

TEMA 2 FUNDAMENTOS DE REDES.pdf



pikopakoi



Fundamentos de Redes



3º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.





**KEEP
CALM
AND
ESTUDIA
UN POQUITO**



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play



18

[Ver mis op](#)

Continúa d



405416_arts_esce
ues2016juniy.pdf

Top de tu gu



7CR



Rocio



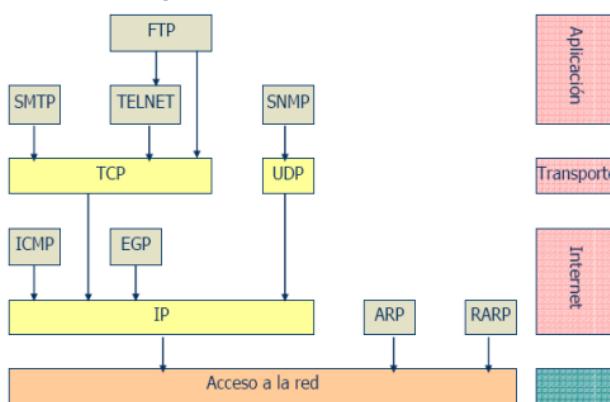
pony

[Inicio](#)

Asign

TEMA 2. SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET.

1. Introducción a las aplicaciones de red.



Servidor:

- Siempre en funcionamiento.
- IP permanente y pública.
- Agrupados en "granjas".

Clients:

- Funcionando intermitentemente.
- Pueden tener IP dinámica y privada.
- Se comunican con el servidor.
- No se comunican entre sí.

Proceso cliente: proceso que inicia la comunicación.

Proceso servidor: proceso que espera a ser contactado -> IP permanente y pública.

- ➔ Proceso envía/ recibe mensajes a/desde su socket.
- ➔ Para recibir mensajes un proceso debe tener un identificador (IP + puerto).

Retardo en cola:

- Para estimar los retardos (tiempos) en cola se usa la teoría de colas:
 - o El uso de un servidor se modela con un sistema M/M/1.
 - o El retardo encola es:

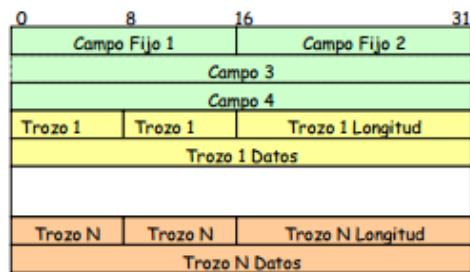
$$R = \frac{\lambda(T_s)^2}{1 - \lambda T_s}$$

Donde T_s es el tiempo de servicio y (λ) el ratio de llegada de solicitudes.

- Esta misma expresión se puede utilizar para calcular el retardo en cola en un router.

¿Qué define un protocolo?

- Tipos de servicios.
- Tipos de mensajes.
- Sintaxis (Estructura de "campos" en el mensaje)
- Semántica (Significado de los "campos")
- Reglas (cuando los procesos envían mensajes/responden a mensajes).

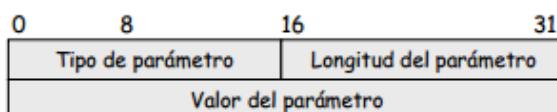


Tipos:

- Protocolo de dominio público.
- Protocolos propietarios.
- In-band versus put-of-band.
- Stateless versus state-full.
- Persistentes versus no-persistentes.

Tendencia: hacer los protocolos flexibles con:

- Una cabecera fija.
- Una serie de "trozos" (obligatorios y opcionales).
 - o 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.



- Los parámetros comienzan en múltiplos de 4 bytes (puede necesitarse relleno).

Características:

- **Pérdida de datos:** Algunas aps (ej: audio) pueden tolerar alguna pérdida de datos; otras (ej: FTP, telnet) requieren transferencia 100% fiable.
- **Requisitos temporales:** Algunas aps (ej: telefonía Internet, juegos interactivos) requieren bajo retraso (delay) para ser efectivas.
- **Rendimiento:** Algunas aps requieren envío de datos a un ritmo determinado.
- **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.

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

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

2. Servicio de nombres de dominio (DNS).

- La comunicación en Internet precisa de direcciones IP.
- Las personas prefieren “nombres”.
- DNS: traducción de nombres direcciones IP (resolución de nombres).
- Estructura jerárquica de dominios.

Parte_local.dominio_niveln.dominio_nivel2.dominio_nivel1

- Nivel1 es el dominio genérico.
- ICANN (Internet Corporation for Assigned Names and Numbers) que suele delegar en centros regionales.

Inicialmente fueron definidos los siguientes 9 dominios genéricos:

- .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 zona geográfica (ej. Es(España); pt(Portugal) ...)

Resolución distribuida:

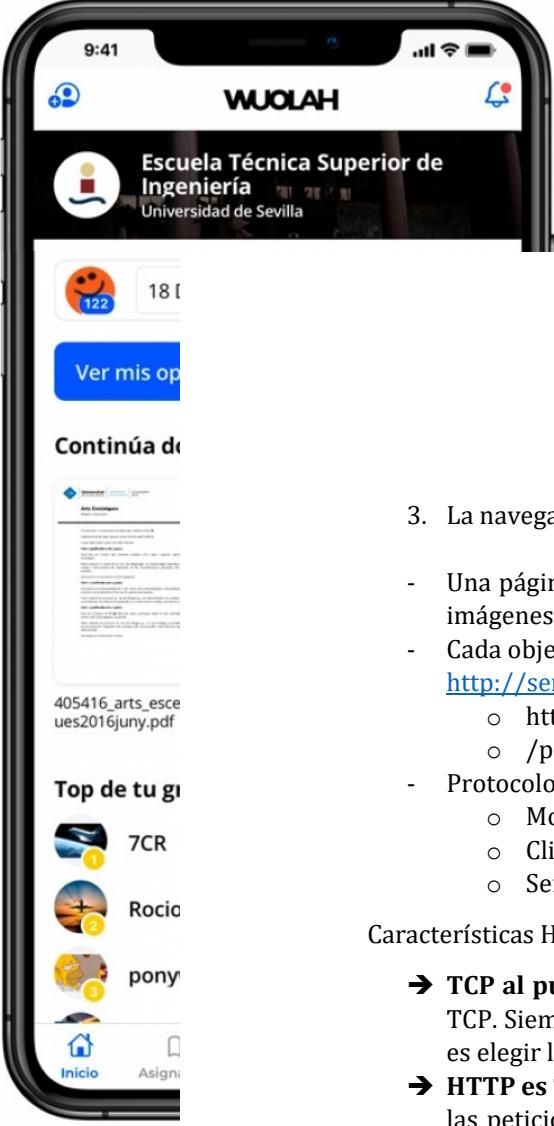
- Servidores “.”
- Servidores de dominio.
- Servidores locales.
- Servidores autorizados y zona.

Gestión de la base de datos DNS:

- ➔ Cada zona debe tener al menos un servidor de autoridad.
- ➔ En cada zona hay servidores primarios (copia master de la db) y secundarios (obtiene la db por transferencia).
- ➔ Además, existe un servicio de cache para mejorar prestaciones.
- ➔ La topología real de servidores es complicada: existe 13 servidores raíz.
- ➔ El root-server F (y otros) tiene un servidor en Madrid.

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 la cache.
- ➔ **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.



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

3. La navegación Web.

- Una página web es un fichero (HTML) formado por objetos ficheros HTML, imágenes JPEG, Java applets, ficheros de audio, ...
- Cada objeto se direcciona por una URL.
[http://servidor\[:puerto\]/path](http://servidor[:puerto]/path)
 - o http: protocolo
 - o /path: servidor web
- Protocolo HTTP:
 - o Modelo cliente-servidor.
 - o Cliente: browser que pide, recibe y muestra objetos web.
 - o Server: envía objetos web en respuesta a peticiones.

Características HTTP:

- ➔ **TCP al puerto 80:** Inicio de conexión TCP, envío HTTP, cierre de conexión TCP. Siempre que se diseña un nuevo protocolo la primera decisión a tomar es elegir la capa de soporte.
- ➔ **HTTP es "stateless" -> Cookies:** El servidor no mantiene información sobre las peticiones de los clientes, es decir, no se guarda estado, implica que el servidor no lo reconoce, esto puede ser problemático y por ello se crearon las cookies (información sobre el usuario). Si no fuese stateless implica una base de datos a la que se puede acceder muy rápido.
- ➔ **Existen dos tipos:**
 - o *No persistente*: Se envía únicamente un objeto en cada conexión TCP.
 - o *Persistente*: Pueden enviarse múltiples objetos sobre una única conexión TCP entre cliente y servidor.
Única conexión TCP, una única descarga, ventaja: no gasta tantos recursos y no hay que hacer una conexión nueva por cada objeto.

Mensajes HTTP:

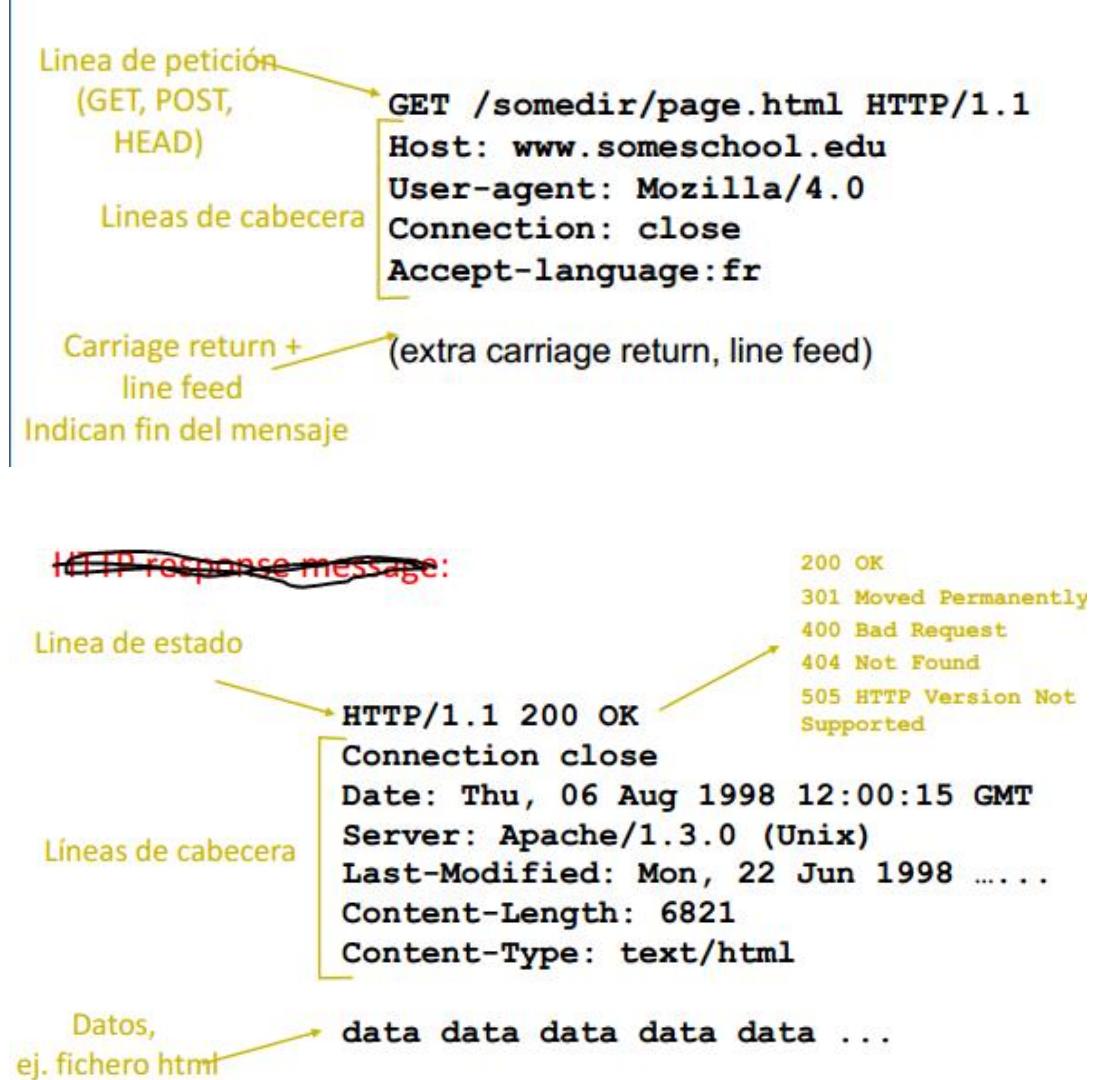
1. (a) Cliente HTTP inicia conexión TCP al servidor HTTP (proceso) en www.ugr.esen puerto 80.
1. (b) Servidor HTTP acepta la conexión y notifica al cliente.
2. Cliente HTTP envía request message del objeto pages/universidad.
3. El servidor HTTP envía el mensaje a través de su socket.
4. Si persistente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

Conclusión: El navegador abre una hebra, y coge la reserva del SO, un puerto, y manda un paquete de solicitud de conexión web, solicitud TCP. El servidor acepta (no tiene por qué aceptar siempre, depende de los recursos, el servidor web puede ser concurrente mayoritariamente y puede recibir varias peticiones y contestarlas, tendrá un límite de clientes para darles servicios).

Una vez establecida la conexión (paso 2) empieza el protocolo TCP. Se envía un paquete de respuesta.

Tipos de mensajes HTTP:

- Dos tipos de mensajes HTTP: request, response HTTP request message:



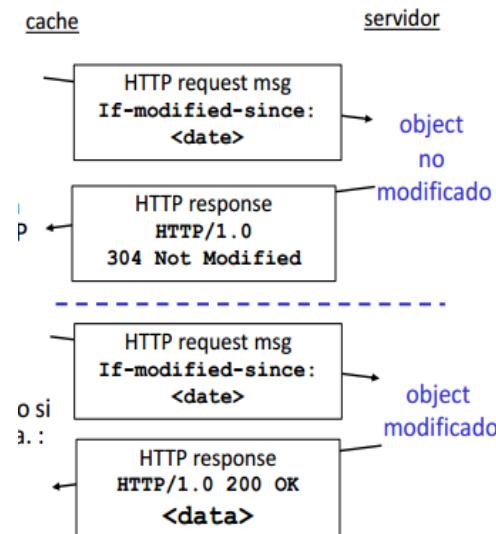
Caché: satisfacer el requerimiento del cliente sin involucrar al servidor destino.

- Usuario configura el browser: Acceso Web vía caché.
- Browser envía todos los requerimientos HTTP al caché.
 - o Si objeto está en caché: caché retorna objeto.
 - o Sino caché requiere los objetos desde el servidor Web, y retorna el objeto al cliente.
 - o Ejemplo de respuesta (servidor a cache/cliente):
 - 1ro y 2do: máximo tiempo que esa página puede tenerse dentro de la caché.
 - 3ro: cuando expira esa página, minimizamos la posibilidad de tener una página obsoleta.

```
HTTP/1.1 200 OK
Date: Fri, 30 Oct 1998 13:19:41 GMT
Server: Apache/1.3.3 (Unix)
Cache-Control: max-age=3600
Expires: Fri, 30 Oct 1998 14:19:41 GMT
Last-Modified: Mon, 29 Jun 1998 02:28:12 GMT
ETag: "3e86-410-3596fbbe"
Content-Length: 1040
Content-Type: text/html
```

Web caché:

- Conditional GET: no enviar objetos si el caché tiene la versión actualizada. Comando que puede mandar la cache cuando recibe una solicitud.
Ventaja: si no se ha modificado la web no hay que volver a reenviarla, cuando una caché hace un conditional get implica que nunca vamos a ver una página desactualizada, implica que se consuman más recursos.
- Caché: especifica la fecha de la copia en el requerimiento HTTP.



If-modified-since:

<date>

If-none-match:

“68689769‡7c876b7e”

- Servidor: responde sin el objeto si la copia de la cache es la última:
 HTTP/1.0 304 Not
 Modified

Tendencias actuales:

- HTTP/2 (mejora la eficiencia en la web, para una única conexión con un único puerto que permitía conexiones en paralelo. También permite cabeceras binarias.)
 - o Nace de SPDY, de Google.
 - o Compatibilidad hacia atrás (HTTP/1.1)
 - o Una conexión, solicitudes en paralelo.
 - o Cabeceras binarias, compresión.
 - o Server push.
- QUIC
 - o Similar a TCP+TLS+HTTP/2
 - o Sobre UDP
 - o Tiempo de conexión reducido.
 - o Mejoras en control de congestión.
 - o Multiplexación, corrección de errores.

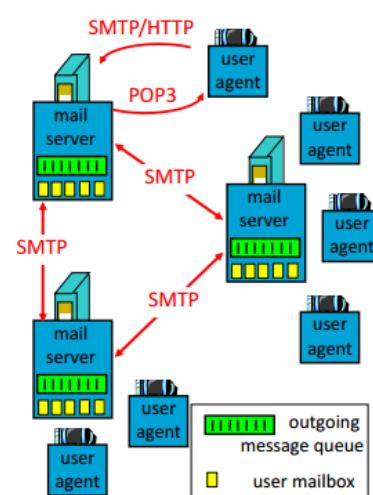
4. El correo electrónico.

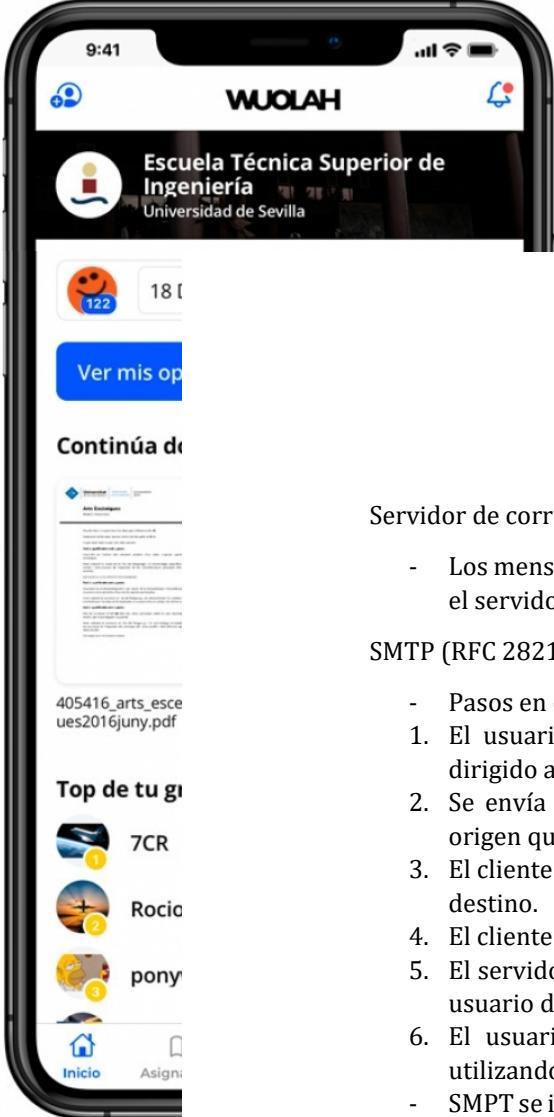
Cuatro componentes principales:

- 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.

Agente de usuario:

- Componer, editar y leer correos mensajes de correo. (Outlook, thunderbird)





Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

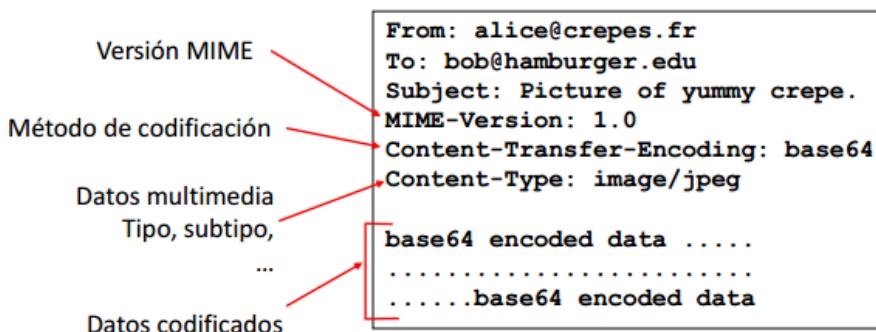
Servidor de correo:

- Los mensajes salientes (outgoing) y entrantes de correo son almacenados en el servidor de correo.

SMTP (RFC 2821):

- 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):
 - o Cliente SMTP: se ejecuta en el mail server que está enviando correo.
 - o Servidor SMTP: se ejecuta en el mail server que está recibiendo correo.
- Usa TCP.
- Tres fases:
 - o Handshaking ("saludo").
 - o Transferencia de mensajes.
 - o Cierre
- La interacción entre cliente SMTP y servidor SMTP se realiza mediante comandos/respuesta.
 - o Comandos: texto ASCII
 - o Respuestas: código de estado y frases.
- Los mensajes deben estar codificados en ASCII de 7 bits -> Extensiones MIME.

MIME: Multimedia mail extensión, RFC 2045, 2056.



Ej: POP3 PROTOCOL

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, 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
```

Ventajas de IMAP:

- Permite organización en carpetas en el lado del servidor (MTA).
- Para ello, mantiene información entre sesiones.
- Permite la descarga de partes de los mensajes.
- Posible acceder con varios clientes (POP también, pero en modo descargar y guardar).

Ventajas de WEB MAIL:

- Organización total en el servidor, accesible desde cualquier cliente con HTTP.
- Seguridad: Uso extendido de HTTPS.

5. Seguridad & protocolos seguros.

Primitivas de seguridad:

- Confidencialidad:
 - o Solo accede a la información quien debe hacerlo.
- Responsabilidad:
 - o Autenticación: Los agentes de la comunicación son quien dicen ser.
 - o No repudio: No se puede negar el autor de una determinada acción.
 - o Control de accesos: Garantía de identidad para el acceso.
- Integridad:
 - o La información no ha sido manipulada.
- Disponibilidad:
 - o Acceso a los servicios.

Protocolos seguros:

➤ Mecanismos de seguridad:

- Cifrado simétrico: $C = K(P) \& P = K(C)$
 - o DES, 3DES, AES, RC4
- Cifrado asimétrico: $C = K^*(P) \& P = K(C)$
 - o Diffie & Hellman, RSA
 - o Muy lento, tarda demasiado.
- Message Authentication Code: $M | F(M, K)$
 - o MD5, SHA-1, ...
 - o MAC: cifrado con clave, evita ataques.
- Firma Digital: $M | F(M, K) \rightarrow$ comprobación con K^+
 - o Estrategia de aplicación al revés del cifrado asimétrico, es un MAC del documento con mi clave privada y cualquiera con mi clave pública puede utilizarlo para comprobarlo.
- Certificado: $(ID + K) | F((ID + K), K - ca)$
 - o Firma digital que firma una autoridad certificadora y nos fiamos de que es verdadero y tiene un prestigio suficiente para creernos lo que firma y es verdad la asociación con mi clave pública.

➤ Seguridad:

- Seguridad (criptográfica) en protocolos:
 - Capa de aplicación:
 - Pretty Good Privacy (PGP): Sistema para encriptar ficheros, de los primeros que surgen. No se basan en una autoridad certificadora sino en una cadena de confianza.
 - Secure Shell (SSH): Telnet sobre un TLS, telnet con una cobertura de criptología pero no usa TLS.
 - Capa de sesión (entre aplicación y transporte)
 - Secure Socket Layer (SSL) -> HTTPS, IMAPS, SSL-POP, VPN
 - Transport Layer Security (TLS): Trabaja mucho en ver qué algoritmos anteriores (mecanismos de seguridad) cuales se usan y cuáles no, el uso de RC4 no es recomendable, desaparece, solo queda AES que no han encontrado un ataque que lo descifre.
 - Diferencia entre SSL y TLS: SSL en versión 3 se llamó TLS. Existe cierta compatibilidad, pero cuando pasó a llamarse SSL hubo un cambio brusco.
 - Capa de red -> IPSec (VPN): TLS es una forma de hacer VPN. Los puertos, en TCP, están en capa de transporte y el IP está en capa de red y el protocolo en protocolo IP.
- Seguridad Perimetral y Gestión de Riesgos:
 - Firewalls, UTM's: Analiza paquetes en capa 3 y capa 4, es decir, mirando IP y puertos y bloqueando servicios no interesantes. Firewall de capa 7: miran el contenido del paquete para identificar spam y cosas que no son seguras, pero son bastante lentos y complicados.
 - Sistemas de detección de intrusiones (IDS) en red (NIDS) o host (HIDS).
IDS: Ejemplo IDS-> WAZUH: empresa granadina con sede en EEUU. Funciona como un antivirus, pero en vez de mirar el código de ejecutables, mira el uso de red dentro del sistema.
NIDS: es el IDS más conocido, es un sistema que analiza paquetes, (sistema de reglas) si el paquete cumple ciertas normas es peligroso.
 - Antivirus.
 - Evaluación de vulnerabilidades: Software que tiene una BD de los peligros. Va puerto por puerto, coteja con su software y mira en la BD de sus vulnerabilidades.
 - Seguridad en Aplicaciones (Firewall capa 7), filtrado web, anti-spam.
 - Advanced Threat Detection.
 - SEMs, SIEMs. (sistemas más gordos y caros)



Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play



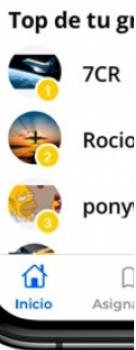
122

Ver mis op

Continúa d

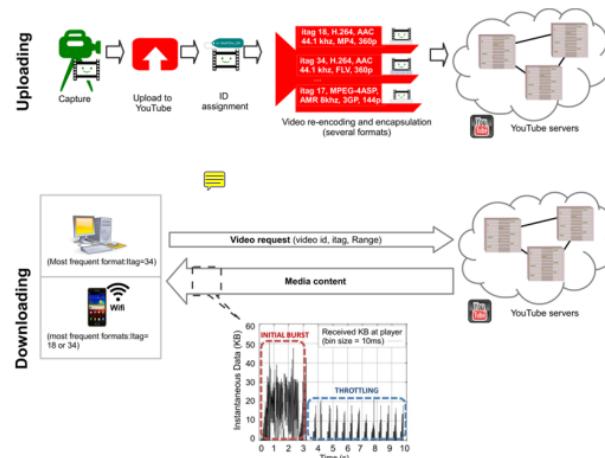


405416_arts_esce ues2016juniy.pdf



6. Aplicaciones multimedia.

- Conceptos:
 - o Aplicaciones multimedia: audio y video.
 - o Calidad de servicio (QoS): capacidad de ofrecer el rendimiento requerido para una aplicación. (Internet es una red en la que no hay garantía de la calidad del servicio, sino que se hace lo posible).
 - o Mejor esfuerzo (best effort): sin garantía de QoS.
- Tipos de aplicaciones:
 - o Flujo de audio y video (streaming) almacenado -> Ej: Youtube.
 - o Flujo de audio y video en vivo -> Ej: Emisoras de radio o IPTV.
 - o Audio y video interactivo -> Ej: Skype.
- Características fundamentales:
 - o Elevado ancho de banda.
 - o Tolerantes a la pérdida de datos.
 - o Delay acotado.
 - o Jitter acotado.
 - o Uso de multicast. (Cuando enviamos streaming a varios destinos, como una radio en tiempo real en internet. Única transmisión escuchando todos).



7. Aplicaciones para interconectividad de redes locales.
 - Son protocolos de red local. Nos permiten mejorar prestaciones dentro de una red local como NAT.
 - DHCP:
 - o Configuración dinámica de direcciones IP.
 - o En redes inalámbricas asignan IP aleatorias al cliente. Es un servicio con base de datos de IP, y se hace una petición al servidor y se asignan las IPs sin falta de configurar nada.
 - o Funcionamiento: servicio sencillo.
 - DynDNS, No-IP,...
 - o Servicios en la red privada, con IP pública variable.
 - o Configuración en router de acceso necesaria.
 - UPnP
 - o “Pervasive adhoc com.”
 - o Comunicación Dispositivo <-> NAT.