Las capas superiores del modelo OSI: sesión, presentación y aplicación

PID_00218446

Ramon Musach Pi

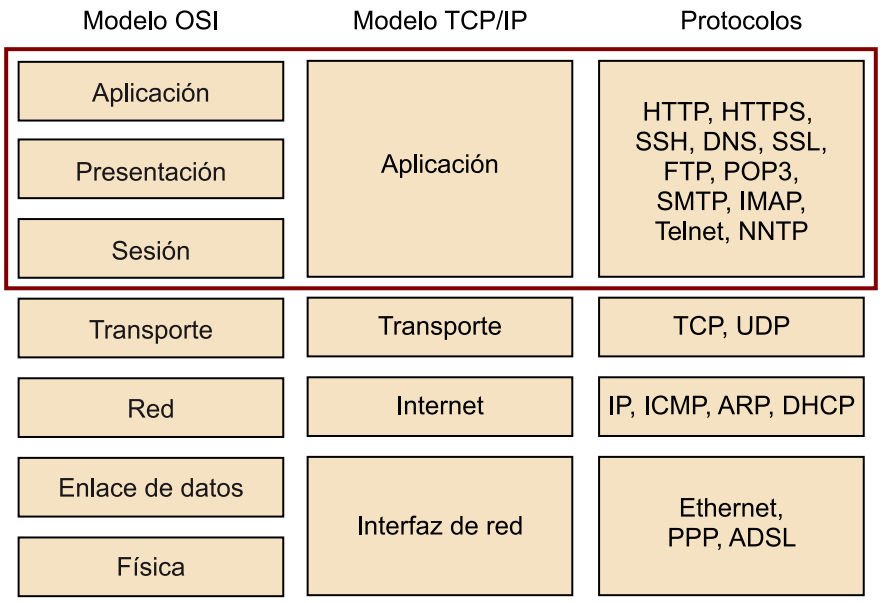
Universitat Oberta de Catalunya

Índice

Introducción

Las capas del modelo TCP/IP no se corresponden exactamente con las capas del modelo OSI. La **capa de aplicación de TCP/IP** es el equivalente a las **capas de sesión, presentación y aplicación** del modelo OSI.

Figura 1



En esta tabla disponemos de las equivalencias entre los dos modelos (OSI y TCP/IP) y los protocolos asociados a cada nivel.

Antes de entrar en detalle en la capa de aplicación, describiremos las capas de sesión y presentación del modelo OSI, con funcionalidades que en el modelo TCP/IP ya vienen integradas dentro de la capa de aplicación.

El nivel de sesión en el modelo OSI

Esta capa del modelo OSI responde a peticiones de servicio de la capa de presentación y obtiene los servicios de la capa de transporte. Organiza las funciones que permiten que dos usuarios se comuniquen mediante la red. Dentro de estas funciones se incluyen tareas de seguridad, contraseñas de usuarios y administración del sistema.

En este nivel se realiza **el establecimiento, la gestión (o utilización) y la finalización (o liberación) de las sesiones de comunicación entre entidades del mismo nivel**. La gestión incluye la sincronización del flujo de datos y el mantenimiento de las sesiones establecidas.

Para la sincronización del intercambio de datos utiliza unos puntos de verificación, denominados *check points*, para que ante una interrupción de la transmisión por cualquier causa esta se pueda retomar desde el último punto de verificación en lugar de repetirla desde el principio.

Ejemplos de protocolos de nivel de sesión son*:* ***session control protocol*** (SCP) y ***remote procedure call*** (RPC).

El nivel de presentación en el modelo OSI

El nivel de presentación responde a peticiones de servicio de la capa de aplicación y obtiene los servicios de la capa de sesión.

La capa de presentación se encarga de traducir la información del formato del computador a un formato comprensible para los usuarios. Incluye el control de las impresoras, la emulación de terminal y los sistemas de codificación. Por lo tanto, podemos afirmar que la capa de presentación se encarga de los aspectos relacionados con **la sintaxis y semántica de la información** que se transmite, realizando las conversiones de representación necesarias para la correcta interpretación de las estructuras de datos y encargándose del significado de la información transportada.

Por lo tanto, incorpora todo un conjunto de **funciones de conversión, compresión y cifrado (o codificación) de los datos del nivel de aplicación**.

Por ejemplo, la compresión de los datos se utiliza para reducir el número de bits que se han de transmitir, y el cifrado, empleando técnicas criptográficas como las que veremos en el próximo módulo, se usa para asegurar la privacidad y su autenticación.

Las implementaciones de la capa de presentación no suelen asociarse a una pila de protocolos, sino que utilizan estándares de formato de datos que sean apropiados para la aplicación, como por ejemplo los formatos de vídeo MPEG, Quicktime, etc.; o los de imágenes, como GIF, JPEG, TIFF, etc.

Tal como hemos comentado, las funcionalidades de las capas de sesión y presentación en el modelo TCP/IP están integradas en la capa de aplicación, y por lo tanto en muchas aplicaciones y protocolos no se hace ninguna distinción entre las capas de presentación y aplicación. Un ejemplo de ello lo tenemos con HTTP, generalmente considerado como protocolo de capa de aplicación, y que también implementa funcionalidades propias de la capa de presentación.

1. El nivel de aplicación

A partir de este apartado trataremos el nivel de aplicación, tal y como lo hace el modelo TCP/IP, integrando funcionalidades de las capas de sesión y de presentación del modelo OSI.

Los servicios de la capa de aplicación facilitan la comunicación entre las aplicaciones de software (programas) que corren sobre esta capa y los servicios que ofrecen las capas inferiores. Así, los protocolos del nivel de la capa de transporte ofrecen servicios a la aplicación, y al mismo tiempo la aplicación requiere unas capacidades que esta capa inferior le debe ofrecer. Entre estas capacidades nos encontramos con **transferencia fiable**, para no perder información por ejemplo en transferencias de ficheros; **ancho de banda**, por ejemplo en transmisiones de imágenes en tiempo real, y **temporización**, para comunicaciones en tiempo real.

Al tratar este nivel, procederemos a describir toda una serie de aplicaciones, denominadas distribuidas, y protocolos de comunicación asociados.

Una **aplicación distribuida** está formada por una colección de ordenadores autónomos enlazados por una red de ordenadores y soportados por un software que permite que los ordenadores autónomos actúen como un servicio integrado.

Las dos arquitecturas distribuidas más empleadas actualmente son **cliente-servidor** (*client/server* en inglés) y de **igual a igual** (*peer-to-peer* o P2P).

1.1. Modelo cliente-servidor

La mayoría de las aplicaciones de software que funcionan en un entorno en red siguen un modelo cliente-servidor.

En este modelo existen dos tipos de componentes que permiten comunicarse entre sí: clientes y servidores.

Los **clientes** son los que hacen las peticiones de servicio, iniciando a menudo la comunicación con el servidor.

Así, podemos hablar de un programa cliente, iniciado por un usuario o por otro programa que se está ejecutando en un *host* y que solicita un servicio determinado a otro *host* de la red (habitualmente un *host* remoto). El proceso finaliza cuando este programa recibe el servicio solicitado.

Los **servidores** son los *hosts* que proveen servicios. A menudo son los que reciben las peticiones que realizan los clientes, las resuelven y devuelven las respuestas a los clientes.

Uno programa servidor es el que se está ejecutando en un *host*, a menudo remoto, que proporciona determinados servicios a múltiples programas cliente. Cuando el programa servidor se inicia, empieza a ofrecer sus servicios, de manera ininterrumpida y continuada, a aquellos clientes que lo soliciten.

Por ejemplo, cuando un usuario utiliza un navegador (aplicación de software del cliente) para abrir una página web, el protocolo denominado HTTP es el que da forma a la solicitud y la envía desde el cliente hasta el servidor. Es el mismo protocolo también el que dará formato y enviará la respuesta del servidor web al navegador del cliente.

1.2. Modelo de igual a igual (*peer-to-peer*, P2P)

Un sistema de igual a igual se caracteriza por ser un sistema distribuido en el que todos los nodos tienen las mismas capacidades y, por lo tanto, en el que la comunicación es simétrica.

Las redes *peer-to-peer* (redes punto a punto o más conocidas como redes P2P) son aquellas redes que no contienen nodos clientes y servidores fijos, sino un número de nodos "iguales" (llamados *peers*, 'pares') que funcionan a la vez como clientes y servidores y otros nodos de la red. Estos sistemas ofrecen y utilizan una serie de recursos distribuidos para llevar a cabo determinadas funciones de manera descentralizada. Estos recursos pueden ser muy variados, como datos, ancho de banda o capacidad de cálculo.

Desde sus inicios, los sistemas y aplicaciones de igual a igual se han ido popularizando en internet con aplicaciones relacionadas con la compartición de ficheros, pero también hay otros muy populares como Skype, que proporciona videollamadas por la red internet, sistemas de mensajería instantánea, sistemas de procesamiento distribuido, juegos, etc.

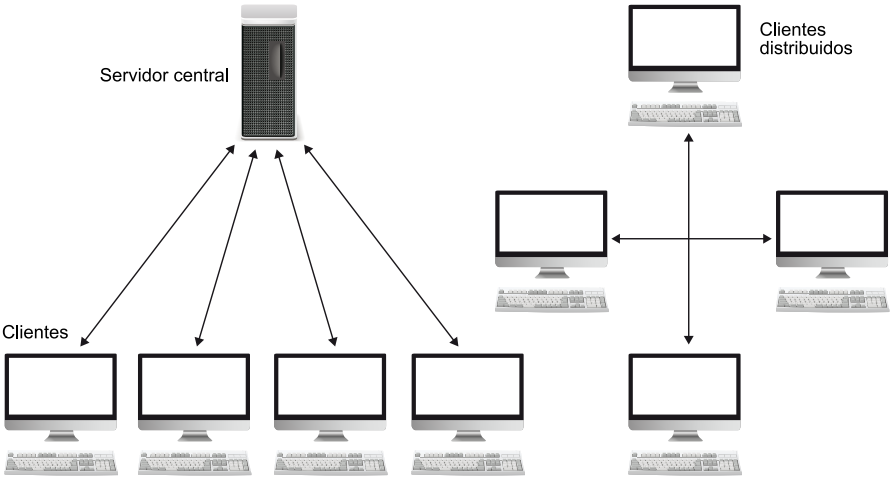
Las aplicaciones de igual a igual se empezaron a popularizar con las aplicaciones de compartición de ficheros. Más en concreto, Napster se popularizó sobre el año 2000. Creado por Shawn Fanning, fue pionero de las redes P2P de intercambio con un servicio de distribución de archivos de música en formato MP3.

Los nodos que forman un sistema o aplicación de igual a igual se organizan con lo que se conoce como **red superpuesta** (*overlay network*, en inglés), que funciona sobre la red física que conecta los nodos.

Ejemplo

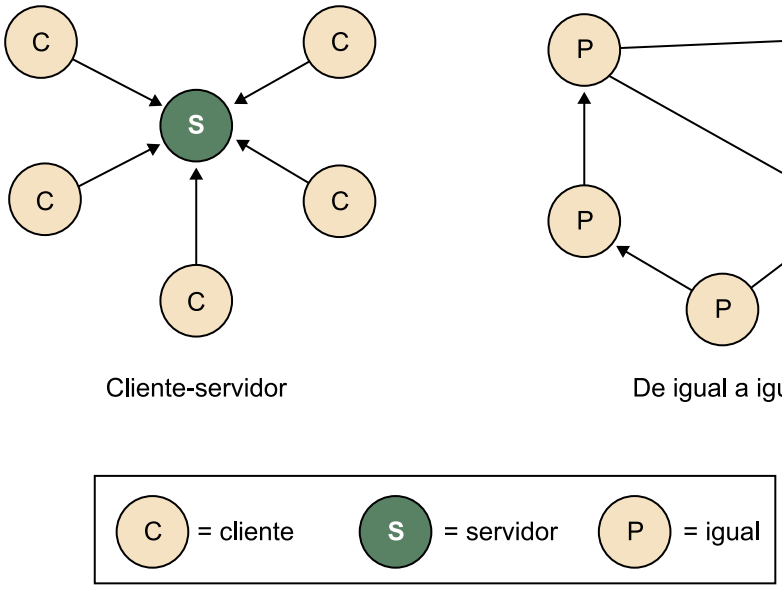
Algunos ejemplos de sistemas P2P son Gnutella, FastTrack/KaZaA, BitTorrent, Overnet/eDonkey2000.

Figura 2



El primer esquema corresponde a un modelo cliente-servidor y el segundo a un sistema distribuido.

Figura 3



De manera más esquematizada, el primer esquema corresponde a un modelo cliente-servidor y el segundo a un sistema *peer-to-peer*.

Los dos modelos descritos son, de hecho, dos maneras de llegar a plantear el diseño de una aplicación, pero debemos tener en cuenta que a menudo las aplicaciones están planteadas como modelos híbridos entre diferentes modelos para satisfacer las necesidades de los usuarios de estas aplicaciones.

2. Protocolos de la capa de aplicación

En este apartado describiremos toda una serie de protocolos de comunicaciones, asociados directamente a aplicaciones que utilizan internet como medio de comunicación. Estas aplicaciones se conocen como aplicaciones distribuidas, dado que están formadas por diferentes partes y cada una se encuentra en máquinas diferentes.

2.1. DNS: servicio de nombres en internet

2.1.1. ¿Qué es?

Cada uno de los ordenadores que se conectan a internet lo hace con una dirección IP única. Considerando la dificultad que supone recordar estas direcciones, se creó un **sistema de nombres de dominio**, en inglés *domain name system* (DNS), que permite traducir direcciones IP con nombres que se pueden recordar más fácilmente.

Un servicio DNS recibe las peticiones que le llegan y realiza rápidamente esta traducción. Por ejemplo, cuando escribimos una dirección web en el navegador, este realiza la consulta al DNS para conocer la dirección IP que le corresponde.

Un **dominio** es un grupo de nodos que pertenecen a una misma organización y que tienen en común una parte de su dirección IP. Un dominio está identificado por un nombre de dominio, que habitualmente está asociado a una organización. El nombre completo de un *host* está formado por el nombre del *host* más el nombre del dominio al que pertenece.

Un nombre de dominio se representa mediante una serie de etiquetas separadas por puntos. Cada etiqueta representa un nivel diferente en la jerarquía de nombres de dominio.

Ejemplo

Por ejemplo, en el nombre de dominio www.uoc.edu, "edu" es el **dominio de nivel superior** (*top-level domain*, TLD), "uoc" es el dominio de segundo nivel, y "www" es el dominio de tercer nivel.

Así pues, cuando escribimos www.uoc.edu, el servicio DNS le dirá al navegador que la dirección IP de www.uoc.edu es 213.73.40.242. Por lo tanto, el DNS vendría ser como un listado telefónico, pero en vez de relaciones de nombres de personas y teléfonos tiene relaciones de dominios y direcciones IP.

Nota

Los nombres de dominio deben estar registrados por la *Internet Corporation for Assigned Names and Numbers* (**ICANN**, https://www.icann.org) o, en su defecto, por una empresa

autorizada para ello por esta entidad. La ICANN se creó en 1998 con el objetivo de encargarse de ciertas tareas que hasta entonces estaban en manos de la IANA, entre las que se encuentra la aprobación y el control de los dominios de internet.

A continuación mostramos algunos de los dominios de nivel superior aprobados por la ICANN:

1) De ámbito genérico:

.cat - para la lengua y cultura catalanas

.com - organizaciones comerciales

.net - estructuras de la red internet

.org - organizaciones de otro tipo (a menudo sin ánimo de lucro o religiosas)

.edu - educación

.info - agencias de información

.int - organizaciones internacionales (i. e., ONU)

.biz - negocios

.mil – militar

2) De ámbito territorial:

.ad - Andorra

.au - Australia

.de - Alemania

.es - España

.fr - Francia

.it - Italia

.jp - Japón

.lu - Luxemburgo

.nl - Países Bajos

.tr - Turquía

Para más información sobre los nombres de dominios y su registro se puede consultar la siguiente guía publicada por ICANN:

"Guía para principiantes para NOMBRES DE DOMINIO"

https://www.icann.org/en/system/files/files/domain-names-beginners-guide-06dec10-es.pdf

Otros servicios importantes que proporciona el DNS son:

- **Traducción de alias** (nombres adicionales que pueden llegar a tener los *hosts*; por ejemplo: www.uoc.edu es un alias de www-org.uoc.edu). Al nombre original se le denomina canónico.

- **Traducción de alias del servidor de correo**. Tengamos en cuenta que una misma organización puede llegar a tener diferentes nombres (alias) para su dominio de correo electrónico y para el servidor web.

- **Distribución de carga**. Redirigiendo, si es el caso, el tráfico a servidores web que disponen de información o servicios replicados, para mejorar el acceso a ellos. Tengamos en cuenta que sitios webs con muchos accesos pueden estar replicados en muchos servidores, cada uno corriendo en un ordenador diferente y con una dirección IP distinta.

2.1.2. Tipo y ubicación de los servidores DNS

Sería sencillo que la red internet dispusiera de un único servidor para llevar a cabo las equivalencias entre las direcciones IP y los nombres de dominio. Pero, tal y como podemos intuir, no sería efectivo por el colapso que se podría producir en las peticiones y respuestas, además del gran volumen que debería tener la base de datos de este servidor.

El DNS es, en realidad, una base de datos de servidores, organizados de manera jerárquica y distribuidos por todo el mundo. Por lo tanto, ningún servidor DNS dispone de todas las equivalencias entre nombres y direcciones IP.

Existen tres tipos de servidores DNS:

1) Servidores DNS raíz. Hay pocos servidores DNS raíz. Cada uno de ellos es en realidad un clúster de servidores reproducidos, por seguridad y fiabilidad.

Podemos encontrar la ubicación de los servidores raíz en: http://www.root-servers.org.

2) Servidores DNS de nivel de dominio superior (*top-level domain*, TLD). Son los responsables de los dominios de primer nivel, como .org, .com, .net, .edu, .us, .cat, .es, etc.

3) Servidores DNS autorizados. Cada organización que disponga de ordenadores accesibles a internet debe disponer de un registro público que permita hacer la traducción de nombres y direcciones IP. Este listado se encuentra en lo que se denomina un DNS autorizado, que puede ser un servidor provisto por la propia organización o por algún proveedor de servicios.

4) Servidores DNS locales. A pesar de que estrictamente no pertenecen a la jerarquía de servidores de DNS, son fundamentales, dado que cuando un nodo se conecta a su ISP (proveedor de acceso a internet), este le proporciona la dirección IP de uno o más servidores DNS locales. Este actúa como **proxy** (o servidor intermediario) y lo envía a la jerarquía de servidores DNS para que resuelva la petición.

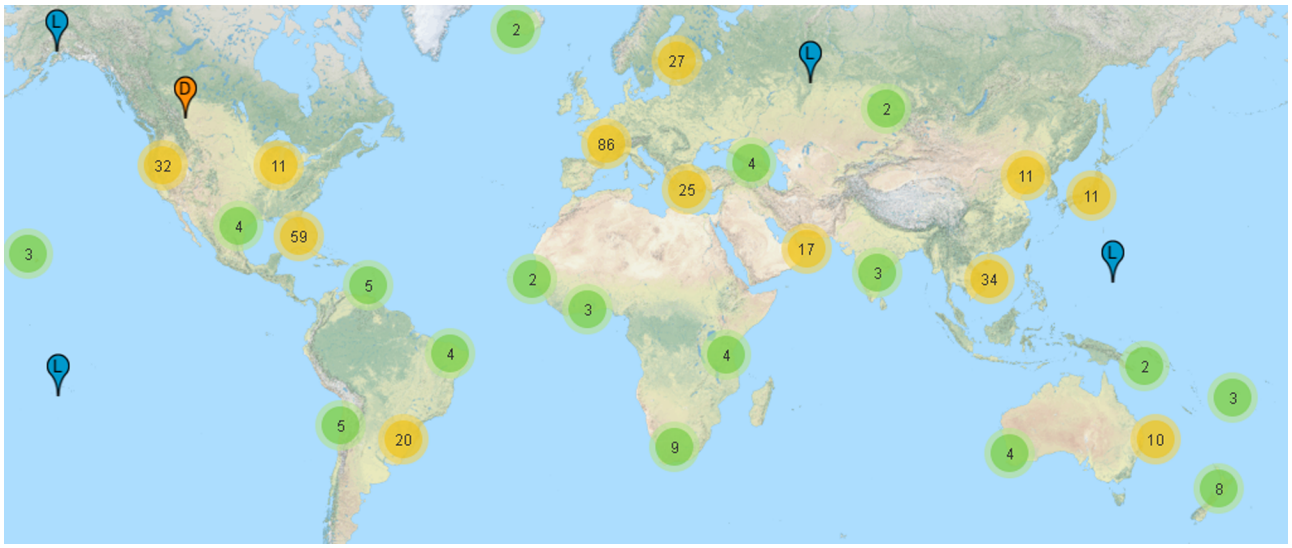
Proxy

Un proxy o servidor intermediario es un programa o dispositivo que realiza una acción en representación de otro. Actúa de intermediario entre el cliente y el servidor, filtrando el tráfico en función de las políticas que se establezcan. Puede responder a peticiones del

cliente siempre que disponga de la respuesta en su memoria caché, así se ahorra solicitarla al servidor.

Podemos encontrar los DNS de las principales operadoras en: http://www.adslayuda.com/dns.html

Figura 4



Mapa del mundo con las ubicaciones de los servidores de DNS. Extraído de http://www.root-servers.org/.

2.1.3. Funcionamiento del servicio DNS

La traducción de los nombres la realizan los denominados servidores DNS, a petición de las aplicaciones del cliente (navegadores, clientes de correo u otras aplicaciones), y de manera transparente al usuario.

A menudo los usuarios utilizan como servidor DNS el que proporciona su proveedor de servicios de internet. La dirección de estos servidores puede llegar a ser configurada de manera manual o automática mediante **DHCP** (***dynamic host configuration protocol***); en otros casos los administradores de red pueden tener configurados sus propios servidores DNS.

Considerando que la equivalencia entre *hosts* y direcciones IP no es permanente, los servidores DNS descartan la información caché después de un cierto tiempo (que puede ser de uno dos días).

El protocolo DNS generalmente transporta las peticiones y respuestas por un puerto UDP, dado que por estructura y nivel de seguridad implementado es más rápido. Pero hay casos en los que se utiliza el puerto TCP, sobre todo cuando hay que transportar respuestas más grandes de 512 bytes de longitud y por razones de necesidad de fiabilidad.

En cuanto a aspectos de seguridad relacionados con los DNS, hay que comentar que cuando se implementó no se consideraron estos aspectos de seguridad. Una vulnerabilidad que se puede dar es la de hacer creer a un servidor DNS una

determinada traducción incorrecta. Es lo que se denomina contaminación de la memoria caché del DNS, que consiste en proporcionar, de forma maliciosa o intencionada, datos no originados por un servidor DNS autorizado.

El servicio ***domain name system security extensions*** (DNSSEC) modifica el DNS para añadir respuestas firmadas digitalmente, de manera que así se autentica el origen de los datos del DNS. A partir del 10 de julio del 2010, todos los **servidores raíces** deben emplear este protocolo DNSSEC: http://www.root-servers.org/

Servidor raíz

Un **servidor raíz** es un servidor de nombres de dominio (DNS) que sabe dónde están los servidores de dominio para cada una de las zonas de mayor nivel en internet. Los servidores raíz son fundamentales para el funcionamiento del DNS, dado que son los que conocen todos los dominios de primer nivel, por lo tanto son el primer paso en la traducción de los nombres de *hosts* a direcciones IP. Admiten un gran volumen de consultas, y en el 2006 había trece repartidos por todo el mundo, con réplicas de estos en varios continentes.

2.2. La web y el HTTP

El servicio web o WWW (*World Wide Web*, red informática mundial) es el que da acceso a la información multimedia, por lo tanto con contenidos de diferentes tipos: texto, imágenes, audio, vídeo, etc. También incluye referencias a otros elementos de información según el modelo de los sistemas hipertexto. En la terminología WWW, a todos estos elementos se los denomina **recursos**.

Un **sistema hipertexto** permite recorrer un documento de manera no necesariamente lineal o secuencial, sino siguiendo las referencias o enlaces que el usuario seleccione, saltando a la parte referenciada. Popularmente, este modo de acceder a la información se conoce con el nombre de *navegar*. Cuando esta navegación, además de hacerse a partir del texto, se realiza mediante elementos multimedia y no solo texto, tales sistemas son denominados **sistemas hipermedia**.

El servicio web es un sistema basado en el modelo cliente/servidor, El servidor almacena la información multimedia y el cliente la solicita, mediante un navegador, y la presenta al usuario.

El protocolo que se utiliza para el diálogo entre el cliente y el servidor es **HTTP** (***hypertext transfer protocol***, protocolo de transferencia de hipertexto), que describiremos en un próximo apartado.

Considerando que el ordenador cliente, mediante el navegador, debe mostrar la información tal y como se desea, es necesario que las especificaciones de las características de esta visualización queden muy claras, y es aquí donde intervienen lo que se denominan los lenguajes de marcas (como HTML y XHTML).

El método general utilizado en el servicio WWW para identificar la información a la que se quiere acceder se conoce con el nombre de **identificadores uniformes de recursos** (URl). La definición del estándar URl se produce en 1998 con la publicación RFC 2396.

Desde un documento se pueden referenciar recursos especificando las direcciones, que se representan mediante la siguiente terminología: esquema: identificador.

Donde:

- **Esquema** puede ser http, ftp, mailto, etc., extendiendo así la funcionalidad del servicio web con el acceso a servidores ftp o servidores de correo desde un cliente web.

- **Identificador** contiene el nombre del recurso y el servidor donde se encuentra.

Cuando el esquema es http o ftp, el servidor empieza con los caracteres "//", y el servidor y el nombre del recurso se separa con un carácter "/".

Ejemplo

http://www.uoc.es/index.html

2.2.1. Los lenguajes de marcas HTML y XHTML

El lenguaje de marcas es una manera de codificar un documento empleando etiquetas (o marcas) que contienen información adicional sobre la estructura, la presentación, etc. Estos lenguajes están diseñados para especificar documentos hipermedia.

Uno de los lenguajes de marcas más empleados es el lenguaje **HTML** (***hypertext markup language***), que se encuentra dentro de la familia de lenguajes (X)HTML. Los fundamentos del lenguaje HTML los estableció Tim Berners-Lee, en 1992, desde el **CERN** (***European Organization for Nuclear Research***). Tim Berners-Lee creó el HTML siguiendo las normas del lenguaje **SGML** (*standard generalized markup language*).

SGML (*standard generalized markup language*) son un conjunto de normas que se publicaron en 1986 con el objetivo de establecer la sintaxis de un documento. Se basa en un sistema de etiquetas que permite organizar la información del documento.

A pesar de que inicialmente no prosperó como estándar, sí que a partir de 1996 la IETF cierra el desarrollo del estándar HTML, y a la vez es adoptado por el W3C (*World Wide Web Consortium*).

Nota

Uno de los puntos de inflexión de la *World Wide Web* es la introducción del navegador web Mosaic, en 1993; un navegador gráfico desarrollado por un equipo del **NCSA** (*National Center for Supercomputing Applications*, Centro Nacional de Aplicaciones de Supercomputación) en la Universidad de Illinois.

Nota

El **W3C** (***World Wide Web Consortium***) y la **IETF** (***Internet Engineering Task Force***) se encargan de definir los estándares web, es decir, las normativas relativas a aspectos de la Word Wide Web.

A partir de la versión HTML 4.01 se genera otro lenguaje, el denominado **XHTML** (***extensible hypertext markup language***); un lenguaje de marcas muy parecido al HTML pero que en lugar de seguir las reglas sintácticas de la SGML sigue las de otro lenguaje de marcas, el **XML** (***extensible markup language***).

En cuanto al presente del mundo web, debemos hablar del **HTML5** (o su variante XML, el XHTML5). Corresponde a la quinta gran revisión del lenguaje básico de la World Wide Web, el HTML. En esta versión se da la circunstancia de que, por primera vez, HTML y XHTML se han desarrollado en paralelo bajo la regulación del Consorcio **W3C**. Incorpora muchas funcionalidades que en los inicios no se tuvieron en cuenta, como soporte de contenidos multimedia (audio, vídeo) con etiquetas que contienen códecs para poder mostrar estos contenidos, soporte para grandes conjuntos de datos, mejoras en formularios, nuevos visores, posibilidad de arrastrar objetos, como puedan ser imágenes, etc.

Desde esta página, http://html5test.com/, podemos llegar a comprobar hasta qué punto los navegadores más populares actualmente, como Google Chrome, Mozilla Firefox, Internet Explorer, etc., están implementando las nuevas funcionalidades de HTML5.

En cuanto a otro concepto relacionado con los documentos hipermedia, tenemos el CSS, que se utiliza para dar estilo a documentos XHTML, y de esta manera poder separar el contenido de la presentación.

El **CSS** (***cascading style sheets***) es un lenguaje de hojas de estilo que permite describir el aspecto y el formato que tendrá un documento escrito en un lenguaje de marcas cuando se muestre por pantalla o cuando se imprima, o incluso cómo se pronunciará la información presente en el documento mediante un dispositivo de lectura. A pesar de que se aplica a HTML y XHTML, también puede ser aplicado a cualquier tipo de documento XML. Sus especificaciones también son mantenidas por el W3C.

2.2.2. HTTP (*hypertext transfer protocol*)

HTTP es un protocolo del nivel de aplicación para sistemas hipermedia colaborativos y distribuidos, que es la base de la web. Se encuentra definido en RFC 1945 y RFC 2616.

Este protocolo se encarga de gestionar la mayor parte del tráfico que circula por internet, dado que está asociada a la solicitud de recursos web.

Sigue el modelo general de peticiones y respuestas entre un cliente y un servidor, basándose en un servicio de transporte fiable. HTTP utiliza primordialmente el protocolo TCP, y por defecto el puerto 80.

Cuando en un navegador se escribe la dirección http://www.uoc.edu se hace una llamada al servicio DNS para que asocie el nombre de dominio con una dirección IP. Cuando ya se conoce, se envía una solicitud ***get*** al servidor web, que responde con una respuesta ***send*** (*get* y *send* son dos operaciones del protocolo HTTP). Dentro de una misma sesión se va produciendo este diálogo.

2.2.3. HTTPS (*hypertext transfer protocol secure*)

Para llevar a cabo transacciones de datos seguros mediante la web, se utiliza el protocolo **HTTPS** (***hypertext transfer protocol secure***). En este protocolo se utiliza una tecnología basada en certificados digitales con el fin de garantizar la autenticación entre los extremos de la transacción. Además, HTTPS garantiza la confidencialidad de los datos, ya que cifra todos los paquetes de datos enviados durante la sesión. Para poder emplear HTTPS, el servidor web debe adquirir un certificado digital a un proveedor de este tipo de servicios. También utiliza TCP, pero con el puerto 443.

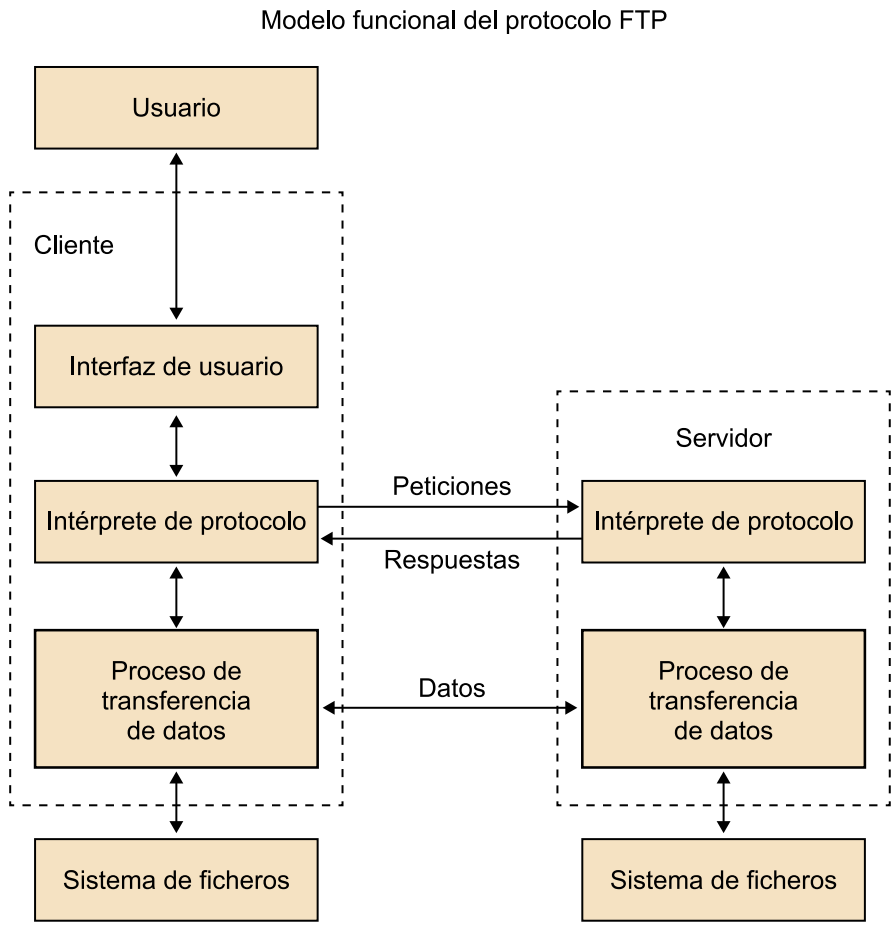
2.3. FTP (*file transfer protocol*)

Esta fue una de las primeras aplicaciones desarrolladas para el entorno en internet. La especificación oficial de este protocolo, FTP (*file transfer protocol*, protocolo de transferencia de ficheros), se publicó en 1985 en el documento RFC 959.

Este protocolo se basa en el modelo cliente/servidor y **permite la transferencia de ficheros en los dos sentidos, con funcionalidades añadidas como las de manipular el sistema de ficheros del servidor: modificarlos, borrarlos, crear y borrar directorios, listar contenidos, etc**. Actúa de manera transparente al usuario, permitiendo la interoperabilidad entre sistemas de ficheros muy diferentes.

En el modelo FTP existen dos entidades, tanto en el servidor como en el cliente, que intervienen en la transferencia de ficheros: **el intérprete de protocolo y el proceso de transferencia de datos**. El primero se encarga del intercambio de pedidos de protocolo, y el segundo, bajo el control del primero, se encarga de intercambiar los datos, es decir, los contenidos de los ficheros que se deben transmitir.

Figura 5



Modelo funcional del protocolo FTP.

El FTP se basa en conexiones TCP y tiene asignados los puertos 21 y 20. De hecho, el puerto 21 se utiliza para el intercambio de órdenes y respuestas, y el 20 para la transferencia de datos. El intérprete de protocolo del servidor ha de estar preparado para recibir peticiones de conexión en un puerto TCP que, como hemos dicho, es el número 21 por defecto.

Las peticiones FTP son cadenas de caracteres ASCII que contienen la petición, parámetros opcionales dependiendo de la petición y un carácter de "final de línea" (carácter <CRLF>).

Cuando se accede a un servidor FTP, en la mayoría de los casos, hay que estar registrado previamente para que acepte las peticiones. Si el servidor permite conexiones anónimas, el acceso será más restringido y solo se permitirán unas funcionalidades concretas, como por ejemplo el acceso a determinados ficheros. Aun así, se debe tener cuidado al abrir el servidor a accesos anónimos, dado que un error en el programa servidor FTP puede permitir intrusiones no deseadas en el sistema.

2.3.1. TFTP (*trivial file transfer protocol*)

Por su complejidad, el protocolo FTP puede no ser el más apropiado para algunas situaciones concretas en las que hay que transferir ficheros de un ordenador a otro. Un ejemplo de esto puede ser el de una estación de trabajo sin disco que carga el sistema operativo por medio de la red, desde un ordenador que actúa como "servidor de arranque" de la estación, proporcionándole los ficheros que necesita. Una pequeña aplicación de la memoria ROM de la estación controla esta transferencia de ficheros.

En casos como este que acabamos de mencionar, de transmisiones simples, se ha definido el TFTP, *trivial file transfer protocol*, especificado en el estándar RFC 1350. Este protocolo se basa en datagramas, no requiere implementar el protocolo TCP, dado que a menudo utiliza UDP, proporcionando solo dos operaciones (leer y escribir ficheros en el servidor), sin ningún tipo de identificación ni autenticación de usuario.

2.4. Correo electrónico en internet

El correo electrónico es la aplicación distribuida que permite el envío de mensajes electrónicos mediante sistemas informáticos. Cuando se especificó esta aplicación se tuvieron muy en cuenta los elementos y las funcionalidades existentes en el correo postal.

La funcionalidad está basada en la filosofía de almacenamiento y reenvío.

Ha habido una gran evolución desde los primeros sistemas, que podían intercambiar solo mensajes de texto ASCII, hasta los correos electrónicos con contenidos multimedia actuales.

Los protocolos asociados a esta aplicación son:

1) SMTP (***simple mail transfer protocol***), para la transferencia de mensajes. Es independiente del formato y el contenido del mensaje.

2) POP3 (***post office protocol***), para el acceso simple a buzones de correo.

3) IMAP4rev1 (***internet message access protocol***), para el acceso complejo a buzones de correo.

También fue necesario definir un formato de mensaje, el RFC 822, que posteriormente se amplió y dio lugar al formato MIME.

El **formato de los mensajes RFC 822** se basa en el formato típico de las cartas postales con información del destinatario, del remitente y el contenido del mensaje.

Este estándar especifica las partes de estos mensajes: cabecera, con toda una serie de campos, y cuerpo del mensaje, con el contenido (opcional).

Como campos obligatorios de la cabecera tenemos: fecha (Date), origen (From) y destinatario (To) o destinatario de copia ciega (Bcc).

La norma RFC 822 define un formato de mensaje y un contenido con una única parte de texto en ASCII de 7 bits. Se consideró que este formato era demasiado simple y que era necesario algún método para superar las limitaciones.

En este contexto, el formato **MIME**, ***multipurpose internet mail extensions***, (RFC 2045 a 2049) redefine el formato del mensaje para permitir, sin perder la compatibilidad con el formato definido por el RFC 822, las características siguientes:

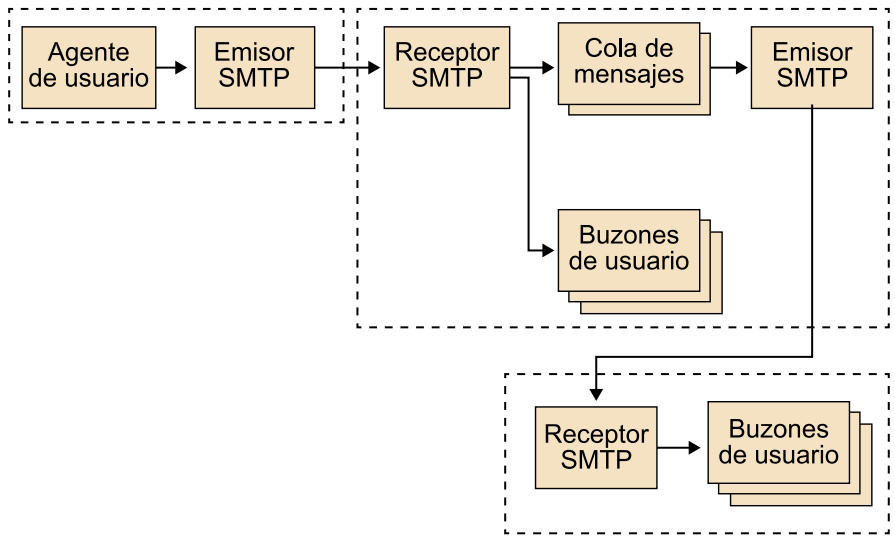
- Contenido de texto no solo ASCII de 7 bits.
- Contenido no textual.
- Contenido con múltiples partes (para permitir adjuntar ficheros).
- Cabeceras con texto no solo ASCII de 7 bits.

2.4.1. SMTP (*simple mail transfer protocol*)

Es el protocolo más utilizado en internet para transferir mensajes de correo electrónico. Ofrece una transferencia fiable y eficiente de mensajes de correo.

La figura siguiente nos muestra el modelo de un sistema SMTP:

Figura 6



Este estándar también sigue el modelo cliente/servidor, en el que entre sus especificaciones el término *usuario* se denomina *agente de usuario*, el *cliente* es el *emisor SMTP* y *servidor* equivale a *receptor SMTP*.

2.4.2. POP3 (*post office protocol* o protocolo de acceso simple a los buzones de correo)

Este protocolo (descrito en el RFC 1939) de la capa de aplicación se definió para dar respuesta a sistemas pequeños, en los que no necesariamente los sistemas clientes están siempre conectados y dispuestos a recibir mensajes en cualquier momento. Permite la recuperación de mensajes de buzones de correo remotos. A diferencia de IMAP (que veremos después), en el esquema de POP3 el almacenamiento de correo se lleva a cabo en el ordenador del usuario.

Asociado al puerto 110 en comunicaciones TCP, el POP3 no especifica ningún método para la remisión de correo; otros protocolos de transferencia de correo, como el SMTP que acabamos de ver, proporcionan esta funcionalidad.

El modelo del POP3 consta de los siguientes elementos: agente de usuario, cliente POP3 y servidor POP3. Cuando el cliente POP3 necesita acceder al buzón, se conecta con el servidor POP3, recupera la información que le interesa y cierra la conexión.

Los tres estados definidos en la norma son los siguientes:

1) Una vez que se ha abierto la conexión, la sesión entra en el estado de autorización, cuando el cliente se debe identificar ante el servidor POP3.

2) Una vez autorizado, la sesión pasa al estado de transacción. En él, el cliente pide acciones al servidor POP3 con los pedidos necesarios, y este las atiende.

3) Cuando el cliente solicita el comando QUIT y lo recibe el servidor, la sesión entra en el estado de actualización. El servidor libera los recursos, se despide y cierra la conexión TCP.

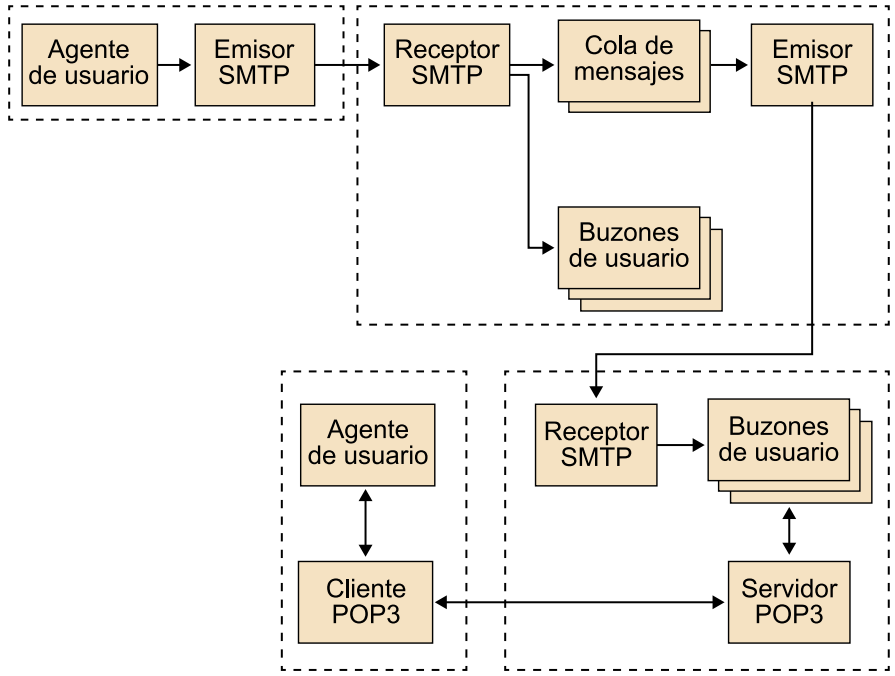
En este proceso, cliente y servidor se intercambian órdenes y respuestas siguiendo el modelo de diálogo de **Telnet**:

- **Órdenes**: órdenes de texto de cuatro caracteres seguidas de espacios y los argumentos que requieran. Finalizan con un <CRLF>.

- **Respuestas**: una cadena de caracteres que empieza por +OK o -ERR más una descripción.

En esta figura se presentan los elementos del modelo funcional del POP3 integrados en un sistema en el que se utiliza el SMTP para enviar el correo y el POP 3 para acceder a los buzones:

Figura 7



2.4.3. IMAP (*internet message access protocol* o protocolo de acceso a mensajes de internet)

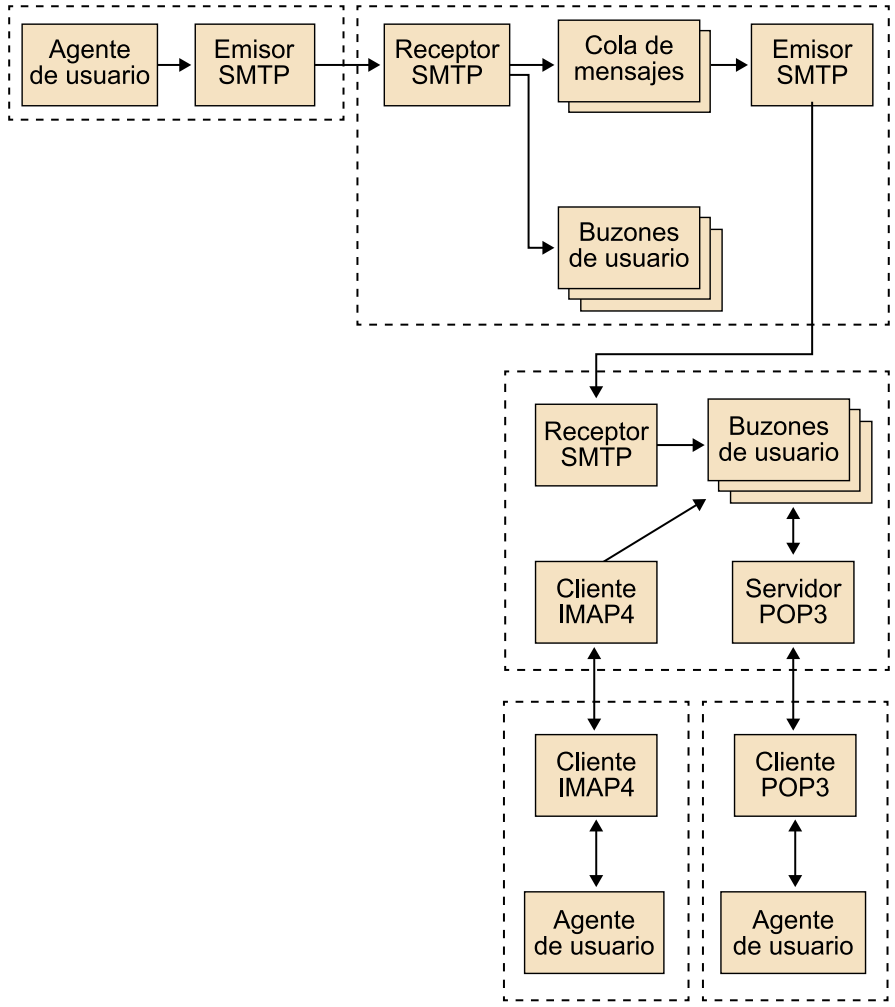
Este protocolo proporciona al usuario acceso remoto al buzón de correo. A diferencia del POP3, los mensajes de correo se depositan en el servidor, donde se almacenan estructurados en carpetas y donde se manipulan. Por lo tanto, es un protocolo más complejo que el anterior.

El protocolo de acceso a mensajes internet, en su versión actual 4 rev 1, se denomina **IMAP4rev1**, y está especificado en el RFC 3501. El IMAP4rev1 (a partir de ahora lo denominaremos IMAP4) permite al usuario disponer de diferentes buzones estructurados de manera jerárquica y, a la vez, poderlos manipular de manera remota, tal como hace con los buzones locales.

El IMAP4 se puede utilizar con cualquier protocolo de transporte fiable. Por norma general, se utiliza el TCP y, en este caso, se utiliza el puerto 143. Todas las interacciones entre cliente y servidor se llevan a cabo a modo de líneas ASCII acabadas con un carácter <CRLF>.

En esta figura se detallan los elementos del modelo funcional del IMAP4 integrados en un sistema en el que se utiliza SMTP para enviar el correo e IMAP4 para acceder a los buzones:

Figura 8



Nota

Debido a las características que hemos explicado, lo más habitual es que las aplicaciones de correo electrónico, por ejemplo, **Outlook de Microsoft**, tengan configurado un protocolo SMTP (para enviar correos electrónicos) y POP3 o IMAP4 (para recibirlos y gestionarlos).

Una de las diferencias más destacadas entre POP3 e IMAP es que, si el programa de correo electrónico utiliza POP3, entonces este descargará todos los mensajes al ordenador del usuario, sin dejar copia en el servidor de mensajería. Por el contrario, si usa IMAP, hará una copia en el ordenador del usuario y siempre mantendrá el mensaje en el servidor. Por lo tanto, en este segundo caso, si se borra un mensaje en el ordenador del usuario, este no queda borrado para siempre. Además, permite poder disponer de diferentes dispositivos clientes sincronizados.

2.5. Servicio de noticias NNTP (*network news transfer protocol*)

El servicio de noticias (en inglés, *news*) permite la remisión de mensajes, al igual que hace el servicio de correo electrónico, pero con la diferencia de que desde el origen no se especifica el destinatario o destinatarios, sino que cualquier usuario con acceso al servicio los puede leer. Por lo tanto, es comparable a un tablón de anuncios, donde todo el mundo puede leer los mensajes que hay colgados en él.

Los servidores de noticias se comunican entre sí para intercambiarse artículos por medio del NNTP (*network news transfer protocol*), especificado en el documento RFC 977. NNTP es un protocolo para la transferencia de información entre clientes mediante el acceso a grupos de noticias o *newsgroups*, los cuales contienen información clasificada por tópicos de interés.

Por norma general, la NNTP utiliza el protocolo de transporte TCP. El número de puerto asignado al servicio de noticias es 119.

2.6. Mensajería instantánea

La evolución de las tecnologías, de los dispositivos móviles y la generalización de internet, propiciada por un abaratamiento de costes, han provocado que muchos usuarios lleguen a comunicarse por internet mediante estos tipos de dispositivos. En este contexto se ha popularizado la comunicación entre usuarios mediante chats y mensajería instantánea (**IM**, *instant messaging*), con una comunicación real, instantánea, de texto, audio y vídeo.

XMPP, acrónimo de *extensible messaging presence protocol*, es un protocolo libre de mensajería instantánea basado en XML y estandarizado por la IETF. El puerto estándar para el XMPP es el 5222. Utiliza una arquitectura cliente-servidor descentralizada, en la que los clientes no se comunican directamente unos con otros, sino que lo hacen mediante los servidores. Aun así, otros sistemas de mensajería instantánea utilizan arquitecturas centralizadas.

Las funcionalidades principales que proporcionan los clientes de mensajería instantánea son comunicación de texto, audio y vídeo, transferencia de ficheros, compartición de escritorios, ubicaciones en los mapas, llamadas telefónicas, etc.

Los principales clientes de **mensajería instantánea móvil** son WhatsApp, Line y Telegram.

Otros clientes de **mensajería instantánea** son Hangouts, Pidgin (multirred), Skype, Trillian (multirred) o Viber.

2.7. Acceso a ordenadores remotos

Una de las aplicaciones del acceso remoto a ordenadores es la de poder administrar un equipo desde otro equipo a través de la red. También es empleado en las organizaciones para permitir que sus trabajadores accedan a los servidores de la empresa, ya sea desde la oficina o desde sus hogares.

El protocolo **Telnet** es el protocolo que se empleó en internet o en redes de área local para proporcionar una comunicación bidireccional interactiva en el acceso a ordenadores remotos. Se basa en el protocolo de transporte TCP, empleando el puerto 23. Normalmente sigue el modelo cliente/servidor.

SSH (acrónimo de *secure shell*) es un protocolo que se utiliza para acceder a máquinas remotas mediante la red. Soluciona la falta de seguridad del protocolo Telnet, al que se le añaden las siguientes funcionalidades: permite el cifrado para evitar que se puedan interceptar los datos que se envían y permite la autenticación con clave pública (que detallaremos en el próximo módulo) para asegurar que el ordenador remoto es quien dice ser.

El puerto estándar para contactar con un servidor SSH es el puerto 22.

Tanto Telnet como SSH son protocolos basados en texto (i. e., no gráficos). Por ello se desarrollaron algunas aplicaciones que permitieran una administración gráfica remota, como **VNC** (*virtual network computing*) y **RDP** (*remote desktop protocol*).

VNC es un desarrollo multiplataforma que dispone de implementaciones libres para diferentes sistemas operativos, de las que destacaremos **TightVNC**. Permite controlar la pantalla de un equipo desde otro equipo de manera remota. Por lo tanto, hay una exportación del escritorio de un equipo (servidor VNC), que es importado a otro equipo remoto (cliente) mediante un visor. El puerto TCP de un servidor VNC es el 5900.

Otra aplicación que también permite una gestión remota, muy utilizada actualmente, es **TeamViewer** (también disponible como aplicación para dispositivos móviles –apps–): http://www.teamviewer.com/es/

El protocolo **RDP** (*remote desktop protocol*) es un protocolo propietario de Microsoft. Es el que usa el servicio denominado *Terminal Services*. Este servicio utiliza por defecto el puerto TCP 3389 en el servidor para recibir las peticiones.

En este apartado también hay que hacer mención a que otra vía para la administración remota es el uso de aplicaciones locales, como navegadores web, que transmiten la interacción del usuario mediante algún protocolo de red hacia el equipo administrado. En el caso de los navegadores web, la interacción utilizará los protocolos HTTP y HTTPS. Con esta opción el cliente no necesita disponer de ninguna aplicación adicional. El componente canvas de HTML5, entre otros factores, ha ayudado a que esto sea posible.

3. Utilidades TCP/IP de los sistemas operativos

En las redes en las que se utiliza el protocolo TCP/IP, los sistemas operativos disponen de un conjunto de **utilidades** o **pequeñas aplicaciones** que nos facilitan información sobre nuestra configuración de red o de las conexiones y sobre el rendimiento de la red. Por lo tanto, nos pueden ser de utilidad para averiguar y solucionar problemas de funcionamiento y rendimiento de la red local.

Entre las utilidades más destacadas en entornos **Windows** (muchas de ellas también disponibles en entornos **Unix/Linux**) encontramos: Ipconfig/ifconfig, Ping, Tracert, hostname, Arp, Netstat, Nslookup, getmac, etc. El **sistema operativo de Apple OS X** también incorpora herramientas con estas funcionalidades: analizador del sistema, preferencias del sistema, utilidad de red (que incluye comandos como ping, traceroute, netstat, finger, etc.), terminal...

A pesar de que por su funcionalidad en algunos casos ya se han tratado en módulos anteriores, considerando que son realmente aplicaciones, realizamos una compilación también en este módulo.

Pasamos a describir brevemente algunas de estas utilidades:

1) ipconfig: En concreto, *ipconfig* puede sernos de utilidad para solucionar un problema de red TCP/IP, dado que nos permite comprobar la configuración de TCP/IP en el equipo que tiene el problema. Podemos utilizar el comando ipconfig para obtener información de la configuración del equipo, incluyendo la dirección IP, la máscara de subred y la puerta de enlace predeterminada. En las primeras versiones de Windows este comando era conocido como winipcfg en lugar de ipconfig.

Permite visualizar los valores de configuración de red del TCP/IP. Para visualizar sus parámetros principales, podemos ejecutar ***ipconfig/all***, que permite visualizar los parámetros de la configuración.

2) ping: El comando *ping* ayuda a comprobar la conectividad en el nivel IP, es decir, permite comprobar si un equipo o dispositivo de la red, con una IP asignada, se encuentra activo dentro de esta red. Podemos utilizar *ping* para enviar una solicitud a un nombre de *host* o una dirección IP de destino. Así podremos comprobar si podemos conectarnos a otros equipos u otros recursos de la red.

Por lo tanto, el comando ping es una orden que envía paquetes a un ordenador remoto y espera su respuesta.

3) hostname: Permite visualizar el nombre de la máquina local.

4) netstat: Windows (y también Linux) nos ofrecen una herramienta que nos va mostrando qué conexiones de red tenemos en cada momento. Para ejecutarla, podemos hacer: **netstat -an**.

Para entender mejor qué conexiones tenemos abiertas, lo mejor es que antes de ejecutar esta orden cerremos todos los programas a excepción de MSDOS (que tenemos activo con Símbolo del sistema), para así ir desde el principio comprobando qué conexiones tenemos y cuáles se van abriendo.

Si queremos que se actualice automáticamente la información, podemos escribir **netstat -an 5** (donde 5 puede ser cualquier número y hace referencia al intervalo de actualización, que puede ser el número de segundos que debe pasar hasta que actualice la información).

Para obtener una pequeña ayuda en relación con netstat ejecutaremos: **netstat/help**

5) tracert: El programa tracert de Windows (traceroute en Unix) permite ver por qué routers pasa una conexión de internet. Proporciona también información sobre el tiempo que tardan los paquetes al ir y volver a estos routers.

Para emplear tracert con Windows, solo hay que ejecutar, por ejemplo: tracert www.google.com.

También podemos encontrar esta herramienta para Mac OS X, yendo a Aplicaciones/Utilidades y abriendo la aplicación "Utilidad de Red", dentro encontraremos la pestaña "traceroute", donde al escribir el dominio o IP, se empezará a trazar la ruta.

6) nslookup: Con esta aplicación nos estamos conectando a nuestros servidores DNS para poder llegar a conocer la IP de un *host* concreto. Por ejemplo, si escribimos nslookup www.yahoo.com, obtendremos la IP de este servidor.

7) getmac: Muestra las direcciones MAC de los adaptadores de red que tengamos instalados en el sistema. Este número identifica de manera única cada adaptador de red.

8) Arp: Cada uno de los equipos que utilizan TCP/IP disponen de una tabla ARP con la que van grabando las direcciones IP y las direcciones MAC asociadas. Con esta orden se visualiza esta tabla con las direcciones estáticas y dinámicas, permitiendo también incorporar nuevas entradas (direcciones estáticas). Cada una de las entradas o registros de esta tabla tienen un tiempo de vida máximo, denominado *time to live* o TTL, y cuando este expira, la entrada correspondiente es borrada de la tabla.

4. Aplicaciones multimedia y sus protocolos

En este apartado describiremos toda una serie de técnicas y protocolos para trabajar con contenidos multimedia en la red internet, pero que también pueden ser extensibles a otros tipos de redes, como por ejemplo redes de videovigilancia.

En los últimos años han aparecido muchas aplicaciones que permiten transmitir audio y vídeo a través de internet, como por ejemplo *streaming* de vídeo, telefonía IP, audio y videoconferencias, radio y TV por internet, etc.

Estas aplicaciones tienen unos requisitos muy diferentes a los de las aplicaciones tradicionales, como la transferencia de ficheros, correo electrónico, etc.

Las aplicaciones multimedia tienen una **alta tolerancia a pérdidas de datos** en la transmisión, y en cambio son **muy sensibles a los retardos temporales** que se puedan producir en esta comunicación. Que no se vean unos pocos bits de un vídeo o sean erróneos puede no ser detectable por el ojo humano, pero en cambio un retardo de unos centenares de milisegundos puede provocar que el vídeo se entrecorte.

Las **redes multimedia** son aquellas que se utilizan primordialmente para el tráfico de voz, audio y vídeo. Considerando que han de trabajar con contenidos multimedia, es necesario que tengan muy en cuenta los retardos, el ***jitter*** y un mínimo ancho de banda; además de los parámetros de calidad que toda red de datos debe disponer.

Jitter

Un *jitter* en telecomunicaciones es una variabilidad en el tiempo de ejecución de los paquetes. Puede tratarse de un retardo o de un adelanto, que en aplicaciones multimedia provoca que no se puedan ejecutar adecuadamente.

Por lo tanto, en las aplicaciones multimedia hay tolerancia en cuanto a la pérdida de datos, siempre y cuando no haya retardos y las pérdidas sean ocasionales.

Para poder satisfacer estos requisitos, se utilizan dos mecanismos clave:

- Mecanismos de compresión específicos para cada tipo de tráfico.
- Protocolos de comunicación que optimicen la transmisión de datos en función de los requisitos mencionados.

Sabemos que el nivel de transporte ofrece toda una serie de servicios al nivel de aplicación, pero cada aplicación tiene unos requisitos diferentes, y por lo tanto el nivel de transporte los debe satisfacer. Los protocolos originales de internet TCP y UDP se concentraban solo con transferencias fiables, dado que las primeras aplicaciones no eran en tiempo real. A medida que se ha desarrollado

este tipo de aplicaciones se han especificado nuevos protocolos de transporte que son los que trataremos en este tema para ofrecer estos nuevos servicios relacionados con el multimedia.

4.1. Ejemplos de aplicaciones multimedia

Actualmente, en la red internet existen muchos tipos de aplicaciones multimedia. Podemos clasificarlas en tres grandes tipos:

1) *Streaming* de audio/vídeo almacenado.
2) *Streaming* en directo.
3) Audio y vídeo en tiempo real interactivo.

Son situaciones muy diferentes al caso clásico de bajarse un contenido multimedia y después visualizarlo, el cual ya quedaría cubierto con una transferencia de ficheros con protocolos HTTP y FTP.

4.1.1. *Streaming* de audio y vídeo almacenados

Con este tipo de aplicaciones, los clientes piden contenidos de audio y vídeo comprimidos que se encuentran almacenados en servidores. Estos contenidos pueden ser programas de televisión o de radio, vídeos, música, etc.

Estas aplicaciones se caracterizan por que los contenidos están **almacenados previamente en un servidor**, así el usuario puede detener la reproducción, rebobinar, etc. Utilizan la **técnica del *streaming*** y una **reproducción continúa**, por lo tanto, sin muchos retardos.

> La reproducción en tiempo real (***streaming*** en inglés) es una técnica que permite reproducir ficheros de audio y de vídeo. Con el *streaming*, el usuario ve una parte del contenido y el resto se va recibiendo a medida que se reproduce.

4.1.2. *Streaming* en directo de audio y vídeo

Este tipo de aplicaciones funciona como el *broadcast* tradicional de radio y televisión, en el que un emisor transmite a muchos receptores, pero en este caso se lleva a cabo en una red como internet.

En este caso, el contenido multimedia no se encuentra almacenado, por lo tanto el usuario no puede avanzar en su reproducción.

Se requiere una reproducción continua, y pueden darse retardos relevantes en el inicio de la reproducción.

4.1.3. Audio y vídeo en tiempo real interactivo

Este tipo de aplicaciones permiten que los usuarios interactúen en tiempo real de manera síncrona. Corresponden a aplicaciones que permiten establecer conexiones de audio y/o vídeo, llamadas telefónicas y/o videoconferencias, como por ejemplo Google Talk o Skype. Son aplicaciones especialmente sensibles a los retardos.

4.2. Compresión de audio y vídeo

Para enviar datos multimedia (audio y vídeo) por internet, primero hay que digitalizar y comprimir esos datos.

La razón de digitalizar la encontramos en la circunstancia de que las redes de ordenadores transmiten bits; y el hecho de comprimir viene dado porque el audio y el vídeo sin comprimir ocupa mucho espacio, y por lo tanto su transmisión puede llegar a consumir mucho ancho de banda.

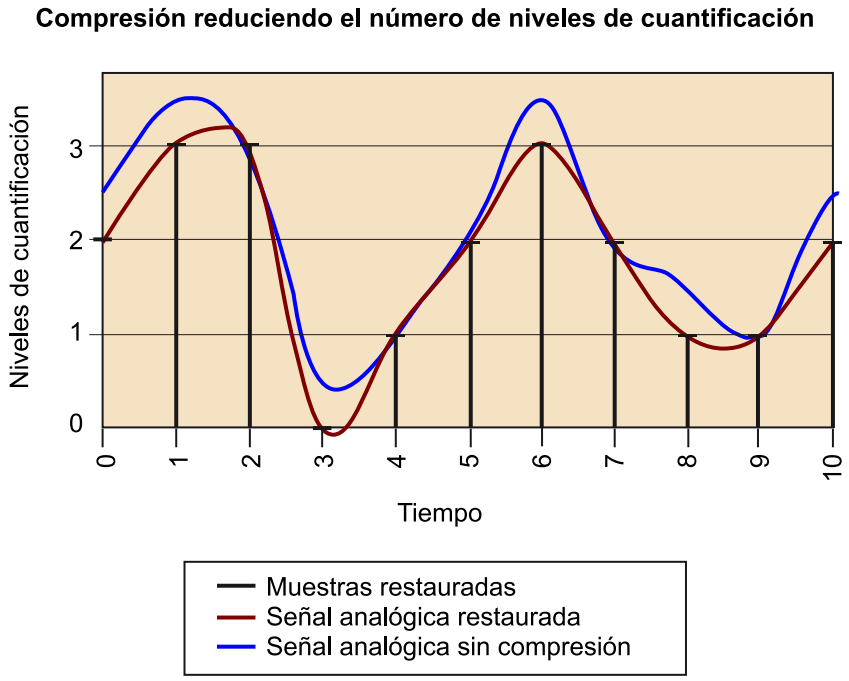
4.2.1. Compresión de audio

Una de las técnicas de compresión de audio es la de tipo **PCM** (*pulse code modulation*), que se basa en la recogida de muestras de audio a una frecuencia determinada. El valor de cada muestra elegida se redondea en un valor discreto y, por lo tanto, se puede representar con un número finito de bits que dependerá del número de valores que podemos tomar de las muestras.

En definitiva, PCM es un procedimiento de modulación utilizado para transformar una señal analógica en una secuencia de bits.

En el caso concreto de PCM empleado en la codificación de voz, se recogen 8.000 muestras por segundo, y cada muestra se representa con 8 bits. Por lo tanto, se tendrá una señal digital con una tasa de 64.000 bits por segundo (64kbs). A partir de la señal digital obtenida se podrá recuperar la señal analógica, a pesar de que lógicamente, por el muestreo utilizado, no será exactamente el mismo que el original. En el caso de recopilar más muestras, la señal analógica descodificada será más parecida a la original.

Figura 9



En este gráfico apreciamos cómo la señal analógica continua (en color azul) se discretiza en las muestras (color negro). Y cómo partir de estas muestras se reconstruye una señal analógica muy parecida a la original (en color rojo).

Para las compresiones de sonido con calidad, una de las técnicas más empleadas es el estándar **MPEG 1 Layer 3**, más conocido como **MP3**. Sus tasas de compresión son de 96 kbps, 128 kbps y 160 kbps.

4.2.2. Compresión de vídeo

El vídeo es una sucesión de imágenes enviadas en una tasa constante, que puede ser de unas 24 o 30 imágenes por segundo. Una imagen sin comprimir presenta una sucesión de píxeles, donde cada píxel se representa con un cierto número de bits que indican el color y la luminosidad.

Las técnicas de compresión de vídeo están basadas en disminuir la redundancia que presentan las imágenes consecutivas del vídeo (redundancia temporal), además de la redundancia propia dentro de la propia imagen (redundancia espacial). Disminuyendo estas redundancias se logra la compresión del vídeo.

Uno de los estándares más empleados en compresión de vídeo es MPEG, en sus diferentes versiones.

4.2.3. Formatos de audio y vídeo

Los formatos de audio más conocidos son:

- **WAV**: Acrónimo de ***waveform audio format***, es un formato de audio originario de Microsoft que permite grabar sonido en diferentes calidades, de 11.025 a 44.100 Hz. Ahora bien, dado que los archivos WAV tienen una calidad de sonido muy alta, suelen ocupar mucho espacio (por ejemplo,

un archivo de audio con una duración de tres minutos puede ocupar entre 20 y 30 MB en disco). Una de las principales ventajas del formato WAV es su posibilidad de conversión en otros formatos (por ejemplo, en MP3).

- **AIFF**: Acrónimo de ***audio interchange file format***. Tiene unas características muy similares a las del formato WAV. Fue diseñado para sistemas operativos Apple.

Los formatos WAV y AIFF ya no se utilizan tanto, pues existen formatos con una compresión mayor y una calidad parecida, como por ejemplo el formato MP3. Los dos nos permiten realizar *streaming*, pero se utilizan como base para otros formatos, como RealAudio. Si se comprimen, pierden mucha calidad.

- **MP3**: Es un formato de compresión de audio creado por el grupo Moving Picture Experts Group, bajo la supervisión del ISO. Los archivos de este formato se identifican con la extensión .**mp3**. Es uno de los formatos más populares en internet, dado que ofrece una buena calidad de sonido, con una elevada compresión de datos. Se puede realizar *streaming* con este formato y también se puede bajar con HTTP o FTP.

- **MP4**: Es un nuevo formato de audio basado en el estándar de codificación avanzada de audio. Se utiliza con diferentes extensiones:
  - .mp4: Extensión oficial para audio, vídeo y contenidos avanzados.
  - .m4a: Es la extensión adoptada por Apple para la distribución de música en iTunes y para ser reproducida en sus dispositivos.

- **MIDI**: Proviene de las siglas *musical instrument digital interface*, la interfaz digital para instrumentos musicales. Es un protocolo de comunicación estándar utilizado para comunicar instrumentos digitales electrónicos y, por lo tanto, datos entre sintetizadores, programas, procesadores de efectos y otros dispositivos. Se trata de un formato utilizado en el ámbito de la composición musical, y generalmente tiene la extensión **.mid**. Ocupan muy poco y en consecuencia son idóneos para conexiones con poco ancho de banda. Con este formato no se puede realizar *streaming*.

- **Real Audio**: Es un formato utilizado en internet para la reproducción en tiempo real, de manera que el archivo se reproduce mientras se descarga. Tiene el problema de que al ser archivos excesivamente grandes, se necesita una gran capacidad de almacenamiento.

Entre la diversidad de formatos de vídeo existentes destacaremos:

- **MPEG**: Estándar desarrollado por ***Moving Picture Experts Group*** (de aquí las siglas MPEG), un grupo de trabajo coordinado entre otros por el ISO. Soporta tres tipos de información: audio, vídeo y *streaming*. Ofrece una alta compresión con pocas pérdidas, por lo tanto es uno de los formatos

más empleados. Desde el año 1991 han ido apareciendo diferentes versiones: MPEG-1, MPEG-2, etc. Desde la web oficial de MPEG podemos ver las características de las diferentes versiones: http://mpeg.chiariglione.org/.

- **QuickTime Movie**: Es un formato propietario de Apple. Su extensión es **.mov**. Se puede visualizar con el reproductor QuickTime, que permite realizar vídeos del mismo formato y dispone de algunas opciones básicas para editarlos. Inicialmente estaba pensado solo como formato de vídeo, pero actualmente se puede emplear para cualquier tipo de medio (imágenes, audio, vídeo, Flash, etc). Con un servidor de vídeo dedicado es posible realizar *streaming* de vídeo de este formato.

- **Real Media**: Es uno de los formatos más empleados para realizar *streaming*. Es un formato propietario de Real Networks.

- **Windows Media Video**: Desarrollado por Microsoft para su reproductor Windows Media Player. Los archivos de este formato tienen extensión **.wmv**, que corresponde al archivo que contiene vídeo; **.wma**, que contiene audio, y **.asf** (*advanced streaming format*) como formato para *streaming*.

- **AVI**, ***audio/video interleaved***: Es un formato que fue definido por Microsoft y que posteriormente fue mejorado y denominado AVI 2.0. Este formato permite almacenar simultáneamente un flujo de datos de vídeo y varios flujos de audio (es decir, que puede contener bandas sonoras en varios idiomas). Ha sido sustituido por el formato Windows Media. Con este formato no se puede realizar *streaming*, dado que se debe almacenar primero el contenido y después reproducirlo.

- **Flash video**: Es un formato de vídeo creado por Flash que permite realizar *streaming*. La extensión de los ficheros en este formato es **.flv**. Funciona con la aplicación Flash Player. Permite las mismas funcionalidades que las animaciones efectuadas con Flash (ficheros **.swf**).

4.3. Protocolos para aplicaciones interactivas en tiempo real

El crecimiento exponencial de internet, con un acceso rápido a la descarga de ficheros grandes, permite que las tecnologías, en definitiva los protocolos asociados, vayan evolucionando.

Como alternativa a la utilización de los servicios estándares de internet FTP y HTTP para la transferencia de datos, sobre todo con información multimedia, se realiza una transferencia que sea procesada como un flujo regular y continuo, sin que haya que esperar a que la información multimedia haya llegado completamente para ser reproducida. Esto es lo que se denomina reproducción en tiempo real (o *streaming*).

La **reproducción en tiempo real (*streaming*)** transmite información multimedia en tiempo real utilizando el protocolo **RTSP** (*real time streaming protocol*) junto a otros protocolos de transporte en tiempo real como **RTP** (*real time transport protocol*) y el control de sesión dinámico **RTCP** (RTP *control protocol*). El *streaming*, y aquí está su ventaja, no utiliza el máximo ancho de banda del que dispone el cliente, sino solo el ancho de banda necesario para ir reproduciendo los contenidos en tiempo real. Además, no se realiza una descarga completa de los contenidos, sino que a medida que se reproducen los va descargando.

Dependiendo de cómo se obtenga la información que se va a difundir, la reproducción en tiempo real se puede dividir en dos categorías:

- **Reproducción del tiempo real en directo**: se lleva a cabo la transmisión de los acontecimientos en el mismo momento que están sucediendo. En este tipo de transmisión se utiliza el término *broadcast* (difusión), dado que se está transmitiendo en directo la misma información a todos los clientes.

- **Reproducción en tiempo real multimedia a la carta**, o **VoD**, *video on demand*, o **AVoD**, *audio and video on demand*, son sistemas que permiten a los usuarios la selección y reproducción de contenidos de audio y vídeo bajo demanda.

4.3.1. RTSP. *Real time streaming protocol*

Este protocolo establece y controla uno o varios *streamings* sincronizados de datos multimedia (audio y vídeo). Lleva el control remoto del envío mediante una red de servidores de datos multimedia.

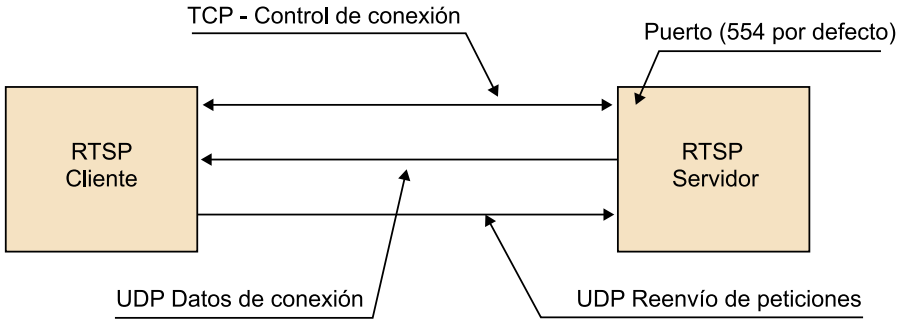
En RTSP no hay conexiones, solo sesiones mantenidas por el servidor. Y cada sesión tiene su identificador. Así, una sesión RTSP no está ligada a una conexión a nivel de transporte, por lo que se puede emplear tanto TCP como UDP. También puede llegar a trabajar junto con otros protocolos de transporte como RTP y RTCP.

Este protocolo está especificado en el RFC 2326.

Las operaciones que se llevan a cabo en este protocolo son:

- Recuperación de datos del servidor de contenidos multimedia.
- Invitación de un servidor de datos a una conferencia.
- Añadir contenido a una presentación existente.

Figura 10



Esquema de funcionamiento del protocolo RTSP.

4.3.2. RTP. *Real time transport protocol*

Es un protocolo de nivel de aplicación desarrollado por el Audio-Video Transporte Working Group de la IETF (Internet Engineering Task Force), que se utiliza para enviar cualquier tipo de formato: PCM, MP3, etc. para audio o H.263 para vídeo.

Es un protocolo complementario a otros protocolos de tiempo real, como SIP o H.323, que describiremos posteriormente.

Funciona sobre el protocolo de transporte UDP. El emisor encapsula un trozo de datos dentro del paquete RTP, que también se encapsula dentro de un segmento UDP, dentro de un paquete IP. El receptor extrae los datos RTP del segmento UDP y los pasa al reproductor para que este descodifique el contenido y lo reproduzca. Por lo tanto, observemos que se reconoce en los extremos, y los enrutadores no se preocupan del contenido de los paquetes IP que circulan por ellos.

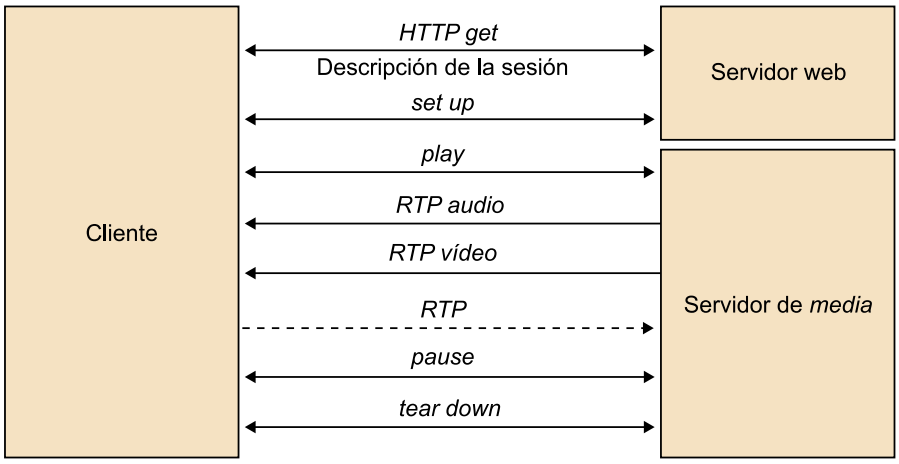
De hecho, RTP se encarga de facilitar la interoperabilidad entre aplicaciones multimedia.

Este protocolo no se preocupa de que los datos lleguen a destino o tengan la calidad adecuada, ni garantiza que los paquetes lleguen ordenadamente.

4.3.3. RTCP. *Real time control protocol*

Este protocolo utiliza el mismo mecanismo de distribución que el protocolo RTP, y se basa en la transmisión periódica de paquetes de control a todos los participantes en una sesión.

Figura 11



Funcionamiento del protocolo RTP/RTCP.

RTCP realiza cuatro funciones básicas:

1) Da información de la calidad de los datos distribuidos.

2) Mantiene un identificador persistente que permite que si algún identificador cambia se puedan recuperar los participantes de la sesión.

3) Controla la tasa de envío, por si hubiera un número demasiado elevado de participantes.

4) Esta función es opcional. Corresponde a la de comunicar un mínimo de información de control de la sesión, como por ejemplo que se muestre la identificación de un participante en la interfaz de usuario.

4.3.4. SIP. *Session initiation protocol*

SIP (*session initiation protocol*) es un protocolo del nivel de aplicación que permite inicializar, modificar y finalizar sesiones interactivas que impliquen elementos multimedia como vídeo, voz, mensajería instantánea, juegos en línea, realidad virtual, etc. Una de sus principales aplicaciones son **conferencias *multicast***, es decir, aplicaciones con diferentes usuarios a la vez. Junto con el protocolo H.323, también es uno de los principales protocolos empleados para **VoIP**.

Nota

Multicast, o difusión selectiva, es el envío de paquetes de información a múltiples destinatarios de una red de manera simultánea.

Nota

VoIP (acrónimo de *Voice over IP*), denominada también telefonía IP, es una tecnología que permite mantener conversaciones con voz en internet, o en cualquier red que utilice el protocolo TCP/IP. Por lo tanto, la señal de voz se envía digitalmente, en paquetes, en lugar de enviarse en forma digital o analógica mediante circuitos de telefonía convencionales.

Trabaja en cooperación con otros protocolos, como **RTP** y **RTCP** para el transporte y control de envío de datos, y **RTSP** para el control del *streaming* de datos multimedia; y permite que aplicaciones que utilizan los usuarios para comunicarse puedan ponerse de acuerdo con el tipo de sesión que quieren compartir.

SIP no ofrece servicios, sino primitivas que se pueden utilizar para ofrecer diferentes tipos de servicios. Permite gestionar sesiones de manera independiente de los protocolos de transporte que haya por debajo de estas. Habitualmente los clientes SIP usan el puerto TCP y UDP 5060 para conectarse a los servidores SIP.

4.3.5. H.323

Es un protocolo alternativo a SIP, muy empleado para transmitir audio y vídeo en tiempo real. Está dedicado a la transmisión de voz sobre IP (VoIP). De manera opcional también puede emplearse como soporte de vídeo.

Permite establecer comunicación entre un equipo conectado a internet y un teléfono conectado a la red telefónica.

Dentro de sus especificaciones incluye cómo negociar las codificaciones de audio y vídeo entre los extremos de la comunicación, cómo se envían las partes de audio y vídeo (utiliza RTP), cómo los equipos se comunican con sus *gatekeepers* (conmutadores virtuales opcionales que permiten la comunicación entre terminales H.323) y cómo los equipos conectados a internet se conectan con los teléfonos conectados a la red telefónica.

Como diferencias entre los protocolos SIP y H.323 tenemos las siguientes:

- H.323 es un servicio integrado de protocolos para llevar a cabo conferencias multimedia, según las especificaciones mencionadas, mientras que SIP solo se encarga del inicio y gestión de la sesión, sin ningún tipo de imposición en el transporte o en los formatos de audio y vídeo soportados.

- H.323 fue definido por la ITU, por lo tanto, desde la vertiente de telefonía, mientras que SIP fue definido por la IETF, por lo tanto, desde la vertiente de estándares de internet.

- H.323 es un estándar más complejo que SIP y, por lo tanto, este es más sencillo de implementar.

4.3.6. Skype

Es un sistema de telefonía de igual a igual (*peer-to-peer*, P2P) que funciona sobre la red internet. Fue desarrollado por el equipo que hizo KaZaA en el 2003, creada por Niklas Zennström y Janus Friis, basándose en el mismo protocolo que utilizaba el denominado **FastTrack**. En el 2011 fue adquirido por Microsoft.

Es la competencia de estándares de transmisión de voz sobre IP como **SIP** o **H.323**.

Entre las funcionalidades que tenemos se hallan el audio y las videoconferencias gratuitas.

En esta página podemos encontrar todas sus funcionalidades:

http://www.skype.com/es/features/

La arquitectura que implementa **FasTrack** es la de una red superpuesta (*overlay network*, en inglés), con dos tipos de nodos: los nodos normales y los supernodos. Un nodo normal corresponde al ordenador en el que el usuario ha instalado la aplicación Skype, y permite realizar llamadas y mensajes de texto. Entra en la red con su usuario y contraseña y de ese modo se conecta a un supernodo, que puede ser cualquier otro nodo con IP pública y bastantes recursos. Otro elemento es que el servidor de entrada es el que almacena a todos los usuarios y palabras de paso de toda la red Skype. Aun así, aparte de esta validación todas las otras operaciones en la red se realizan de manera totalmente descentralizada.

Considerando que su código es cerrado y el protocolo propietario, es difícil la interoperabilidad con otros sistemas.