Práctica 01

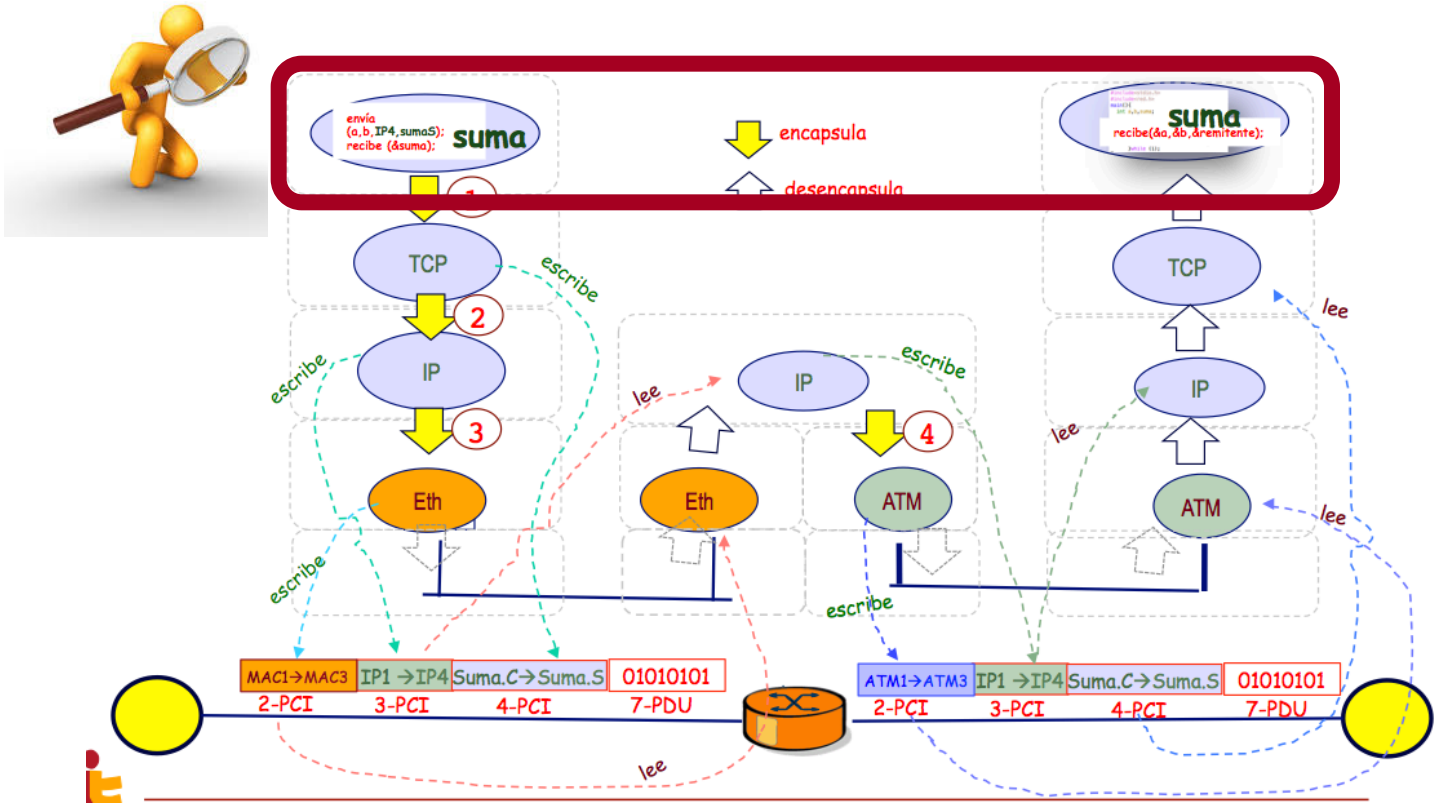
Aplicaciones Web y DNS

Antonio J. Estepa Alonso

Departamento de
Ingeniería Telemática

Índice

1. WireShark
2. Introducción a la Aplicación Web
3. Introducción a la Aplicación DNS

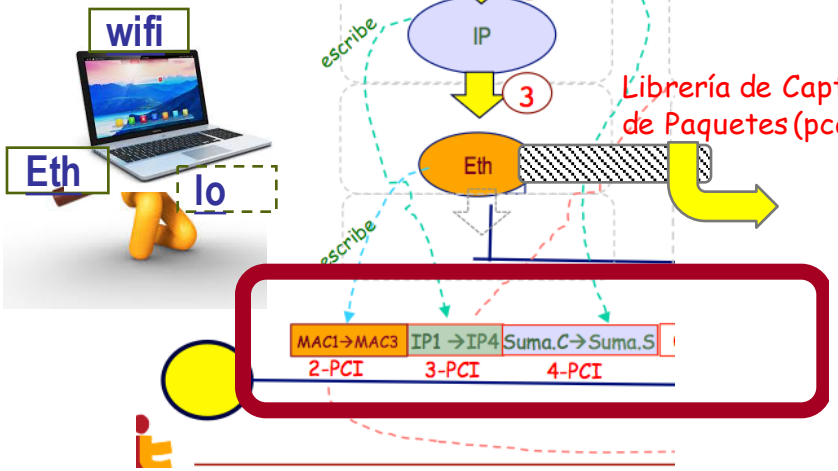


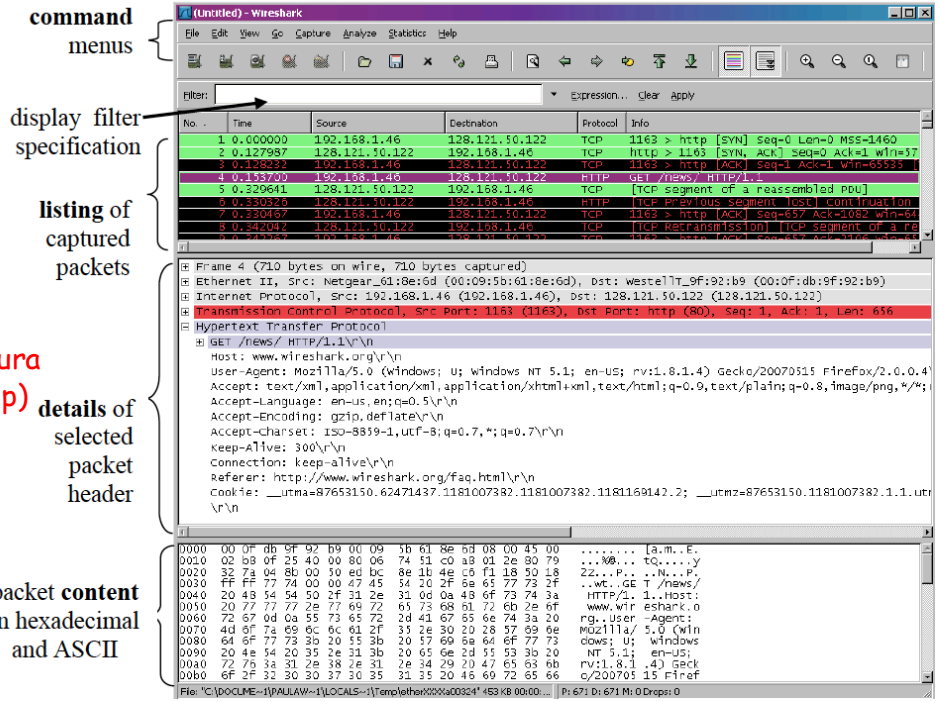
Wireshark

- **Analizador de Protocolos Gratuito**
  - **Representa e interpreta todas las cabeceras que viajan encapsuladas en las tramas que lee la NIC de tu equipo.**
  - **Nosotros nos centraremos en las A-PDUs aunque el programa muestra las cabeceras de todos los niveles**

Recuerda:

Un portátil suele tener al menos 3 interfaces diferentes





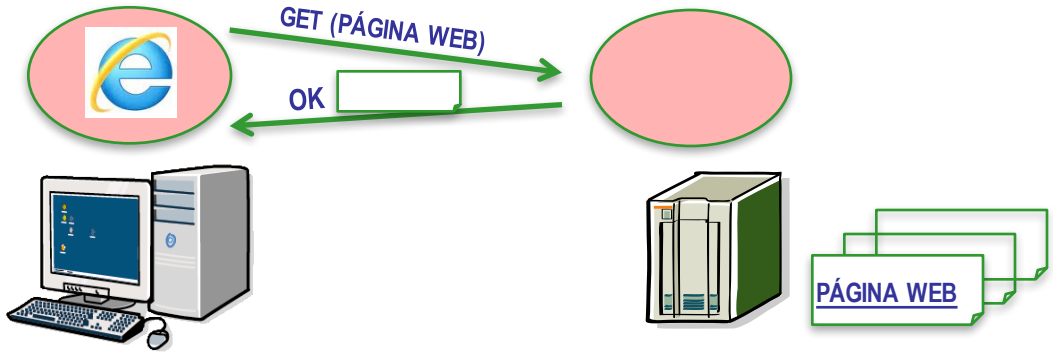
Wireshark

- ¿Cómo puedo aprender a manejarlo?
  - Haz la Práctica 0 del libro de referencia (wiresharklab)
  - Mira la ayuda del programa
  - Consulta tutoriales en Internet
    - Nivel Básico:
      - https://www.youtube.com/watch?v=TkCSr30UojM
      - https://cs.gmu.edu/~astavrou/courses/ISA_674_F12/Wireshark-Tutorial.pdf
    - Nivel Medio:
      - http://www-scf.usc.edu/~csci571/Special/Tutorials/wireshark_html/wireshark.html

- ¿Cómo se que he aprendido lo que necesito?
  - Porque siempre eres capaz de capturar los paquetes en el interfaz de red (NIC) que quieres y no necesitas descargarte las capturas ya hechas por el profesor
  - Porque sabes encontrar la información que buscas.

Aplicaciones Web y DNS

- A continuación se ofrece un resumen de la “teoría” que hay detrás de cada aplicación.
- Sin el conocimiento del funcionamiento básico y protocolos que usan estas dos aplicaciones es difícil hacer las prácticas.
- Por ello, en primer lugar es necesario LEER el funcionamiento de estas dos aplicaciones en el libro de referencia.
- Después puedes leer estas transparencias, donde se ofrece un resumen de los puntos más importantes del libro.
- Finalmente, puedes hacer los ejercícios de la práctica.
- Adicionalmente, puedes hacer las prácticas 1 y 2 del laboratorio wireshark del libro de referencia.

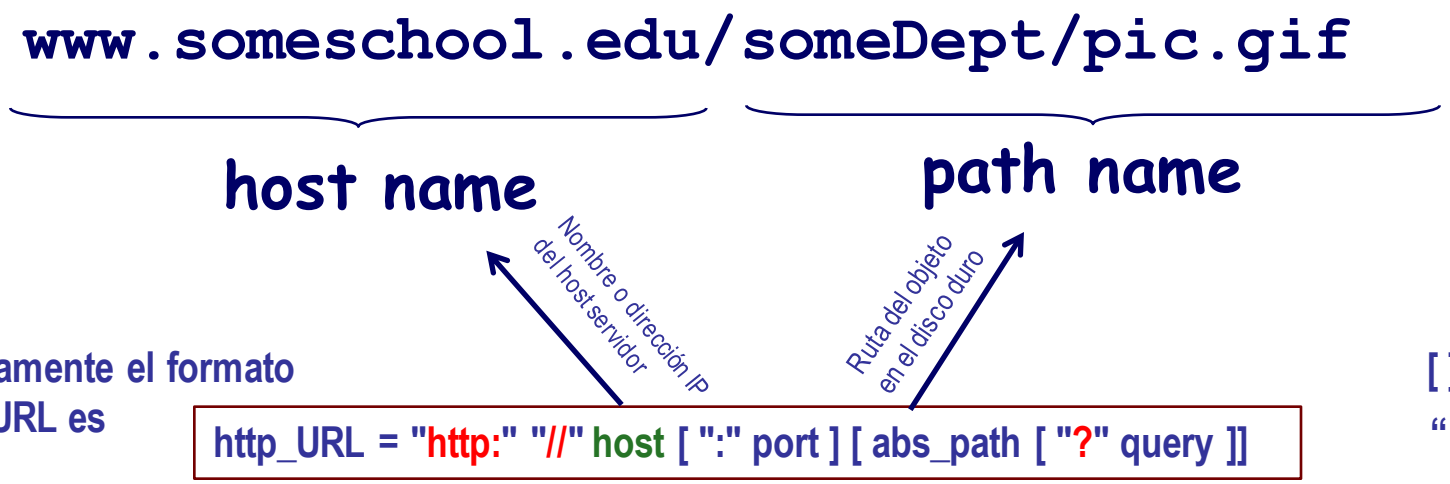


Aplicaciones de Prácticas

La web clásica

Web y HTTP: conceptos previos

- Página Web: es un documento que está compuesto por objetos.
- Una página web consiste en un fichero base en formato HTML que incluye referencia a otros objetos que están también almacenados como ficheros de tipo imágenes JPEG, applets de java, audio, etc..
- Cada objeto es direccionable (accesible) por una URL (Universal Resource Locator)
- Ejemplo de URL: masai.us.es/index.html



Genéricamente el formato de una URL es

http_URL = "http:" "//" host [":" port] [abs_path ["?" query]]

[ ] opcional

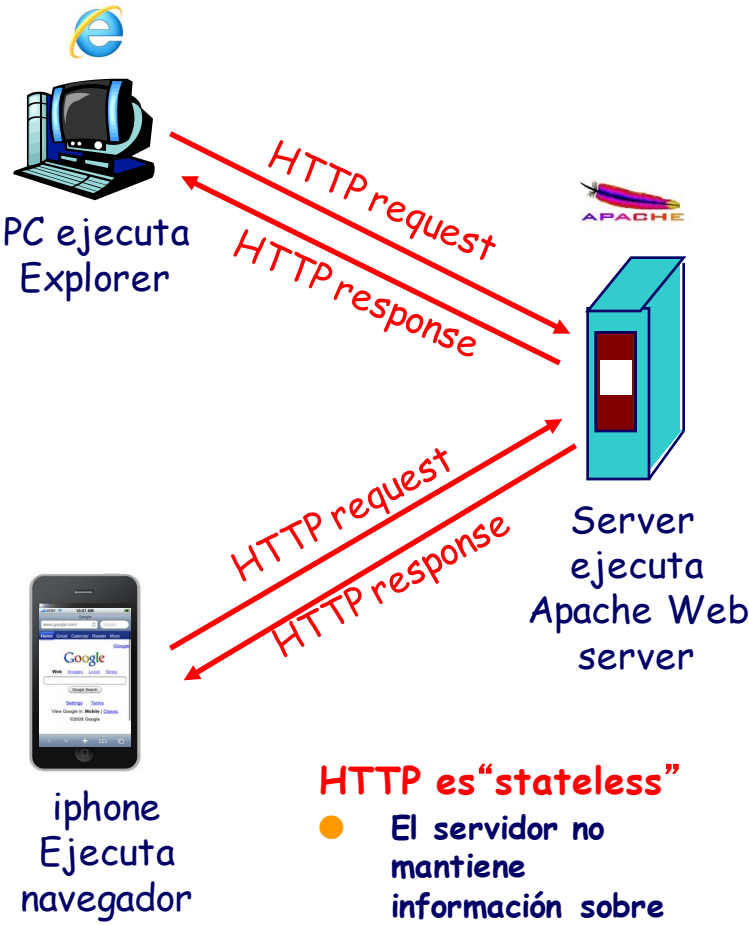
" " literal

HTTP (Hypertext Transfer Protocol)

- Protocolo para la aplicación Web.
  - Tipo textual
  - Dos tipos de mensajes HTTP
    - Petición (request)
    - Respuesta (response)

- Arquitectura C/S
  - Cliente: (navegador) envía peticiones y recibe y representa los objetos web
  - Servidor Web: envía objetos en respuesta a las peticiones del cliente

- La aplicación web usa el servicio de TCP para el transporte de mensajes
  - TCP es un protocolo orientado a conexión



HTTP es "stateless"

- El servidor no mantiene información sobre peticiones pasadas

Conexiones usadas por HTTP

- La conexión debe ser solicitada por el cliente y aceptada por el servidor (puerto 80 TCP) previamente al envío de datos.
  - El cliente crea un socket TCP y le pide que se conecte al socket del servidor

- Una vez establecida la conexión, los dos procesos de aplicación pueden intercambiar mensajes del protocolo HTTP
- Una vez finalizado el intercambio, cierran la conexión

HTTP No persistente

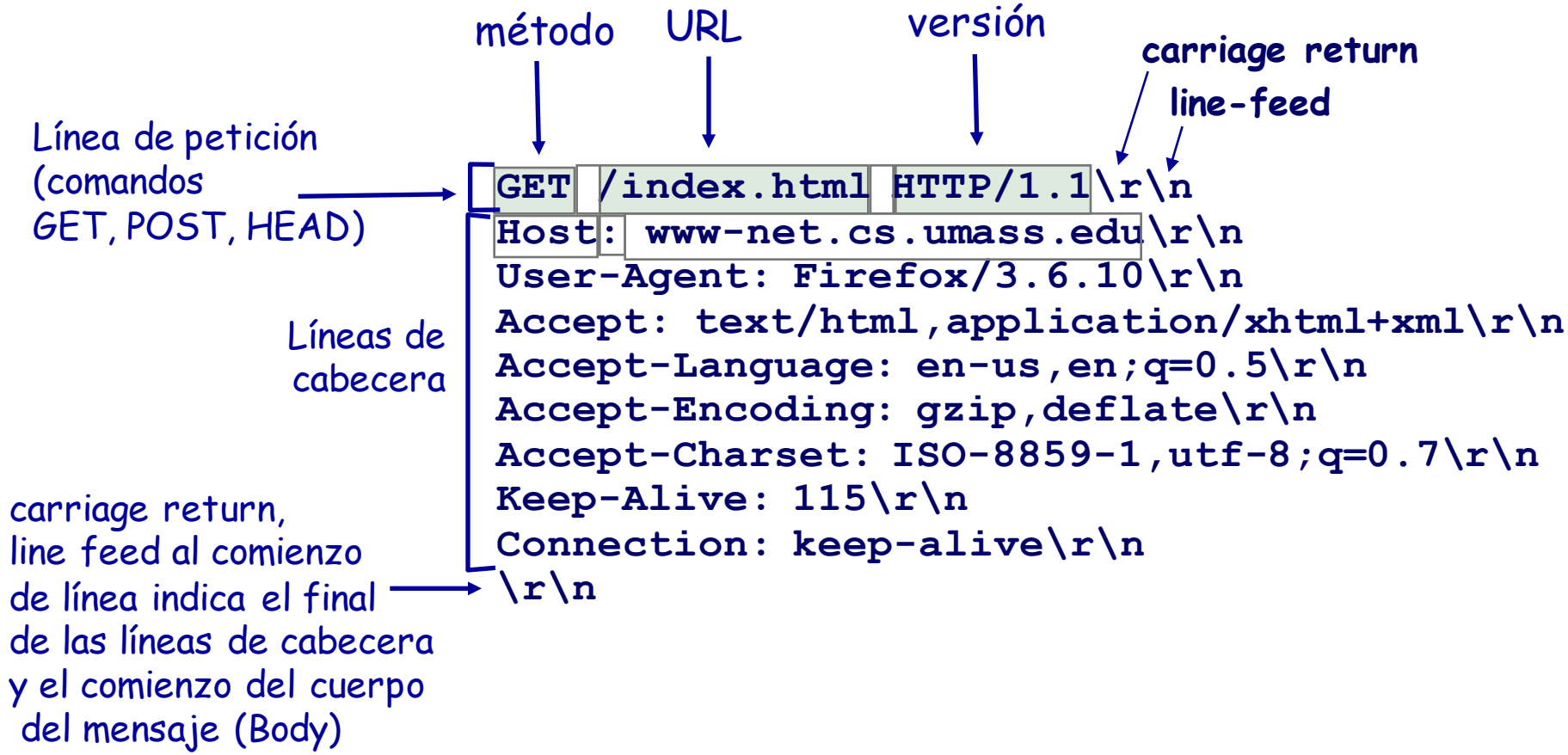
- Por una conexión TCP sólo se puede enviar como máximo un objeto
- Cada objeto requiere su propia conexión TCP.

HTTP persistente

- Por una conexión TCP se pueden enviar multiples objetos

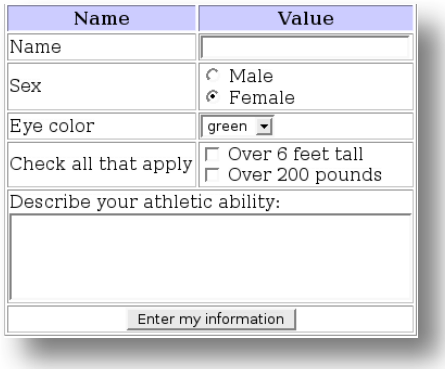
Mensajes de Petición HTTP (RFC2612)

- Protocolo textual (legibles por personas)



Enviando al servidor información de formularios

- Los mensajes de petición solicitan objetos pero…
- HTML permite el uso de formularios dentro de una página web
  - El formulario esta compuesto por objetos de recolección de datos de usuario (cuadros de texto, selectores, etc..)
  - La información que entra el usuario debe ser subida al servidor para su procesamiento (C→S)
    - Normalmente a través de algún lenguaje de programación (p.e. php) .. Páginas dinámicas .
- HTTP permite subir esta información (C→S)
  - Método POST
    - Sube la información en el cuerpo del mensaje
  - Método de la URL (usa el método GET)
    - Sube la información en el campo URL de la línea de petición:
      - www.miweb.com/search?Name=Antonio&EyeColor=2



Métodos de las peticiones según el protocolo

HTTP/1.0 (RFC 1945)

- GET (solicita objeto)
  - Sube información de formularios en URL
- POST(solicita objeto)
  - Sube información de formularios en el cuerpo
- HEAD(solicita objeto)
  - Pide al servidor que no incluya el objeto pedido en la respuesta

HTTP/1.1 (RFC 2616)

- GET, POST, HEAD
- PUT
  - Sube fichero en el cuerpo a la ruta espcificada en URL
- DELETE
  - Borra fichero especificado en el campo URL

Mensaje de Respuesta HTTP

200 OK
request succeeded, requested object later in this msg
301 Moved Permanently
requested object moved, new location specified later in this msg (Location:)
400 Bad Request
request msg not understood by server
404 Not Found
requested document not found on this server
505 HTTP Version Not Supported

Status Line (protocol status code status phrase)

header lines

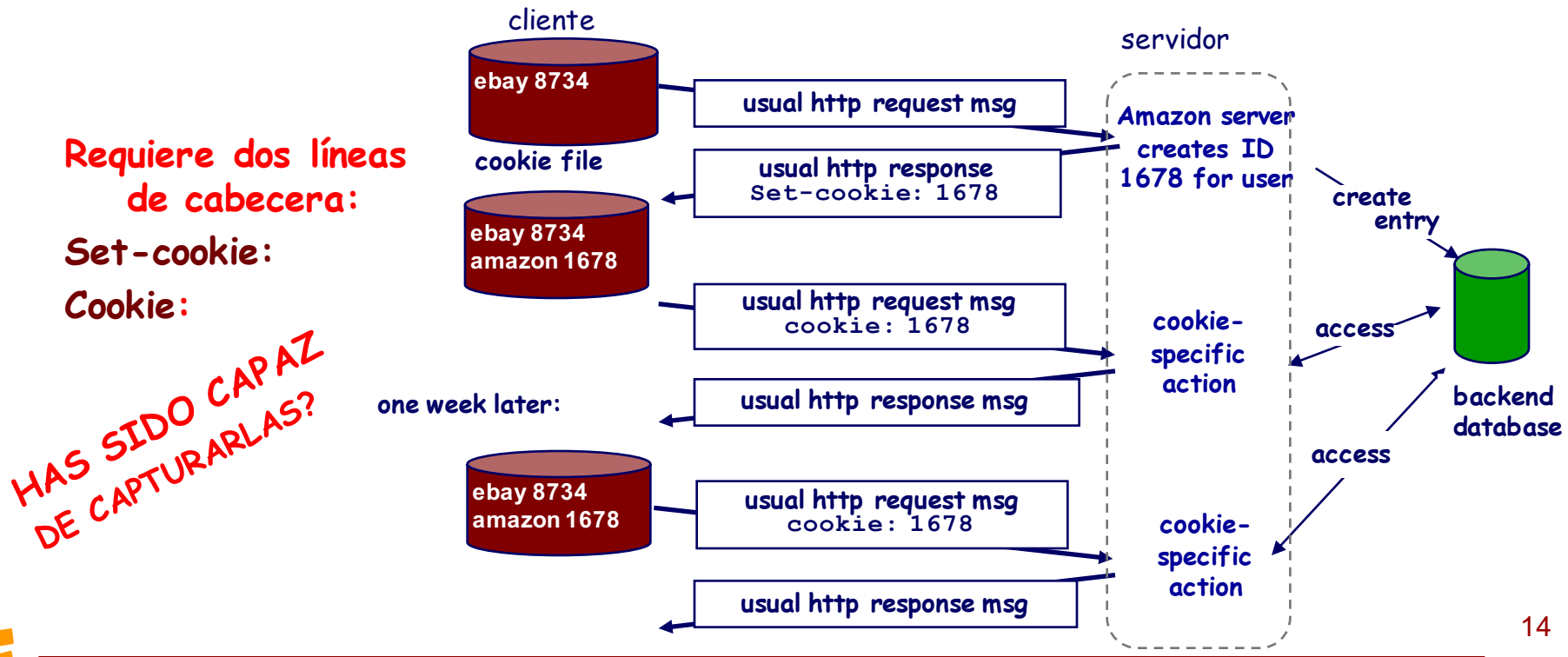
data, e.g., requested HTML file

```
HTTP/1.1 200 OK\r\n
Date: Sun, 26 Sep 2010 20:09:20 GMT\r\n
Server: Apache/2.0.52 (CentOS)\r\n
Last-Modified: Tue, 30 Oct 2007 17:00:02 GMT\r\n
ETag: "17dc6-a5c-bf716880"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 2652\r\n
Keep-Alive: timeout=10, max=100\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html; charset=ISO-8859-1\r\n
\r\n
data data data data data ...
```

Servidores “con estado”: cookies

RFC 6265

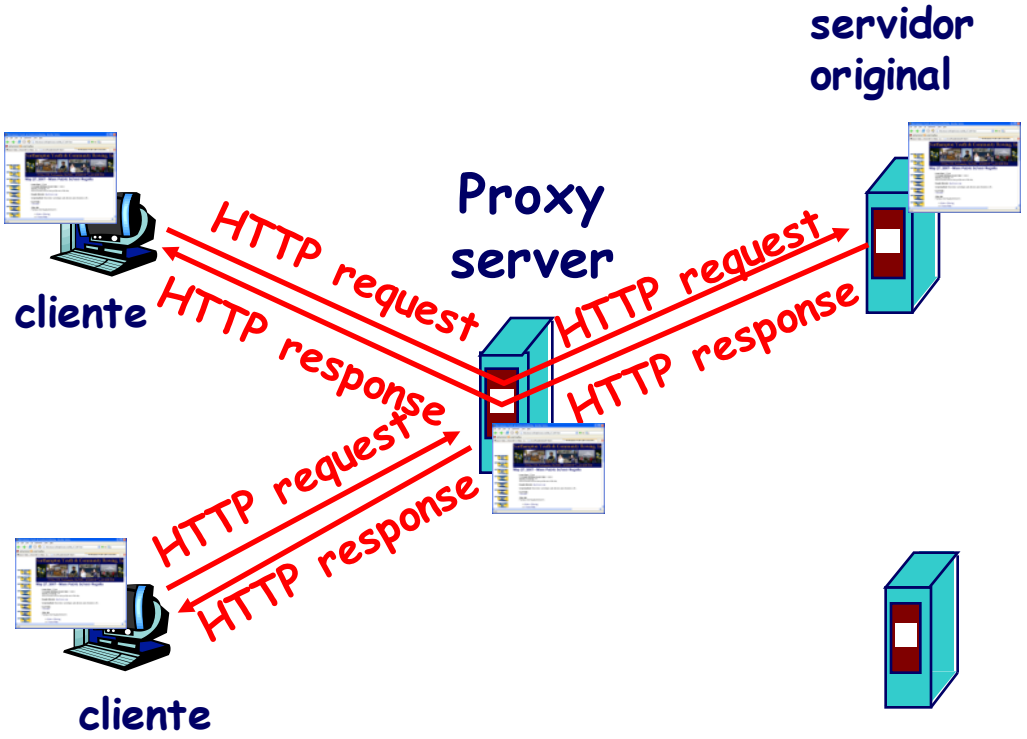
- Los extremos del protocolo pueden mantener el estado durante múltiples transacciones (el servidor quiere “conocerte”)
  - P.e. Para saber tu actividad pasada o personalizar una página
- Mecanismo



Cache Web (Servidores Proxy)

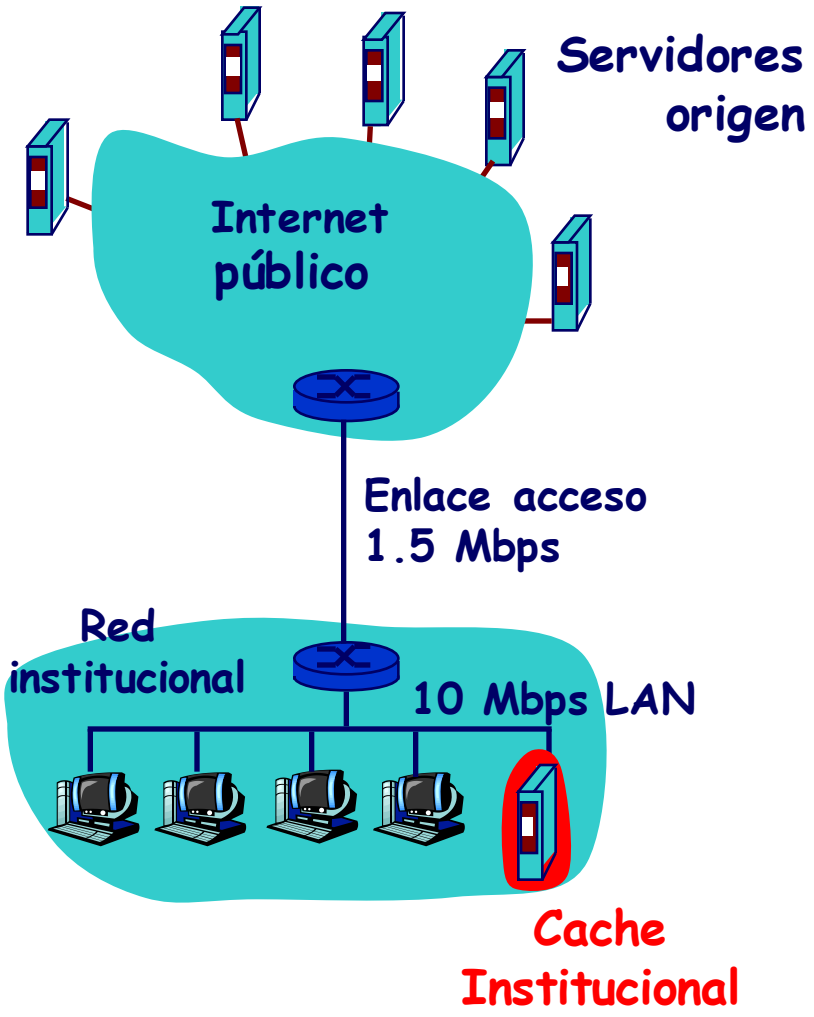
- Objetivo: satisfacer las peticiones de los clientes sin involucrar a los servidores

- Usuarios: configuran el navegador para usar el proxy
- El navegador envía todas las peticiones al proxy
  - Si el proxy las tiene en caché las envía a los usuarios
  - Si no, las solicita al servidor original antes de enviársela al cliente.



Cache Webs

- El Proxy o cache web hace las veces de cliente y servidor
- Suele ser instalado por los ISPs (universidad, empresa, etc..)
- ¿Qué ventajas tiene?
  - Reduce el tiempo de respuesta para atender la petición del cliente
  - Reduce el tráfico en la red de acceso del cliente
  - Incrementa la seguridad



… y si cambia algún objeto en el Servidor Web ?

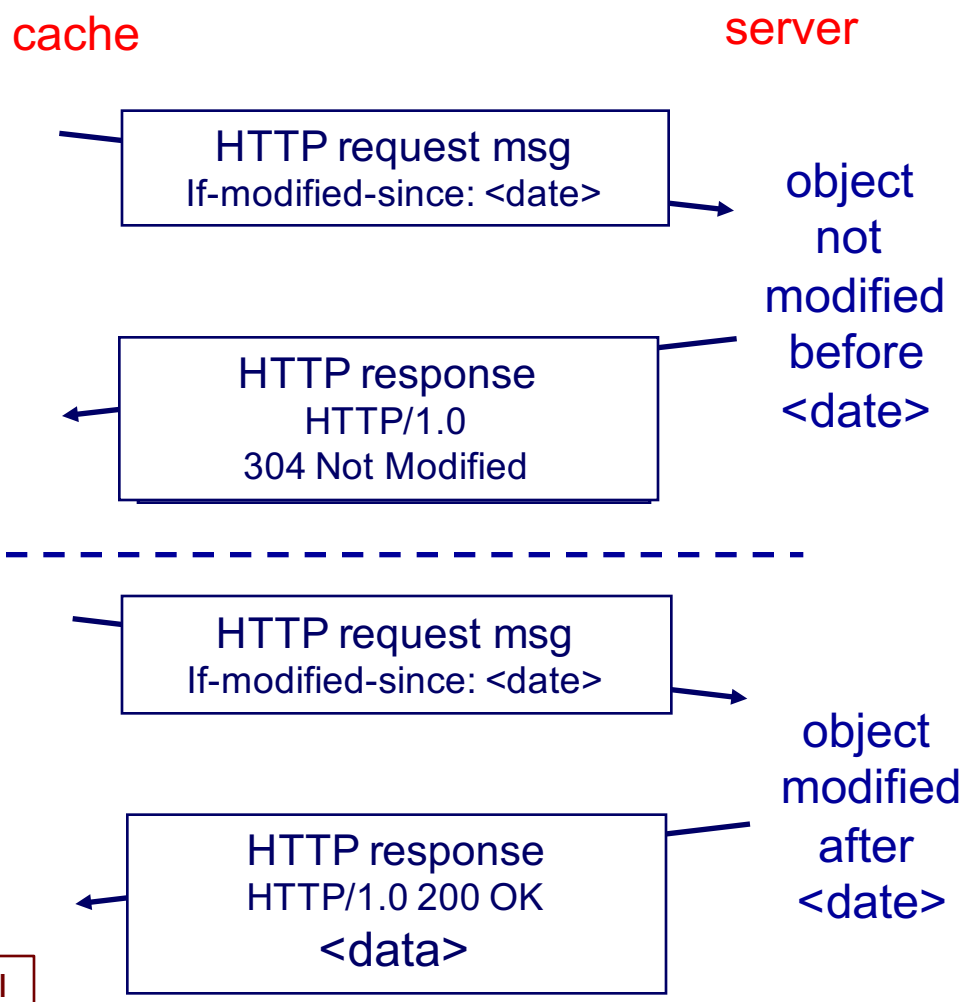
- Conditional GET
- El servidor no envía el objeto si el proxy tiene una versión actualizada
  - La fecha de la versión en cache se envía como una cabecera

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

  - El servidor comprueba la fecha del caché con la del objeto que almacena. Si esta actualizada, le envía

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

Los propios navegadores suelen mantener un cache local y usan conditional Get para asegurarse de que su cache esta actualizada



Ahora ya puedes hacer la parte I de la práctica.

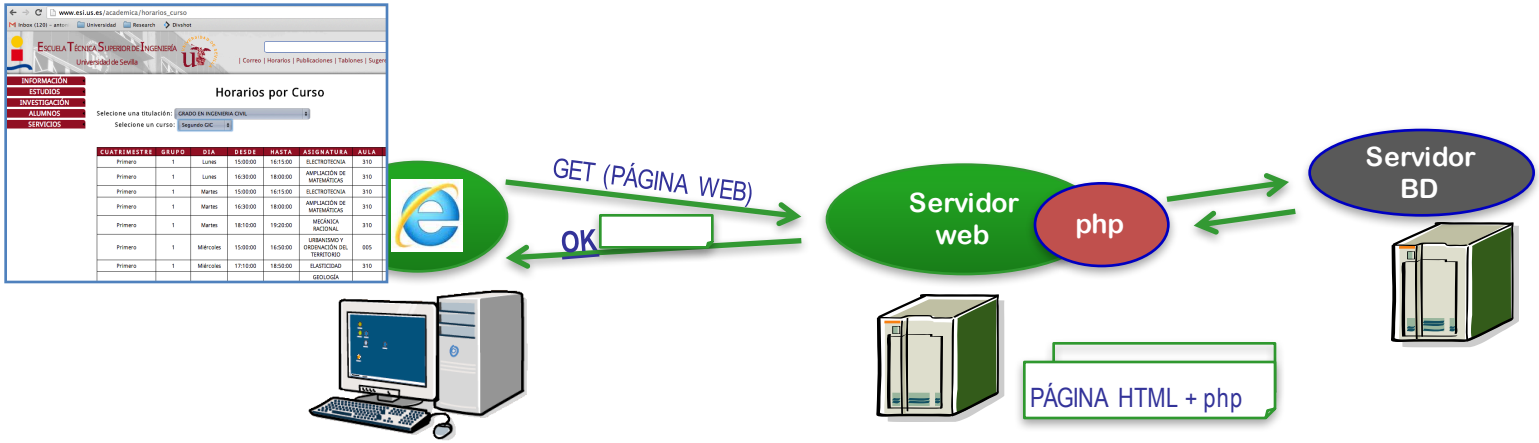
- Consulta estas transparencias y el libro cuando tengas dudas
- Puedes hacerla en linux o windows
- Anota las dudas que te surjan y no seas capaz de aclararlas.
- Haz el examen de auto-evaluación (en algunas respuestas puede cambiar la versión de software o el tiempo de creación de los ficheros debido a una reparación en masai.us.es)
- Lee cada uno de los puntos “Qué deberías saber a partir de ahora”. Auto-evalúate.

¿Has tenido algún problema en …?

- Poner en marcha Wireshark para capturar por el interfaz de red deseado
  - Grabar / Cargar capturas
- Interpretar el Interfaz de Usuario de Wireshark
  - Identificar las diferentes cabeceras encapsuladas
  - Identificar origen y destino de cada paquete a nivel 3 y 4
- Capturar mensajes de petición HTTP
- Capturar mensajes de respuesta HTTP
- Buscar e interpretar el significado de cabeceras según RFC
- Identificar todos los segmentos que pertenecen al mismo mensaje de aplicación
- Capturar el uso del Conditional GET?
- Entender su uso por parte de navegadores o Proxys?
  - Entender sus ventajas?

La Web para usos “no clásicos”

- Los servidores y clientes web podrían ejecutar una lógica que utilizase los datos de los mensajes HTTP
  - P.e. El contenido de la línea de petición, líneas de cabecera o cuerpo del mensaje (información de formularios)
- Una vez ejecutada esa lógica, el servidor devolverá un mensaje de respuesta HTTP
  - el cuerpo del mensaje no tiene por qué ser un fichero estático sino algo “generado” ad-hoc (página dinámica)
- Otras veces, el servidor enviará un trozo de código al para que éste lo ejecute (javascript)



Aplicaciones de Ejemplo

DNS – Sistema de nombres de dominio (práctica 2)

Aplicación DNS: Domain Name System

- Es mas fácil para las personas recordar nombres que direcciones numéricas

- Sin embargo, para las máquinas es al revés (dirección IP: 182.12.52.88)
  - Los routers usan la dirección IP para identificar al destino.

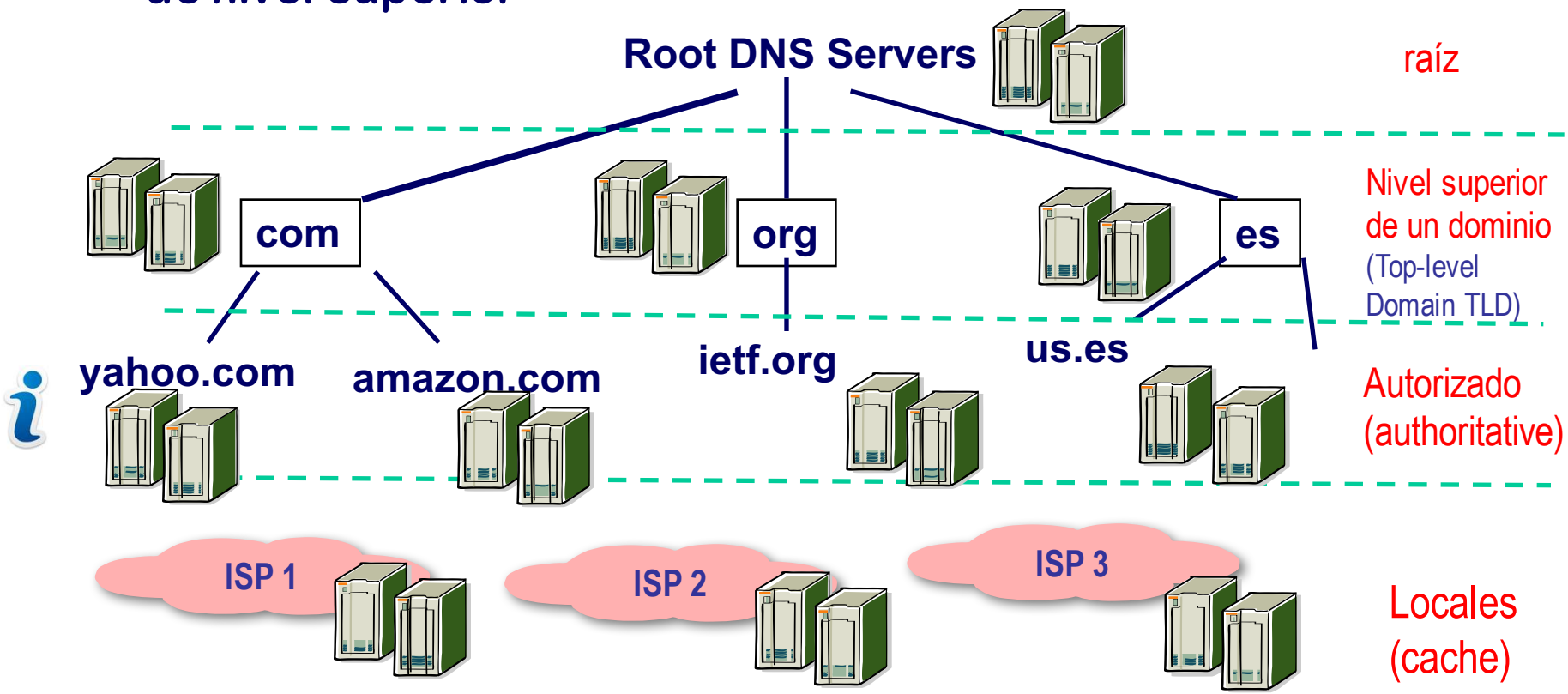
- El sistema de Nombres de dominio (DNS) es una aplicación distribuida
- Compuesta por:
  - Una base de datos distribuida que asocia los nombres de las máquinas → direcciones IP
  - Un protocolo de aplicación para la consulta (resolución) de la base de datos
    - Usa servicio UDP (puerto 53)
- Es básica en Internet
  - Reside en el borde de la red

Aplicación DNS: Domain Name System

- Tiene varias aplicaciones (además de la evidente)
  - Traducción de nombre a dirección IP
  - Hosts Aliasing (múltiples nombres para la misma máquina)
    - Nombre canónico, nombres alias.
  - Alias de servidores de email
  - Distribución de carga
    - Servidores Web replicados con diferentes IPs y bajo el mismo nombre.
- Es un aplicación C/S, pero no hay un único servidor. Los datos se distribuyen entre varios servidores
  - Evita
    - Único punto de fallo
    - Excesivo volumen de tráfico
    - Excesivo tiempo de respuesta
    - Excesivo Mantenimiento (añadir, cambiar, borrar datos)

La Base de Datos de DNS es distribuida y jerárquica

- Varios servidores DNS en cada nivel de la jerarquía.
- Los servidores de un nivel conocen, al menos la dirección IP de los servidores dentro de su dominio y la de algún servidor de nivel superior



Jerarquía de Servidores DNS

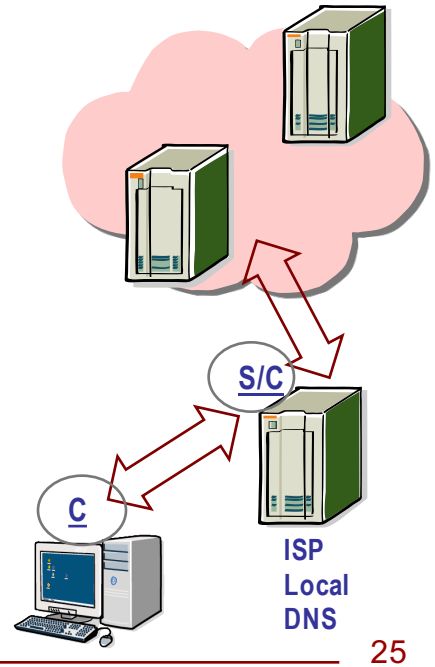
- Raíz (root)
  - Los servidores Locales pueden contactar con él si no tienen otra alternativa.
  - Contacta con servidor de nombres autorizado (authoritative) si no conoce el mapeo para resolver un nombre
  - Hay 13 distribuidos por el mundo (con réplicas)

http://www.root-servers.org

- Nivel superior del dominio (TLD)
  - Responsables de dominios de nivel superior (com , net, org) y y todos los dominios de nivel superior nacional (uk, es, etc..)
    - Network solutions mantiene com http://www.networksolutions.com

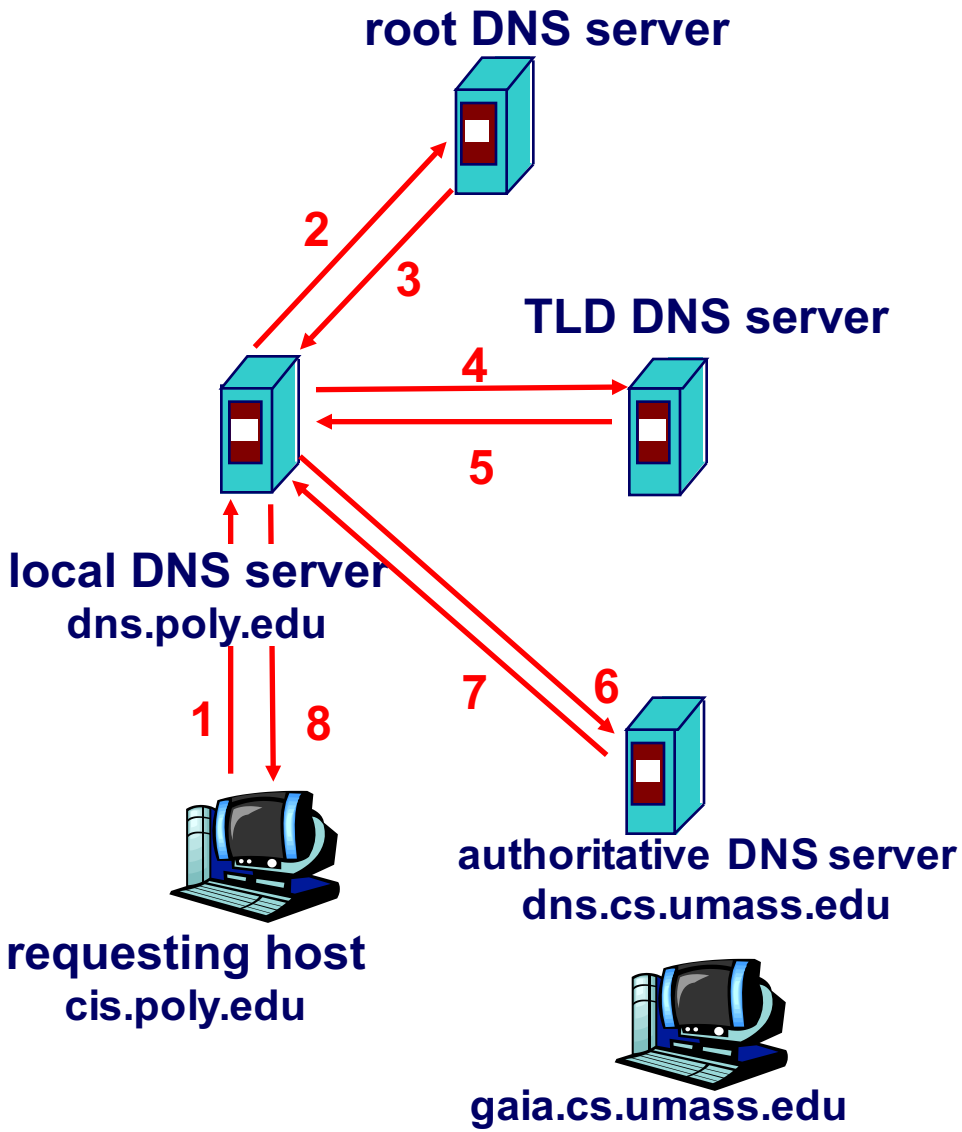
- Servidores Autorizados (authoritative)
  - Servidores DNS de una organización (dominio).
  - Ofrecen mapeos "autorizados" o fiables para los servidores de un dominio
  - Pueden ser mantenidos por la propia organización o un SP (prov.)

- Servidores de Nombre Locales (no pertenecen a la jerarquía)
  - Cada ISP tiene al menos uno (default name server)
  - Los clientes DNS de los hosts realizan su consulta siempre a los servidores Locales (actúan como una especie de proxy)



Ejemplo Resolución DNS

- El host cis.poly.edu quiere saber la IP de gaia.cs.umass.edu



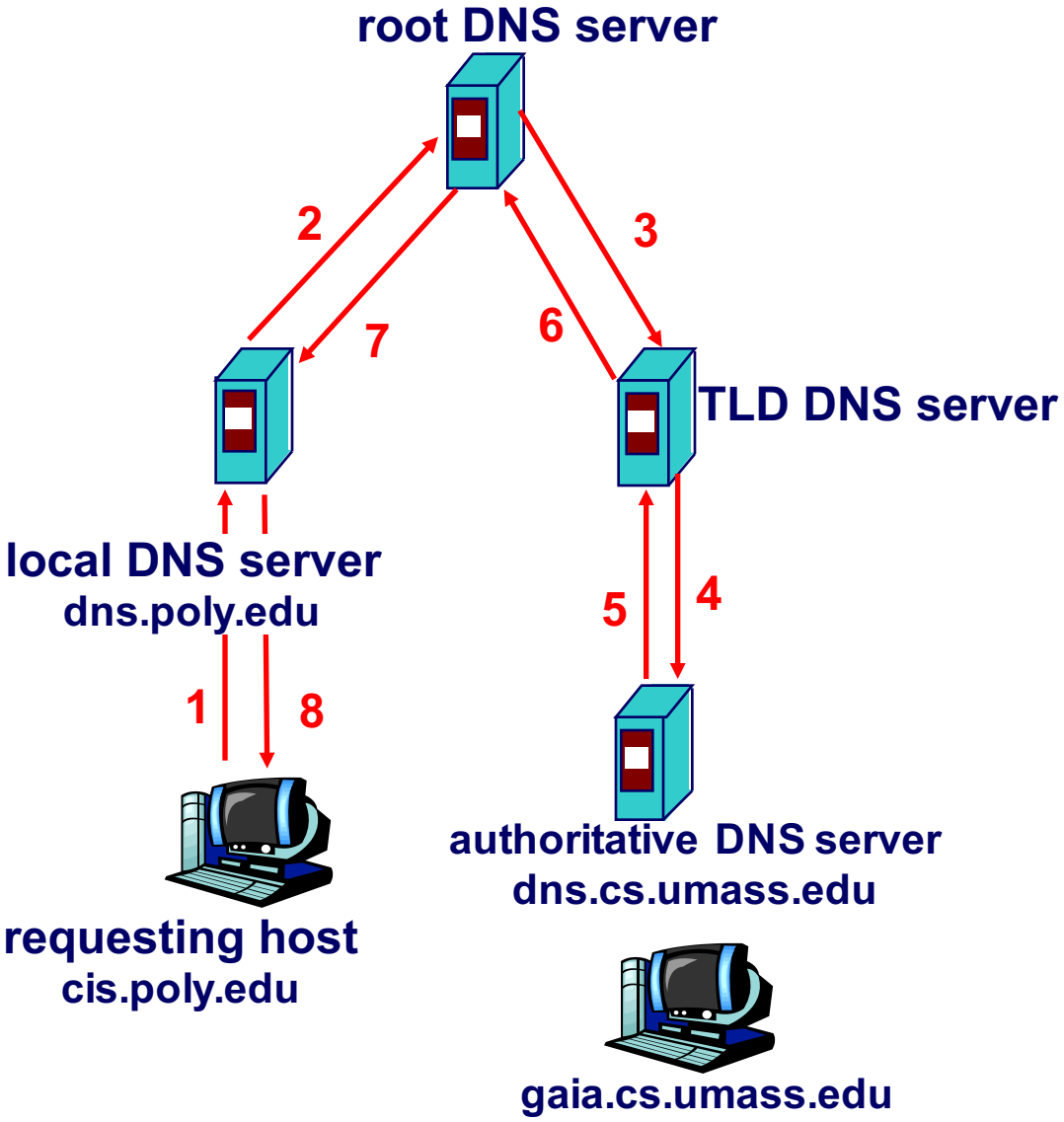
Consulta ITERATIVA :

- El Servidor contactado responde con el nombre del servidor con el que contactar
- "No lo se, pero pregunta a este"

También podría haberse resuelto de forma recursiva

Ejemplo Resolución DNS (II)

- El host cis.poly.edu quiere saber la IP de gaia.cs.umass.edu



Consulta RECURSIVA:

- El Servidor contactado es el responsable de resolver

Hay una caché local en cada servidor donde guarda las resoluciones aprendidas (cada nueva entrada caduca después de un tiempo)

Los TLD suelen estar en la caché de los locales (el root no se suele usar)

Registros de Recursos DNS (DNS Resource Record)

- Es la unidad información que se almacena en la BBDD distribuida

Formato RR: `(name, value, type, ttl)`

- Cuatro tipos de Registros de Recursos (principales)

Type=A

- `name` →hostname
- `value` → dirección IP

Type=NS

- `name` →dominio (p.e. foo.com)
- `value` → hostname de un servidor DNS autorizado para el dominio

Type=CNAME

- `name` → alias de algún nombre "canonico" (el real)
- `www.ibm.com` es realmente `servereast.backup2.ibm.com`
- `value` → nombre "canonico"

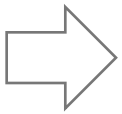
Type=MX

- `value` → nombre del servidor de email asociado al dominio (`name`)

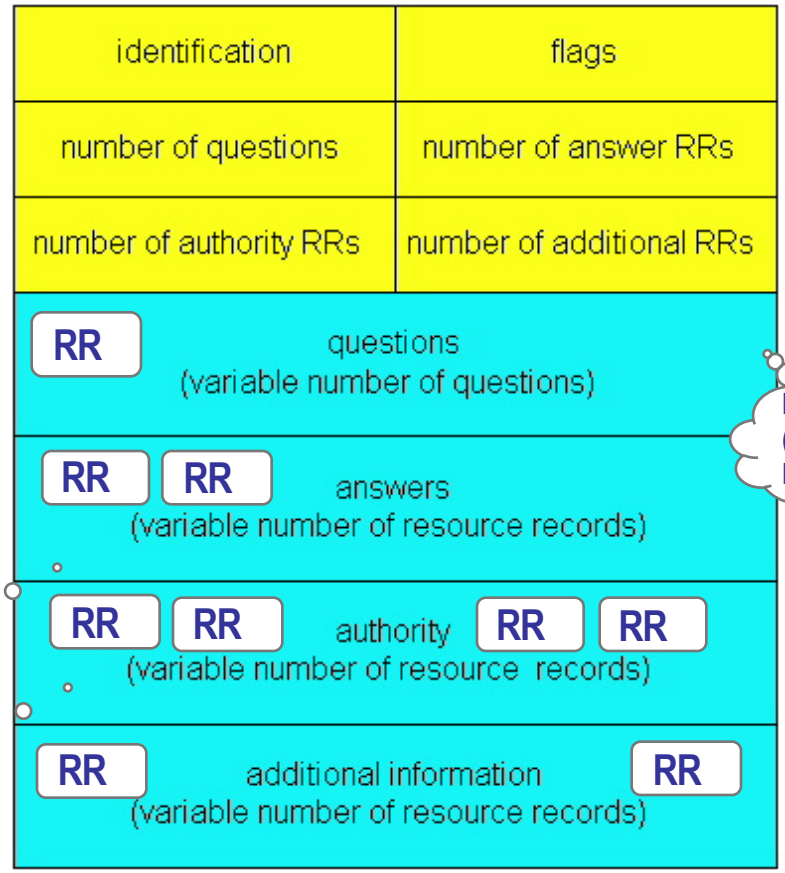
Protocolo de la aplicación DNS: mensajes

- Protocolo de petición/respuesta (cliente/servidor)
- Usa el servicio de transporte UDP (pto 53)
- Ambos tipos de mensajes con igual formato
- Cabecera (A-PCI)
  - Identificador: # 16bits usado para correlacionar req/res
  - Flags:
    - Req / Res (Query o Reply)
    - Se desea recursividad
    - Se dispone de recursividad
    - La respuesta es

RFC 882
RFC 883
(1983)



RFC 1034
RFC 1035
(1987)



¿cómo inserto un nuevo registro en el DNS?

- Por ejemplo: nueva startup "botijos.com"
- Registro el nombre "botijos.com" en un Registrar DNS. (p.e network solutions)
  - Le doy el nombre, dirección IP de los servidores autorizados (primario y secundario)
  - El registrar inserta estos dos RRs en el servidor com TLD

```
(botijos.com, dns1.botijos.com, NS)
(dns1.botijos.com, 212.212.212.1, A)
```

- Configuro los Servidores Autorizados (o lo alquilo) para mi dominio
  - Un registro tipo A para www.botijos.com; (servidor web)
  - Un registro tipo MX para botijos.com (servidor email)

Lista de Registrars http://www.internic.net/alpha.html

Ahora ya puedes hacer la parte II de la práctica.

- Consulta estas transparencias y el libro cuando te surjan dudas
- Puedes hacerlo en linux o windows.
  - Para consultar dns en linux puedes usar varios programas, p.ej: dig o nslookup. Busca manuales o tutoriales para conocer sus opciones de uso.
    - Dig:
      - http://www.rickconner.net/spamweb/tools-dig.html
      - https://www.linode.com/docs/networking/dns/use-dig-to-perform-manual-dns-queries
- Anota las dudas que te surjan y no seas capaz de aclarar.
- Haz el examen de auto-evaluación (en algunas respuestas puede cambiar la versión de software o el tiempo de creación de los ficheros debido a una reparación en masai.us.es)
- Lee cada uno de los puntos “Qué deberías saber a partir de ahora”. Auto-evalúate.

¿Has tenido algún problema en ...

- Capturar los mensajes del protocolo DNS?
- Realizar consultas al DNS local e interpretar sus respuestas?
- Entender cómo se da de alta un nuevo nombre en el sistema?