TEMA 2
SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

Fundamentos de Redes

2015/2016

➢Bibliografía Básica:

Capítulo 2 (2.1, 2.2, 2.4, 2.5), James F. Kurose y Keith W. Ross. ***COMPUTER NETWORKING. A TOP-DOWN APPROACH***, 5ª Edición, Addison-Wesley, 2010, ISBN: 9780136079675.

Capítulo 11, Pedro García Teodoro, Jesús Díaz Verdejo y Juan Manuel López Soler. ***TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES***, Ed. Pearson, 2016, ISBN: 978-0-273-76896-8.

➢Para saber más…

Capítulos 7 y 8, James F. Kurose y Keith W. Ross. ***COMPUTER NETWORKING. A TOP-DOWN APPROACH***, 5ª Edición, Addison-Wesley, 2010, ISBN: 9780136079675.

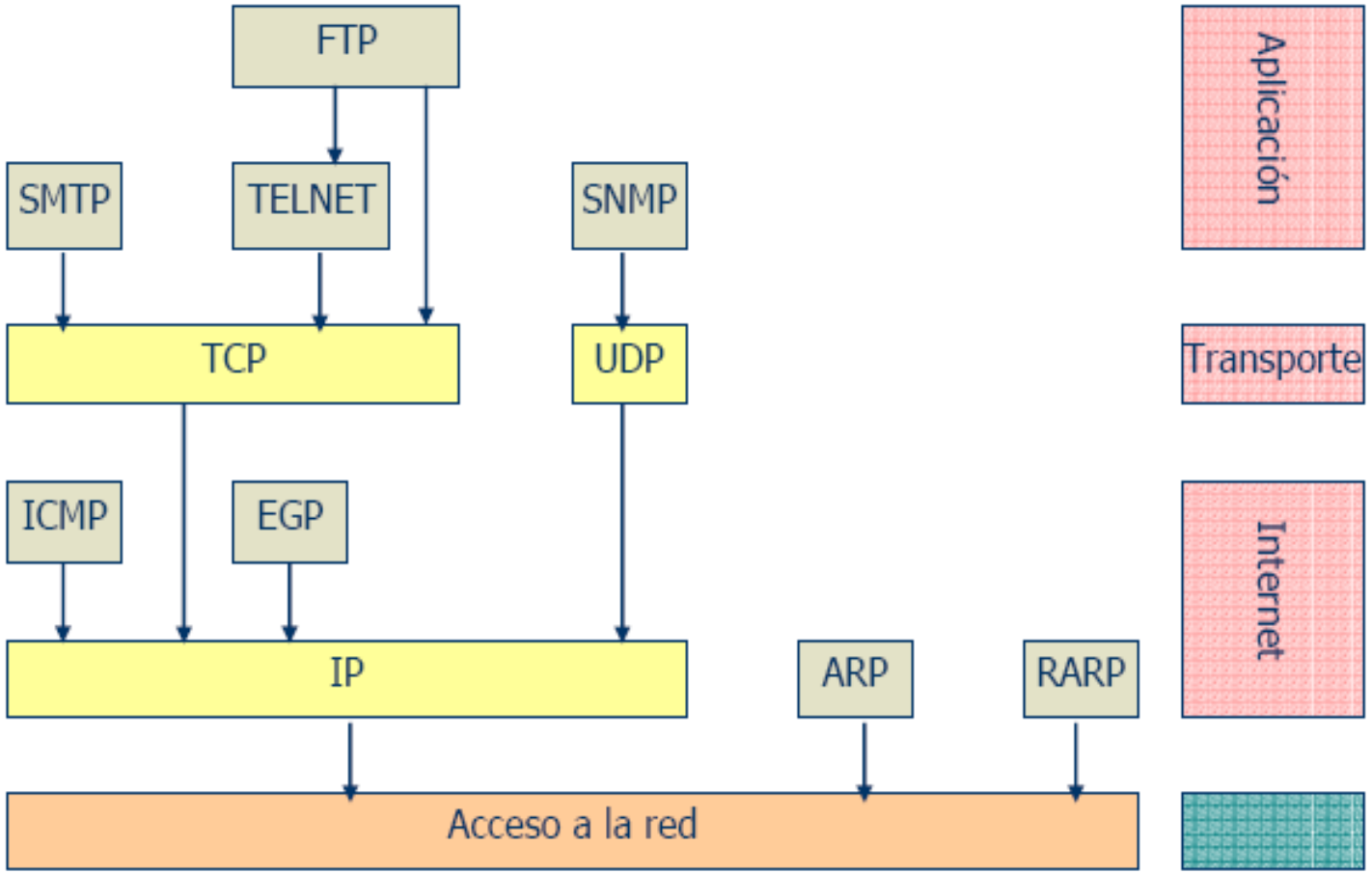
➢ Agradecimientos:

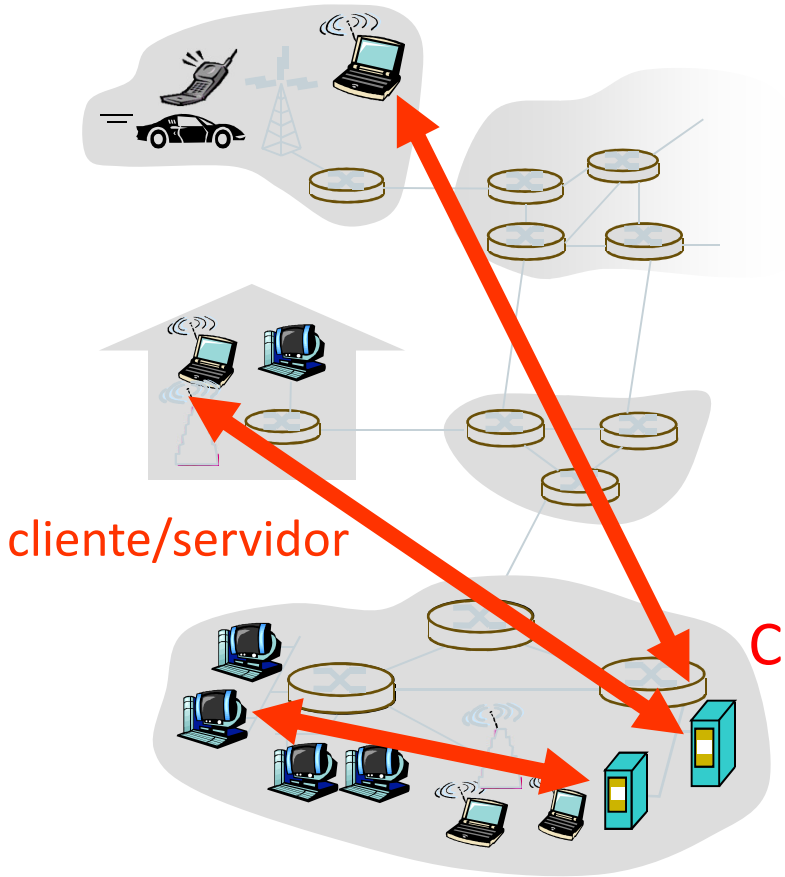
Estas transparencias están inspiradas en las transparencias utilizadas por Kurose y Ross en de la Universidad de Massachusetts.

Tema 2. SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

1. **Introducción a las aplicaciones de red**
2. Servicio de Nombres de Dominio (DNS)
3. La navegación Web
4. El Correo electrónico
5. Protocolos seguros
6. Aplicaciones multimedia
7. Aplicaciones para interconectividad de redes locales
8. Cuestiones y ejercicios

Estructura de protocolos





Servidor:

- Siempre en funcionamiento
- IP permanente & pública
- Agrupados en “granjas”
- http://www.xatakandroid.com/mundo-android/la-imagen-de-la-semana-google-muestra-el-corazon-de-internet
- https://www.youtube.com/watch?v=zRwPSFpLX8I

Clientes:

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

INTRODUCCIÓN A LAS APLICACIONES DE RED: PROCESOS CLIENTE Y SERVIDOR

Proceso Cliente : proceso que inicia la comunicación

Proceso Servidor: proceso que espera a ser contactado

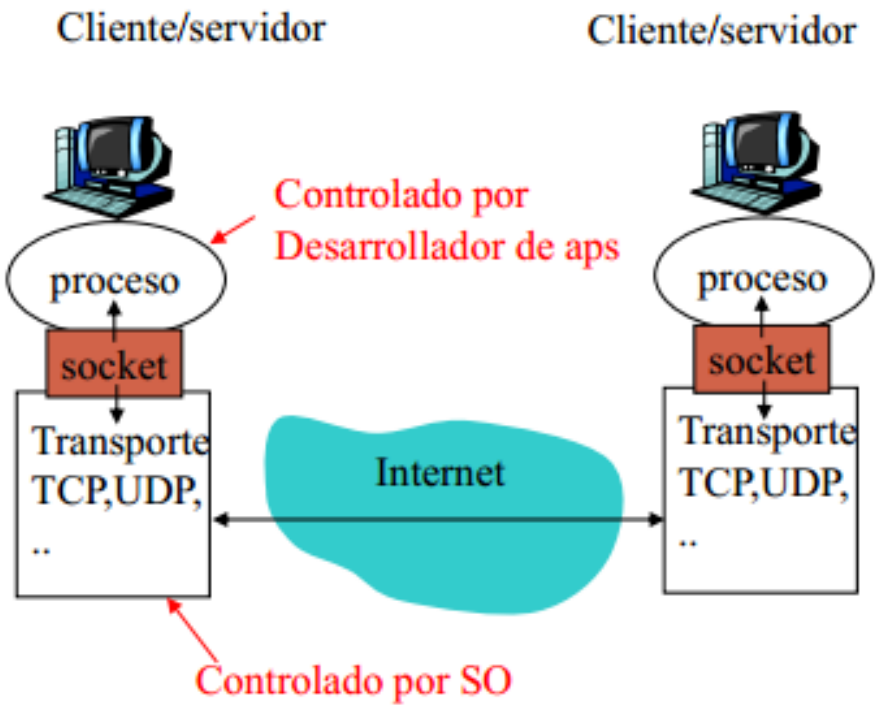
➔IP permanente & pública

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

Ej: servidor web *gaia.cs.umass.edu*:

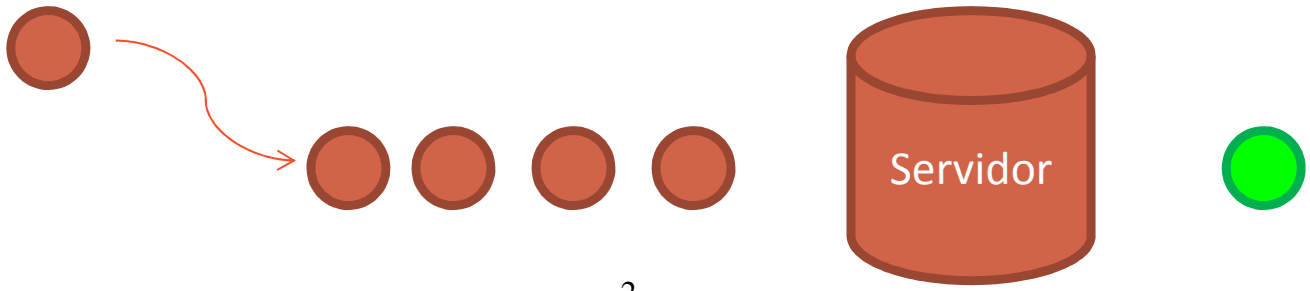
Dirección IP: 128.119.245.12

Número de puerto: 80



INTRODUCCIÓN A LAS APLICACIONES DE RED: 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 (ver ***TRANSMISIÓN DE DATOS Y REDES DE COMPUTADORES***)

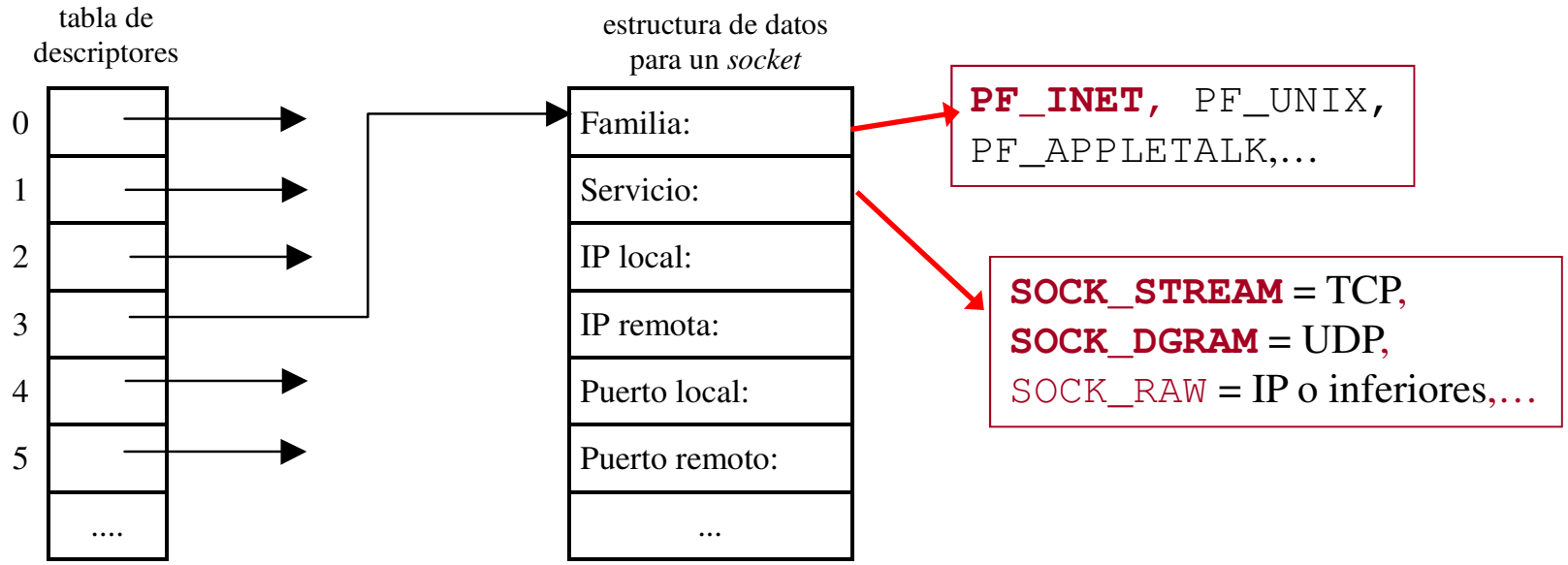


  - El retardo en cola es: $$R = \frac{\lambda \cdot (T_s)^2}{1 - \lambda \cdot T_s}$$

donde Ts (distribución exponencial) es el tiempo de servicio y
λ (Poisson) el ratio de llegada de solicitudes.

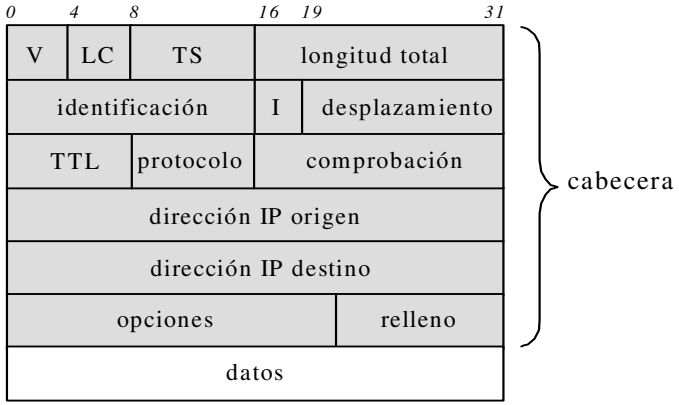
  - Esta misma expersión se puede utilizar para calcular el retardo en cola en un router.

- Definimos **SOCKET** como un descriptor de una transmisión a través del cual la aplicación puede enviar y/o recibir información hacia y/o desde otro proceso de aplicación.
- Es una "puerta" de acceso entre la aplicación y los servicios de transporte.
- En la práctica un socket es un puntero a una estructura:



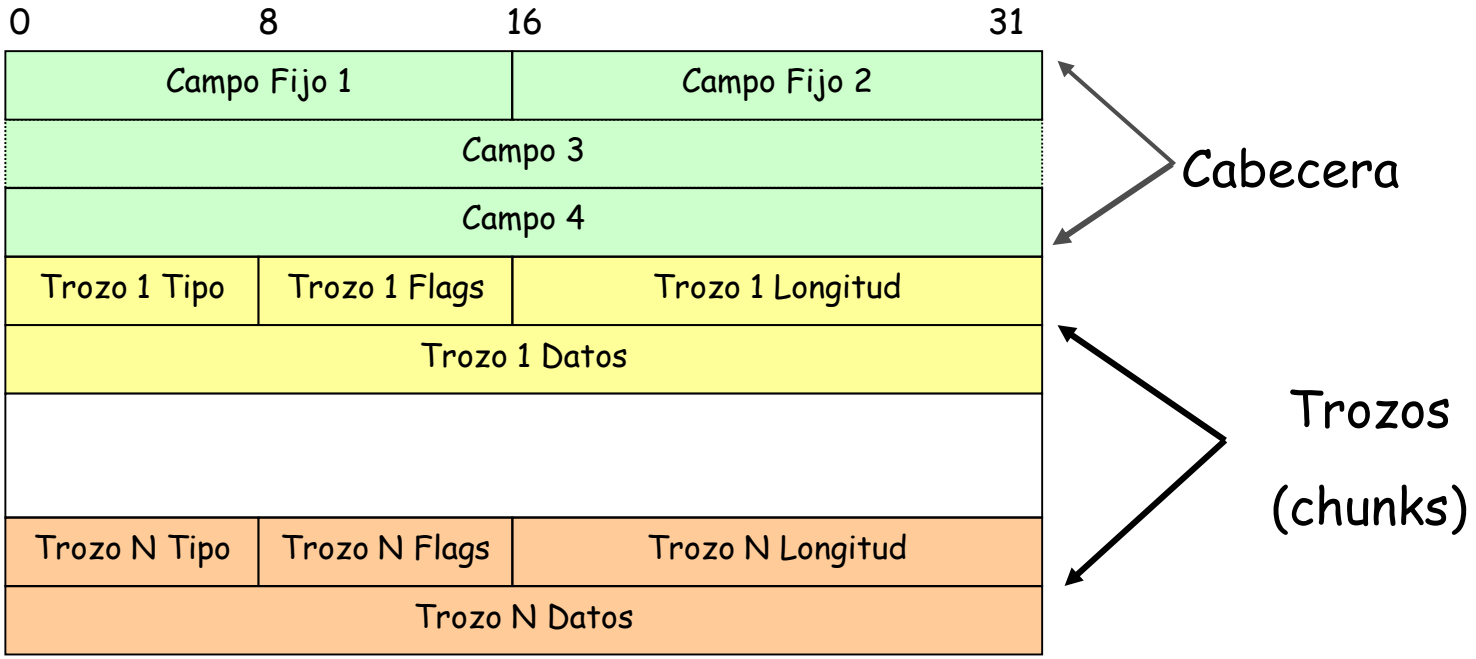
INTRODUCCIÓN A LAS APLICACIONES DE RED: ¿QUÉ DEFINEN LOS PROTOCOLOS DE APLICACIÓN?

- **¿Qué define un protocolo?**
  - **Tipos de servicio**
    - Orientado o no orientado a conexión
    - Realimentado o no
  - **Tipos de mensajes**
    ej., request, response,
  - **Sintaxis:**
    Definición y estructura de"campos" en el mensaje
    En aplicación generalmente son orientados a texto (HTTP)
    Aunque hay excepciones (DNS)
    Tendencia para otras capas : usar formato Type-Length-Value
  - **Semántica:**
    Significado de los "campos"
  - **Reglas:**
    Cuándo los procesos envian mensajes/responden a mensajes

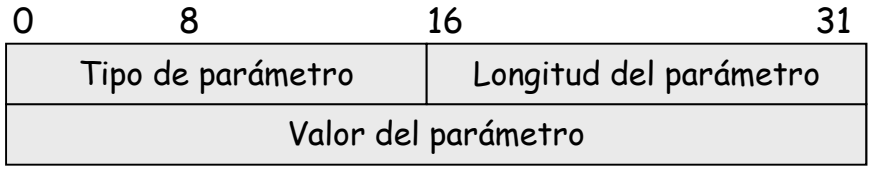


- **Tipos:**
  - **Protocolos de dominio público** ➔ Definidos en RFCs
    ej., HTTP, SMTP
  - **Protocolos propietarios:**
    ej., Skype, IGRP
  - **In-band *versus* out-of-band**
  - **stateless *versus* state-full**
  - **persistentes *versus* no-persistentes**

- **Tendencia: hacer los protocolos flexibles con:**
  - **Una cabecera fija**
  - **Una serie de "trozos" (obligatorios y opcionales)**



- **Tendencia: hacer los protocolos flexibles con:**
  - **Una cabecera fija**
  - **Una serie de "trozos" (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:



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

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

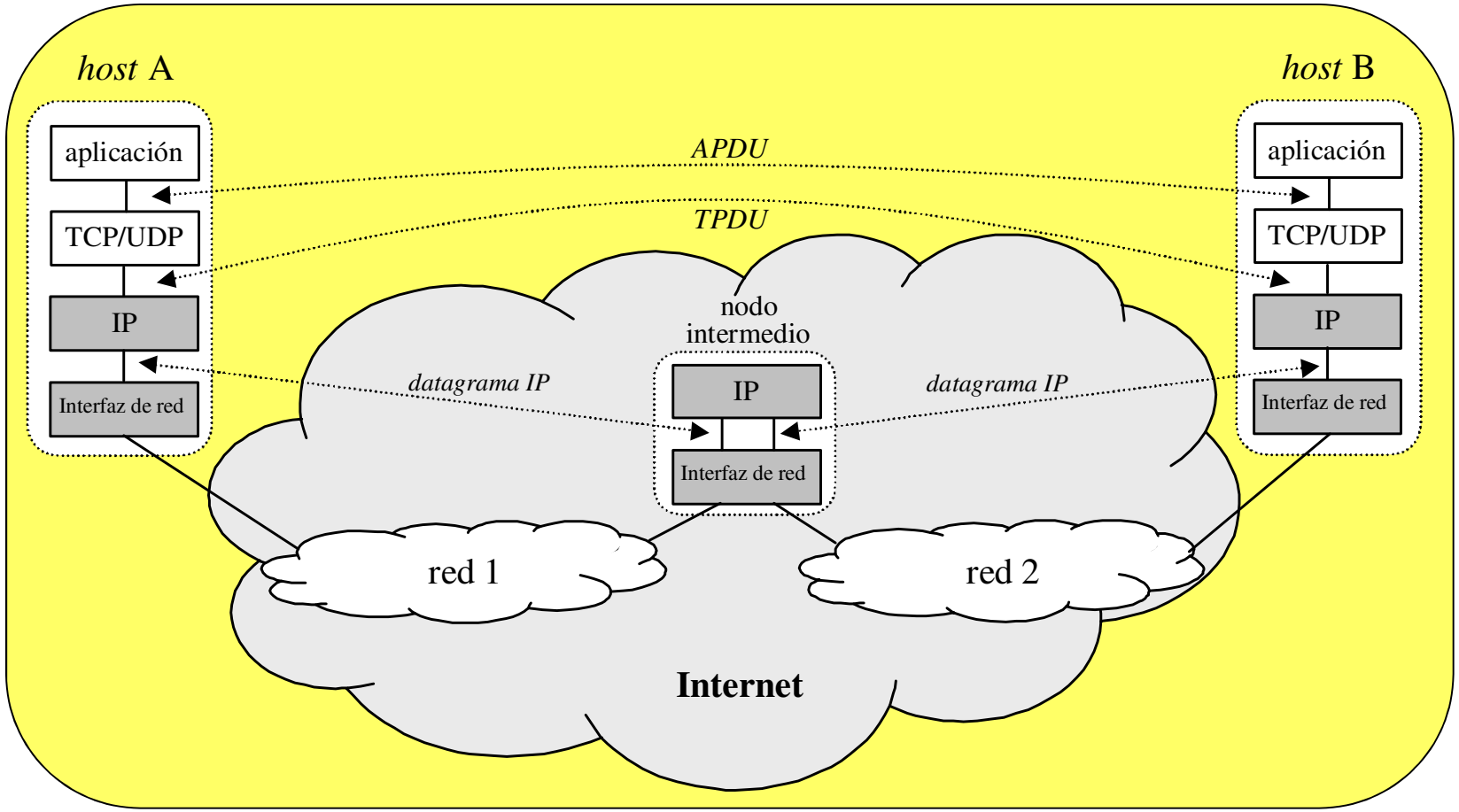
Rendimiento (Throughput)

Algunas aps requieren envío de datos a una tasa determinada

Seguridad

Encriptación, autenticación, no repudio, …

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


host A
host B
aplicación
TCP/UDP
IP
Interfaz de red
APDU
TPDU
datagrama IP
datagrama IP
nodo intermedio
IP
Interfaz de red
aplicación
TCP/UDP
IP
Interfaz de red
red 1
red 2
Internet

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,

¿Para qué existe UDP?

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

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

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

Tema 2. SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

1. Introducción a las aplicaciones de red
2. **Servicio de Nombres de Dominio (DNS)**
3. La navegación Web
4. El Correo electrónico
5. Protocolos seguros
6. Aplicaciones multimedia
7. Aplicaciones para interconectividad de redes locales
8. Cuestiones y ejercicios

SERVICIO DE NOMBRES DE DOMINIO (DNS)

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

150.214.20.3 <-> goliat.ugr.es

- Estructura jerárquica en dominios:

  Parte_local.dominio_niveln. … .dominio_nivel2.dominio_nivel1.

- Al dominio de nivel1 se le denomina dominio genérico.

- El dominio raiz o “.” está gestionado por el **ICANN** (Internet Corporation for Assigned Names and Numbers; http://www.icann.org), que suele delegar en centros regionales.

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

.com -> organizaciones comerciales

.edu -> instituciones educativas, como universidades, de EEUU.

.gov -> instituciones gubernamentales estadounidenses

.mil -> grupos militares de estados unidos

.net -> proveedores de Internet

.org -> organizaciones diversas diferentes de las anteriores

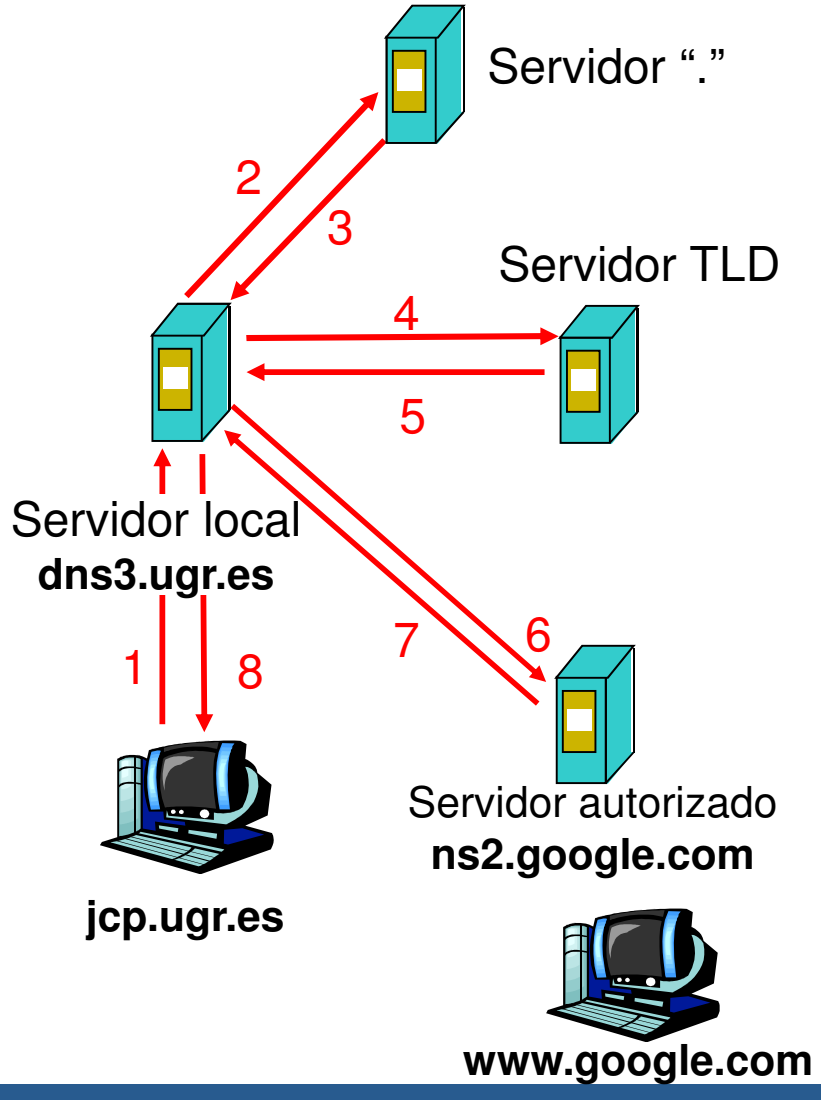
.arpa-> propósitos exclusivos de infraestructura de Internet

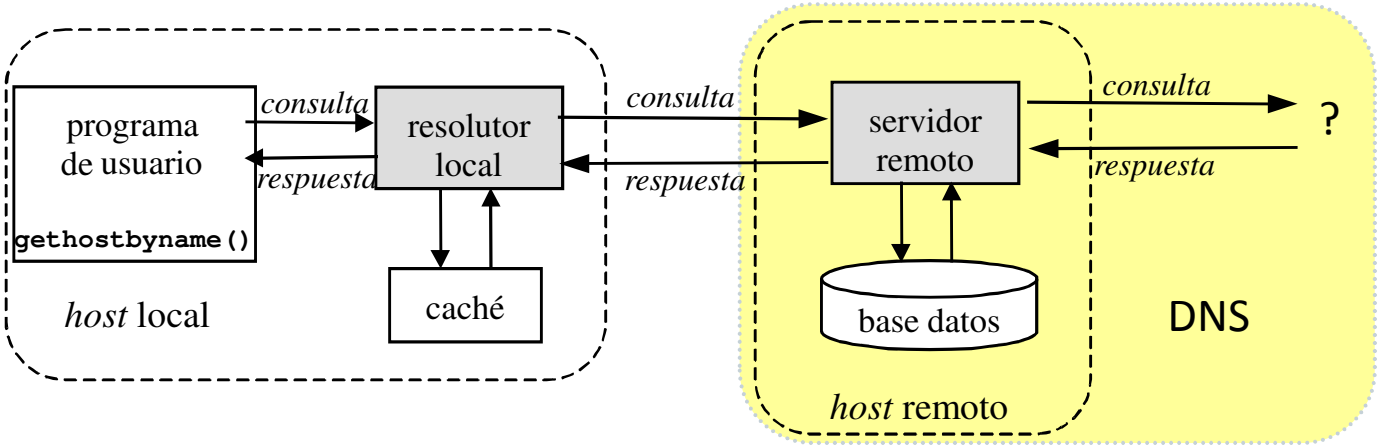
.int -> organizaciones establecidas por tratados internacionales entre gobiernos

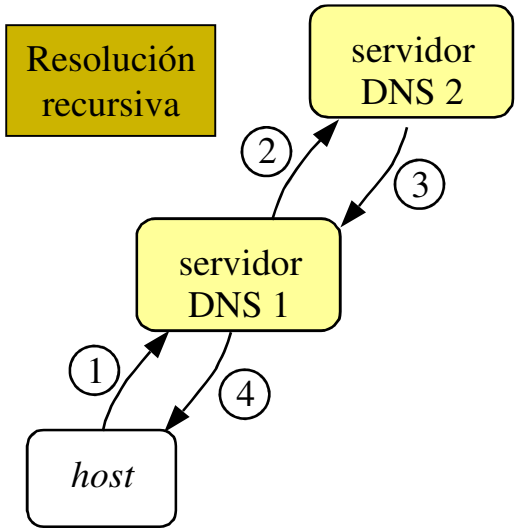
.xy -> indicativos de la zona geográfica (ej. es (España); pt (portugal); jp (Japón)…

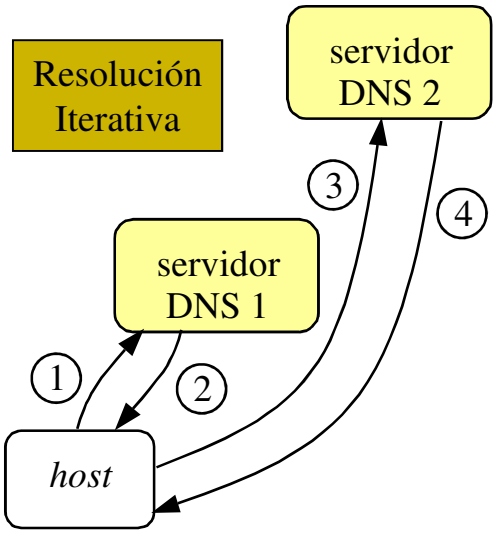
SERVICIO DE NOMBRES DE DOMINIO (DNS)

- DNS es un protocolo para el acceso a una base de datos distribuida con una gestión distribuida:
  - Servidores "."
  - Servidores de dominio (Top-Level domain o TLD)
  - Servidores Locales
- jcp.ugr.es ➔ www.google.com
  i. Consulta al resolver local
  ii. Conexión con DNS local (IP conocida) ¿cómo?
  i. DNS local ➔ IP de destino
  ii. Conexión destino
- Resolución iterativa o recursiva
- Uso de caché



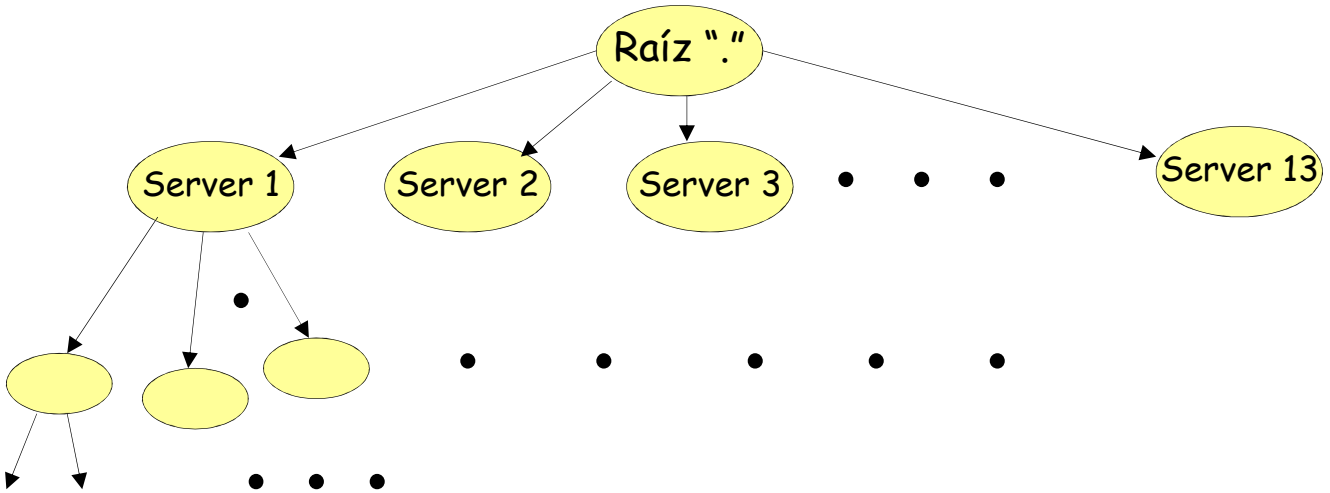

programa
de usuario
gethostbyname()
consulta
respuesta
resolutor
local
caché
host local
consulta
respuesta
servidor
remoto
base datos
host remoto
consulta
?
respuesta
DNS


Resolución
recursiva
servidor
DNS 2
servidor
DNS 1
host
1
2
3
4


Resolución
Iterativa
servidor
DNS 2
servidor
DNS 1
host
1
2
3
4

Gestión de la base de datos distribuida y jerárquica:

- Está formada por un conjunto de servidores cooperativos que almacenan parcialmente la base de datos (**B**erkeley **I**nternet **N**ame **D**omain).
- Cada servidor es responsable de lo que se denomina ZONA.
- Una zona es un conjunto de nombres de dominio contiguos de los que el servidor tiene toda la información y es su autoridad.
- Los servidores autoridad deben contener toda (no "cacheada") la información de su zona.
- La autoridad puede delegarse jerárquicamente a otros servidores



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** raiz (A-M) (ver http://www.root-servers.org)
- El root-server F (y otros) tiene un servidor en Madrid (**Espanix: punto neutro**)
- Cuando un cliente (resolver) solicita una resolución de nombres a su servidor puede ocurrir:
  - 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.

Root-servers http://www.root-servers.org/

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

¿Cómo es la base de datos DNS?

- Todo dominio está asociado al menos a un registro **R**esource **R**ecord.

- Cada **RR** es una tupla con 5 campos:

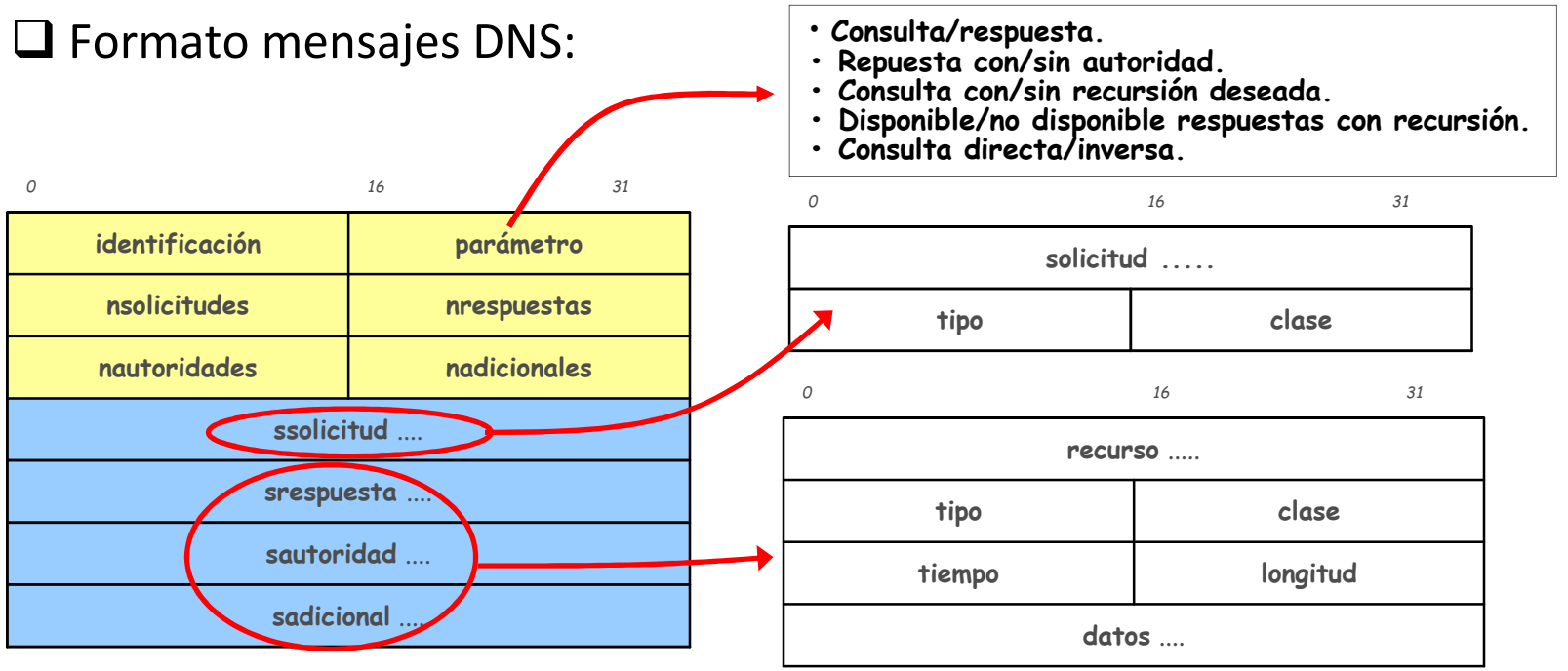
  Nombre del dominio: nombre del dominio al que se refiere el RR.
  Tiempo de vida: tiempo de validez de un registro (para la cache).
  Clase: en Internet siempre IN.
  Tipo: Tipo de registro.

SOA	Registro con la autoridad de la zona.
NS	Registro que contiene un servidor de nombres.
A	Registro que define una dirección IP.
MX	Registro que define un servidor de correo electrónico.
CNAME	Registro que define el nombre canónico de un nombre de dominio.
HINFO	Información del tipo de máquina y sistema operativo.
TXT	Información del dominio.

  Valor: Contenido que depende del campo tipo

- Existe una base de datos asociada de **resolución inversa** para traducir direcciones IP en nombres de dominio. (in-addr.arpa)

❑ Formato mensajes DNS:

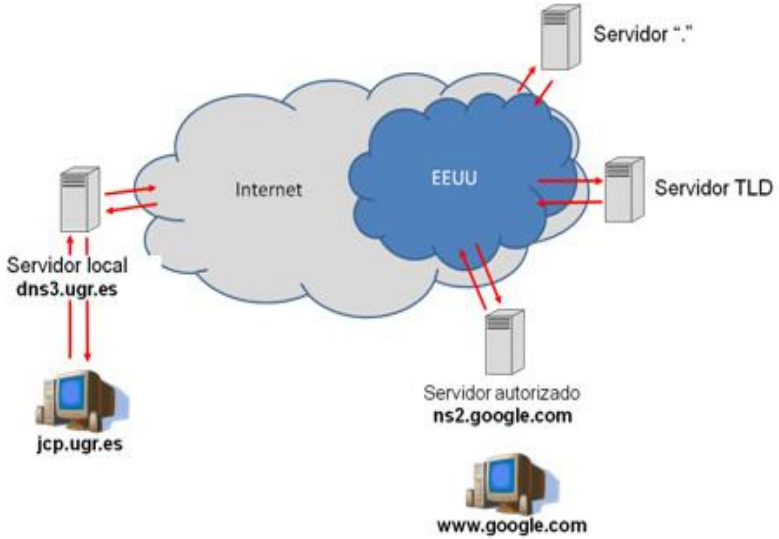


❑ DNS se ofrece en el puerto 53 mediante UDP normalmente o TCP (para respuestas grandes > 512 bytes).

❑ Más información:

- RFC 1034 y RFC 1035 (actualizados 3597 y 3658)
- /usr/doc/HOWTO/trans/es/DNS-COMO
- man named, nslookup, resolver, host.conf, dig
- DNSSEC

6. En la siguiente figura se ilustra un ejemplo de acceso DNS por parte de una máquina (jcp.ugr.es) que quiere acceder a los servicios de www.google.com. Para obtener la dirección IP del servidor, es necesario que la consulta pase por todos los servidores del gráfico. Considerando unos retardos promedio de 8 µs dentro de una red LAN, de 12 ms en cada acceso a través de Internet (4 ms si la conexión se restringe a EEUU) y de 1 ms de procesamiento en cada servidor:



Calcule el tiempo que se tardaría si la solicitud al servidor local es recursiva, pero el propio servidor local realiza solicitudes iterativas.

Especifique una política (recursiva-iterativa) más rápida de solicitudes y el tiempo que tardaría la solicitud en ser respondida. ¿Qué desventaja tiene sobre la solución anterior?

Tema 2. SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

1. Introducción a las aplicaciones de red
2. Servicio de Nombres de Dominio (DNS)
3. **La navegación Web**
4. El Correo electrónico
5. Protocolos seguros
6. Aplicaciones multimedia
7. Aplicaciones para interconectividad de redes locales
8. Cuestiones y ejercicios

- 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

- Protocolo HTTP

  Modelo cliente-servidor

  cliente: *browser* que pide, recibe y muestra objetos web

  server: envia objetos web en respuesta a peticiones

➢ Características HTTP

TCP al puerto 80

Inicio de conexión TCP, envío HTTP, cierre de conexión TCP

HTTP es “stateless” ➔ Cookies

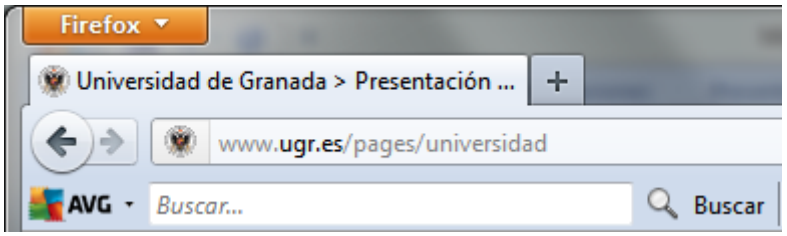
El servidor no mantiene información sobre las peticiones de los clientes

Existen dos tipos

No persistente ➔ Se envia únicamente un objeto en cada conexión TCP.

Persistente ➔ Pueden enviarse multiples objetos sobre una única conexión TCP entre cliente y servidor

LA NAVEGACIÓN WEB: MENSAJES HTTP



1a. Cliente HTTP inicia conexión TCP al servidor HTTP (proceso) en www.ugr.es en puerto 80

1b. Servidor HTTP acepta la conexión y notifica el cliente

2. Cliente HTTP envia *request message* del objeto pages/universidad

3. El servidor HTTP envia el mensaje a través su socket

tiempo

4. Si persistente ➔Envío de más objetos

5. Cierre de conexión TCP

6. Nuevas conexiones TCP

Dos tipos de mensajes HTTP: *request, response*

HTTP request message:

Linea de petición (GET, POST, HEAD)

Lineas de cabecera

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

Carriage return + line feed

(extra carriage return, line feed)

Indican fin del mensaje

Dos tipos de mensajes HTTP: *request, response*

HTTP response message:

Linea de estado

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 …...
Content-Length: 6821
Content-Type: text/html
```

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

Líneas de cabecera

Datos,
ej. fichero html

```
data data data data data ...
```

8. Compare el rendimiento en términos temporales de HTTP persistente y no persistente considerando los siguientes parámetros:

Descarga de una página web con 10 objetos incrustados
Tiempo de Establecimiento de conexión TCP → 5 ms
Tiempo de Cierre de conexión TCP → 5 ms
Tiempo de solicitud HTTP → 2 ms
Tiempo de respuesta HTTP (página web u objeto) → 10 ms

LA NAVEGACIÓN WEB: Protocolo HTTP 1.1 (RFC 2616)

- MÉTODOS
  - **OPTIONS**: solicitud de información sobre las opciones disponibles
  - **GET**: solicitud de un recurso (puede ser condicional)
  - **HEAD**: igual que GET pero el servidor no devuelve el "cuerpo" sólo cabeceras
  - **POST**: solicitar al servidor la aceptación y subordinación a la URI especificada, de la "entidad" (datos) incluida en la solicitud,
  - **PUT**: solicitud de sustituir la URI especificada con los "datos" incluidos en la solicitud.
  - **DELETE**: solicitud de borrar la URI especificada.
- CÓDIGOS DE RESPUESTA
  - **1xx** indican mensajes exclusivamente informativos
  - **2xx** indican algún tipo de éxito
  - **3xx** redireccion al cliente a otra URL
  - **4xx** indican un error
  - **5xx** indican un error
- CABECERAS (hasta 46 definiciones en HTTP 1.1)

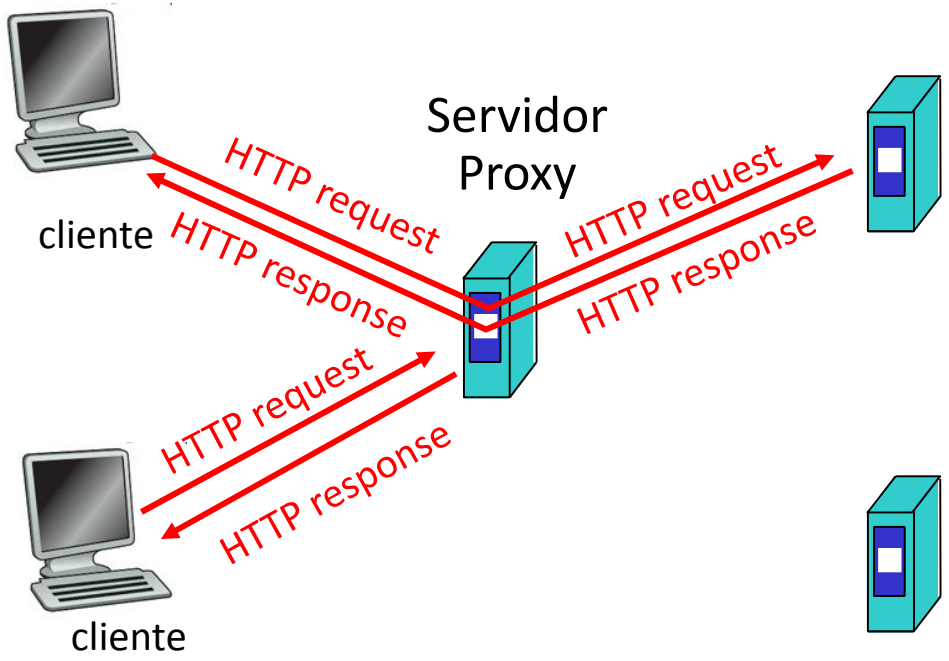
  `From: , User-Agent:, Content-Type:, Content-Length:,……`

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

Web cache

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

- Usuario configura el browser: Acceso Web vía cache
- browser envía todos los requerimientos HTTP al cache
  - Si objeto está en cache: cache retorna objeto
  - Sino cache requiere los objetos desde el servidor Web, y retorna el objeto al cliente



- Ejemplo de respuesta

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: "3e86-410-3596fbbc“
Content-Length: 1040
Content-Type: text/html

Web cache

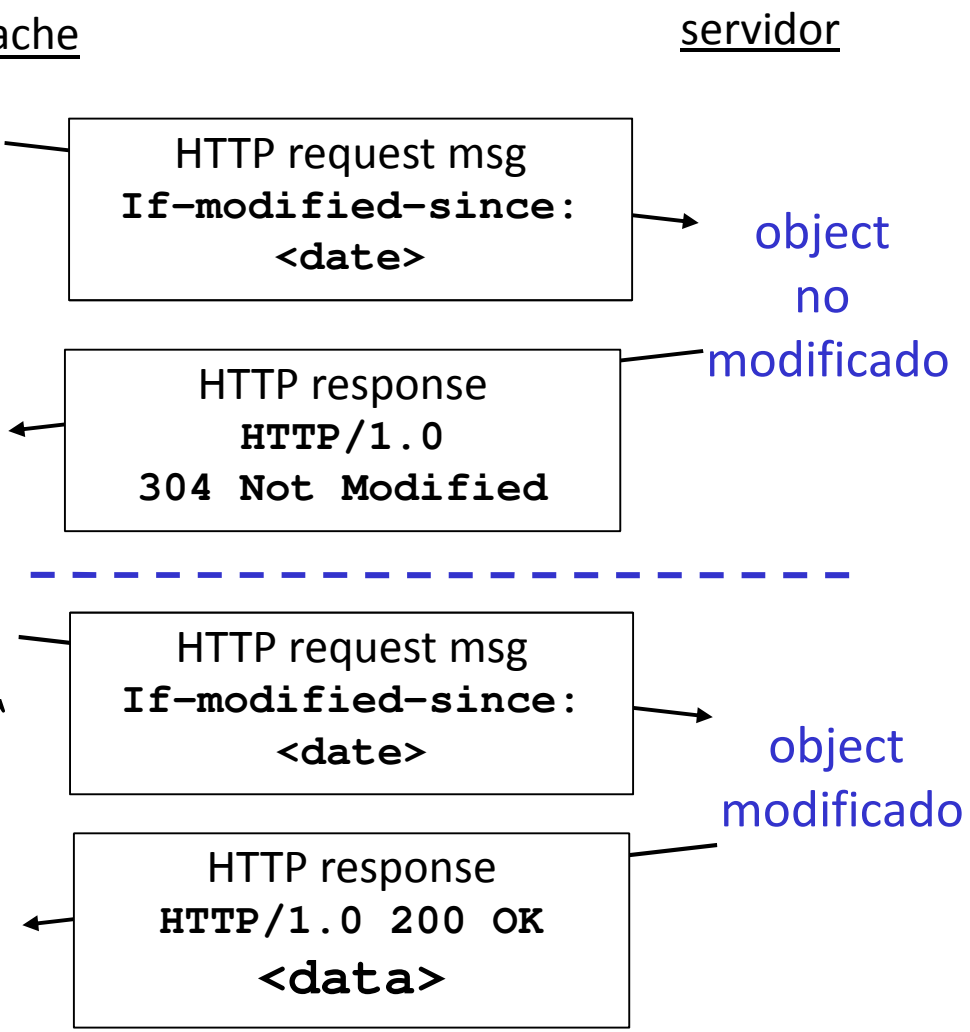
- Objetivo: no enviar objetos si el cache tiene la versión actualizada
- Cache: especifica la fecha de la copia en el requerimiento HTTP

```
If-modified-since:
  <date>
```

```
If-None-Match:
  "686897696a7c876b7e"
```

- servidor: responde sin el objeto si la copia de la cache es la última. :

```
HTTP/1.0 304 Not
  Modified
```



Tema 2. SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

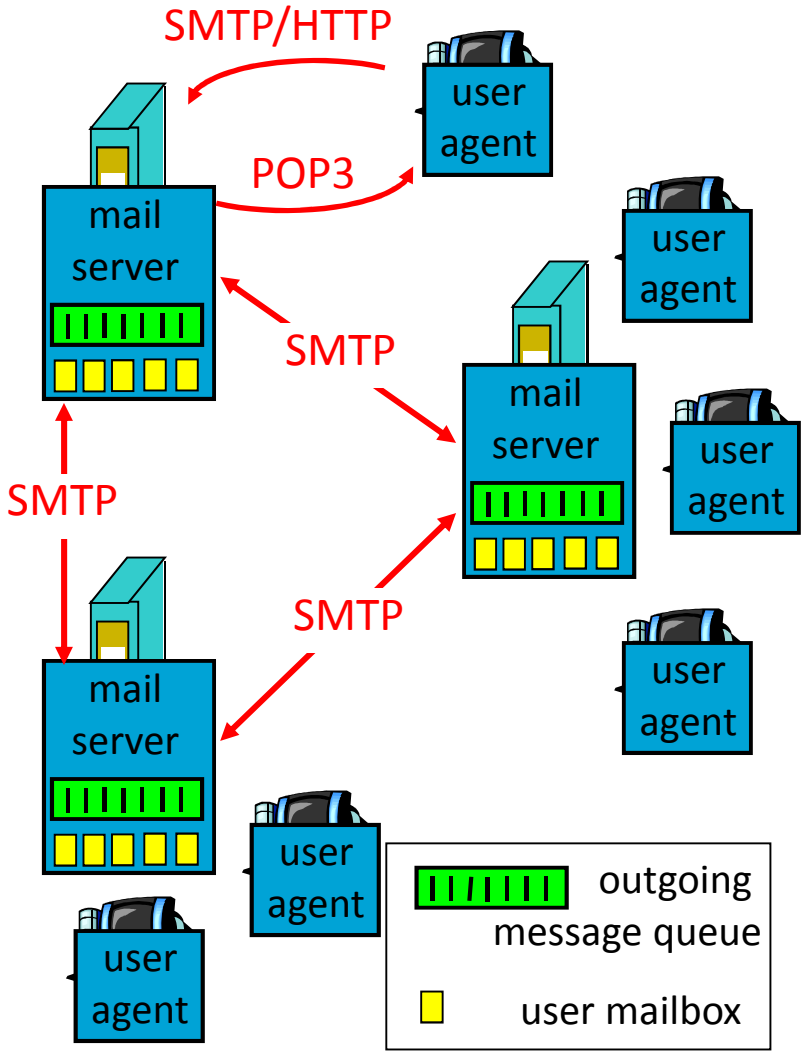
1. Introducción a las aplicaciones de red
2. Servicio de Nombres de Dominio (DNS)
3. La navegación Web
4. **El Correo electrónico**
5. Protocolos seguros
6. Aplicaciones multimedia
7. Aplicaciones para interconectividad de redes locales
8. Cuestiones y ejercicios

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
  - Procolos de descarga:
    POP3, IMAP, HTTP

- Agente de usuario
  - Componer, Editar y Leer correos mensajes de correo
    Ej. Outlook, Thunderbird

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



EL CORREO ELECTRÓNICO: SMTP (RFC 2821)

- SMTP se implementa mediante dos programas (incluidos ambos en cada mail server):
  - Cliente SMTP: se ejecuta en el mail server que está enviando correo
  - Servidor SMTP: se ejecuta en el mail server que está recibiendo correo
  - `"sendmail"` http://en.wikipedia.org/wiki/Sendmail

- Usa TCP en el puerto 25

- Tres fases
  - *Handshaking* ("saludo")
  - Transferencia de mensajes
  - Cierre

- La interacción entre cliente SMTP y servidor SMTP se realiza mediante comandos/respuesta
  - comandos: texto ASCII
  - respuestas: código de estado y frases

- Los mensajes deben estar codificados en ASCII de 7 bits!! ➔ Extensiones MIME

EL CORREO ELECTRÓNICO: SMTP (RFC 2821)

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

EL CORREO ELECTRÓNICO: 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

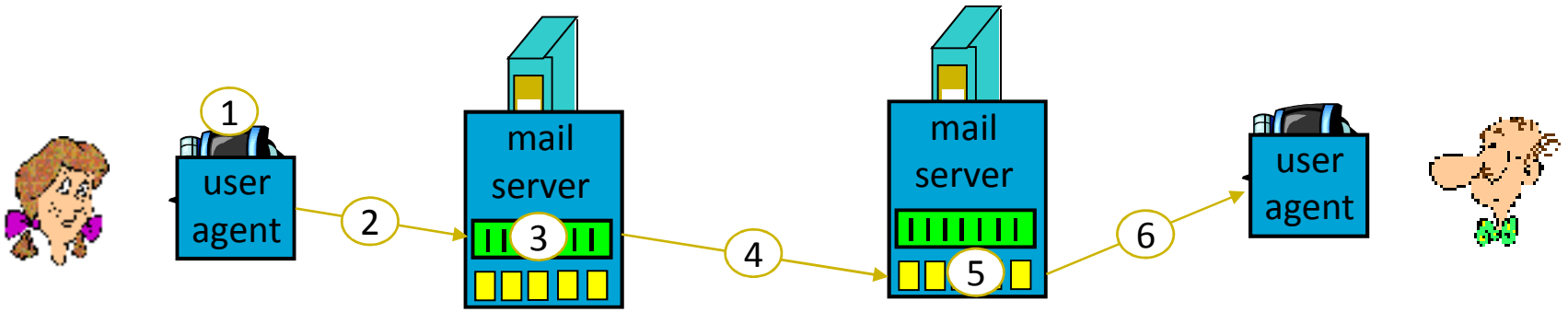
2) Se envia con SMTP o HTTP el mensaje al servidor de correo del usuario origen que lo sitúa en la cola de mensajes salientes

3) El cliente SMTP abre una conexión TCP con el servidor de correo del usuario destino

4) El cliente SMTP envia el mensaje sobre la conexión TCP

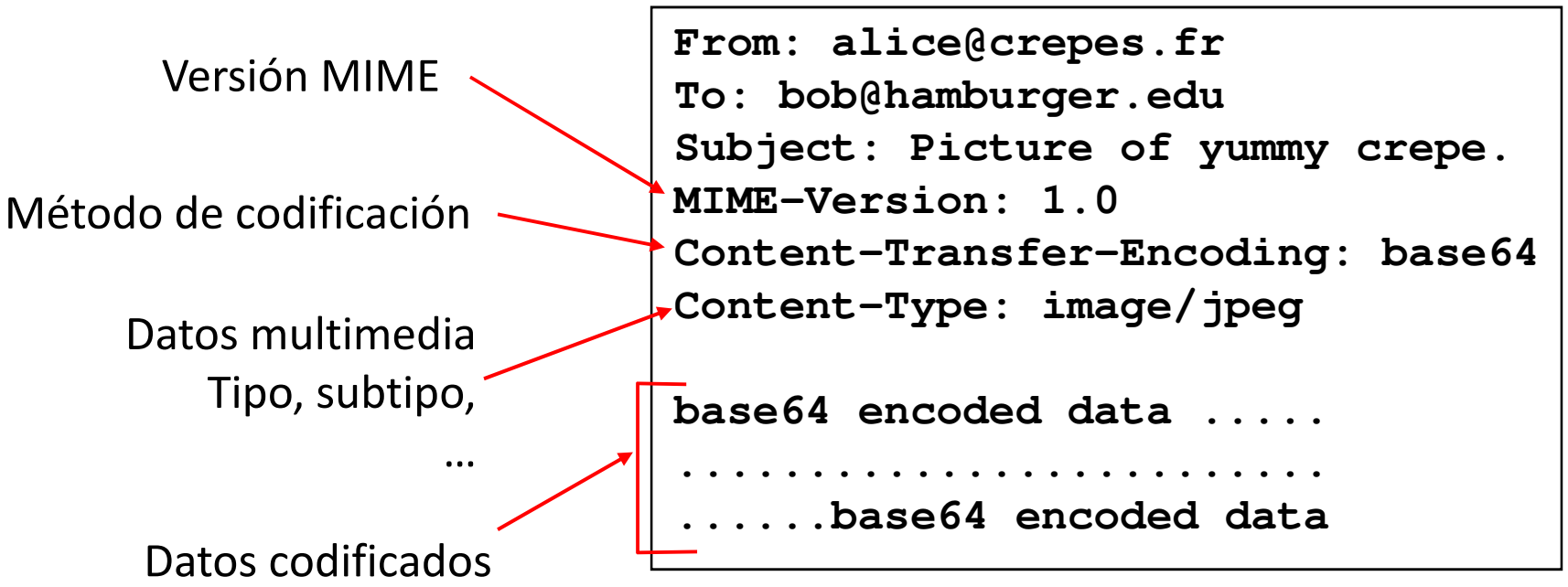
5) El servidor de correo del usuario destino ubica el mensaje en el mailbox del usuario destino

6) El usuario destino invoca su Agente de Usuario para leer el mensaje utilizando POP3, IMAP o HTTP



EL CORREO ELECTRÓNICO: EXTENSION MIME

- MIME: multimedia mail extension, RFC 2045, 2056



EL CORREO ELECTRÓNICO: PROTOCOLOS DE ACCESO AL CORREO

Ej: POP3 PROTOCOL PORT = 110

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

- Listado de puertos relacionados con e-mail:

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

Tema 2. SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

1. Introducción a las aplicaciones de red
2. Servicio de Nombres de Dominio (DNS)
3. La navegación Web
4. El Correo electrónico
5. **Seguridad & protocolos seguros**
6. Aplicaciones multimedia
7. Aplicaciones para interconectividad de redes locales
8. Cuestiones y ejercicios

➢ Primitivas de seguridad

- **Confidencialidad**
  - Sólo accede a la información quien debe hacerlo.

- **Responsabilidad**
  - 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.

- **Integridad**
  - La información no ha sido manipulada.

- **Disponibilidad**
  - Acceso a los servicios

➢ Mecanismos de Seguridad

- **Cifrado Simétrico:** C = K(P) & P = K(C)
  - DES, 3DES, AES, RC4

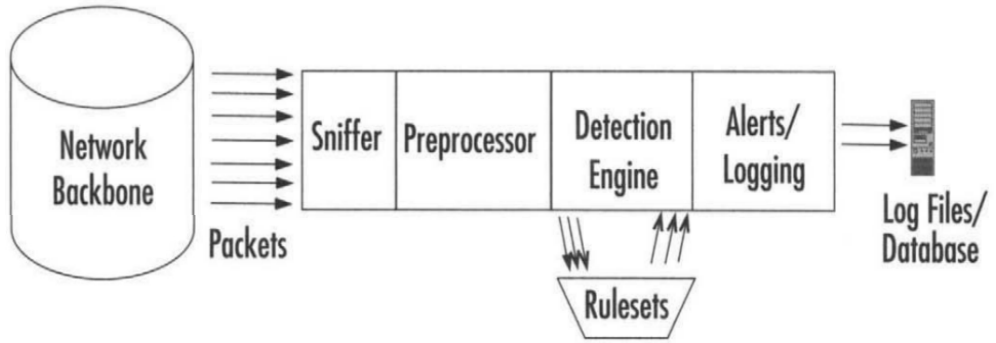
- **Cifrado Asimétrico:** C = K^{+}(P) & P = K^{-}(C)
  - Diffie & Hellman, RSA

- **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 Perimetral:
    - *Firewalls,* sistemas de detección de intrusiones (IDS) y de respuesta (IRS)



PROTOCOLOS SEGUROS

- Seguridad:
  - Seguridad Perimetral:
    - *Firewalls,* sistemas de detección de intrusiones (IDS) y de respuesta (IRS)
  - 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, SSL-POP, VPN
      - Transport Secure Layer (TSL)

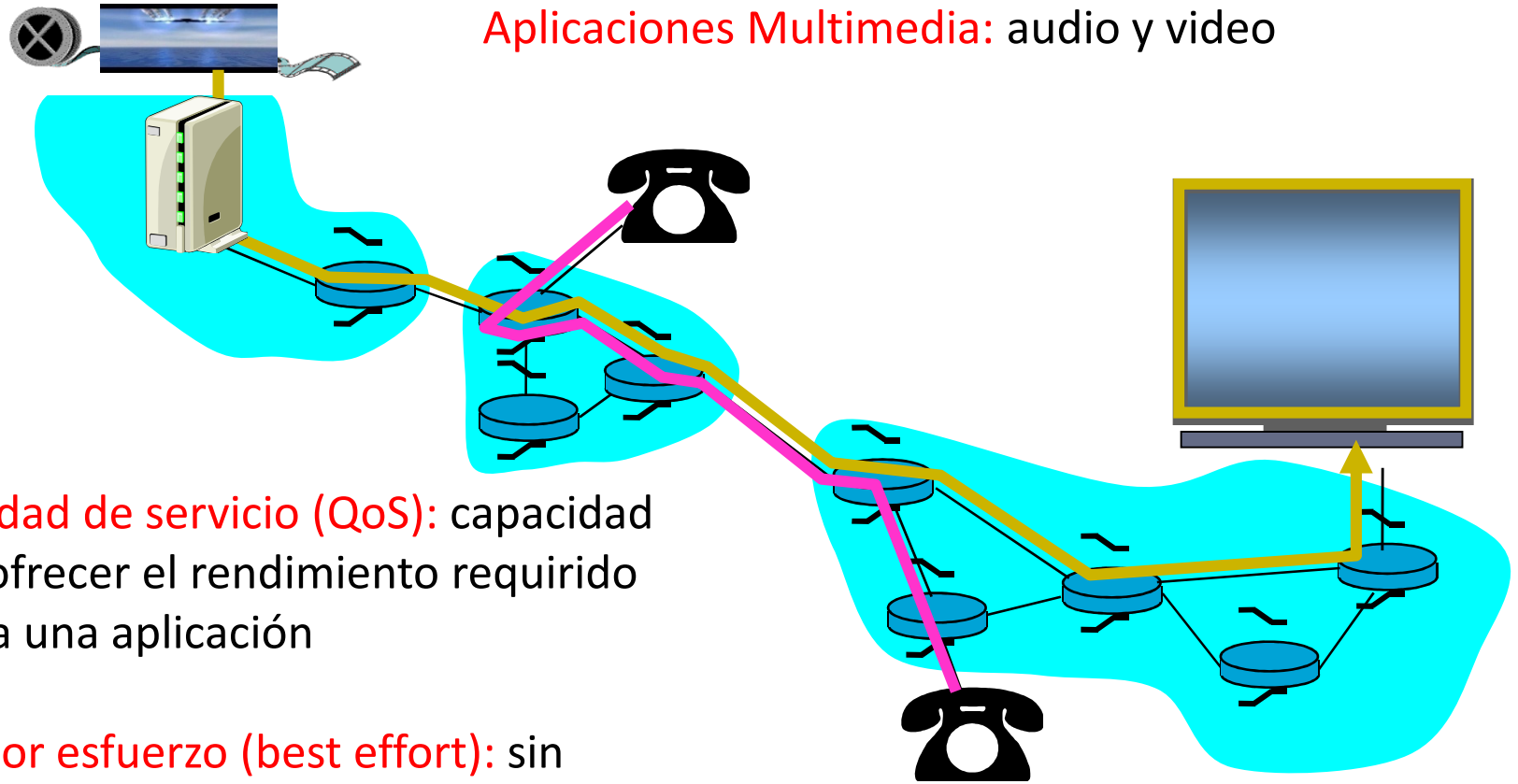
        http://heartbleed.com/

    - Capa de Red ➔ IPSec (VPN)

Tema 2. SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

1. Introducción a las aplicaciones de red
2. Servicio de Nombres de Dominio (DNS)
3. La navegación Web
4. El Correo electrónico
5. Protocolos seguros
6. **Aplicaciones multimedia**
7. Aplicaciones para interconectividad de redes locales
8. Cuestiones y ejercicios

➢ Conceptos

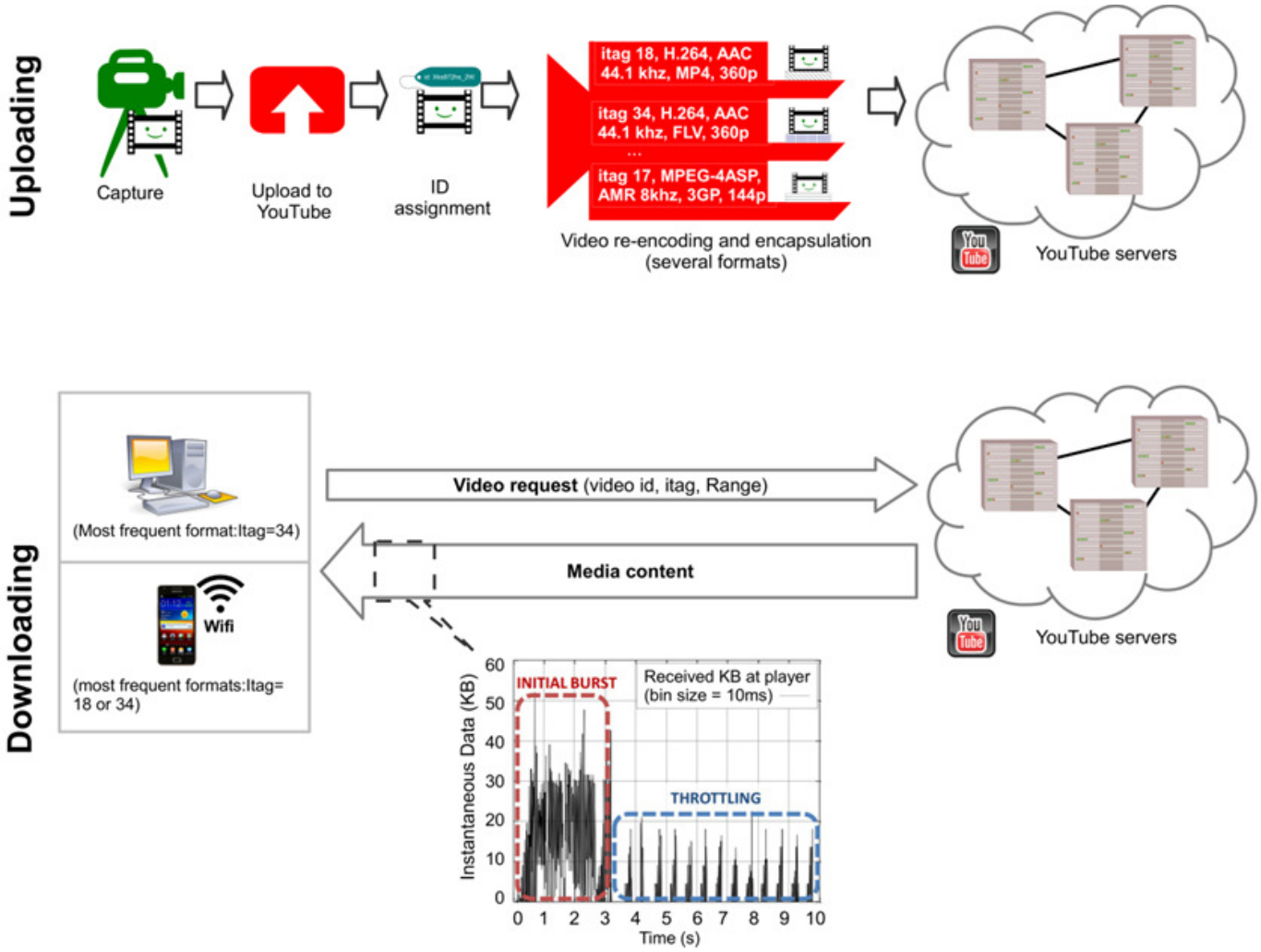


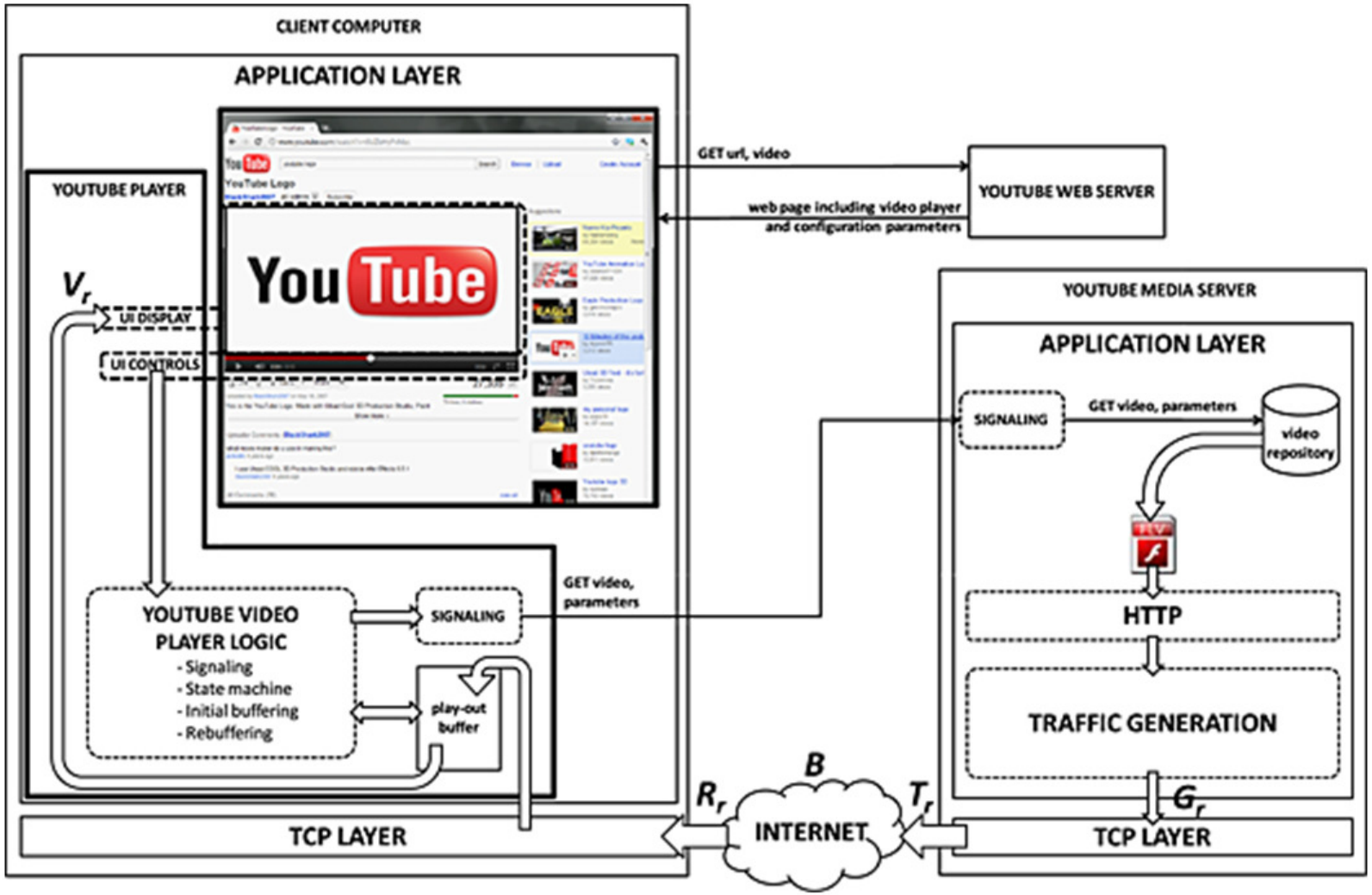
Aplicaciones Multimedia: audio y video

Calidad de servicio (QoS): capacidad de ofrecer el rendimiento requirido para una aplicación

Mejor esfuerzo (best effort): sin garantía de QoS

- **Tipos de aplicaciones**
  - Flujo de audio y video (streaming) almacenado ➔ Ej YouTube
  - Flujo de audio y vide en vivo ➔ Ej. emisoras de radio o IPTV
  - Audio y vídeo interativo ➔ Ej. Skype
- **Características fundamentales**
  - Elevado ancho de banda
  - Tolerantes a la pérdida de datos
  - Delay acotado
  - Jitter acotado
  - Uso de multicast


Uploading
Capture
Upload to YouTube
ID assignment
itag 18, H.264, AAC 44.1 khz, MP4, 360p
itag 34, H.264, AAC 44.1 khz, FLV, 360p
itag 17, MPEG-4ASP, AMR 8khz, 3GP, 144p
Video re-encoding and encapsulation (several formats)
YouTube servers
Downloading
(Most frequent format:Itag=34)
Wifi
(most frequent formats:Itag= 18 or 34)
Video request (video id, itag, Range)
Media content
YouTube servers
Received KB at player (bin size = 10ms)
INITIAL BURST
THROTTLING
Instantaneous Data (KB)
Time (s)


CLIENT COMPUTER
APPLICATION LAYER
YOUTUBE PLAYER
YouTube Logo
V_r
UI DISPLAY
UI CONTROLS
YOUTUBE VIDEO PLAYER LOGIC
- Signaling
- State machine
- Initial buffering
- Rebuffering
SIGNALING
play-out buffer
GET video, parameters
TCP LAYER
GET url, video
web page including video player and configuration parameters
YOUTUBE WEB SERVER
YOUTUBE MEDIA SERVER
APPLICATION LAYER
SIGNALING
GET video, parameters
video repository
HTTP
TRAFFIC GENERATION
G_r
TCP LAYER
B
R_r
INTERNET
T_r

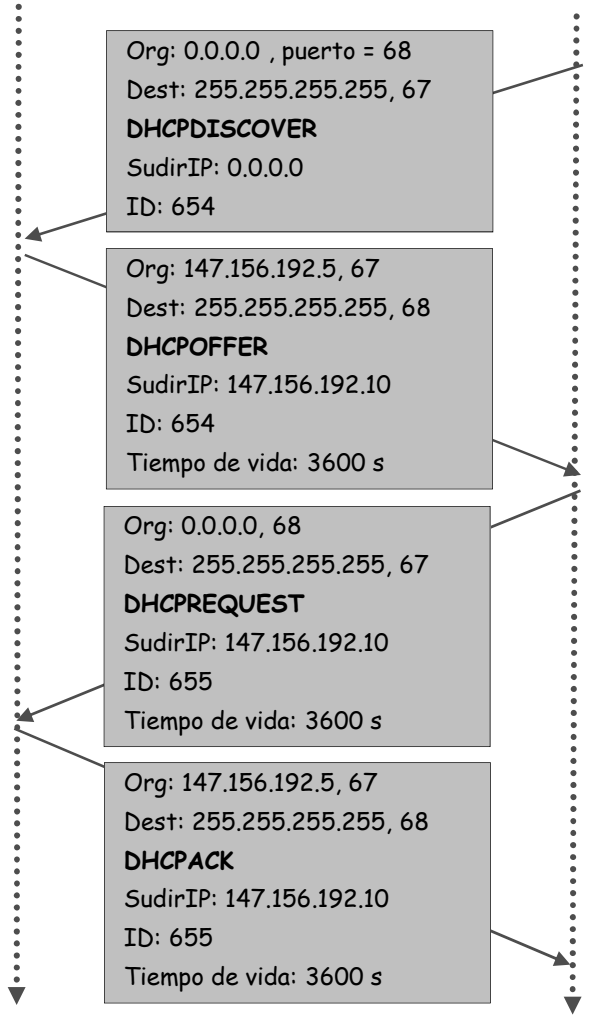
Tema 2. SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

1. Introducción a las aplicaciones de red
2. Servicio de Nombres de Dominio (DNS)
3. La navegación Web
4. El Correo electrónico
5. Protocolos seguros
6. Aplicaciones multimedia
7. **Aplicaciones para interconectividad de redes locales**
8. Cuestiones y ejercicios

DHCP (Dynamic Host Configuration Protocol)

Servidor DHCP
147.156.192.5

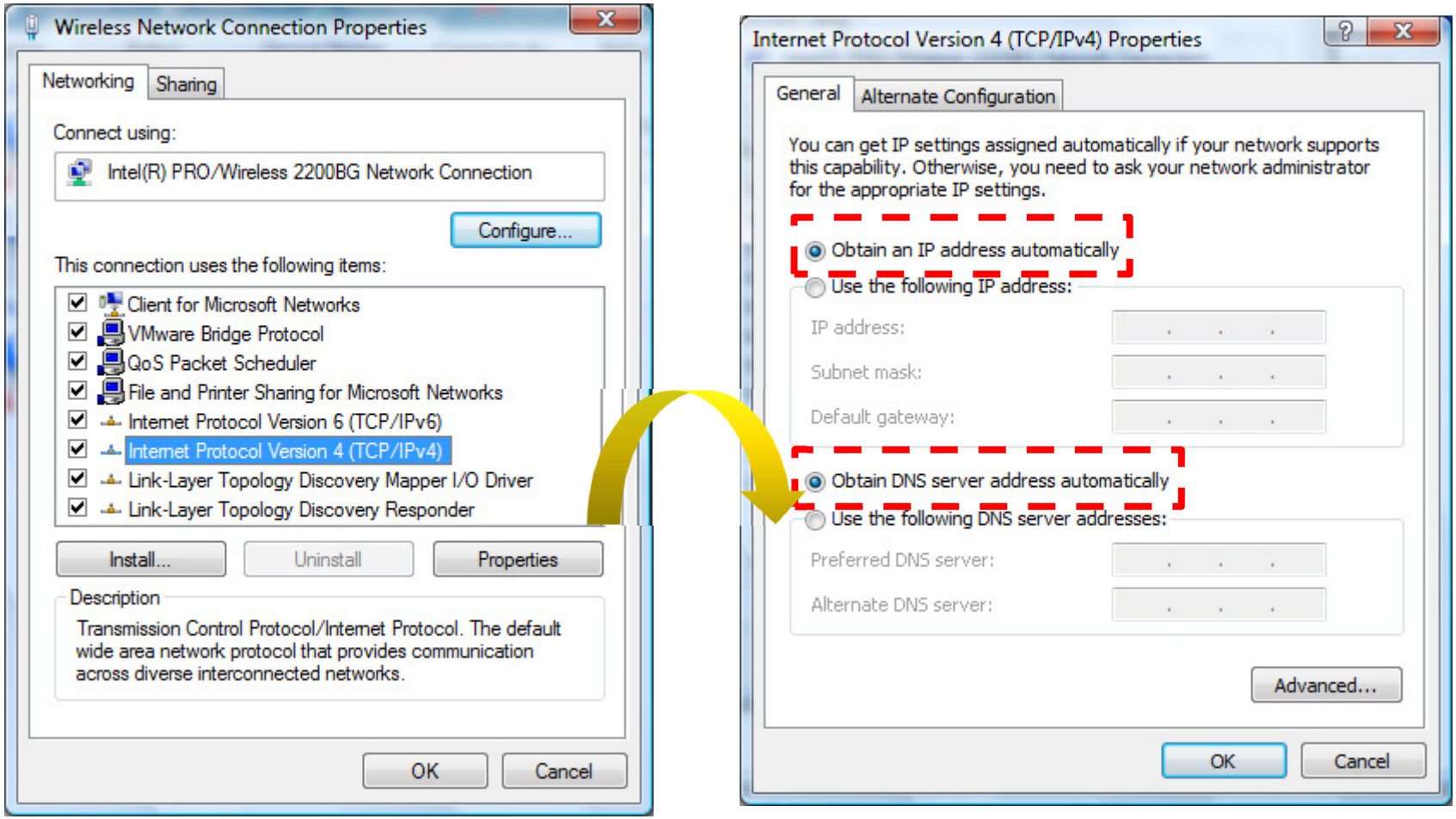
Cliente DHCP
IP: ?



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

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

Configuración de un cliente MS Windows:



Configuración de un cliente Linux (Fedora Core dist.):

```
# Sample /etc/sysconfig/network-scripts/ifcfg-eth0 :

DEVICE=eth0
BOOTPROTO=dhcp
HWADDR=00:0C:29:CE:63:E3
ONBOOT=yes
TYPE=Ethernet
```

Configuración de un servidor de Linux (*dhcpd*):

```
# Sample /etc/dhcpd.conf

default-lease-time 600;max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "mydomain.org";
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.10 192.168.1.100;
  range 192.168.1.150 192.168.1.200;
}

# Static IP address assignment
host haagen {
  hardware ethernet 08:00:2b:4c:59:23;
  fixed-address 192.168.1.222;
}
```

Tema 2. SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

1. Introducción a las aplicaciones de red
2. Servicio de Nombres de Dominio (DNS)
3. La navegación Web
4. El Correo electrónico
5. Protocolos seguros
6. Aplicaciones multimedia
7. Aplicaciones para interconectividad de redes locales
8. **Cuestiones y ejercicios**

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

La telefonía móvil
WhatsApp
YouTube
Spotify
Comercio electrónico

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

a. **Calcule la velocidad de transmisión requerida. ¿Es la velocidad del enlace de acceso suficiente?**
b. **¿y si se dobla la velocidad del enlace? ¿cuál sería el tiempo de cola que esperaría en promedio cada solicitud en el enlace descendente antes de ser enviada? Considere que cada registro se envía por separado, con una cabecera de tamaño despreciable**
c. **Si, alternativamente, se diseña una caché que permite evitar un 70% de los accesos a la BD ¿cuál sería el tiempo de cola que esperaría en promedio cada solicitud en el enlace descendente? ¿qué solución es mejor, la b. o esta?**

CUESTIONES Y EJERCICIOS

1. Explicar por qué cuando solicitamos http://www.google.com desde nuestro navegador, se muestra la URL servida desde (www.google.es)

   ¿qué relación tienen esos 2 nombres de dominio?
   ¿guarda google información sobre nuestra localización?¿cómo se obtiene?
   ¿qué herramientas e información se necesita?
   ¿qué ocurre y cómo influye si configuro en mi navegador como lenguaje preferido "francés"?
   ¿pueden servirse páginas dependiendo de nuestra localización? ¿en su caso, con qué precisión?

   Sugerencia: Usar el analizador http://www.wireshark.org para mostrar trazas

TEMA 2
SERVICIOS Y PROTOCOLOS DE APLICACIÓN EN INTERNET

Fundamentos de Redes

2015/2016