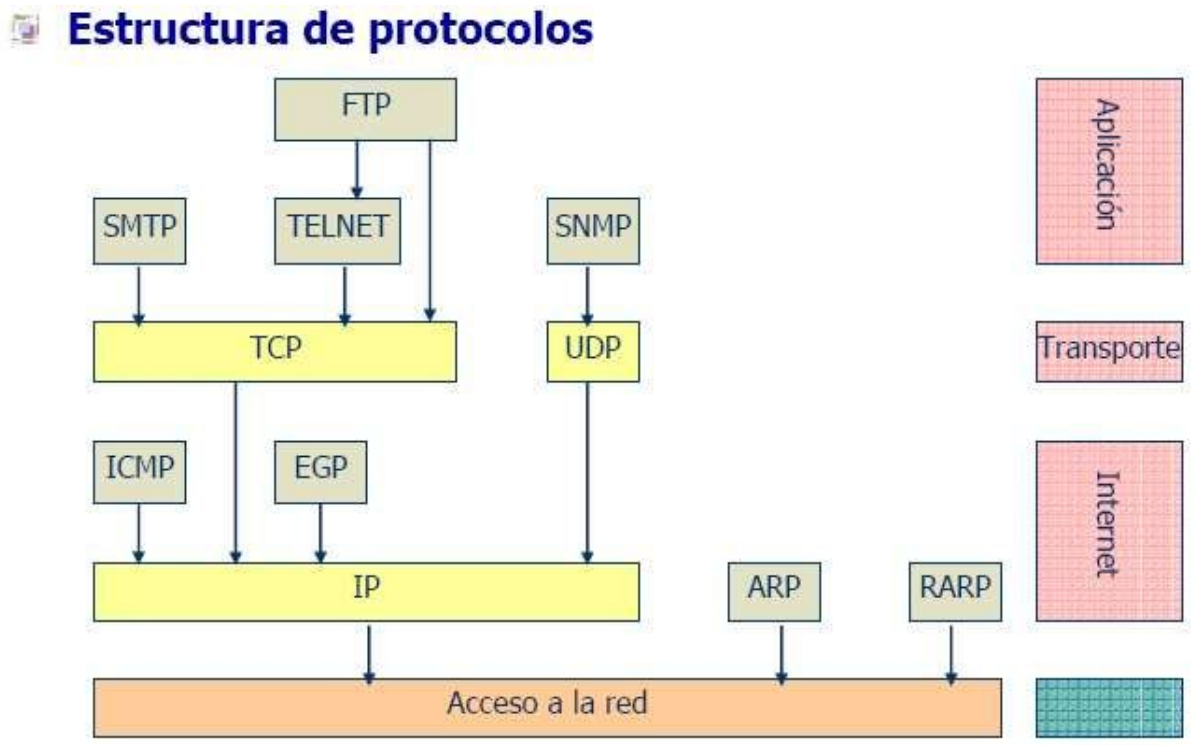
TEMA 2. SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

Introducción a las aplicaciones de red

Estructura de protocolos:



En este tema se fija la visión sobre la capa de aplicación de TCP/IP. Hay que tener siempre presente que la arquitectura de aplicaciones es muy diferente a la arquitectura de redes. Para este tema, nos vamos a centrar en la arquitectura de aplicaciones, más específicamente en el modelo cliente/servidor.

Arquitectura Cliente/Servidor

Una buena parte de las aplicaciones estandarizadas en Internet se basa en el modelo de interacción *cliente-servidor* siguiendo el modelo de red TCP/IP. El modelo cliente-servidor está compuesto por dos tipos de procesos:

- Cliente: es el proceso que inicia la comunicación. Funciona de manera intermitente, puede tener IP dinámica y privada, y se comunica con el servidor. Dos procesos clientes no se comunican entre sí.
- Servidor: es el proceso que espera a ser contactado. Está siempre en funcionamiento, su IP es permanente y pública, y pueden estar agrupados en granjas.

Sockets

Un socket es la interfaz entre la capa de aplicación y la capa de transporte en un host. También se refiere a la API entre la aplicación y la red. Cada proceso envía/recibe mensajes a/desde su socket.
Para interaccionar con una aplicación remota a través de la interfaz *socket*, la aplicación debe abrir la comunicación, invocando a la correspondiente llamada al sistema, la cual devuelve un descriptor, luego se escribe de y se lee en él y por último se cierra. Para recibir mensajes un proceso debe tener un identificador (IP + puerto). Un ejemplo de identificador sería:
Servidor web gaia.cs.umass.edu con dirección IP 128.119.245.12 y número de puerto 80.

Retardo en cola

Para estimar los retardos (tiempos) en cola se usa la teoría de colas. El uso de un servidor se modela con un sistema M/M/1. Este sistema tiene un solo servidor donde las llegadas están determinadas por un proceso de Poisson y los tiempos de servicio de un trabajo tienen una distribución exponencial.
El retardo en cola viene dado por la siguiente expresión:

$$R = \frac{\lambda \times (Ts)^2}{1 - \lambda \times Ts}$$

donde Ts es el tiempo de servicio y λ el ratio de llegadas de solicitudes.
Esta misma expresión se puede utilizar para calcular el retardo en cola de un router.
Esta ecuación solo es útil cuando Ts < λ y ambos son inferiores a 1.

¿Qué definen los protocolos de Internet?

Un protocolo define:

- Tipos de servicios.
- Tipos de mensajes. Por ejemplo request y response.
- Sintaxis. “Campos” y su estructura en el mensaje.
- Semántica. Significado de los “campos”.
- Reglas. Cuándo los procesos envían/responden a mensajes.

Existen varios tipos de protocolos según la clasificación:

- Protocolos de dominio público vs protocolos propietarios. Los protocolos de dominio público están definidos en RFCs. Un ejemplo de protocolo de dominio público es HTTP y un ejemplo de protocolo propietario es Skype.
- In-band vs out-of-band. Las señales de control se transmiten en el mismo canal que los datos o en canales diferentes (truco: todo en lo mismo “all-in” o todo separado “get out”).

- Stateless vs stateful. Un servidor es stateless si trata cada petición de manera independiente y no se relaciona con ninguna petición previa. Al contrario, un servidor stateful almacena información de consultas previas.
- Persistentes vs no persistentes. Es persistente si los datos siguen existiendo después de la ejecución del programa. Ocurre al contrario si es no persistente. Se podría ver como una base de datos frente a la ejecución de un programa secuencial.

La tendencia es hacer los protocolos flexibles de manera que se componen de dos partes principales:

- Una cabecera fija.
- Una serie de *trozos* (chunks) (obligatorios y 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.
        - Formato TLV (Type-Length-Variable) para los parámetros definidos por tipo de parámetro, longitud de parámetro y valor del parámetro.
        - Los parámetros comienzan en múltiplos de 4 bytes (pueden necesitar relleno).

Características

- Pérdida de datos (data loss). Algunas apps pueden tolerar alguna pérdida de datos (ej. audio); otras requieren transferencia 100% fiable (ej. FTP).
- Requisitos temporales (time sensitive). Algunas apps requieren bajo retraso (delay) para ser efectivas (ej. juegos interactivos).
- Rendimiento (throughput). Algunas apps requieren envío de datos a un ritmo determinado (sensible al ancho de banda) mientras que otras apps utilizan el ancho que esté disponible (elásticas).
- Seguridad. Encriptación, autenticación, no repudio,...

Protocolos de transporte

- Servicio TCP:
    - Orientado a conexión.
    - Transporte fiable.
    - Control de flujo.
    - Control de congestión.
- Servicio UDP:
    - No orientado a conexión.
    - Transporte no fiable.
    - Sin control de flujo.
    - Sin control de congestión. Esto hace que el emisor pueda enviar los datos al ritmo que desee (el throughput end-to-end real puede ser menor debido a la limitada capacidad de transmisión o a la congestión).

TCP y UDP (capa de transporte) al ser usuarios del protocolo IP (capa de red) no garantizan:

- Retardo acotado.
- Fluctuaciones acotadas.
- Mínimo throughput.
- Seguridad.

Servicio de Nombres de Dominio (DNS)

La comunicación en Internet precisa de direcciones IP (se verán en el tema 4), pero las personas prefieren identificar por nombres que por identificadores (ej: ¿conoces a mi amigo 78577540?). Para poder pasar de nombres a IPs se utiliza la resolución de nombres.

150.214.20.3 <-> goliat.ugr.es

La estructura jerárquica en dominios es:

Parte_local.dominio_niveln. … .dominio_nivel2.dominio_nivel1

El nivel 1 es el dominio genérico.
Inicialmente fueron definidos los siguientes 9 dominios genéricos (RFC 1591):
.com -> organizaciones comerciales.
.edu -> instituciones educativas, como universidades.
.gov -> instituciones gubernamentales.
.mil -> grupos militares.
.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)).

Resolución distribuida

La autoridad de los nombres de dominio está distribuida. La autoridad en orden descendente de los nombres es:

- Servidores “.” (punto). Son los servidores DNS.
- Servidores de dominio (Top-Level domain o TLD). Son responsables de los dominios de alto nivel como .org o .es.
- Servidores locales. Cada ISP (Internet Service Provider) tiene un servidor DNS asociado (también llamado un nombre predeterminado de servidor). Cuando un host se conecta a un ISP, el ISP le proporciona una (o más) dirección (direcciones) IP de su servidor DNS.
- Servidores autorizados y zona.Toda organización con hosts accesibles públicamente (como servidores Web y servidores de correo) deben proveer registros DNS de acceso público que mapee los nombres de los hosts a direcciones IP.

Un ejemplo de conexión podría ser:
jcp.ugr.es -> www.google.es

A. Conexión con el servidor local (IP conocida).
B. DNS local -> IP de destino.
C. Conexión destino.

Gestión de la base de datos DNS

Para afrontar el problema del escalado, el DNS utiliza una gran cantidad de servidores, organizados de forma jerárquica y distribuidos por todo el mundo. Cada zona debe tener al menos un servidor de autoridad. En cada zona hay servidores primarios (copia master de la base de datos) y secundarios (obtiene la base de datos por transferencia). Además, existe un servicio de caché para mejorar prestaciones.
La topología real de servidores es complicada: existen 13 servidores raíz (en realidad, cada servidor es una red de servidores replicados, por seguridad y fiabilidad).
Se debe remarcar que la base de datos no es lo mismo que la caché almacenada en DNS. La base de datos indica los dominios reservados (DNS internos) mientras que la caché está formada por los sitios visitados recientemente (solicitudes recientes al exterior).

Respuesta del servidor

- Respuesta CON autoridad: el servidor tiene autoridad sobre la zona en la que se encuentra el nombre solicitado y devuelve la dirección IP.
- Respuesta SIN autoridad: el servidor no tiene autoridad sobre la zona en la que se encuentra el nombre solicitado, pero lo tiene en caché o en una copia.
- No conoce la respuesta: el servidor preguntará a otros servidores de forma recursiva o iterativa. Normalmente se “eleva” la petición a uno de los servidores raíz.

La navegación Web

Una página Web es un fichero (HTML) formado por objetos como ficheros HTML, imágenes JPEG, Java applets, ficheros de audio,... Cada uno de estos objetos se direcciona por una URL (Uniform Resource Locator).
El protocolo HTTP sigue el modelo cliente-servidor donde:

- Cliente: navegador que pide, recibe y muestra objetos web.
- Servidor: envía objetos web en respuesta a peticiones.

Características de HTTP

- TCP al puerto 80: inicio de conexión TCP, envío HTTP, cierre de conexión TCP.
- HTTP es “stateless” -> cookies: el servidor no mantiene información sobre las peticiones de los clientes. Esto simplifica el diseño del servidor y permite a los servidores manejar miles de peticiones simultáneas; sin embargo, normalmente es deseable identificar a los usuarios, ya sea porque el sitio desea restringir el acceso o porque se quiere servir contenido como una función de identificación de usuario. Las cookies permiten mantener la información de los usuarios.
- Existen dos tipos de conexiones:
    - No persistente: en cada conexión se envía únicamente un objeto.
    - Persistente: en cada conexión puede enviarse múltiples objetos.

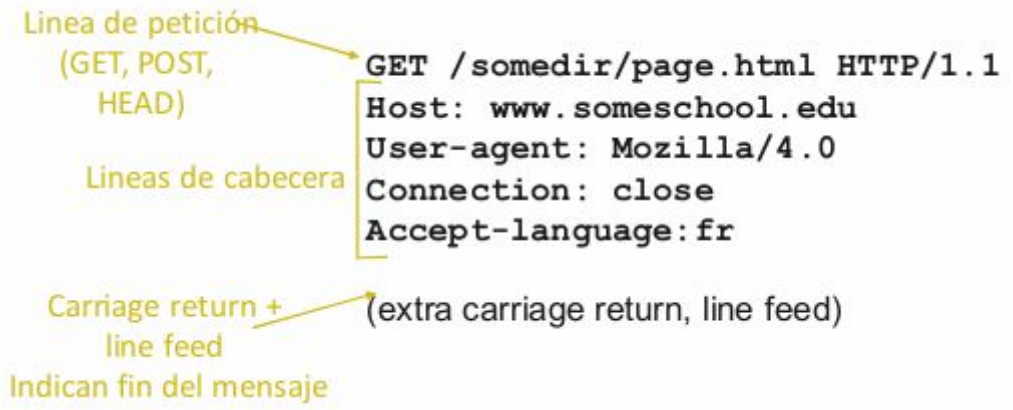
Mensaje HTTP

1a. Cliente HTTP inicia la conexión TCP al servidor HTTP (proceso) en www.ugr.es en el puerto 80.
1b. Servidor acepta la conexión y notifica al cliente.
2. Cliente HTTP envía *request message* del objeto pages/universidad.
3. Servidor HTTP envía el mensaje a través de su socket.
4. Si persistente -> envío de más objetos.
5. Cierre de conexión TCP.
6. Nuevas conexiones TCP.

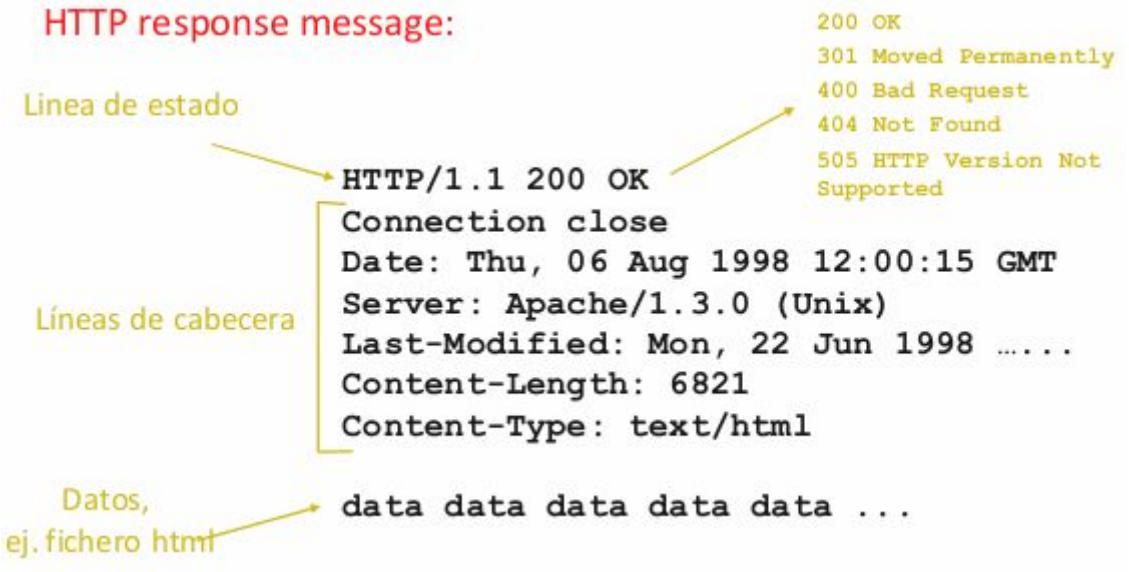
Tipos de mensajes HTTP

Existen dos tipos de mensajes HTTP: request y response.

- HTTP request message:



- HTTP response message:



Cache

El objetivo de la caché es satisfacer el requerimiento del cliente sin involucrar al servidor destino.

- Usuario configura el browser: acceso web vía caché.
- Browser envía todos los requerimientos HTTP a la caché:
    - Si el objeto está en caché: la caché devuelve el objeto.
    - Si no está en caché: requiere los objetos desde el servidor web, y devuelve el objeto al cliente.
- Conditional GET: no enviar objetos si la caché tiene la versión actualizada.
- Caché: especifica la fecha de la copia en el requerimiento HTTP.
- Servidor: responde sin el objeto si la copia de la caché es la última.

Tendencias actuales

- HTTP/2
    - Nace de SPDY de Google.
    - Compatibilidad hacia atrás (HTTP/1.1).
    - Una conexión, solicitudes en paralelo.
    - Cabeceras binarias, compresión.
    - Server push.
- QUIC
    - Similar a TCP+TLS+HTTP/2.
    - Sobre UDP.
    - Tiempo de conexión reducido.
    - Mejoras en control de congestión.
    - Multiplexación, corrección de errores,...

El correo electrónico

Como el correo postal, el e-mail es un medio de comunicación asíncrono; la gente envía y lee mensajes cuando es conveniente para ellos, sin tener que coordinarse con otras personas. Las cuatro componentes principales son:

- Cliente de correo (user agent).
- Servidor de correo (mail server o mail transfer agent).
- Simple mail transfer protocol (SMTP).
- Protocolos de descarga: POP3, IMAP, HTTP.

Los agentes de usuario permiten componer, editar y leer mensajes de correo (ej. Thunderbird).
Los mensajes salientes (outgoing) y entrantes (incoming) de correo son almacenados en el mismo servidor de correo. Una analogía sería ver el servidor de correo como la casa de correos, donde están los mensajes a recibir y a enviar.

SMTP

Para describir el Simple Mail Transfer Protocol (SMTP) primero se analizan los pasos en el envío/recepción de correo.

1. El usuario origen compone mediante su Agente de Usuario un mensaje dirigido a la dirección de correo del usuario destino.
2. Se envía con SMTP o HTTP el mensaje al servidor de correo 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 del usuario destino.
4. El cliente SMTP envía el mensaje sobre la conexión TCP.
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 para leer el mensaje utilizando POP3, IMAP o HTTP.

SMTP se implementa mediante dos programas (incluidos ambos en cada mail server):

- Cliente SMTP: se ejecuta en el mail server que está enviando el correo.
- Servidor SMTP: se ejecuta en el mail server que está recibiendo el correo.

Usa TCP y está compuesto por tres fases:

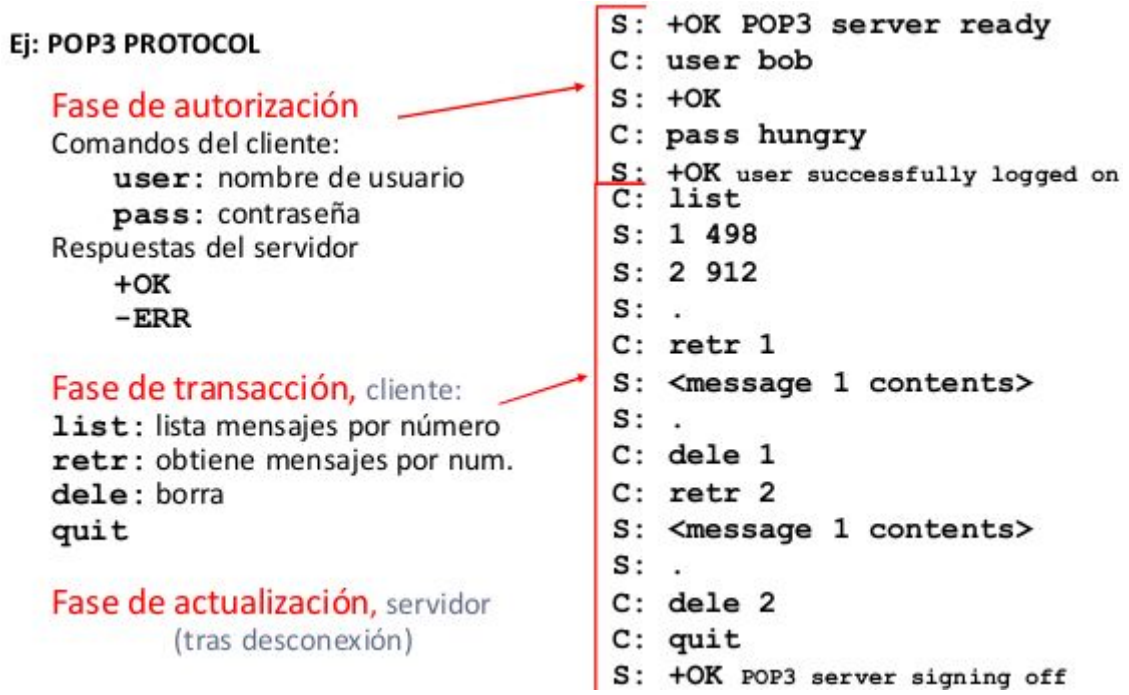
- Handshaking.
- Transferencia de mensajes.
- Cierre.

La interacción entre cliente SMTP y servidor SMTP se realiza mediante comandos (texto ASCII) /respuesta (código de estado y frases).

Los mensajes deben estar codificados en ASCII de 7 bits (extensiones MIME).

Protocolos de acceso al correo

- POP3: es un protocolo simple pero su funcionalidad está limitada. Empieza cuando el agente de usuario abre una conexión TCP al servidor de correo. Una vez establece la conexión TCP, recorre tres fases: autorización, transacción y actualización.



- IMAP: para solucionar los problemas que tenía POP3 se inventó IMAP. Tiene más funcionalidades que POP3, pero es más complejo. Las ventajas que ofrece IMAP son:
    - Organización en carpetas en el lado del servidor (MTA).
    - Mantiene información entre sesiones.
    - Permite la descarga de parte de los mensajes.
    - Se puede acceder con varios clientes (con POP3 también, pero solo en modo descargar y guardar).
- Web MAIL: el agente de usuario es un navegador web y el usuario se comunica con su mailbox vía HTTP. Permite la organización total en el servidor, accesible desde cualquier cliente con HTTP y aumenta su seguridad con el uso extendido de HTTPS.

Algunos puertos utilizados son 110 (POP3), 143(IMAP), 443(HTTPS).

Seguridad y protocolos seguros

Primitivas de seguridad

- Integridad: la información no ha sido manipulada.
- Confidencialidad/privacidad: sólo accede a la información quien debe hacerlo.
- Autenticación: los agentes de la comunicación son quien dicen ser.
- No repudio: no se puede negar el autor de una determinada acción.
- Control de accesos: garantía de identidad para el acceso.
- Disponibilidad: acceso a información o servicios de los usuarios autorizados cuando éstos los solicitan o necesitan.

Mecanismos de seguridad

- Cifrado simétrico: C = K(P) & P = K(C)
    - DES, 3DES, AES.
- Cifrado asimétrico: C = K^+(P) & P = K^-©
    - Diffie & Hellman, RSA.
- Message Authentication Code: M | F(M,K)
    - MD5, SHA-1.
- Firma digital: M | F(M,K^-). Comprobación con K^+
- Certificado: (ID + K^+) | F((ID + K^+), K^{-CA})

Seguridad

- Seguridad (criptográfica) en protocolos:
    - Capa de aplicación
        - Pretty Good Privacy (PGP)
        - Secure Shell (SSH)
    - Capa de sesión (entre aplicación y transporte)
        - Secure Socket Layer (SSL) -> HTTPS, IMAPS
        - Transport Secure Layer (TSL)
    - Capa de red -> IPSec (VPN)
- Seguridad perimetral y gestión de riesgos:
    - Firewalls, UTMs.
    - Sistemas de detección de intrusiones (IDS) en red (NIDS) o host (HIDS).
    - Antivirus.
    - Evaluación de vulnerabilidades.
    - Seguridad en aplicaciones, filtrado web, anti-spam.
    - Advanced Threat Detection.
    - SEMs, SiEMs.

Aplicaciones multimedia

Conceptos

- Aplicaciones multimedia: son las relativas a audio y vídeo.
- Calidad de servicio (QoS): capacidad de ofrecer el rendimiento requerido para una aplicación.
- Mejor esfuerzo (best effort): sin garantías de QoS.

Tipos de aplicaciones

- Flujo de audio y video (streaming) almacenado (ej.: Youtube).
- Flujo de audio y video en vivo (ej.: emisoras de radio).
- Audio y vídeo interactivo (ej.: Skype).

Características fundamentales

- Elevado ancho de banda.
- Tolerantes a la pérdida de datos.
- Delay acotado.
- Jitter acotado.
- Uso de multicast.

Aplicaciones para interconectividad de redes locales

DHCP

- Configuración dinámica de direcciones IP.
- Se utiliza para la asignación dinámica de direcciones IP.

DyDNS, No-IP,...

- Servicios en la red privada, con IP pública variable.
- Es necesaria la configuración en router de acceso.

UPnP

- “Pervasive adhoc”.
- Los dispositivos se comunican mediante NAT (Network Address Translation).