

Capítulo 2: Capa Aplicación - III

ELO322: Redes de Computadores Agustín J. González

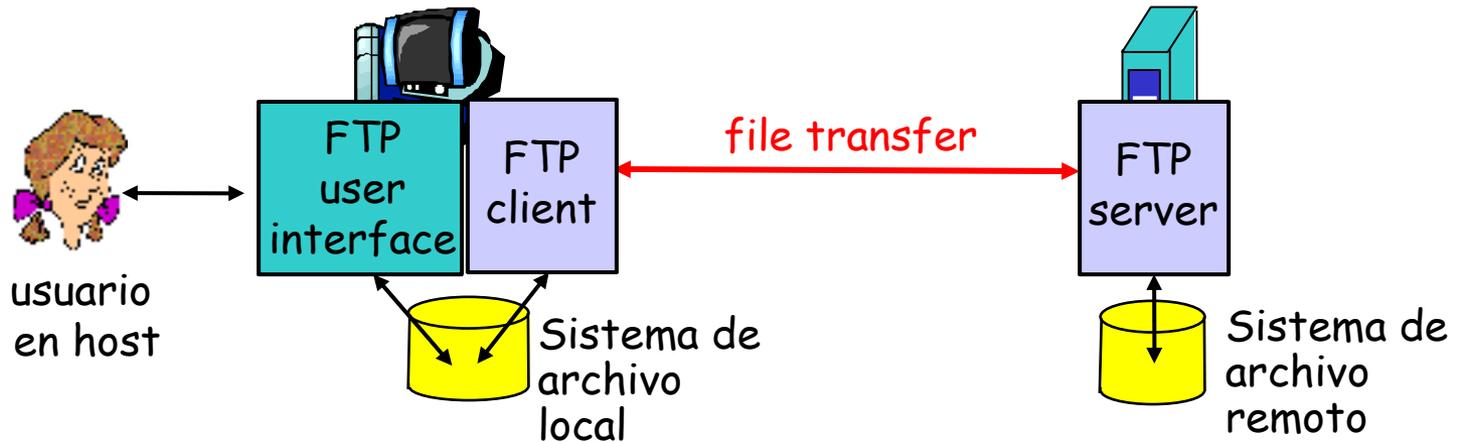
Este material está basado en:

- Material de apoyo al texto *Computer Networking: A Top Down Approach Featuring the Internet* 3rd edition. Jim Kurose, Keith Ross Addison-Wesley, 2004.
- Material del curso anterior ELO322 del Prof. Tomás Arredondo V.

Capítulo 2: Capa Aplicación

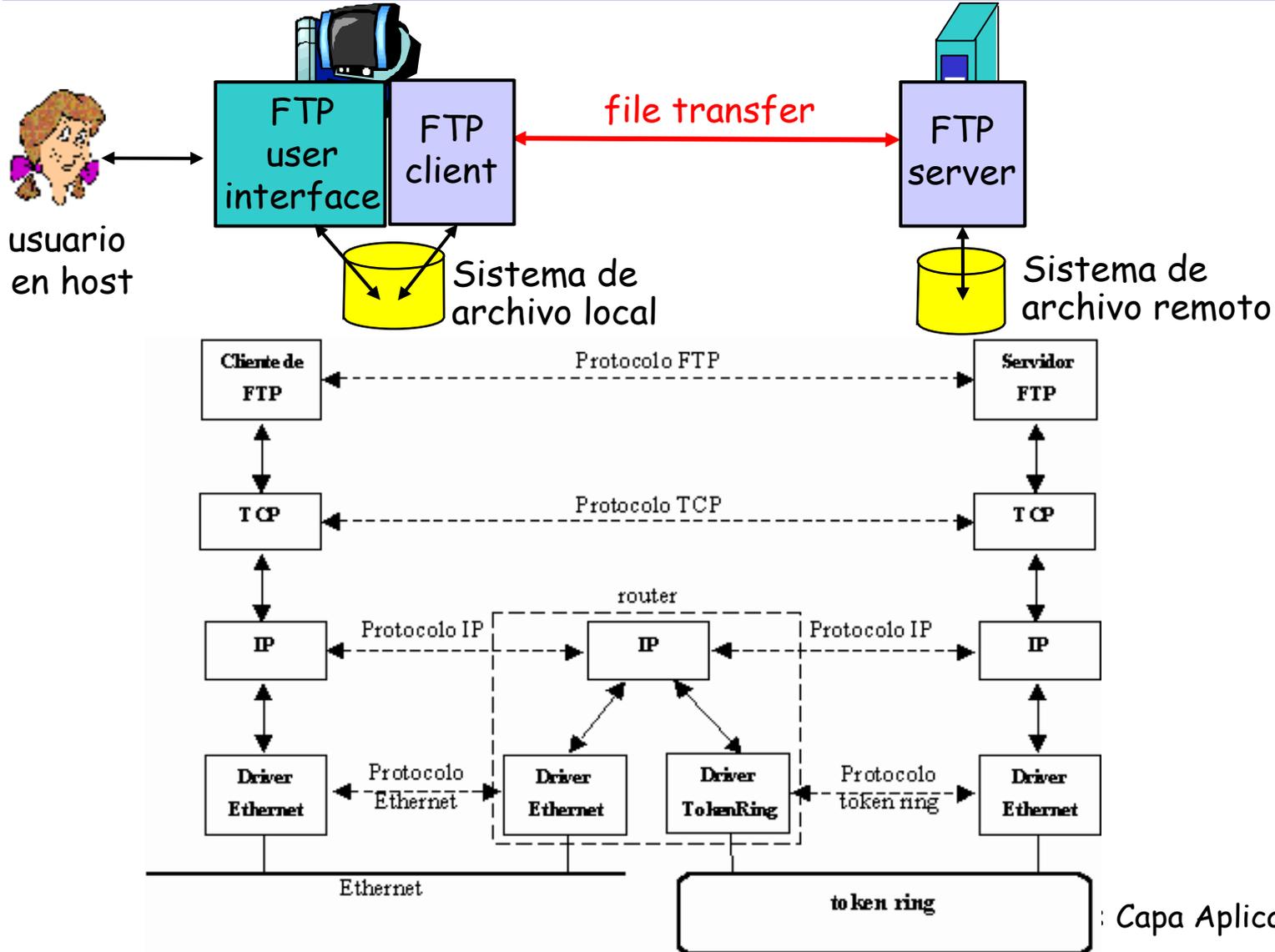
- ❑ 2.1 Principios de la aplicaciones de red
- ❑ 2.2 Web y HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Correo Electrónico
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P Compartición de archivos
- ❑ 2.7 Programación de Socket con TCP
- ❑ 2.8 Programación de socket con UDP
- ❑ 2.9 Construcción de un servidor WEB

FTP: El protocolo de transferencia de archivos (File Transfer Protocol)



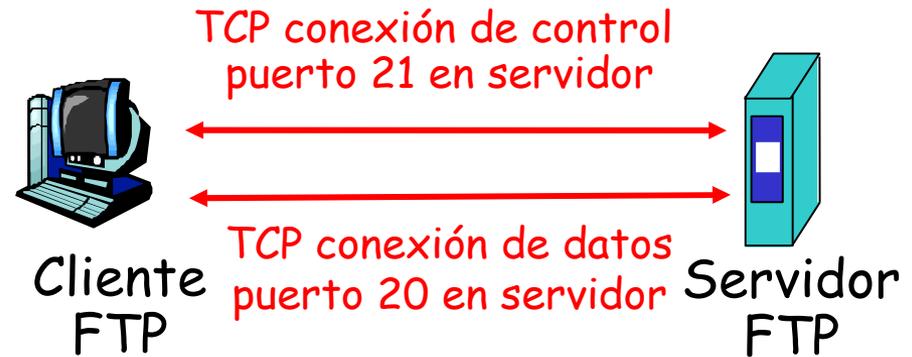
- ❑ Transferencia de archivos a/desde el host remoto
- ❑ Sigue modelo cliente/servidor
 - *cliente*: sitio que inicia la transferencia (ya sea a/desde sitio remoto)
 - *servidor*: host remoto
- ❑ ftp: RFC 959
- ❑ Servidor ftp: puerto 21, cliente en cualquier puerto.
Ver netstat -p tcp

FTP: La capa aplicación se apoya en la inferiores. Hacemos abstracción



FTP: Conexiones separadas de control y datos

- ❑ Cliente FTP contacta servidor FTP en puerto 21, especificando TCP como protocolo de transporte
- ❑ El cliente obtiene autorización sobre el control de la conexión
- ❑ El cliente navega el directorio remoto enviando comandos sobre la conexión de control.
- ❑ Cuando el servidor recibe una petición de transferencia de archivo, el servidor abre una conexión de datos hacia el cliente. Éste es Modo Activo.
- ❑ Después de la transferencia de un archivo, el servidor cierra la conexión de datos.



- ❑ El servidor abre una segunda conexión TCP de datos para transferir otro archivo.
- ❑ Conexión de control: "out of band" (fuera de banda)
- ❑ Servidor FTP mantiene "estado": directorio actual, cuenta de usuario conectado.
- ❑ Existe modo activo y pasivo

FTP comandos, respuestas

Muestra de comandos:

- ❑ Son enviados como texto ASCII vía el canal de control
- ❑ **USER** *username*
- ❑ **PASS** *password*
- ❑ **LIST** retorna la lista de archivos del directorio actual
- ❑ **RETR** *filename* baja un archivo (gets)
- ❑ **STOR** *filename* almacena (puts) archivo en host remoto

Algunos códigos retornados

- ❑ Código estatus y frases (como en HTTP)
- ❑ 331 Username OK, password required
- ❑ 125 data connection already open; transfer starting
- ❑ 425 Can't open data connection
- ❑ 452 Error writing file

Capítulo 2: Capa Aplicación

- ❑ 2.1 Principios de la aplicaciones de red
- ❑ 2.2 Web y HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Correo Electrónico
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P Compartición de archivos
- ❑ 2.7 Programación de Socket con TCP
- ❑ 2.8 Programación de socket con UDP
- ❑ 2.9 Construcción de un servidor WEB

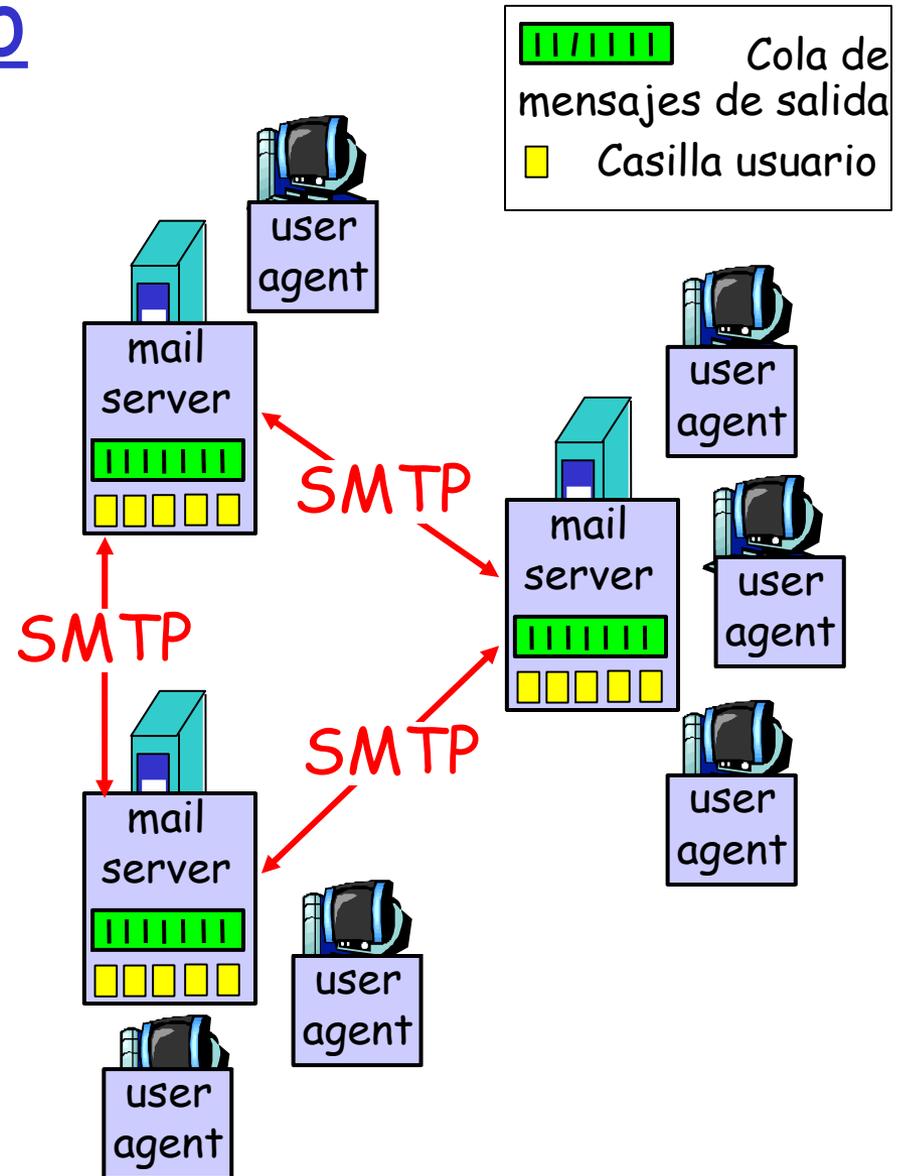
Correo Electrónico

Tres mayores componentes:

- ❑ Agente usuario o cliente de correo
- ❑ Servidor de correo
- ❑ Simple Mail Transfer Protocol: SMTP

Agente Usuario

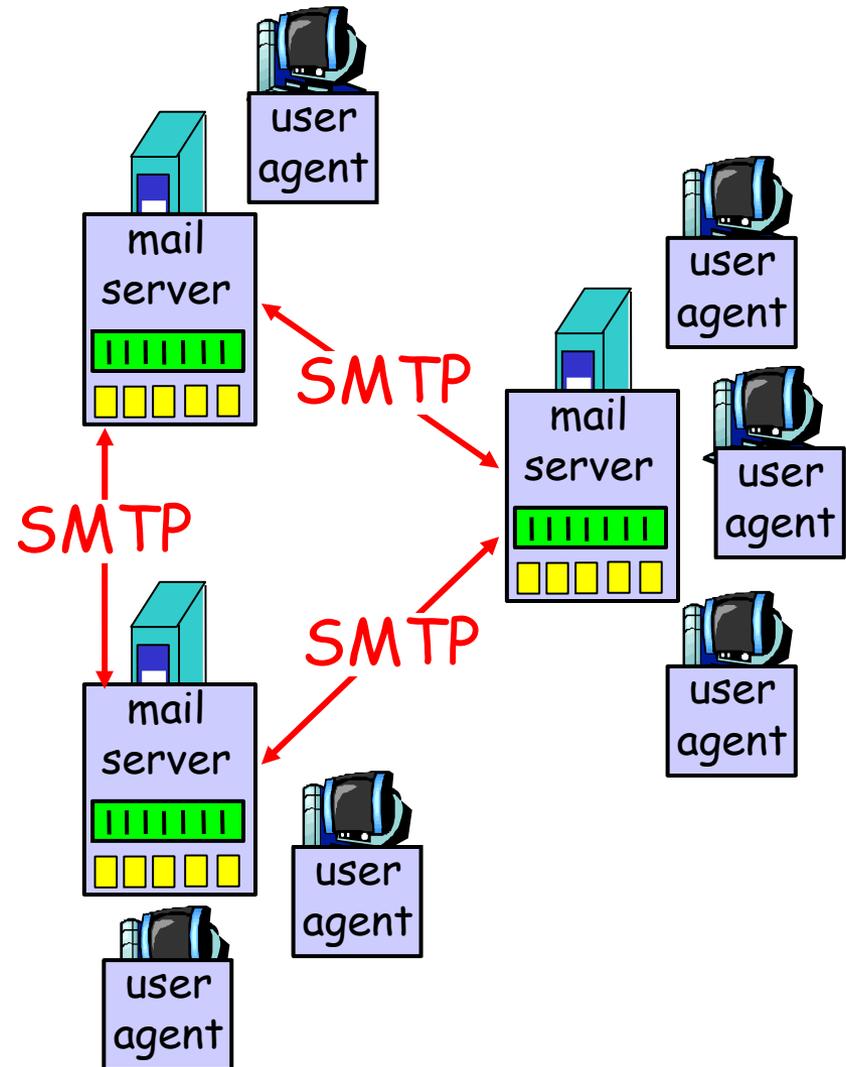
- ❑ También conocido como "lector de correo"
- ❑ Escritura, edición, lectura de mensajes de correos
- ❑ e.g., Eudora, Outlook, elm, Mozilla Thunderbird
- ❑ Mensajes de salida y entrada son almacenados en servidor



Correo Electrónico: Servidor de correo

Servidor de Correo

- ❑ **Casilla** contiene mensajes de entrada para el usuario
- ❑ **Cola de mensajes** de los correos de salida
- ❑ **SMTP: Protocolo** entre servidores de correo para enviar mensajes e-mail
 - cliente: servidor que envía el correo
 - "servidor": servidor que recibe el correo

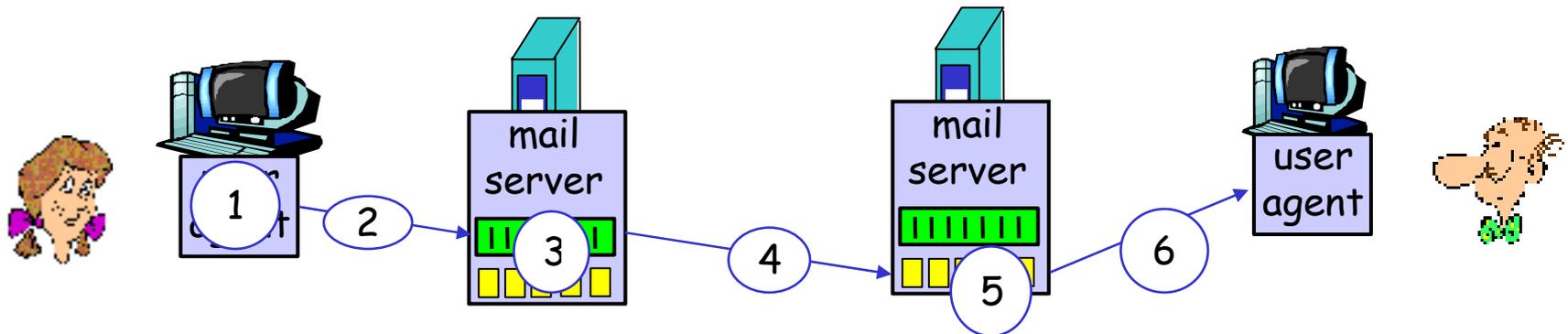


Correo Electrónico: SMTP [RFC 2821]

- ❑ Usa TCP para transferir confiablemente mensajes e-mail desde el cliente al servidor, puerto 25 en servidor.
- ❑ Transferencia directa: servidor envía correos al servidor receptor
- ❑ Tres fases de la transferencia
 - handshaking (apretón de manos para establecer conexión)
 - transferencia de mensajes
 - cierre
- ❑ Interacción comandos/respuestas
 - **comandos:** Texto ASCII
 - **respuesta:** código de estatus y frase.
- ❑ Mensajes deben ser enviados en ASCII de 7-bits
¿Qué pasa con las fotografías y archivos binarios?

Escenario: Alicia envía mensaje a Bob

- 1) Alicia usa agente usuario para componer el mensaje para bob@someschool.edu
- 2) El agente de Alicia envía en mensaje a su servidor de correo; el mensaje es puesto en cola de salida
- 3) Lado cliente de SMTP abre una conexión TCP con el servidor de correo de Bob
- 4) El cliente SMTP envía el mensaje de Alicia por la conexión TCP
- 5) El servidor de correo de Bob pone el mensaje en su casilla
- 6) Bob invoca su agente usuario para leer el mensaje



Ejemplo de Interacción SMTP

Luego de: \$telnet hamburger.edu 25 <enter>

S: 220 hamburger.edu

C: HELO crepes.fr

S: 250 Hello crepes.fr, pleased to meet you

C: MAIL FROM: <alice@crepes.fr>

S: 250 alice@crepes.fr... Sender ok

C: RCPT TO: <bob@hamburger.edu>

S: 250 bob@hamburger.edu ... Recipient ok

C: DATA

S: 354 Enter mail, end with "." on a line by itself

C: Do you like ketchup?

C: How about pickles?

C: .

S: 250 Message accepted for delivery

C: QUIT

S: 221 hamburger.edu closing connection

Prueben la interacción SMTP :

- ❑ `telnet servername 25`
- ❑ Ver respuesta 220 desde el servidor
- ❑ Ingresar los comandos HELO, MAIL FROM, RCPT TO, DATA, QUIT

Lo de arriba nos permite enviar correo sin usar el cliente de correo.

/ Equivalentemente, si usted sabe cómo escribir una aplicación que maneje sockets, usted sabe cómo enviar e-mail desde su aplicación.*/*

- ❑ Hoy muchos servidores están configurados para aceptar sólo conexiones seguras que no permiten el uso de telnet para envío de correo. La USM usa TLS (Transport Layer Security)

SMTP: palabras finales

- ❑ SMTP usa conexiones persistentes
- ❑ SMTP requiere que el mensaje (encabezado y cuerpo) sean en ASCII de 7-bits
- ❑ Servidor SMTP usa CRLF. CRLF para terminar el mensaje; es decir, una línea con sólo un punto en ella.

Comparación con HTTP:

- ❑ HTTP: pull (saca contenido desde servidor)
- ❑ SMTP: push (pone contenido en servidor)
- ❑ Ambos tienen interacción comando/respuesta en ASCII, y tienen códigos de estatus
- ❑ HTTP: cada objeto es encapsulado en su propio mensaje
- ❑ SMTP: múltiples objetos son enviados en un mensaje multiparte

Formato de mensajes de correo (comando DATA)

SMTP: protocolo para intercambio de mensajes de correo

RFC 822: estándar para el formato de los mensajes:

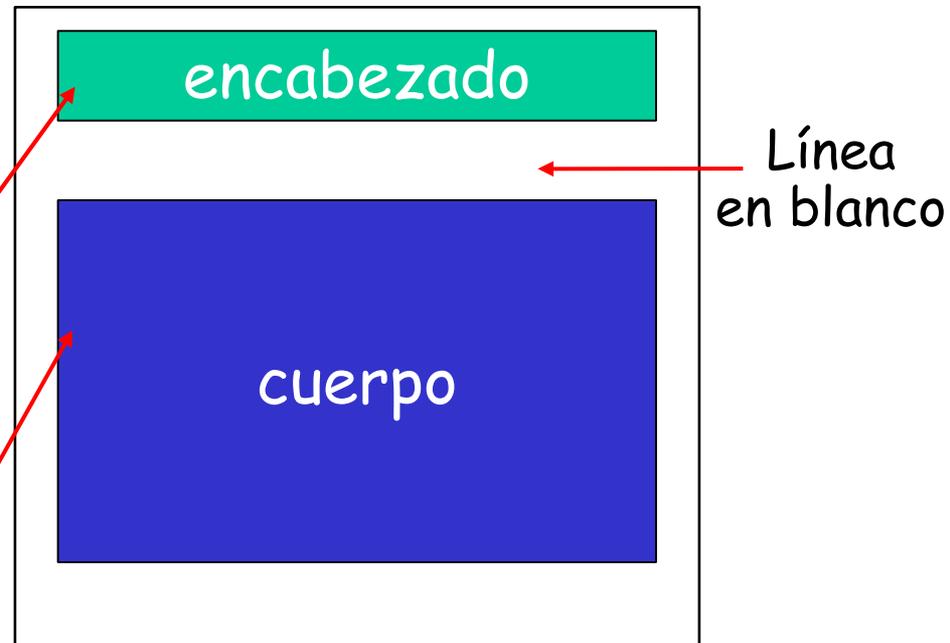
□ E.g. líneas de encabezado (opcional), entre otros:

- To:
- From:
- Subject:

diferente a los comandos SMTP!

□ Cuerpo

- El "mensaje", sólo caracteres ASCII



Formato de mensaje: extensiones multimedia

- ❑ MIME: "multimedia mail extension", RFC 2045, 2056
- ❑ Líneas adicionales en el encabezado del mensaje declaran el tipo de contenido MIME
- ❑ La codificación Base64 usa sólo los caracteres: A-Z, a-z, 0-9 y +/=

Versión MIME

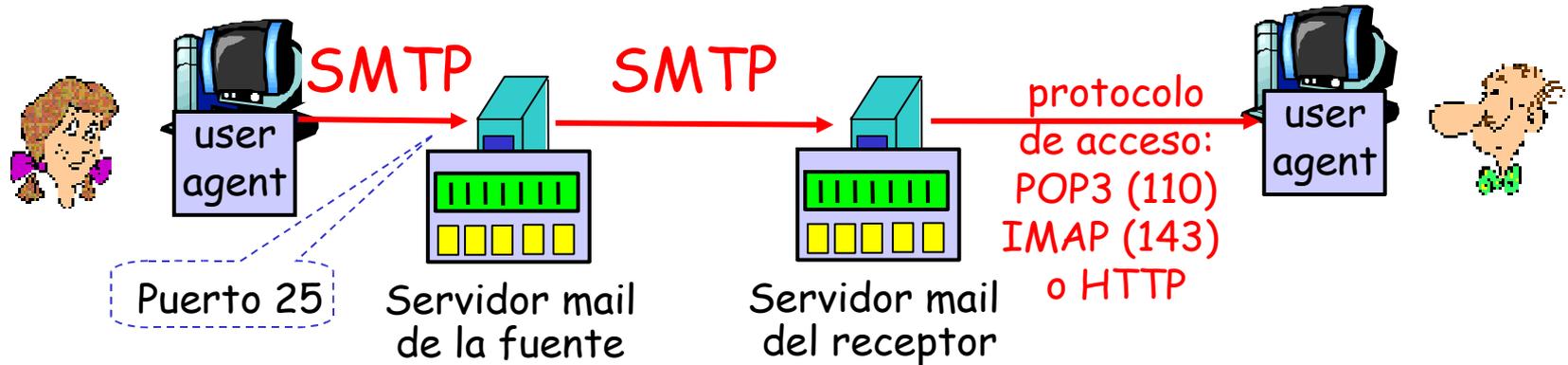
Método de codificación usado

Tipo datos multimedia, subtipo, declaración de parámetros

Datos binarios codificados en base64

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data .....
.....base64 encoded data
```

Protocolos de acceso de correo



- SMTP: permite envío y almacenamiento de correo en servidor del destinatario
- Protocolo de acceso a correo: permite extraer correo desde el servidor
 - POP: Post Office Protocol [RFC 1939]
 - autenticación (agent <-->server) y bajada
 - IMAP: Internet Mail Access Protocol [RFC 1730]
 - Más características (más complejo)
 - Permite manipulación de los mensajes almacenados en el servidor
 - HTTP: Hotmail , Yahoo! Mail, etc.

Protocolo POP3

Fase de autorización

- ❑ Comandos del cliente:
 - user: declara username
 - pass: password
- ❑ Respuestas del servidor:
 - +OK
 - -ERR

Fase transaccional, cliente:

- ❑ list: lista números de mensajes
- ❑ retr: extrae mensajes por su número
- ❑ dele: borra
- ❑ quit

```
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
```

Tamaño del mensaje

POP3 (más) e IMAP

Más sobre POP3

- ❑ Ejemplo previo usa modo "bajar y borrar".
- ❑ Bob no puede releer el correo si cambia el cliente
- ❑ "bajada y conserva": obtiene copia de los mensajes en diferentes clientes.
- ❑ POP3 no mantiene el estado de una sesión a otra ("stateless")

IMAP

- ❑ Mantiene todos los mensajes en un lugar: el servidor
- ❑ Permite que el usuario organice sus correos en carpetas
- ❑ IMAP mantiene el estado del usuario de una sesión a otra:
 - Nombre de carpetas mapeo entre Ids (identificadores) de mensajes y nombres de carpetas.

/* Si usted sabe programar sockets, usted puede escribir un cliente de correo. */

Capítulo 2: Capa Aplicación

- ❑ 2.1 Principios de la aplicaciones de red
- ❑ 2.2 Web y HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Correo Electrónico
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P Compartición de archivos
- ❑ 2.7 Programación de Socket con TCP
- ❑ 2.8 Programación de socket con UDP
- ❑ 2.9 Construcción de un servidor WEB

DNS: Domain Name System (Sistema de nombres de dominio)

Personas: muchos identificadores:

- ROL, RUT, name, # pasaporte

Host y router en Internet:

- Dirección IP (32 bit) - usada para direccionar datagramas (ideal para router)
- "nombre", e.g., www.yahoo.com - son usados por humanos

Q: Quién mapea entre direcciones IP y nombres?

Domain Name System:

- *Base de datos distribuida* implementada en una jerarquía de muchos *servidores de nombres*
- *Protocolo de capa aplicación* permite a host, routers, y servidores de nombre comunicarse para *resolver* nombres (traducción dirección/nombre)
- No está orientado al uso directo de los usuarios.
 - nota: función central de la Internet implementada como protocolo de capa aplicación
 - La idea es dejar la complejidad en la "periferia" de la red. Routers almacenan información de DNS

DNS

Servicios DNS

- ❑ Traducción de nombre de host a dirección IP
- ❑ Alias para host
 - Nombre canónico o alias
 - Nombre canónico: CNAME en RFC 1035
- ❑ Alias para servidor de correo
- ❑ Distribución de carga

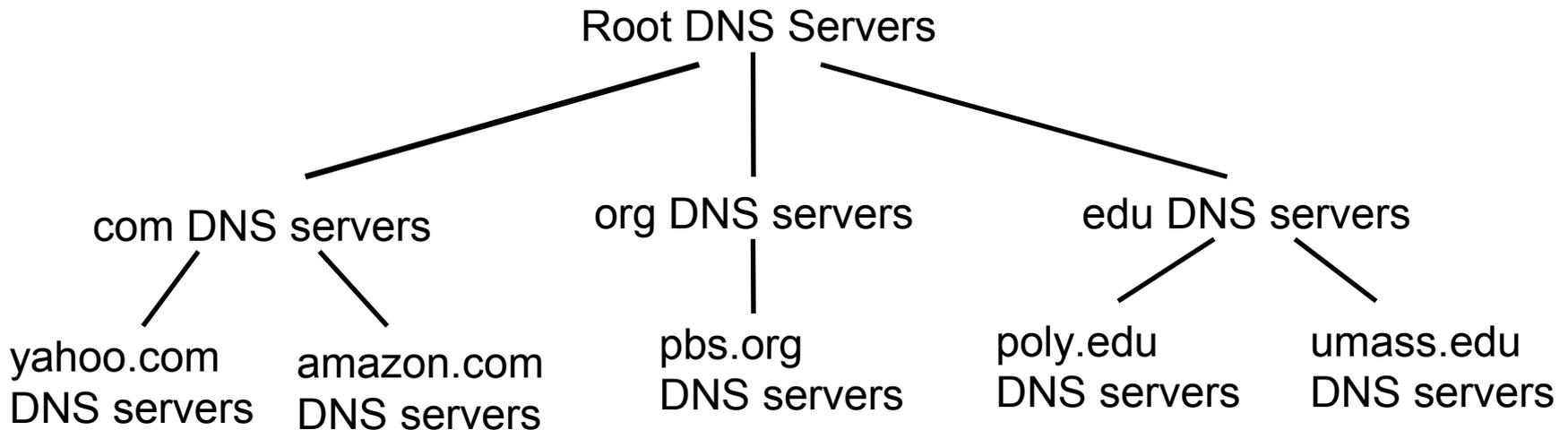
Servidores Web replicados: conjunto de direcciones IP para un nombre canónico (e.g. relay1.west-coast.amazon.com), servidores DNS rota entre direcciones IP

¿Por qué no centralizar DNS?

- ❑ Único punto de falla
- ❑ Volumen de tráfico, muchos necesitan el DNS
- ❑ Sería una base de datos centralizada distante con grandes retardos de acceso.
- ❑ Mantenición, es mejor que cada dominio gestione sus nombres

Porque no es escalable!

Base de datos jerárquica y distribuida



Cliente desea IP de www.amazon.com; 1^{ra} aprox. :

- ❑ Cliente consulta al servidor raíz para encontrar servidor DNS de com
- ❑ Cliente consulta servidor DNS TLD (Top Level Domain) de com para obtener servidor DNS de amazon.com
- ❑ Cliente consulta servidor DNS amazon.com para obtener dirección IP de www.amazon.com

DNS: servidores de nombre raíz

- Son contactados por servidor de nombre local cuando no puede resolver un nombre
- Servidor nombre raíz:
 - Contacta servidor de nombre autoritario de la zona superior (e.g. com) si mapeo del nombre es desconocido para él
 - Obtiene mapeo (propio o desde otro servidor raíz)
 - Retorna mapeo al servidor de nombre local



TLD y Servidores Autoritarios

- **Top-level domain (TLD) servers:** responsable por com, org, net, edu, etc., y todos los dominios superiores de cada país: uk, fr, ca, jp, cl, etc..
 - Network solutions mantiene servidores para el TLD de com
 - Educause para el TLD de edu
 - Nic (network information center) para el TLD de cl (www.nic.cl)
- **Servidores DNS autoritarios:** son servidores DNS de las organizaciones y proveen mapeos autoritarios entre hostname e IP (e.g., Web y mail).
 - Éstos pueden ser mantenidos por la organización o el proveedor de servicio

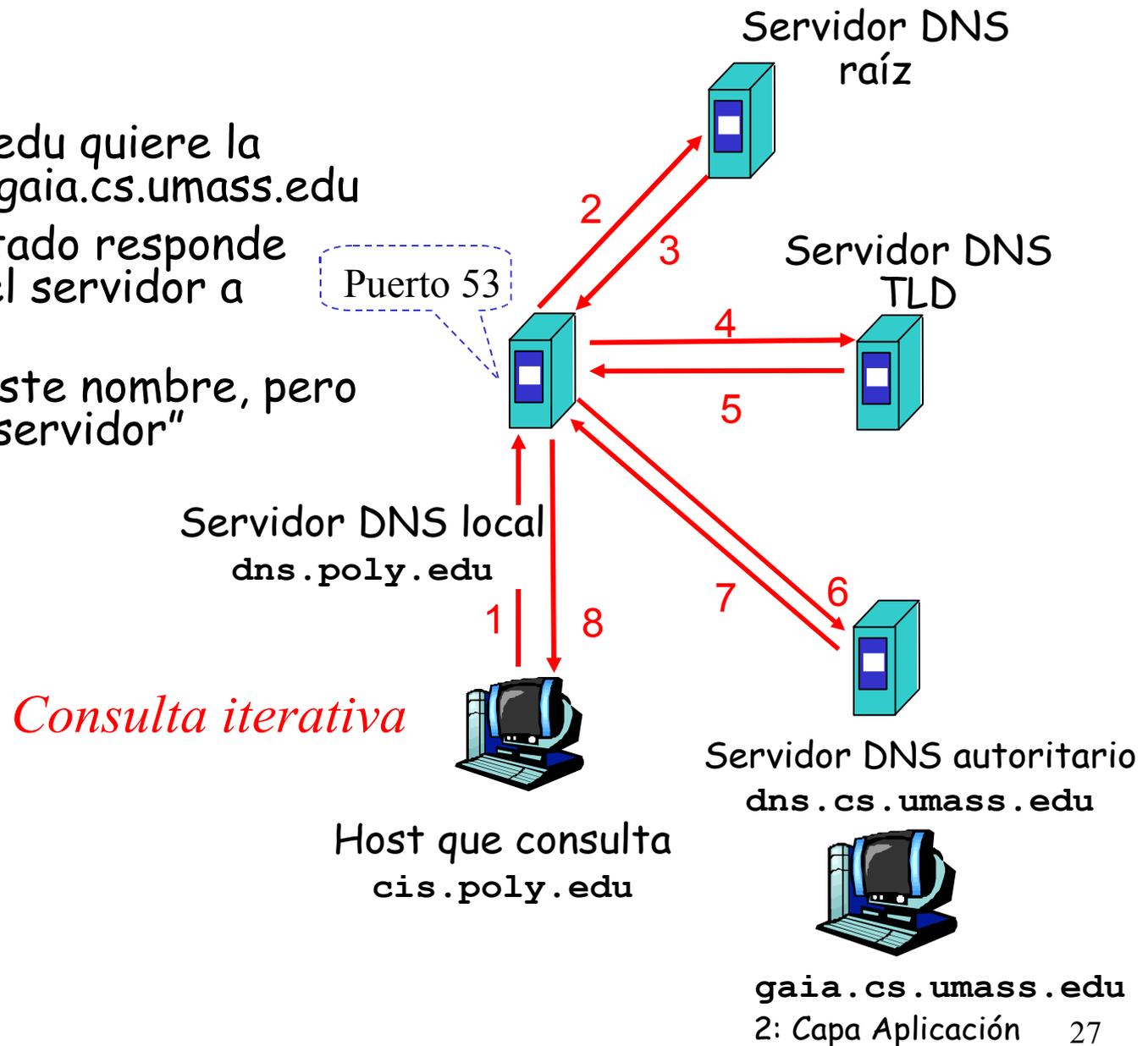
Servidor de nombre local

- ❑ No pertenece estrictamente a la jerarquía
- ❑ Cada ISP (ISP residencial, compañía, universidad) tiene uno.
 - También son llamados "servidor de nombre por omisión" (default name server)
- ❑ Cuando un host hace una consulta DNS, ésta es enviada a su servidor DNS local
 - Actúa como proxy, re-envía consulta dentro de la jerarquía.

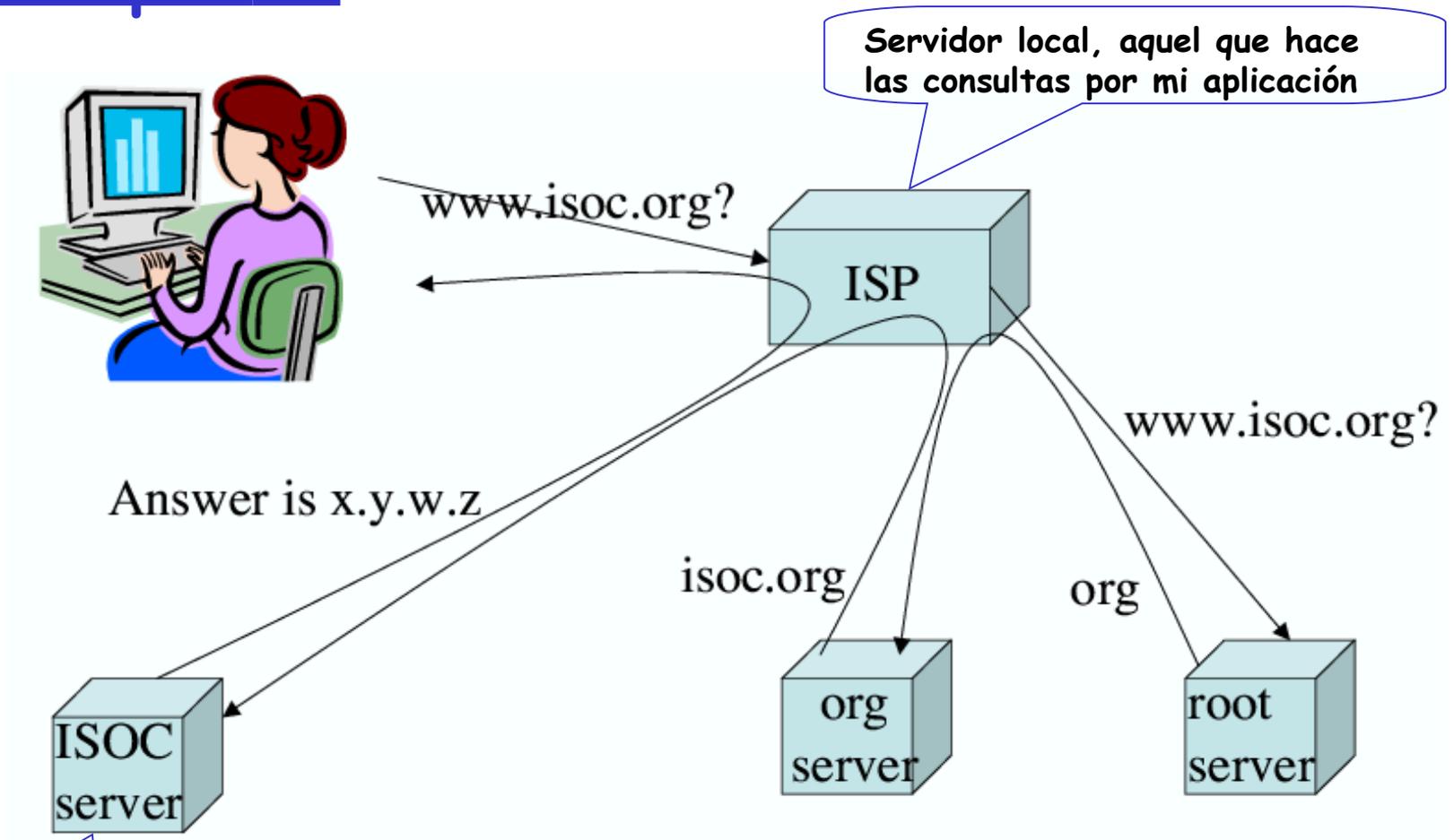
Ejemplo 1

Consulta iterativa:

- ❑ Host en cis.poly.edu quiere la dirección IP de gaia.cs.umass.edu
- ❑ Servidor contactado responde con el nombre del servidor a contactar
- ❑ "Yo no conozco este nombre, pero pregunta a este servidor"



Ejemplo 2



Servidor autoritario, aquel que define el mapeo nombre <-> IP

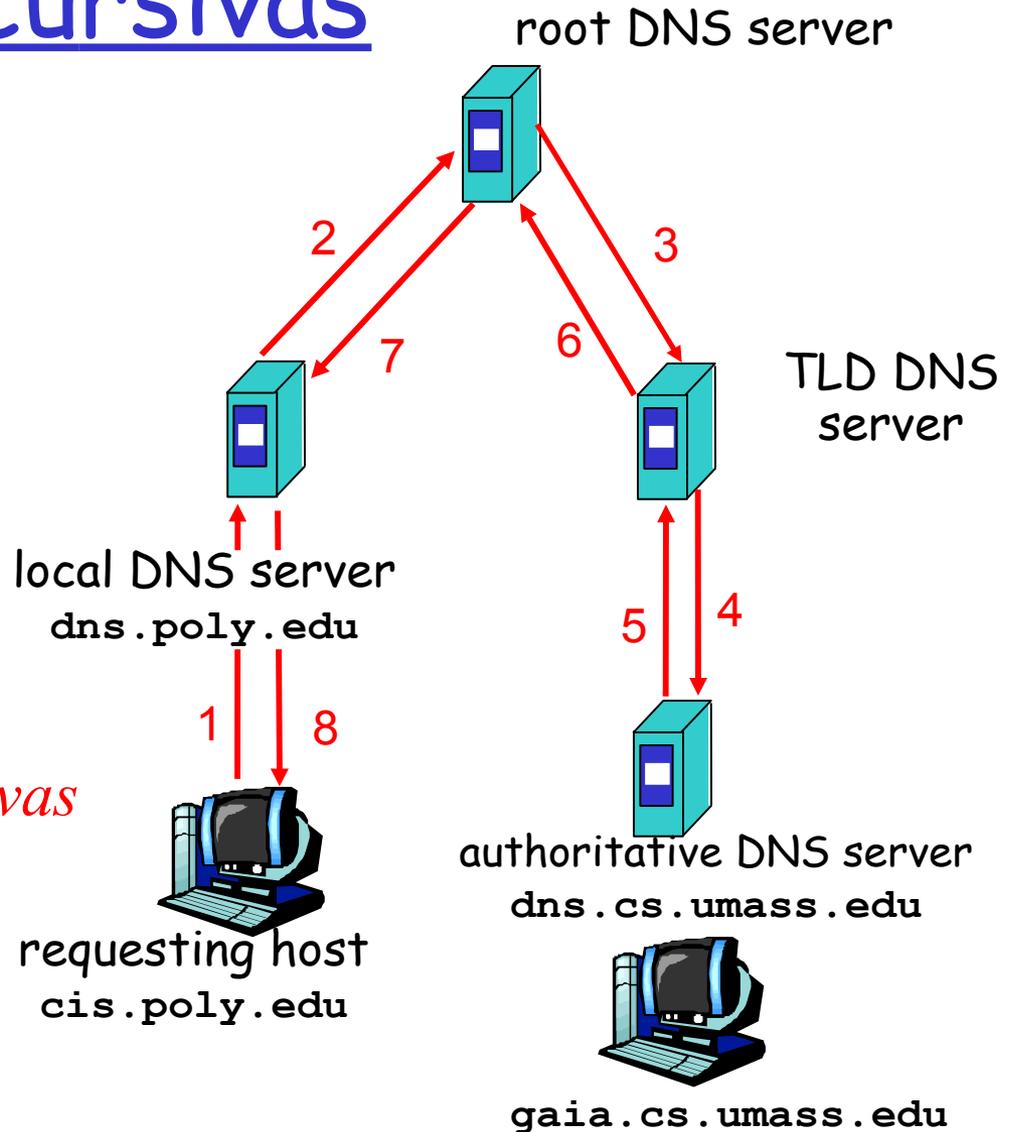
Consulta iterativa

Consultas Recursivas

Consulta recursiva :

- Pone la carga de la resolución de nombre al servidor contactado.
- ¿Qué pasa en situaciones de alta carga?

Consultas Recursivas



Ejemplo

- ❑ Hacer algo del tipo:
\$ nslookup www.elo.utfsm.cl
- ❑ Luego:
\$ nslookup 200.1.17.5
- ❑ Finalmente:
\$ nslookup www.google.com
- ❑ y
\$ nslookup 64.233.161.99
- ❑ Estando en aragorn hacer:
\$ nslookup 200.1.17.195

DNS: Cache y actualización de registros

- Una vez que un servidor de nombre conoce un mapeo, éste *guarda* (caches) el mapeo
 - Las entradas del cache expiran (desaparecen) después de algún tiempo
 - Servidores TLD típicamente están en cache de los servidores de nombre locales
 - Así los servidores de nombre raíz no son visitados con frecuencia
- Mecanismos de Actualización/notificación están bajo diseño por el IETF (Internet Engineering Task Force)
 - RFC 2136
 - <http://www.ietf.org/html.charters/dnsind-charter.html>

Registros DNS

DNS: es una base de datos distribuida que almacena registros de recursos (resource records, **RR**)

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

□ Type=A

- name es un hostname
- value es una dirección IP

□ Type=NS

- name es un dominio (e.g. foo.com)
- value es la dirección IP (nombre) del servidor autoritario que sabe cómo obtener las direcciones IP de este dominio.

□ Type=CNAME

- name es un alias para algún nombre real

www.ibm.com es realmente

servereast.backup2.ibm.com

value es el nombre real

□ Type=MX

value es el nombre del servidor de correo asociado con name

Inserción de registros en DNS

- Ejemplo: Recién se crea una empresa "Network Utopia"
- Debemos registrar el nombre networkutopia.com en un **administrador de dominio** (e.g., Network Solutions)
 - Necesitamos proveer el nombre y la dirección IP de nuestro servidor de nombre autoritario (primario y secundario)
 - Administrador del dominio inserta dos RRs en el servidor TLD com:

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

- Incorporar en el servidor autoritario un registro Tipo A para www.networkutopia.com y un registro Tipo MX para networkutopia.com
- **¿Cómo la gente obtiene la dirección IP de nuestro Servidor WEB?**
- En Chile debemos acceder al NIC Chile para arrendar un nombre de dominio.

Capítulo 2: Capa Aplicación

- ❑ 2.1 Principios de la aplicaciones de red
- ❑ 2.2 Web y HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Correo Electrónico
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS
- ❑ 2.6 P2P Compartición de archivos
- ❑ 2.7 Programación de Socket con TCP
- ❑ 2.8 Programación de socket con UDP
- ❑ 2.9 Construcción de un servidor WEB