# Capa de Aplicación

Redes de Computadores

FIEC04705

Sesión 22

# Agenda

- Terminología
- Protocolos para envío de correo electrónico
- Demostración de Wireshark
- DNS
- P2P

# Terminología

# Terminología

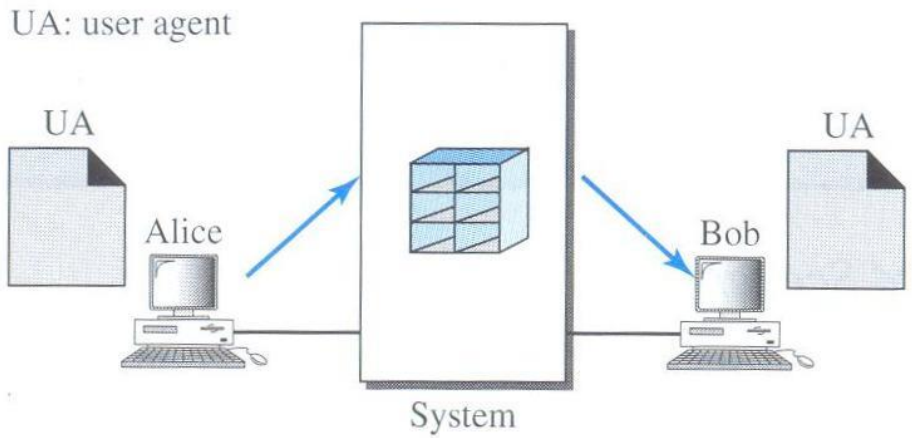
- **Multipurpose Internet Mail Extensions (MIME)** es un protocolo suplementario que **permite enviar datos no ASCII** a través de un e-mail. Se puede pensar que es un conjunto de funciones de software que transforman dato no ASCII (flujo de bits) a ASCII y viceversa.

# Protocolos para envío de correo electrónico

# Arquitectura de un e-mail

- **Primer escenario**: cuando el remitente y el destinatario de un email están en el mismo sistema, solo se requiere dos user agents.

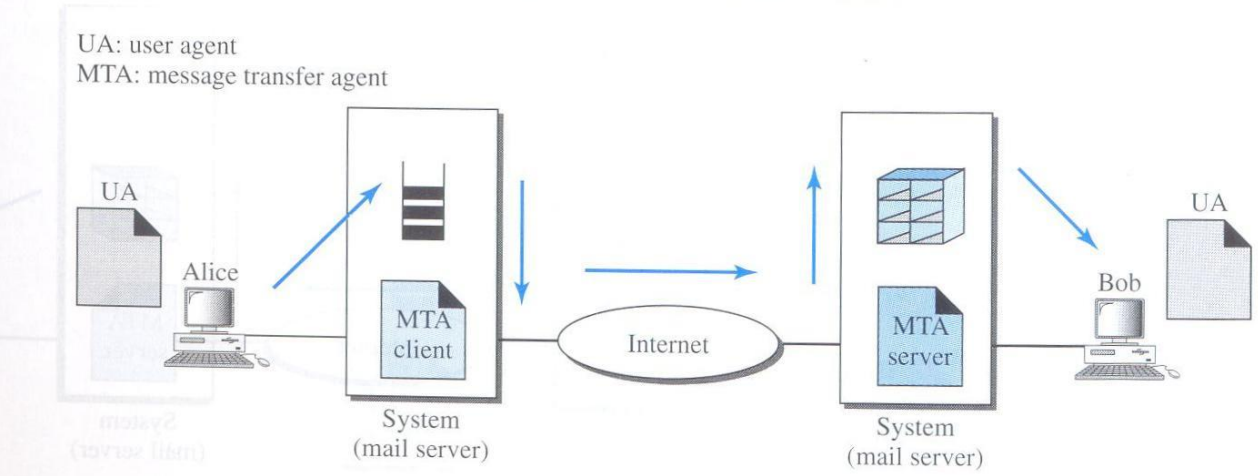
**Figure 26.6** *First scenario in electronic mail*



# Arquitectura de un e-mail

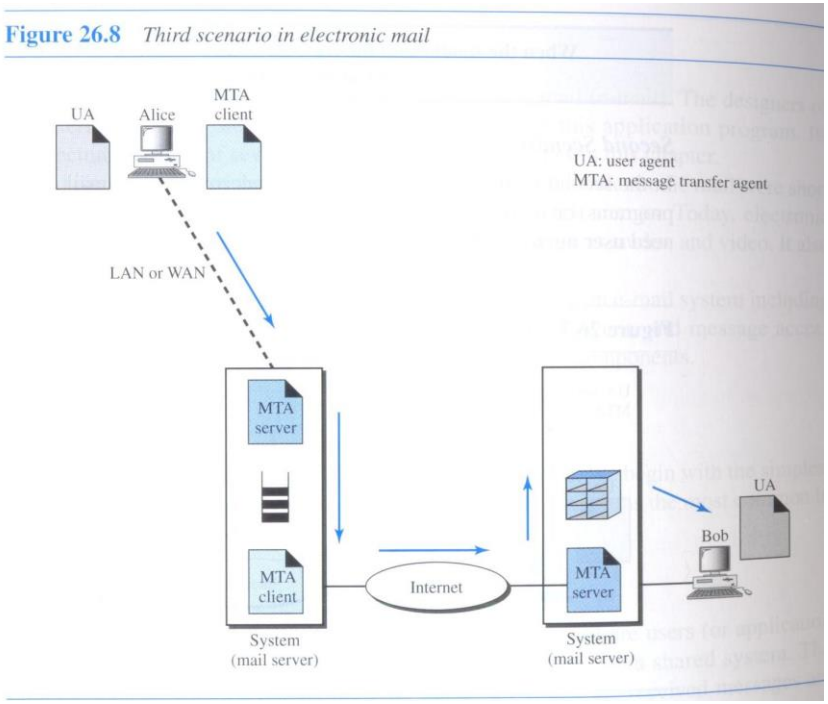
- **Segundo escenario**: cuando el remitente y el destinatario de un e-mail están en diferentes sistemas, se requiere dos UAs y un par de MTAs (cliente y servidor)

**Figure 26.7** *Second scenario in electronic mail*



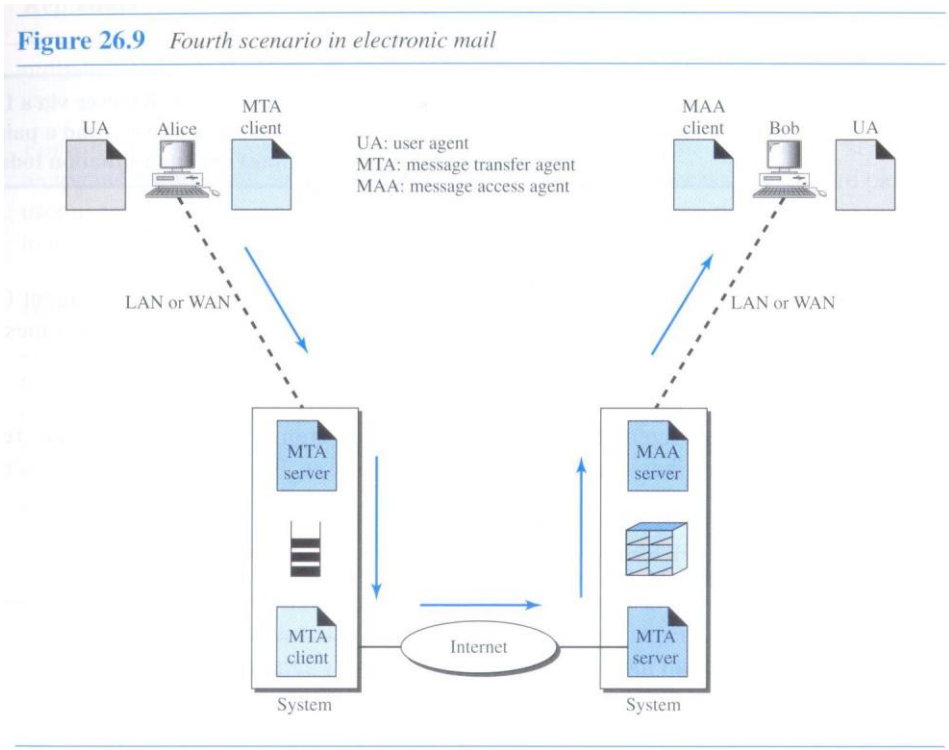
# Arquitectura de un e-mail

- **Tercer escenario**: Cuando el remitente está conectado al servidor de correo por medio de una LAN o WAN, se requiere dos UAs y dos pares de MTAs (cliente y servidor)



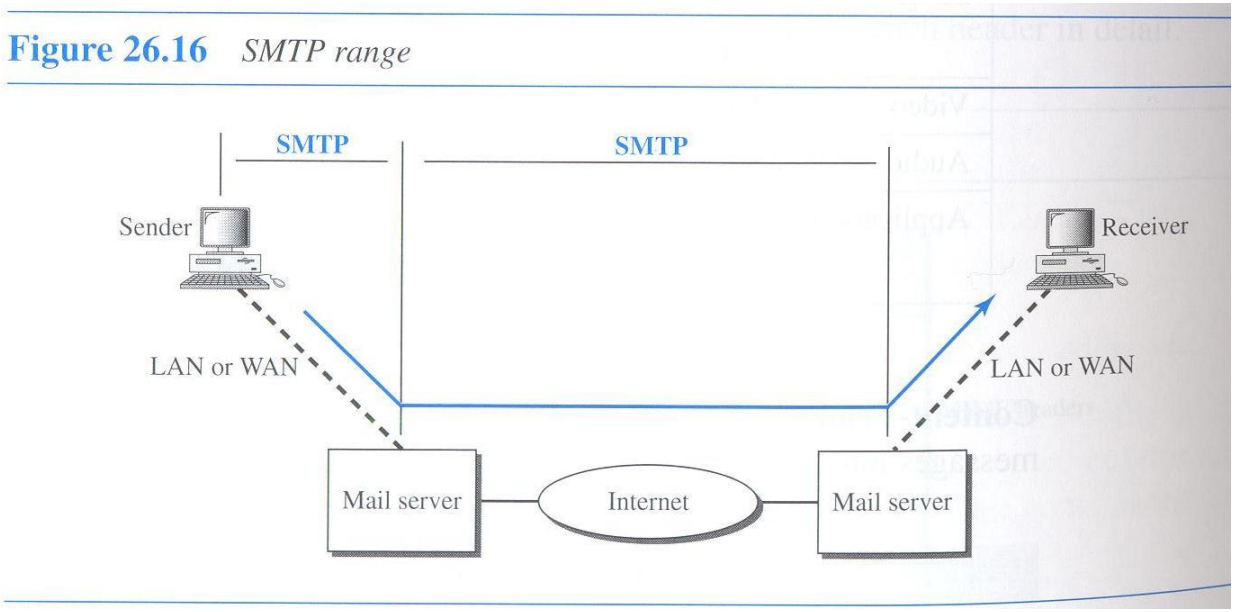
# Arquitectura de un e-mail

- **Cuarto escenario**: Cuando el remitente y el destinatario están conectados al servidor de correo por medio de una LAN o WAN, se requiere dos UAs, dos pares de MTAs, y un par de MAAs.



# SMTP

- En el cuarto escenario, SMTP es utilizado dos veces: entre el remitente y su servidor de correo y entre los dos servidores de correo.
- Otro protocolo es requerido entre el destinatario y su servidor de correo.

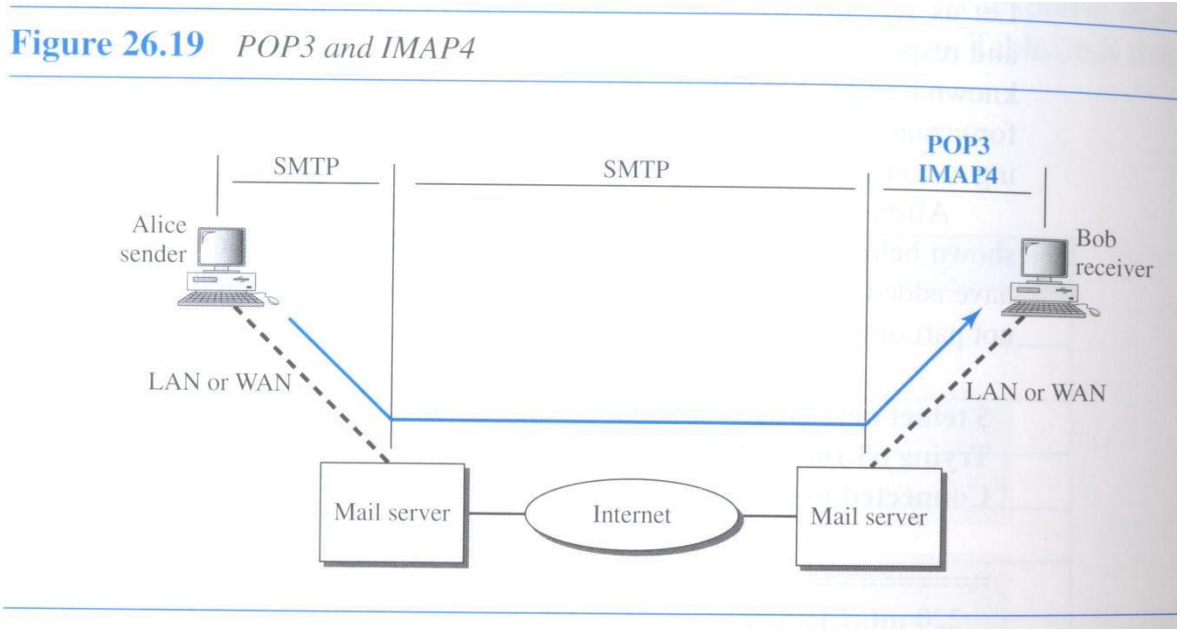


# Simple Mail Transfer Protocol - SMTP

- **User Agent (UA)**: es un programa que compone, lee, responde y reeenvía mensajes. Existen dos tipos:
  - Command-Driven: mail, pine elm
  - GUI-Based: Eudora, Netscape, Outlook
- **Message Transfer Agent (MTA)**: un componente SMTP que transfiere el mensaje a través del Internet.
- El protocolo que define el cliente y servidor MTA en el Internet es conocido como SMTP.

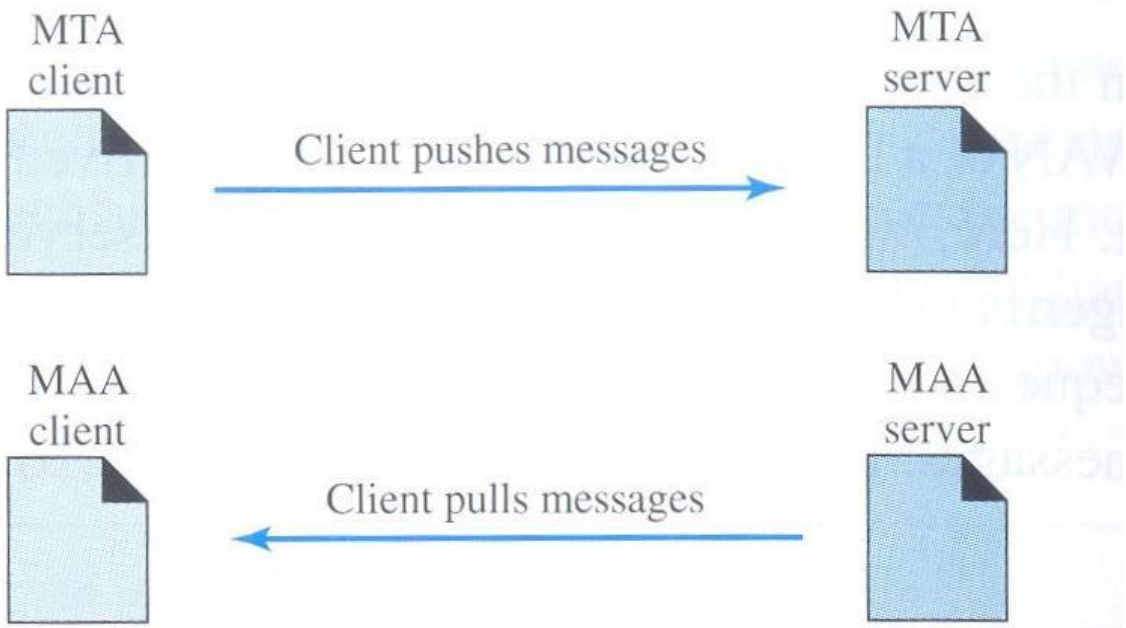
# Message Access Agents (MAA): POP e IMAP

- SMTP es un protocolo *push*: coloca el mensaje del cliente en el servidor.
- En el tercer paso se requiere un protocolo pull: el cliente debe traer los mensajes del servidor. Se utiliza uno de los Message Access Protocols: POP3 o IMAP4.



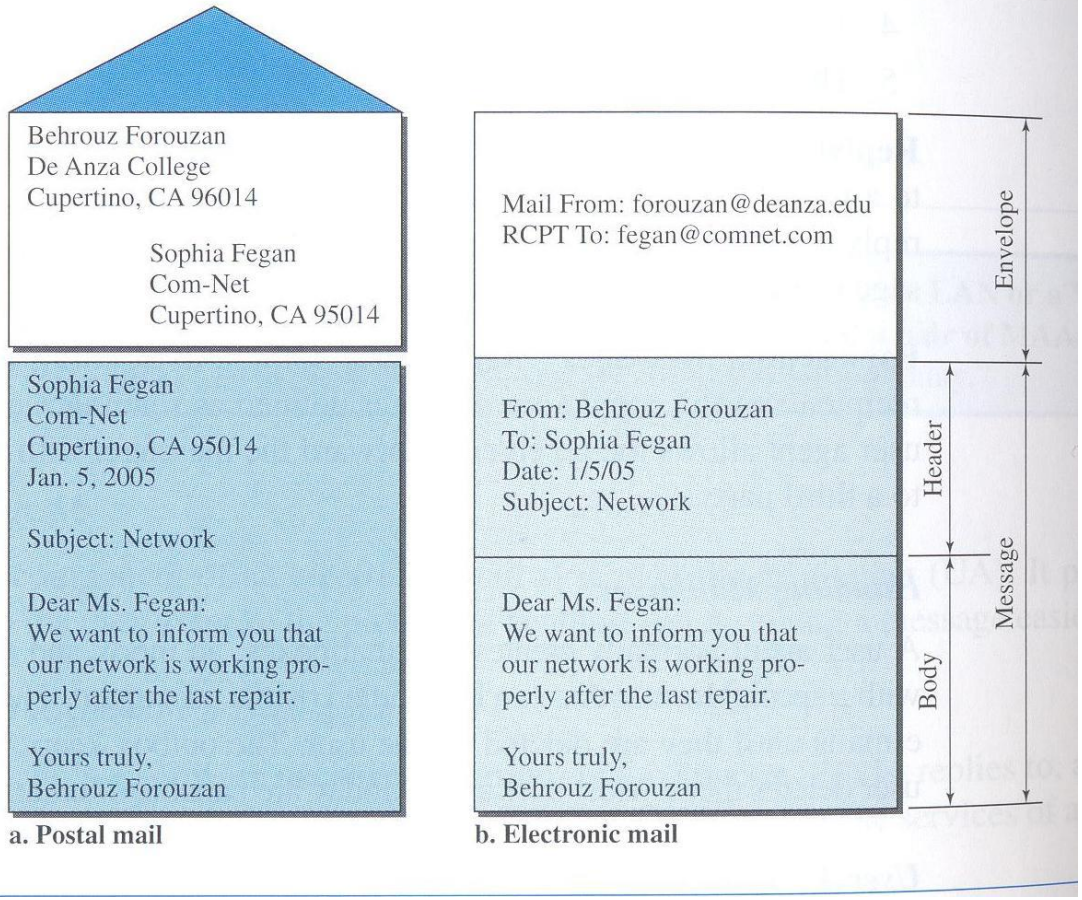
# MTA vs. MAA

**Figure 26.10** *Push versus pull in electronic email*



# Formato de un e-mail

**Figure 26.12** *Format of an e-mail*

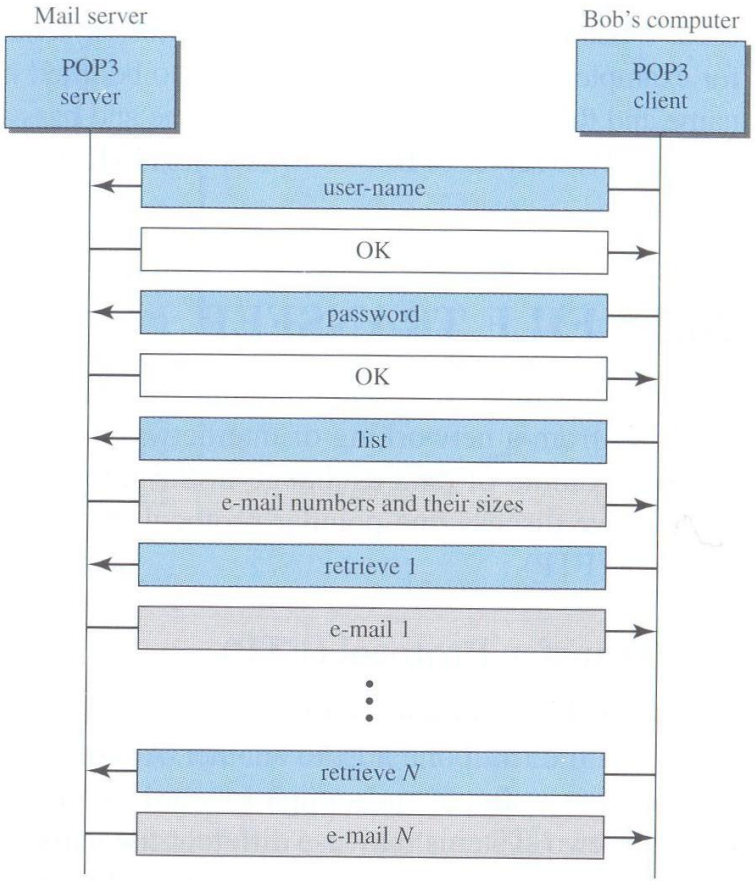


# Post Office Protocol, version 3 - POP3

- El cliente del software POP3 es instalado en el computador destinatario
- El servidor del software POP3 es instalado en el servidor.
- POP3 tiene dos modos:
  - Delete: el buzón es borrado luego de traer los corrreos.
  - Keep: el correo es leido pero mantenido en el buzón para posteriores recuperaciones.

# Post Office Protocol, version 3 - POP3

**Figure 26.20** *The exchange of commands and responses in POP3*



# Internet Mail Access Protocol, version 4 - IMAP4

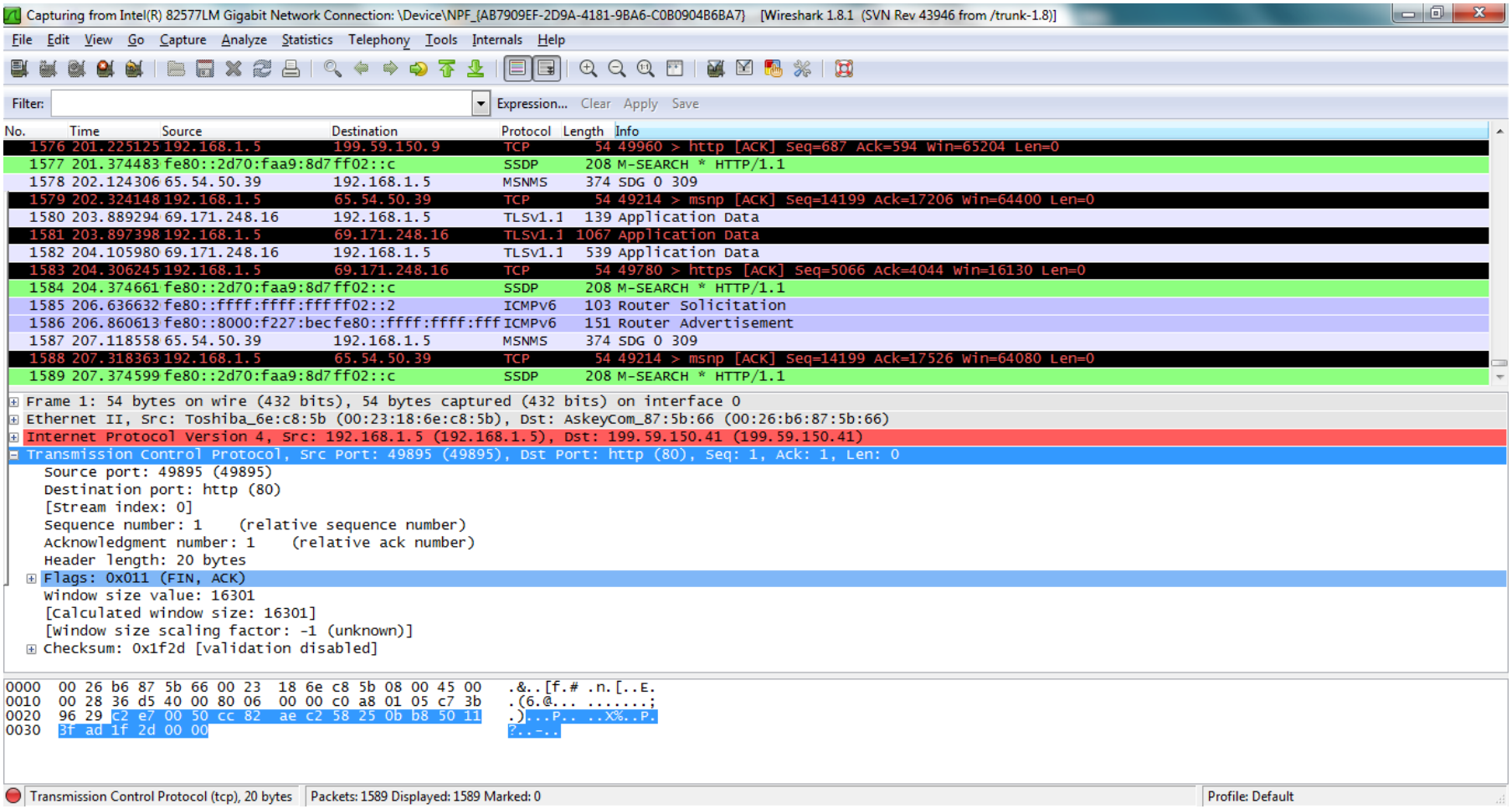
- Similar a POP3 pero con mejores características:
  - Un usuario puede chequear la cabecera del email antes de descargarlo
  - Un usuario puede buscar un texto en el contenido de un email antes de descargarlo
  - Un usuario puede descargar un correo parcialmente. Por ejemplo, omitir un contenido multimedia
  - Un usuario puede crear, borrar o renombrar buzones en el servidor de correo
  - Un usuario puede crear una jerarquía de buzones en una carpeta para almacenamiento de correos.

# Demostración de Wireshark

# Wireshark

- Wireshark es un analizador de protocolos de red que permite capturar y explorar interactivamente el tráfico en una red de computadores.
- Características:
  - Inspección de cientos de protocolos
  - Captura en línea y análisis offline
  - Explorador de paquetes en tres paneles
  - Lectura y escritura de varios formatos de archivos de captura
  - Soporte para desencriptación de muchos protocolos
  - Muchas otras...

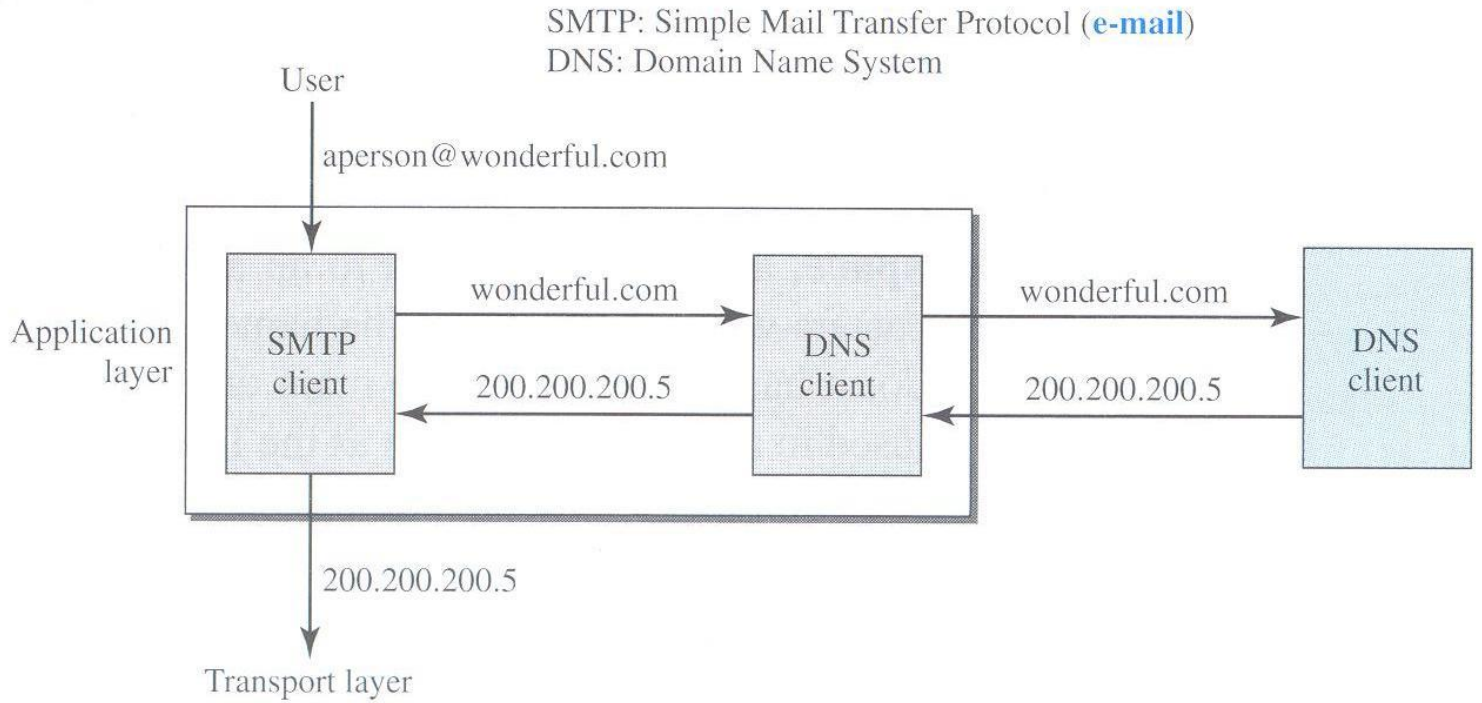
# Demo Wireshark



# DNS

# Domain Name System - DNS

**Figure 25.1** *Example of using the DNS service*

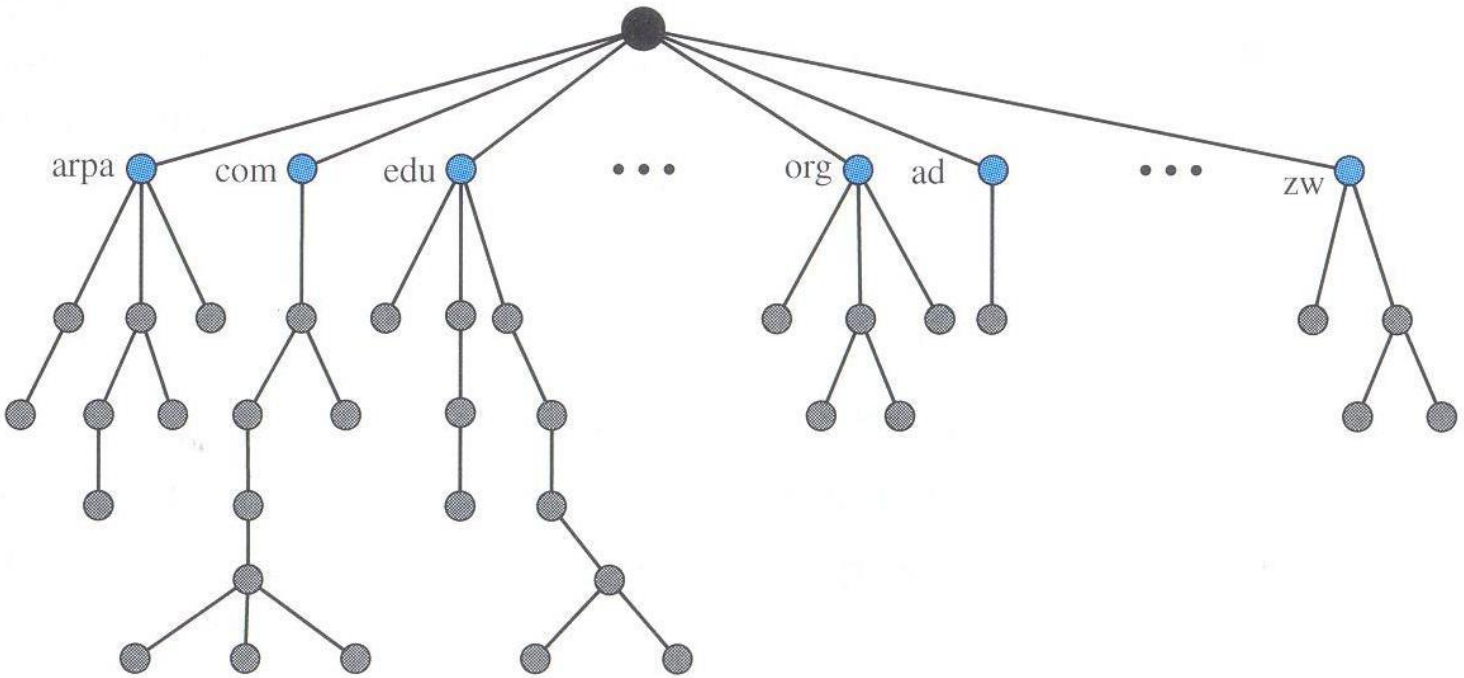


# DNS

- Sirve para mapear direcciones IP a nombres y viceversa.
- DNS está diseñado como una aplicación cliente/servidor.
- El espacio de nombres tiene una estructura jerárquica en forma de árbol que se denomina *domain name space*.
- El domain name es una secuencia de labels (nombre del nodo en el árbol) separado por puntos.

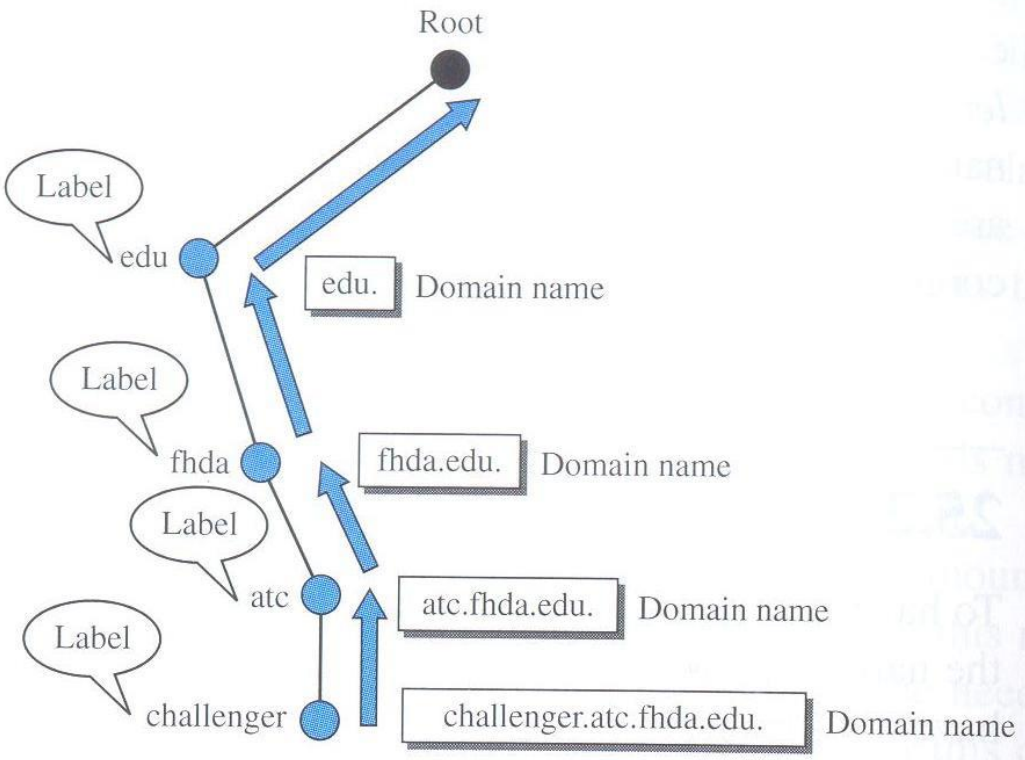
# DNS

**Figure 25.2** *Domain name space*



# Domain name y labels

**Figure 25.3** *Domain names and labels*



# Zonas y dominios

**Figure 25.6** *Hierarchy of name servers*

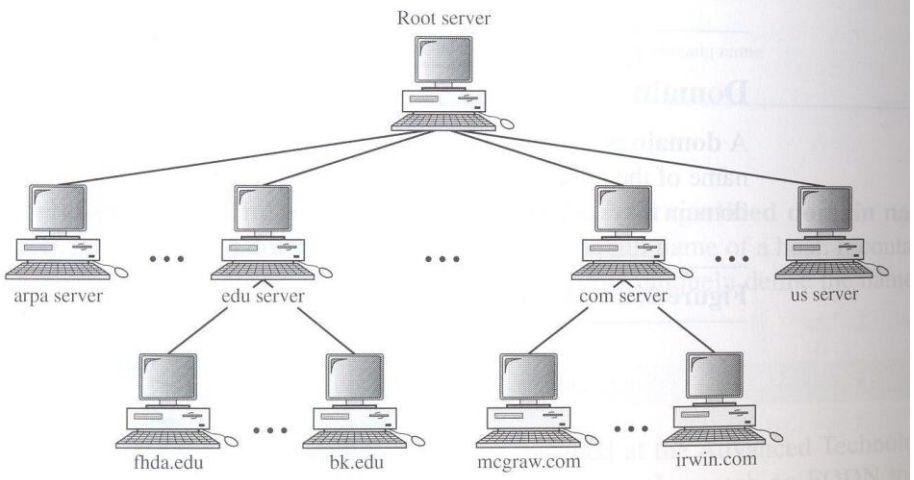
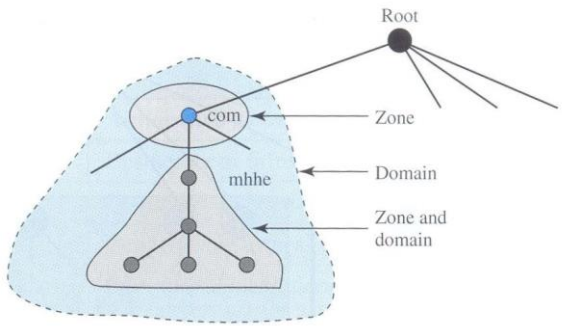



**Figure 25.7** *Zones and domains*



# DNS en el Internet

**Figure 25.8** *DNS used in the Internet*

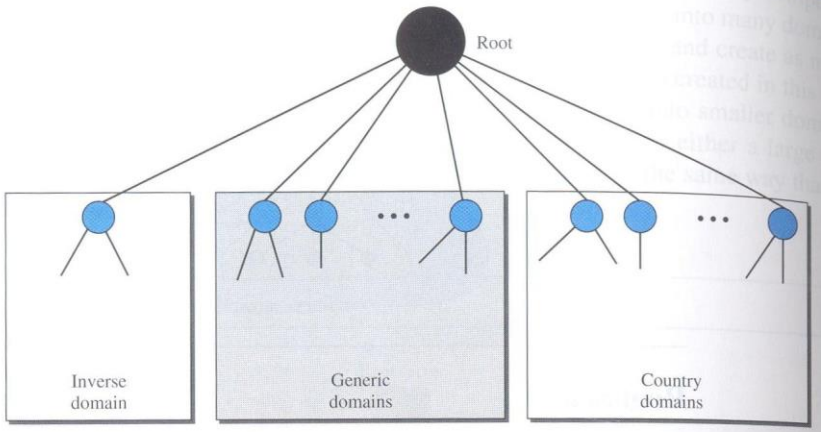
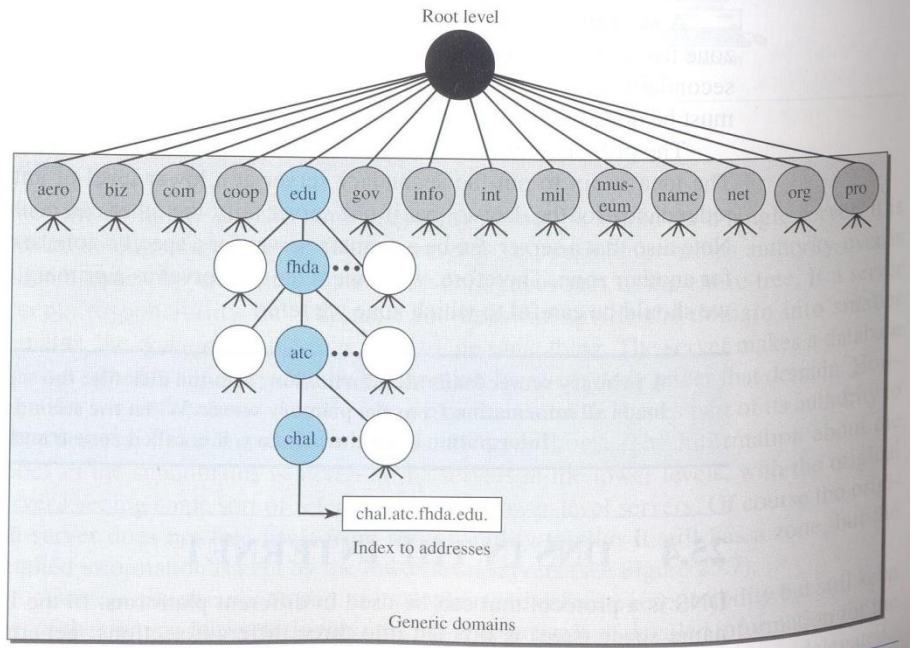



**Figure 25.9** *Generic domains*

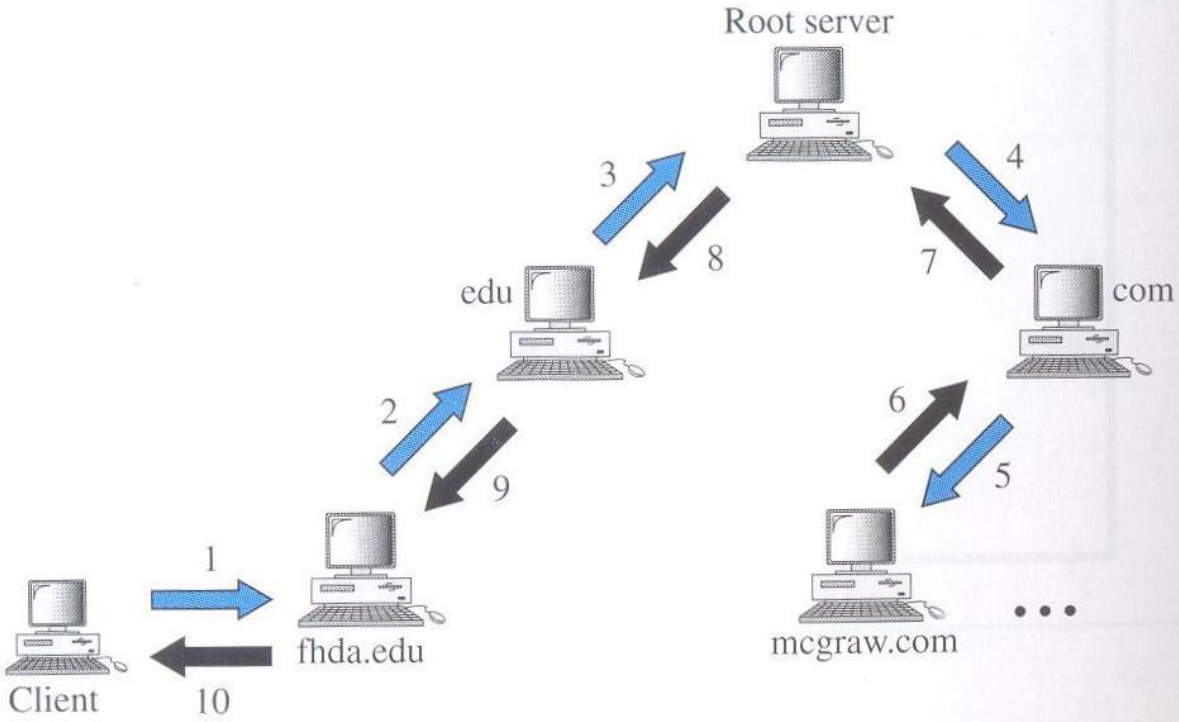


# Resolución

- Resolución es mapear un nombre a una direción o viceversa.
- *Resolver* es el DNS cliente.
- El resolver accede al servidor DNS más próximo. Si el servidor tiene la información, el responde al resolver; de otra manera, refiere el resolver a otros servidores o pregunta a otros servidores para proveer la información.
- Existen dos tipos: Recursive vs. Iterative

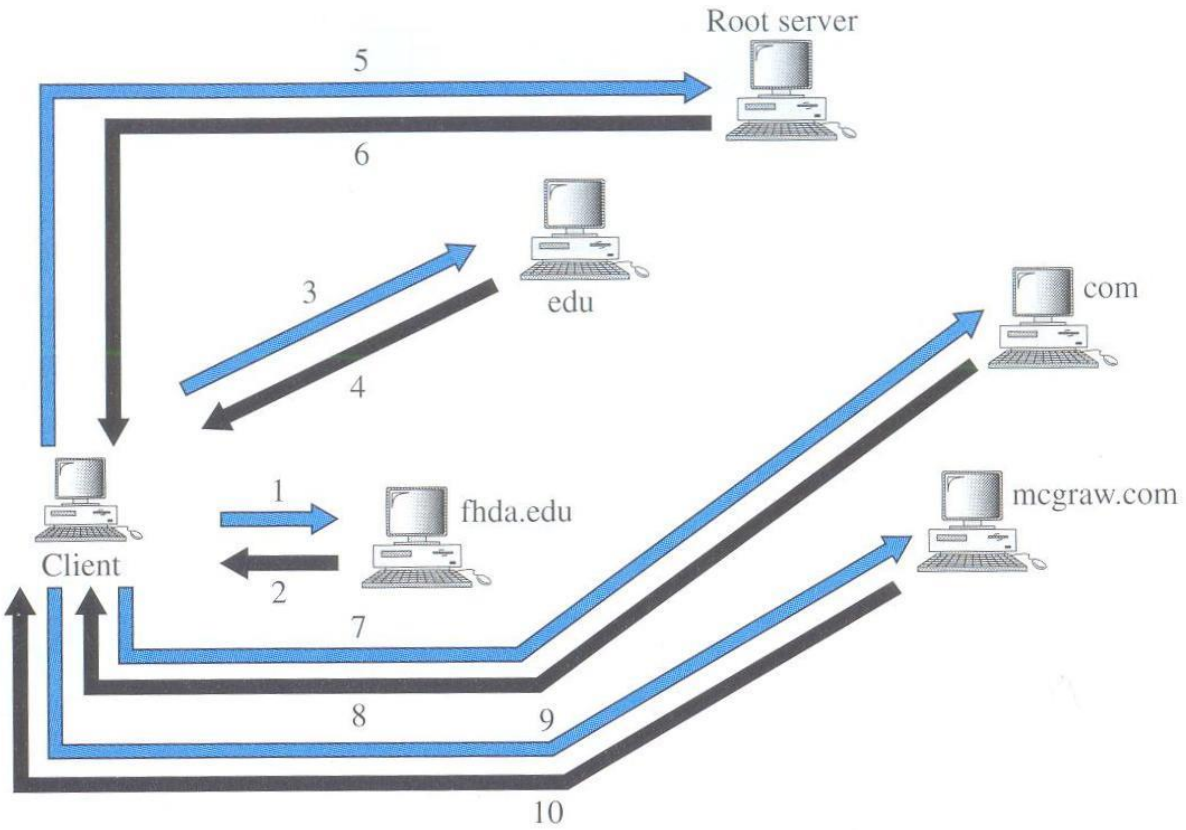
# Recursive resolution

**Figure 25.12** *Recursive resolution*



# Iterative resolution

**Figure 25.13** *Iterative resolution*



# DNS Caching

- Cuando un servidor pregunta por un mapeo de otro servidor y recibe la respuesta, este almacena la información en su memoria caché antes de enviarla al cliente.
- Caché mejora el tiempo de respuesta de la resolución, pero se presta para ataques de potenciales intrusos.

# Mensajes DNS

- DNS tiene dos tipos de mensajes:
  - Query
  - Response

**Figure 25.14** *Query and response messages*

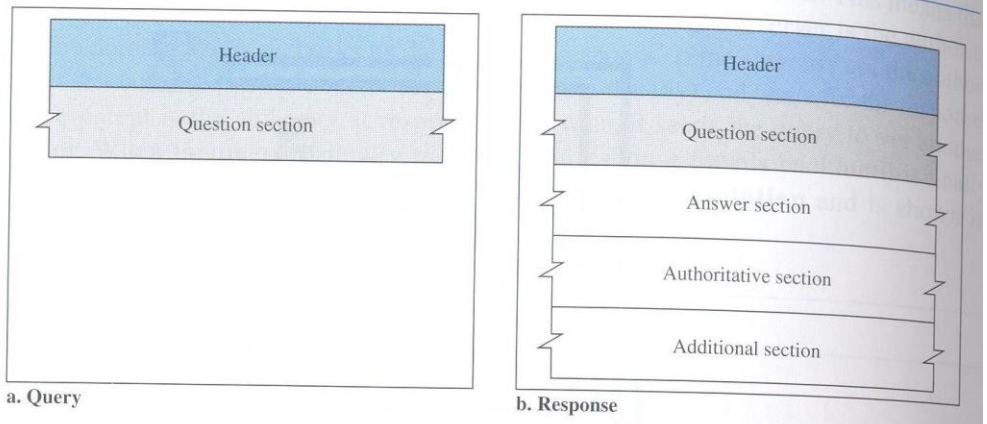



**Figure 25.15** *Header format*

| Identification | Flags |
|---|---|
| Number of question records | Number of answer records (all 0s in query message) |
| Number of authoritative records (all 0s in query message) | Number of additional records (all 0s in query message) |

# DNS Query

```
13:30:08.018705 IP 10.4.130.214.51103 > 147.188.128.102.53: 1313+ A?
google.com. (28)
13:30:08.047483 IP 147.188.128.102.53 > 10.4.130.214.51103: 1313 5/13/0
A 74.125.230.114, A 74.125.230.115, A 74.125.230.116, A 74.125.230.112,
A 74.125.230.113 (319)
```

# DNS Response

```
$ dig google.com
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 34072
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.            IN   A

;; ANSWER SECTION:
google.com.        300  IN   A    209.85.143.99
google.com.        300  IN   A    209.85.143.104

;; AUTHORITY SECTION:
google.com.        172800    IN   NS   ns4.google.com.
google.com.        172800    IN   NS   ns1.google.com.
google.com.        172800    IN   NS   ns2.google.com.
google.com.        172800    IN   NS   ns3.google.com.

;; Query time: 21 msec
;; SERVER: 147.188.192.4#53(147.188.192.4)
;; WHEN: Wed Feb  2 18:29:31 2011
;; MSG SIZE  rcvd: 132
```

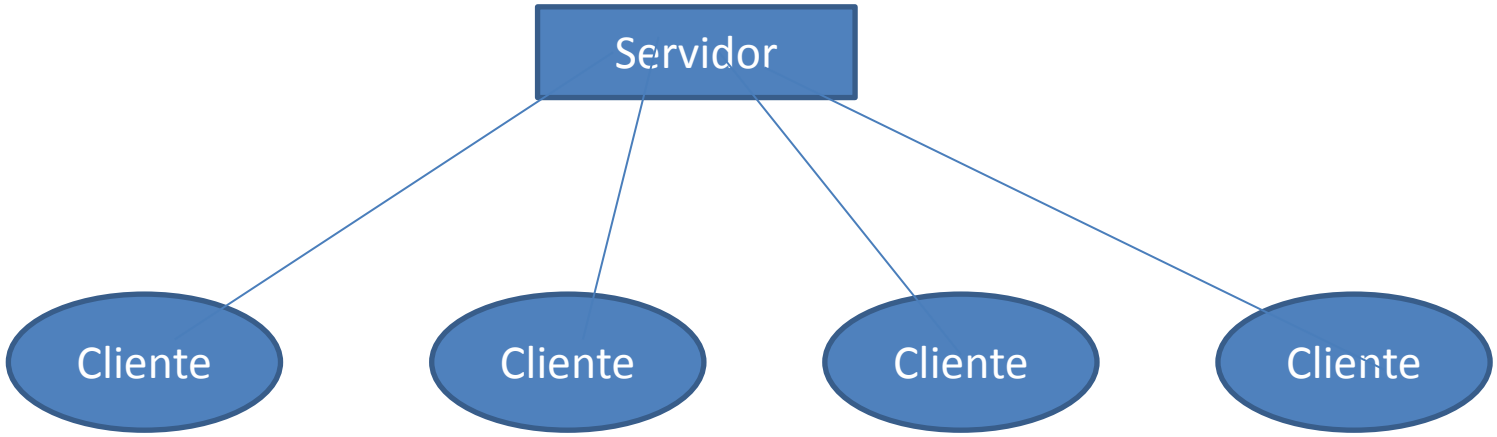
# Registrars

- Un *registrar* es una entidad comercialmente acreditada por *Internet Corporation for Assigned Names and Numbers (ICANN)* que agrega nuevos dominios al DNS.
- Un *registrar* primero verifica que el nombre de dominio requerido es único y luego lo ingresa en la base de datos DNS.
- Una organización debe entregar al *registrar* el nombre del servidor (Por ejemplo: ws.wonderful.com) y la dirección IP del servidor (200.200.200.5).

# P2P

# Cliente Servidor

- Modelo cliente-Servidor:
  - Trabaja muy bien la mayor parte del tiempo
  - Sin embargo:
    - Pone toda la carga en un servidor
    - Si el servidor falla, todo lo demás también



# Peer to Peer

- No hay servidor central
- Cada peer actúa como cliente y como servidor
- Distribuye la carga
- Muchos diferentes tipos de redes P2P:
  - Centralizado vs. Descentralizado
  - Fully connected vs. Not fully connected

# P2P Centralizado

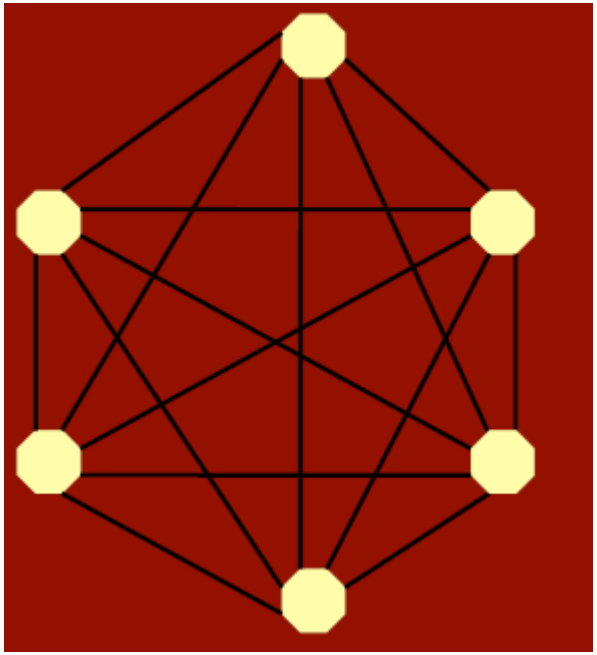
- P2P centralizado tiene un coordinador que dirije la actividad de los peers
- El uso de un coordinador facilita el inicio y búsqueda de la red
- El coordinador puede controlar la red
- Único punto de falla (Single point of failure)
- Ejemplos: Napster, BBC iPlayer (P2P version)

# P2P Descentralizado

- No tiene un único coordinador
- Mucho más difícil de configurar
- No hay un punto único de fallo
- Mayor privacidad
- Ejemplo: Gnutella

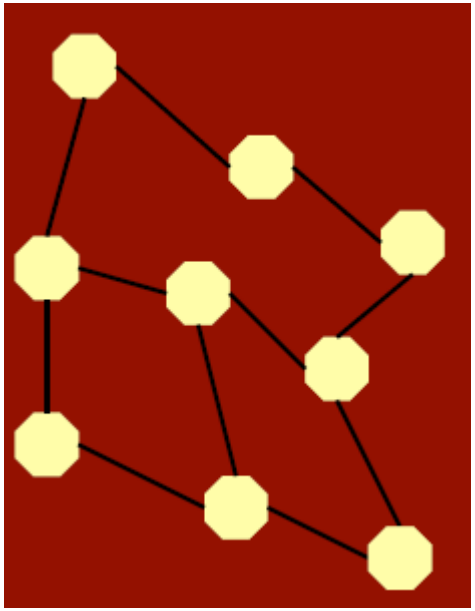
# P2P Fully Connected

- Cada nodo se conecta a todos los otros nodos
- Hace sencillo el paso de mensajes
- No es escalable



# P2P Not Fully Connected

- Cada nodo se conecta solo a unos pocos
- Más común
- Escalable: fácil de mantener
- No tan eficiente



# Redes P2P para compartir archivos

- Napster
  - La primera gran red de compartición de archivos
  - En línea entre los años 1999-2001
  - Coordinador central
- Gnutella
  - La red para compartir archivos más popular a mediados de los 2000s
  - Sin coordinador central
  - Red no estructurada
- BitTorrent
  - Muchas pequeñas redes P2P
  - Un  seguimiento individual para coordinar cada descarga
  - Recompensas para quienes suben archivos

# Puntos para recordar

- SMTP, POP3, IMAP4
- Características de Wireshark
- DNS
- Tipos de redes peer to peer

# Próxima Sesión

- Seguridad en redes de computadores