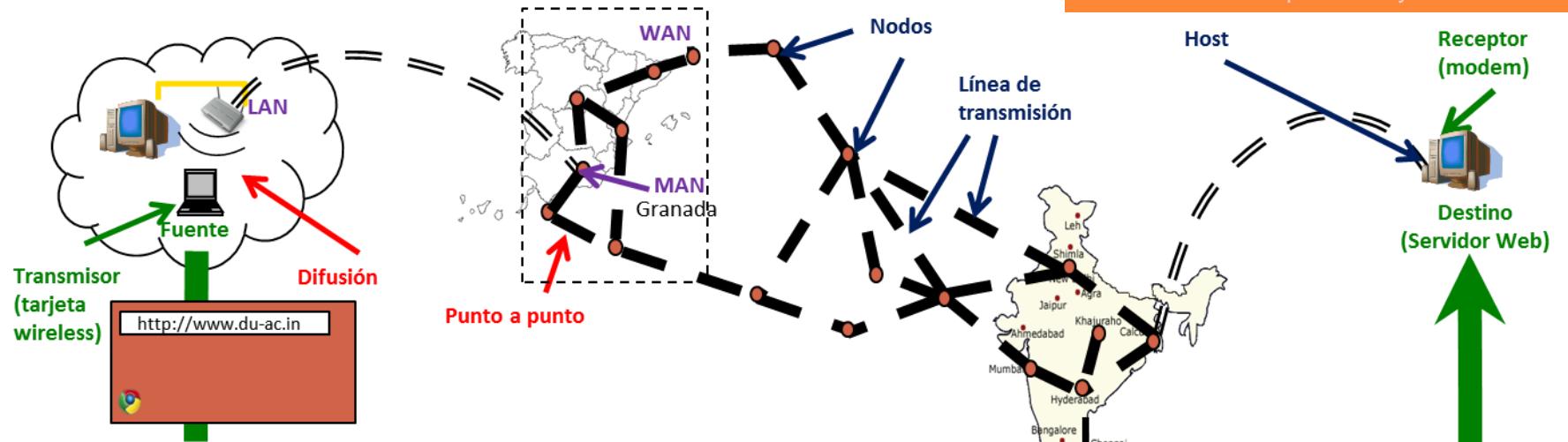
- **Redes troncales** (ATM, SDH, SONET, etc) de grandes operadores de telecomunicaciones (ISP de nivel 1).
- **Redes de acceso** (xDSL, RDSI, FTTH, etc) del ISP
- **Intranets** (Ethernet) del usuario: zona pública + zona privada



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** \Leftrightarrow **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).

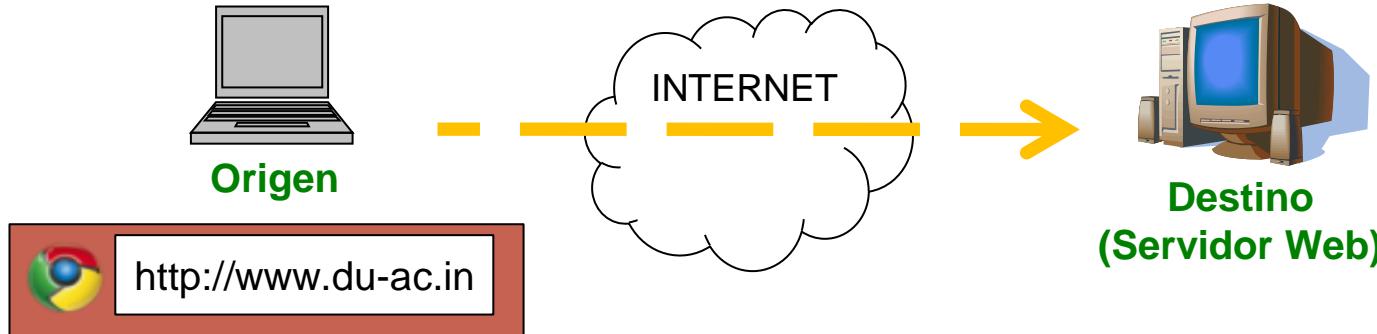
Direccionamiento

- 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.
- En la actualidad, conviven **dos versiones del protocolo IP: IPv4** y la nueva versión **IPv6**.
- 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.

Direccionamiento

- 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.
 - **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.

Direccionamiento



- **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

Ejemplo



RedIRIS (www.rediris.es)

- Red académica e investigación RedIRIS es la Gran Instalación Telemática del Plan Nacional de I+D+i, creada para potenciar los resultados de la investigación española.
 - Es una Red de datos para facilitar el desarrollo científico.
 - Es una herramienta de colaboración para los investigadores.
 - Es un elemento básico para experimentos científicos.
 - Es un banco de pruebas de nuevas tecnologías y servicios.
 - Es un elemento de ciertos instrumentos científicos.
 - Es una ayuda para impulsar la Sociedad de la Información.
- RedIRIS ofrece sus servicios a más de 350 instituciones (incluyendo todas las universidades españolas y la mayoría de los centros de investigación públicos). Esto incluye más de 150.000 investigadores y más de 2.000.000 de usuarios potenciales.

Ejemplo



Red IRIS (www.rediris.es)

Puntos de
Presencia (PoP) son
lugares físicos
donde están los
equipos de un ISP



TEMA 1. Introducción

- 1.1. Sistemas de comunicación y redes.
- 1.2. Diseño y estandarización de redes.
- 1.3. Terminología y servicios.
- 1.4. Internet: Arquitectura y direccionamiento.
- **1.5. Cuestiones y ejercicios.**

Ejercicios

- Boletín de ejercicios resueltos Tema1 (Prado).
- Cuestiones y ejercicios del Capítulo 1 de Kurose, Ross.
- Cuestiones y ejercicios del Capítulo 1 de García-Teodoro, Díaz-Verdejo, López-Soler.

Entonces... ¿Tenemos ya delegad@?

Para que sea el/la intermediario/a para la comunicación entre la clase y los profesores de la asignatura.



¿Preguntas?

O comentarios, sugerencias, inquietudes



Fundamentos de Redes

Tema 2.

Servicios y Protocolos de Aplicación en Internet

Antonio M. Mora García



Bibliografía

Básica

- James F. Kurose, Keith W. Ross. Redes de computadoras. Un enfoque descendente. 7º Edición. Editorial Pearson S.A., 2017.

CAPÍTULO 2 (2.1, 2.2, 2.4, 2.5)

- P. García-Teodoro, J.E. Díaz-Verdejo, J.M. López-Soler. Transmisión de datos y redes de computadores, 2ª Edición. Editorial Pearson, 2014. **CAPÍTULOS 11 y 12.3**



Complementaria

- James F. Kurose, Keith W. Ross. Redes de computadoras. Un enfoque descendente. 7º Edición. Editorial Pearson S.A., 2017.

CAPÍTULOS 7 y 8



Índice

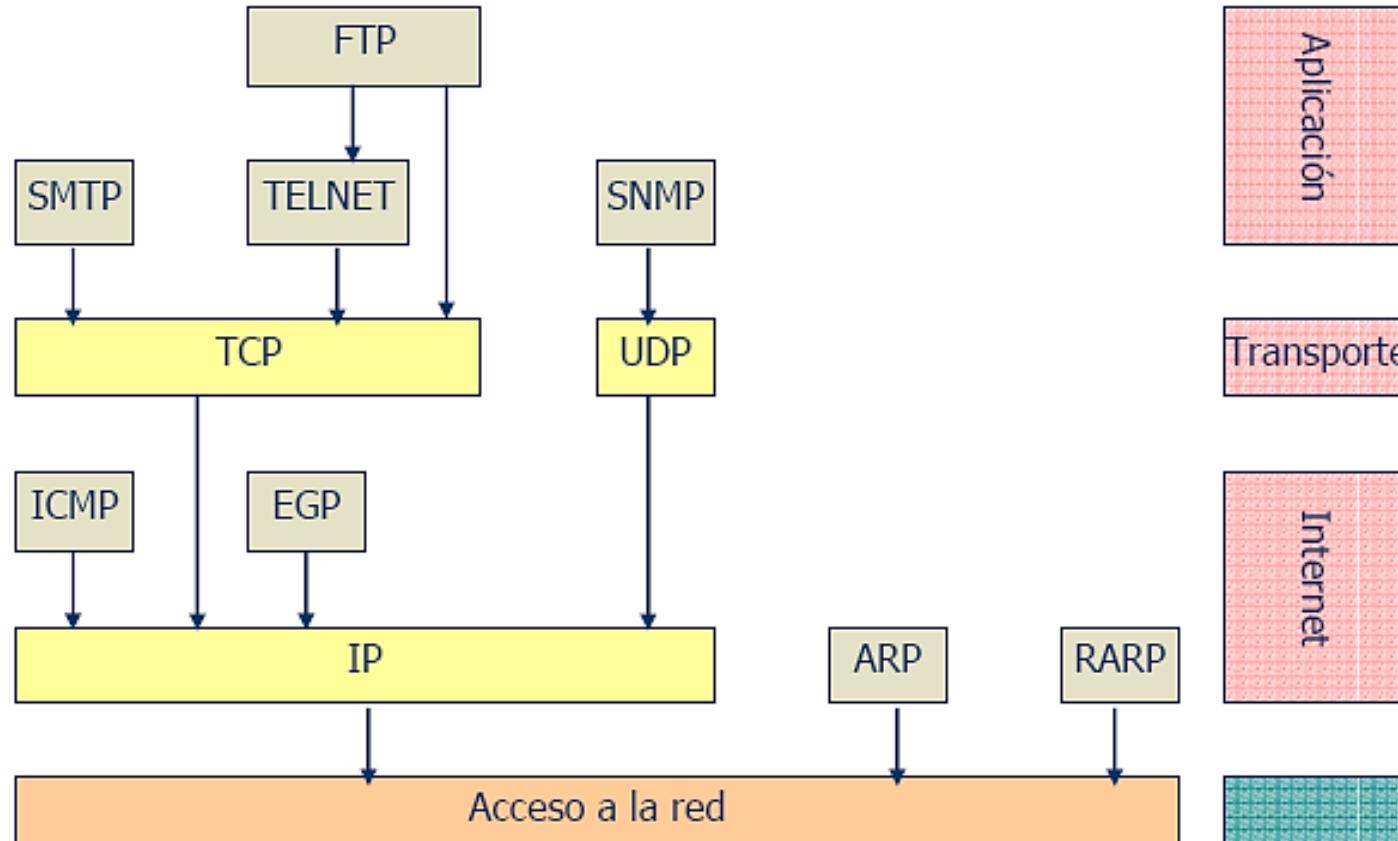
- **2.1.** Introducción a las aplicaciones de red
- **2.2.** Servicio de Nombres de Dominio (DNS)
- **2.3.** Navegación web
- **2.4.** Correo electrónico
- **2.5.** Protocolos seguros
- **2.6.** Aplicaciones multimedia
- **2.7.** Aplicaciones para interconectividad de redes locales
- **2.8.** Cuestiones y ejercicios

TEMA 2. Servicios y Protocolos en Internet

2.1. Introducción a las aplicaciones de red

- 2.2. Servicio de Nombres de Dominio (DNS)
- 2.3. Navegación web
- 2.4. Correo electrónico
- 2.5. Protocolos seguros
- 2.6. Aplicaciones multimedia
- 2.7. Aplicaciones para interconectividad de redes locales
- 2.8. Cuestiones y ejercicios

Estructura de protocolos



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.

Protocolos TCP/IP

CAPA DE RED

- La **base** de la familia de protocolos es el **nivel de Red (IP, Internet Protocol)**.
- Es un **protocolo de conmutación 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

Protocolos TCP/IP

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 o vídeo

Protocolos TCP/IP

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.
- 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.
- **Cliente y servidor** no tienen que estar necesariamente en computadoras separadas, sino que **pueden ser programas** diferentes que se ejecuten **en la misma máquina**.

Arquitectura Cliente/Servidor

VENTAJAS

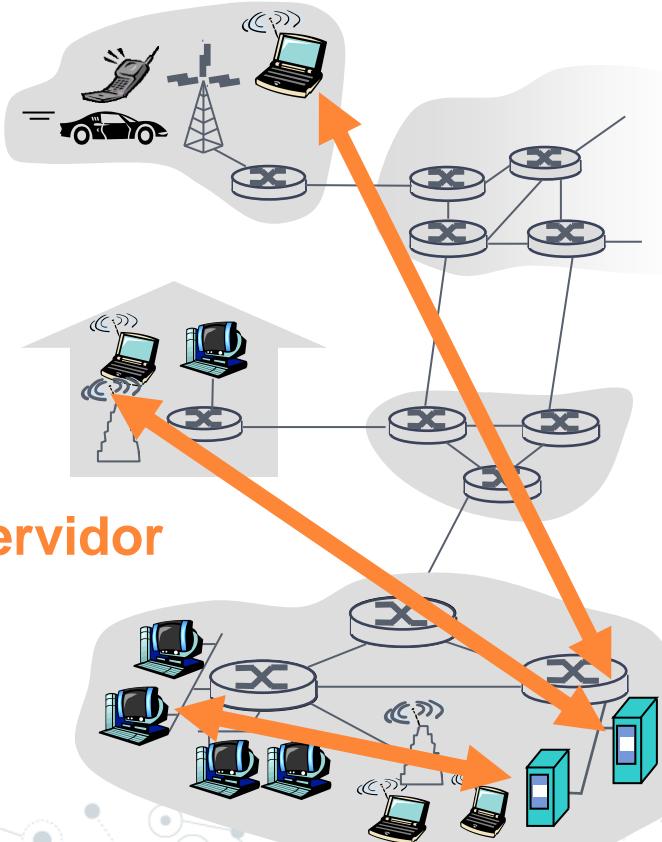
- **Recursos centralizados:** debido a que el servidor suele ser el centro de la red, puede administrar los recursos que son comunes a todos los usuarios (clientes).
- **Seguridad mejorada:** la cantidad de puntos de entrada que permite el acceso a los datos no es importante. El servidor garantizará 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.

Arquitectura Cliente/Servidor

DESVENTAJAS

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

Arquitectura Cliente/Servidor



Servidor

- Siempre en funcionamiento
- IP permanente y pública
- Agrupados en “granjas”
- Pueden comunicarse entre sí para optimizar el servicio

Clients

- Funcionando intermitentemente
- Pueden tener IP dinámica y privada
- Se comunican con el servidor
- No se comunican entre sí en relación a un servicio

Procesos cliente y servidor

Proceso Cliente: proceso que inicia la comunicación

Proceso Servidor: proceso que espera ser contactado

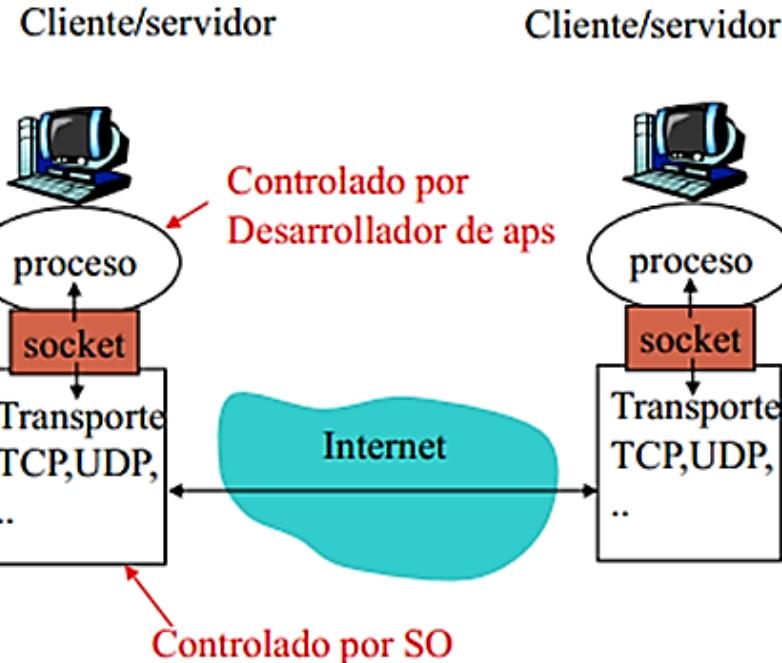
- Un proceso envía/recibe mensajes a/desde un **socket**.
- Cada proceso debe tener un **identificador** compuesto por su **dirección IP + número de puerto**

Ejemplo:

servidor web *gaia.cs.umass.edu*

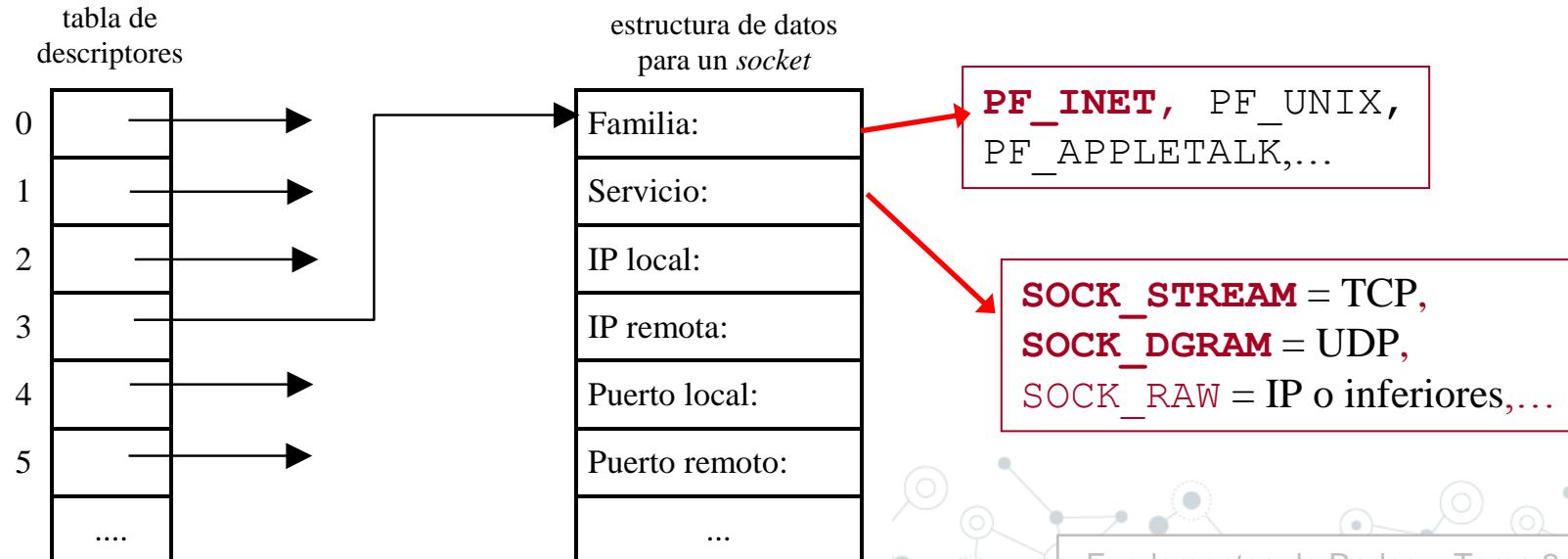
Dirección IP: 128.119.245.12

Núm. Puerto: 80



Interfaz Socket

- 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:



Interfaz Socket

PROPIEDADES (Según el tipo de socket)

- 1) La **fiabilidad de la transmisión**: ningún dato transmitido se pierde.
- 2) La conservación del **orden de los datos**: los datos llegan en el orden en el que han sido emitidos.
- 3) La **no duplicación de datos**: sólo llega a destino un ejemplar de cada dato emitido.
- 4) 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.
- 5) **Delimitación de los mensajes**: los límites de los mensajes emitidos se pueden encontrar en el destino.
- 6) El envío de **mensajes (urgentes)**: posibilidad de enviar datos fuera del flujo normal, accesibles inmediatamente.

Interfaz Socket

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

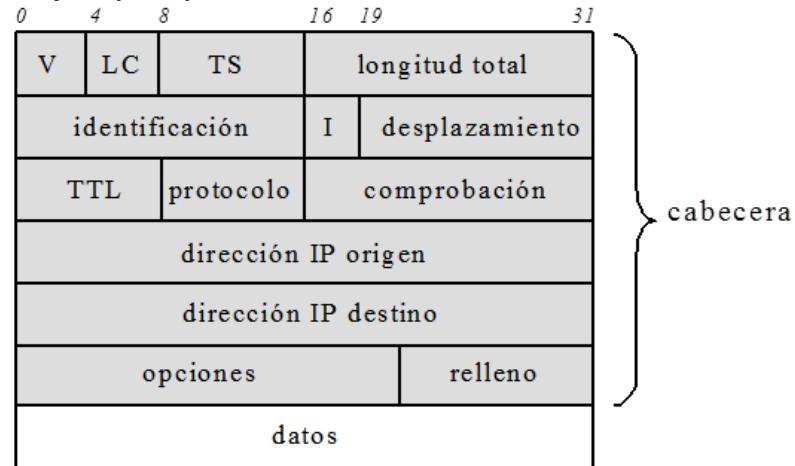
- **La semántica**

- Significado de los “campos”

- **Las reglas**

- Cuándo los procesos envían mensajes/responden a mensajes

Ejemplo: protocolo IP



Protocolos

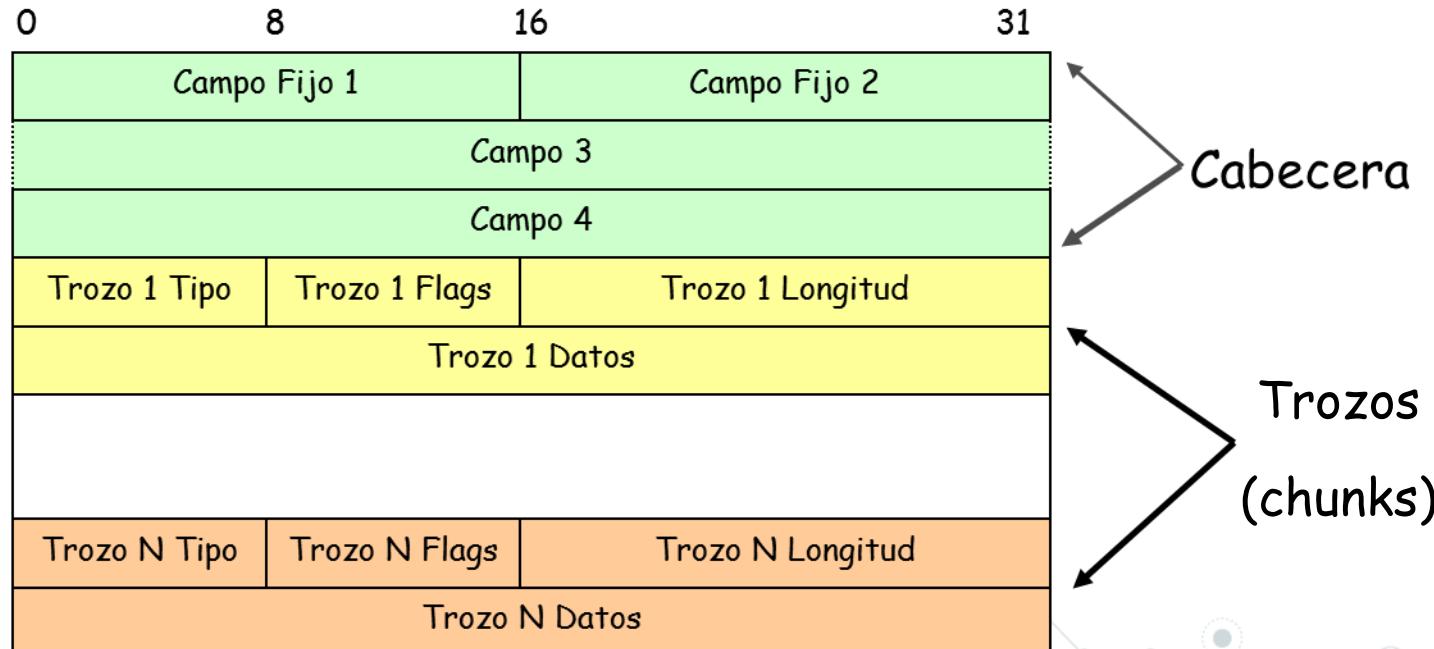
TIPOS DE PROTOCOLOS

- 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:** (Llamado “Urgent Data” en TCP) Útil para la separación de dos tipos diferentes de datos. No significa que será entregado más rápido o con mayor prioridad que los datos en el flujo de datos in-band.
(Ej: Protocolo FTP, puertos 20 (Datos) y 21 (Control))
- 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.

Protocolos

TENDENCIA: PROTOCOLOS FLEXIBLES

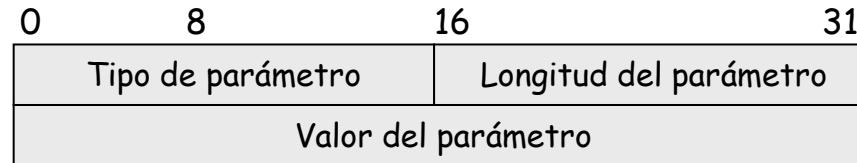
- Una cabecera fija.
- Uno o varios “trozos” (obligatorios u opcionales).



Protocolos

TENDENCIA: PROTOCOLOS FLEXIBLES

- Una cabecera fija.
- 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)



Protocolos

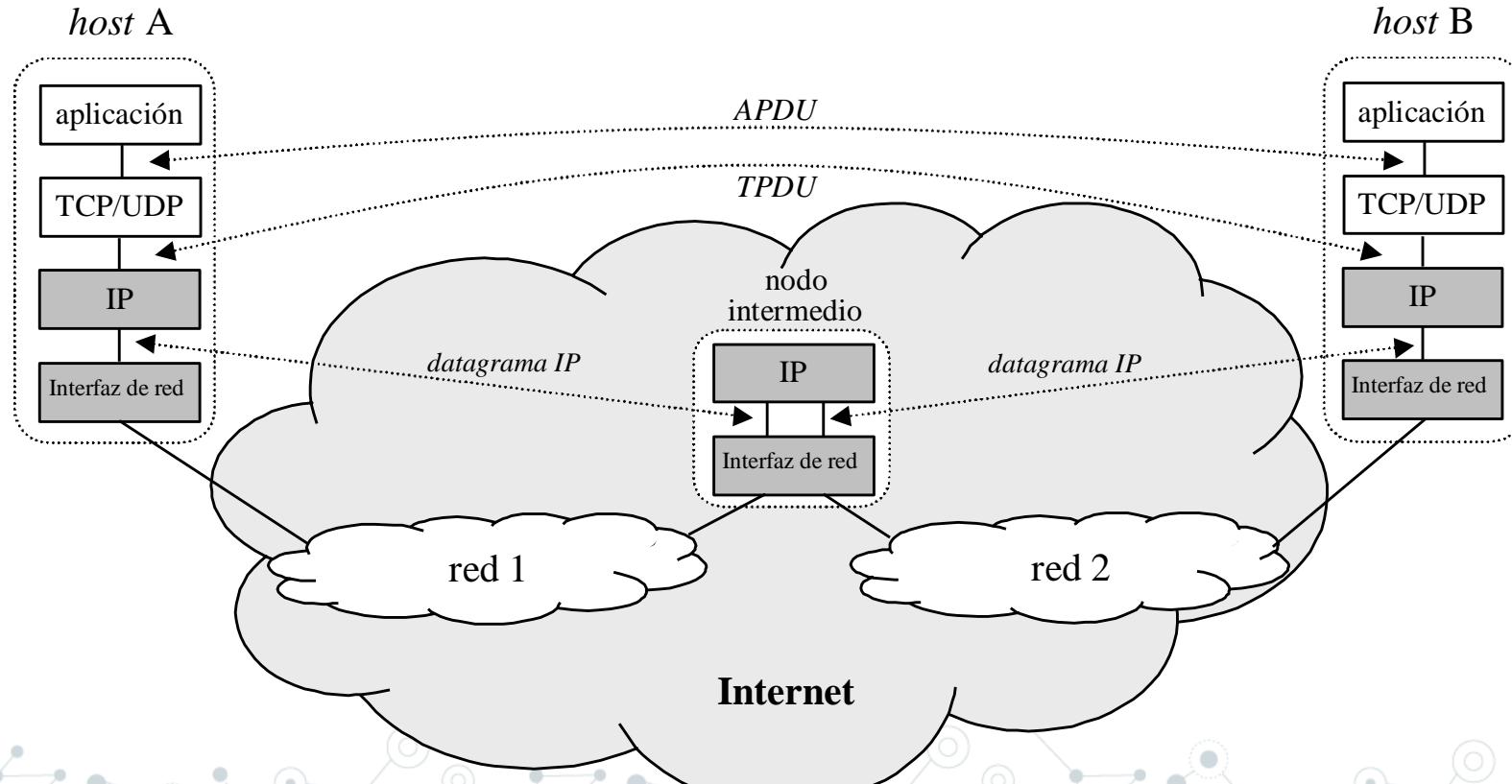
PROPIEDADES

- **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, ...

Protocolos

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

Protocolos



Protocolos

Application	Application layer protocol	Underlying transport protocol
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

TEMA 2. Servicios y Protocolos en Internet

- 2.1. Introducción a las aplicaciones de red
- **2.2. Servicio de Nombres de Dominio (DNS)**
- **2.3. Navegación web**
- **2.4. Correo electrónico**
- **2.5. Protocolos seguros**
- **2.6. Aplicaciones multimedia**
- **2.7. Aplicaciones para interconectividad de redes locales**
- **2.8. Cuestiones y ejercicios**

Introducción

- La comunicación en Internet **precisa de direcciones IP**.
- Las **direcciones IP son difíciles de memorizar** o recordar
- Necesitamos asignar a dichas direcciones nombres significativos conocidos como nombres de dominio.

www.ugr.es <-----> 150.214.204.25

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

Introducción

EN LOS COMIENZOS 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.

Introducción

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 maquina 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

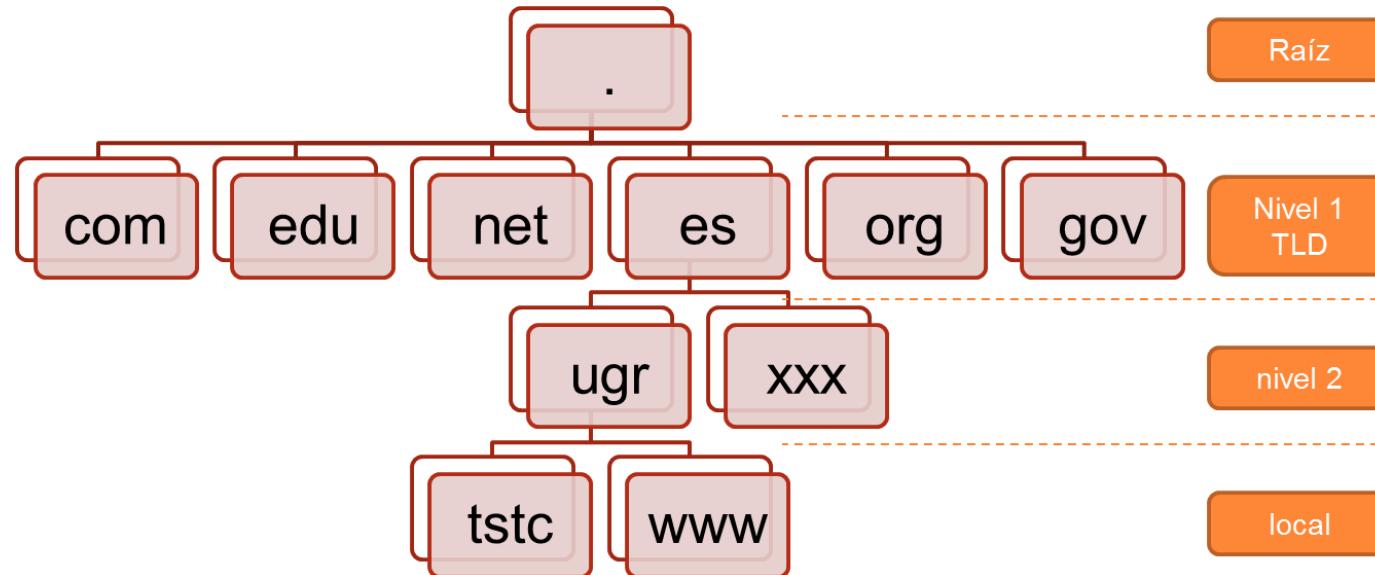


- **Sistema global:** identifica únicamente en todo Internet.
 - ICANN (*Internet Corporation for Assigned Names and Numbers*). Es la autoridad central que controla la entrada de cualquier sistema nuevo. www.icann.org
 - NIC (*Network Information Center*). Organizaciones que permiten descentralizar esas 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.



Sistema de Nombres de Dominio

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.

Sistema de Nombres de Dominio

SINTAXIS DEL NOMBRE DE DOMINIO

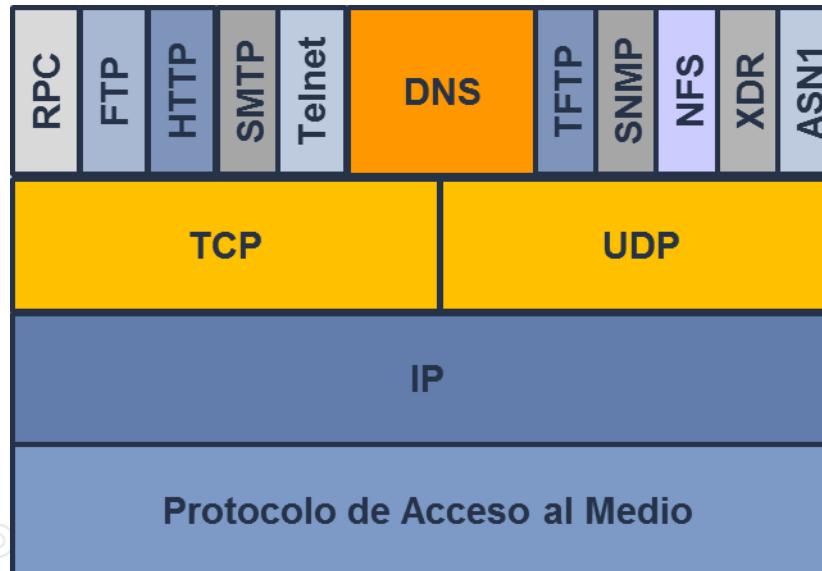
- Cadena de **hasta 255 caracteres**, formada por **etiquetas separadas por puntos** (long. etiqueta < 64 caracteres) que **indican un nivel en la jerarquía**.
- No se distinguen mayúsculas y minúsculas.
- Tipos de nombres de dominio:
 - **Absolutos**: terminados con “.” (Ej: “ugr.es.”)
 - **Relativos**: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

Sistema de Nombres de Dominio

- 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); jp (Japón)...

Servicio de Nombres de Dominio

- 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

Servidores DNS

- 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.

Servidores Raíz

(13) Root-Servers <http://www.root-servers.org/>

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 Ejercito, Maryland, USA.

Servidor I: NORDUnet, Estocolmo, Suecia.

Servidor J: (TBD), Virginia, USA.

Servidor K: RIPE-NCC, Londres, Inglaterra.

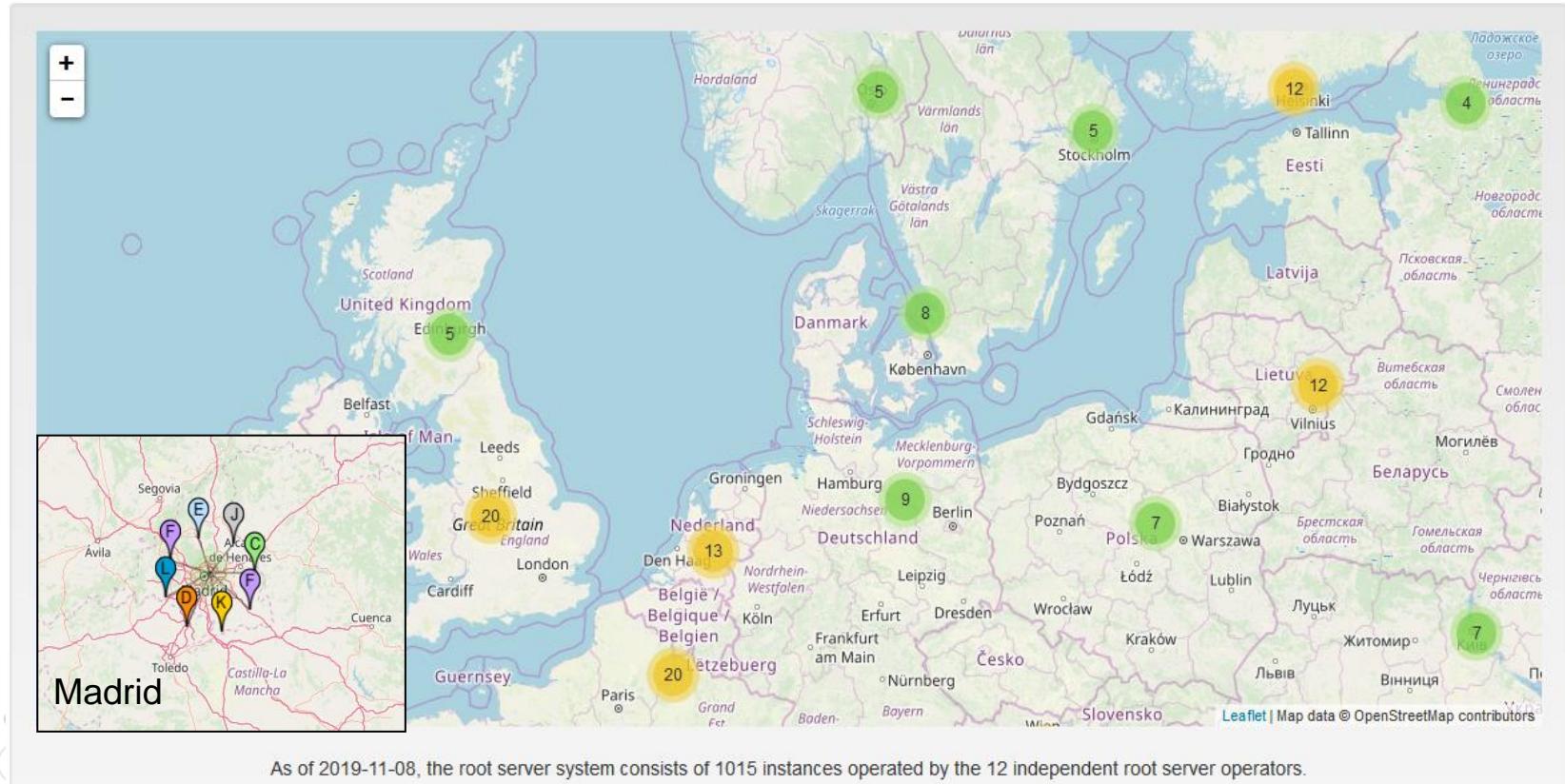
Servidor L: (TBD), California, USA.

Servidor M: Wide Project, Universidad de Tokyo, Japón.



Servidores Raíz

Gestionados por organismos públicos o compañías



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) y/o **ISP**.
- Cada país dispone de **autoridades de registro**.

Delegación de la autoridad

InterNIC

- Home
- Registrars
- Whois
- F

InterNIC—Public Information Regarding Internet Domain Name Registration Services

Do you have a complaint or dispute?

Your Registrar or Domain Name:

- Domain Name Transfer Dispute
- Unsolicited Renewal or Transfer Solicitation
- Your Registrar is Not on the Accredited List
- Unauthorized Transfer of Your Domain Name
- Trademark Infringement
- Registrar Services Dispute
 - Failure to answer phones or respond to email messages
 - Financial Transaction Issues
- Uniform Domain Name Dispute Resolution (UDRP) Intake Report System

Inaccurate Whois Information

Spam or Viruses

IP Address Issues

Content on a Website

Information about Registrars

- Search Accredited Registrar Directory
 - Alphabetical List
 - List by Location
 - List by Language Supported
- Have a Problem with a Registrar?
 - Complaint Form
 - Helpful Hints

Information about Whois

- Search Whois
- Report Inaccurate Whois Listing

FAQs and Information

- FAQs (ICANN)
- Domain Transfer FAQs
- Explanation of Domain Name System
- Glossary of Terms

Arsys - Registra tu dominio... | ATENCIÓN 24/7 | 902 115 530 | 941 620 100 | EMAIL | CHAT | ES | EN | PROFESIONALES | SOPORTE | ÁREA DE CLIENTE

arsys Todos los Productos | Dominios | Hosting / Crear Web | Correo / Herramientas | Servidores | Soluciones

Registra tu dominio

- > Protege tu marca
- > 1.500 nuevas extensiones
- > Página de bienvenida
- > Gestión DNS

VER DOMINIOS

20 AÑOS DE INNOVACIÓN EN DOMINIOS, HOSTING Y CLOUD

DALE LA VUELTA A TU NEGOCIO

www.indicatudominio.com

BUSCAR

3x2 en todos los dominios

.com	.es	.org
10€	10€	10€

Delegación de la autoridad

- En una **zona** existe un **administrador local** que puede delegar en otros administradores.
Ejemplo: “ugr.es.” puede delegar en el Dep. de TSTC (“tstc.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.

- Pueden dar la **respuesta** bien **por tener autoridad** sobre el dominio en cuestión **o** bien **por tenerla en su caché**.

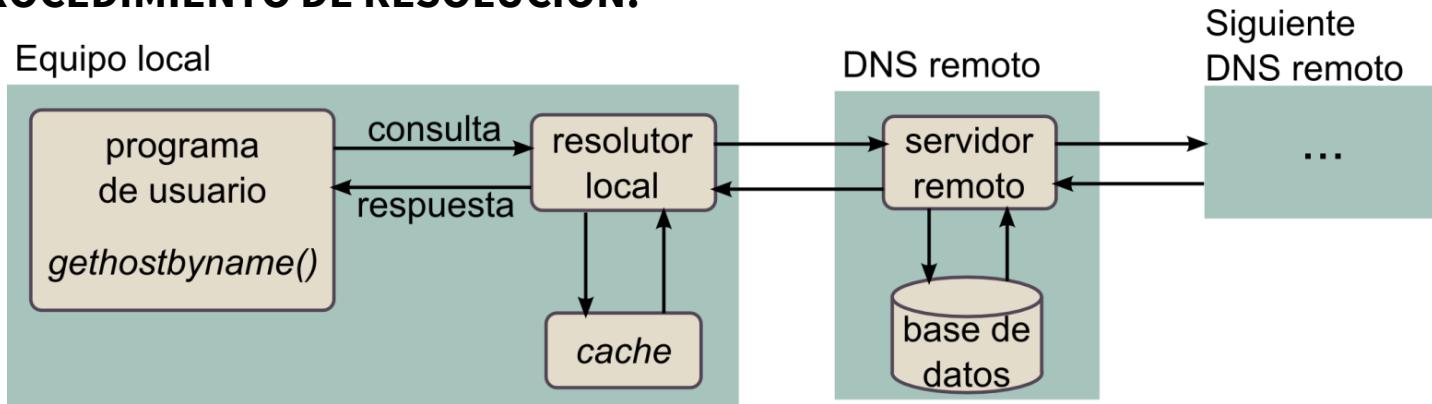
- En caso de no tenerla pueden consultar a otros servidores DNS.



Funcionamiento del servicio DNS

PROCEDIMIENTO DE RESOLUCIÓN:

PASO 1

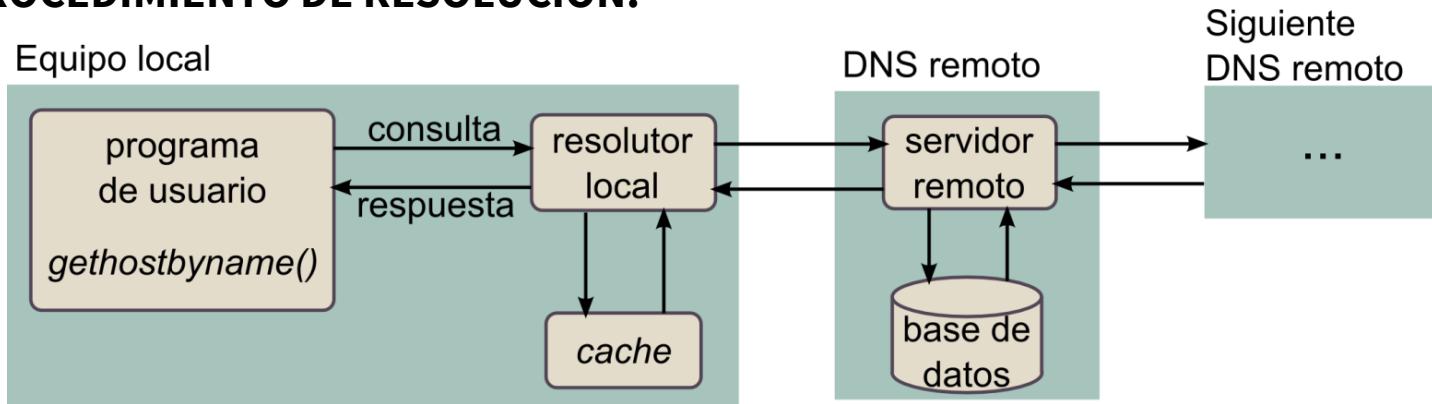


- El **Cliente DNS** (programa de usuario) hace consulta de resolución de nombre de dominio al **resolutor local** (en el mismo equipo). Éste intenta resolver la petición consultando:
 - **Información local de DNS** (ficheros de configuración con asignaciones de IPs).
 - **Información en caché** (memoria con resoluciones recientes).
- Si encuentra la respuesta, se la pasa al programa cliente.

Funcionamiento del servicio DNS

PROCEDIMIENTO DE RESOLUCIÓN:

PASO 2

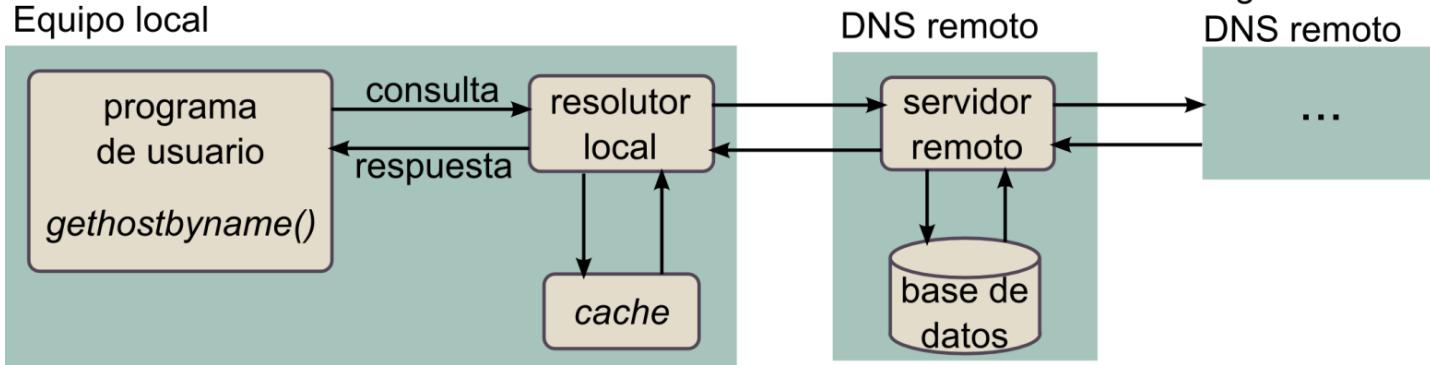


- Si el **resolutor local** no encuentra respuesta, **pasa la consulta** al servidor DNS que tenga asignado el equipo local por defecto (puede haber dos).
- El **servidor DNS comprueba** si dispone de la información para la solicitud en su base de datos, es decir, **si tiene autoridad** sobre ese dominio.

Funcionamiento del servicio DNS

PROCEDIMIENTO DE RESOLUCIÓN:

PASO 3

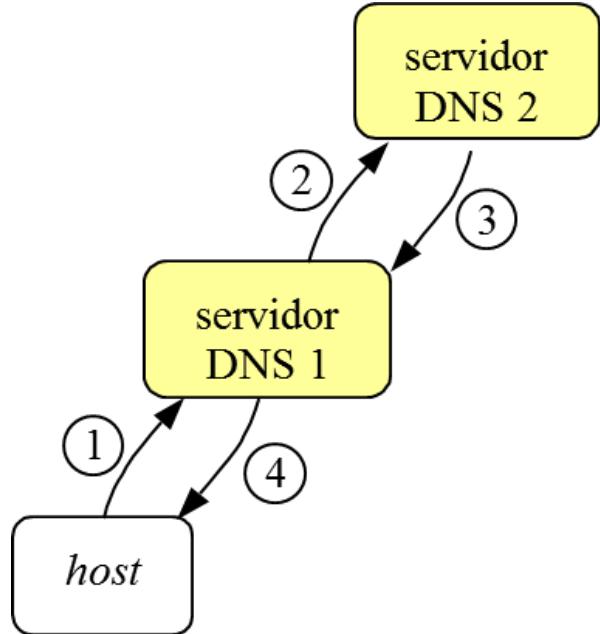


- Si el **servidor DNS no tiene autoridad**, debe encontrar al servidor que la tenga:
 - Consulta a los **servidores raíz** por el servidor con autoridad para el **TLD**.
 - Consulta al **servidor con autoridad** para el **dominio de segundo nivel**.
 - Y así **sucesivamente** hasta llegar al **servidor con autoridad para el dominio local**, que sería el que enviaría la respuesta.

Funcionamiento del servicio DNS

RESOLUCIÓN DISTRIBUIDA:

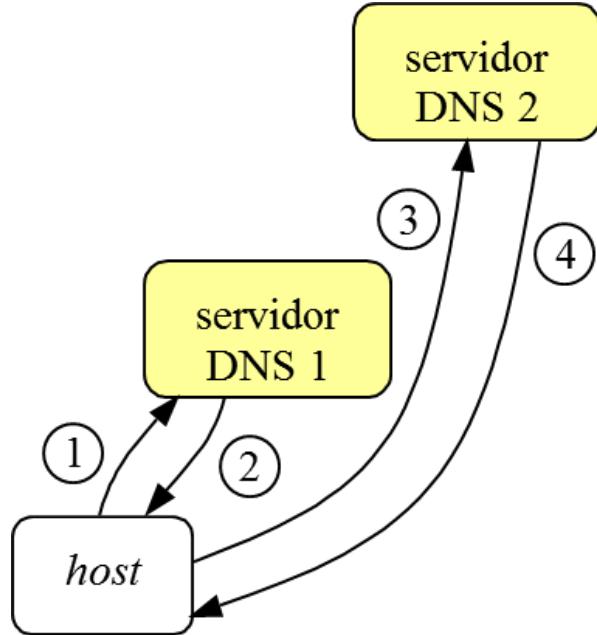
- **Recursiva:** el servidor DNS 1 (servidor por defecto) recibe la consulta del host y determina que es el servidor DNS 2 quien tiene autoridad sobre el dominio. Transfiere la consulta al servidor DNS 2 y recibe la respuesta, que luego pasa al host. Este último no será consciente de los saltos o consultas que haya tenido que realizar DNS 1.



Funcionamiento del servicio DNS

RESOLUCIÓN DISTRIBUIDA:

- **Iterativa:** el servidor DNS 1 envía al host la información sobre el siguiente servidor al que tendrá que consultar y será el host quién realice la siguiente consulta a DNS 2.

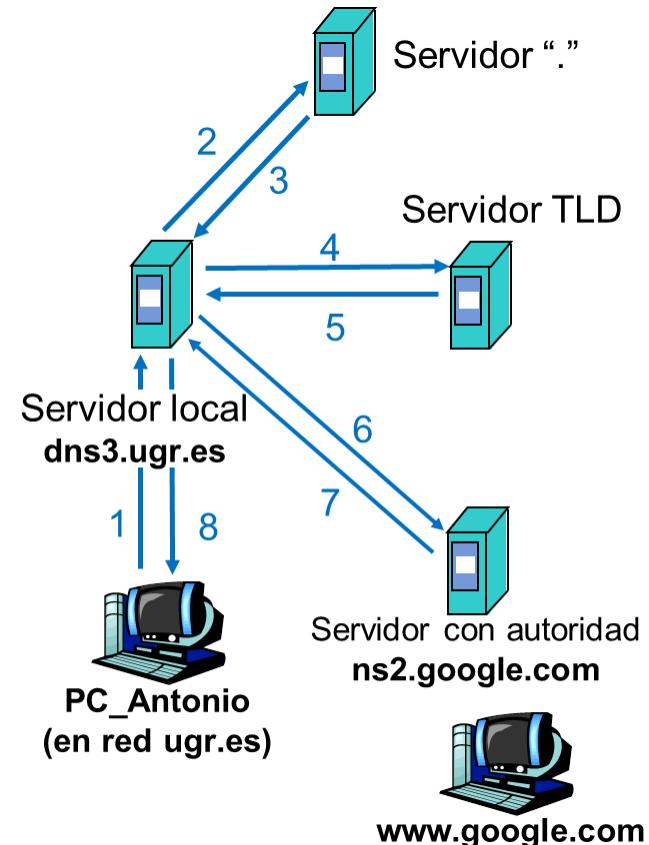


Funcionamiento del servicio DNS

PROCEDIMIENTO DE RESOLUCIÓN DE DNS DISTRIBUIDA:

- Ejemplo de funcionamiento del servicio de DNS para una consulta sobre la IP del nombre de dominio “www.google.com” realizada desde un equipo en la red de la UGR (ugr.es).

>>> Resolución de forma recursiva respecto al PC e iterativa respecto a servidor local de DNS <<<



Formato de la BD de DNS

- Todo **dominio** está **asociado** al menos a un **Resource Record (Registro de recursos)**.
- El formato de los registros es de la siguiente forma:

[Nombre_dominio] [TTL] [Clase] Tipo Valor_tipo

- Cuando un cliente (*resolutor*) da un **nombre de dominio al DNS**, lo que **recibe son los RR asociados** a ese nombre.
- Normalmente existen varios RR por dominio.

Formato de la BD de DNS

- **Nombre_dominio:** puede haber más de un registro por dominio. Se puede omitir, tomando por defecto el último nombre de dominio indicado.
- **TTL:** tiempo de vida (estabilidad del registro). La información *altamente estable* tiene un valor grande (86400 seg. o un día), mientras que la *volátil* recibe un valor pequeño (60 seg.).
- **Clase:** Actualmente sólo se utiliza *IN*, para información de Internet.

Formato de la BD de DNS

- **Valor_tipo:** es un número o texto ASCII dependiendo del tipo.

Tipo de Registro	Descripción
SOA <i>Start Of Authority</i>	Inicio de autoridad, identificando el dominio o la zona. Fija una serie de parámetros para esta zona.
NS <i>Name Server</i>	El nombre de dominio se hace corresponder con el nombre de un servidor con autoridad para dicho dominio.
A <i>Addres</i>	Dirección IP correspondiente al dominio (formato 32 bits). Si este tiene varias direcciones IP, habrá un registro por cada una de ellas.
CNAME	Es un alias que se corresponde con el nombre canónico verdadero.
MX <i>Mail eXchanger</i>	Indica la dirección IP del servidor de e-mail que corresponde a un nombre de dominio.
TXT	Texto, es una forma de añadir comentarios a la Base de Datos. Por ejemplo, para dar la dirección postal del dominio.
PTR <i>Pointer</i>	Puntero, hace corresponder una dirección IP con un nombre de dominio. Se usa en archivos dirección-nombre, la inversa del tipo A.
HINFO	Información del Host, como tipo y modelo de computadora.
WKS	Servicios públicos (Well-Known Services). Puede listar los servicios de las aplicaciones disponibles en el ordenador.

Formato de la BD de DNS

- Ejemplo de fichero de DNS (servidor BIND)

```
lабредес.pri. 94400 IN SOA eihal.лабредес.pri. admin.лабредес.pri.(
    2008042401 ; Serial
    28800       ; Refresh (seconds)
    14400       ; Retry (seconds)
    360000      ; Expire (seconds)
    86400 )     ; Minimum TTL(seconds)

лабредес.pri.      IN NS   eihal.лабредес.pri.
лабредес.pri.      IN MX   10 mailserver.лабредес.pri.
controler.лабредес.pri. IN CNAME eihal.лабредес.pri.

еihal.лабредес.pri. IN A   172.18.140.21
eil40146.лабредес.pri. IN A   172.18.140.146
mailserver.лабредес.pri. IN A   172.18.140.148
voipserver.лабредес.pri. IN A   172.18.140.149
```

Consultas inversas

- Existen registros en la base de datos para la **resolución inversa** (traducir direcciones IP a nombres de dominio).
- Se refieren al dominio especial “**in-addr.arpa**”.
- El dominio asociado a la **dirección IP w.x.y.z** se almacenará:
 - Campo Nombre_dominio → z.y.x.w.in-addr.arpa.
 - Campo PTR → nombre del dominio correspondiente a la IP.
- La inversión de los bytes es necesaria debido a que los nombres de dominio son más genéricos por la derecha, al contrario que ocurre con las direcciones IP.

Formato de los mensajes DNS

- Los mensajes de consulta y respuesta intercambiados entre clientes y servidores DNS tienen un formato sencillo. Un servidor añade la información requerida a la consulta original y la envía de vuelta.

Cabecera.
Consulta (o consultas).
(En la respuesta) RR de respuesta.
(En la respuesta) RR que identifican servidores con autorización.
(En la respuesta) RR con información adicional.



Formato de los mensajes DNS

- Campos de la **cabecera**:

Campo	Descripción
ID	Identificador para hacer corresponder una respuesta con su petición.
Parámetros	Consulta o respuesta. Consulta normal o inversa. En respuestas, si es de un servidor con autoridad. Recursivo o no. En respuestas, si la recursión está disponible. En respuestas, código de error.
Num. de consultas	Proporcionado en una consulta y en una respuesta.
Num. de respuestas	Proporcionado en una respuesta.
Num. de registros de autoridad	Proporcionado en una respuesta. La información de los registros de autoridad incluye los nombres de los servidores que contienen los datos de confianza.
Num. de registros adicionales	Proporcionado en una respuesta. La información incluye las direcciones de los servidores de confianza.

Formato de los mensajes DNS

- Campos de una **consulta** (puede haber varias en una petición):

Campo	Descripción
Nombre	Nombre de dominio o dirección IP en el subárbol IN-ADDR.ARPA
Tipo	Tipo de consulta, por ejemplo A o NS
Clase	IN para Internet, se representa como 1

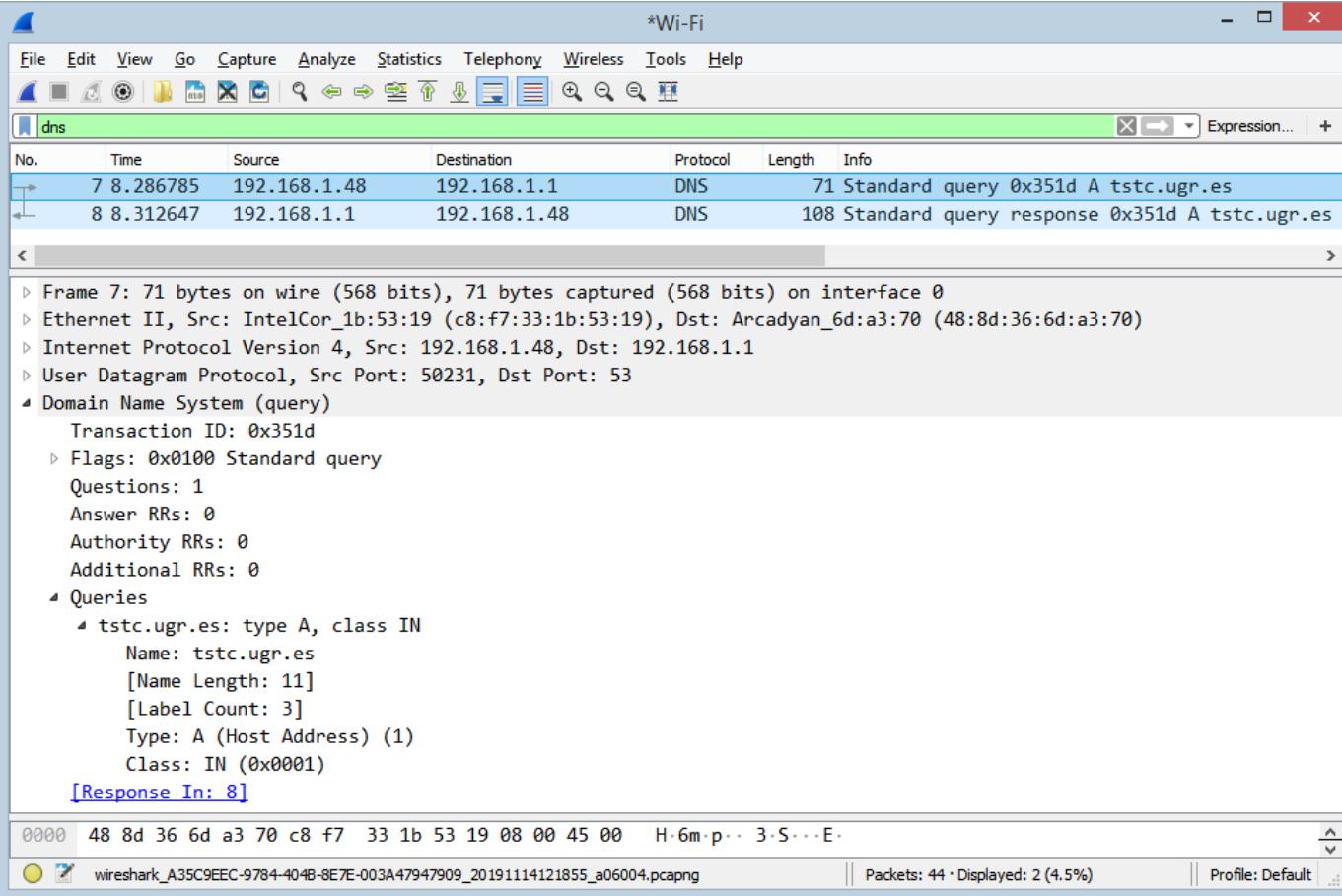
Formato de los mensajes DNS

- Campos de **respuesta, información de autoridad e información adicional**. Secuencia de RR.

Campo	Descripción
Nombre	Nombre del nodo para este registro.
Tipo	Tipo de registro, como SOA o A, indicado por un código numérico.
Clase	IN, se representa como 1.
TTL	Tiempo de vida, un entero con signo de 32 bits que indica cuánto tiempo puede permanecer el registro en la caché.
RDLENGTH	Tamaño del campo de datos de recursos.
RDATA	La información, por ejemplo, para un registro de direcciones, es la dirección IP. Para un registro SOA, incluye más datos.

Formato de los mensajes DNS

Ejemplo de
consulta a
“tstc.ugr.es”
(captura Wireshark)



The screenshot shows a Wireshark capture window titled "*Wi-Fi". The packet list pane displays two DNS-related frames:

No.	Time	Source	Destination	Protocol	Length	Info
7	8.286785	192.168.1.48	192.168.1.1	DNS	71	Standard query 0x351d A tstc.ugr.es
8	8.312647	192.168.1.1	192.168.1.48	DNS	108	Standard query response 0x351d A tstc.ugr.es

The details pane provides a hierarchical breakdown of the selected DNS query (Frame 7):

- Frame 7: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface 0
- Ethernet II, Src: IntelCor_1b:53:19 (c8:f7:33:1b:53:19), Dst: Arcadyan_6d:a3:70 (48:8d:36:6d:a3:70)
- Internet Protocol Version 4, Src: 192.168.1.48, Dst: 192.168.1.1
- User Datagram Protocol, Src Port: 50231, Dst Port: 53
- Domain Name System (query)
 - Transaction ID: 0x351d
 - Flags: 0x0100 Standard query
 - Questions: 1
 - Answer RRs: 0
 - Authority RRs: 0
 - Additional RRs: 0
- Queries
 - tstc.ugr.es: type A, class IN
 - Name: tstc.ugr.es
 - [Name Length: 11]
 - [Label Count: 3]
 - Type: A (Host Address) (1)
 - Class: IN (0x0001)

The bottom pane shows the raw hex and ASCII data for the selected frame.

Formato de los mensajes DNS

Ejemplo de respuesta de “ugr.es” por “tstc”
captura Wireshark)



The screenshot shows a Wireshark capture window titled "dns". The packet list pane displays two DNS messages:

No.	Time	Source	Destination	Protocol	Length	Info
7	8.286785	192.168.1.48	192.168.1.1	DNS	71	Standard query 0x351d A tstc.ugr.es
8	8.312647	192.168.1.1	192.168.1.48	DNS	108	Standard query response 0x351d A tstc.ugr.es

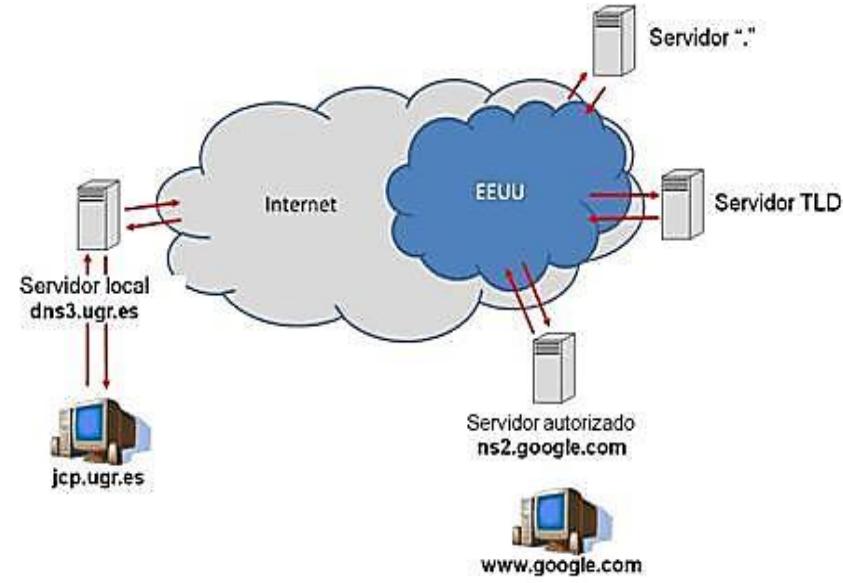
The details pane shows the second DNS response message (packet 8) expanded:

- Internet Protocol Version 4, Src: 192.168.1.1, Dst: 192.168.1.48
- User Datagram Protocol, Src Port: 53, Dst Port: 50231
- Domain Name System (response)
 - Transaction ID: 0x351d
 - Flags: 0x8180 Standard query response, No error
 - Questions: 1
 - Answer RRs: 2
 - Authority RRs: 0
 - Additional RRs: 0
- Queries
- Answers
 - tstc.ugr.es: type CNAME, class IN, cname uniweb.ugr.es
 - Name: tstc.ugr.es
 - Type: CNAME (Canonical NAME for an alias) (5)
 - Class: IN (0x0001)
 - Time to live: 172800
 - Data length: 9
 - CNAME: uniweb.ugr.es
 - uniweb.ugr.es: type A, class IN, addr 150.214.204.201
 - Name: uniweb.ugr.es
 - Type: A (Host Address) (1)
 - Class: IN (0x0001)
 - Time to live: 127173
 - Data length: 4
 - Address: 150.214.204.201

Ejercicio

EJERCICIO 6 - RELACIÓN DE PROBLEMAS DEL TEMA 2

- En la siguiente figura se ilustra un ejemplo de acceso DNS por parte de una máquina (jcp.ugr.es) que quiere acceder a los servicios de www.google.com. Para obtener la dirección IP del servidor, es necesario que la consulta pase por todos los servidores del gráfico. Considerando unos retardos promedio de 8 μ s dentro de una red LAN, de 12 ms en cada acceso a través de Internet (4 ms si la conexión se restringe a EEUU) y de 1 ms de procesamiento en cada servidor:
 - Calcule el tiempo que se tardaría si la solicitud al servidor local es recursiva, pero el propio servidor local realiza solicitudes iterativas.
 - Especifique una política (recursiva-iterativa) más rápida de solicitudes y el tiempo que tardaría la solicitud en ser respondida. ¿Qué desventaja tiene sobre la solución anterior?



TEMA 2. Servicios y Protocolos en Internet

- 2.1. Introducción a las aplicaciones de red
- 2.2. Servicio de Nombres de Dominio (DNS)
- **2.3. Navegación web**
- 2.4. Correo electrónico
- 2.5. Protocolos seguros
- 2.6. Aplicaciones multimedia
- 2.7. Aplicaciones para interconectividad de redes locales
- 2.8. Cuestiones y ejercicios

Introducción

- 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

Introducción

- 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**".

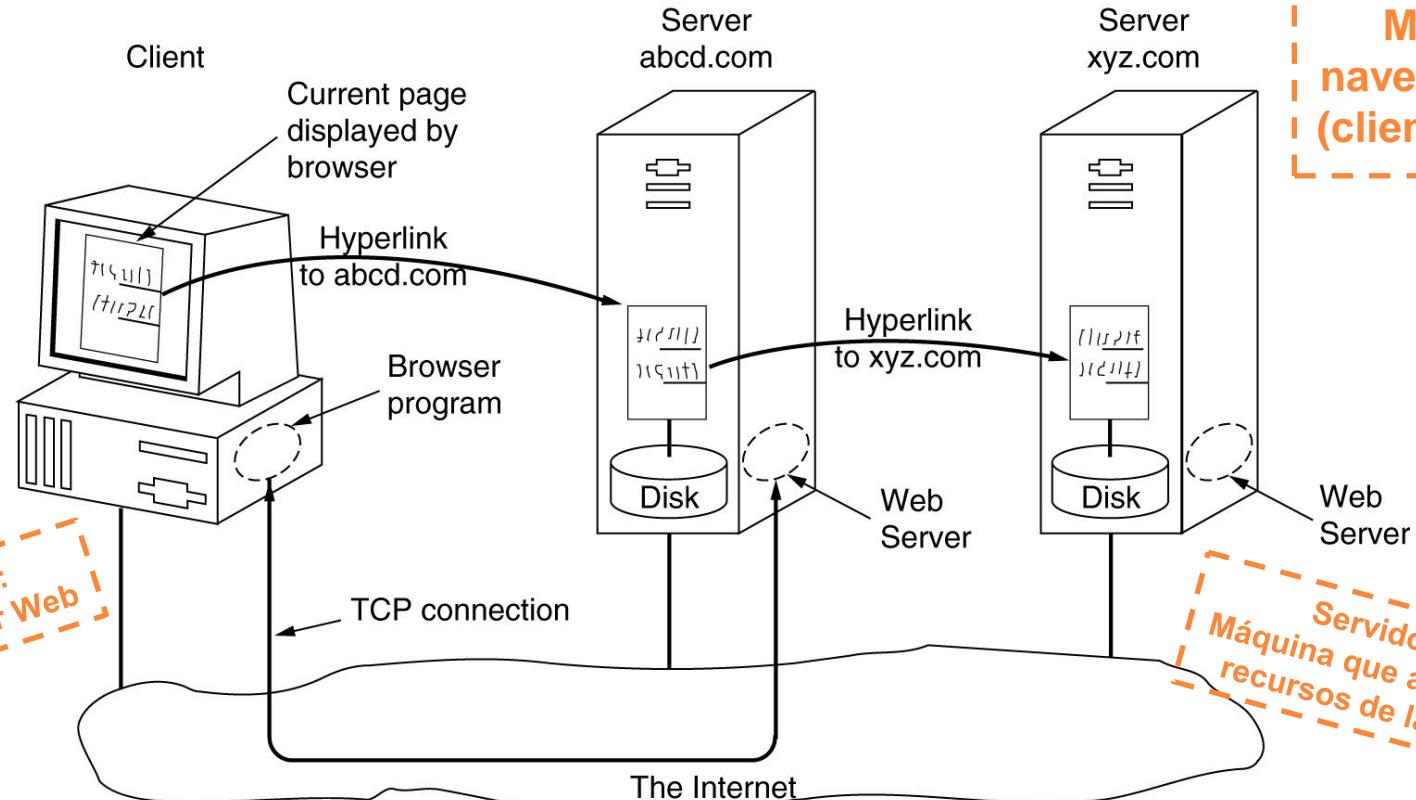
Redes privadas que
usan tecnología de
Internet

Introducción

- 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.

Una URL
es un URI

Introducción



Modelo de navegación Web (cliente/servidor)

Cliente:
Navegador Web

Servidor:
Máquina que aloja los recursos de la web

Cliente web

- 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 los 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 icono GIF, un vídeo en MPEG... El servidor indica el tipo MIME. 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).

MIME:
Multipurpose Internet
Mail Extensions

Cliente web

PROCESAMIENTO

<http://www.epsg.upv.es/historia.php?modo=presentacion&titulo=Hist%F2ria>

1. El navegador determina la URL (de un enlace)
2. Accede al servicio DNS para averiguar la dirección IP de *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. Y envía “*GET /historia.php?modo=presentacion&titulo=Hist%F2ria*”
6. El servidor manda el fichero *historia.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 *historia.php* y los recursos asociados

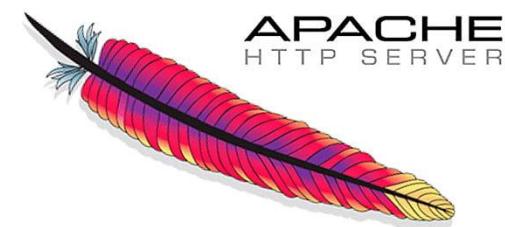


Servidor web

- Máquina que aloja los recursos de las páginas web.
- Escucha conexiones de tipo TCP en el puerto 80.
- Entregan 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



Servidor web

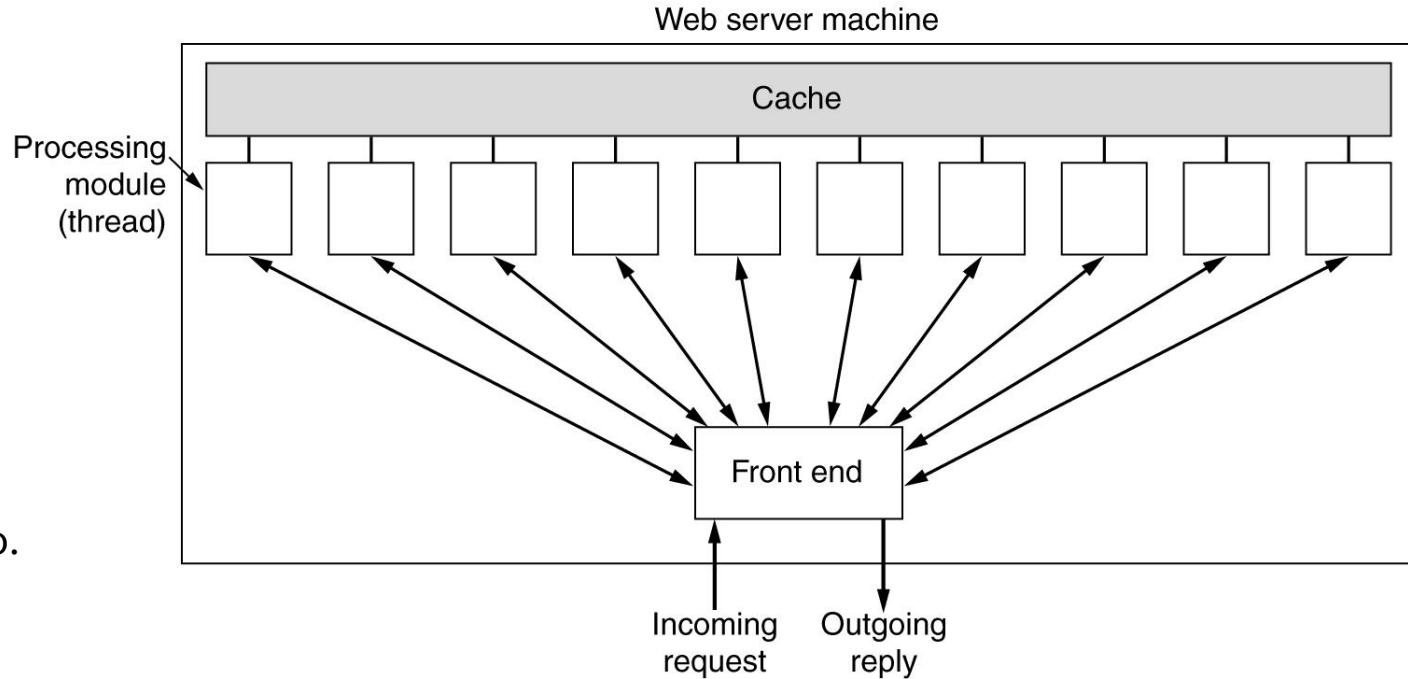
PROCESAMIENTO (avanzado)

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

Servidor web

ARQUITECTURA HABITUAL

Servidor web multihilo con módulos de acceso (Front End) y de procesamiento.



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.
- Esta 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**)
 - servidor → envía mensaje con el estado de la operación y el resultado (**response**)

Protocolo HTTP

CARACTERÍSTICAS PRINCIPALES

- 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:
 - **GET** → para recoger un objeto
 - **POST** → para enviar información al servidor
 - **HEAD** → para solicitar las características de un objeto (Ejemplo: fecha de modificación de un documento HTML).

Protocolo HTTP

CARACTERÍSTICAS PRINCIPALES

- Protocolo “**stateless**” → El servidor no mantiene información de las peticiones de los clientes.
 - Cookie: información breve (y estructurada) enviada por un sitio web y almacenada en el navegador. La puede consultar el sitio web en futuras visitas
- Dos tipos de servicio:
 - **No persistente** → Se envía únicamente un objeto en cada conexión TCP.
 - **Persistente** → Pueden enviarse múltiples objetos sobre una única conexión TCP entre cliente y servidor

Protocolo HTTP

FUNCIONAMIENTO

1a. El Cliente HTTP inicia conexión TCP al servidor HTTP (proceso) en www.ugr.es en el puerto 80 (segmento SYNC de TCP)

2. El Cliente HTTP envía ***request message*** para el objeto

1b. El Servidor HTTP acepta la conexión y solicita al cliente abrir la conexión (SYNC+ACK)
1c. El cliente confirma (ACK)

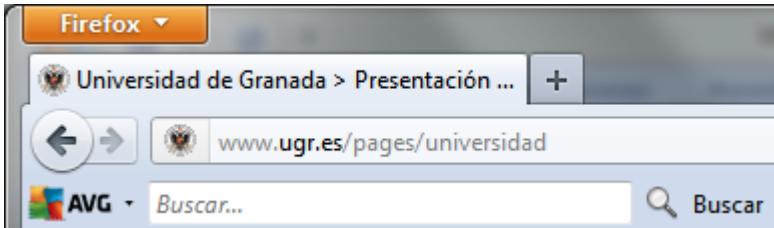
3. El servidor HTTP devuelve la respuesta (***response message***)

4. Si es persistente → Envío de más objetos por la misma conexión TCP

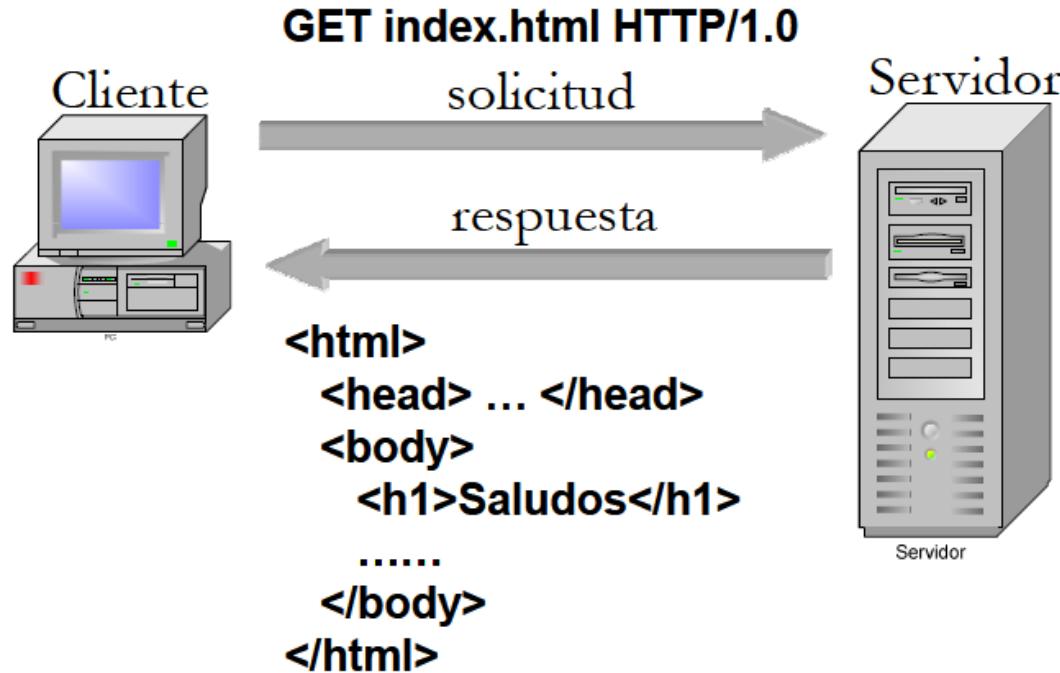
5. Cierre de conexión TCP (liberación de recursos)

6. Nuevas conexiones TCP

tiempo



Protocolo HTTP



Ejercicio

EJERCICIO 8 - RELACIÓN DE PROBLEMAS DEL TEMA 2

- Compare el rendimiento en términos temporales de HTTP persistente y no persistente considerando los siguientes parámetros:
 - Descarga de una página web con 10 objetos incrustados
 - Tiempo de Establecimiento de conexión TCP → 5 ms
 - Tiempo de Cierre de conexión TCP → 5 ms
 - Tiempo de solicitud HTTP → 2 ms
 - Tiempo de respuesta HTTP (página web u objeto) → 10 ms

Protocolo HTTP 1.1 (RFC 2616)

MENSAJE REQUEST

- HTTP request message (solicitudes del cliente al servidor).

Línea de petición
(GET, POST,
HEAD)

Líneas de cabecera

Carriage Return (CR) +
Line Feed (LF)
Indican fin del mensaje

GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language:fr

(extra CR LF)

Protocolo HTTP 1.1 (RFC 2616)

MENSAJE RESPONSE

- HTTP response message: (respuestas del servidor al cliente).

Línea de estado

HTTP/1.1 200 OK

200 OK
301 Moved Permanently
400 Bad Request
404 Not Found
505 HTTP Version Not Supported

Líneas de cabecera

Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998
Content-Length: 6821
Content-Type: text/html

Datos,

Ej: fichero html

data data data data data ...

Protocolo HTTP 1.1 (RFC 2616)

MÉTODOS (Acciones solicitadas en los *request messages* del cliente)

- **OPTIONS:** solicitud de información sobre las opciones disponibles
- **GET:** solicitud de un recurso (puede ser condicional)
(Se envía al pulsar sobre un enlace o al teclear una URL directamente en el navegador)
- **HEAD:** igual que GET pero el servidor no devuelve el “cuerpo” sólo cabeceras
(Utilizado por los gestores de cachés de páginas, para saber cuándo es necesario actualizar la copia que se tiene de un fichero)
- **POST:** solicitud al servidor para que subordine a la URI especificada, los datos incluidos en la solicitud.
(Ej: Envío de datos de un formulario. El servidor pasará los datos a un proceso encargado de su utilización)
- **PUT:** solicitud de sustituir la URI especificada con los datos incluidos en la solicitud.
- **DELETE:** solicitud de borrar la URI especificada.

Protocolo HTTP 1.1 (RFC 2616)

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

CABECERAS y CAMPOS (47 *request fields*, 49 *response fields*)

From: User-Agent:, Content-Type:, Content-Length:, ...

http://en.wikipedia.org/wiki/List_of_HTTP_header_fields

Protocolo HTTP 1.1 (RFC 2616)

CAMPOS DE CABECERA COMUNES PARA PETICIONES Y RESPUESTAS

- **Content-Type:** descripción MIME de la información contenida en este mensaje.
- **Content-Length:** longitud en bytes de los datos enviados, expresado en base decimal.
- **Content-Encoding:** formato de codificación de los datos enviados en este mensaje. Sirve, por ejemplo, para enviar datos comprimidos o encriptados.
- **Date:** fecha local de la operación. Las fechas deben incluir la zona horaria en que reside el sistema que genera la operación. Por ejemplo: Sunday, 12-Dec-96 12:21:22 GMT+01. No existe un formato único en las fechas.

Protocolo HTTP 1.1 (RFC 2616)

CAMPOS DE CABECERA SOLO PARA PETICIONES DEL CLIENTE

- **Accept:** campo opcional que contiene una lista de tipos MIME aceptados por el cliente.
- **Authorization:** clave de acceso que envía un cliente para acceder a un recurso de uso protegido o limitado. La información incluye el formato de autorización empleado, seguido de la clave de acceso propiamente dicha.
- **From:** campo opcional que contiene la dirección de correo electrónico del usuario del cliente Web que realiza el acceso.

Protocolo HTTP 1.1 (RFC 2616)

CAMPOS DE CABECERA SOLO PARA PETICIONES DEL CLIENTE

- **If-Modified-Since:** permite realizar operaciones GET condicionales, en función de si la fecha de modificación del objeto requerido es anterior o posterior a la fecha proporcionada. Puede ser utilizada por los sistemas de almacenamiento temporal de páginas. Es equivalente a realizar un HEAD seguido de un GET normal.
- **Referer:** contiene la URL del documento desde donde se ha activado este enlace. De esta forma, un servidor puede informar al creador de ese documento de cambios o actualizaciones en los enlaces que contiene. No todos los clientes lo envían.
- **User-agent:** cadena que identifica el tipo y versión del cliente que realiza la petición. Por ejemplo, los browsers de Netscape envían cadenas del tipo User-Agent: Mozilla/3.0.

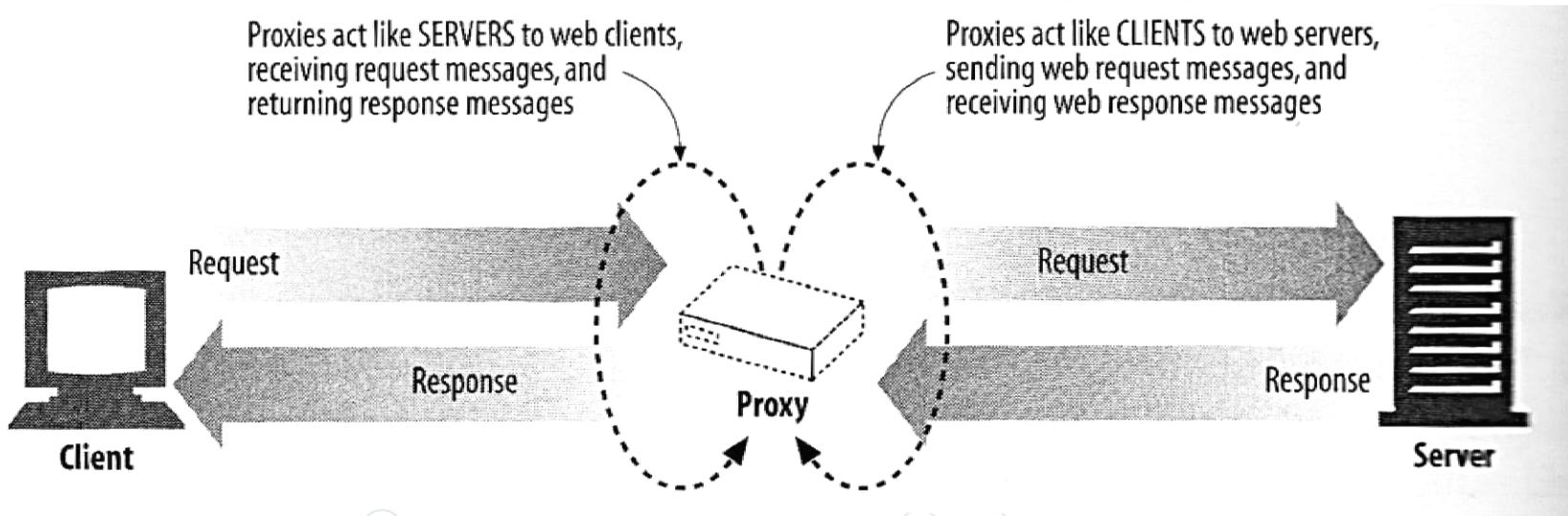
Protocolo HTTP 1.1 (RFC 2616)

CAMPOS DE CABECERA SOLO PARA RESPUESTAS DEL SERVIDOR

- **Allow:** informa de los comandos HTTP opcionales que se pueden aplicar sobre el objeto al que se refiere esta respuesta. Por ejemplo, Allow: GET, POST.
- **Expires:** fecha de expiración del objeto enviado. Los sistemas de cache deben descartar las posibles copias del objeto pasada esta fecha. Por ejemplo, Expires: Thu, 12 Jan 97 00:00:00 GMT+1. No todos los sistemas lo envían.
- **Last-modified:** fecha local de modificación del objeto devuelto. Se puede corresponder con la fecha de modificación de un fichero en disco, o, para información generada dinámicamente desde una base de datos, con la fecha de modificación del registro de datos correspondiente.

Servidor Proxy

- Servidor que se sitúa entre el cliente y el servidor web, que hace papel de servidor de cara al cliente y de cliente de cara al servidor.



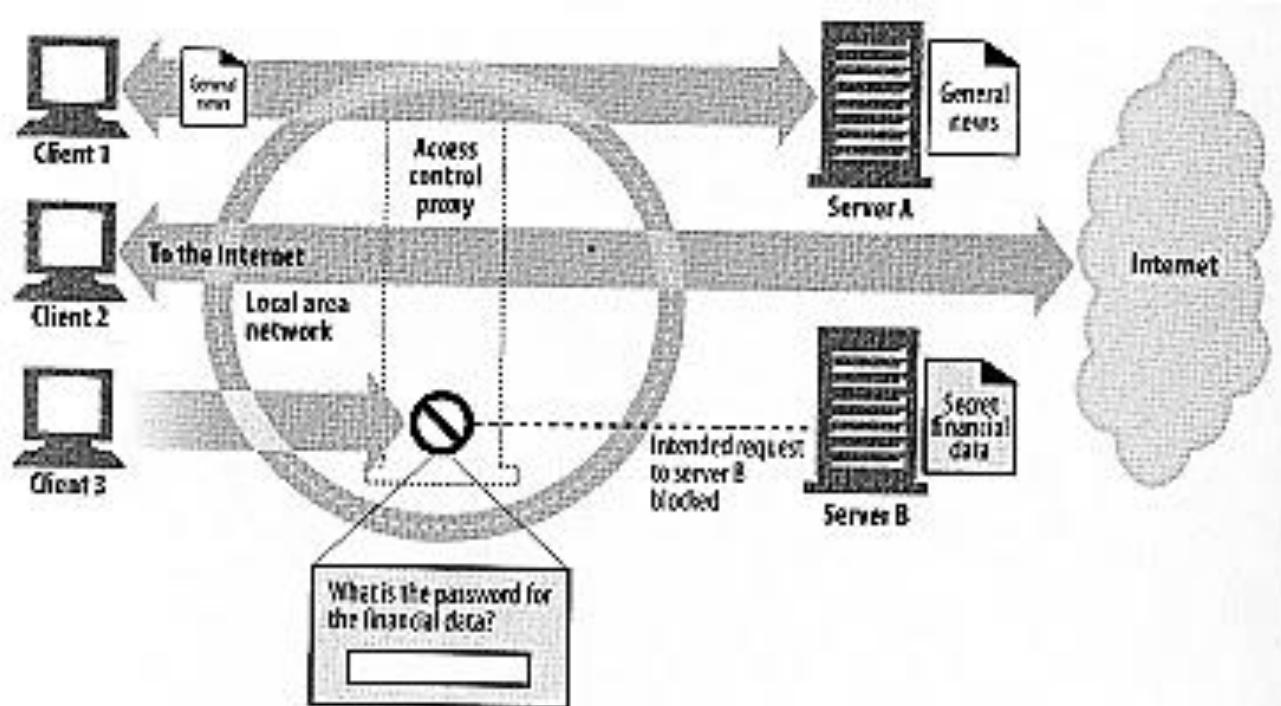
Servidor Proxy

VENTAJAS

- **Control:** Sólo el proxy hace la petición real al servidor, por tanto se pueden limitar y restringir los derechos de los usuarios y dar permisos sólo al proxy.
- **Ahorro:** Sólo uno de los usuarios (el proxy) ha de estar equipado para hacer el trabajo real.
- **Velocidad:** Si varios clientes van a pedir el mismo recurso, el proxy puede hacer caché (guardar la respuesta de una petición para darla directamente cuando otro usuario la pida). Así no tiene que volver a contactar con el servidor destino, y se resuelve antes la petición.
- **Filtrado:** El proxy puede negarse a responder algunas peticiones si detecta que están prohibidas.

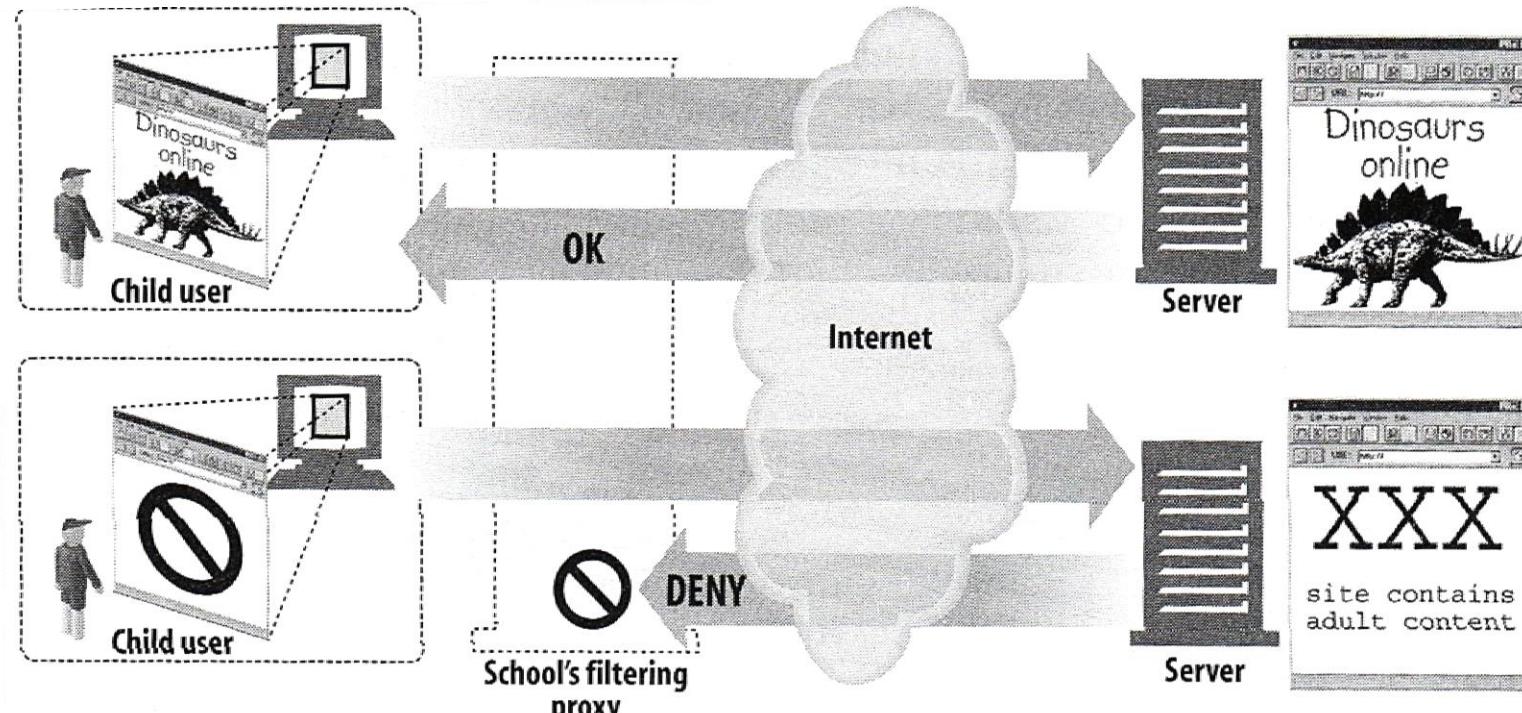
Servidor Proxy

EJEMPLO CONTROL DE ACCESO A DOCUMENTOS CENTRALIZADO



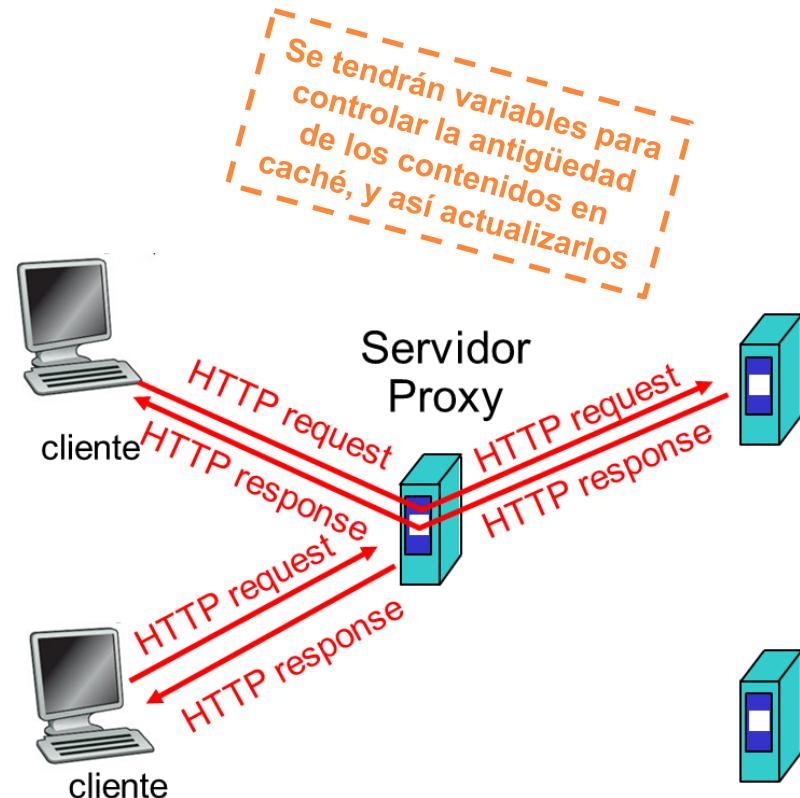
Servidor Proxy

EJEMPLO FILTRO DE PROTECCIÓN A MENORES



Servidor Caché

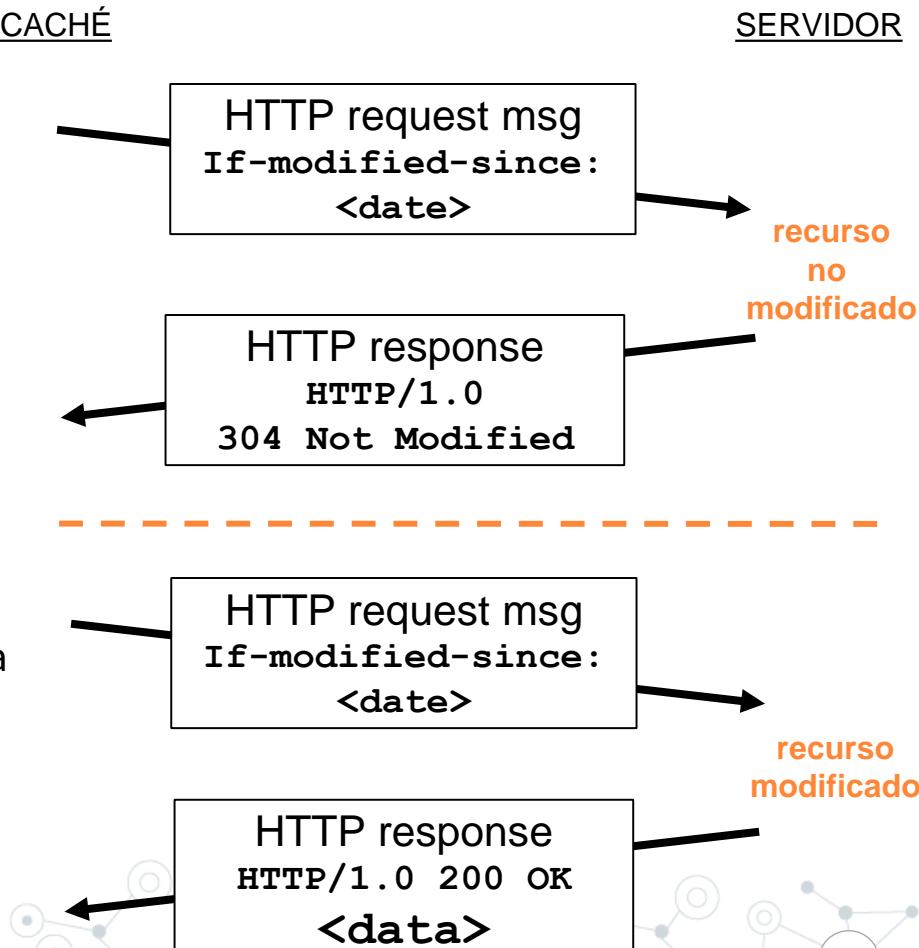
- Servidor **proxy** que **almacena las páginas visitadas** (recientemente) por uno o varios clientes.
- El **navegador/cliente** enviará todas las **peticiones HTTP al servidor caché**:
 - Si recurso está en la caché → se devuelve el objeto local
 - Si no → el servidor caché solicita el recurso al servidor destino, actualiza la caché con el mismo, y lo sirve al cliente
- La caché puede estar en el propio ordenador del usuario (cliente).



Servidor Caché

- **Objetivo:** reducir el tráfico si la cache tiene una versión actualizada del objeto/recurso.
- La **cache solicita** el recurso condicionado a la fecha de la copia local, usando:
`If-modified-since: <date>`
- Si el recurso es más reciente de esa fecha, el servidor lo envía.
- El servidor responde sin el recurso si la copia de la cache está actualizada:

`HTTP/1.0 304 Not Modified`



Cookies (RFC 2109)

- Las **cookies** son pequeños **ficheros de texto** que se intercambian los **clientes y servidores HTTP**, para solucionar una de las principales deficiencias del protocolo: la falta de información de estado entre dos transacciones (**stateless**). Fueron introducidas por Netscape.
- La primera vez que un **usuario accede** a un determinado **documento** de un **servidor**, éste **proporciona una cookie** que contiene datos que **relacionarán posteriores operaciones**.
- El **cliente almacena** la **cookie** en su sistema para usarla después. En los **futuros accesos** a este **servidor**, el navegador podrá **proporcionar** la **cookie original**, que servirá de **nexo entre este acceso** y los anteriores.
- Todo este proceso se realiza automáticamente, sin intervención del usuario.
- Usos habituales: compra electrónica, recuerdo de contraseñas, preferencias.



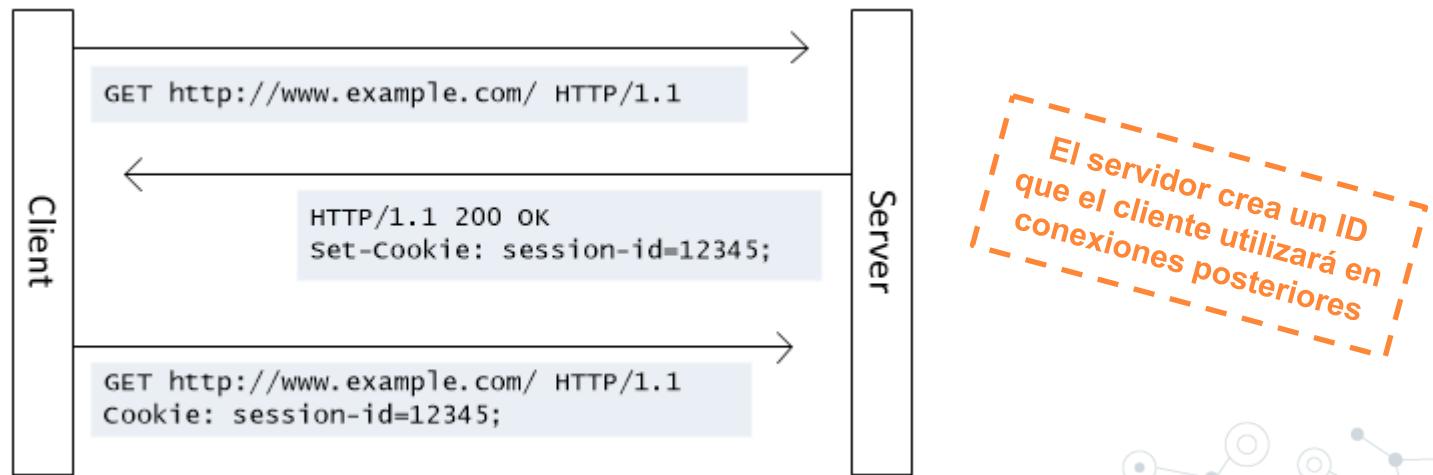
Cookies (RFC 2109)

USO DE LAS COOKIES

- Una **cookie** es simplemente una serie de **líneas de texto**, con **pares variable/valor**. Existe un conjunto predefinido de **nombres de variable**, necesarias para el correcto funcionamiento.
- Por ejemplo:
 - **Domain**: conjunto de direcciones Internet para el que es válida la cookie. Se puede dar una dirección única (www.mitienda.es) o un rango (.netscape.com).
 - **Path**: fija el subconjunto de URLs para las que sirve esta cookie.
 - **Version**: Permite seleccionar entre diferentes versiones del modelo de cookies.
 - **Expires**: Fecha de expiración de la información. Si no se incluye, los datos son descartados al finalizar la sesión con el cliente Web.

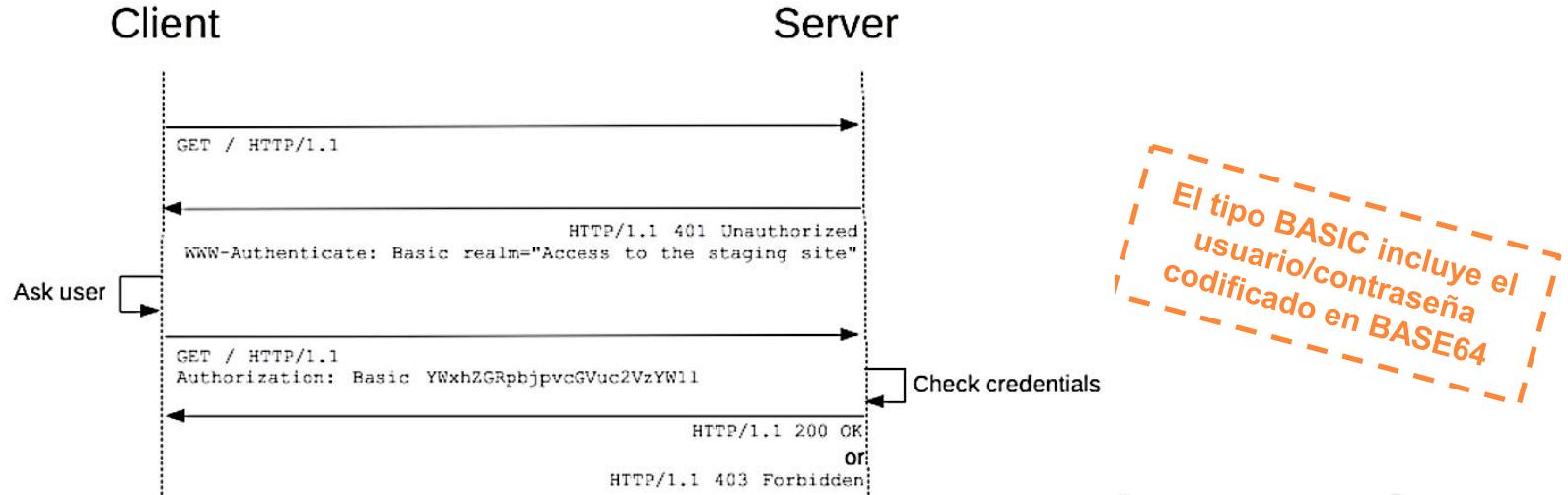
Cookies (RFC 2109)

- Un **servidor HTTP envía** los diferentes campos de una **cookie** con la cabecera **HTTP Set-Cookie**: *Set-Cookie: Domain=www.unican.es; Path=/; Nombre=Luis; Expires Fri, 15-Jul-97 12:00:00 GMT*
- Cuando se **accede** a una **URL** que **verifica el par dominio/path registrado**, el **cliente enviará** automáticamente la **información** de los diferentes **campos** de la **cookie** con la cabecera **HTTP Cookie**: *Cookie: Domain=www.unican.es; Path=/; Nombre=Luis*



Acceso restringido

- **HTTP no es seguro**, pero incluye cabeceras **WWW-Authenticate** (servidor) y **Authorization** (cliente) para **restringir el acceso** a recursos.
- HTTPS → Versión segura de HTTP. Encripta las transmisiones de peticiones y respuestas.



TEMA 2. Servicios y Protocolos en Internet

- 2.1. Introducción a las aplicaciones de red
- 2.2. Servicio de Nombres de Dominio (DNS)
- 2.3. Navegación web
- **2.4. Correo electrónico**
- 2.5. Protocolos seguros
- 2.6. Aplicaciones multimedia
- 2.7. Aplicaciones para interconectividad de redes locales
- 2.8. Cuestiones y ejercicios

Introducción

- 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.



Introducción

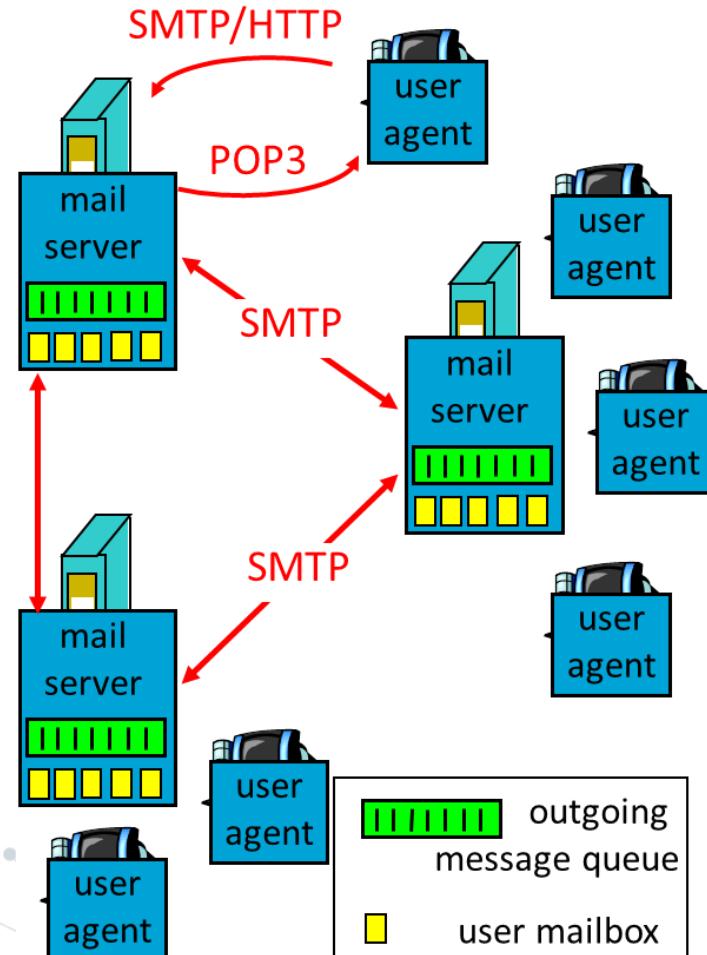
- El uso de la **arroba (@)** se incorporó en 1971, al ser un carácter que no existe en ningún nombre en el mundo.
- La **arroba divide** la dirección entre el **usuario** y el **dominio del proveedor de correo** (máquina en la que se aloja el correo).
- En inglés la arroba se lee “at” (en).
- El **nombre 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**.

Introducción

- 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.

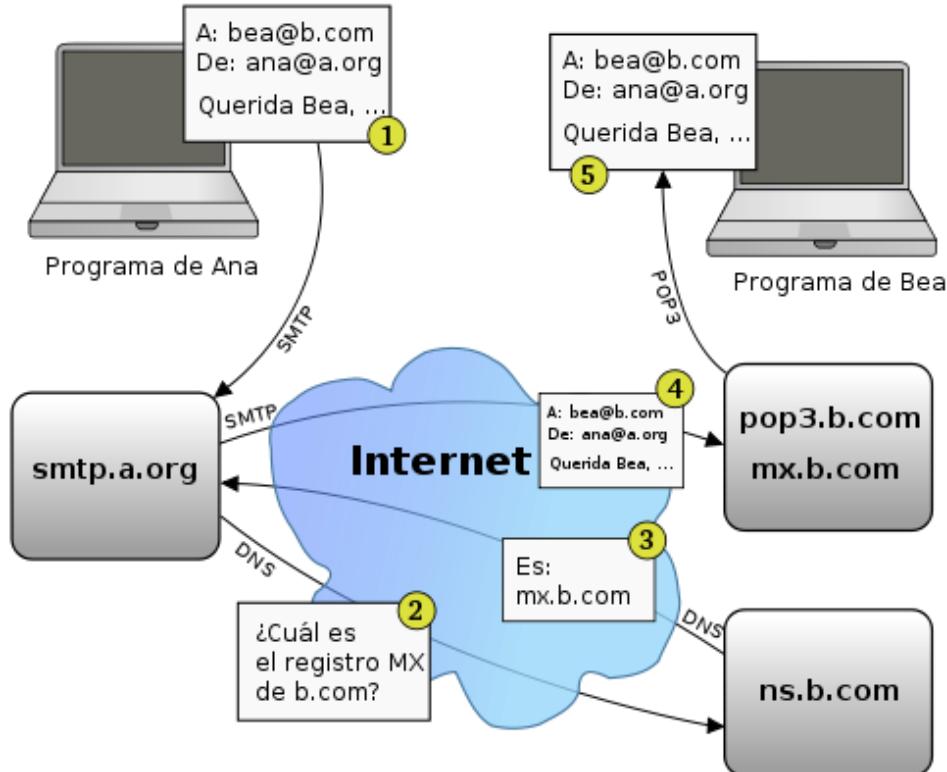
Introducción

- **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



Introducción

- Es un **servicio** que **hace uso de DNS**:



MX es el servidor de correo asociado a un dominio en DNS

SMTP (RFC 5321)

- **Simple Mail Transfer Protocol.**
- Protocolo **cliente/servidor** sencillo.
- Funciona mediante **TCP** a través del **puerto 25** (en el servidor):
 - Handshaking (“saludo”)
 - Transferencia de mensajes
 - Cierre
- **SMTP es un protocolo:**
 - Orientado a texto.
 - Orientado a conexión
 - Statefull
- La interacción entre cliente SMTP y servidor SMTP se realiza mediante comandos/respuesta:
 - Comandos → texto ASCII
 - Respuestas → código de estado y frases explicativas

Inicialmente los mensajes se codificaban con ASCII de 7 bits!!!

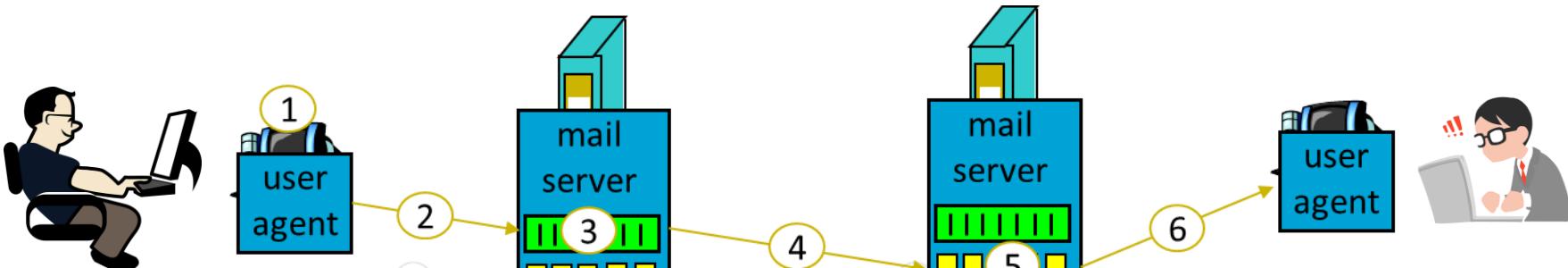
Posteriormente con MIME se pueden enviar ASCII de 8 bits y formatos enriquecidos

SMTP (RFC 5321)

PASOS EN EL ENVÍO Y RECEPCIÓN DE 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.

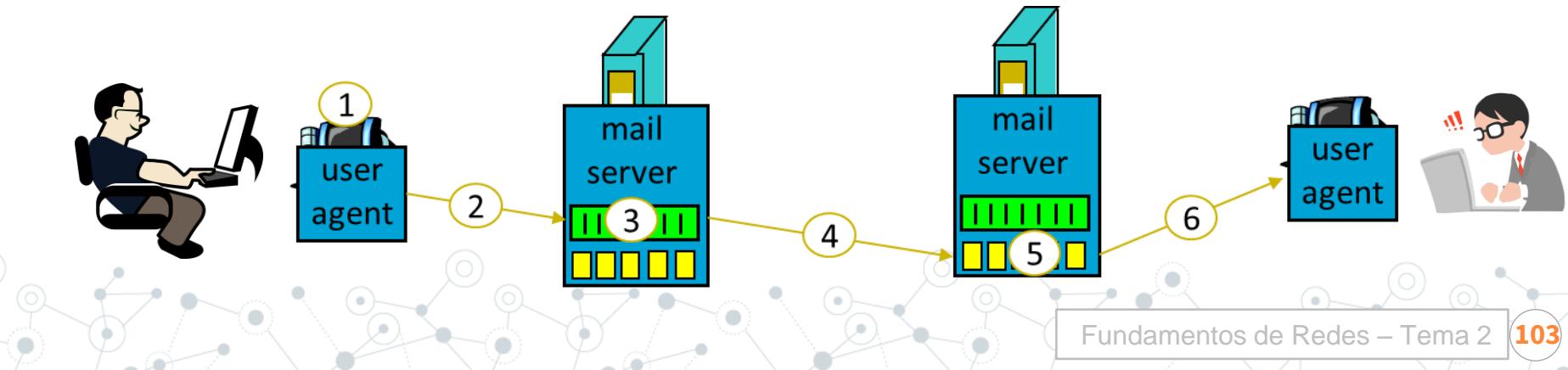
HTTP si se usa
Webmail



SMTP (RFC 5321)

PASOS EN EL ENVÍO Y RECEPCIÓN DE CORREO

- 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.



SMTP (RFC 5321)

COMANDOS SMTP (Cliente)

<u>Comando</u>	<u>Descripción</u>
HELO (ahora EHLO)	Identifica el remitente al destinatario.
MAIL FROM	Identifica una transacción de correo e identifica al emisor.
RCPT TO	Se utiliza para identificar un destinatario individual . 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 última línea debe llevar únicamente el carácter punto ":" seguido de <CR><LF>.
RSET	Aborta la transacción de correo actual.
NOOP	No operación. Indica al extremo que envíe una respuesta positiva. Keepalives
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 valido.
EXPN	Pide al receptor la confirmación de una lista de correo 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 inviertan los papeles , 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.

SMTP (RFC 5321)

RESPUESTAS SMTP (Servidor)

<u>Código</u>	<u>Descripción</u>
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.

SMTP (RFC 5321)

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

SMTP (RFC 5321)

CAPTURA WIRESHARK

cliente -> serv_correo TCP 3612 > 25 [SYN] Seq=0 Win=64512 Len=0 MSS=1460
 serv_correo -> cliente TCP 25 > 3612 [SYN, ACK] Seq=0 Ack=1 win=65535 Len=0 MSS=1460
 cliente -> serv_correo TCP 3612 > 25 [ACK] Seq=1 Ack=1 win=64512 Len=0
 serv_correo -> cliente SMTP Response: 220 SERV-CORREO Spectrum Server 1.0 ESMTP ready
 cliente -> serv_correo SMTP Command: EHLO alfon
 serv_correo -> cliente SMTP Response: 250-SERV-CORREO
 cliente -> serv_correo SMTP Command: AUTH LOGIN
 serv_correo -> cliente SMTP Response: 334 VXNlcm5hbwu6
 cliente -> serv_correo SMTP Command: XXXzdGVtYXN4xxxhbmn1LmVz
 serv_correo -> cliente SMTP Response: 334 UGF2c3dvcmQ6
 cliente -> serv_correo SMTP Command: XXXETUFGXXXW3Q==
 serv_correo -> cliente TCP 25 > 3612 [ACK] Seq=243 Ack=71 win=65465 Len=0
 serv_correo -> cliente SMTP Response: 235 2.0.0 Authentication successful
 cliente -> serv_correo SMTP Command: MAIL FROM: <alfon@miservidor.com>
 serv_correo -> cliente SMTP Response: 250 2.1.0 Sender <alfon@miservidor.com> ok
 cliente -> serv_correo SMTP Command: RCPT TO: <destinatario@otroservidor.com>
 serv_correo -> cliente SMTP Response: 250 2.1.5 Recipient <alfon@miservidor.com> ok (local)
 cliente -> serv_correo SMTP Command: DATA
 serv_correo -> cliente SMTP Response: 354 Enter mail, end with CRLF.CRLF
 cliente -> serv_correo SMTP DATA fragment, 1460 bytes
 cliente -> serv_correo SMTP DATA fragment, 740 bytes
 serv_correo -> cliente TCP 25 > 3612 [ACK] Seq=411 Ack=3061 win=65535 Len=0
 cliente -> serv_correo SMTP EOM:
 serv_correo -> cliente TCP 25 > 3612 [ACK] Seq=411 Ack=4521 win=65535 Len=0
 serv_correo -> cliente SMTP Response: 250 2.0.0 48456cd9-0001e6ea Message accepted for delivery
 cliente -> serv_correo SMTP Command: QUIT
 serv_correo -> cliente SMTP Response: 221 2.0.0 SMTP closing connection
 serv_correo -> cliente TCP 25 > 3612 [FIN, ACK] Seq=505 Ack=6732 Win=65524 Len=0
 cliente -> serv_correo TCP 3612 > 25 [ACK] Seq=6732 Ack=506 win=64008 Len=0
 cliente -> serv_correo TCP 3612 > 25 [FIN, ACK] Seq=6732 Ack=506 win=64008 Len=0
 serv_correo -> cliente TCP 25 > 3612 [ACK] Seq=506 Ack=6733 win=65524 Len=0

TCP 3-way Handshake

Servidor: Preparado (220)

Cliente: Inicio de diálogo (EHLO)

Servidor: OK (250)

Srv: Solicita usuario y contraseña

Cli: Se los envía (cifrados)

Srv: OK (235)

Cli: FROM, Srv: OK

Cli: TO, Srv: OK

Mensaje

Fin mensaje

Cli: Salir

Srv: OK

Fin conexión TCP

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.

MIME (Multipurpose Internet Mail Protocol Extensions)

- **Cabeceras del 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 entrecomillada-imprimible.7.2.*

MIME (Multipurpose Internet Mail Protocol Extensions)

- **Content-Type:** La lista inicial de tipos y subtipos especificada por el RFC 1521 es:

<u>Tipo</u>	<u>Subtipo</u>	<u>Descripción</u>
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.

MIME (Multipurpose Internet Mail Protocol Extensions)

- **Content-Type:** Tipo *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 solicitando 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.

MIME (Multipurpose Internet Mail Protocol Extensions)

- **Content-Type:** Tipo *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 vídeo. 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.

MIME (Multipurpose Internet Mail Protocol Extensions)

- **Content-Type:** Tipo *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 vídeo de las películas.
- El subtipo ***digest*** se usa cuando se juntan muchos mensajes en un mensaje compuesto.

POP3 (RFC 1939)

- **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 (RFC 1939)

- 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.

POP3 (RFC 1939)

- **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.

POP3 (RFC 1939)

- 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 (RFC 2060)

- **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)

IMAP (RFC 2060)

- 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

- 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.

IMAP es más rápido

POP3 vs IMAP4

- 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.

POP3 vs IMAP4

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

TEMA 2. Servicios y Protocolos en Internet

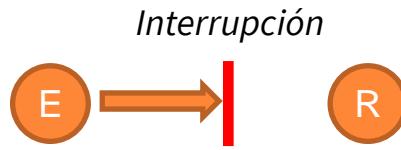
- 2.1. Introducción a las aplicaciones de red
- 2.2. Servicio de Nombres de Dominio (DNS)
- 2.3. Navegación web
- 2.4. Correo electrónico
- **2.5. Protocolos seguros**
- 2.6. Aplicaciones multimedia
- 2.7. Aplicaciones para interconectividad de redes locales
- 2.8. Cuestiones y ejercicios

Introducción

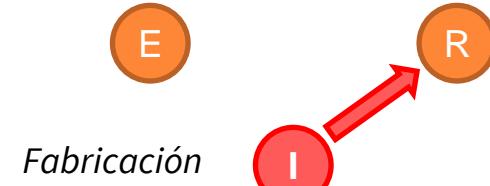
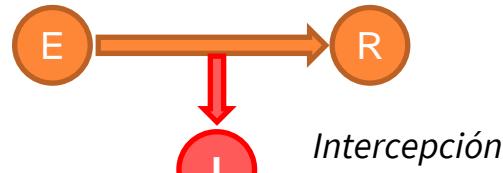
- Una **red de comunicaciones es segura** cuando se **garantizan todos los aspectos** de la misma ⇔ no hay protocolos ni redes 100% seguros.
- ¿Qué es seguridad? → múltiples aspectos:
 - **Confidencialidad/privacidad**: el contenido de la información es comprensible sólo por entidades autorizadas.
 - **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.

Introducción

- ¿En qué **nivel/capa** se debe situar la **seguridad**? en **TODOS**... el grado de seguridad *lo fija el punto más débil*.
- **Ataque de seguridad**: cualquier **acción** intencionada o no que **menoscaba** cualquiera de los **aspectos de la seguridad**.



TIPOS DE ATAQUES



Introducción

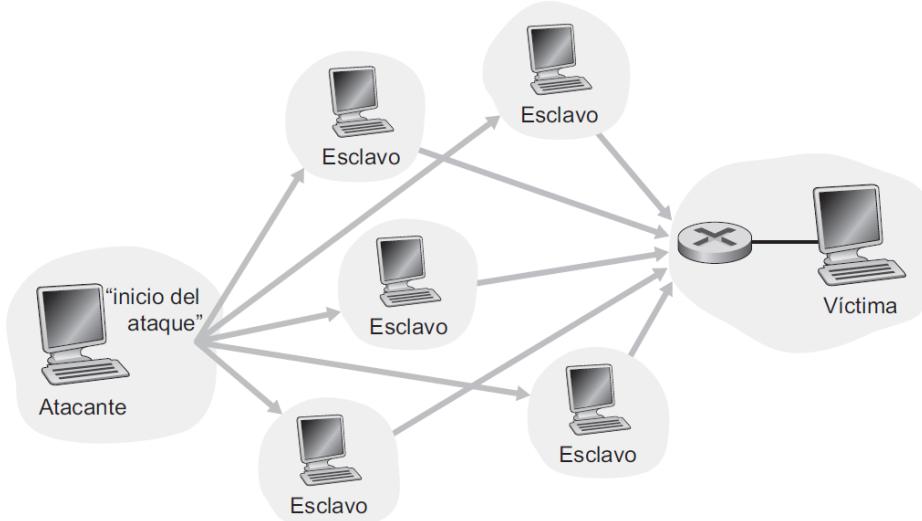
- **Ejemplos de Tipos de ataques:**

- ***Sniffing*** → vulneración a la confidencialidad, escuchar (husmear). **[Intercepción]**
- ***Poofing (phishing)*** → suplantación de la identidad de entidades. **[Fabricación]**
- ***Man in the Middle (MitM)*** → hombre en medio. **[Intercepción/Modificación]**
- ***Distributed Denial of Service (DDoS)*** → denegación de servicio distribuido, Ej: *Flooding* (inundación) **[Interrupción]**
- ***Malware*** → troyanos, gusanos, spyware, backdoors, rootkits, ransomware, keyloggers

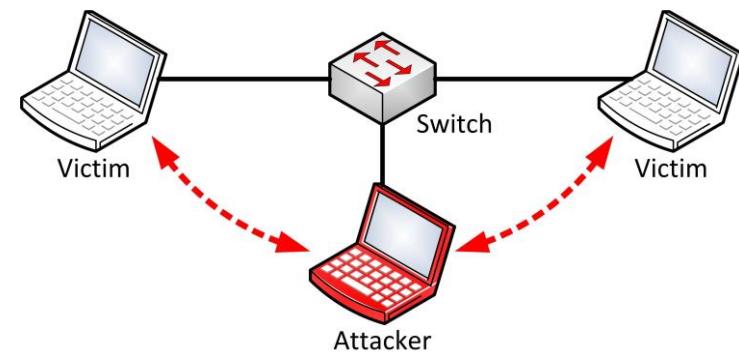
Introducción

EJEMPLOS DE ATAQUES

Ataque DDoS



Ataque MitM



Introducción

MECANISMOS DE SEGURIDAD

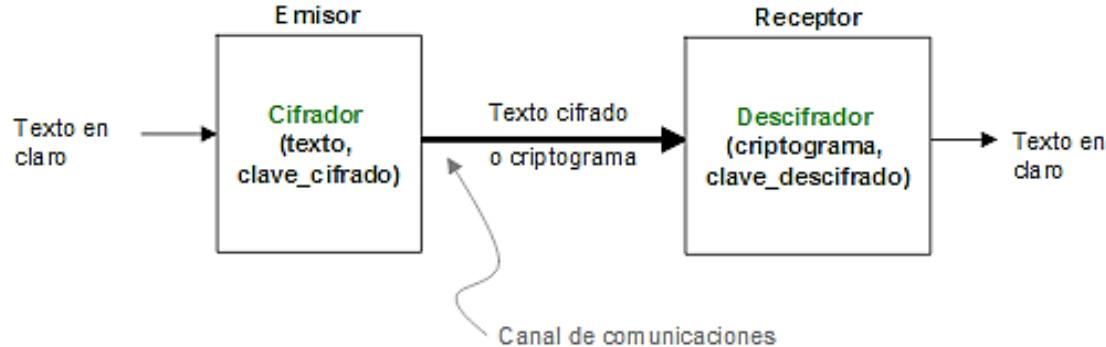
- De **prevención**:
 - mecanismos de autenticación e identificación.
 - mecanismos de control de acceso.
 - mecanismos de separación (física, temporal, lógica, criptográfica y fragmentación).
 - mecanismos de seguridad en las comunicaciones (cifrado de la información).
- De **detección**:
 - IDS (Intruder Detection System)
- De **recuperación**:
 - copias de seguridad (backup).
 - mecanismos de análisis forense: averiguar alcance, las actividades del intruso en el sistema y cómo entró.

Mecanismos de seguridad

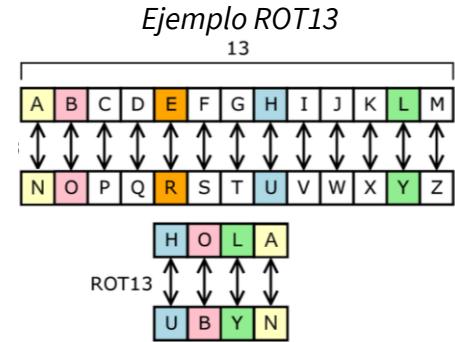
- **Mecanismos más utilizados:**
 - Cifrado (simétrico y asimétrico)
 - Autenticación con clave secreta (reto-respuesta)
 - Intercambio de Diffie-Hellman (establecimiento de clave secreta)
 - Funciones Hash. Hash Message Authentication Code (HMAC).
 - Firma Digital.
 - Certificados digitales.

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



- Tipos de criptosistema:
 - **Simétricos** o de clave privada (DES, *Data Encryption Standard*)
 - **Asimétricos** o de clave pública (RSA, *Rivest-Shamir-Adleman*)



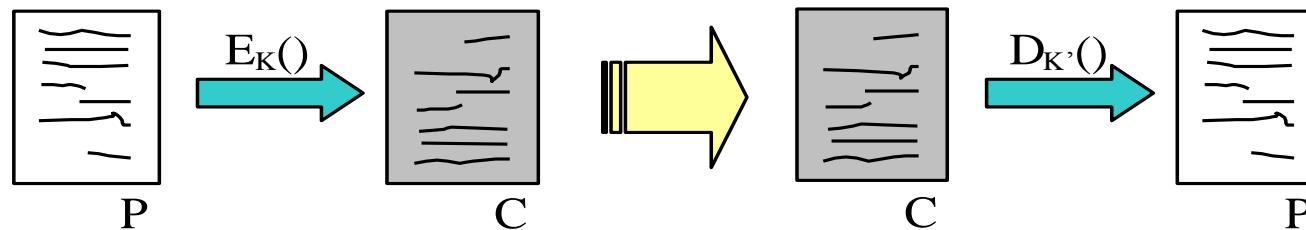
Ejemplo ASCII

A	01000001
B	01000010
C	01000011
D	01000100
E	01000101
F	01000110
G	01000111
H	01001000
I	01001001
J	01001010

Mecanismos de seguridad - Cifrado

- El **cifrado** es un procedimiento para garantizar la **confidencialidad**:
 - Se parte de un **Texto llano/claro** (*plain text*)
 - Se aplica un **algoritmo de cifrado** conocido como $E_K()$
 - Y un **algoritmo de descifrado** llamado $D_{K'}()$
 - Ambos **dependen** respectivamente de una **clave de cifrado K** y de una **clave de descifrado K'** .

Cifrado \Leftrightarrow Encriptación
 Descifrado \Leftrightarrow
 Desencriptación



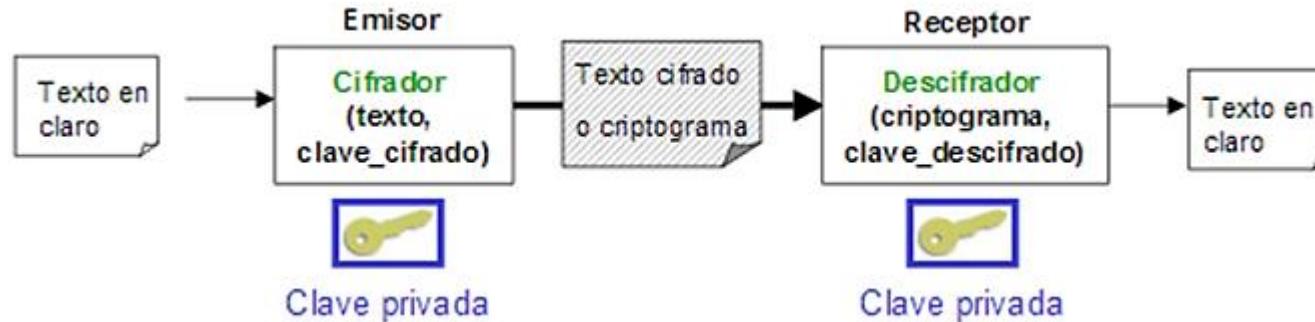
- El texto plano P se cifra y se convierte en C, se transmite y posteriormente se descifra C para obtener P de nuevo.

Mecanismos de seguridad - Cifrado

CIFRADO SIMÉTRICO – ALGORITMOS DE CLAVE SECRETA

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

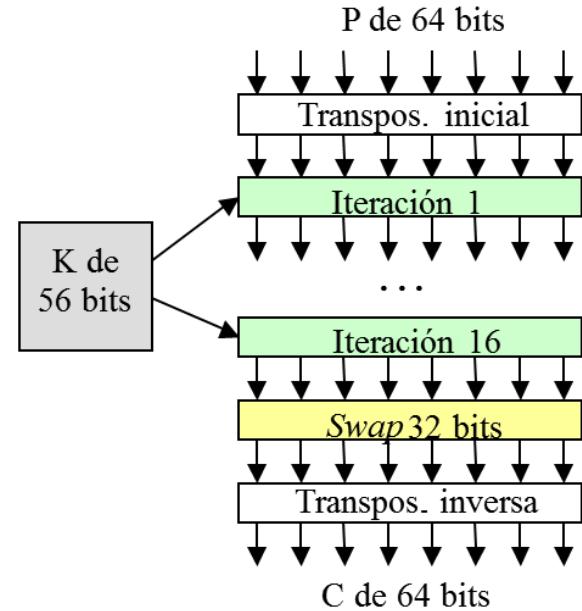
$$K = K'$$



Mecanismos de seguridad - Cifrado

CIFRADO SIMÉTRICO – ALGORITMOS DE CLAVE SECRETA

- Algoritmo **DES** (*Data Encryption Standard, IBM 1975*):
 - Se hace una transposición al bloque inicial de bits P
 - 16 iteraciones aplicando la clave K de 56 bits
[ver transparencia siguiente]
 - Intercambio de 32 bits de orden más alto por los más bajos
 - Transposición inversa de 1)

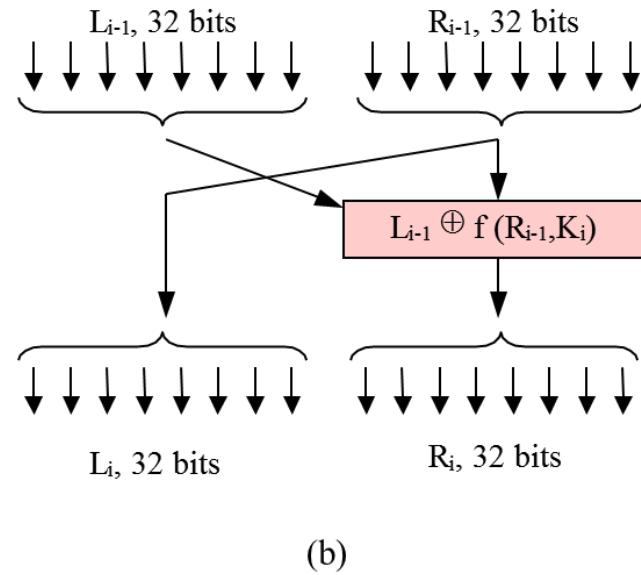


(a)

Mecanismos de seguridad - Cifrado

CIFRADO SIMÉTRICO – ALGORITMOS DE CLAVE SECRETA

- Algoritmo **DES** (*Data Encryption Standard, IBM 1975*):
 - 16 iteraciones aplicando la clave K de 56 bits (cada iteración (b))
 - 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.



(b)

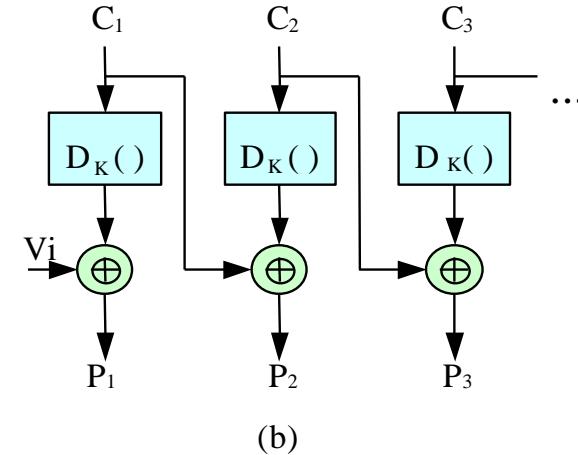
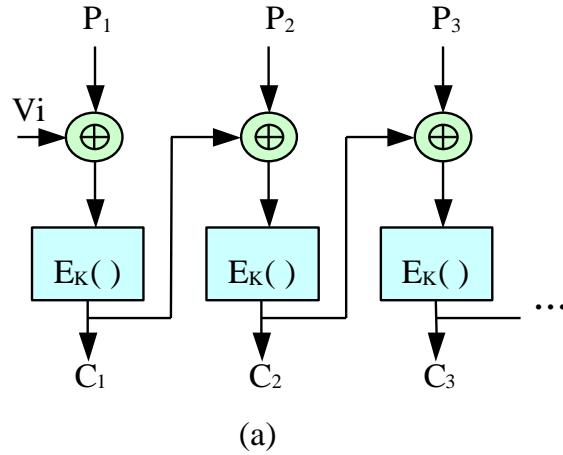
Mecanismos de seguridad - Cifrado

CIFRADO SIMÉTRICO – ALGORITMOS DE CLAVE SECRETA

- 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.

DES: Complejidad para descubrir la clave “sólo” 2^{56}

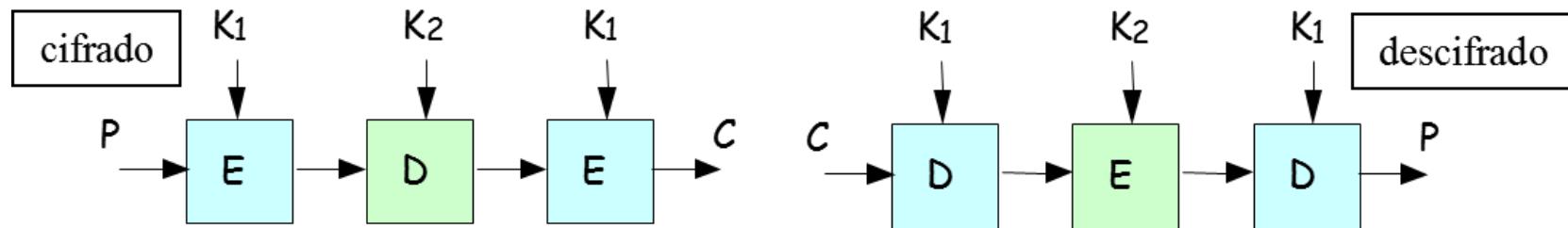


Mecanismos de seguridad - Cifrado

CIFRADO SIMÉTRICO – ALGORITMOS DE CLAVE SECRETA

- **3DES:**

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



Mecanismos de seguridad - Cifrado

CIFRADO SIMÉTRICO – ALGORITMOS DE CLAVE SECRETA

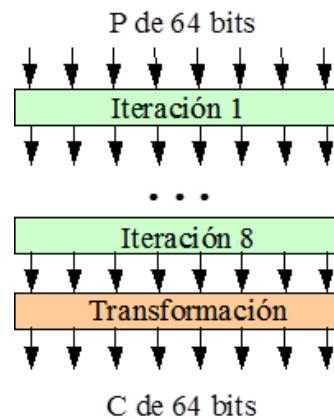
- **IDEA** (*International Data Encryption Algorithm*):

- Utiliza claves de 128 bits.
- Puede operar en tiempo real.
- Fácil de implementar en hardware.
- 8 iteraciones
- Aplica operaciones:

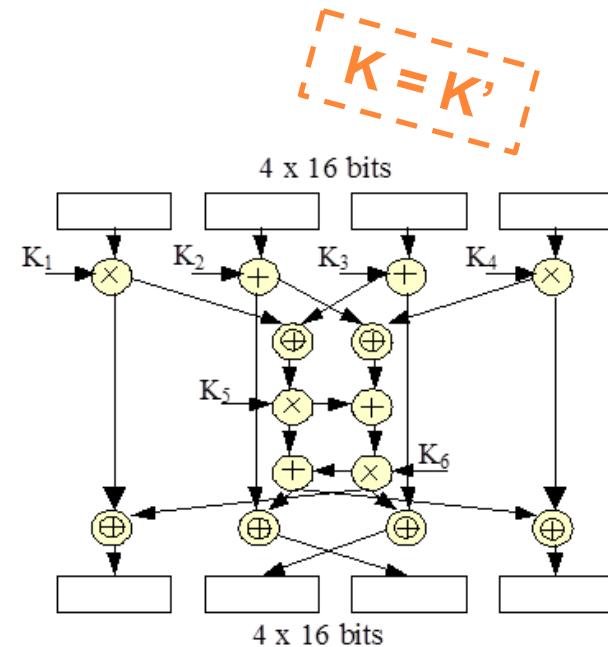
XOR

Suma módulo 2^{16}

Multiplicaciones módulo $2^{16}+1$



(a)



(b)

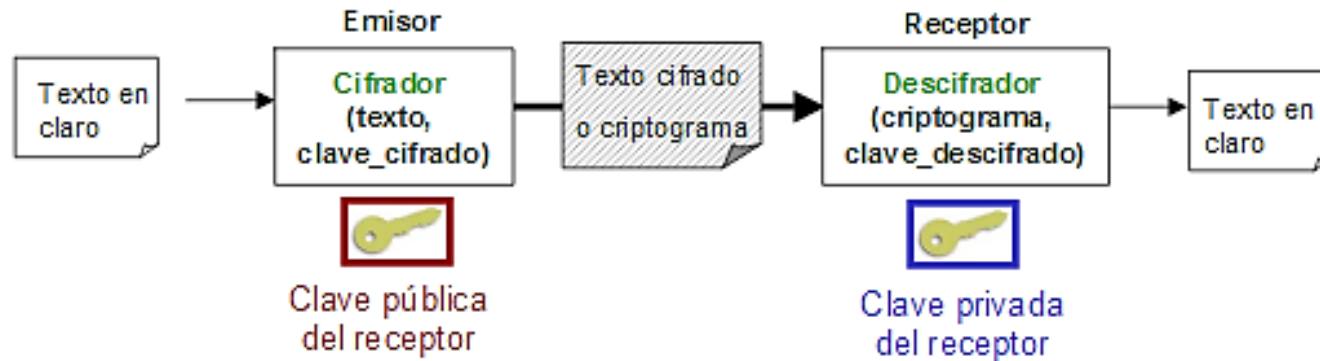
https://es.wikipedia.org/wiki/International_Data_Encryption_Algorithm

Mecanismos de seguridad - Cifrado

CIFRADO ASIMÉTRICO – ALGORITMOS DE CLAVE PÚBLICA

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

$K \neq K'$



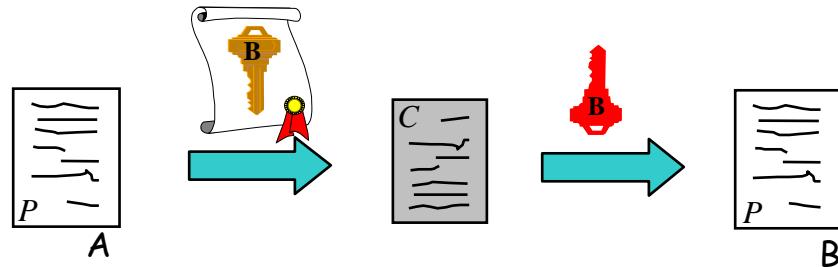
Mecanismos de seguridad - Cifrado

CIFRADO ASIMÉTRICO – ALGORITMOS DE CLAVE PÚBLICA

- **Dos claves por usuario (B):** una **pública** K_{PUB_B} y otra **privada** K_{PRI_B} **distintas**
- Conocida K_{PUB_B} es **imposible conocer** K_{PRI_B}
- **Claves diferentes** para **cifrar y descifrar**:

Cifrar → $C = E_{K_{pubB}}(P)$

Descifrar: $P = D_{K_{priB}}(C)$



- ¿Y si enviamos $C=E_{K_{privA}}(P)$? → **autenticación**

$$K \neq K'$$

Mecanismos de seguridad - Cifrado

CIFRADO ASIMÉTRICO – ALGORITMOS DE CLAVE PÚBLICA

- **RSA** (*Rivest, Shamir, Adleman*)

$K \neq K'$

- 1) Elegimos p y q primos grandes ($>10^{100}$)
- 2) $n = (p \cdot q)$ y $z = (p-1) \cdot (q-1)$ (función de Euler)
- 3) Elegimos d coprimo con z (no tienen factores primos en común)
- 4) Calculamos e tal que $e \cdot d \bmod z = 1$ (algoritmo de Euclides)
- 5) $K_{pub} = (e, n)$ y $K_{pri} = (d, n)$, de modo que:

$$* C = P^e \bmod n$$

$$* P = C^d \bmod n$$

<https://es.wikipedia.org/wiki/RSA>

Mecanismos de seguridad - Cifrado

CIFRADO ASIMÉTRICO – ALGORITMOS DE CLAVE PÚBLICA

- EJEMPLO RSA

$$p = 3, q = 11$$

$$n = p \cdot q = 33, \quad z = (n-1) \cdot (p-1) = 20 \quad (= 5 \cdot 2 \cdot 2 \text{ en factores primos})$$

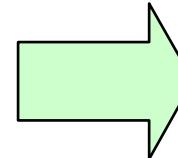
d = 7, coprimo respecto a z

$$e = 3, e \cdot d \bmod z = 1$$

$$K_{pub} = (3, 33) \text{ y } K_{pri} = (7, 33)$$

$$\begin{aligned} C &= P^e \bmod n \\ P &= C^d \bmod n \end{aligned}$$

Simbólico	Numérico	P^3	$P^3 \bmod 33$	C^7	$C^7 \bmod 33$	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

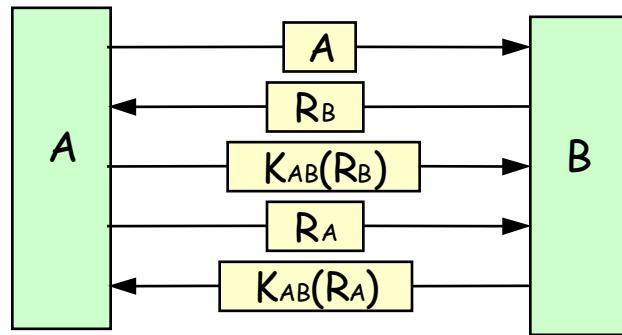


Mecanismos de seguridad - Autenticación

AUTENTICACIÓN Y CIFRADO DE 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:**

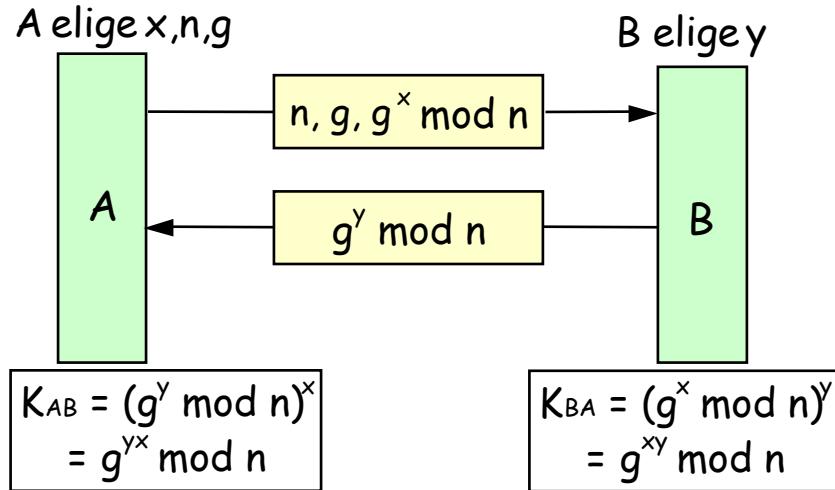
- La respuesta es la contraseña → ataque replay
- Contraseña con identificador → ataque replay con id
- Contraseña de un solo uso

Mensaje nonce
(sólo se genera una vez)

Mecanismos de seguridad – Clave secreta

ESTABLECIMIENTO DE CLAVE SECRETA

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



EJEMPLO:

$$g=7, n=23$$

1. A elige $x = 3$ y calcula $R1 = 7^3 \text{ mod } 23 = 21$.
2. A envía el número 21 a B.
3. B elige $y = 6$ y calcula $R2 = 7^6 \text{ mod } 23 = 4$.
4. B envía el número 4 a A.
5. A calcula la clave privada/simétrica $K = 4^3 \text{ mod } 23 = 18$.
6. B calcula la clave privada/simétrica $K = 21^6 \text{ mod } 23 = 18$.

El valor de K es el mismo para A y B:

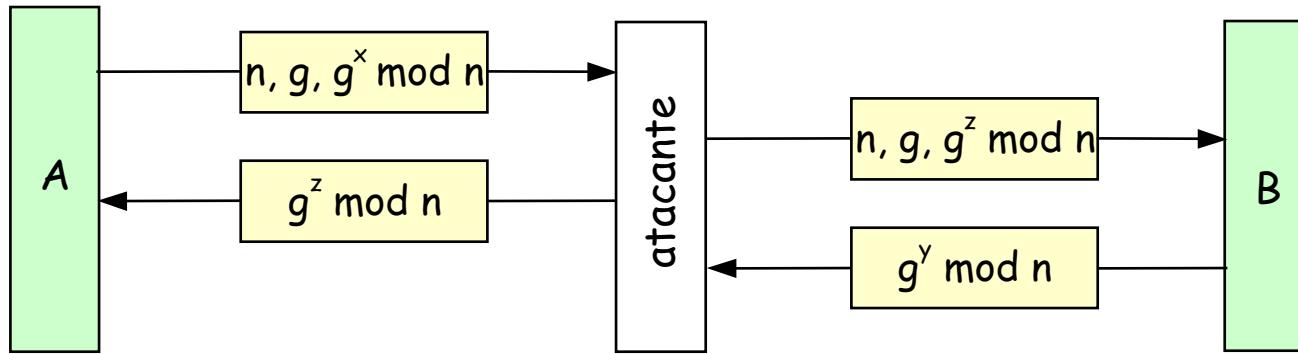
$$g^{xy} \text{ mod } n = 7^{18} \text{ mod } 23 = 18$$

Usando números grandes no es vulnerable a escucha del canal

Mecanismos de seguridad – Clave secreta

ESTABLECIMIENTO DE CLAVE SECRETA

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



Vulnerable a ataque MitM

Mecanismos de seguridad – 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' tal que

$$M \neq M' \text{ y } H(M) = H(M')$$

- Ejemplos de funciones HASH: MD5, SHA-1, SHA-512
- Las funciones **Hash** se usan para **garantizar integridad + autenticación** (clave K):

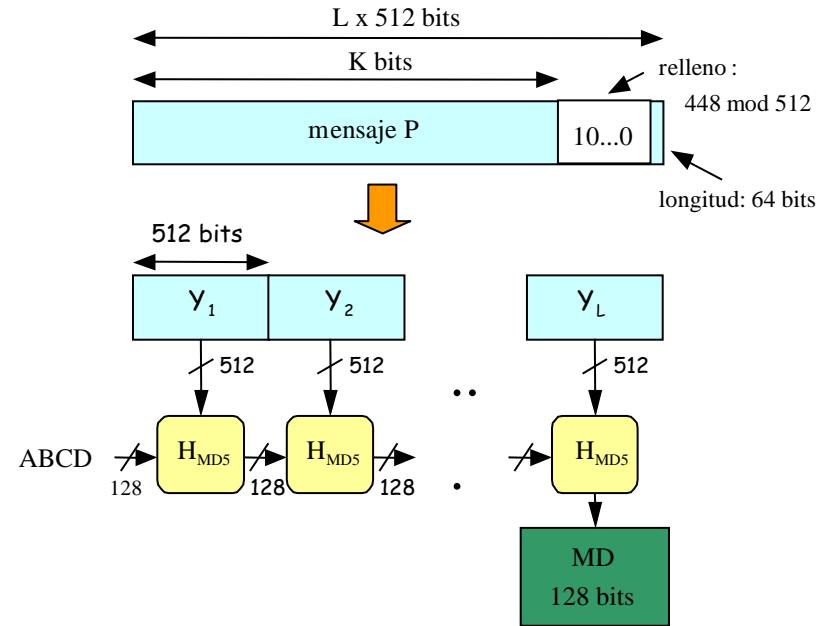
Hash Message Authentication Code (HMAC): $M + H(K|M)$ pero para evitar ataques de extensión se usa $M + H(K | H(K | M))$

Mecanismos de seguridad – Funciones Hash

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.

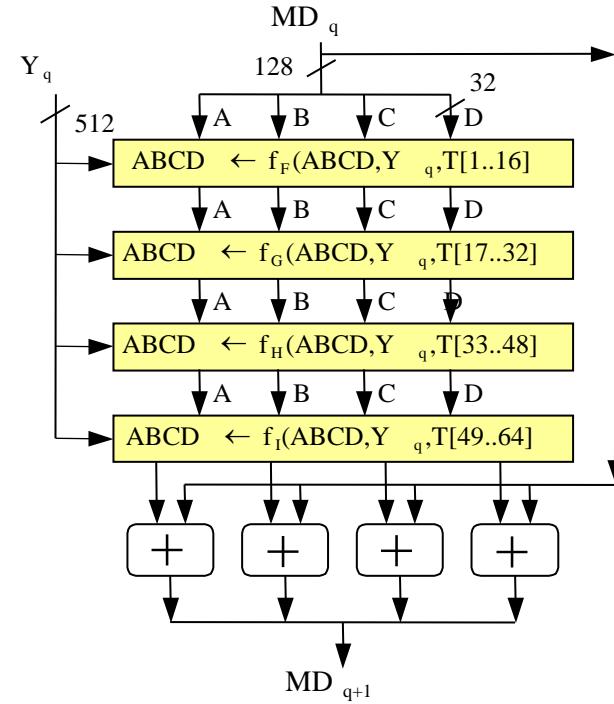
ABCD son 4 registros de 32 bits con valores constantes hexadecimales



Mecanismos de seguridad – Funciones Hash

MD5 (Message Digest 5, RFC 1321)

- 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.
 - Se hacen desplazamientos de bits.
 - Se hacen varias pasadas.
 - Se hace una suma final módulo 2^{32} .
 - La salida de un bloque será la entrada del siguiente.

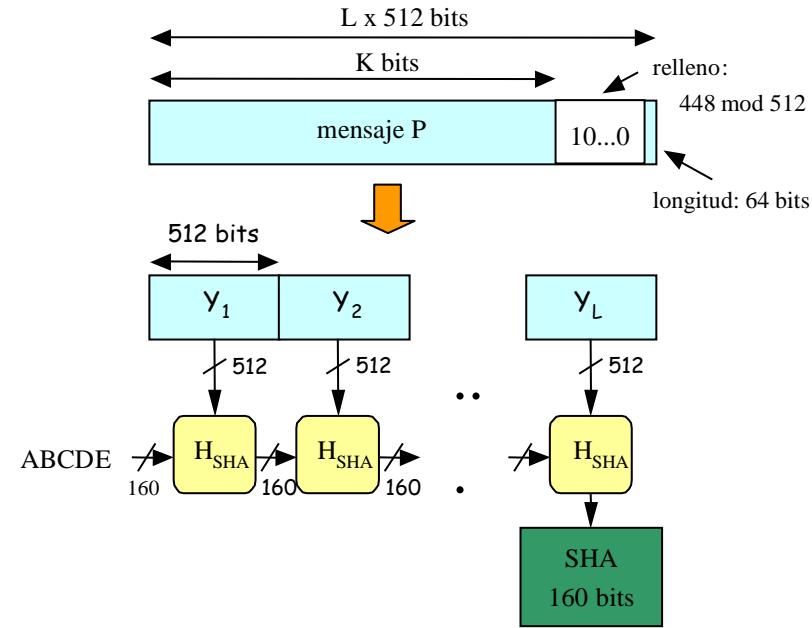


Mecanismos de seguridad – Funciones Hash

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.

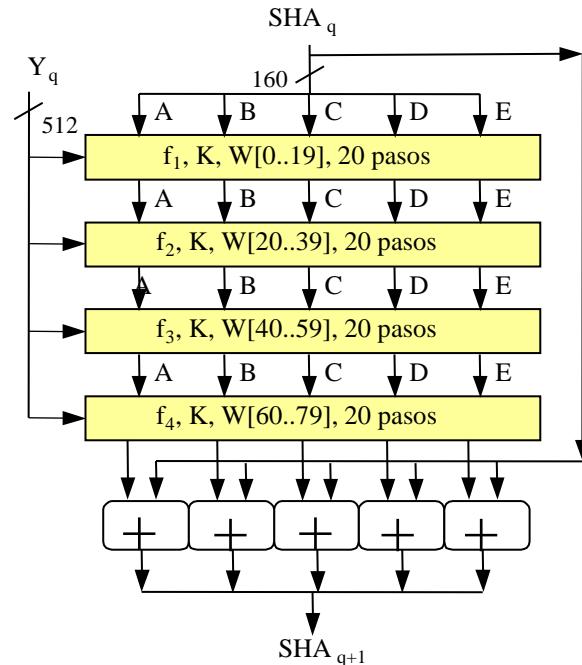
ABCDE son 5 registros de 32 bits con valores constantes hexadecimales (diferentes de los de MD5)



Mecanismos de seguridad – Funciones Hash

SHA-1 (Secure Hash Algorithm 1, RFC 3174)

- 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 2^{32} .
 - La salida de un bloque será la entrada del siguiente.

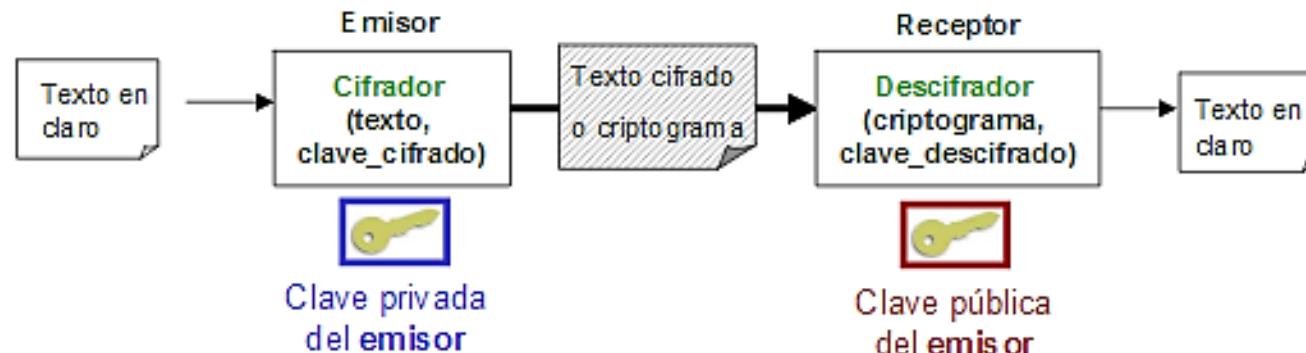


Mecanismos de seguridad – Firma Digital

- 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

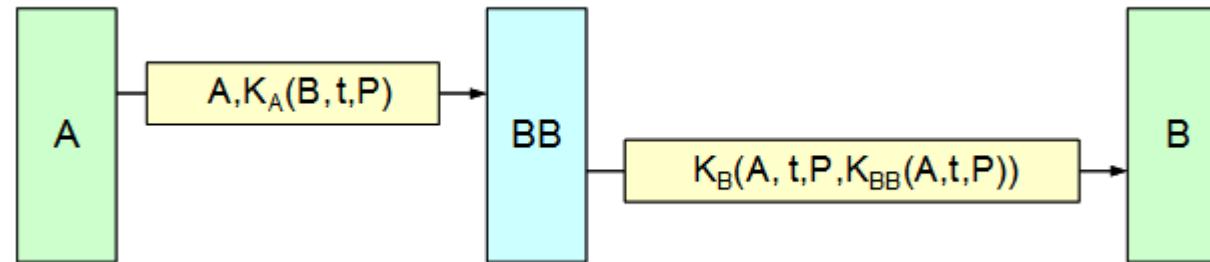
- Que el **receptor** pueda **autenticar al emisor**.
- Que **no** haya **repudio** (que el emisor no pueda alegar que él no envió el mensaje).
- Que el **emisor** tenga **garantías de no falsificación** de su mensaje (integridad).



Mecanismos de seguridad – Firma Digital

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, K_A , 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, K_B , 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 K_{BB} .

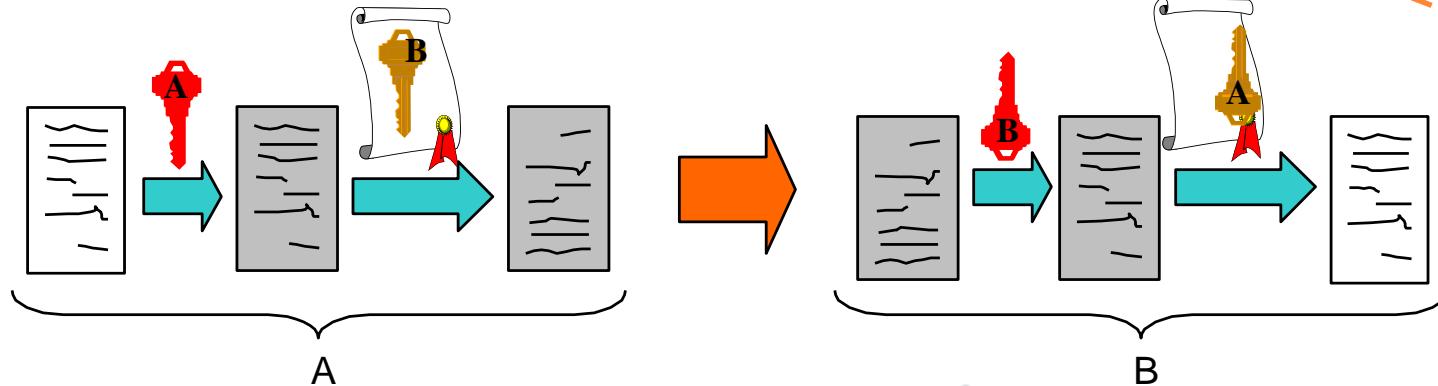


Mecanismos de seguridad – Firma Digital

FIRMA DIGITAL CON CLAVE ASIMÉTRICA. DOBLE CIFRADO

- Uno para proporcionar privacidad, con K_{pubB}
- Otro, previo, para autenticación, con K_{priA}
- Para firmar, enviar $K_{\text{pubB}}(K_{\text{priA}}(P))$
- En el receptor $K_{\text{pubA}}(K_{\text{priB}}(K_{\text{pubB}}(K_{\text{priA}}(P))))=P$

PROBLEMA
 Garantizar el no repudio:
 Asociación fehaciente e
 indisoluble de A con su clave
 pública K_{pubA}



Mecanismos de seguridad – 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 (AC)

- **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 .
- **Formato** de certificados: principalmente **X.509**.

Mecanismos de seguridad – Certificados digitales

AUTORIDADES DE CERTIFICACIÓN (AC)

- 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)
 - VeriSign (www.verisign.com)
 - CAMERFIRMA (www.camerfirma.es)
 - CERES (www.cert.fnmt.es)

*No es lo mismo Firma Digital
(un uso en una transmisión)
que Certificado Digital (muchos usos,
acreditar nuestra identidad)*

Mecanismos de seguridad – Certificados digitales

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.

Mecanismos de seguridad – Certificados digitales

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

Mecanismos de seguridad – Certificados digitales

CERTIFICADO X.509

```

Certificate:
  Data:
    Version: 1 (0x0)
    Serial Number: 7829 (0x1e95)
    Signature Algorithm: md5WithRSAEncryption
    Issuer: C=ZA, ST=Western Cape, L=Cape Town, O=Thawte Consulting cc,
             OU=Certification Services Division,
             CN=Thawte Server CA/Email=server-certs@thawte.com
    Validity
      Not Before: Jul  9 16:04:02 1998 GMT
      Not After : Jul  9 16:04:02 1999 GMT
    Subject: C=US, ST=Maryland, L=Pasadena, O=Brent Baccala,
             OU=FreeSoft, CN=www.freesoft.org/Email=baccala@freesoft.org
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (1024 bit)
        Modulus (1024 bit):
          00:b4:31:98:0a:c4:bc:62:c1:88:aa:dc:b0:c8:bb:
          33:35:19:d5:0c:64:b9:3d:41:b2:96:fc:f3:31:e1:
          66:36:d0:8e:56:12:44:ba:75:eb:e8:1c:9c:5b:66:
          70:33:52:14:c9:ec:4f:91:51:70:39:de:53:85:17:
          16:94:6e:ee:f4:d5:6f:d5:ca:b3:47:5e:1b:0c:7b:
          c5:cc:2b:6b:c1:90:c3:16:31:0d:bf:7a:c7:47:77:
          8f:a0:21:c7:4c:d0:16:65:00:c1:0f:d7:b8:80:e3:
          d2:75:6b:c1:ea:9e:5c:5c:ea:7d:c1:a1:10:bc:b8:
          e8:35:1c:9e:27:52:7e:41:8f
        Exponent: 65537 (0x10001)
    Signature Algorithm: md5WithRSAEncryption
    93:5f:8f:5f:c5:af:bf:0a:ab:a5:6d:fb:24:5f:b6:59:5d:9d:
    92:2e:4a:1b:8b:ac:7d:99:17:5d:cd:19:f6:ad:ef:63:2f:92:
    ab:2f:4b:cf:0a:13:90:ee:2c:0e:43:03:be:f6:ea:8e:9c:67:
    d0:a2:40:03:f7:ef:6a:15:09:79:a9:46:ed:b7:16:1b:41:72:
    0d:19:aa:ad:dd:9a:df:ab:97:50:65:f5:5e:85:a6:ef:19:d1:
    5a:de:9d:ea:63:cd:cb:cc:6d:5d:01:85:b5:6d:c8:f3:d9:f7:
    8f:0e:fc:ba:1f:34:e9:96:6e:6c:cf:f2:ef:9b:bf:de:b5:22:
    68:9f
  
```

Autoridad
Certificadora

Datos del
usuario

Algoritmo y
clave pública
del usuario

Algoritmo y
clave privada
de la AC

Implementación de mecanismos de seguridad

- **Seguridad perimetral**
 - Firewalls, IDS (sistemas detección intrusiones), IRS (sistemas respuesta intrusiones)

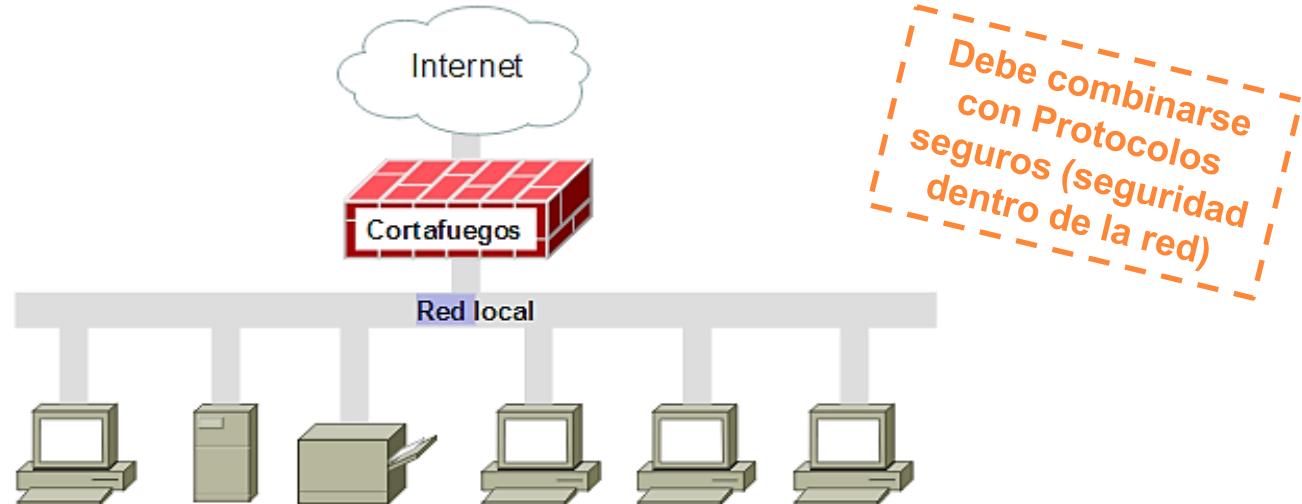
PROTOCOLOS DE SEGURIDAD

- **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...**

Implementación de mecanismos de seguridad

CORTAFUEGOS (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).



Implementación de mecanismos de seguridad

CORTAFUEGOS (*FIREWALL*) – 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.

Implementación de mecanismos de seguridad

CORTAFUEGOS (FIREWALL) – TÉCNICAS APLICADAS

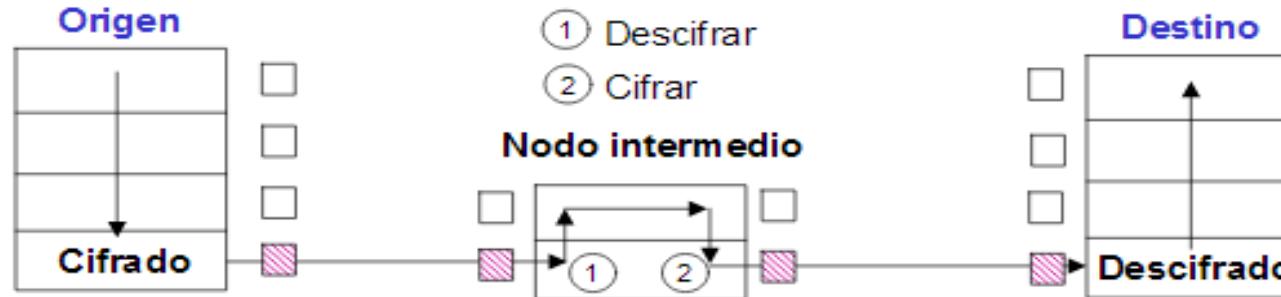
- **Filtrado de paquetes:**
 - Reglas que especifican qué tipos de paquetes pueden circular en cada sentido y cuáles se bloquearán.
 - Las reglas se basan en las cabeceras de los paquetes.
- **Servicios de proxy:**
 - Son aplicaciones especializadas que funcionan en un cortafuegos.
 - Hacen de intermediarios 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.

Implementación de mecanismos de seguridad

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

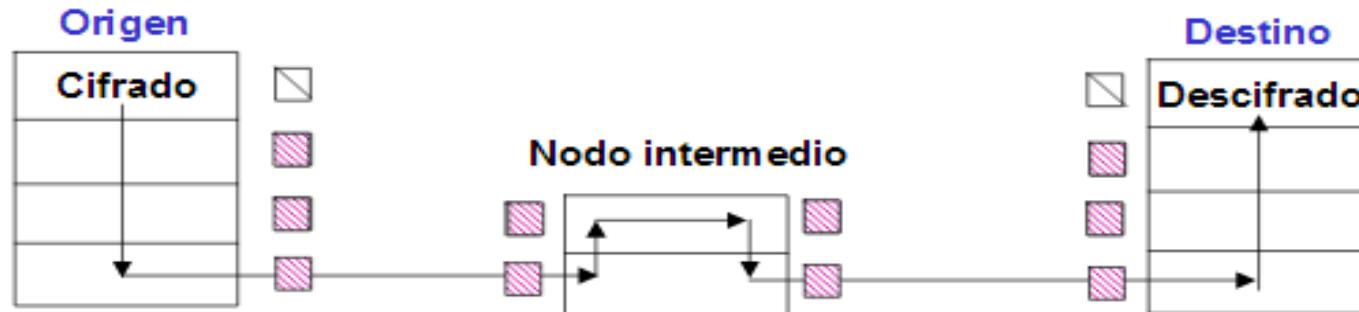


Implementación de mecanismos de seguridad

CIFRADO EN REDES

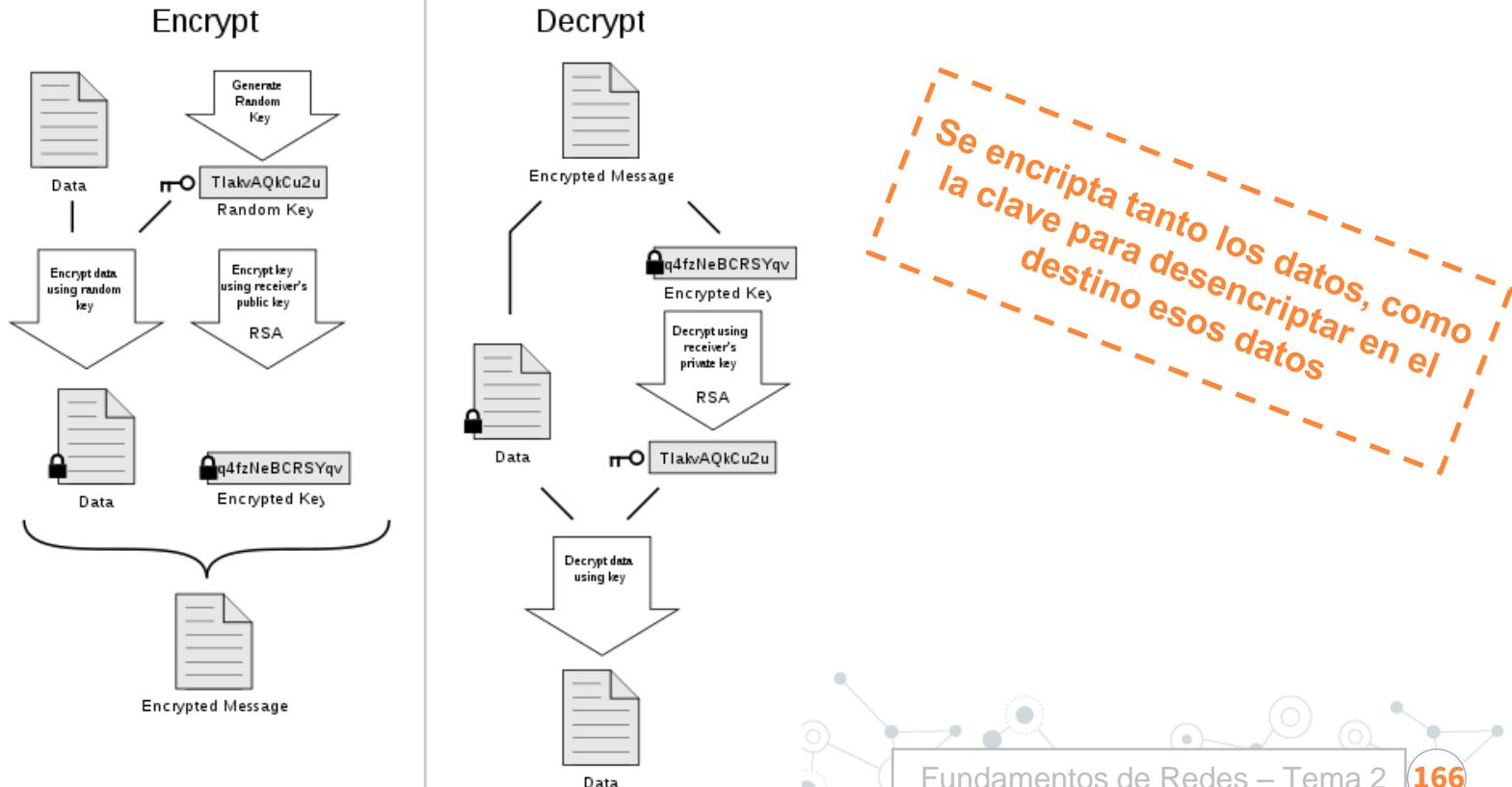
- **Cifrado extremo a extremo:**

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



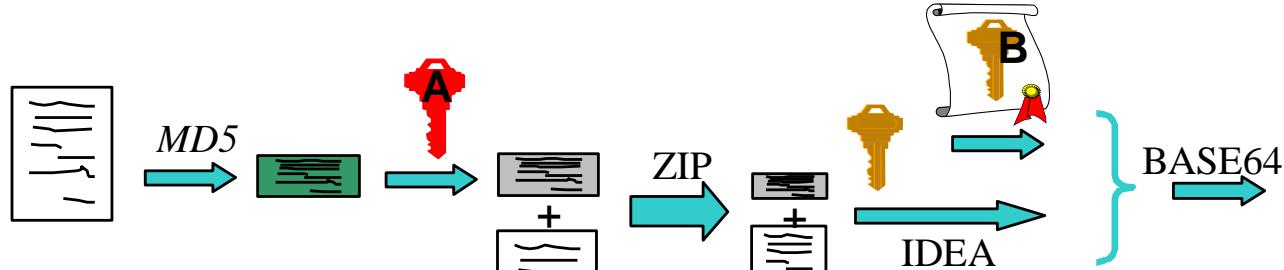
Implementación de mecanismos de seguridad

PRETTY GOOD PRIVACY (PGP) (Usado para correo electrónico seguro y otros documentos en Internet)



Implementación de mecanismos de seguridad

PRETTY GOOD PRIVACY (PGP) (Usado para correo electrónico seguro y otros documentos en Internet)



- I . Resumen/Hash (integridad)
- I . Firma Digital (identidad de A)
- I . Agrupar datos y comprimirlos
- I . Encriptar con clave aleat. K (seguridad)
- I . Encriptar con clave publica de B (confidencialidad)
- I . Codificación adicional B64

Emisor:

- $R = MD5(P)$
- $FD = K_{prA}(R)$
- $Z = ZIP(FD + P)$
- $C = IDEA_K(Z) + K_{puB}(K)$
- $M = B64(C)$

Receptor:

- $C = B64^{-1}(M)$
- $K = K_{prB}(K_{puB}(K))$
- $Z = IDEA_K^{-1}(IDEA_K(Z))$
- $FD + P = ZIP^{-1}(Z)$
- $R = K_{puA}(FD)$
- $R' = MD5(P)$
- $R' = R ??$

Implementación de mecanismos de seguridad

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.

Implementación de mecanismos de seguridad

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.

Implementación de mecanismos de seguridad

TRANSPORT LAYER SECURITY (SSL/TLS)

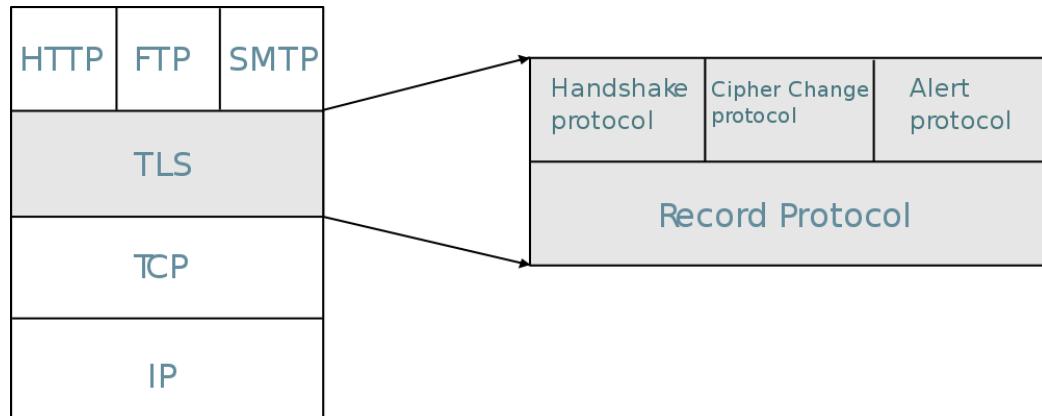
- **SSL (Secure Socket Layer)** → Desarrollado por Netscape en 1994 y puesto en dominio público para la **definición de canales seguros sobre TCP**.
- **TLS (Transport Layer Security)** → Sucesor y **mejora sobre SSL**.
 - Corrige **vulnerabilidades** de SSL y **permite la autenticación** de emisor y 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

Implementación de mecanismos de seguridad

TRANSPORT LAYER SECURITY (SSL/TLS) – Capas

- **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 cifrada con K_{PUB_SERVER} ó
 - Diffie-Hellman
- **SSL Alert protocol**
 - Informa sobre errores en la sesión
- **Change Cipher Espec Protocol**
 - Para notificar cambios en el cifrado



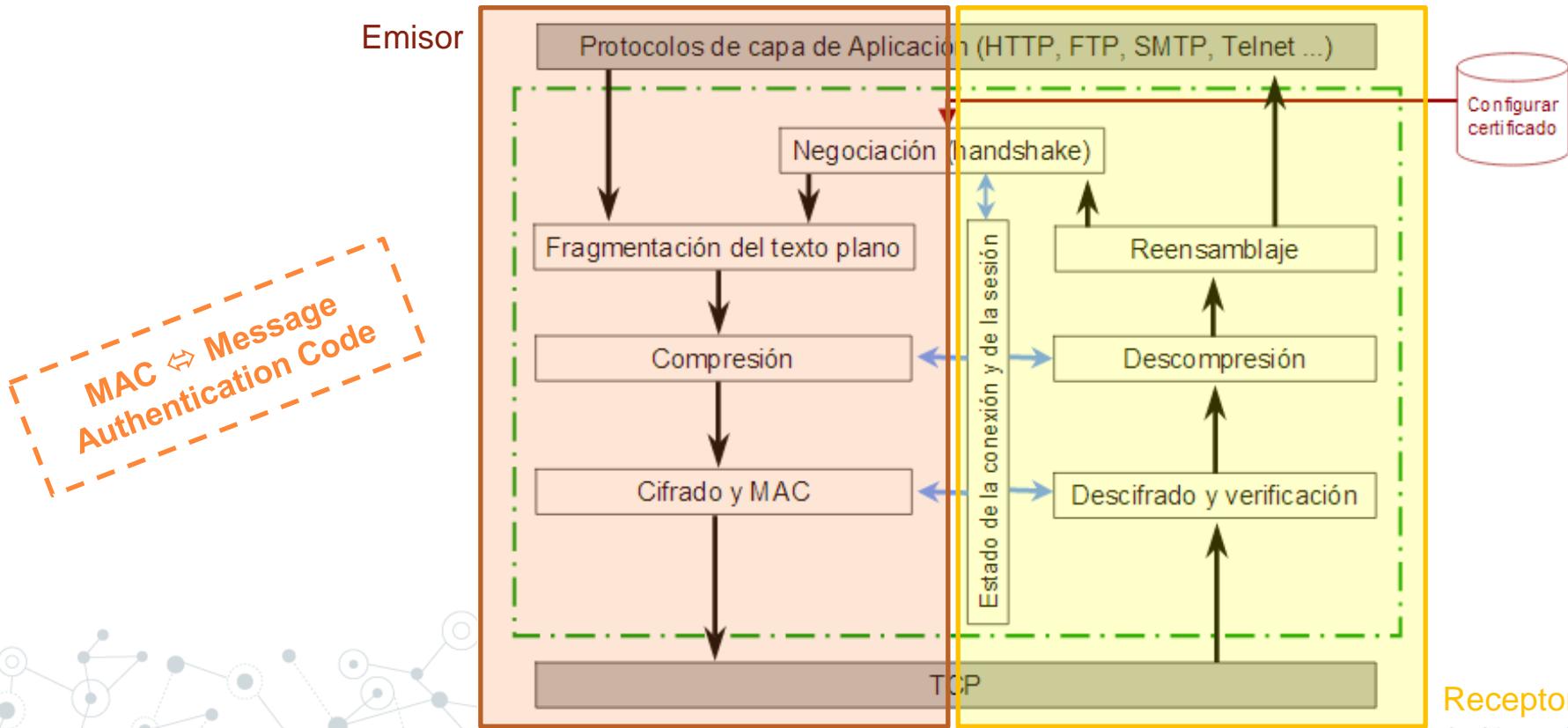
Implementación de mecanismos de seguridad

TRANSPORT LAYER SECURITY (SSL/TLS) – Funcionamiento

- 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.
- El **cliente elegirá** un **sistema criptográfico** y **verificará** la **clave pública del servidor**.
- Entonces se **generará** una **clave privada** (de uso único) cifrada con la clave del servidor.
Si alguien pudiese descifrar la información, sólo conseguiría romper esa conexión/sesión, ya que una sesión posterior requeriría una clave privada diferente.
- 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).

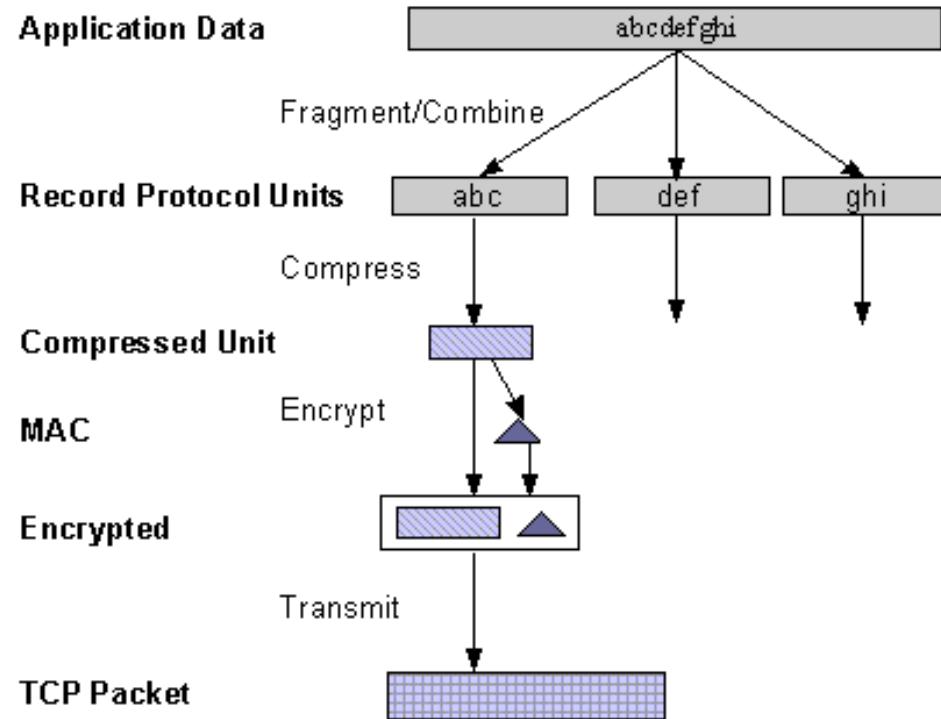
Implementación de mecanismos de seguridad

TRANSPORT LAYER SECURITY (SSL/TLS) – Arquitectura



Implementación de mecanismos de seguridad

TRANSPORT LAYER SECURITY (TLS/SSL)



Implementación de mecanismos de seguridad

TRANSPORT LAYER SECURITY (SSL/TLS)

- 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	646	LDAP sobre SSL
telnets	992	TELNET sobre SSL
imaps	993	IMAP sobre SSL
ircs	994	IRC sobre SSL
pop3s	995	POP3 sobre SSL
ftps-data	989	FTP-Datos sobre SSL
ftps-control	990	FTP-Control sobre SSL

Implementación de mecanismos de seguridad

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 en 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)

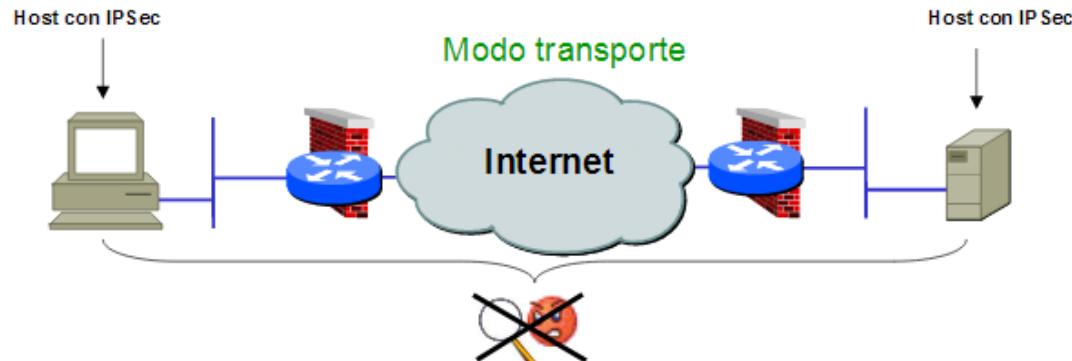
Implementación de mecanismos de seguridad

IPSec (IP Security)

- 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



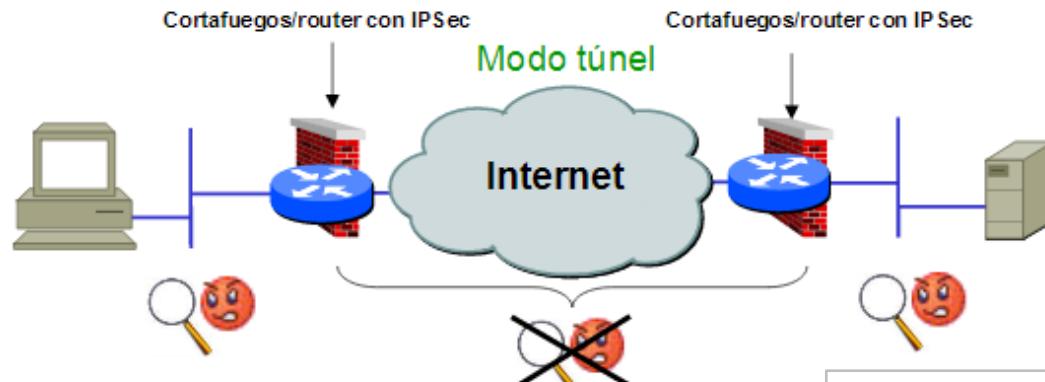
Implementación de mecanismos de seguridad

IPSec

- IPSec tiene 2 modos de operación:

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

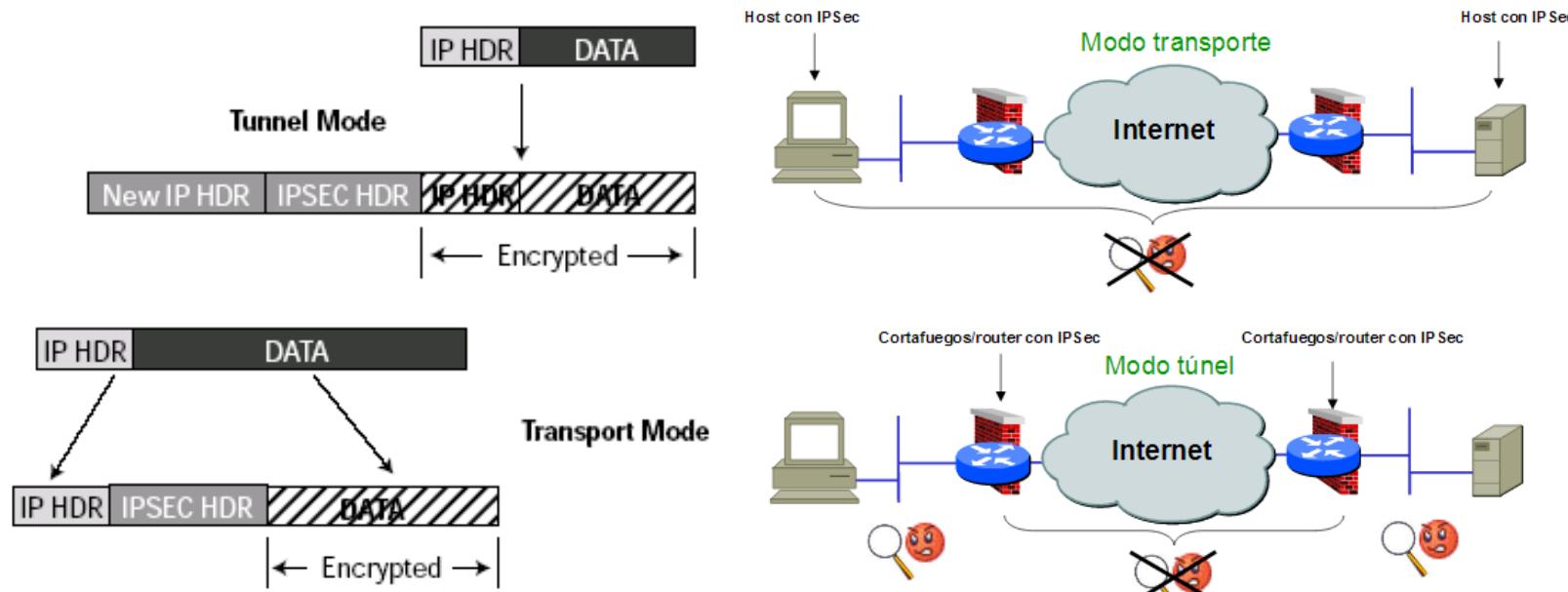
- se protegen paquetes IP (capa de red)
- para la comunicación segura entre routers/gateways de seguridad sólo se puede usar este modo
- permite incorporar IPSec sin afectar a los hosts
- se integra fácilmente con VPNs



Implementación de mecanismos de seguridad

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 2. Servicios y Protocolos en Internet

- 2.1. Introducción a las aplicaciones de red
- 2.2. Servicio de Nombres de Dominio (DNS)
- 2.3. Navegación web
- 2.4. Correo electrónico
- 2.5. Protocolos seguros
- **2.6. Aplicaciones multimedia**
- 2.7. Aplicaciones para interconectividad de redes locales
- 2.8. Cuestiones y ejercicios

Definiciones

- Para A. Bartolomé (1994):

“Los sistemas Multimedia, en el sentido que hoy se da al término, son básicamente sistemas interactivos con múltiples códigos”

- Según Fred Hoffstetter:

“Multimedia es el uso del ordenador para presentar y combinar: texto, gráficos, audio y vídeo con enlaces que permitan al usuario navegar, interactuar, crear y comunicarse”.

Concepto de aplicación multimedia

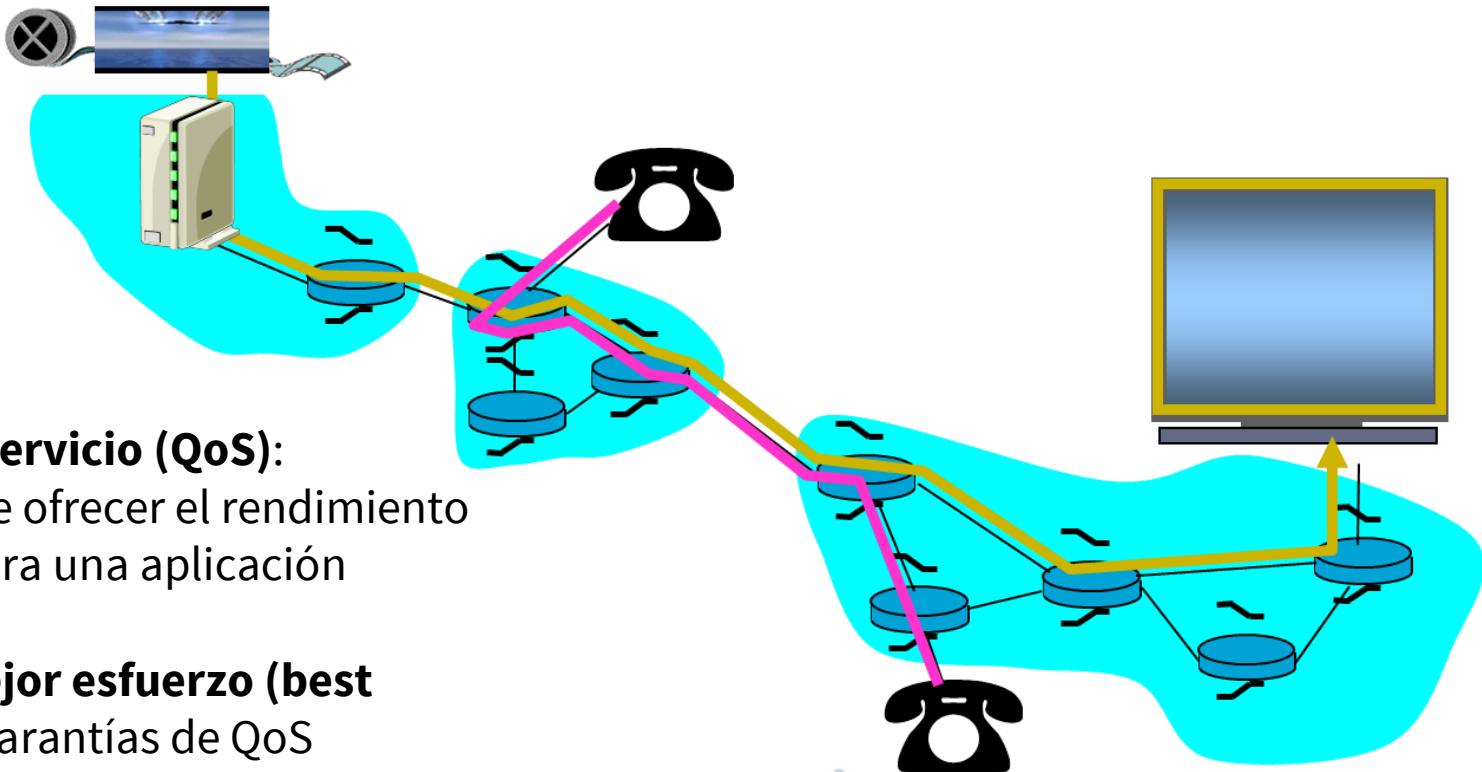
- El término **multimedia** hace referencia al **uso combinado** de **diferentes medios de comunicación**: texto, imagen, sonido, animación y video.
- Los **programas informáticos** que **utilizan de forma combinada** y coherente con sus objetivos **diferentes medios**, y **permiten la interacción** 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 coheramente.
 - **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



Aplicaciones multimedia en la red

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).

Problemas habituales

Network Delivery Issues

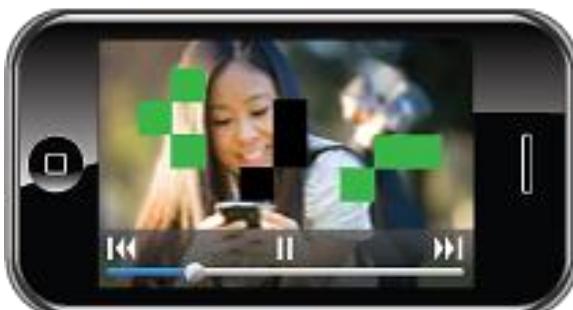


Stalling

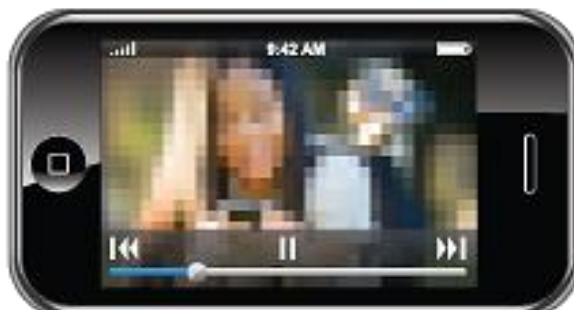
Low Quality Source or Overly Aggressive Optimization



Blurriness

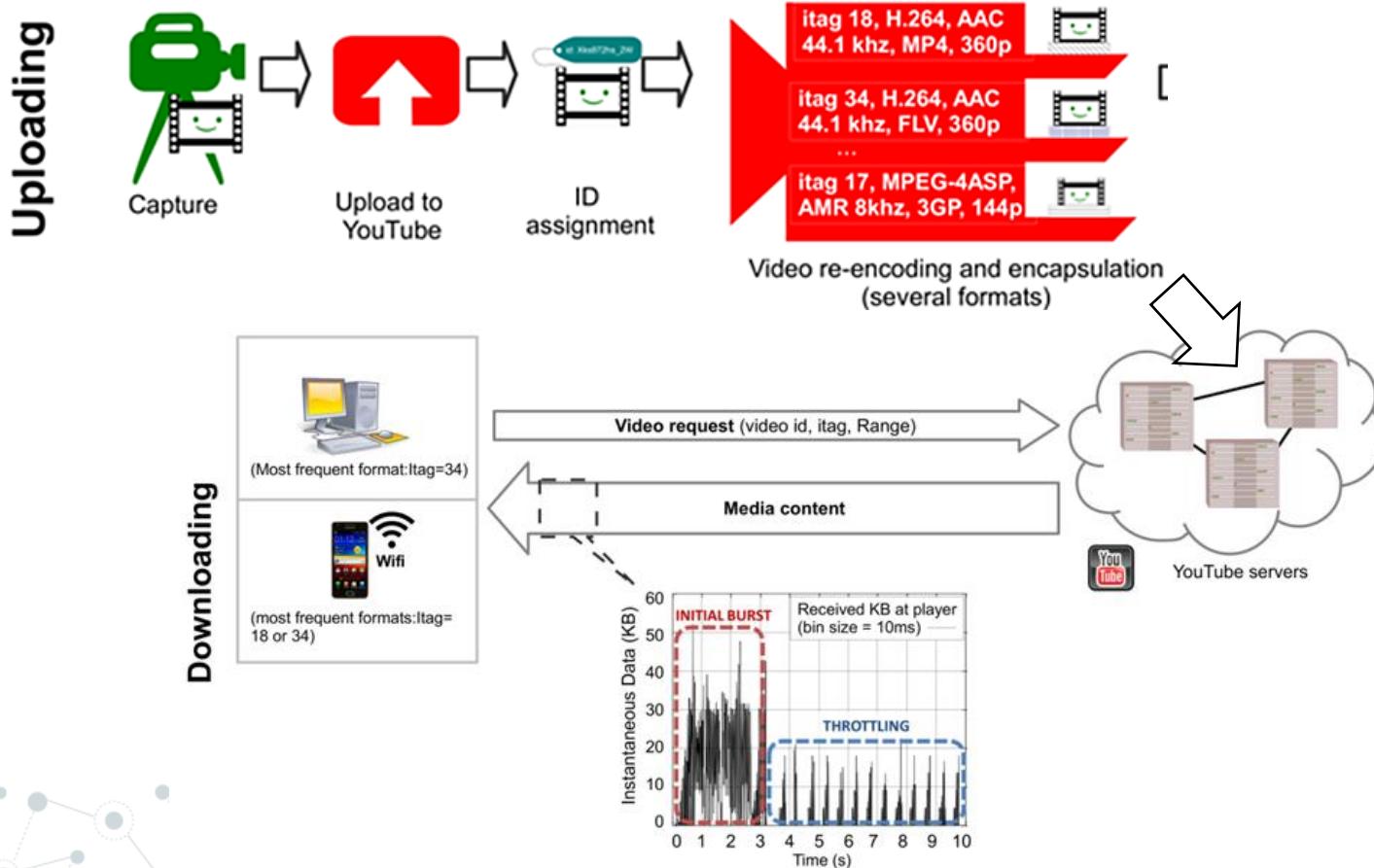


Macroblocking (loss of packets)

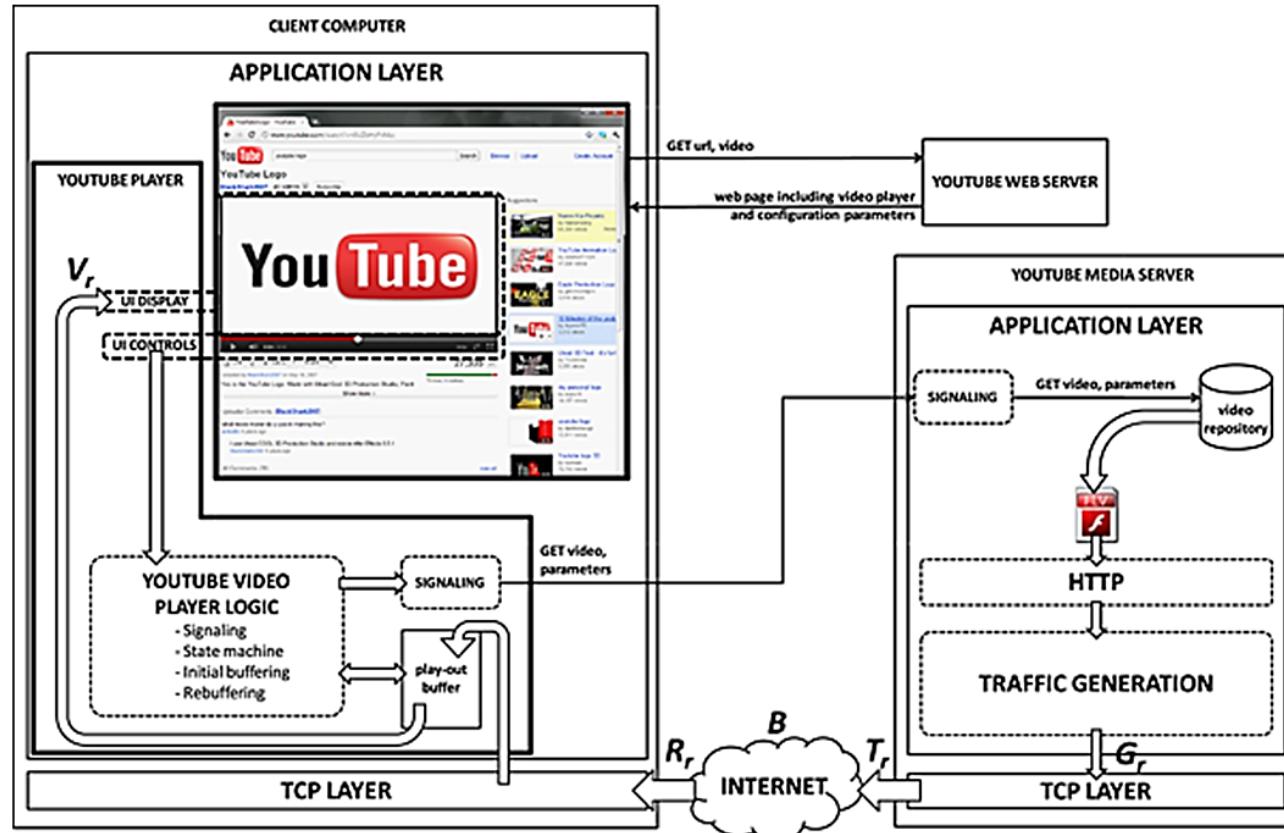


Blockiness

Ejemplo: Youtube



Ejemplo: Youtube



TEMA 2. Servicios y Protocolos en Internet

- 2.1. Introducción a las aplicaciones de red
- 2.2. Servicio de Nombres de Dominio (DNS)
- 2.3. Navegación web
- 2.4. Correo electrónico
- 2.5. Protocolos seguros
- 2.6. Aplicaciones multimedia
- **2.7. Aplicaciones para interconectividad de redes locales**
- 2.8. Cuestiones y ejercicios

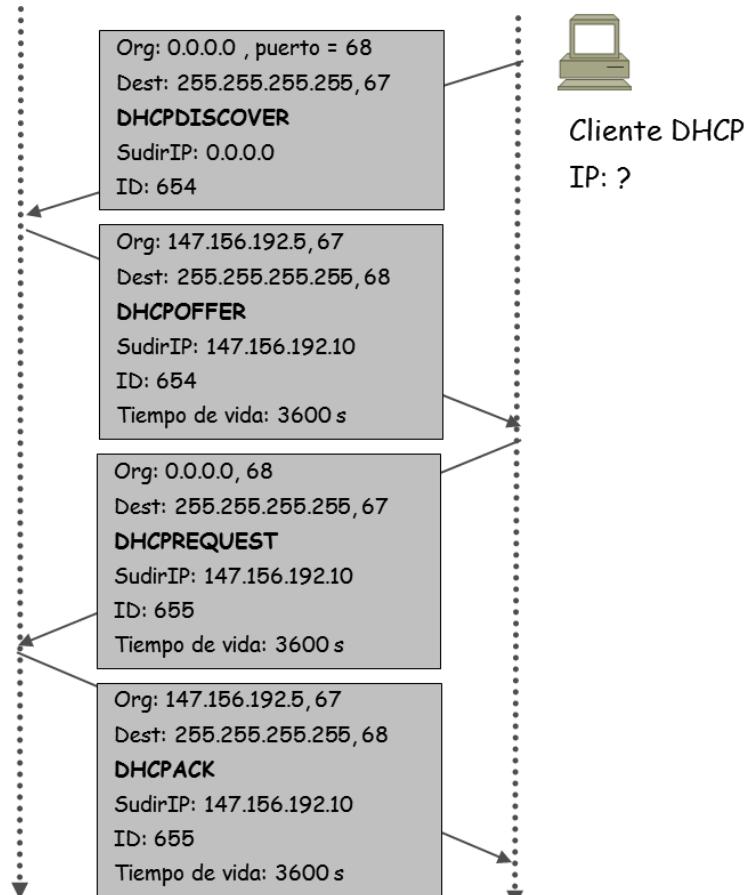
DHCP (Dynamic Host Configuration Protocol)

Asignación de IPs de forma dinámica en una red privada



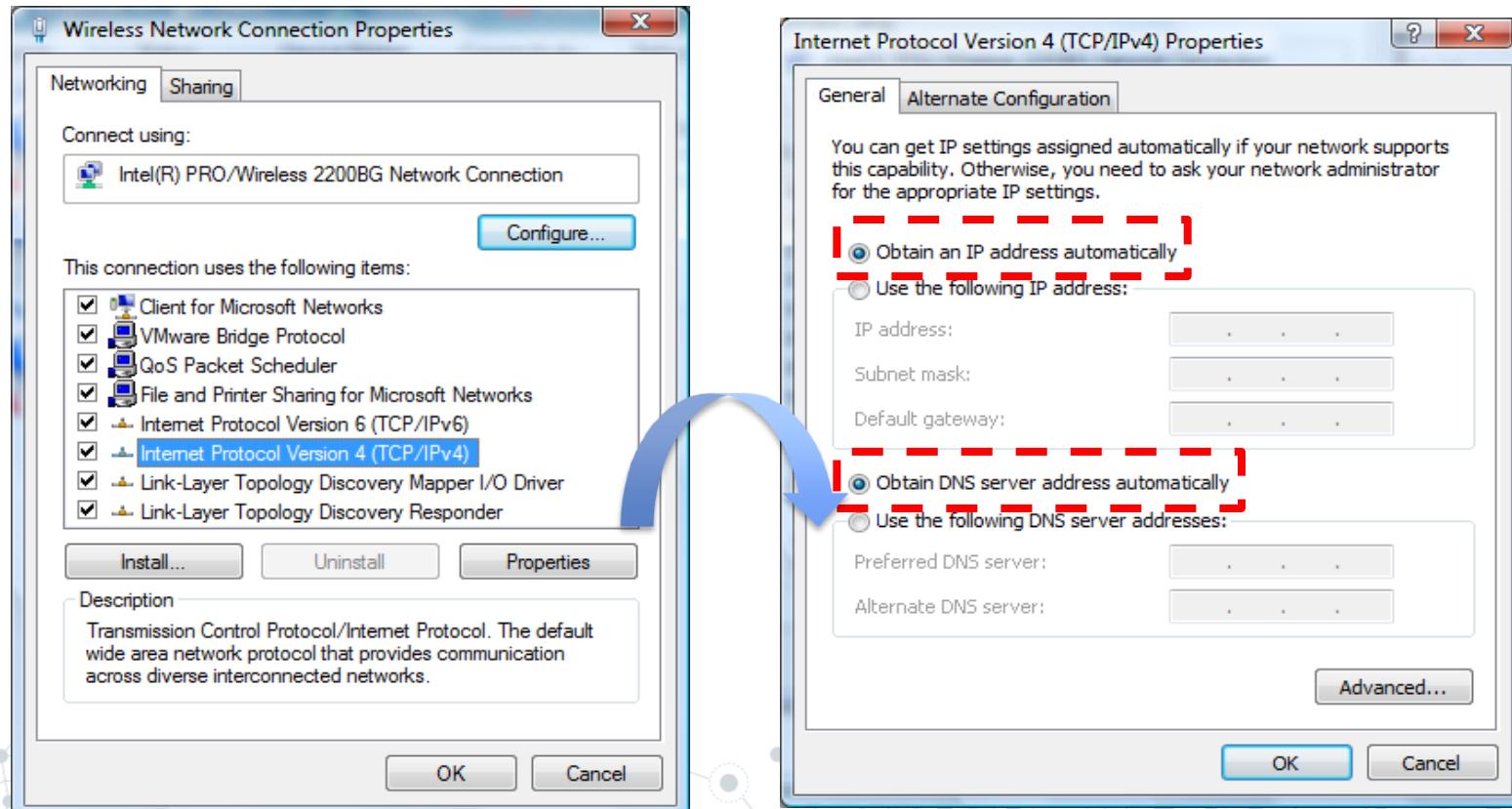
Para asignar las direcciones se usa **DHCP** (RFC 2131-3396), protocolo usuario de UDP (**puerto 67**)

- El host (cliente) envía un mensaje *broadcast*: "DHCP discover"
- El server DHCP responde con un mensaje "DHCP offer"
- El host solicita una dirección IP, mensaje "DHCP request"
- El server DHCP envía la dirección IP: mensaje "DHCP ack"



DHCP (Dynamic Host Configuration Protocol)

Configuración de un cliente DHCP en MS Windows:



DHCP (Dynamic Host Configuration Protocol)

Configuración de un cliente DHCP en Linux (Fedora Core Distribution):

```
# Sample /etc/sysconfig/network-scripts/ifcfg-eth0 :  
  
DEVICE=eth0  
BOOTPROTO=dhcp  
HWADDR=00:0C:29:CE:63:E3  
ONBOOT=yes  
TYPE=Ethernet
```

Configuración de un servidor DHCP en Linux (Fedora Core Distribution):

```
# Sample /etc/dhcpd.conf  
  
default-lease-time 600;max-lease-time 7200;  
option subnet-mask 255.255.255.0;  
option broadcast-address 192.168.1.255;  
option routers 192.168.1.254;  
option domain-name-servers 192.168.1.1, 192.168.1.2;  
option domain-name "mydomain.org";  
subnet 192.168.1.0 netmask 255.255.255.0 {  
    range 192.168.1.10 192.168.1.100;  
    range 192.168.1.150 192.168.1.200;  
}  
  
# Static IP address assignment  
host haagen {  
    hardware ethernet 08:00:2b:4c:59:23;  
    fixed-address 192.168.1.222;
```

TEMA 2. Servicios y Protocolos en Internet

- 2.1. Introducción a las aplicaciones de red.
- **2.2. Servicio de Nombres de Dominio (DNS)**
- **2.3. Navegación web**
- **2.4. Correo electrónico**
- **2.5. Protocolos seguros**
- **2.6. Aplicaciones multimedia**
- **2.7. Aplicaciones para interconectividad de redes locales**
- **2.8. Cuestiones y ejercicios**

Ejercicio

RELACIÓN DE EJERCICIOS DEL TEMA 2. EJERCICIO 3

- Discuta las características de las siguientes aplicaciones en términos de su tolerancia a la pérdida de datos, los requisitos temporales, la necesidad de rendimiento mínimo y la seguridad.

Telefonía móvil

WhatsApp

YouTube

Spotify

Comercio electrónico

Ejercicio

RELACIÓN DE EJERCICIOS DEL TEMA 2. EJERCICIO 4

- ¿Es posible que un host tenga varias direcciones IP y un único nombre de dominio? Discuta un caso
- ¿Es posible que un host tenga varios nombres de dominio y una única dirección IP? Discuta un caso
- ¿Es posible que varios host tengan el mismo nombre de dominio, aunque direcciones IP distintas? Discuta un caso.

Ejercicio

RELACIÓN DE EJERCICIOS DEL TEMA 2. EJERCICIO 9.a

- Una sucursal con 50 empleados en Granada tiene una red interna basada en FastEthernet (100Mbps) que se conecta a Internet con una red de acceso ADSL de 0,5 Mbps de subida y 1,5 Mbps de bajada. Cada empleado, en el desempeño de su trabajo, realiza un promedio de 2000 solicitudes de información a la hora a un servidor de Base de Datos ubicado en la central del banco, en Madrid, donde cada solicitud supone el envío por parte del servidor de un promedio de 10 registros de 1KB cada uno. Adicionalmente, la modificación de datos tras algunas de estas solicitudes supone el envío promedio de 100 actualizaciones, de 10 registros de media, a la hora desde la sucursal al servidor. El resto de los servicios telemáticos se restringe.
- a) Calcule el promedio de la velocidad de transmisión requerida. ¿Es la velocidad del enlace de acceso suficiente?

¿Preguntas?

O comentarios, sugerencias, inquietudes



Fundamentos de Redes

Tema 3

Capa de Transporte en Internet

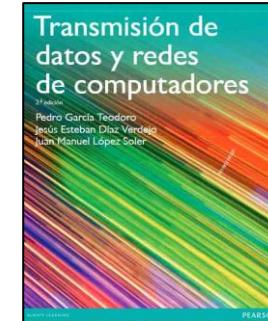
Antonio M. Mora García



Bibliografía

Básica

- P. García-Teodoro, J.E. Díaz-Verdejo, J.M. López-Soler.
Transmisión de datos y redes de computadores, 2^a Edición.
Editorial Pearson, 2014. **CAPÍTULO 10**



Complementaria

- James F. Kurose, Keith W. Ross. Redes de computadoras. Un enfoque descendente. 7^o Edición. Editorial Pearson S.A., 2017.

CAPÍTULO 3



Índice

- **3.1.** Introducción a los protocolos de Capa de Transporte
- **3.2.** Protocolo de datagrama de usuario (UDP)
- **3.3.** Protocolo de control de transmisión (TCP)
 - Multiplexación/demultiplexación
 - Control de conexión
 - Control de errores y de flujo
 - Control de congestión
- **3.4.** Extensiones TCP
- **3.5.** Cuestiones y ejercicios

TEMA 3. Capa de transporte en Internet

- **3.1. Introducción a los protocolos de Capa de Transporte**
- 3.2. Protocolo de datagrama de usuario (UDP)
- 3.3. Protocolo de control de transmisión (TCP)
 - Multiplexación/demultiplexación
 - Control de conexión
 - Control de errores y de flujo
 - Control de congestión
- 3.4. Extensiones TCP
- 3.5. Cuestiones y ejercicios

Capa de transporte

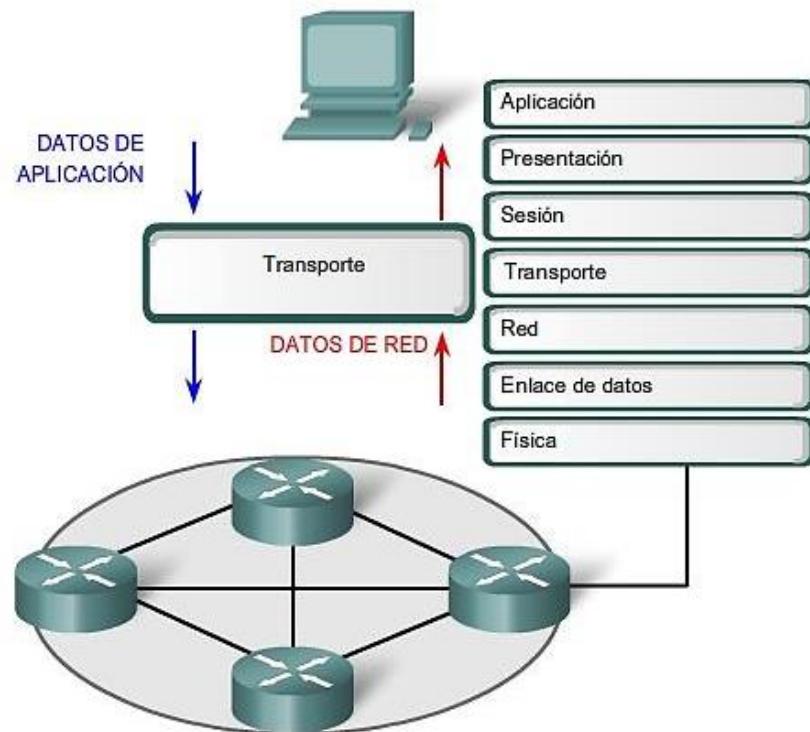
- Las redes de **datos e Internet** nos dan **soporte** para **establecer una comunicación continua y confiable** entre los equipos.
- En un único dispositivo, se pueden utilizar **varios servicios** (correo electrónico, acceso Web, mensajería instantánea, etc).
- Los **datos de cada** una de estas **aplicaciones** 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 **pasarlos a la capa de red** garantizando que lleguen a su destino independientemente la red o redes físicas que utilicen.

Capa de transporte

- **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
 - La introducción de redundancia en la información
- Dos **protocolos:** **TCP y UDP.**

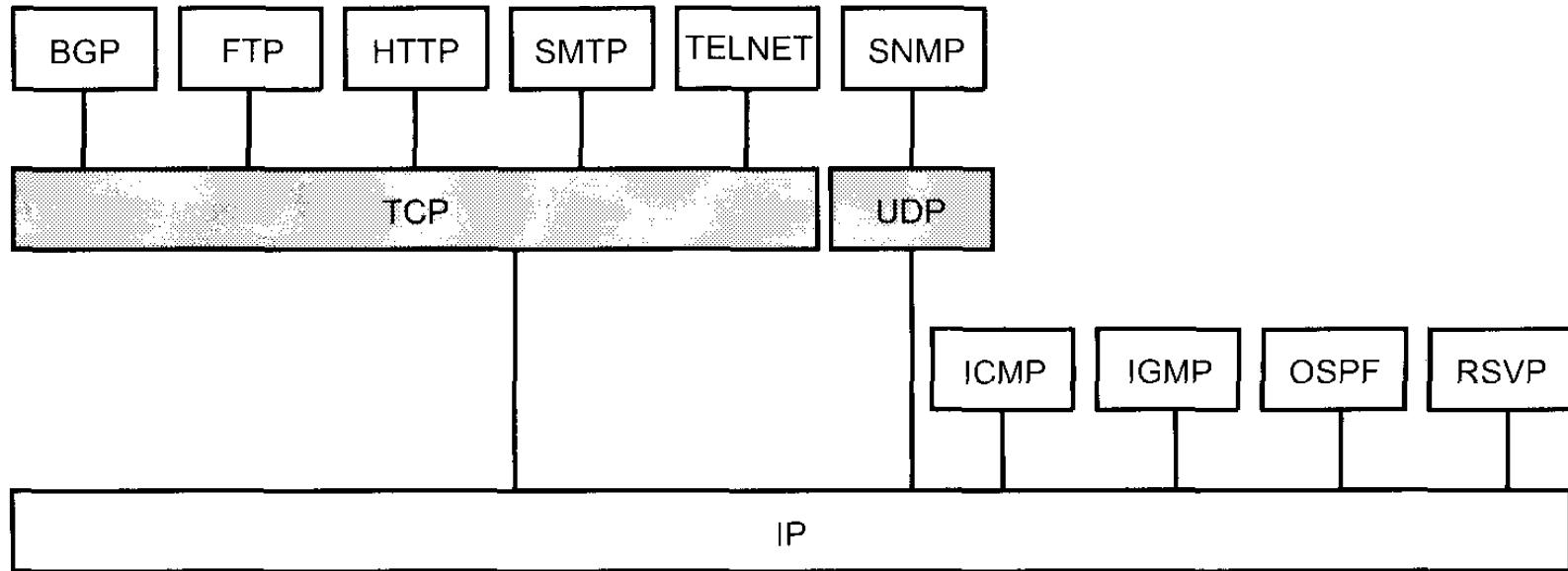
Capa de transporte

- 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.



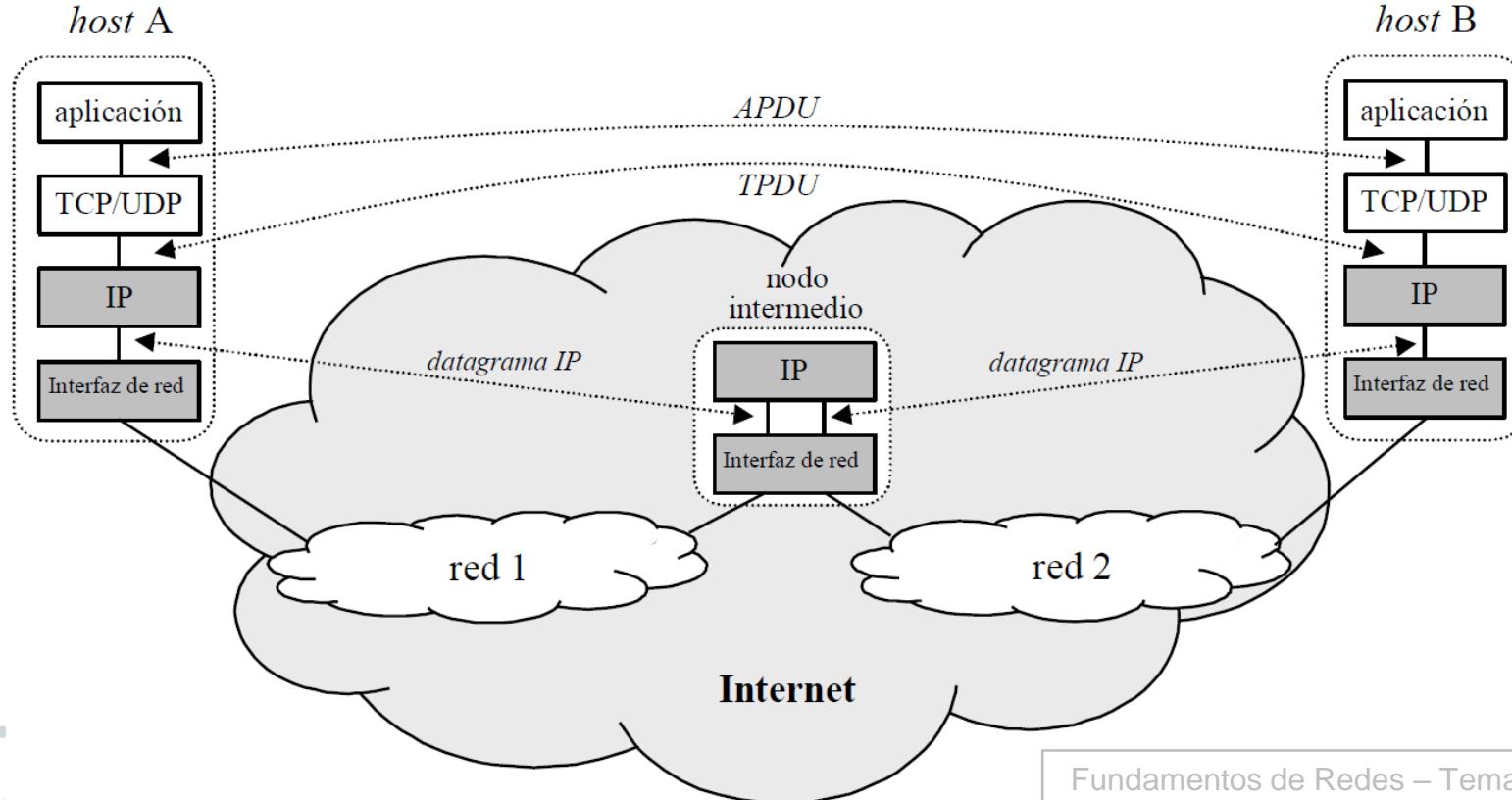
Capa de transporte

- **Protocolos** de las capas de **aplicación** y **transporte**.



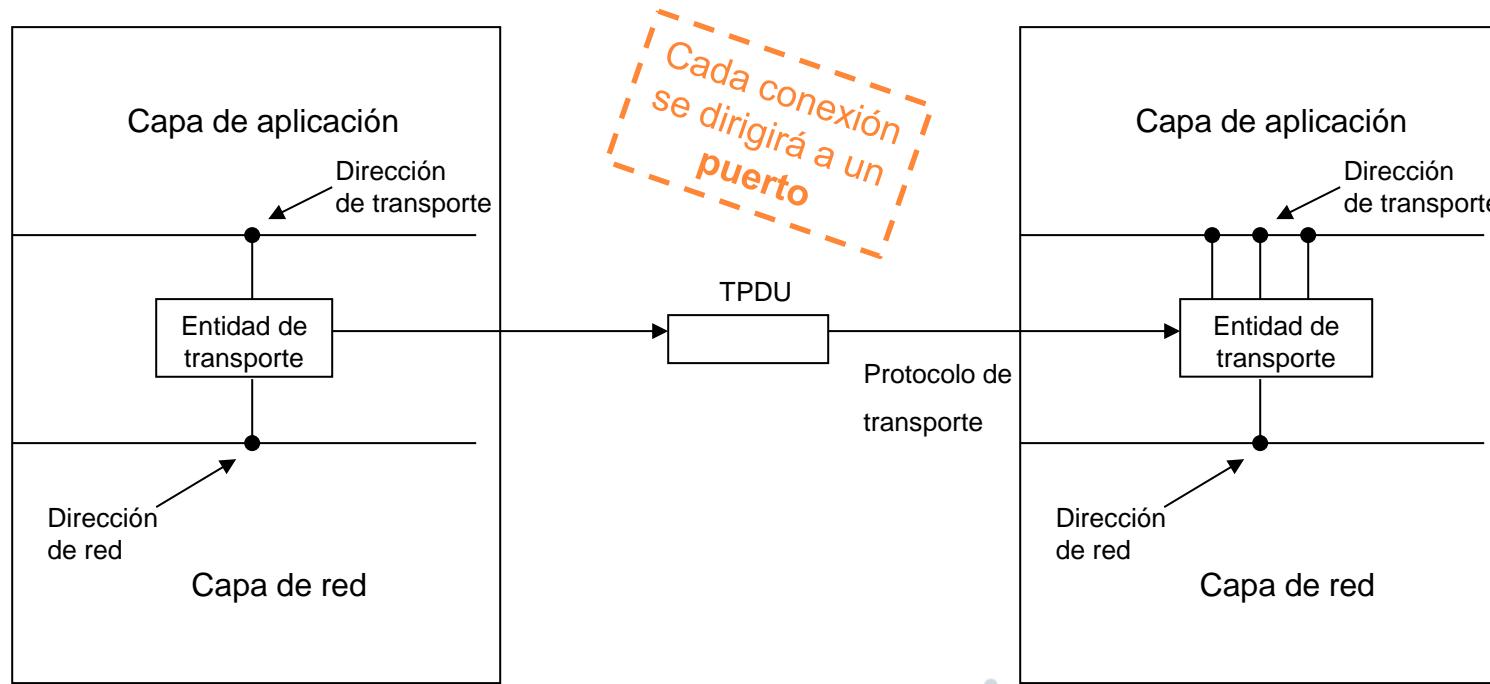
Capa de transporte

- La **comunicación** entre entidades de **transporte** es **extremo a extremo** (*end-to-end*).



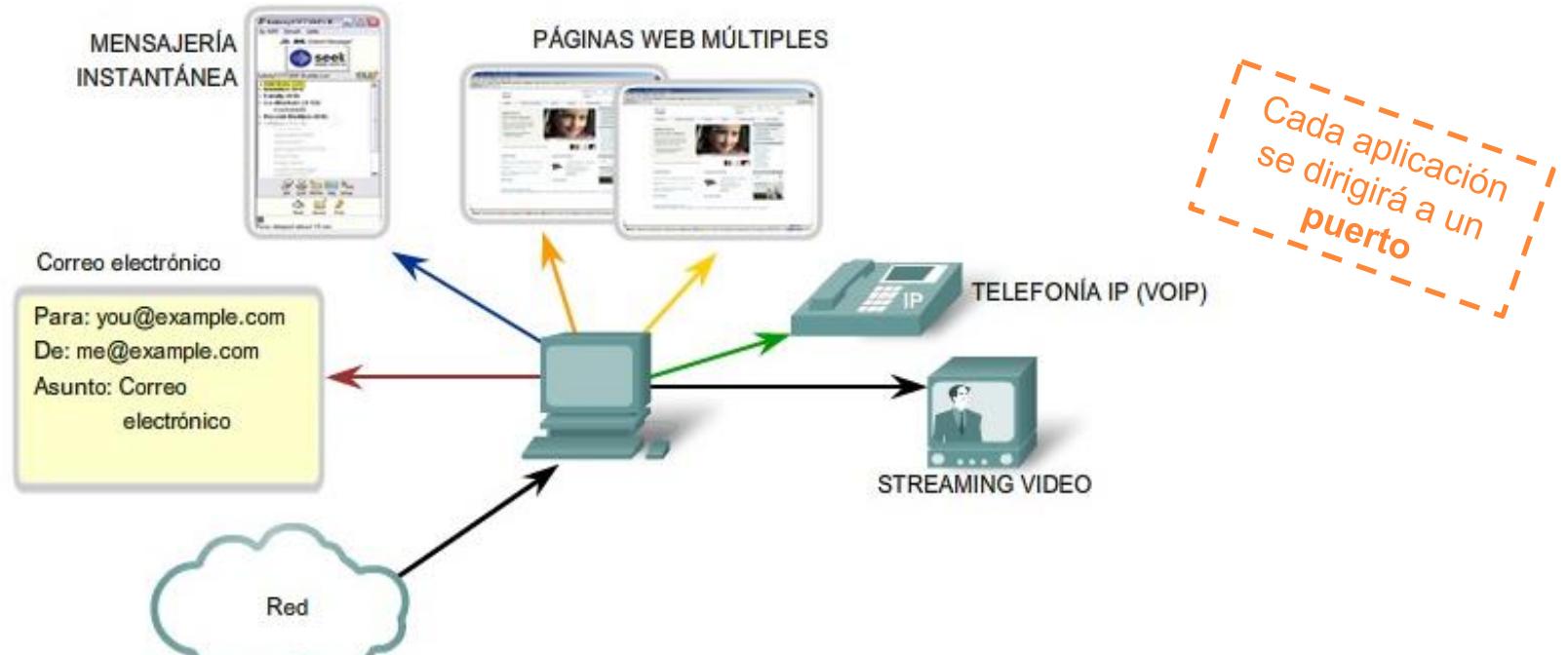
Capa de transporte

- Permite realizar **multiplexación de comunicaciones** (de aplicaciones).



Capa de transporte

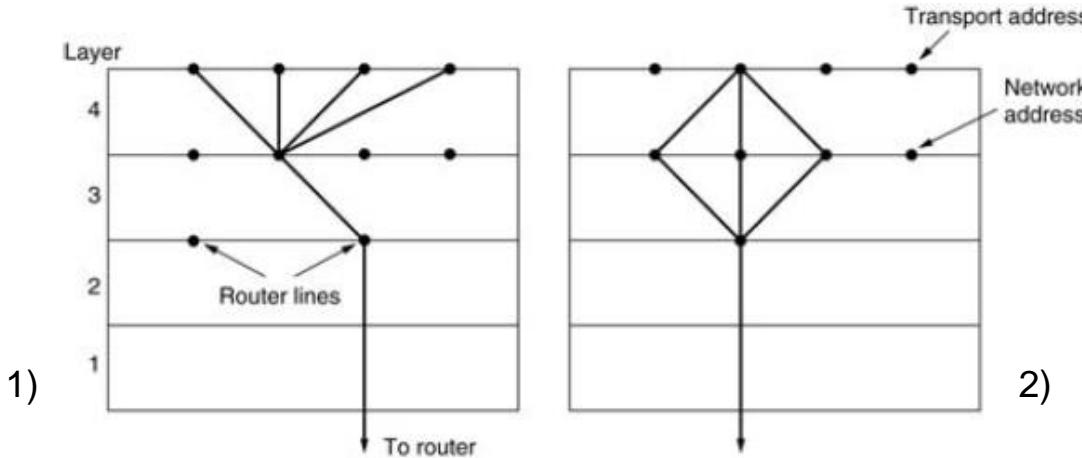
- Permite realizar **multiplexación de comunicaciones** (de aplicaciones).



Capa de transporte

- **Multiplexación/Demultiplexación:**

- 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 quiere aumentar el caudal o reducirse el retardo en una conexión de transporte.



Propósito de la capa de transporte

- 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 y manejo de cada parte.
 - Reensamblaje de segmentos.
 - Identificación de diferentes aplicaciones en origen y destino.
 - Multiplexación y Demultiplexación del tráfico de las aplicaciones.

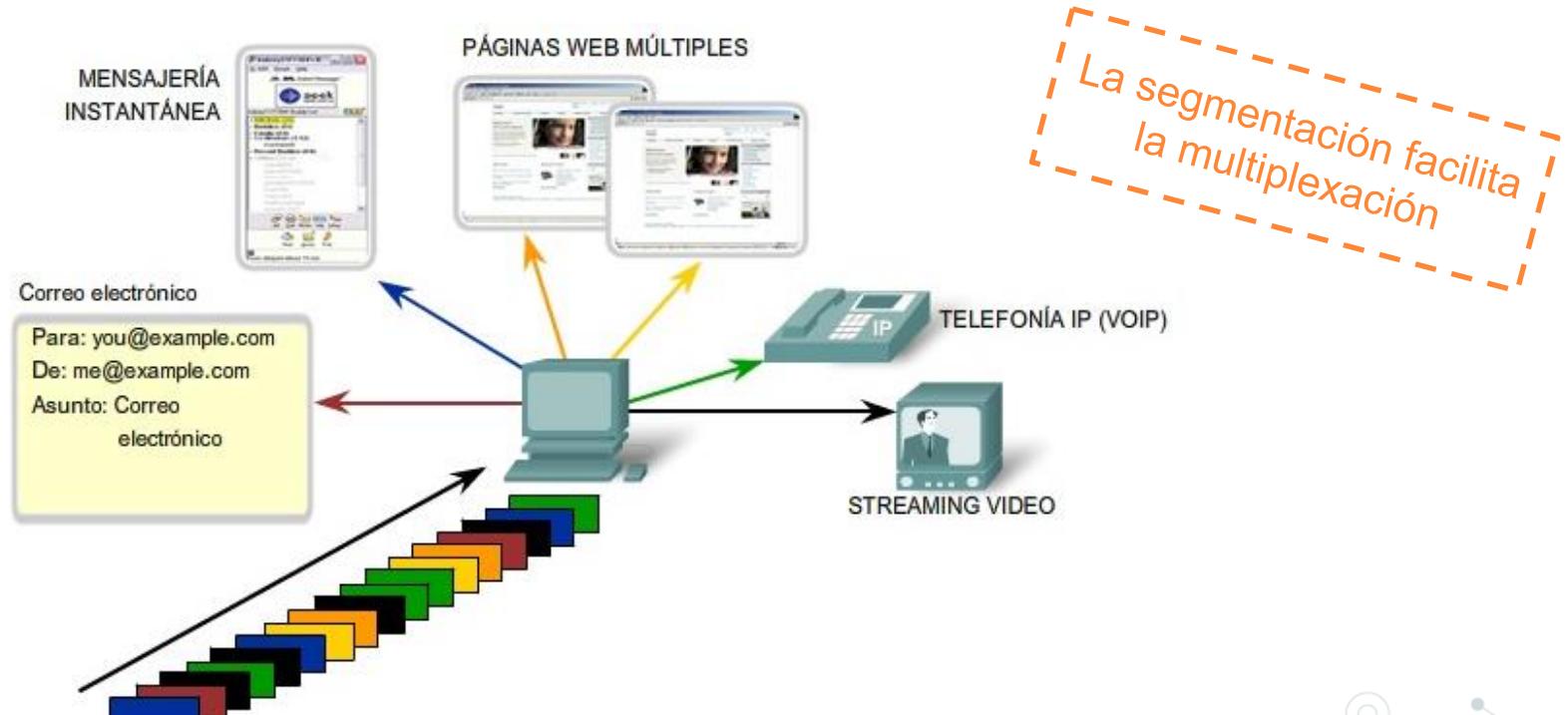
Propósito de la capa de transporte

- **Segmentación de datos 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

Propósito de la capa de transporte

- Segmentación de datos y manejo de cada parte.



Propósito de la capa de transporte

- **Reensamblaje de segmentos:**
 - Al recibir los datos, 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 pasarlos a la capa de aplicación**.
- **Restricciones:**
 - 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.

Propósito de la capa de transporte

- **Identificación de diferentes aplicaciones:**

- 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**.

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.

Puertos

ASIGNACIÓN DE PUERTOS - 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

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

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**

TEMA 3. Capa de transporte en Internet

- 3.1. Introducción a los protocolos de Capa de Transporte
- **3.2. Protocolo de datagrama de usuario (UDP)**
- 3.3. Protocolo de control de transmisión (TCP)
 - Multiplexación/demultiplexación
 - Control de conexión
 - Control de errores y de flujo
 - Control de congestión
- 3.4. Extensiones TCP
- 3.5. Cuestiones y ejercicios

Introducción

- El **protocolo UDP (User Datagram Protocol)** se define en la RFC 768.
- 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).

Funcionalidad
“Best Effort”

Introducción

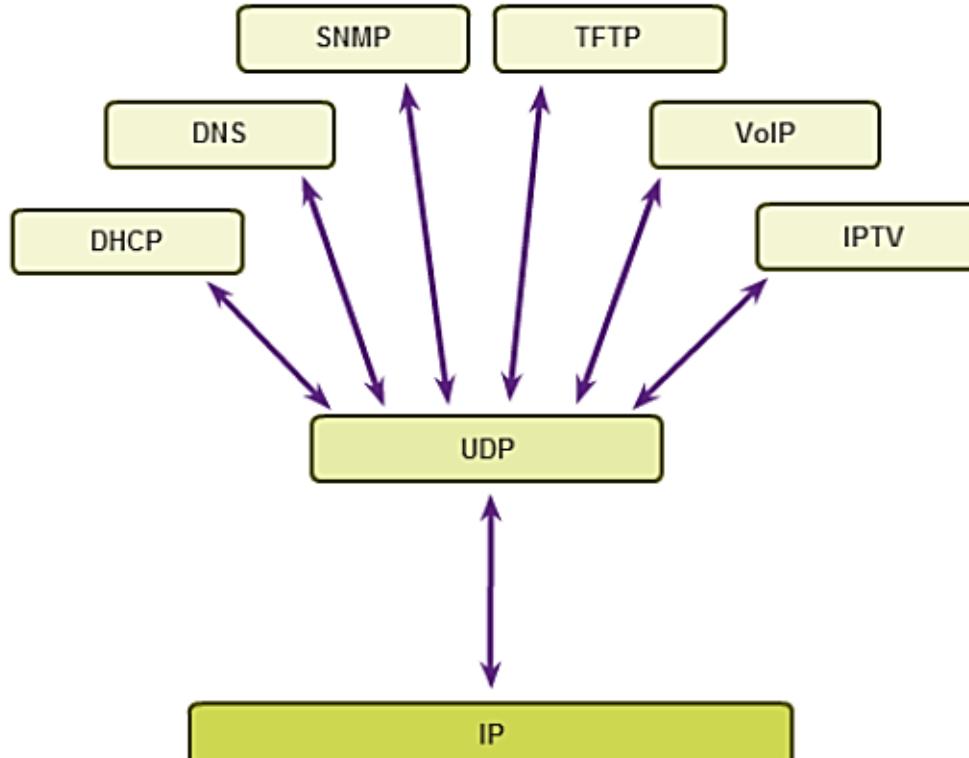
- **UDP utiliza IP** (capa Red), pero agrega la capacidad para distinguir entre varias aplicaciones de destino dentro de un mismo sistema destino.
- Una **aplicación** que utiliza UDP, **asume la responsabilidad** por los **problemas** de **confiabilidad**, incluyendo la pérdida, duplicación y retraso 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.

Utilización

- **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.

Utilización

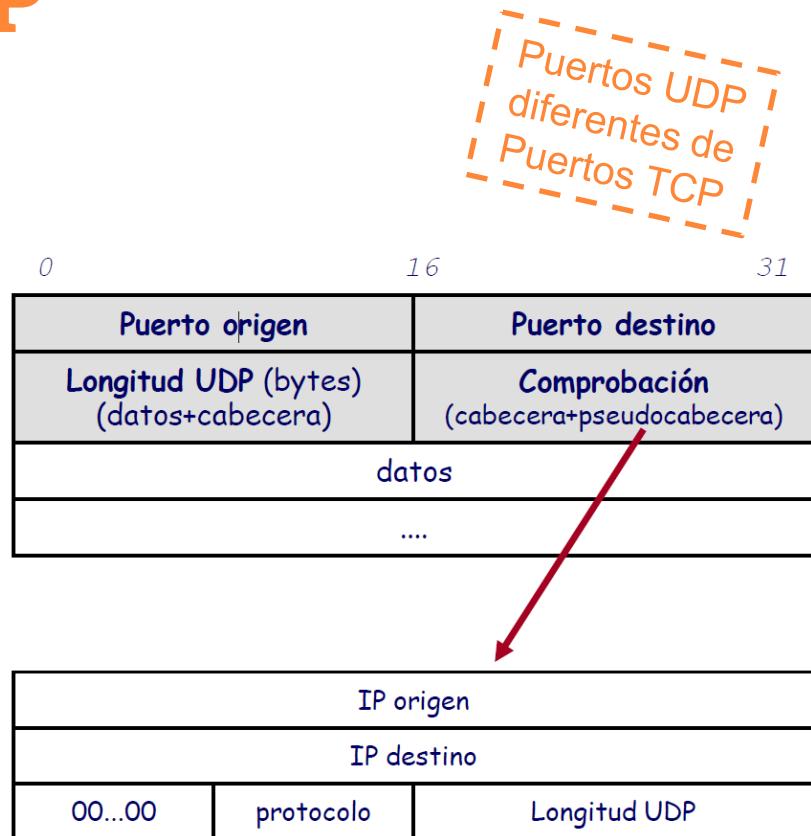
- UDP suele utilizarse para:



Formato Datagrama UDP

- Cabecera:**

- Los números de puerto identifican los procesos emisor y receptor
- Longitud UDP → longitud de la cabecera UDP + longitud de datos
(valor mínimo 8 bytes)
- Datos → PDU de la capa superior

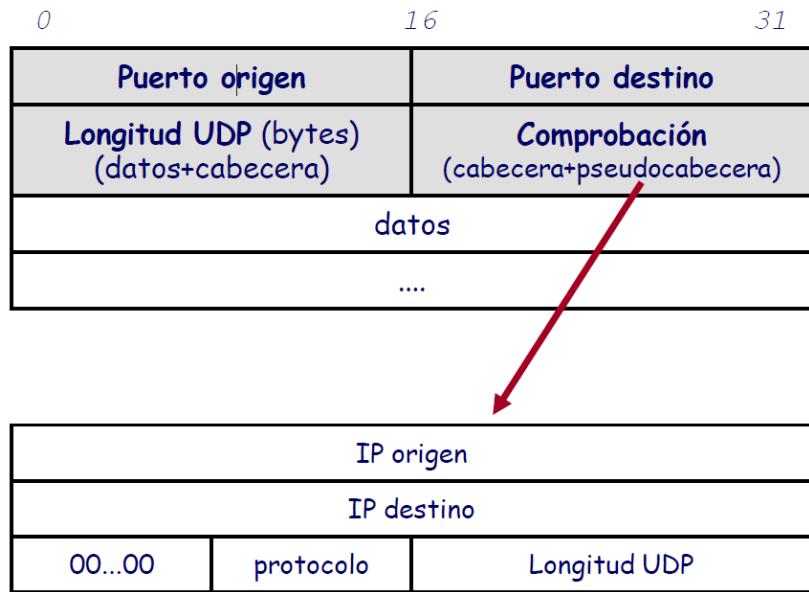


Formato Datagrama UDP

- Cabecera:**

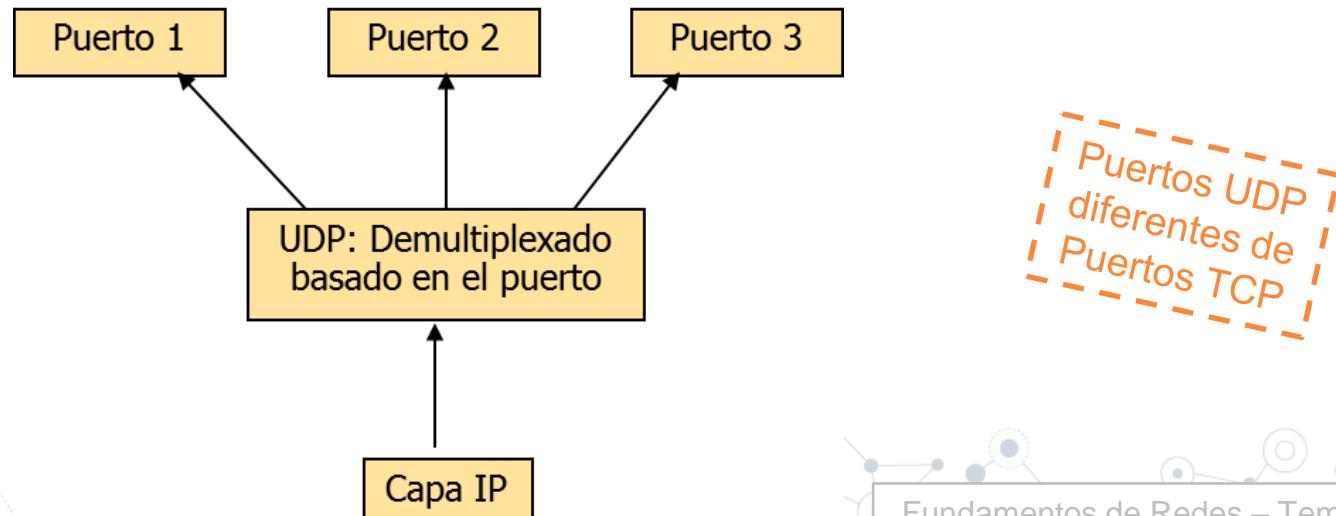
- Comprobación (*Checksum*):

- . 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).



TEMA 3. Capa de transporte en Internet

- 3.1. Introducción a los protocolos de Capa de Transporte
- 3.2. Protocolo de datagrama de usuario (UDP)
- **3.3. Protocolo de control de transmisión (TCP)**
 - Multiplexación/demultiplexación
 - Control de conexión
 - Control de errores y de flujo
 - Control de congestión
- 3.4. Extensiones TCP
- 3.5. Cuestiones y ejercicios

Características del *Transmission Control Protocol*

- RFC 793 (1122, 1323, 2018, 2581).
- 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.

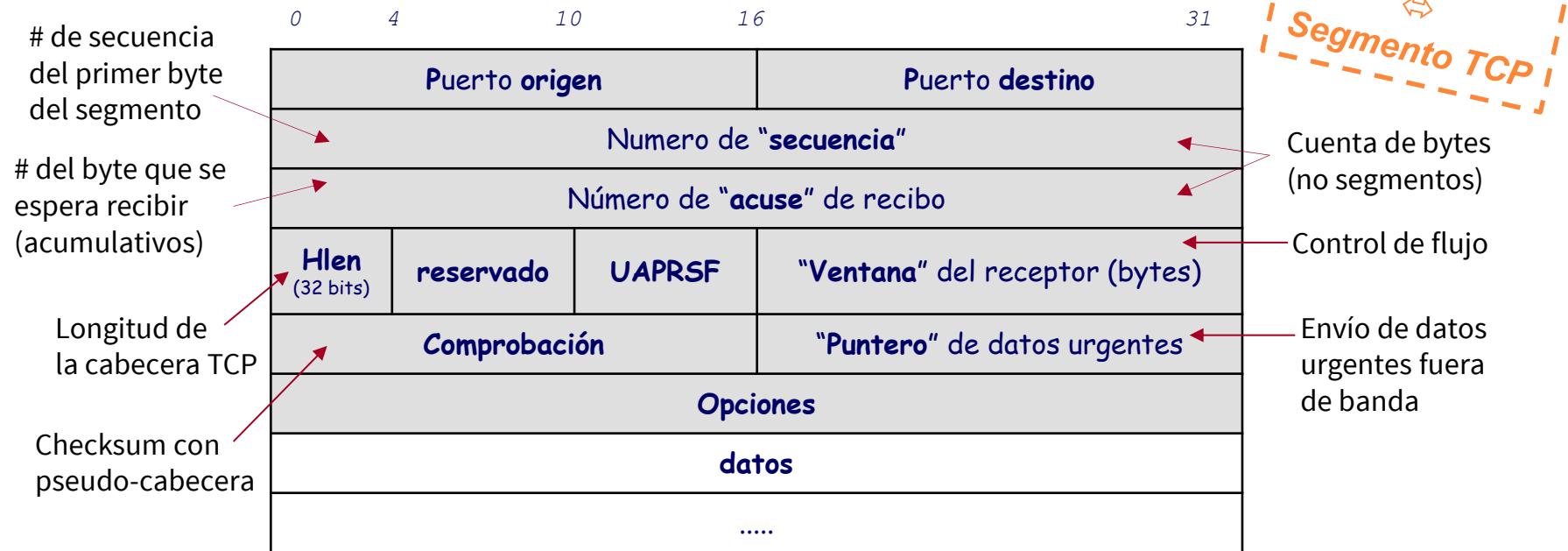
Características del *Transmission Control Protocol*

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.

Características del *Transmission Control Protocol*

CABECERA TCP



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

TEMA 3. Capa de transporte en Internet

- 3.1. Introducción a los protocolos de Capa de Transporte
- 3.2. Protocolo de datagrama de usuario (UDP)
- **3.3. Protocolo de control de transmisión (TCP)**
 - Multiplexación/demultiplexación
 - Control de conexión
 - Control de errores y de flujo
 - Control de congestión
- 3.4. Extensiones TCP
- 3.5. Cuestiones y ejercicios

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**:

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

Puertos TCP diferentes de Puertos UDP

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

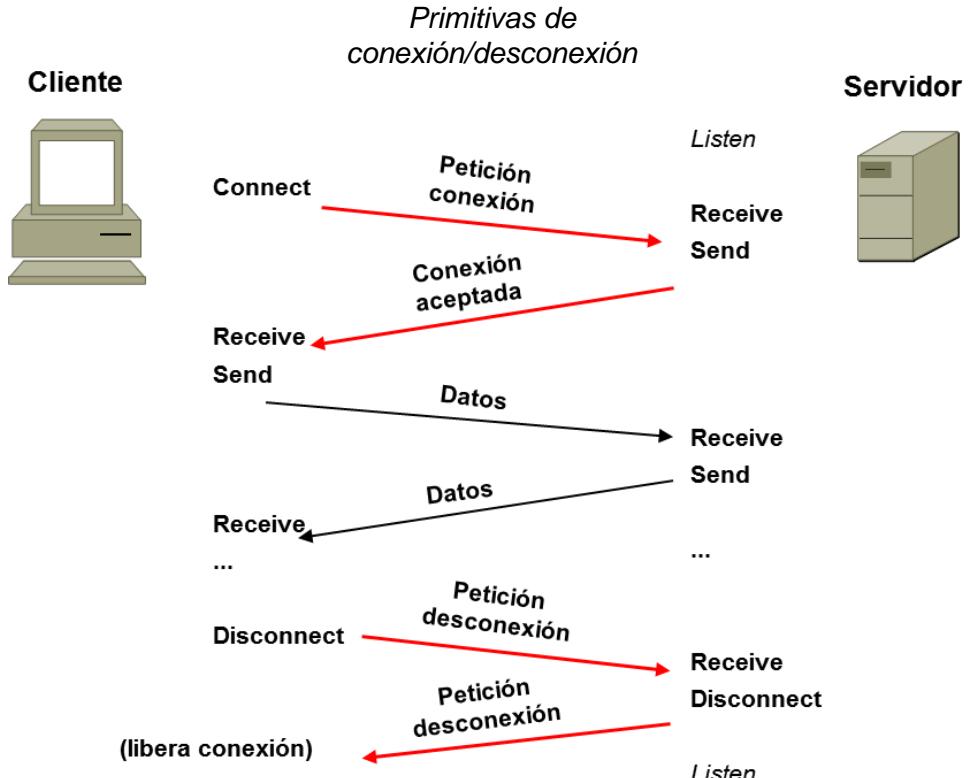
TEMA 3. Capa de transporte en Internet

- 3.1. Introducción a los protocolos de Capa de Transporte
- 3.2. Protocolo de datagrama de usuario (UDP)
- **3.3. Protocolo de control de transmisión (TCP)**
 - Multiplexación/demultiplexación
 - **Control de conexión**
 - Control de errores y de flujo
 - Control de congestión
- 3.4. Extensiones TCP
- 3.5. Cuestiones y ejercicios

Control de la conexión

- TCP es **orientado a 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).

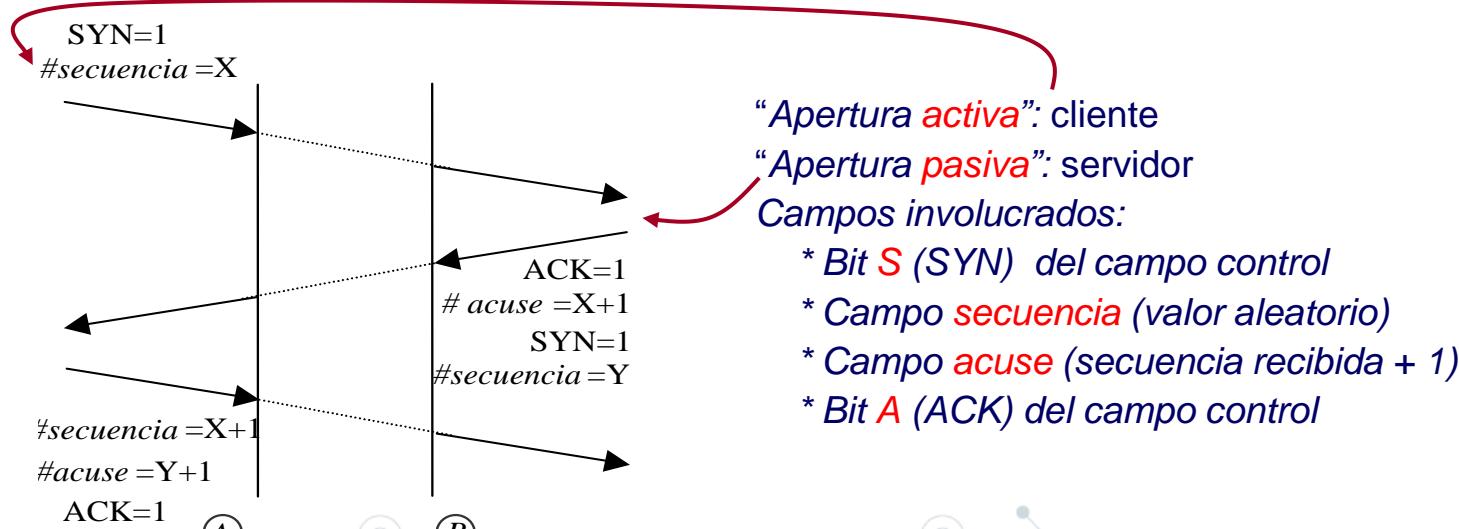
Es un mecanismo de sincronización entre emisor y receptor. **Para garantizar comunicación ordenada y sin errores**



Control de la conexión

ESTABLECIMIENTO

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



Control de la conexión

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 $4 \mu\text{s}$ aproximadamente. En este caso el contador se da la vuelta (y el **ISN** reaparece) al cabo de 4 horas 46 min.

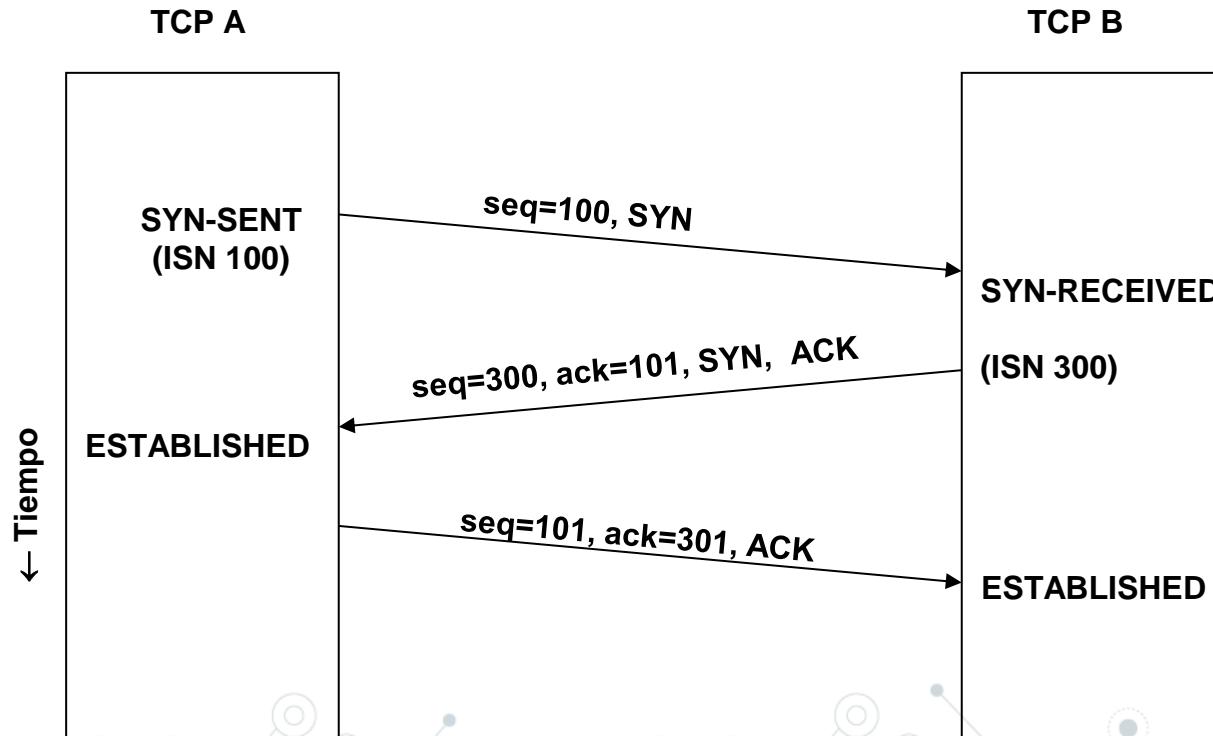
Control de la conexión

NÚMEROS DE SECUENCIA

- 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).

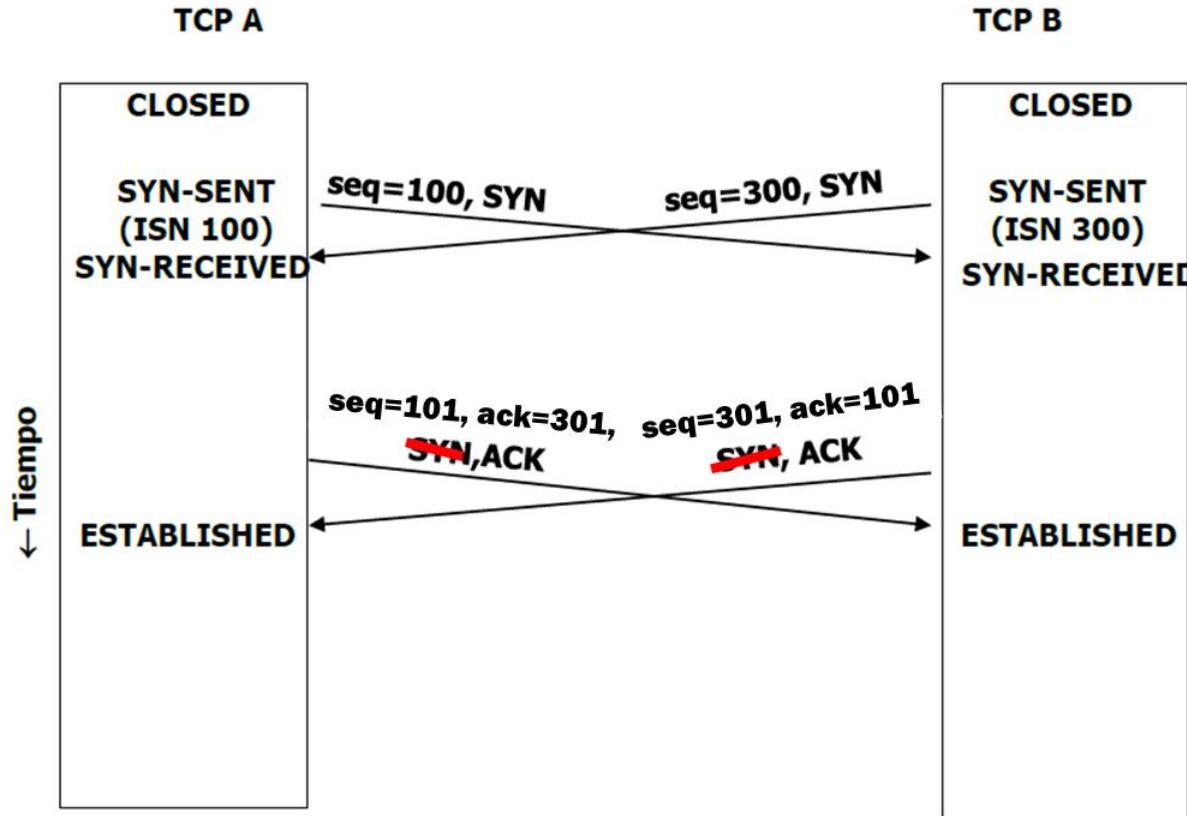
Control de la conexión

- Ejemplo: *three-way handshaking*



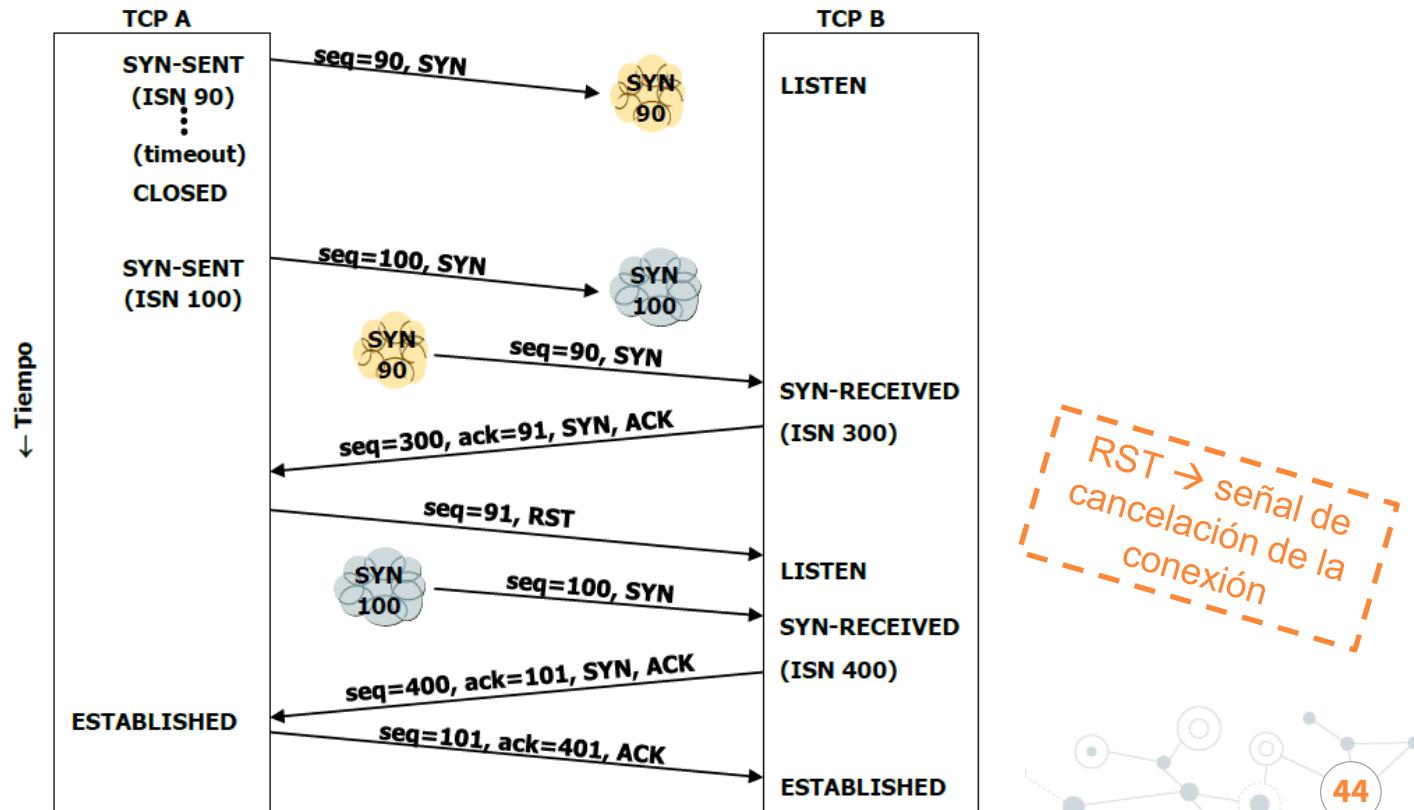
Control de la conexión

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



Control de la conexión

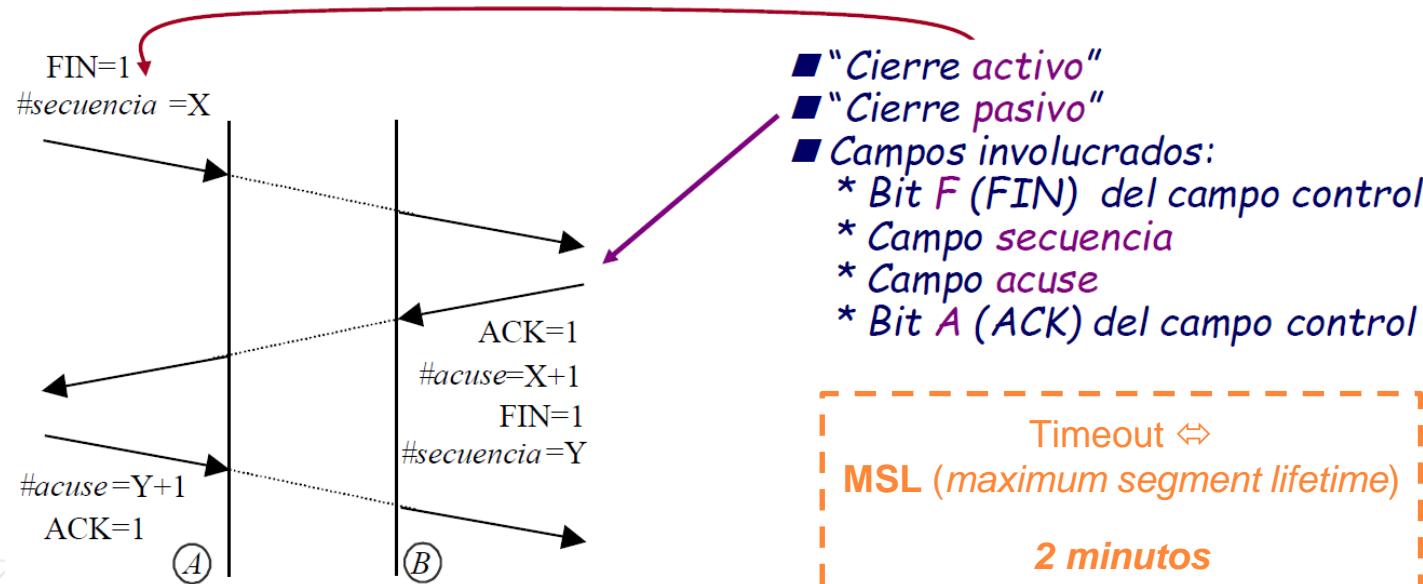
- Ejemplo: ***three-way handshaking (SYN retrasados y duplicados)***



Control de la conexión

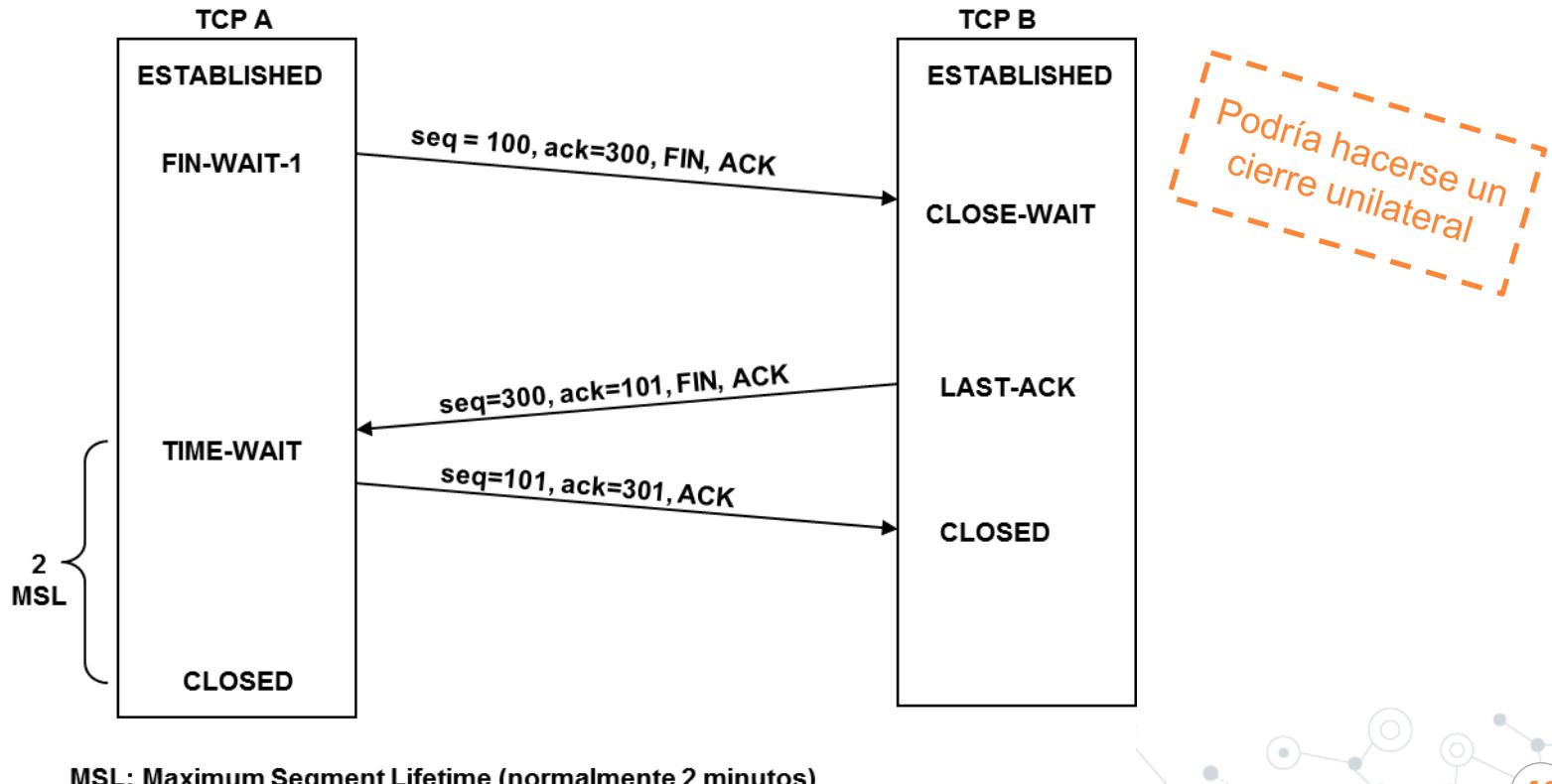
CIERRE

- 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*.



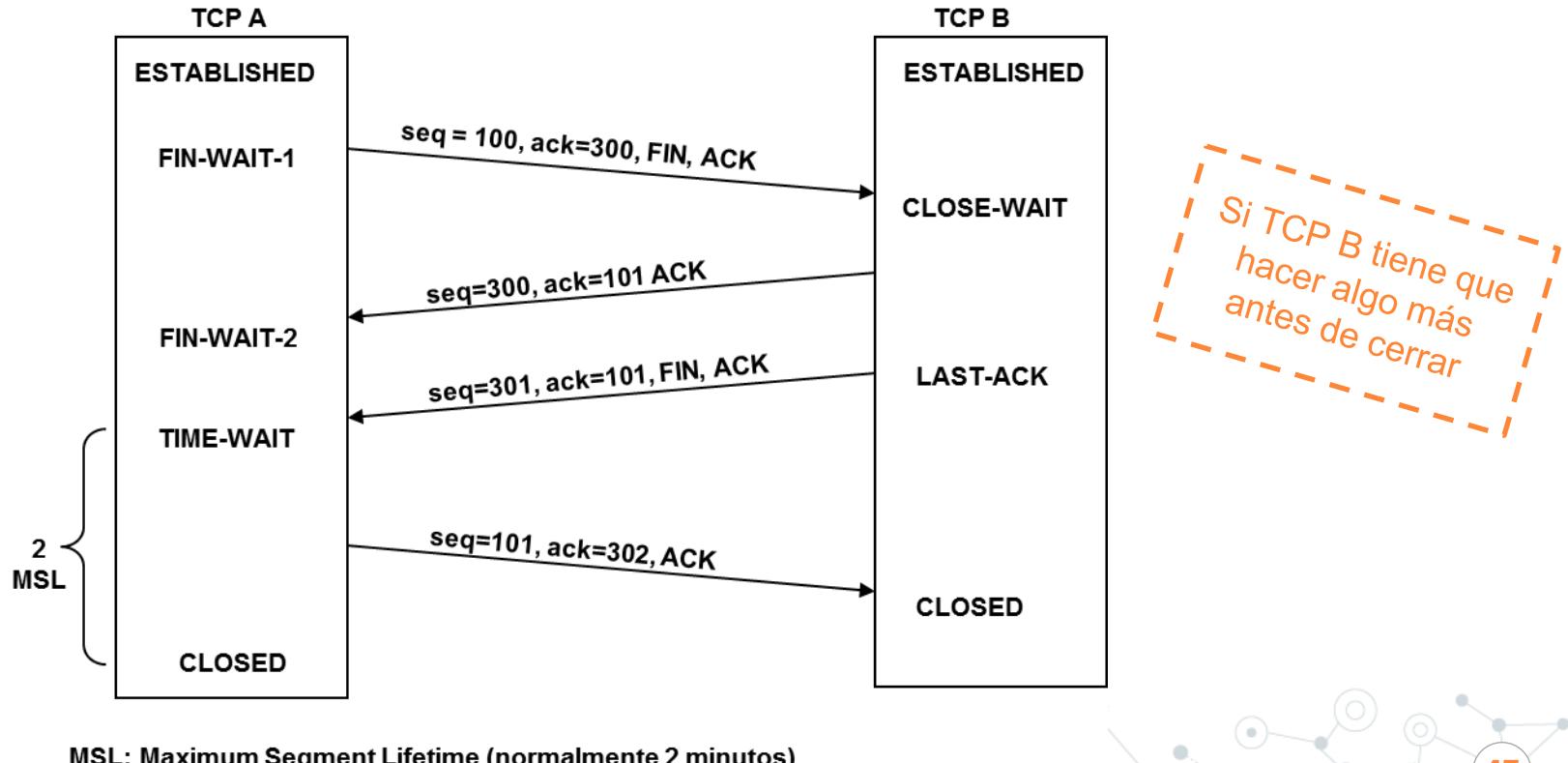
Control de la conexión

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



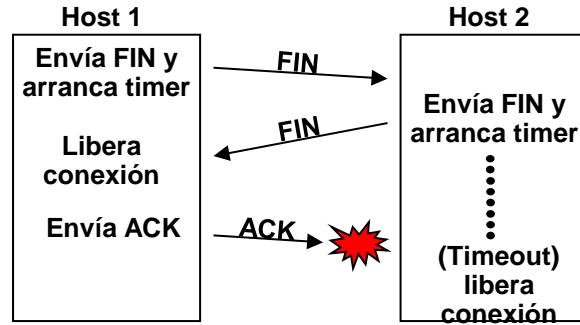
Control de la conexión

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

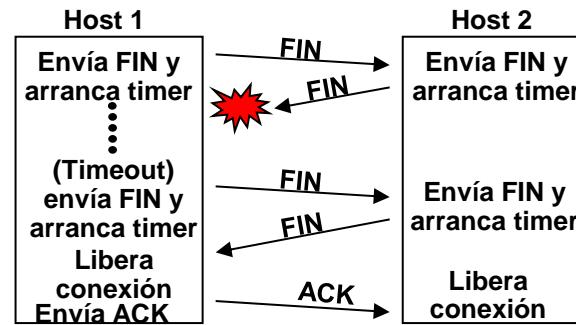


Control de la conexión

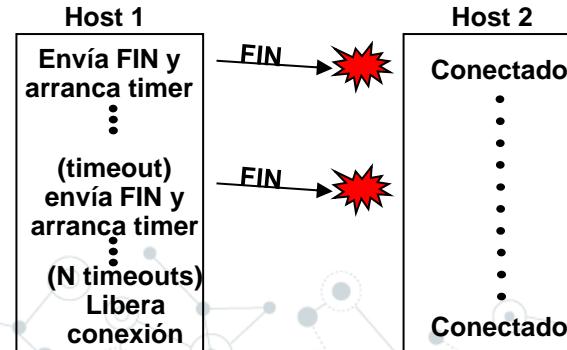
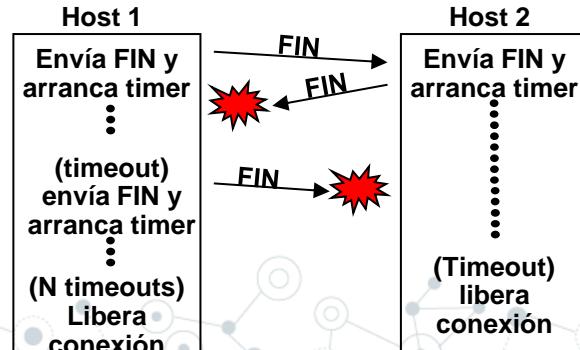
- Ejemplos de cierres anormales:



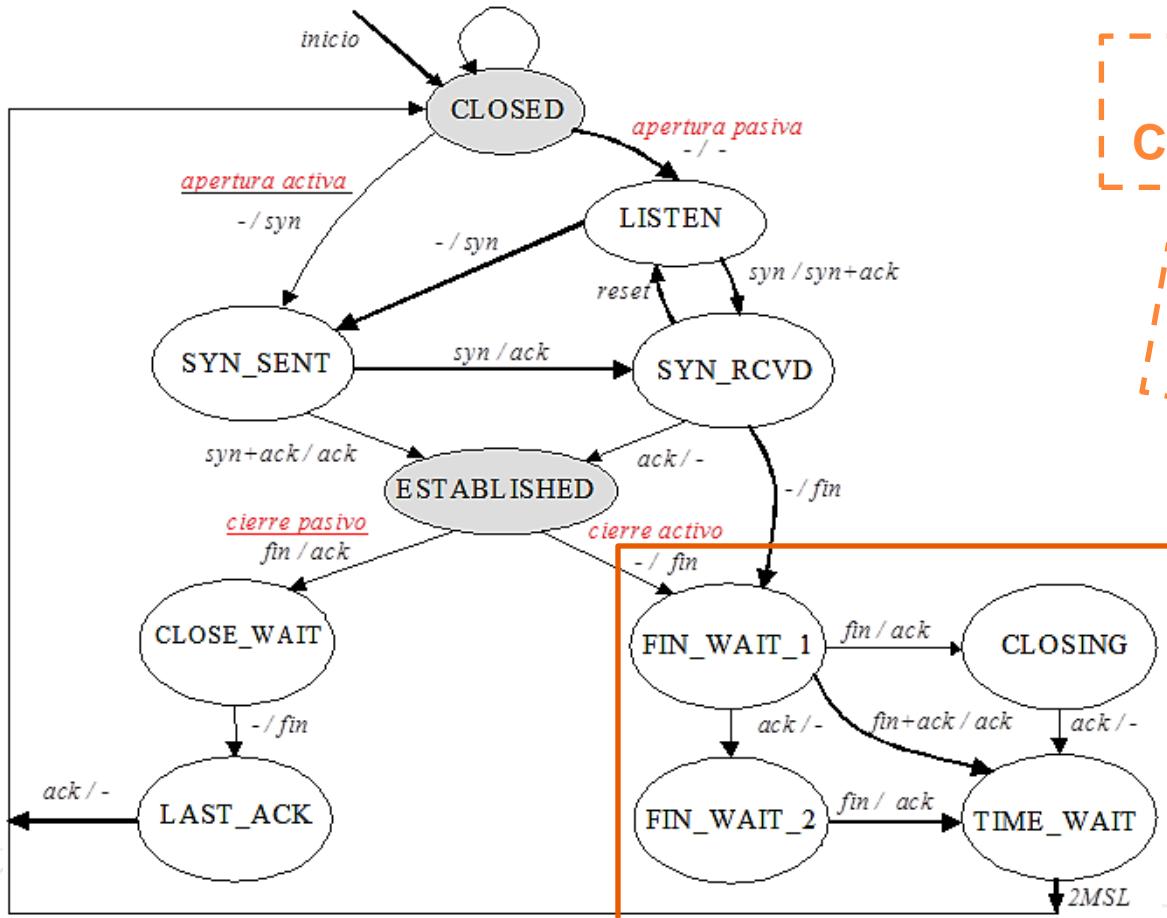
Pérdida de ACK final



Pérdida de respuesta FIN



Autómata de estados finitos TCP



Establecimiento y
Cierre de la conexión

LEYENDA: a/b
Segmento a recibido
Segmento b transmitido

Distintas posibilidades
de cierres de conexión

Intercambio de datos

TCP ⇄ APPLICACIÓN

- 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.

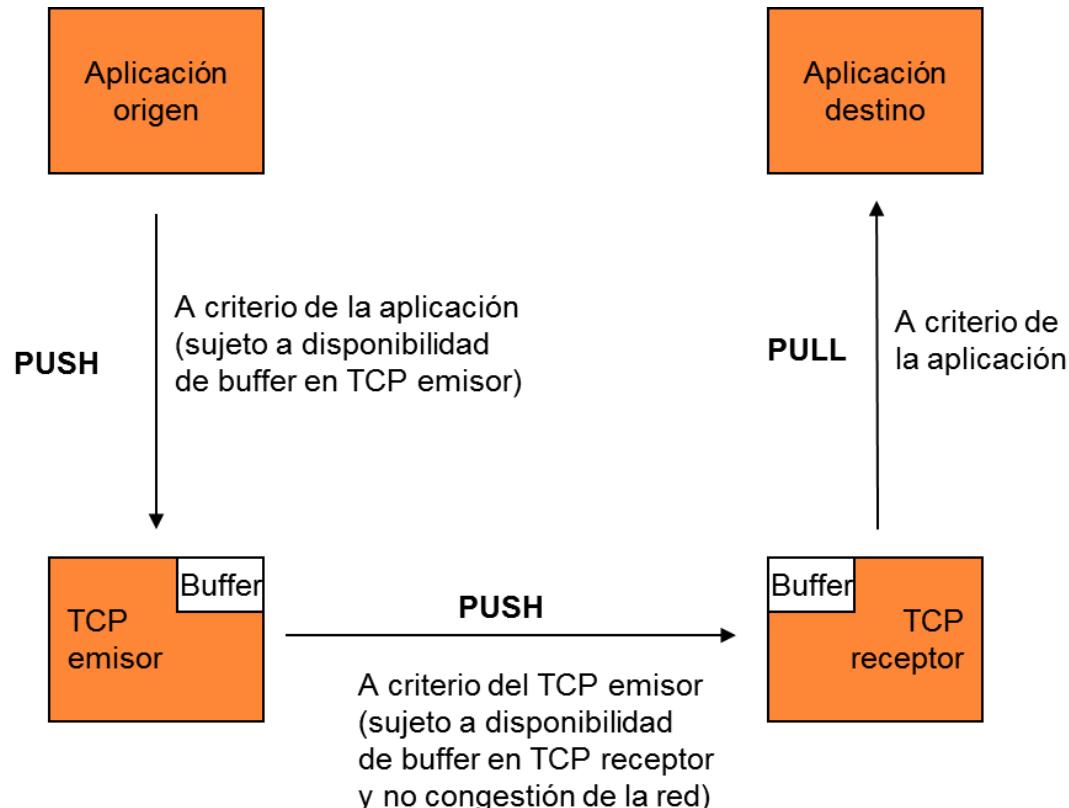
Intercambio de datos

TCP ⇄ 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.

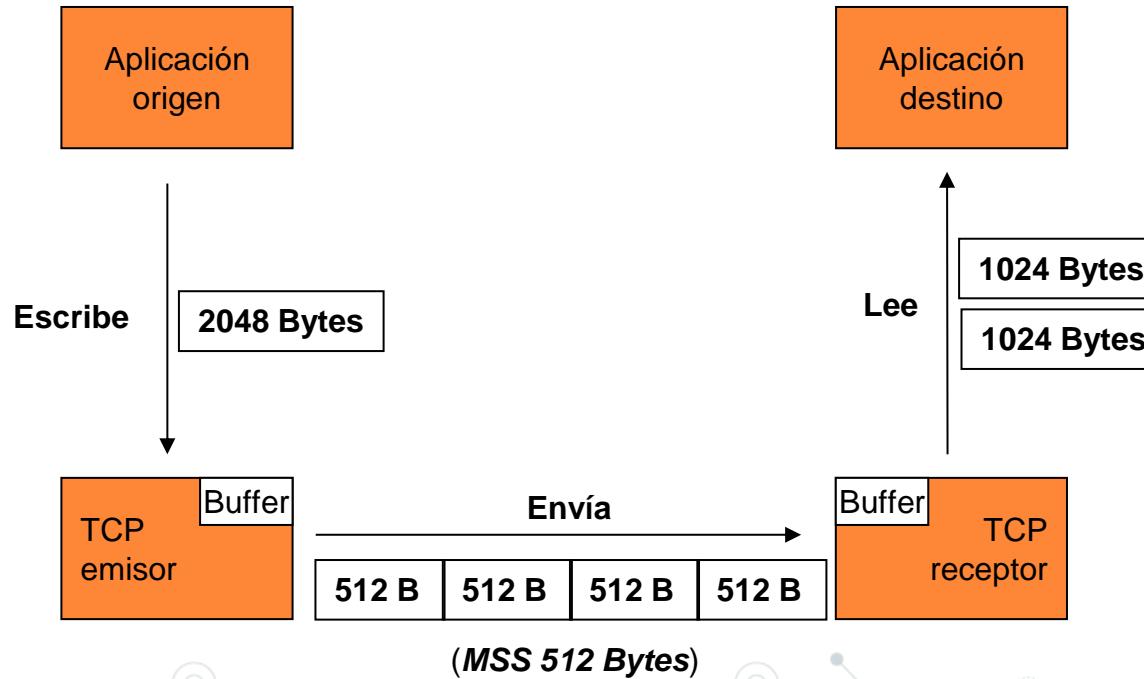
Intercambio de datos

TCP ⇄ APPLICACIÓN y TCP ⇄ TCP



Intercambio de datos

TCP ⇄ APPLICACIÓN y TCP ⇄ TCP



Intercambio de datos

CASOS EXCEPCIONALES

- **Datos “Pushed” (bit PSH):**

La aplicación pide al TCP emisor que envíe esos datos lo antes posible. 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

TEMA 3. Capa de transporte en Internet

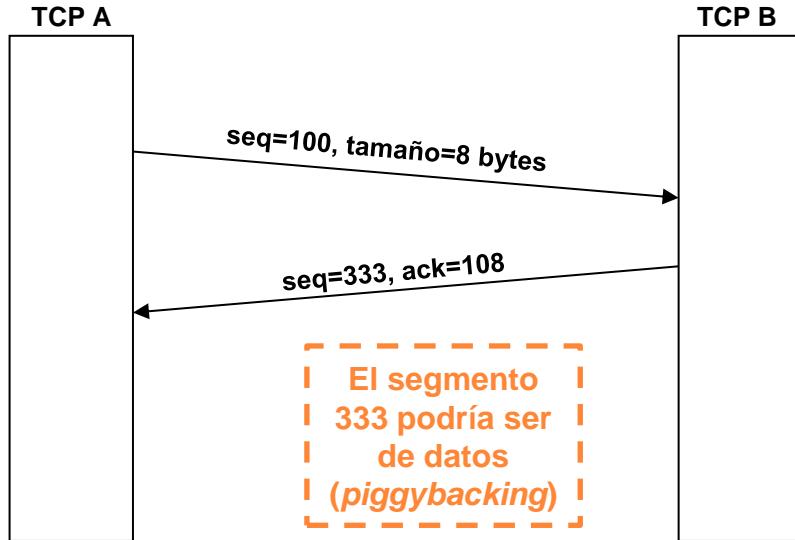
- 3.1. Introducción a los protocolos de Capa de Transporte
- 3.2. Protocolo de datagrama de usuario (UDP)
- **3.3. Protocolo de control de transmisión (TCP)**
 - Multiplexación/demultiplexación
 - Control de conexión
 - **Control de errores y de flujo**
 - Control de congestión
- 3.4. Extensiones TCP
- 3.5. Cuestiones y ejercicios

Control de errores

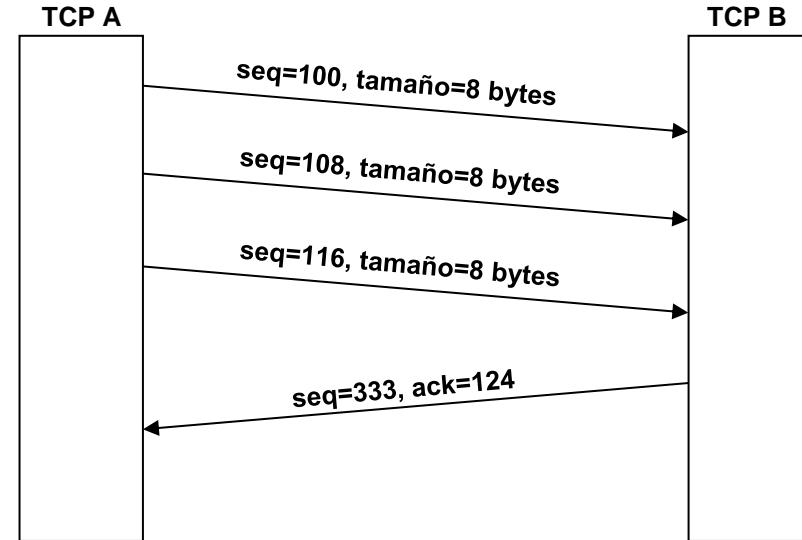
- Para el control de errores se sigue un **esquema ARQ** (*Automatic Repeat-reQuest*) con **confirmaciones positivas y acumulativas**.
- Campos** involucrados:
 - Campo **secuencia**: offset (en bytes) dentro del mensaje.
 - Campo **acuse**: número de byte esperado en el receptor.
 - Bit **A** (ACK) 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)

Control de errores

- ARQ: Funcionamiento habitual



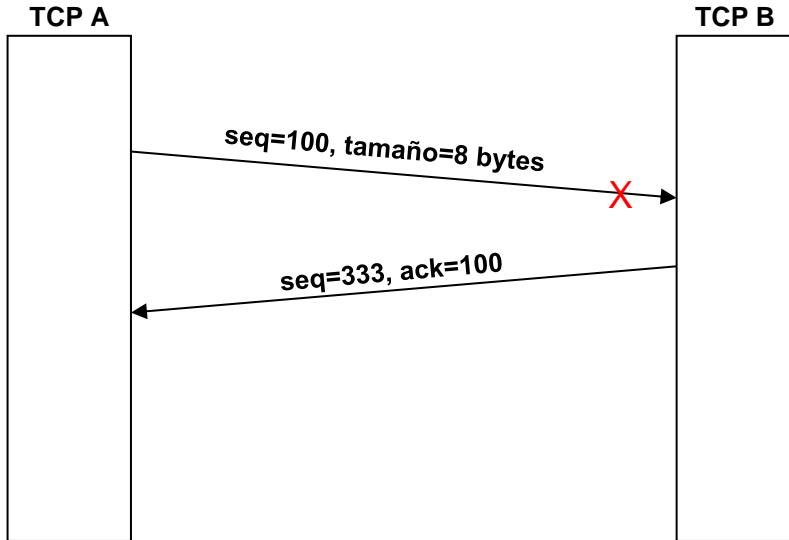
Confirmación simple



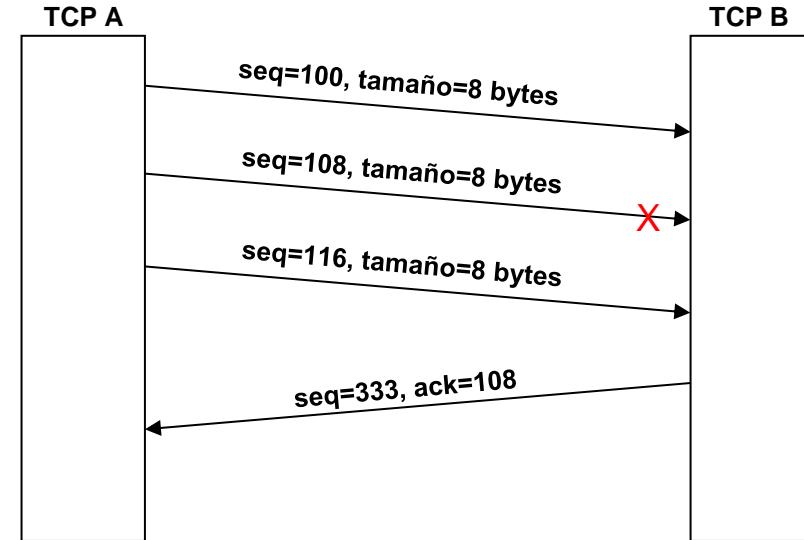
Confirmación acumulativa

Control de errores

- ARQ: Error en segmento



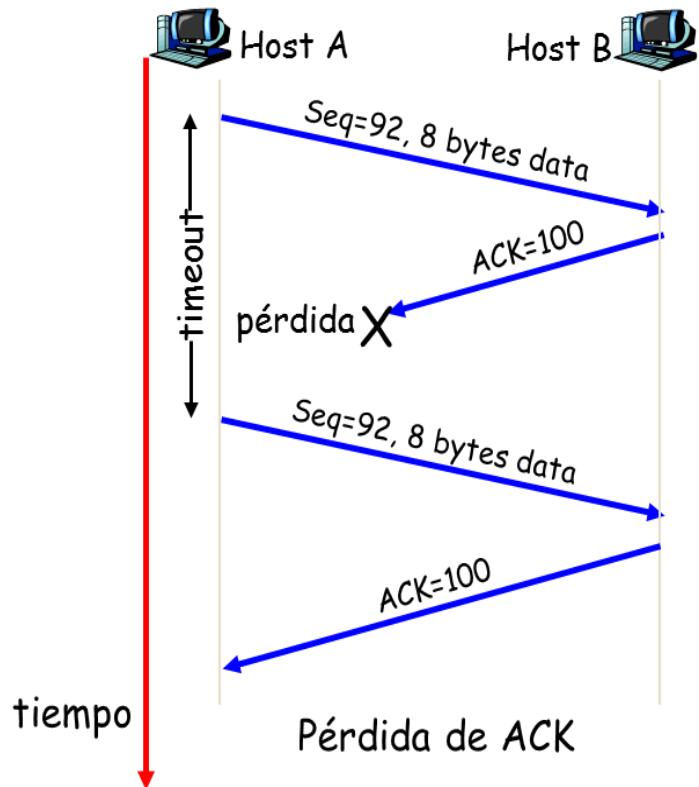
Solicitud de reenvío



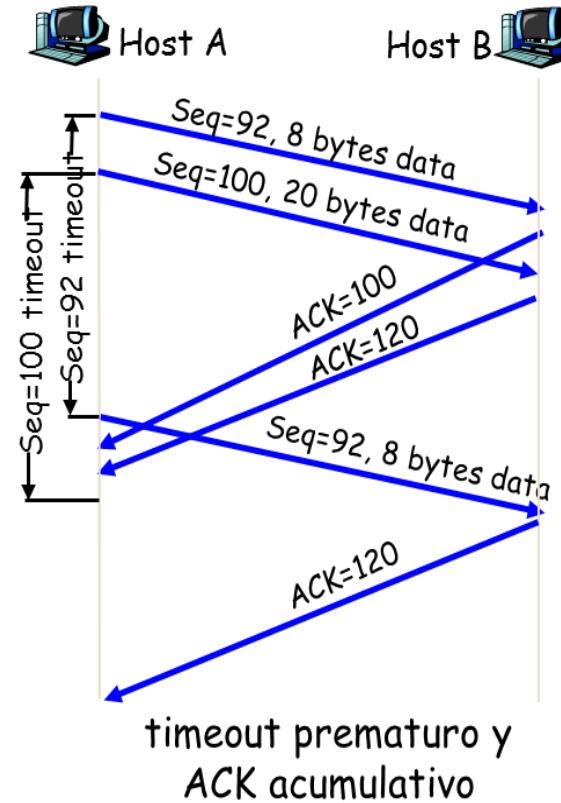
Solicitud de reenvío (desde el error)

Control de errores

- ARQ: Retransmisión



(Gráfico J.F. Kurose)



Control de errores

- ARQ: Protocolo de Generación de ACKS (RFC 1122, 2581)

Evento	Acción del TCP receptor
Llegada ordenada de segmento, sin discontinuidad, todo lo anterior ya confirmado.	Retrasar ACK. Esperar recibir al siguiente segmento hasta 500 mseg. Si no llega, enviar ACK.
Llegada ordenada de segmento, sin discontinuidad, hay pendiente un ACK retrasado.	Inmediatamente enviar un único ACK acumulativo.
Llegada desordenada de segmento con # de sec. mayor que el esperado, discontinuidad detectada.	Enviar un ACK duplicado, indicando el # de sec. del siguiente byte esperado.
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.

Control de errores

- ARQ: ¿Cómo estimar los *timeouts*?
 - Debe ser **mayor que el tiempo de ida y vuelta (RTT, Round Trip Time)**, pero ¿cuánto?
 - Si es **demasiado pequeño**: timeouts prematuros → retransmisiones innecesarias
 - Si es **demasiado grande**: reacción lenta a pérdida de segmentos → baja eficacia
 - Para situaciones cambiantes ... la mejor solución es **adaptarse dinámicamente**.

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

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

$$Desviacion_{nueva} = (1-\beta) \cdot Desviacion_{vieja} + \beta \cdot | RTT_{medido} - RTT_{nuevo} |$$

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

Kurose & Ross

Control de errores

- ARQ: ¿Cómo estimar los *timeouts*?

- 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*:

$$tout_{nuevo} = \gamma \cdot tout_{viejo}, \gamma = 2$$

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

$$RTTnuevo = (1-\alpha) \cdot RTTviejo + \alpha \cdot RTTmedido, \alpha \in [0, 1]$$

$$Desviacion_{nueva} = (1-\beta) \cdot Desviacion_{vieja} + \beta \cdot | RTTmedido - RTTnuevo |$$

$$\text{Timeout} = RTTnuevo + 4 * Desviacion$$

Kurose & Ross

Control de errores

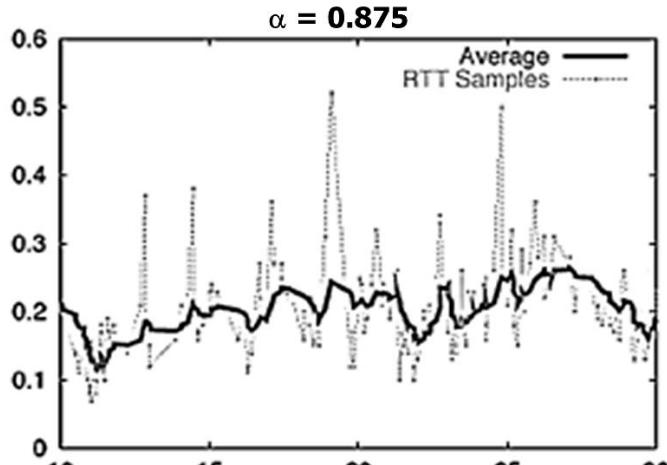
- ARQ: ¿Cómo estimar los *timeouts*?

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

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

$$Desviacion_{nueva} = (1-\beta) \cdot Desviacion_{vieja} + \beta \cdot |RTT_{medido} - RTT_{nuevo}|$$

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

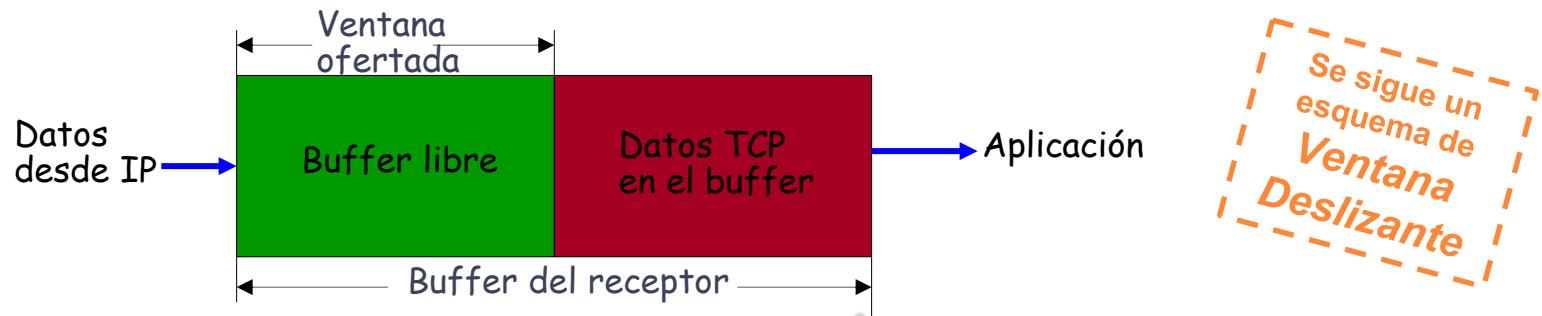


Ejemplo de RTT medidos y estimados entre Amherst, Massachusetts y St. Louis, Missouri.

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**:

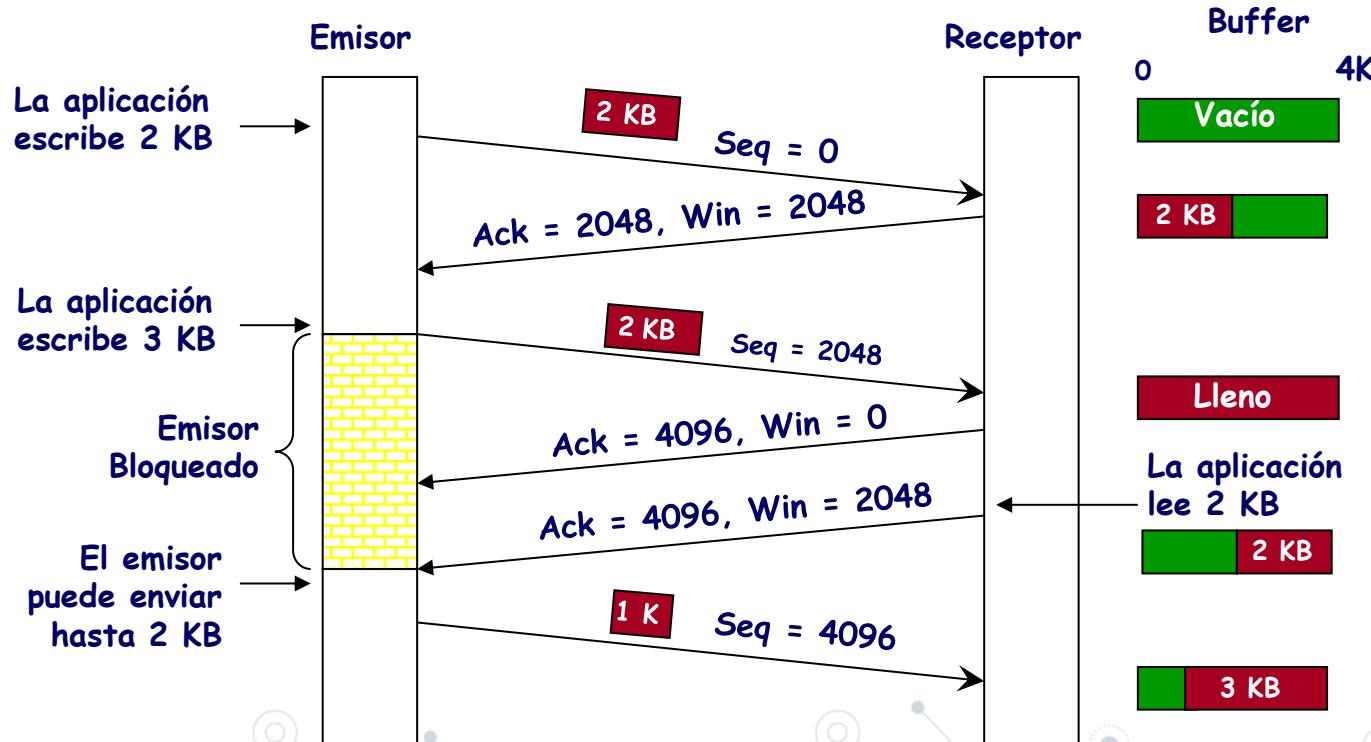
ventana útil emisor = ventana ofertada receptor - bytes en tránsito



Control de flujo

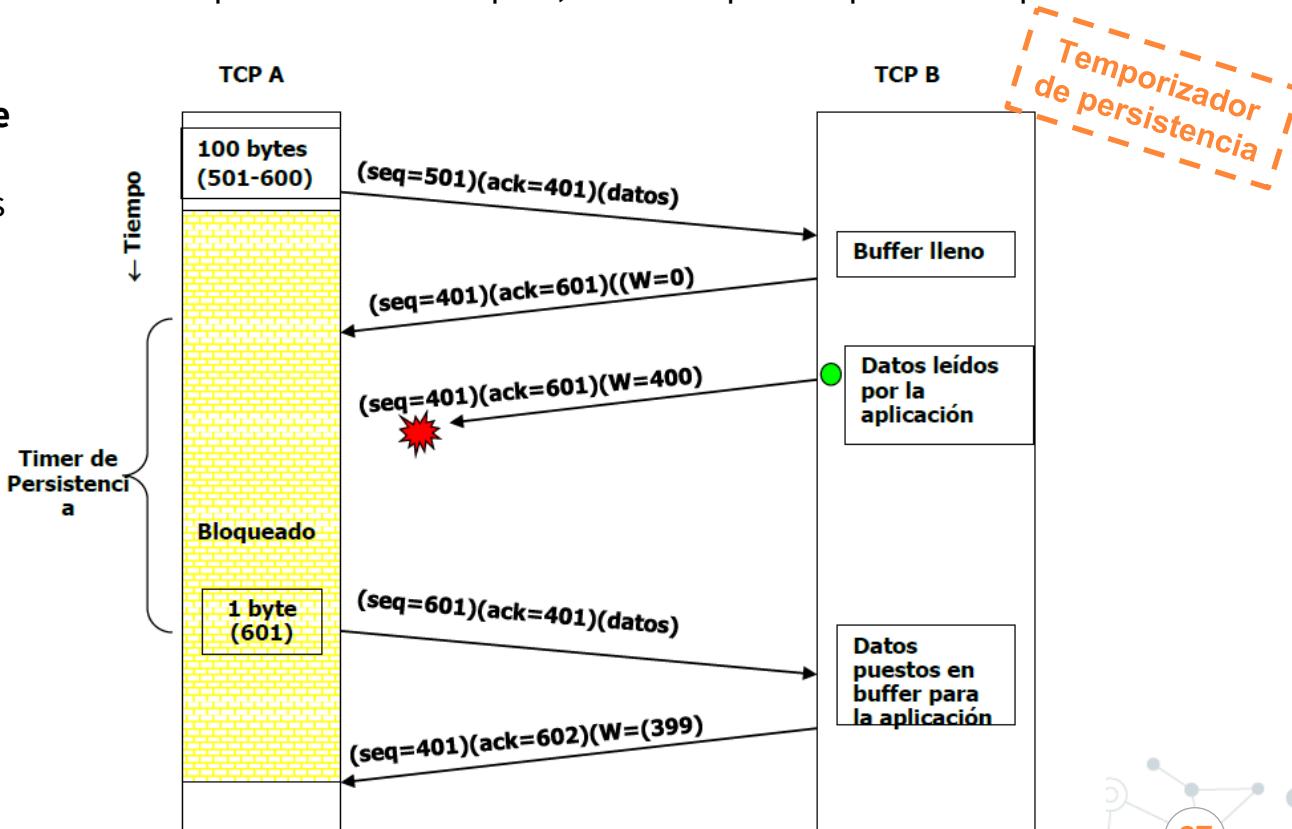
- El TCP **receptor informa** en cada segmento **al emisor** del **espacio** que le queda **libre** en el **buffer** para esa comunicación. Para ello usa el campo **tamaño de ventana (WIN)**.
- Anunciando una **ventana cero** el **receptor puede bloquear al emisor**, y ejercer así **control de flujo**.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

Control de flujo



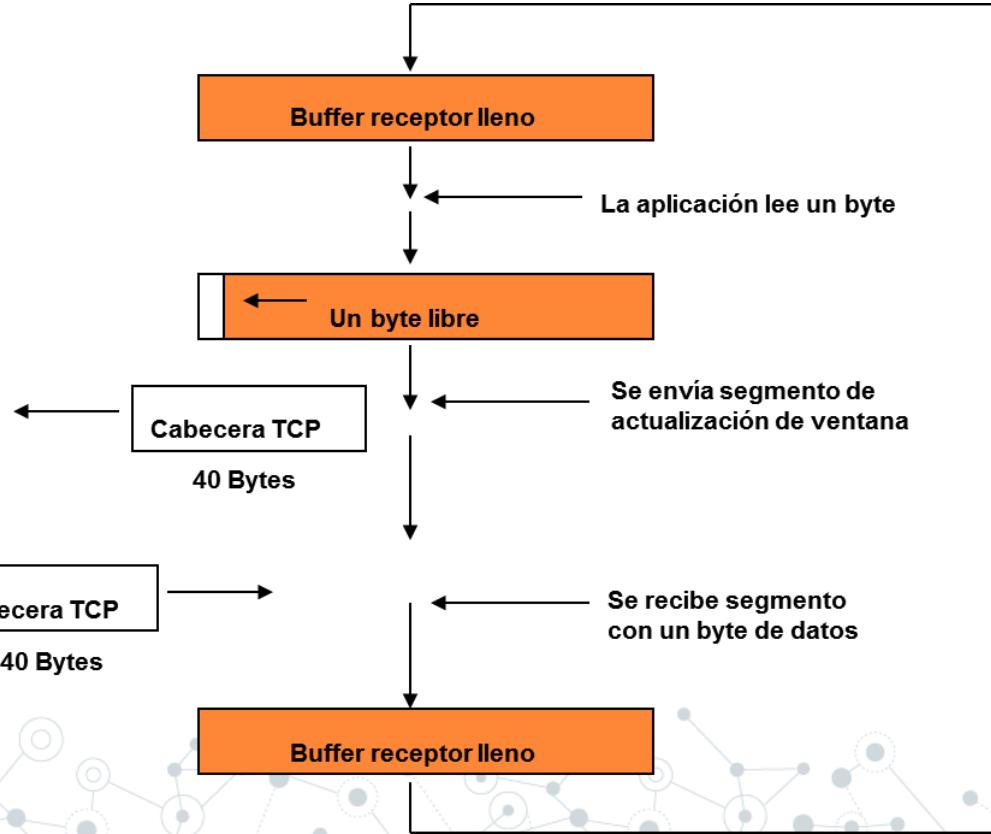
Control de flujo

- 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.
- Possible mejora: la **ventana optimista** (RFC 813) o solución de Clark.



Control de flujo

Síndrome de la ventana tonta (RFC 813)



Control de flujo

Solución de Clark (RFC 813).

- 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
 - La mitad del espacio disponible en el buffer.

TEMA 3. Capa de transporte en Internet

- 3.1. Introducción a los protocolos de Capa de Transporte
- 3.2. Protocolo de datagrama de usuario (UDP)
- **3.3. Protocolo de control de transmisión (TCP)**
 - Multiplexación/demultiplexación
 - Control de conexión
 - Control de errores y de flujo
 - **Control de congestión**
- 3.4. Extensiones TCP
- 3.5. Cuestiones y ejercicios

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**.

Control de congestión

- 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.

Control de congestión

SLOW START (PRUEBA Y ERROR)

- El emisor utiliza **dos ventanas** y un **umbral**.

```
Bytes_permitidos_enviar =
    min{VentanaCongestion, VentanaDelReceptor}
```

VentanaDelReceptor: utilizada para el control de flujo (de tamaño variable) según el campo "ventana" recibido

VentanaCongestion:

Inicialmente VentanaCongestion = $1 \cdot \text{MSS}$

Inicio lento

Si VentanaCongestion < umbral, por cada ACK recibido
 $\text{VentanaCongestion} += \text{MSS}$ (**crecimiento exponencial**)

Prevención de la congestión

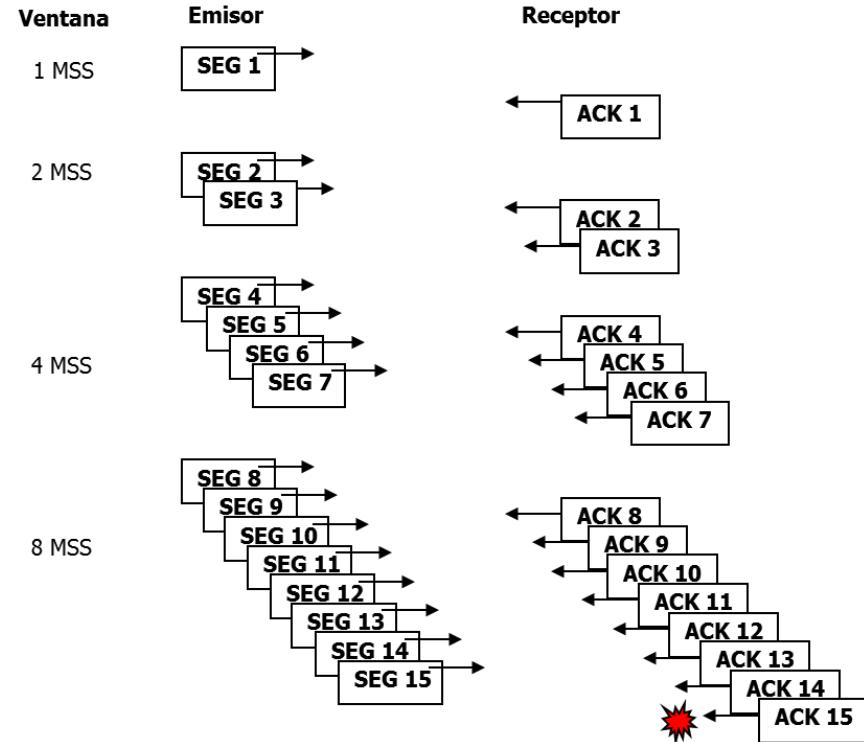
Si VentanaCongestion > umbral, cada vez que se recibe todos los ACKs pendientes
 $\text{VentanaCongestion} += \text{MSS}$ (**crecimiento lineal**)

Si hay timeout entonces
 $\text{umbral} = \text{VentanaCongestion}/2$ y $\text{VentanaCongestion} = \text{MSS}$

Control de congestión

SLOW START (PRUEBA Y ERROR) – PRIMERA FASE

- Inicialmente la ventana de congestión tiene el tamaño de un MSS (Maximum Segment Size)
- 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)
- Si la ventana de congestión supera a la de control de flujo se aplica ésta con lo cual aquella deja de crecer

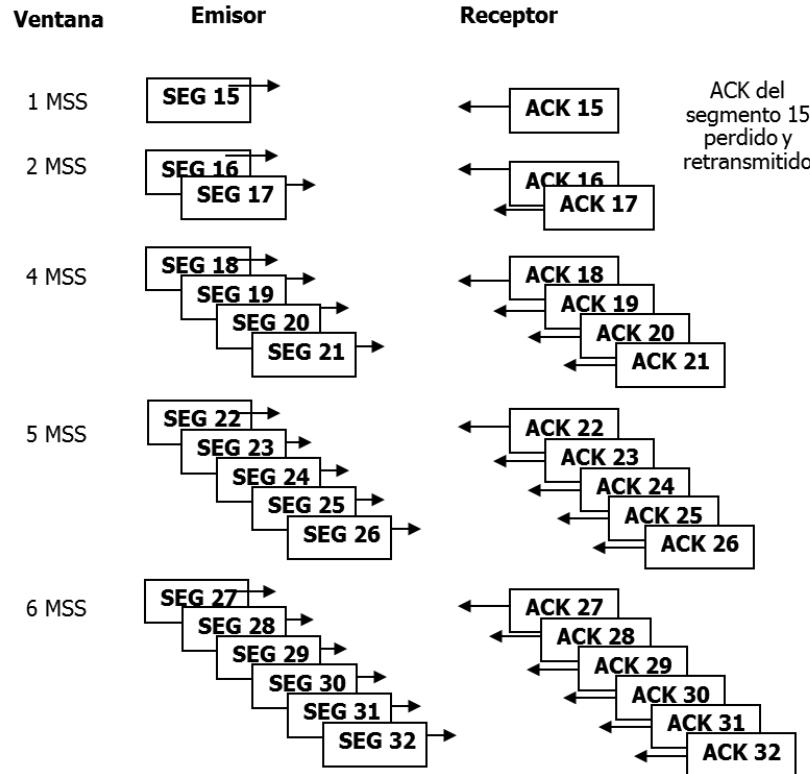


Control de congestión

SLOW START (PRUEBA Y ERROR) – SEGUNDA FASE

Cuando se pierde un segmento:

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



TEMA 3. Capa de transporte en Internet

- 3.1. Introducción a los protocolos de Capa de Transporte
- 3.2. Protocolo de datagrama de usuario (UDP)
- 3.3. Protocolo de control de transmisión (TCP)
 - Multiplexación/demultiplexación
 - Control de conexión
 - Control de errores y de flujo
 - Control de congestión
- **3.4. Extensiones TCP**
- 3.5. Cuestiones y ejercicios

Variantes de 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**

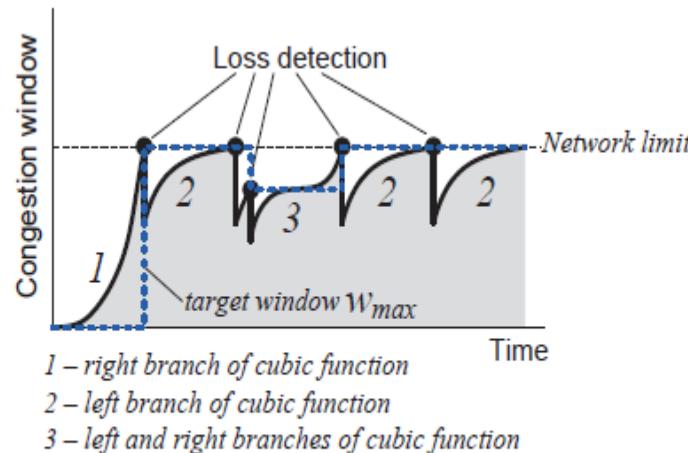
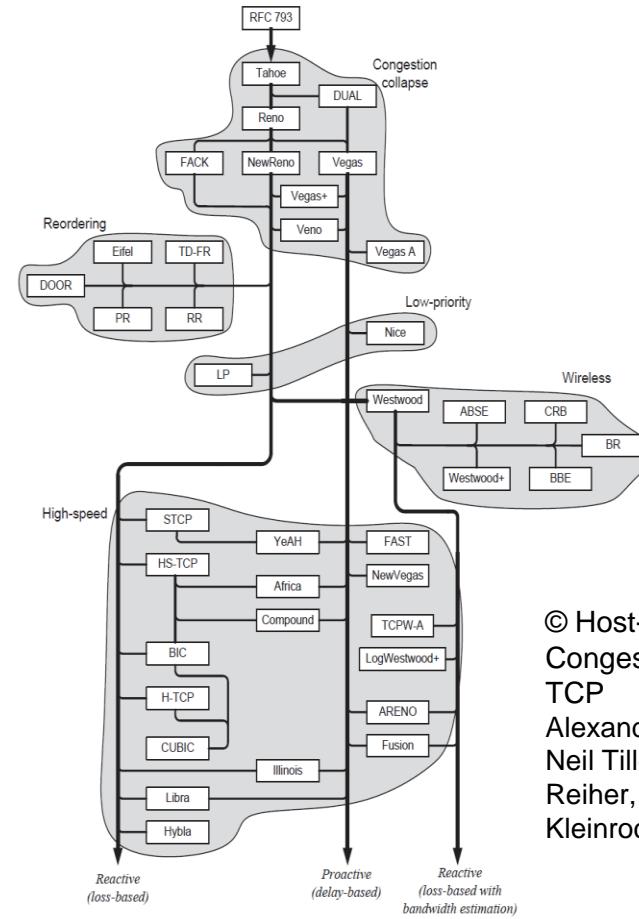


Fig. 50. Congestion window dynamics in CUBIC



57. Evolutionary graph of variants of TCP congestion control.

© Host-to-Host Congestion Control for TCP
Alexander Afanasyev, Neil Tilley, Peter Reiher, and Leonard Kleinrock

Variantes de TCP

- Adaptación de TCP a redes actuales (RFC 1323, 2018).

- **Ventana escalada:**

- Opción TCP en segmentos SYN:

- Hasta $2^{14} \times 2^{16}$ bytes ($= 2^{30}$ bytes = 1GB) autorizados.

- **Estimación RTT:**

- Opción TCP de sello de tiempo, en todos los segmentos.

- **PAWS (“Protect Against Wrapped Sequence numbers”):**

- Sello de tiempo y rechazo de segmentos duplicados.

- **SACK:**

- Confirmaciones selectivas.

¿Preguntas?

O comentarios, sugerencias, inquietudes



Fundamentos de Redes

Tema 4

Redes Conmutadas e Internet

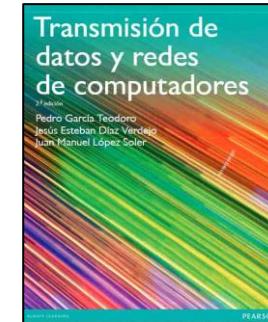
Antonio M. Mora García



Bibliografía

Básica

- P. García-Teodoro, J.E. Díaz-Verdejo, J.M. López-Soler.
Transmisión de datos y redes de computadores, 2^a Edición.
Editorial Pearson, 2014. **CAPÍTULOS 6 y 9**



Complementaria

- James F. Kurose, Keith W. Ross. Redes de computadoras. Un enfoque descendente. 7^º Edición. Editorial Pearson S.A., 2017.
CAPÍTULO 4



Índice

- **4.1.** Funcionalidades
- **4.2.** Comutación
- **4.3.** El protocolo IP
- **4.4.** Asociación con la capa de enlace: El protocolo ARP
- **4.5.** El protocolo ICMP
- **4.6.** Cuestiones y ejercicios

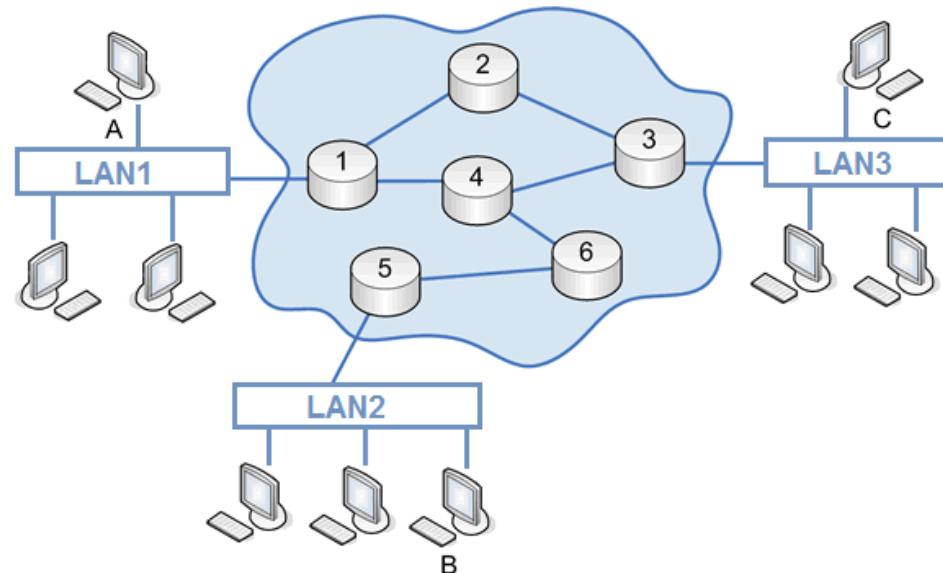
TEMA 4. Redes Conmutadas e Internet

- **4.1. Funcionalidades**
- 4.2. Conmutación
- 4.3. El protocolo IP
- 4.4. Asociación con la capa de enlace: El protocolo ARP
- 4.5. El protocolo ICMP
- 4.6. Cuestiones y ejercicios

Capa de Red

FUNCIONES Y SERVICIOS EN TCP/IP

- 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.



EJEMPLOS DE PROTOCOLOS DE RED

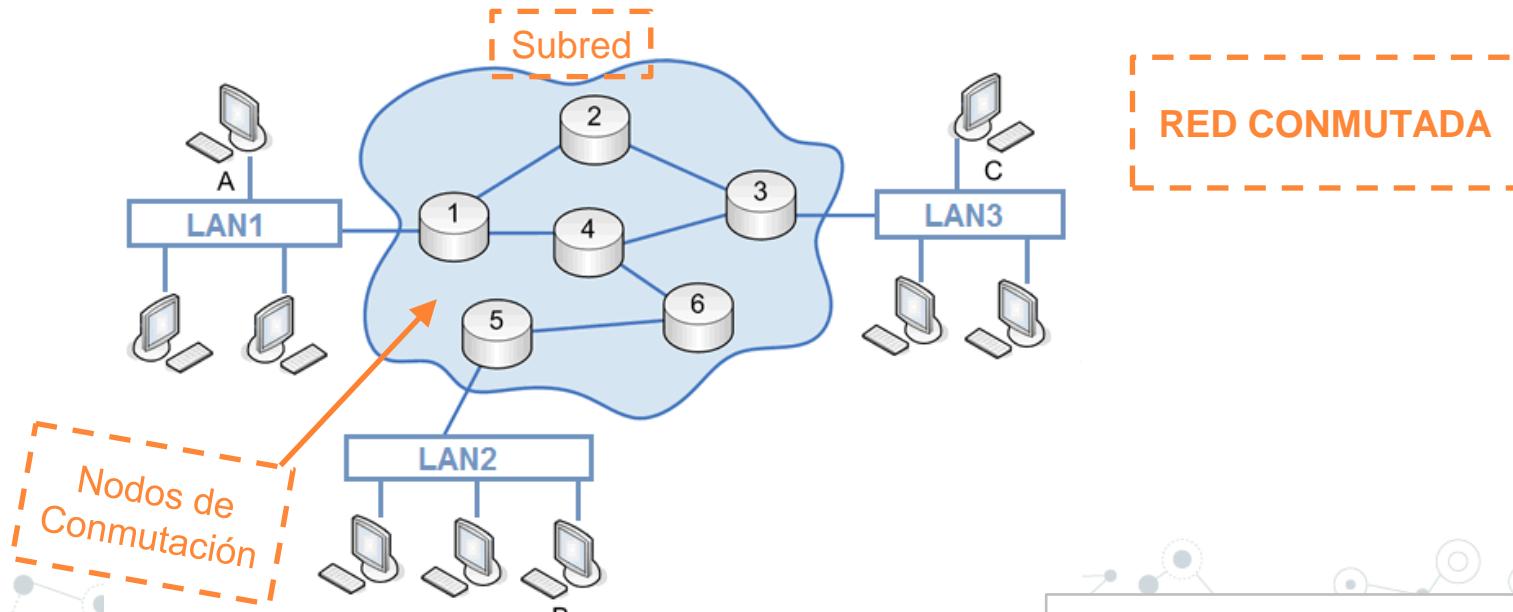
X.25 (https://es.wikipedia.org/wiki/Norma_X.25)

IP

Capa de Red

FUNCIONES Y SERVICIOS EN TCP/IP

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



Capa de Red

FUNCIONES Y SERVICIOS EN TCP/IP

Funciones del protocolo TCP

En el emisor	<ul style="list-style-type: none">• Divide la información en paquetes• Agrega un código detector de errores para comprobar si el paquete llega correctamente a su destino• Pasa el paquete al protocolo IP para que gestione su envío
En el receptor	<ul style="list-style-type: none">• Recibir los paquetes que pasa el protocolo IP• Ordena los paquetes, y comprueba que están todos y que son correctos.• Extrae la información útil de los paquetes• Si detecta un paquete que no ha llegado o que es incorrecto, genera un paquete para ser enviado al emisor, indicándole que lo ha de enviar de nuevo.

TEMA 4. Redes Conmutadas e Internet

- 4.1. Funcionalidades
- **4.2. Conmutación**
- 4.3. El protocolo IP
- 4.4. Asociación con la capa de enlace: El protocolo ARP
- 4.5. El protocolo ICMP
- 4.6. Cuestiones y ejercicios

¿Qué es la conmutación?

- 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**.

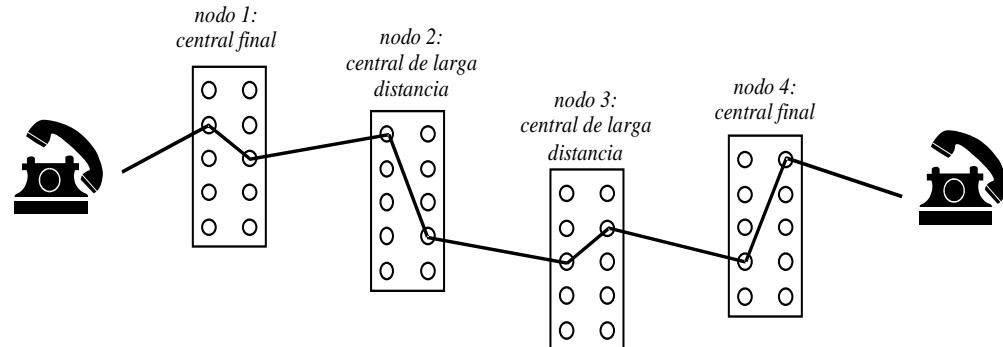
CONMUTACIÓN ⇔ REDIRECCIÓN

¿Qué es la conmutación?

- La conmutación para conectar redes entre sí funciona en la Capa 3 del modelo OSI (Capa de Red).
- Los **servicios** fundamentales que **emplean** técnicas de **conmutación** son:
 - Servicio telefónico
 - Servicio telegráfico
 - Servicio de datos
- **Tecnologías de conmutación:**
 - de circuitos
 - de paquetes (datagramas o circuitos virtuales)

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.
- Ejemplo: *Red telefónica conmutada*



Comutación de circuitos

Servicio
orientado
a conexión

- **Pasos:** (1) Conexión, (2) Transmisión, (3) Desconexión.
- **Establecimiento del circuito:** el host emisor solicita a un cierto nodo de conmutación el establecimiento de conexión hacia un host receptor. Este nodo 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...
- **Transferencia de datos:** una vez establecido el circuito exclusivo para esta transmisión, se transmite desde el emisor hasta el receptor conmutando sin demoras de nodo en nodo (los nodos tienen reservado un canal lógico para ello).
- **Desconexión del circuito:** Terminada la transferencia, el emisor o el receptor indican a su nodo de conmutación 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.



Comutación de circuitos

- **Ventajas:**

- 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 → se transmite a la máxima velocidad posible).
- No hay encamamiento (una vez establecido el circuito) ⇔ transmisión más rápida.
- Simplicidad de gestión en nodos (se recibe siempre por la misma entrada y se transmite siempre por la misma salida).

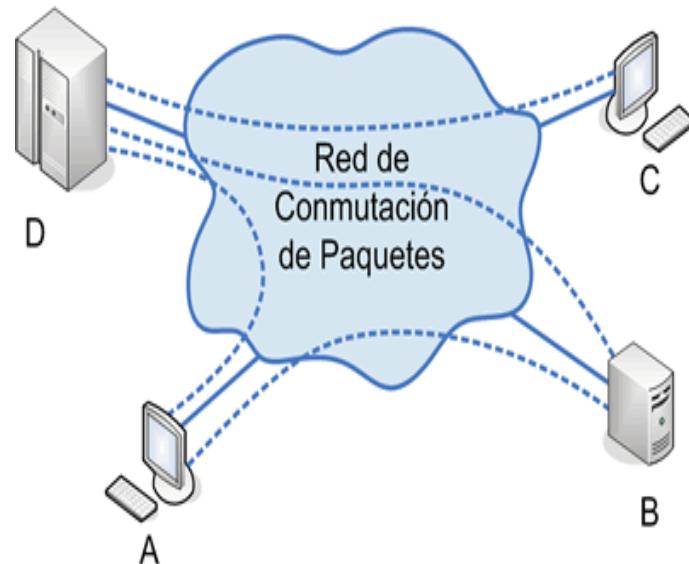
Comutación de circuitos

- **Desventajas:**

- 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).

Comutación de paquetes

- **No es necesario** establecer una **conexión previa**.
- Un **paquete** consta de dos partes:
 - Datos útiles.
 - Información de control (para determinar la ruta a seguir a lo largo de la red hasta el destino).
- Los paquetes permanecen muy poco tiempo en memoria, por lo que resulta muy rápida.
- La comutación de paquetes admite **dos variantes** distintas, según el modo de funcionamiento: **Datagrama** y **Circuitos Virtuales**



Comutación de paquetes

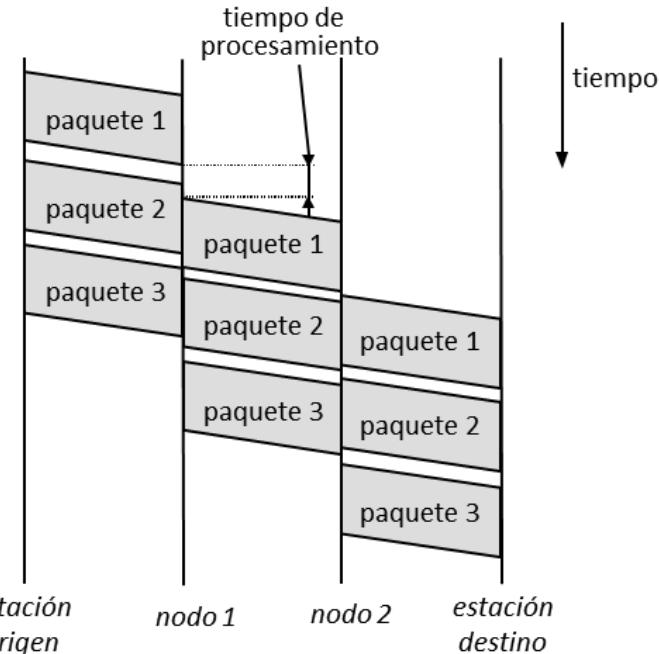
PROCEDIMIENTO:

- Cuando un host quiere enviar **información** a otro lo **divide en paquetes**.
- Se lo **pasará a un nodo intermedio** que será el encargado de transmitirlo al siguiente hacia el destino.
- Cada **nodo intermedio** realiza las siguientes **funciones**:
 - **Almacenamiento y retransmisión** (*store and forward*): el paquete se detiene (se almacena) el tiempo necesario para procesarlo.
 - **Control de ruta** (*routing*): Selección de un nodo del camino por el 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 un orden.

Comutación de paquetes

CONMUTACIÓN DE DATAGRAMAS:

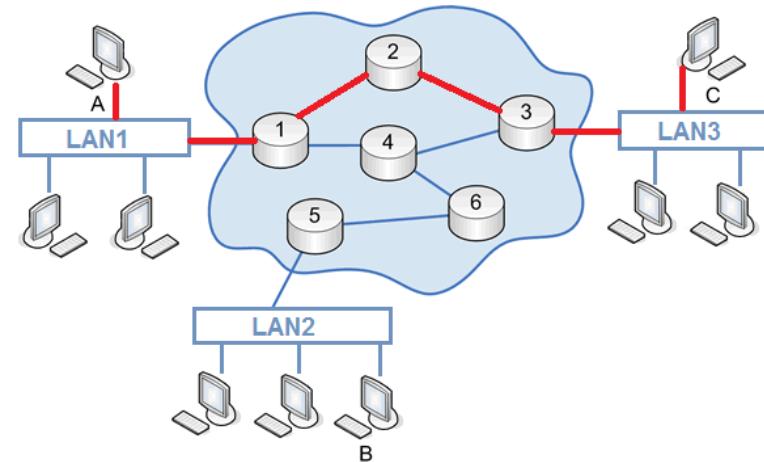
- No hay conexión
- Envío en unidades de datos (**paquetes**) **independientes**
- En cada salto: **almacenamiento y re-envío**
- Cada **paquete** debe contener las **direcciones origen y destino**
- Los **paquetes**, pueden seguir **rutas diferentes** y pueden llegar desordenados
- Ejemplo: **IP**



Comutación de paquetes

CONMUTACIÓN 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)*



Comutación de paquetes

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.

Comutación de paquetes

DESVENTAJAS DE CIRCUITOS VIRTUALES FRENTE A DATAGRAMAS:

- En datagramas no hay que establecer la conexión → para **pocos paquetes**, es **más rápida** la **comutación 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 hace.
- 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).

TEMA 4. Redes Conmutadas e Internet

- 4.1. Funcionalidades
- 4.2. Conmutación
- **4.3. El protocolo IP**
- 4.4. Asociación con la capa de enlace: El protocolo ARP
- 4.5. El protocolo ICMP
- 4.6. Cuestiones y ejercicios

Introducción

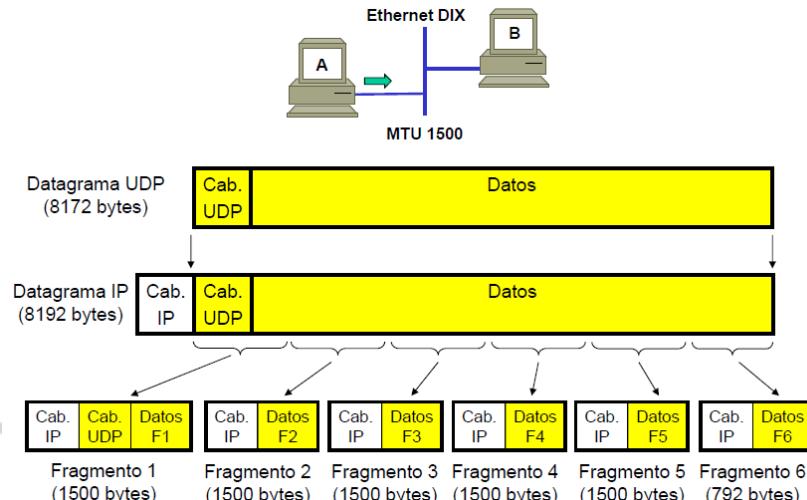
IPv4

- Especificado en el RFC 791 (1349, 2474, 6864).
- 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.

Introducción

IPv4

- 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.
- IP **gestiona la fragmentación**: adaptar el tamaño del datagrama a las diferentes Maximum Transfer Units (MTUs) de las subredes hasta llegar al destino.



Introducción

- Cada entidad en Internet se **identifica por su dirección IP**.



Servidor
Webmail
130.206.192.39



www.google.com
172.194.34.209

Cada dirección IP es única en Internet



Servidor
Spotify
78.31.8.101



www.youtube.com
172.194.34.206



www.ugr.es
150.214.204.25
dns3.ugr.es
150.214.191.10
pop.ugr.es
150.214.20.3

Direcciones IP

IPv4

- Una **dirección IP** \Leftrightarrow **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

Direcciones IP

IPv4

- Cada dirección IP tiene **dos partes** bien diferenciadas:
 - Un **identificador de la subred** (parte izquierda de la IP)
 - Un **identificador del dispositivo** dentro de esa subred (parte derecha de la IP).
- Cada **subred** tiene un **identificador único en la intranet** (red privada).
- Cada **dispositivo (interfaz)** tiene un **identificador único en la subred**.

Direcciones IP

IPv4

- La **máscara de red** es un patrón de ‘1s’ que **determina qué bits** de la IP completa corresponden al **identificador de subred**.

Ejemplo:

Dirección IP: 200.27.4.112 → 11001000.00011011.00000100.01110000

Máscara: 255.255.255.0 → 11111111.11111111.11111111.00000000

- La máscara se puede representar de **forma compacta**, indicando el número de ‘1s’ que tiene.

Ejemplo:

255.255.255.0 → 11111111.11111111.11111111.00000000 ⇔ /24

La dirección anterior con la máscara sería: 200.27.4.112/24

Direcciones IP

IPv4

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

Ejemplo:

200.27.4.112 → 11001000.00011011.00000100.01110000

&

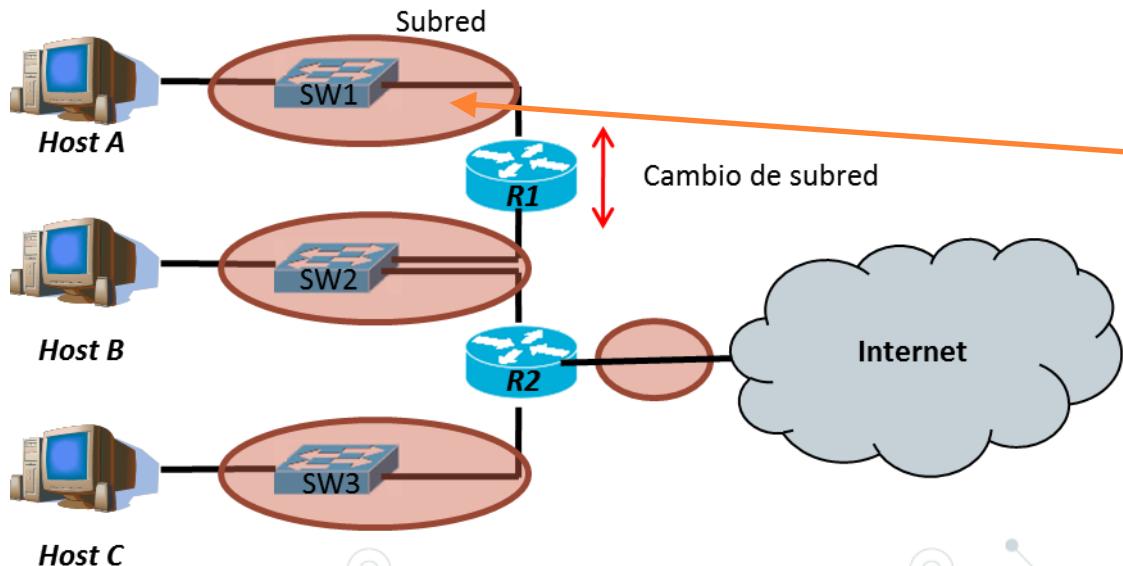
&

255.255.255.0 → 11111111.11111111.11111111.00000000

Subred ➔ 200.27.4.0 ⇔ 11001000.00011011.00000100.00000000

Subredes

- Podemos considerar Internet como un conjunto de subredes interconectadas
- ¿Qué es una subred? ¿Qué es un switch? ¿Qué es un router?

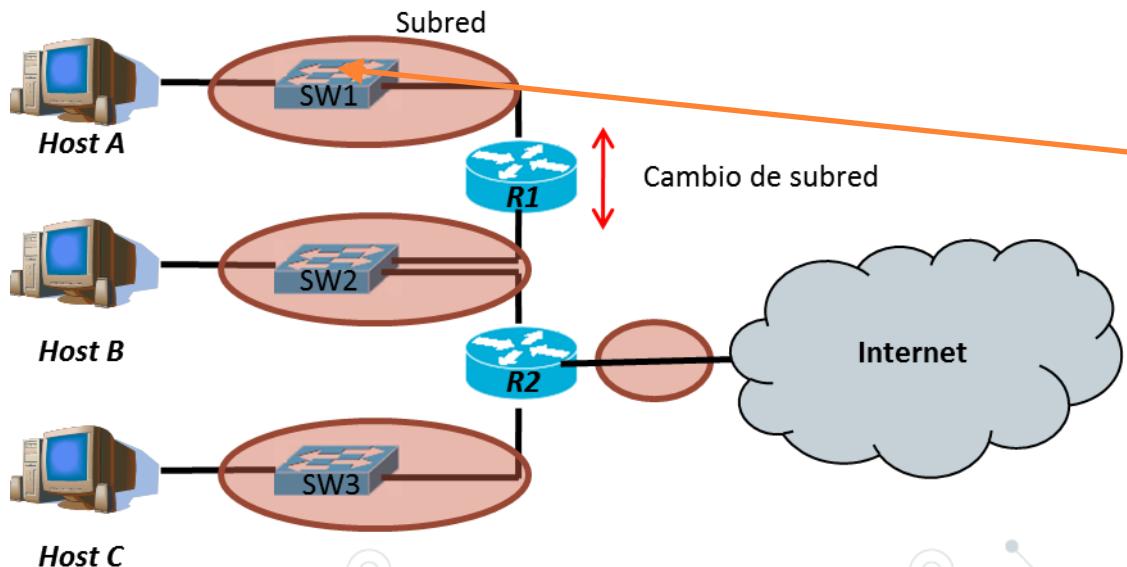


SUBRED

Líneas de transmisión e infraestructura de red que permite la **conexión directa** de dispositivos IP sin intermediarios (un switch se considera transparente)

Subredes

- Podemos considerar Internet como un conjunto de subredes interconectadas
- ¿Qué es una subred? ¿Qué es un switch? ¿Qué es un router?

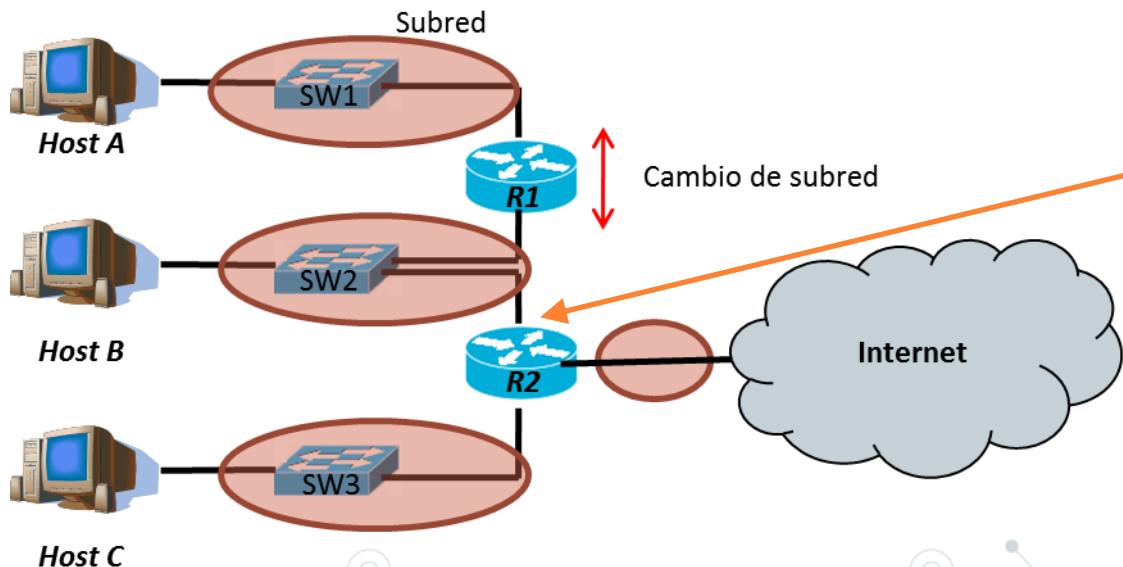


SWITCH

O conmutador.
Se usa para crear redes de computadoras. Son “transparentes”.
Trabaja a nivel de enlace (Capa 2 de OSI).

Subredes

- Podemos considerar Internet como un conjunto de subredes interconectadas
- ¿Qué es una subred? ¿Qué es un switch? ¿Qué es un router?



ROUTER

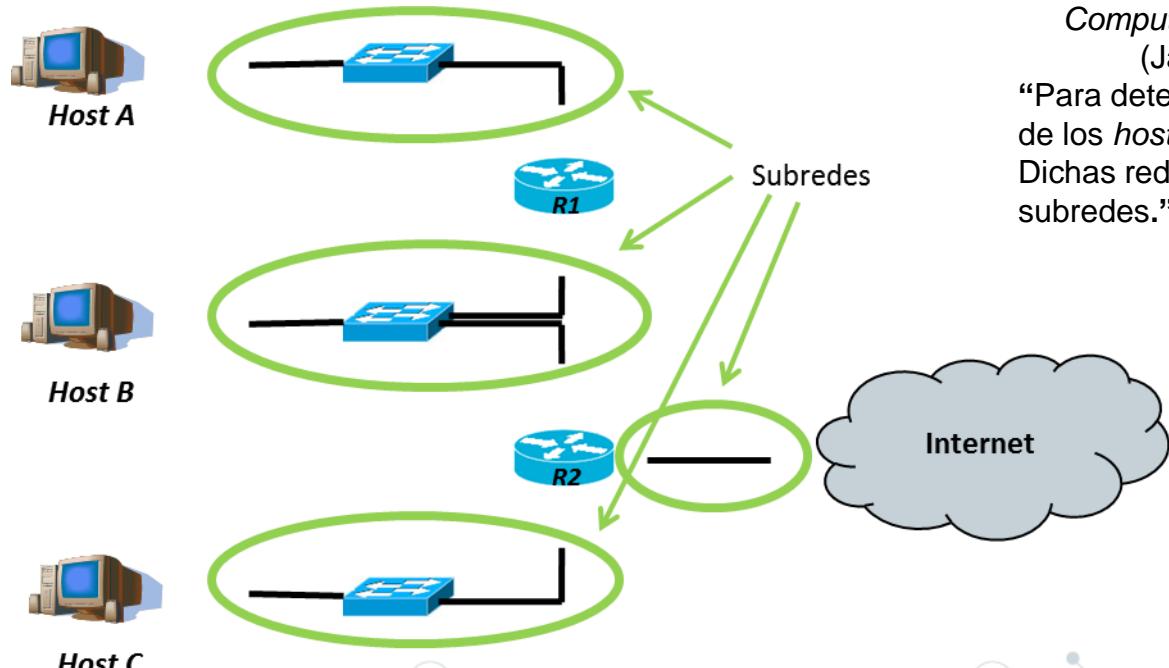
O encaminador.
Se usa para conectar redes entre sí. Es un punto de separación, ya que limita el tráfico entre las redes.

Redirige los paquetes hacia el destino de una transmisión.

Trabaja a nivel de red
(Capa 3 de OSI).

Subnetting

- ¿Cómo determinar las subredes en un esquema de red?



Computer Networking. A Top-down Approach.

(James F. Kurose y Keith W. Ross)

“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

Subnetting

- ¿Cómo se elige la máscara? → Según el número de dispositivos que necesitemos direccionar en la subred, tal que se ajusta para no desaprovechar direcciones.

Dirección IP → 200.27.4.112 = 11001000.00011011.00000100.01110000

Máscara → 255.255.255.0 = 11111111.11111111.11111111.00000000

Número de dispositivos = $2^{\text{número_ceros}} - 2$

Ej: 8 ceros (/24) permite 254 dispositivos

El -2 viene de que la primera IP y última son reservadas

Recuérdese:
Cada subred tiene
un identificador
único en nuestra
intranet

Subnetting

(Máscara /24)

La dirección de Red/Subred tiene todo a 0s en la parte de host

- 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)

La dirección de Difusión/Broadcast tiene todo a 1s en la parte de host

Tipos de direcciones IP

PÚBLICAS:

- Cada dirección se asigna a sólo 1 dispositivo (una interfaz) en toda la Internet global.
- Se asignan centralizadamente.

PRIVADAS:

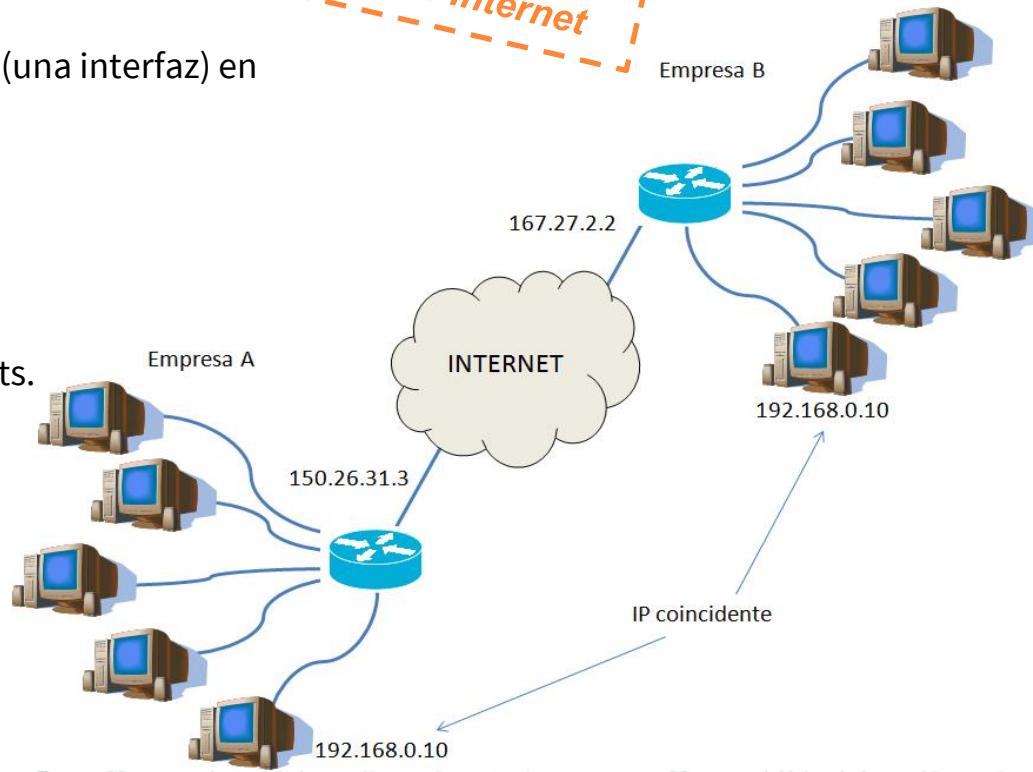
- Sólo sirven para tráfico dentro de las intranets.
- Se pueden repetir en distintas intranets.
- Las asigna el usuario según su criterio.
- Rangos de IPs privadas:

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

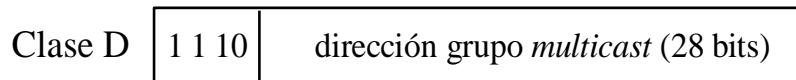
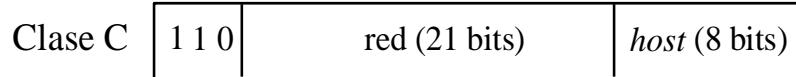
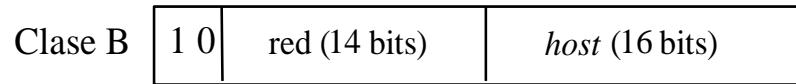
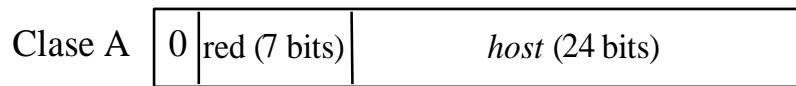
*Las IPs públicas
son únicas en
todo Internet*



Clases de direcciones IP

- Especificadas en RFCs 1166 y 5737.
- Originariamente se definieron **5 clases de direcciones IP**.
- Clases **A, B, C → Jerárquicas a dos niveles**:

identificador de red + identificador de dispositivo (host)



Internet Assigned Numbers Authority



Internet Corporation for
Assigned Names and Numbers

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

- Reglas especiales:

- host = 00...0 → identifica a una red, nunca es una dirección origen, no se usa para dispositivos
- host = 11...1 → difusión en la red especificada, es una dirección destino, no se usa para dispositivos
- 127.0.0.0 → autobucle (loopback)

- Reserva de direcciones privadas (RFC 1918):

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...

IPv6

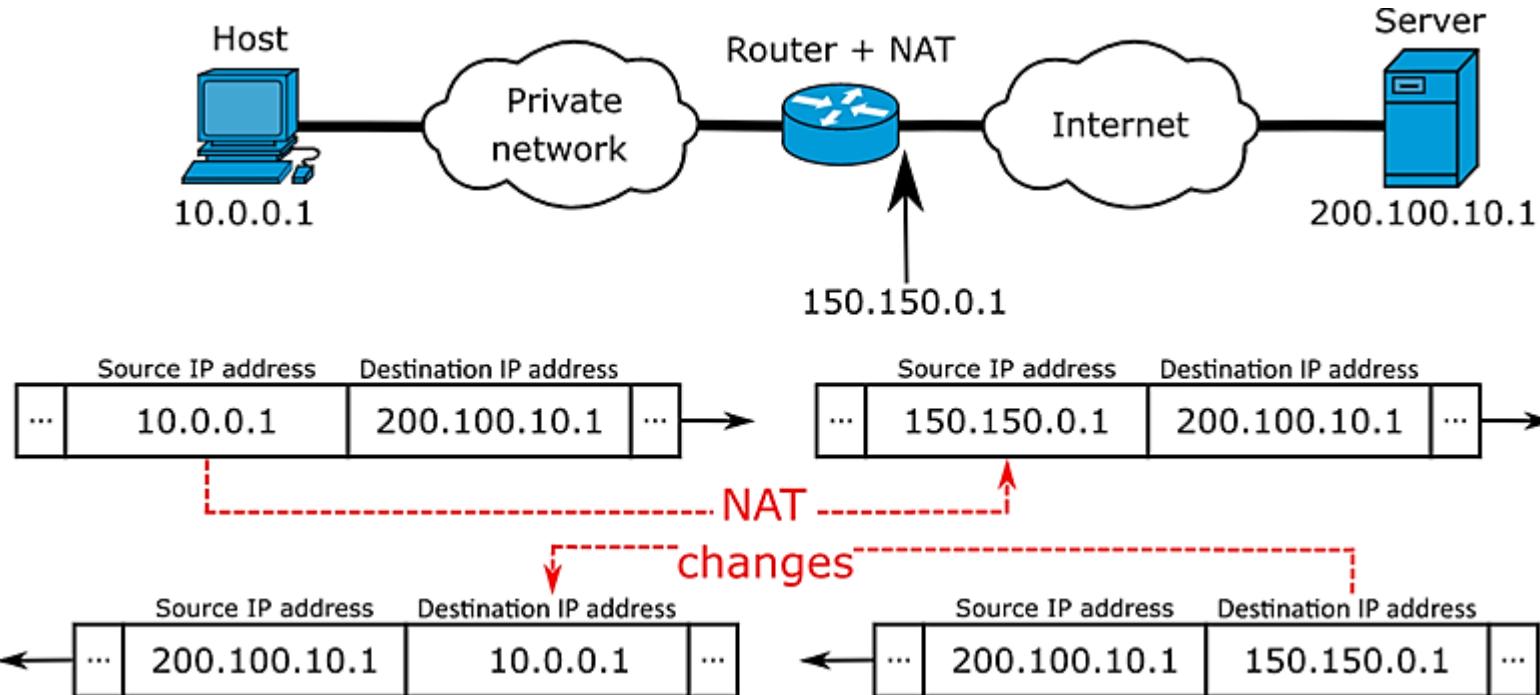
- IPv6 usa un esquema de **direccionamiento de 128 bits**.
- **Notación hexadecimal.** 8 grupos de 4 dígitos, separados por “:”.
- Cada dígito hexadecimal corresponde a 4 dígitos en binario (4 bits).
- Rango: 0000:0000:0000:0000:0000:0000:0000:0000 a
FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
- 340.282.366.920.938.463.463.374.607.431.768.211.456 (340 sextillones) direcciones diferentes.
- **Compatible con IPv4.**

NAT (Network Address Translation)

- RFC 1631, 2663, 3022.
- Consiste en **traducir un conjunto de direcciones IPv4 en otras**.
- Permite que una red con direccionamiento privado se pueda conectar a Internet (direccionamiento 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)

NAT (Network Address Translation)



NAT (Network Address Translation)

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.

NAT (Network Address Translation)

PROTO	TCP
SADDR	10.0.0.3
DADDR	128.32.32.68
SPORT	1049
DPOR	80
FLAGS	SYN
CKSUM	0x1636

1. El cliente intenta conectarse al servidor web 128.32.32.68 y envía un paquete SYN con su dirección IP interna 10.0.0.3 (privada).

PROTO	TCP
SADDR	24.1.70.210
DADDR	128.32.32.68
SPORT	40960
DPOR	80
FLAGS	SYN
CKSUM	0x2436

2. El dispositivo NAT ve la configuración del paquete, añade una nueva entrada a su tabla de traducción. Luego modifica el paquete usando su dirección IP externa (pública), cambia el puerto y el chequeo de integridad del paquete.



PROTO	TCP
SADDR	128.32.32.68
DADDR	10.0.0.3
SPORT	80
DPOR	1049
FLAGS	SYN, ACK
CKSUM	0x8041

4. El dispositivo NAT mira su tabla de traducción, y encuentra la que corresponde a direcciones y puertos origen/destino. Reescribe el paquete utilizando los puertos y direcciones internas.

Original	NAT
10.0.0.3:1049	24.1.70.210:40960
...	...

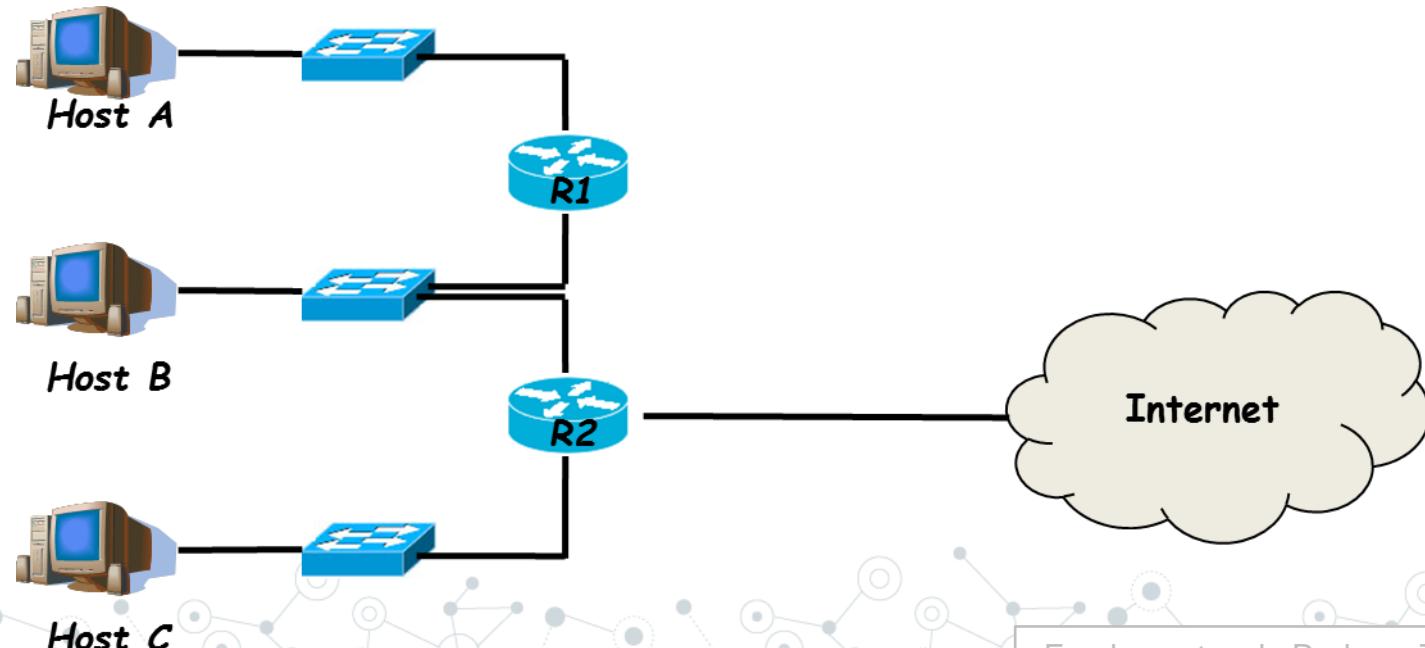
PROTO	TCP
SADDR	128.32.32.68
DADDR	24.1.70.210
SPORT	80
DPOR	40960
FLAGS	SYN, ACK
CKSUM	0x8041

3. El servidor responde con un paquete SYN, ACK. El paquete se envía a la dirección IP externa (pública) del dispositivo NAT.

Ejercicio

ASIGNACIÓN DE DIRECCIONES IP

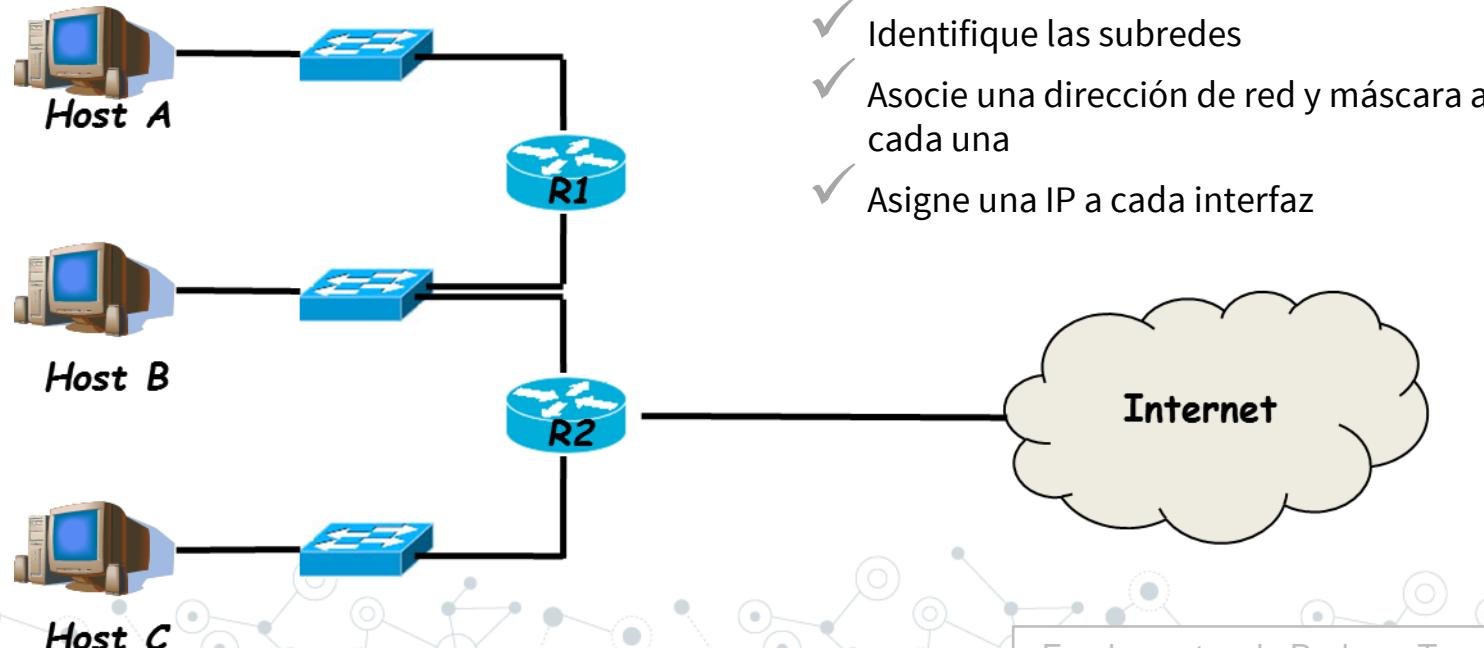
- Subredes corporativas: 30 dispositivos cada una, direcciones privadas en el rango 192.168.0.0
- Subred de acceso: dirección pública (ISP)



Ejercicio

ASIGNACIÓN DE DIRECCIONES IP

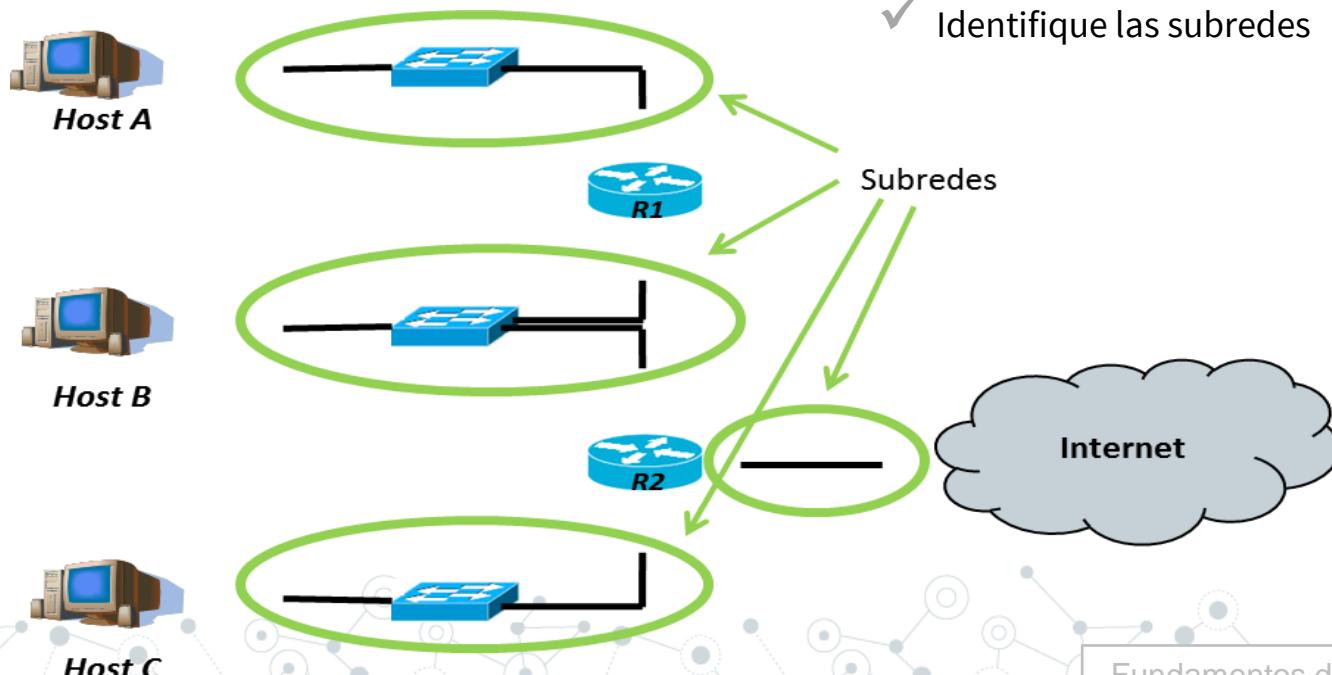
- Subredes corporativas: 30 dispositivos cada una, direcciones privadas en el rango 192.168.0.0
- Subred de acceso: dirección pública (ISP)



Ejercicio

ASIGNACIÓN DE DIRECCIONES IP

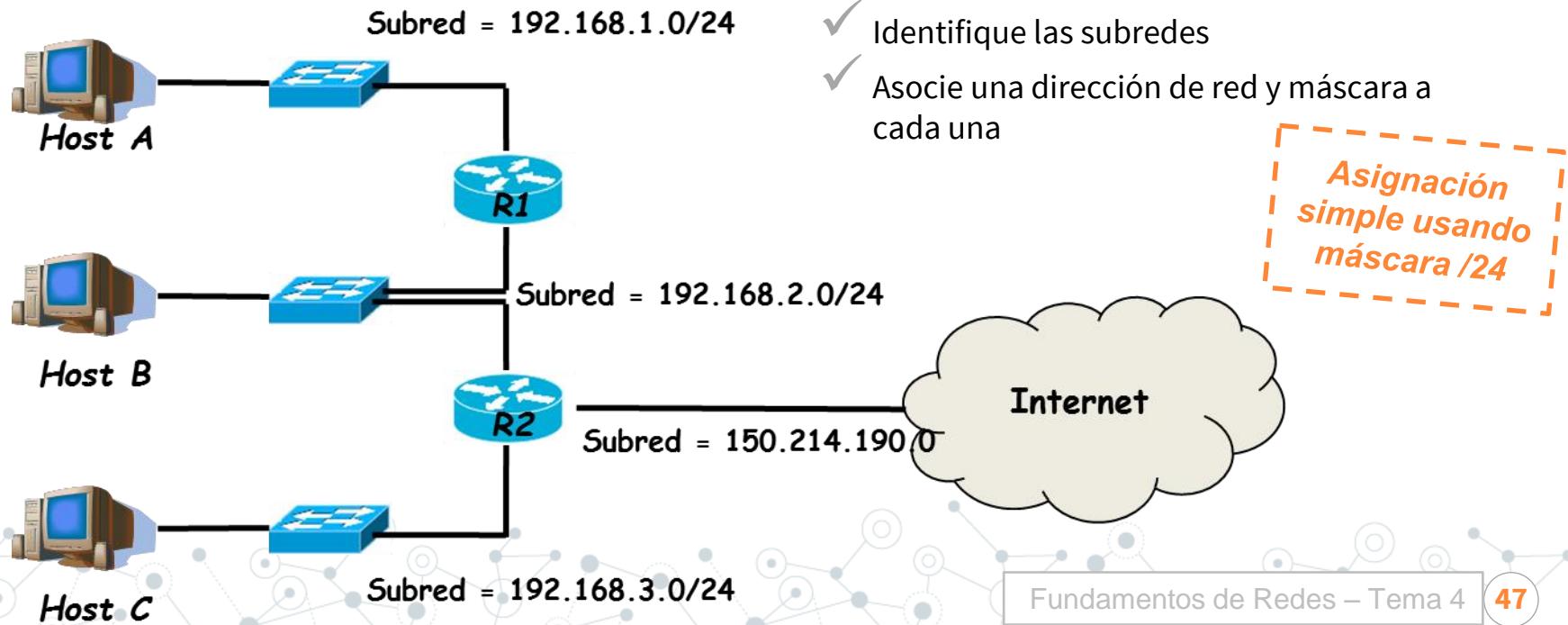
- Subredes corporativas: 30 dispositivos cada una, direcciones privadas en el rango 192.168.0.0
- Subred de acceso: dirección pública (ISP)



Ejercicio

ASIGNACIÓN DE DIRECCIONES IP

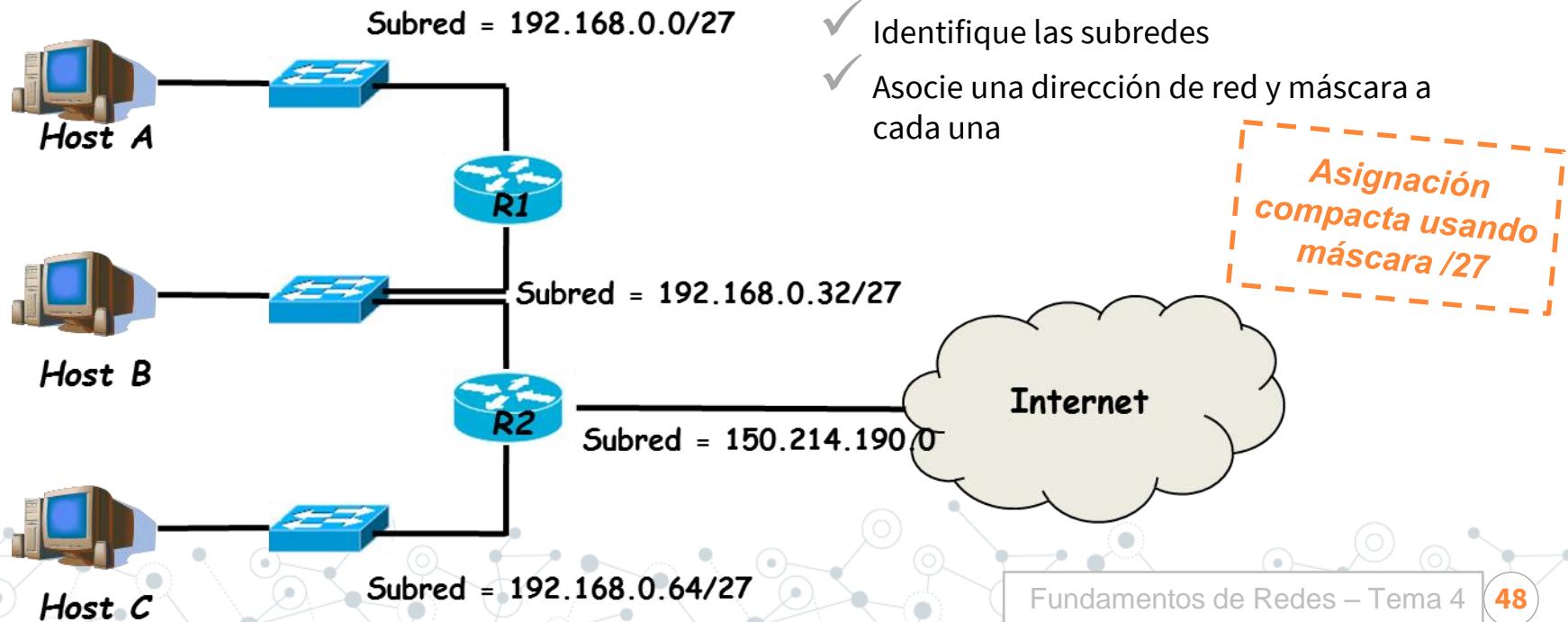
- Subredes corporativas: 30 dispositivos cada una, direcciones privadas en el rango 192.168.0.0
- Subred de acceso: dirección pública (ISP)



Ejercicio

ASIGNACIÓN DE DIRECCIONES IP

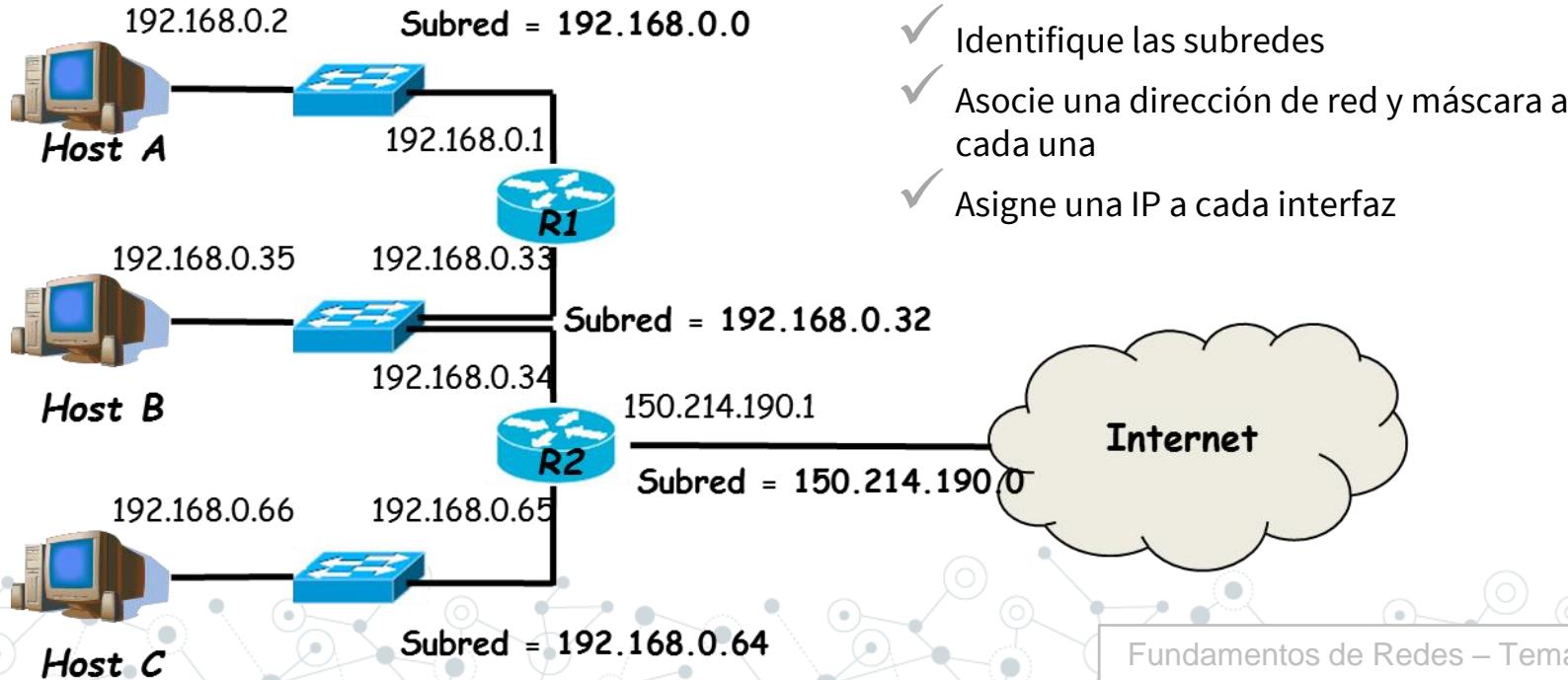
- Para direccionar 30 dispositivos → 5 bits en la parte de hosts. $32-5=27$ bits para red → máscara /27
- Dirección pública ISP: 2 bits, /30, consideramos por ejemplo 150.214.190.0 (UGR)



Ejercicio

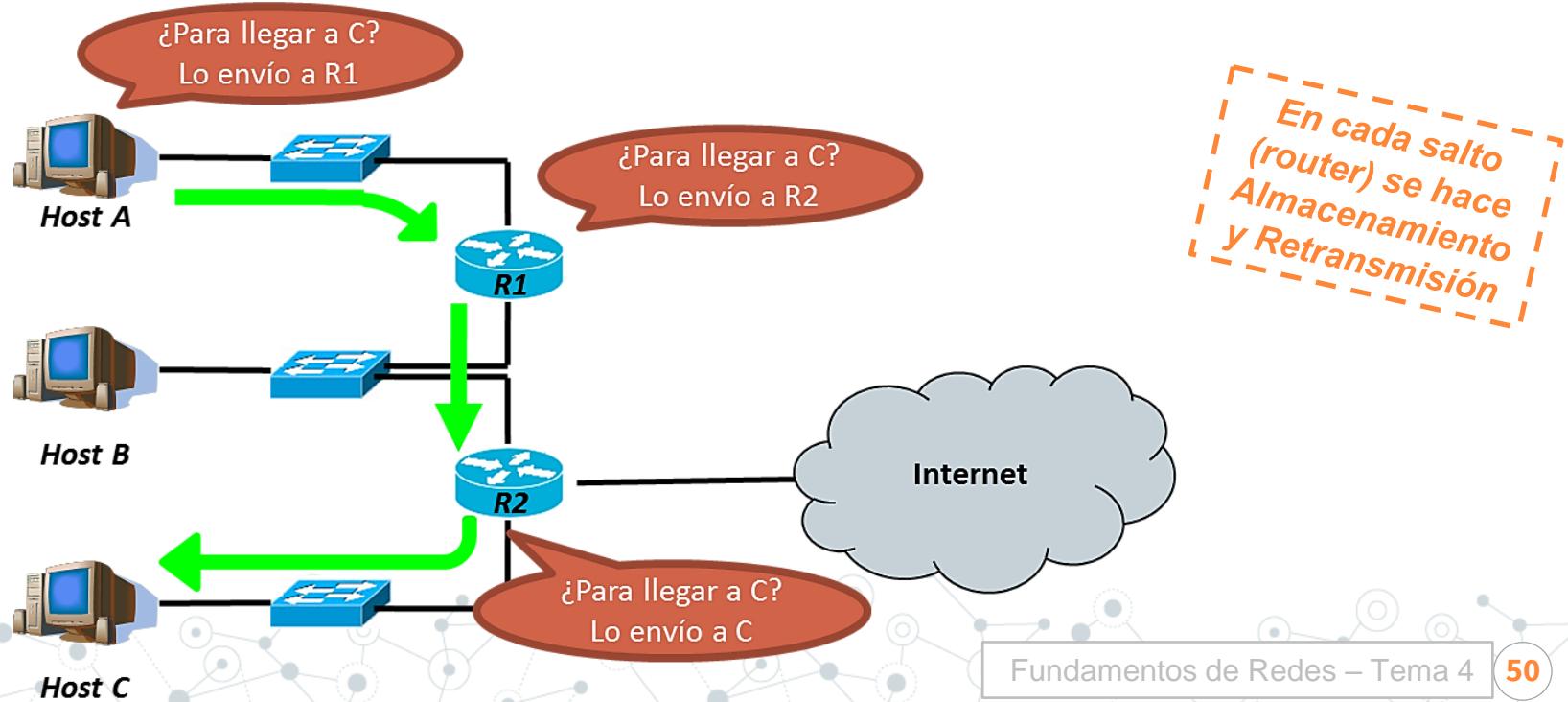
ASIGNACIÓN DE DIRECCIONES IP

- Para direccionar 30 dispositivos → 5 bits en la parte de hosts. $32-5=27$ bits para red → máscara /27
- Dirección pública ISP: 2 bits, /30, consideramos por ejemplo 150.214.190.0 (UGR)



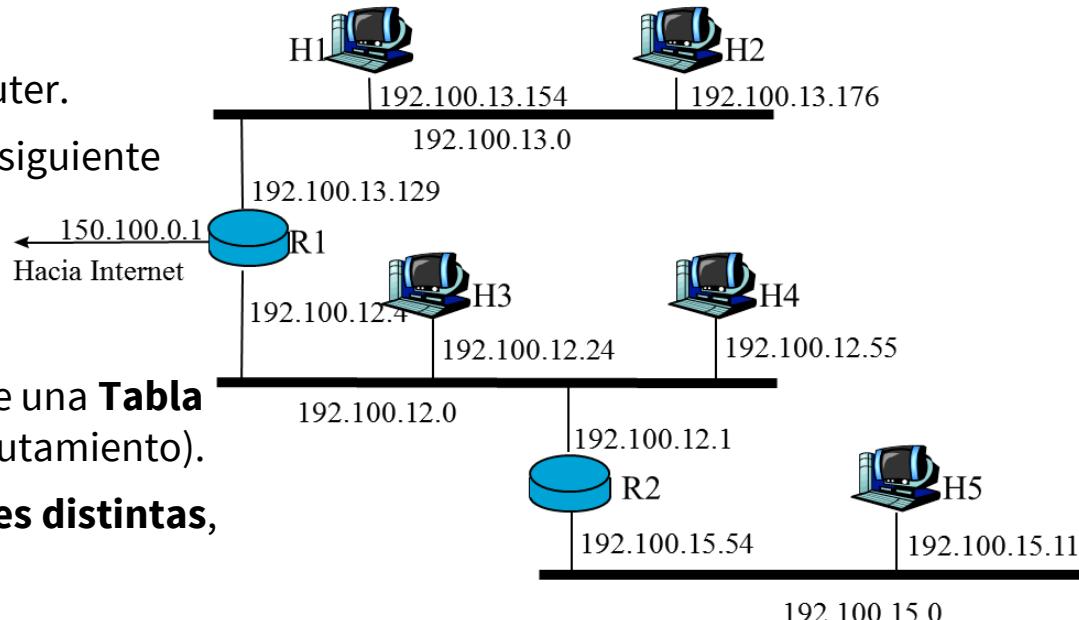
Encaminamiento (Enrutamiento)

- **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).



Encaminamiento (Enrutamiento)

- El encaminamiento se realiza **salto a salto** y **datagrama a datagrama** (IP es no orientado a conexión).
- Modos de encaminamiento:
 - **directo** → lo resuelve el propio router.
 - **no directo** → lo resuelve el router siguiente en la ruta.
- Cada dispositivo (host o router) tiene una **Tabla de encaminamiento** (o Tabla de enrutamiento).
- 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
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

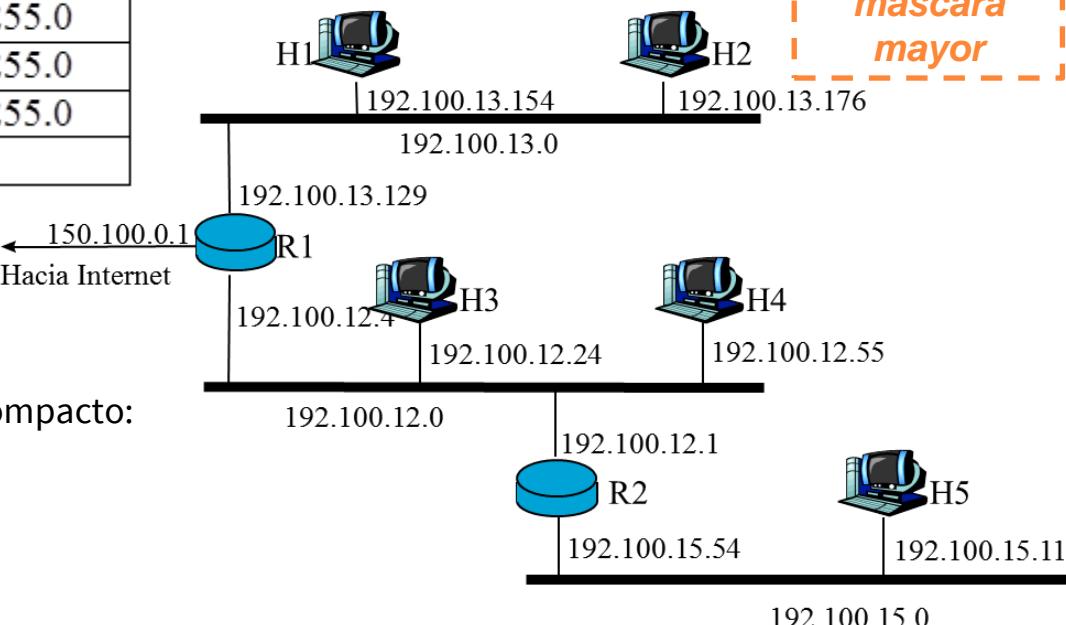
- ¿Faltaría alguna entrada?
Una específica a la red 150.100.0.0/30
- La máscara se puede indicar en formato compacto:

255.255.255.0 \Leftrightarrow /24

255.255.255.192 \Leftrightarrow /26

255.255.255.252 \Leftrightarrow /30

Si hay dos entradas en conflicto se elige la más restrictiva \Leftrightarrow máscara mayor



Encaminamiento (Enrutamiento)

i	Destino (D_i)	Salto siguiente (S_i)	Máscara (M_i)	Flags	Interfaz(I_i)
1	127.0.0.1	*	255.255.255.255	H	lo
2	192.100.12.0	*	255.255.255.0	-	eth0
.	192.100.13.0	*	255.255.255.0	-	eth1
.	192.100.15.0	192.100.12.1	255.255.255.0	G	eth0
N	Default	150.100.0.222	0.0.0.0	G	eth2

PROCESO DE ENCAMINAMIENTO (EN CADA NODO Y PARA CADA DATAGRAMA)

- Se extrae la dirección destino: IP_DESTINO del datagrama
- Por cada entrada i con $i = 1, \dots, N$, de la tabla de encaminamiento se calcula:

$$IP_i = IP_DESTINO \text{ AND} (\&) \text{ MASCARA}_i$$
- Si $IP_i = D_i$ y
 si es routing directo (*) → reenviar el datagrama al destino final por la interfaz i
 o si no es routing directo → reenviar 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)
- Si se ha barrido toda la tabla y no hay coincidencia con ninguna fila → error (posible mensaje ICMP)

Ejemplo encaminamiento

TABLA DE ENCAMINAMIENTO

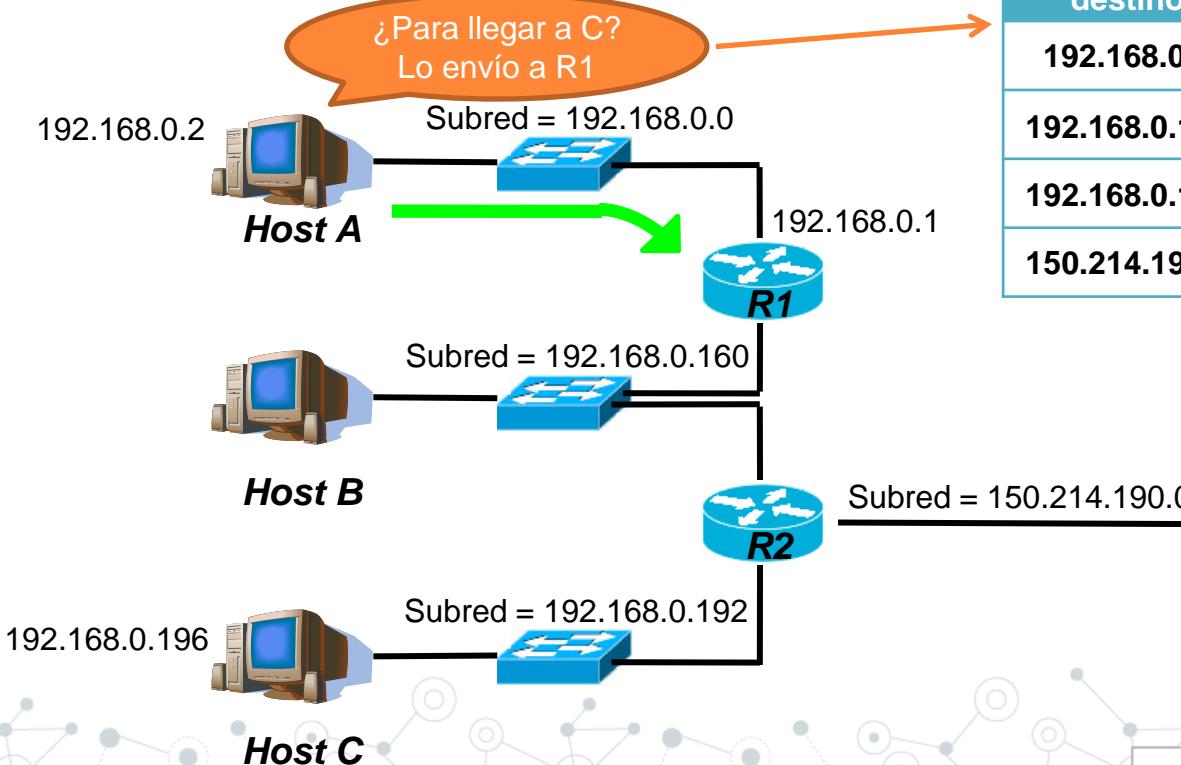


Tabla de Host A

Ejemplo encaminamiento

- Se comprueba la **tabla de Host A**.
- Dirección de destino (DD): 192.168.0.196
- Para cada entrada (fila en la tabla)
- DD & Máscara = A
- ¿A = Dirección de destino en tabla?
 SI → elegir el "Siguiente Nodo" → consultar TABLA ARP
 NO → seguir buscando

Esto se repite en cada router

Tabla de Host A

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

➤ 192.168.0.196 & /27 = 11000000.10101000.00000000.110**00100** & /27 = 192.168.0.192

¿192.168.0.192 = 192.168.0.0? NO

➤ 192.168.0.196 & /27 = 11000000.10101000.00000000.110**00100** & /27 = 192.168.0.192

¿192.168.0.192 = 192.168.0.160? NO

➤ 192.168.0.196 & /27 = 11000000.10101000.00000000.110**00100** & /27 = 192.168.0.192

¿192.168.0.192 = 192.168.0.192? SÍ ➔ **Siguiente Nodo = 192.168.0.1**

➤ 192.168.0.196 & /30 = 11000000.10101000.00000000.110001**00** & /30 = 192.168.0.196

¿192.168.0.196 = 150.214.190.0? NO

Ejemplo encaminamiento

PROBLEMA

- La topología implica sólo un camino de salida desde Host A → ¿realmente necesitamos 4 entradas?

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

Podemos agrupar entradas

Podremos agrupar entradas de la tabla que tengan distinto destino, pero el mismo salto siguiente

Ejemplo encaminamiento

PROBLEMA

- La topología implica sólo un camino de salida desde Host A → ¿realmente necesitamos 4 entradas?

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

Podemos agrupar entradas

Buscamos los bits en común (iguales):

192.168.0.160 ⇔ **11000000.10101000.00000000.1**0100000

192.168.0.192 ⇔ **11000000.10101000.00000000.1**1000000

La máscara del agrupamiento indicará el número de bits iguales → **/25**

La dirección agrupada será la parte común y el resto de bits a 0:

11000000.10101000.00000000.10000000

La entrada quedaría como:

192.168.0.128/25

Ejemplo encaminamiento

PROBLEMA

- La topología implica sólo un camino de salida desde Host A → ¿realmente necesitamos 4 entradas?

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1



Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.128	/25	192.168.0.1
150.214.190.0	/30	192.168.0.1

Podemos agrupar entradas

No merece la pena agrupar direcciones muy diferentes, porque la entrada agrupada será muy genérica

Ejemplo encaminamiento

PROBLEMA

- La tabla del ejemplo NO direcciona Internet (ej. www.google.com = 172.194.34.209)
- La topología implica sólo un camino de salida desde Host A → ¿realmente necesitamos 4 entradas?

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.0	/27	-
192.168.0.160	/27	192.168.0.1
192.168.0.192	/27	192.168.0.1
150.214.190.0	/30	192.168.0.1

!!Usar la entrada por defecto!! → /0

La entrada por defecto se suele añadir para dirigir el tráfico hacia fuera de la red (hacia Internet)

Aunque en este ejemplo, se puede usar para dirigir el tráfico a las demás subredes también.

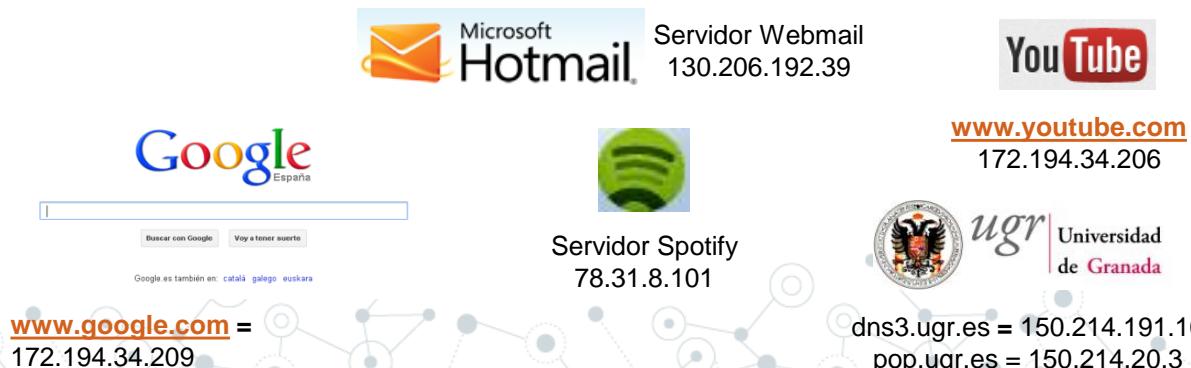
Ejemplo encaminamiento

PROBLEMA

- La tabla del ejemplo NO direcciona Internet (ej. www.google.com = 172.194.34.209)
- La topología implica sólo un camino de salida desde A → ¿realmente necesitamos 4 entradas?

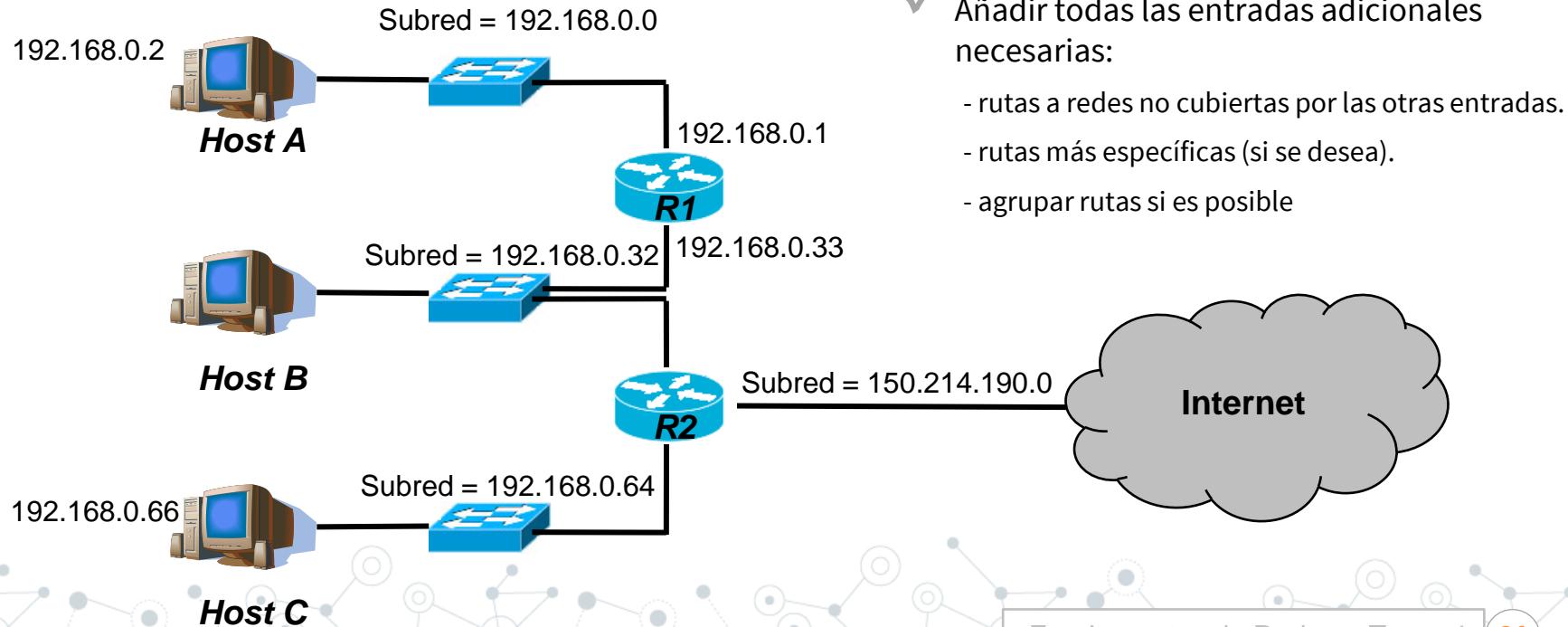


*0.0.0.0
default*



Ejercicio

- Diseñar la Tabla de encaminamiento en R2



Ejercicio

- Diseñar la Tabla de encaminamiento en R2

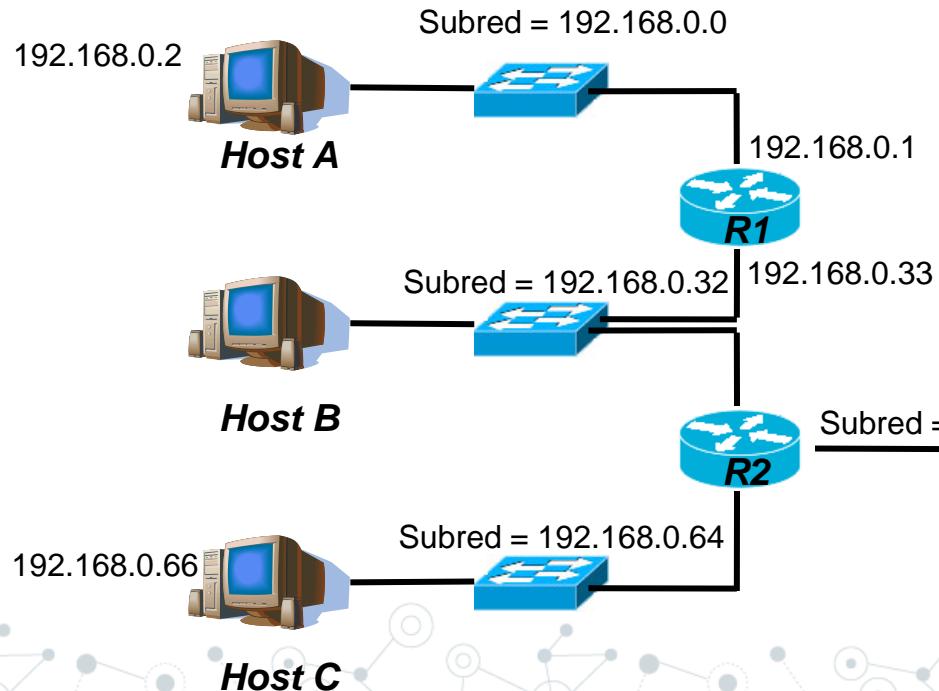
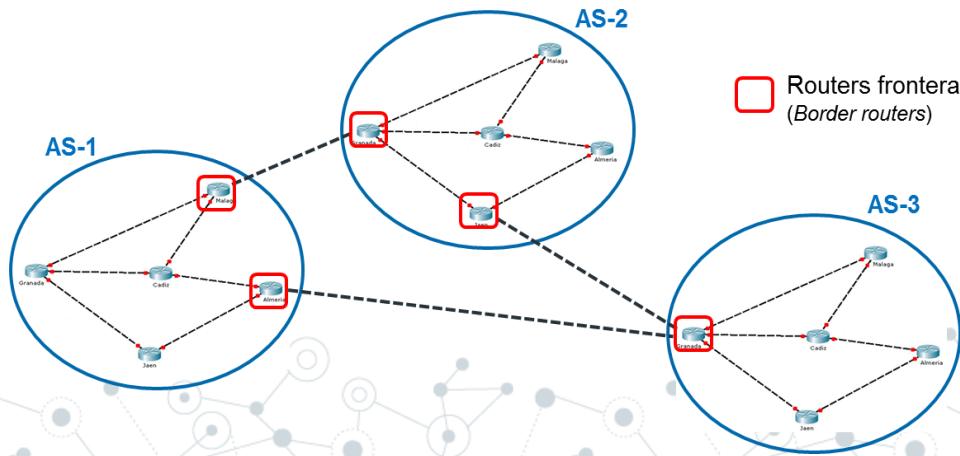


Tabla de R2

Dirección IP destino	Máscara	Siguiente nodo
192.168.0.32	/27	-
192.168.0.64	/27	-
150.214.190.0	/30	-
0.0.0.0	/0	150.214.190.2
192.168.0.0	/27	192.168.0.33

Sistemas Autónomos

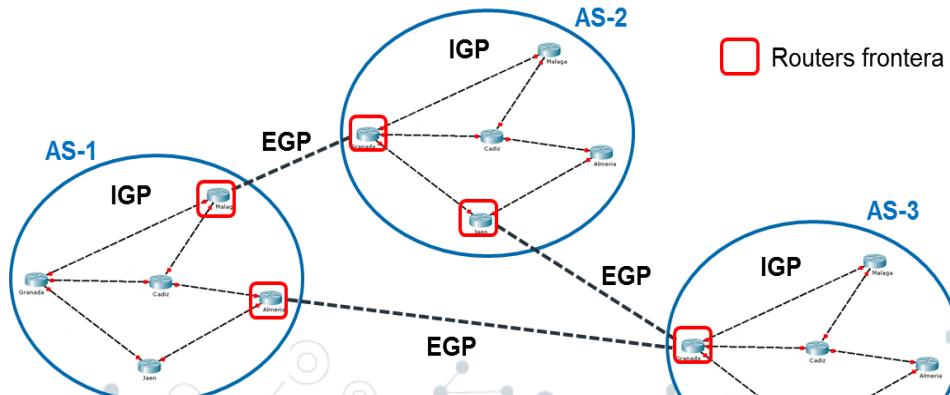
- Para **facilitar la administración y aumentar la escalabilidad** Internet se jerarquiza en **Sistemas Autónomos (SA)**.
- Un **SA es un conjunto de redes y routers** administrados por una autoridad.
- Cada **SA informa a los otros SA** de las **redes accesibles**.
Existe un router responsable de esto, denominado **router exterior** (o *router frontera*).
- Cada **SA se identifica por un entero de 16 bits** (DESDE 2007 ES 32-BITS). Ej: Rediris → AS766



Sistemas Autónomos

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.
- Ejemplo: RIP

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.
- Ejemplo: OSPF

RIP

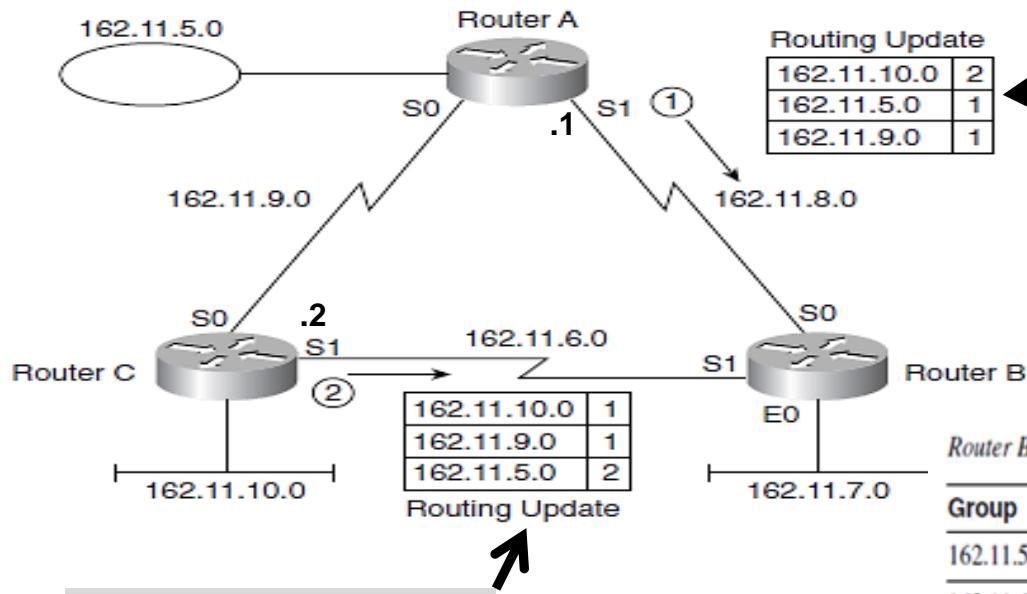
- **Routing Information Protocol** (RFC 1058, 2453, 4822).
- Protocolo de la **capa de aplicación** (opera sobre **UDP** puerto 520).
- Adopta un **algoritmo vector-distancia** (métrica basada en **número de saltos**).
- No considera la congestión de la red ni la velocidad de los enlaces.
- Una red directamente 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.

RIP

- **Routing Information Protocol** (RFC 1058, 2453, 4822).
- 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 (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.

RIP (Ejemplo)

Routers A and C Advertising to Router B



1.- Router A envía actualización a Router B indicando en cuántos saltos Router B podría alcanzar estas redes

3.- Router B compila el mejor camino posible y actualiza el coste. Además, aparecen dos rutas de igual coste (métrica) a la misma red.

2.- RouterC envía actualización a RouterB indicando en cuántos saltos RouterC puede alcanzar estas redes

Router B Routing Table

Group	Outgoing Interface	Next Router	Metric
162.11.5.0	S0	162.11.8.1 (Rout A)	2
162.11.6.0	S1		1
162.11.7.0	E0		1
162.11.8.0	S0		1
162.11.9.0	S0	162.11.8.1, 162.11.6.2	2
162.11.10.0	S1	162.11.6.2 (Rout C)	2

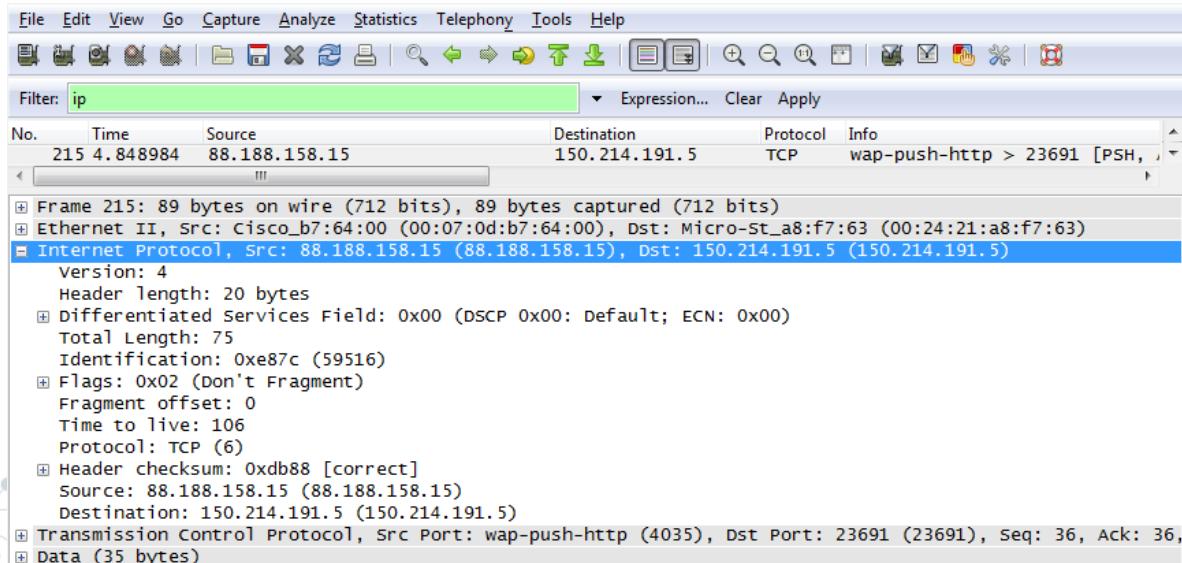
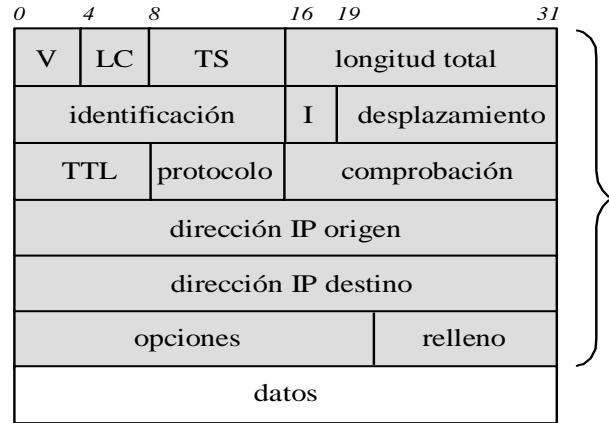
OSPF

- **Open Shortest Path First** (RFC 2328).
- 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: **coste = $10^8/BW$** .
Ej: para un enlace con BW = 1 Mbps
 $\text{coste} = 10^8/10^6 = 100$
- El **coste** de los enlaces **se podrá determinar en tiempo real** → 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 ⇔ balanceo de carga

OSPF

- 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



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).

Formato Datagrama IP

0	16	31
Versión	Tamaño Cabecera	Tipo de Servicio
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).

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	

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.

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		

Opciones:
Hasta 40 bytes.
Permite hacer
funciones de test y
depuración sobre la
red (sello de tiempo,
registro de ruta, etc).

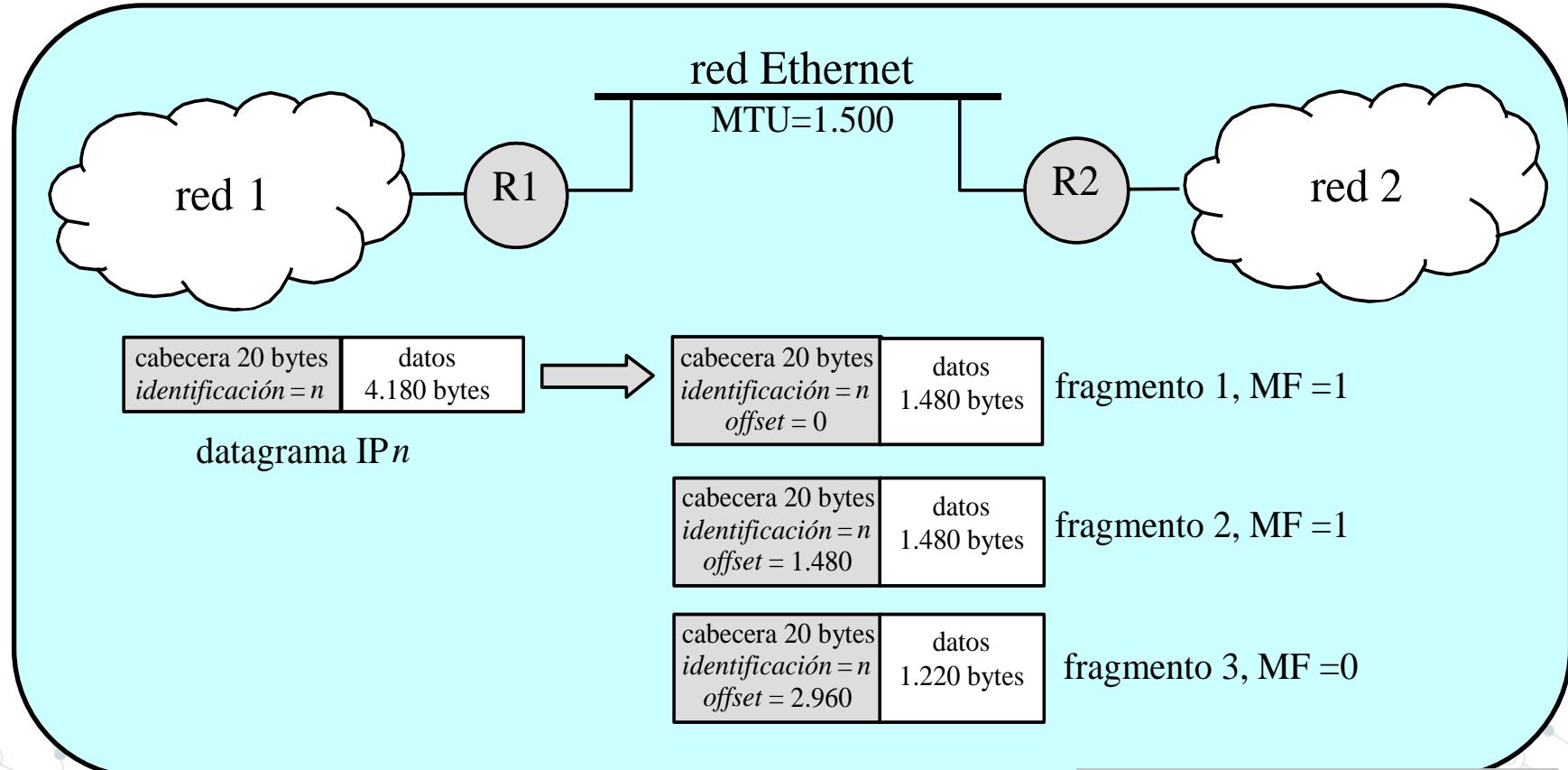
Relleno:
Bits a 0 para
completar una
palabra de 32 bits en
la cabecera.

Fragmentación IP

- Tamaño máximo: $2^{16}-1 = 65.535$ bytes.
- Adaptarse a la MTU (Maximum Transfer Unit).
- 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 (HTT 8ms)
Classical IP over ATM	9180

Fragmentación IP



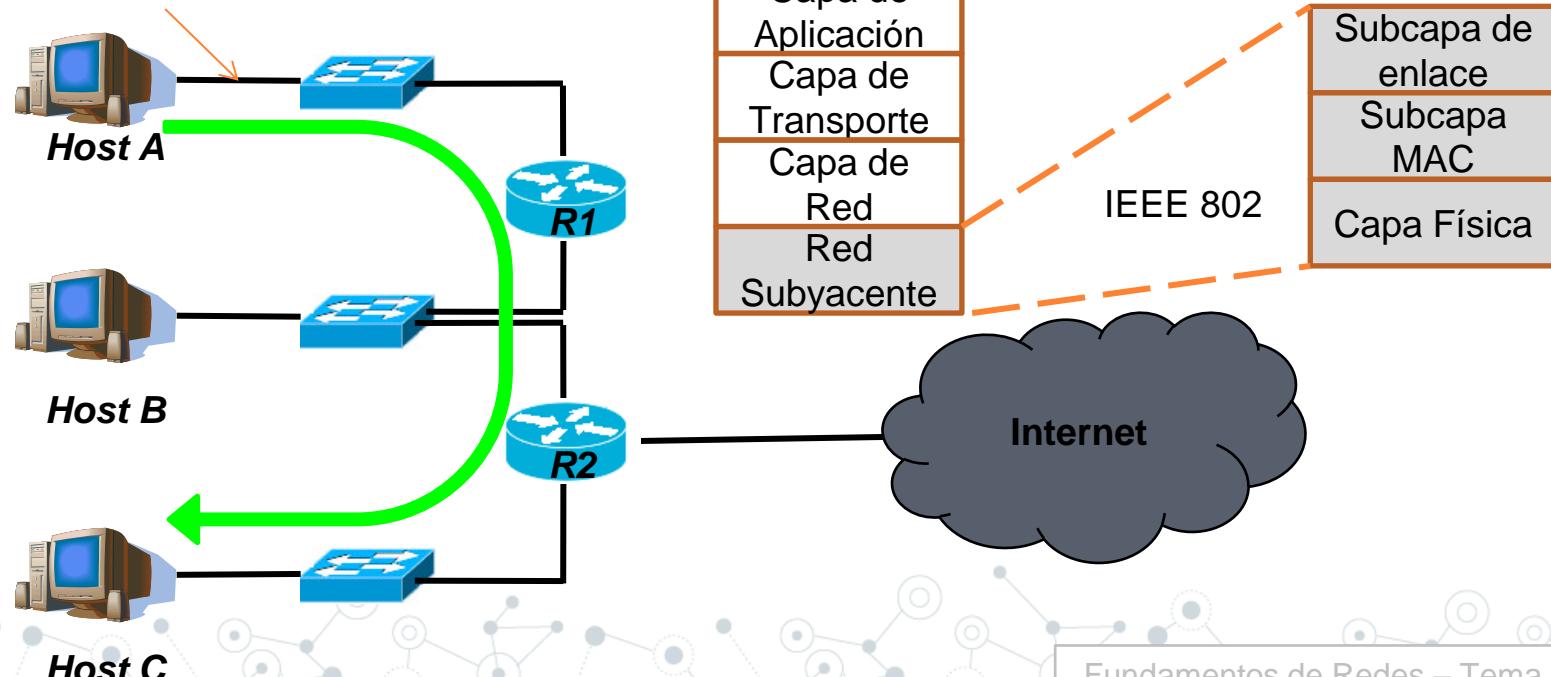
TEMA 4. Redes Conmutadas e Internet

- 4.1. Funcionalidades
- 4.2. Conmutación
- 4.3. El protocolo IP
- **4.4. Asociación con la capa de enlace: El protocolo ARP**
- 4.5. El protocolo ICMP
- 4.6. Cuestiones y ejercicios

Direcciones MAC

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

A debe conocer la MAC de R1

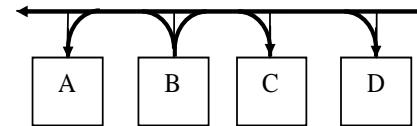


ARP

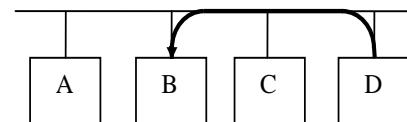
- Tras la redirección IP → Enviar a la Medium Access Control (MAC) del siguiente nodo. Se usan en redes Ethernet (cableadas) y Wifi.
- Formato (6 bytes): HH-HH-HH-HH-HH-HH Ej. 00-24-21-A8-F7-6A
- Son únicas, asignadas por IEEE en lotes de 2^{24} para cada fabricante
- Dirección de difusión (broadcast) FF-FF-FF-FF-FF-FF

ARP

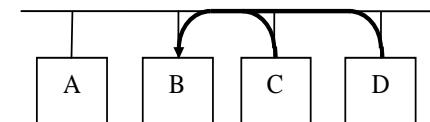
- Address Resolution Protocol
- Obtener MAC a partir de IP:
B pregunta MAC de D [(a) y (b)]



(a)



(b)



(c)

RARP

- Rerverse ARP (RARP)
- Obtener IP a partir de MAC: (a) y (c)

ARP

- Formato ARP

	0	8	16	31
Htipo		Ptipo		
Hlen	Plen	Operación		
Hemisor (bytes 0-3)				
Hemisor (bytes 4-5)		Pemisor (bytes 0-1)		
Pemisor (bytes 2-3)		Hsol (bytes 0-1)		
Hsol (bytes 2-5)				
Psol (bytes 0-3)				

Screenshot of Wireshark showing an ARP request frame.

Filter: arp

No.	Time	Source	Destination	Protocol	Info
6	0.106885	AsustekC_a2:68:bd	Broadcast	ARP	who has 150.214.191.10? Telnet?

```

Frame 6: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
Ethernet II, Src: AsustekC_a2:68:bd (90:e6:ba:a2:68:bd), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (0x0001)
  [Is gratuitous: False]
  Sender MAC address: AsustekC_a2:68:bd (90:e6:ba:a2:68:bd)
  Sender IP address: 150.214.191.178 (150.214.191.178)
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 150.214.191.10 (150.214.191.10)

```

TEMA 4. Redes Conmutadas e Internet

- 4.1. Funcionalidades
- 4.2. Conmutación
- 4.3. El protocolo IP
- 4.4. Asociación con la capa de enlace: El protocolo ARP
- **4.5. El protocolo ICMP**
- 4.6. Cuestiones y ejercicios

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
- Cabecera de 32 bits
 - Tipo (8 bits): tipo de mensaje
 - Código (8 bits): subtipo de mensaje
 - Comprobación (16 bits)



Mensaje ICMP

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

ICMP

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: icmp Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
2	0.000719	150.214.20.130	150.214.191.5	ICMP	Destination unreachable (Port)

Frame 2: 120 bytes on wire (960 bits), 120 bytes captured (960 bits)
Ethernet II, Src: Cisco_b7:64:00 (00:07:0d:b7:64:00), Dst: Micro-st_a8:f7:63 (00:24:21:a8:f7:63)
Internet Protocol, Src: 150.214.20.130 (150.214.20.130), Dst: 150.214.191.5 (150.214.191.5)
Internet Control Message Protocol
Type: 3 (Destination unreachable)
Code: 3 (Port unreachable)
Checksum: 0xfe7c [correct]
Internet Protocol, Src: 150.214.191.5 (150.214.191.5), Dst: 150.214.20.130 (150.214.20.130)
User Datagram Protocol, Src Port: netbios-ns (137), Dst Port: netbios-ns (137)
NetBIOS Name Service

¿Preguntas?

O comentarios, sugerencias, inquietudes