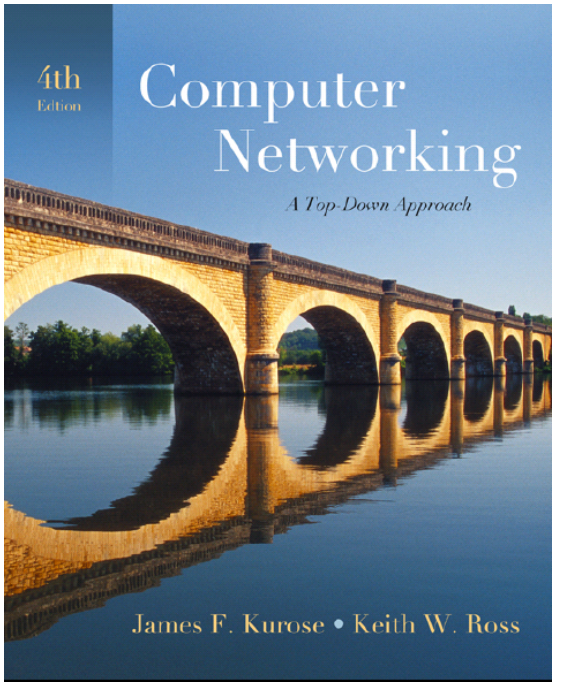
Introducción a las redes de Computadoras
Capítulo 2
Clase 2



Computer Networking: A Top Down Approach, 4th edition.
Jim Kurose, Keith Ross
Addison-Wesley, July 2007.

Nota acerca de las transparencias del curso:

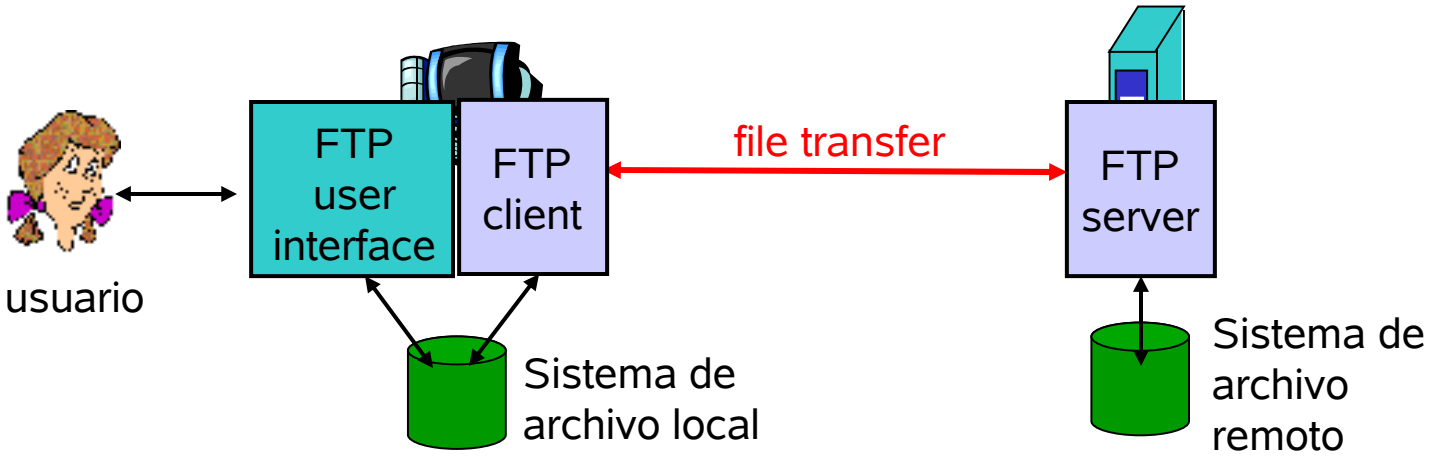
Estas transparencias están basadas en el sitio web que acompaña el libro, y

han sido modificadas por los docentes del curso.

Capítulo 2: Capa de Aplicación

- 2.1 Principles of network applications
- 2.2 Web y HTTP
- 2.3 FTP
- 2.4 Correo Electronico
  - SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 Aplicaciones P2P
- 2.7 Programación de Socket con TCP
- 2.8 Programación de Socket con UDP

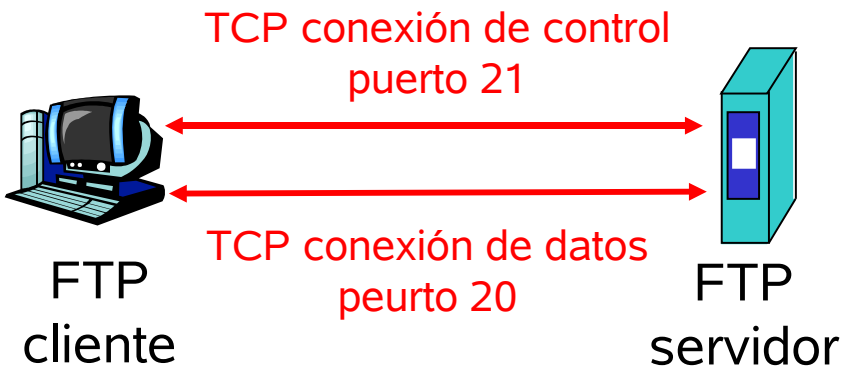
FTP: File Transfer Protocol



- Se transfiere al y desde el equipo remoto
- Arquitectura cliente/servidor
  - *cliente:* inicia la conexión
  - *servidor:* remote host
- ftp: RFC 959
- ftp servidor: puerto 21

FTP: separación control, datos

- Cliente FTP conecta al servidor FTP en el puerto 21, utilizando TCP como protocolo de transporte
- El cliente es autorizado en la conexión de control
- El cliente neavega en el sistema de directorio enviando comandos en la conexión de control.
- Cuando el servidor recibe un comando de transferencia de archivo inicia una conexión TCP en el puerto 20
- Luego de transferir el archivo el servidor cierra la conexión



- El servidor abre otra conexión TCP para transferir otro archivo.
- La conexión de control se encuentra fuera de la transferencia de datos.
- Servidor FTP mantiene estado: directorio actual, autenticación

Comandos y respuestas FTP

Comandos:

- Enviados como texto ASCII
- **USER *username***
- **PASS *password***
- **LIST** devuelve la lista de archivos en el directorio actual
- **RETR filename** obtiene un archivo
- **STOR filename** guarda un archivo

Códigos de retorno

- Codigo de estado y descripción (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 de Aplicación

- 2.1 Principles of network applications
- 2.2 Web y HTTP
- 2.3 FTP
- 2.4 Correo Electronico
  - SMTP, POP3, IMAP
- 2.5 DNS
- 2.6 Aplicaciones P2P
- 2.7 Programación de Socket con TCP
- 2.8 Programación de Socket con UDP

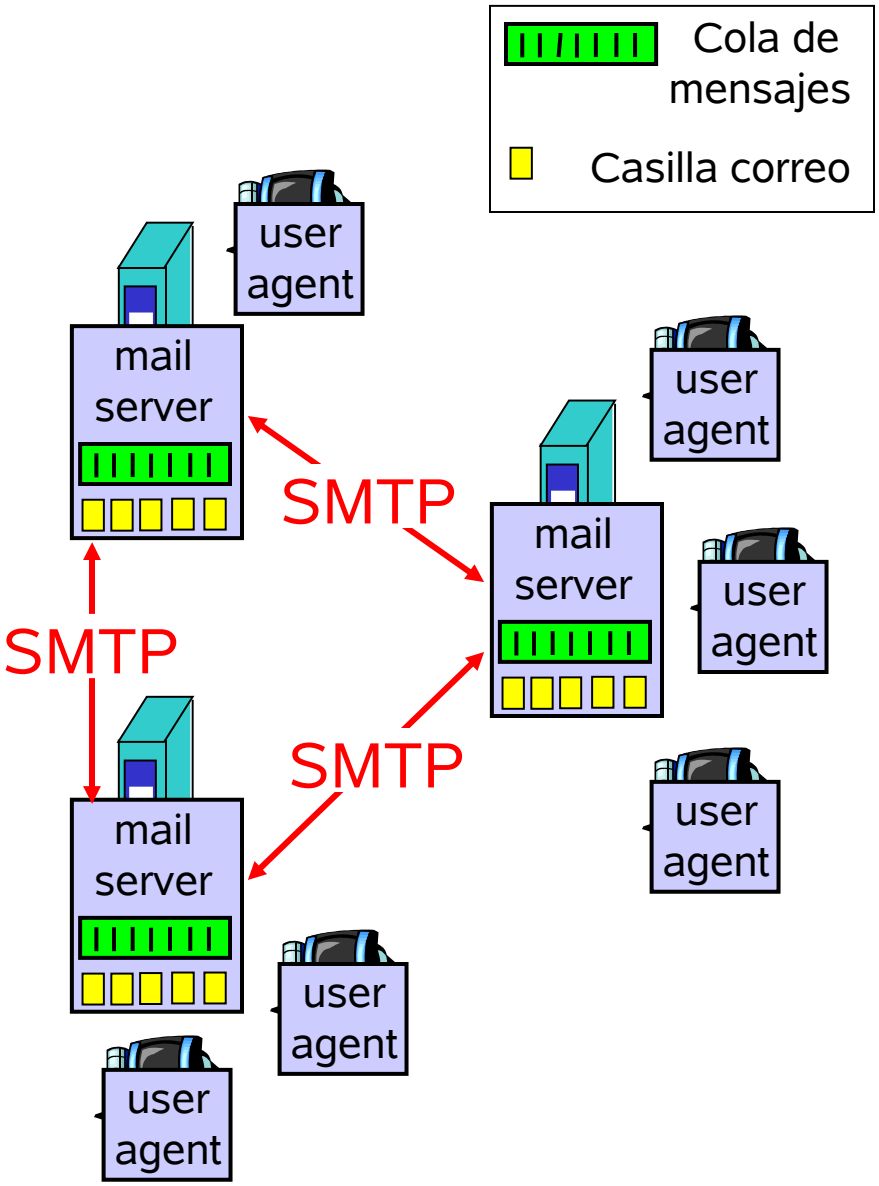
Correo Electrónico

Tres componentes:

- Usuarios
- Servidores
- SMTP: Simple Mail Transfer Protocol

Usuarios

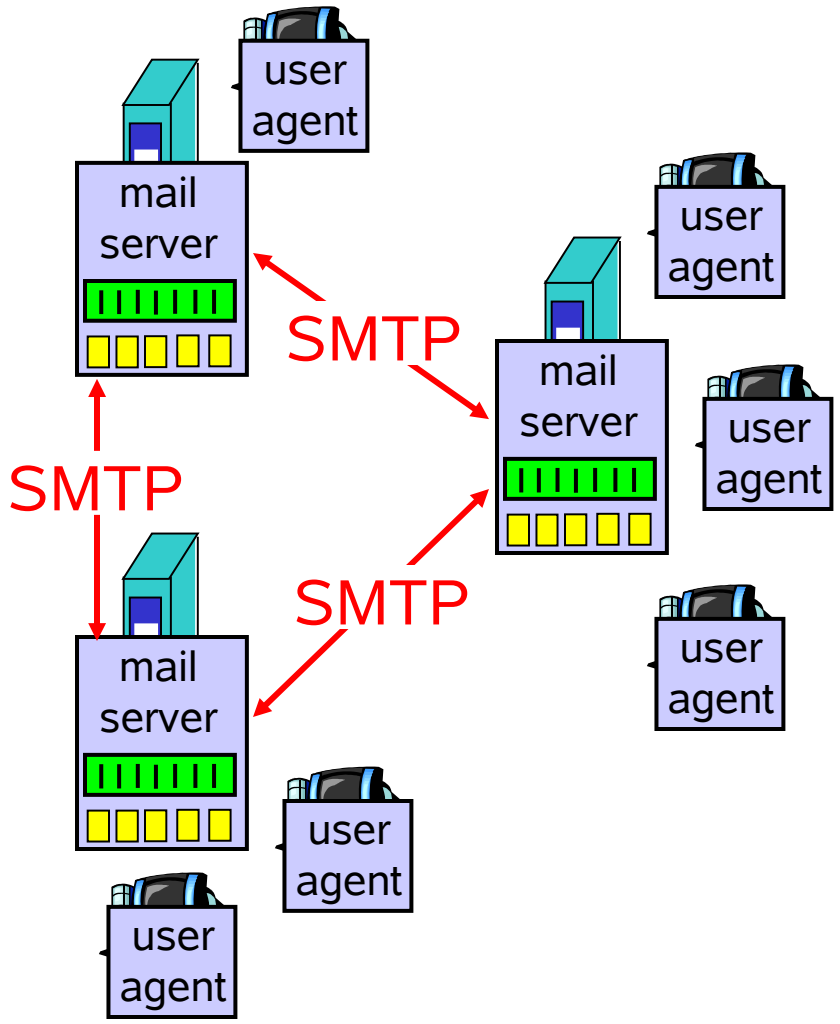
- Aplicación para leer correo
- Crear editar y leer mensajes
- Eudora, Outlook, elm, Mozilla Thunderbird
- Mensajes son guardados en servidor



Correo Electrónico: Servidores

Servidores

- Casilla de correo (mailbox) contiene los mensajes entrantes
- Cola de mensajes (message queue) mensajes salientes
- Protocolo SMTP entre usuarios y servidores
  - cliente: envia mail
  - servidor: recibe mail

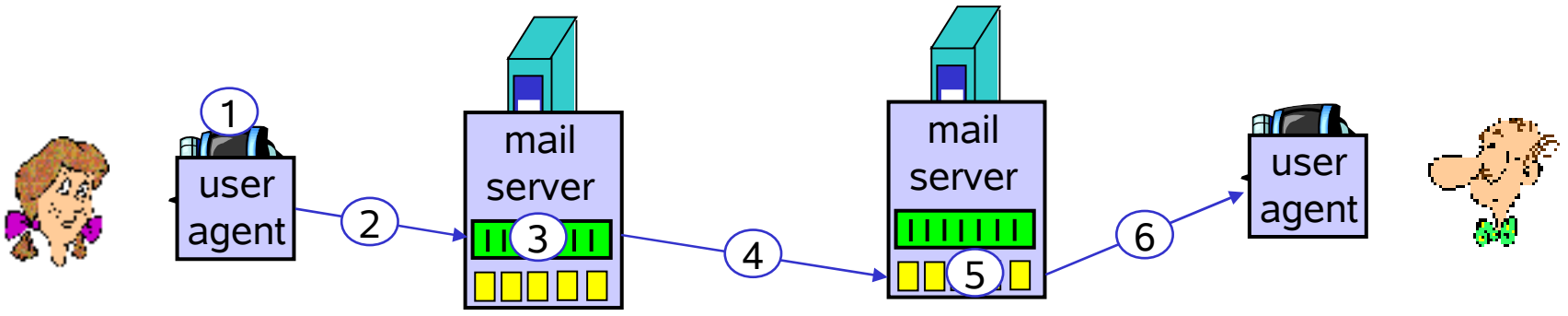


Correo Electrónico: SMTP [RFC 2821]

- Utiliza TCP para enviar mensajes en forma confiable del cliente al servidor en el puerto 25
- Tres fases de transferencia
  - Saludo (handshaking)
  - Transferencia de mensajes
  - Finalización
- Conexión directa servidor - servidor
- Interacción comandos y respuestas
  - comandos: texto ASCII
  - respuesta: codigo de estado y descripción
- Mensajes en ASCII 7-bit

Escenario: Alicia envía un mensaje a Roberto

1) Alicia usa aplicación para crear mensaje a `roberto@fing.edu.uy`
2) Alicia usa aplicación para enviar mensaje a su servidor de correo. El mensaje es puesto en una cola de mensajes
3) El servidor de Alicia abre una conexión TCP con el servidor de Roberto
4) El servidor SMTP de Alicia envia el mensaje por la conexión TCP
5) El servidor de Roberto coloca el mensaje en la casilla de Roberto
6) Roberto usa su aplicación para leer el mensaje



Ejemplo SMTP

```
220 smtp-s03.adinet.com.uy ESMTP Service ready
HELO notebook
250 smtp-s03.adinet.com.uy
MAIL FROM: <gabgg@adinet.com.uy>
250 MAIL FROM:<gabgg@adinet.com.uy> OK
RCPT TO: <gabgg@adinet.com.uy>
250 RCPT TO:<gabgg@adinet.com.uy> OK
DATA
354 Start mail input; end with <CRLF>.<CRLF>
Esto es una prueba linea 1
Esto es una prueba linea 2
.
250 <47D975C5006879DA> Mail accepted
QUIT
221 smtp-s03.adinet.com.uy QUIT
```

Try SMTP interaction for yourself:

- **`telnet adinet.com.uy 25`**
- Esperar respuesta 220
- Ingresar comandos HELO, MAIL FROM, RCPT TO, DATA, QUIT

SMTP

- SMTP usa conexiones persistentes
- SMTP requiere que el mensaje (cabezal y cuerpo) este en ASCII 7-bit
- SMTP usa `CRLF.CRLF` para determinar el fin de mensaje

Comparación con HTTP:

- HTTP: se extraen datos del servidor
- SMTP: se envían datos
- Ambos tienenn comandos ASCII con estados
- HTTP: cada objeto esta encapsulado en su propio mensaje
- SMTP: muchos objetos en un solo mensaje

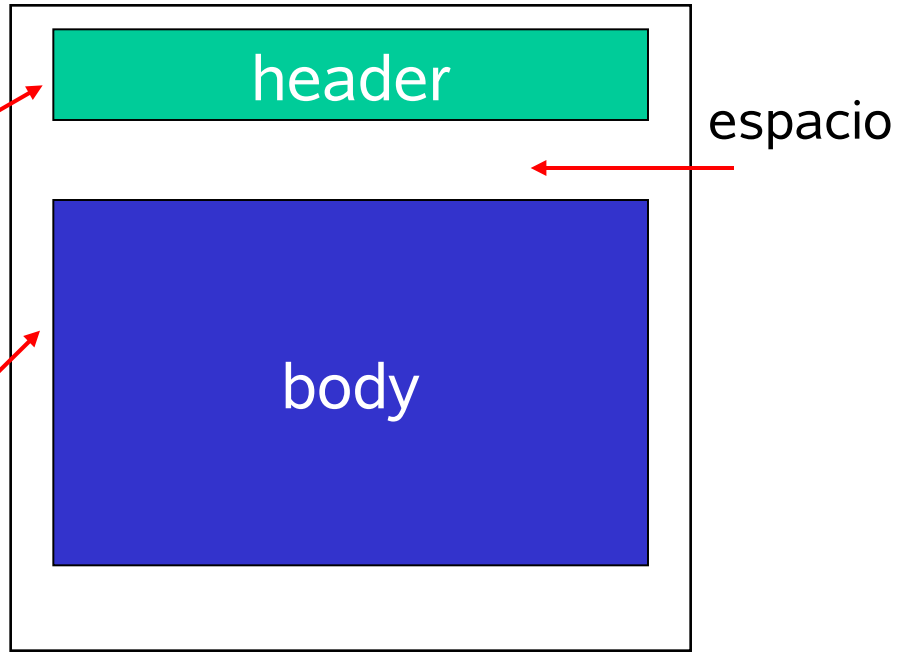
Formato del mensaje

SMTP: protocolo para intercambiar mensajes de correo

RFC 822: estándar para formato del mensaje

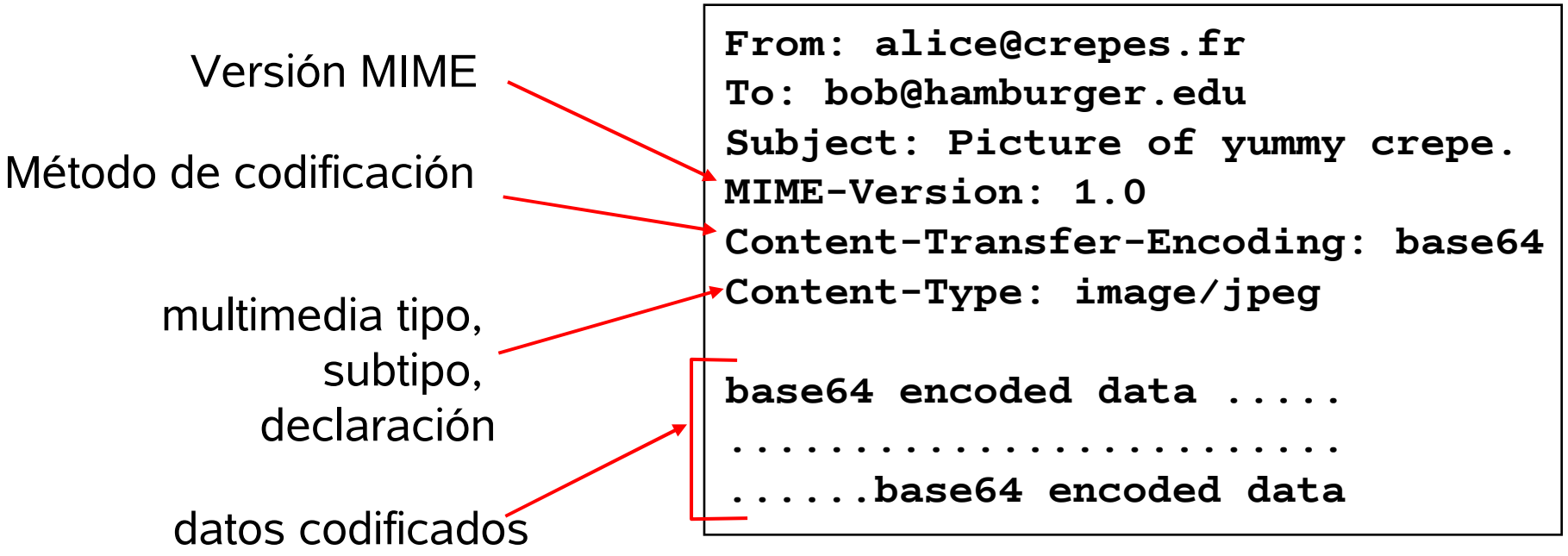
- Cabezal (header)
  - To:
  - From:
  - Subject:

  diferente de comandos SMTP
- Cuerpo (body)
  - el mensaje, soloASCII

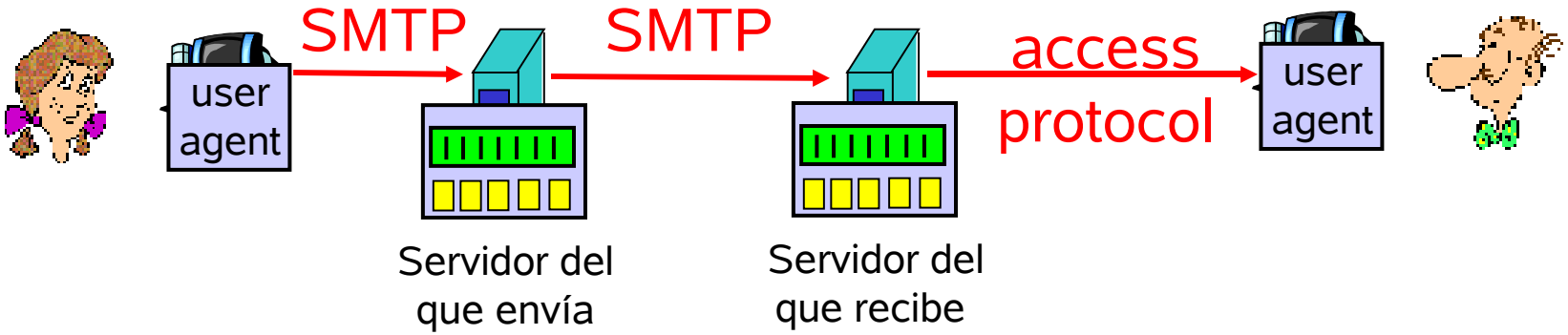


Formato del mensaje: extensiones multimedia

- MIME: multimedia mail extension, RFC 2045, 2056
- lineas adicionales declaraan el tipo de contenido MIME



Protocolo de acceso a correo



- SMTP: envio/almacenamiento
- Protocolo de acceso a correo: obtener del servidor
  - POP: Post Office Protocol [RFC 1939]
    - autorizzación (usuario <--> servidro) y bajada
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - Mas funcionalidad (mas complejo)
    - Manipulación de mensajes almacenados en el servidor
  - HTTP: gmail, Hotmail, Yahoo! Mail, etc.

POP3

Autorización

- Comandos del cliente:
  - **`user:`** declare username
  - **`pass:`** password
- Respuestas del servidor
  - **`+OK`**
  - **`-ERR`**

Interacción

- **`list:`** número de mensajes
- **`retr:`** obtiene mensaje por número
- **`dele:`** borra mensaje
- **`quit`**

```
S: +OK POP3 server ready
C: user roberto
S: +OK
C: pass roberto123
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
```

POP3 e IMAP

More about POP3

- En el ejemplo se obtiene y borra el mensaje
- Roberto no puede volver a leer el mensaje en otro cliente de correo
- Se puede obtener el mensaje sin borrar
- POP3 no tiene estado entre sesiones

IMAP

- Se guardan todos los mensajes en el servidor
- Se pueden organizar los mensajes en directorios
- IMAP mantiene estado entre sesiones:
  - Nombres de directorios, mensajes y directorios.