

Servicios de Internet

Area de Ingeniería Telemática
<http://www.tlm.unavarra.es>

Arquitectura de Redes, Sistemas y Servicios
3º Ingeniería de Telecomunicación

Servicio: e-mail

Contenido

- Arquitectura
- SMTP
- Formato de los mensajes
- Protocolos de acceso al mail

Contenido

- **Arquitectura**
- **SMTP**
- **Formato de los mensajes**
- **Protocolos de acceso al mail**

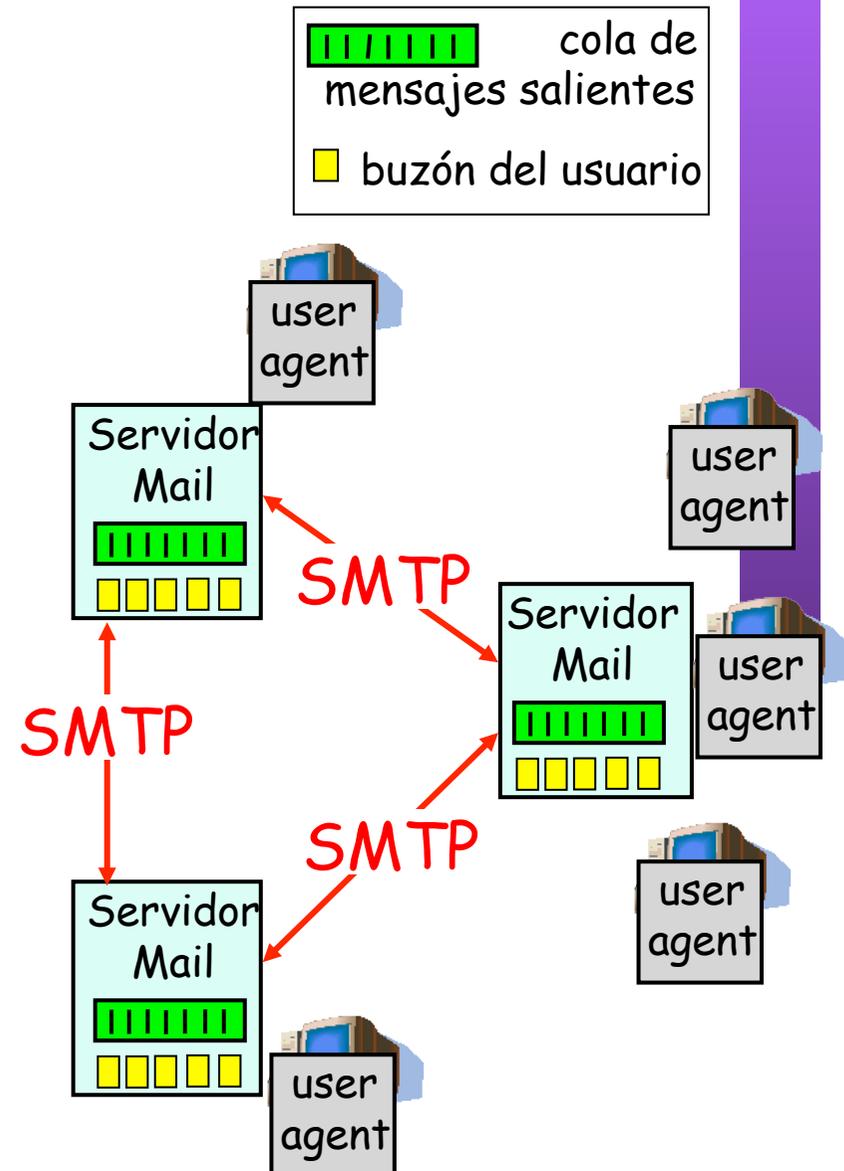
Electronic Mail

Tres elementos principales:

- Agentes de usuario (*user agents*)
- *Mail servers*
- Simple Mail Transfer Protocol:
SMTP

User Agent

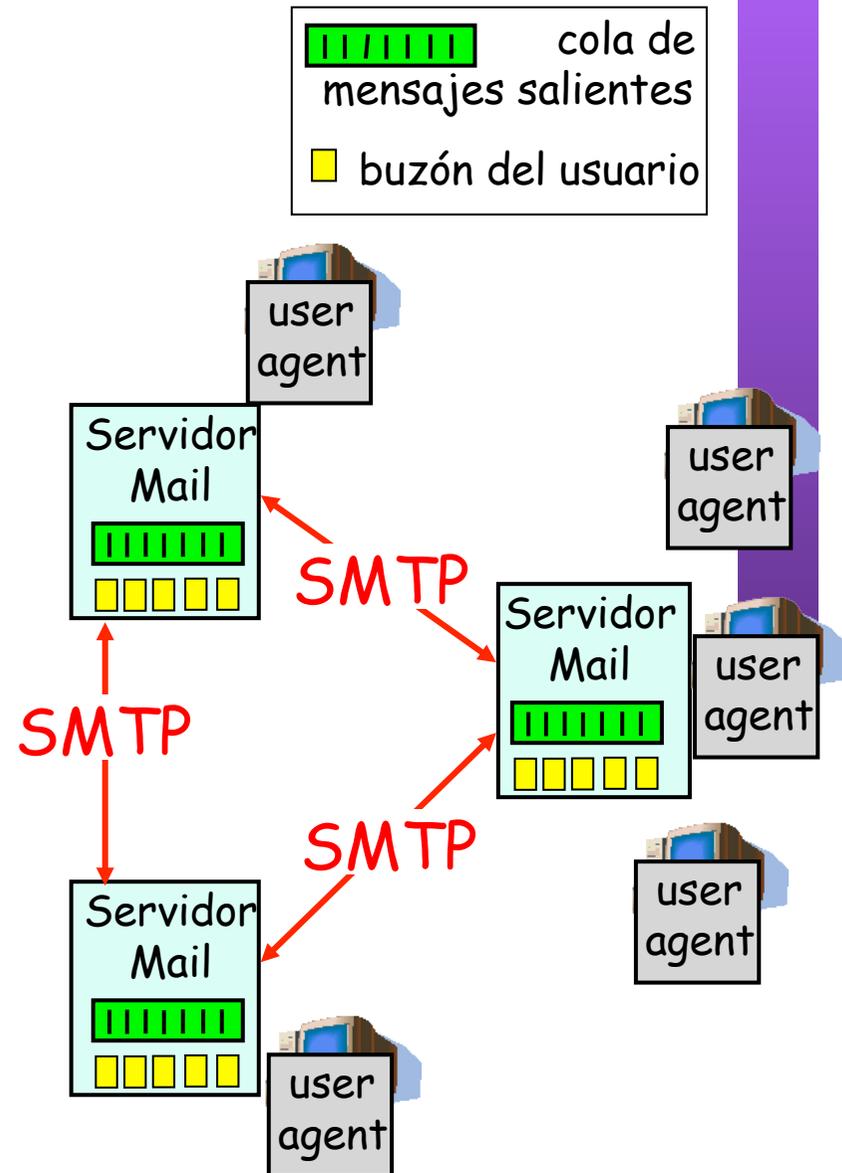
- alias “programa de correo”
- Componer, editar, leer mensajes de correo
- ej., Eudora, Outlook, elm, Netscape Messenger
- Mensajes salientes y entrantes en el servidor



E-Mail: Servidores

Servidores de Mail:

- **Mailbox** contiene los mensajes entrantes para el usuario
- **Cola de mensajes** salientes (a enviar)
- **Protocolo SMTP** entre servidores de correo para enviar mensajes
 - cliente: el servidor de correo que envía
 - “servidor”: el servidor de correo que recibe



Contenido

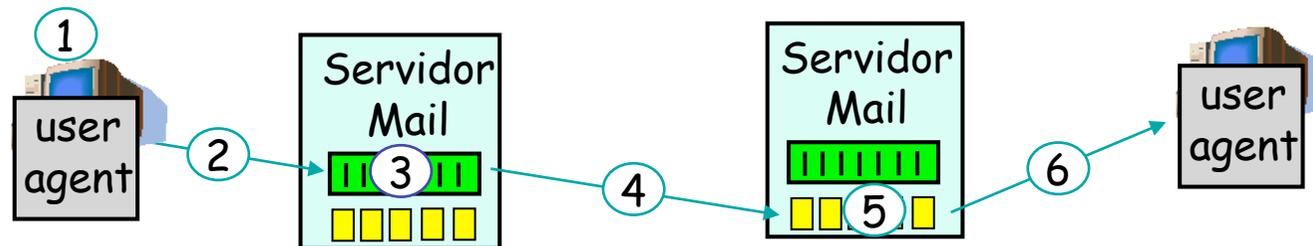
- Arquitectura
- **SMTP**
- Formato de los mensajes
- Protocolos de acceso al mail

E-Mail: SMTP [RFC 2821]

- Emplea **TCP** para entregar de forma fiable los mensajes entre el cliente y el servidor
- Puerto **25**
- Transferencia directa: del servidor del emisor al servidor del receptor
- **Tres fases** en la transferencia
 - handshaking (el saludo)
 - transferencia de mensajes
 - cierre
- Interacción mediante comandos y respuestas
 - comandos: texto ASCII
 - respuestas: código de estado y frase de estado (texto ASCII)
- Los mensajes deben estar en **ASCII** de 7 bits

Ejemplo: Usuario 1 envía mensaje a Usuario 2

- 1) Usuario 1 emplea un UA para crear el mensaje para usuario2@micasa.com
- 2) El programa envía el mensaje a su servidor de correo y lo coloca en una cola de mensajes
- 3) El Servidor de Mail, como cliente, abre una conexión TCP con el Servidor de Usuario 2
- 4) Envía el mensaje de Usuario 1 empleando SMTP sobre esa conexión TCP
- 5) El servidor de mail de Usuario 2 coloca el mensaje en su buzón
- 6) Usuario 2 lanza su UA para leer el mensaje (volveremos a esta parte)



Ejemplo de SMTP

[Conexión del cliente con el servidor (puerto 25)]

- 220 unavarra.es ESMTP Sendmail 8.9.3/8.9.1 (IRIS 3.0); Fri, 29 Apr 2005 14:00:19 +0200 (MET DST)
 - **HELO daniel.tlm.unavarra.es**
 - 250 unavarra.es Hello s169m159.unavarra.es [130.206.169.159], pleased to meet you
 - **MAIL FROM: <daniel.morato@unavarra.es>**
 - 250 <daniel.morato@unavarra.es>... Sender ok
 - **RCPT TO: danielmorato@yahoo.com**
 - 250 danielmorato@yahoo.com... Recipient ok
 - **DATA**
 - 354 Enter mail, end with "." on a line by itself
 - **Hola**
 - **Aqui, saludandome a mi mismo**
 - .
 - 250 OAA24057 Message accepted for delivery
 - **QUIT**
 - 221 unavarra.es closing connection
- [Cierre de la conexión TCP]*

Probando SMTP

- `nc servername 25`
- `ó`
- `telnet servername 25`
- Pruebe los comandos HELO, MAIL FROM, RCPT TO, DATA, QUIT
- Con esos comandos puede enviar un email sin emplear un programa de email
- Si el servidor de SMTP es el programa `sendmail` incluso ofrece ayuda con el comando HELP

Algo más sobre SMTP

- Conexiones persistentes
- Requiere que el mensaje (cabecera y contenido) esté en ASCII de 7 bits
- El servidor de SMTP emplea **CRLF** . **CRLF** para reconocer el final del mensaje

Comparación con HTTP:

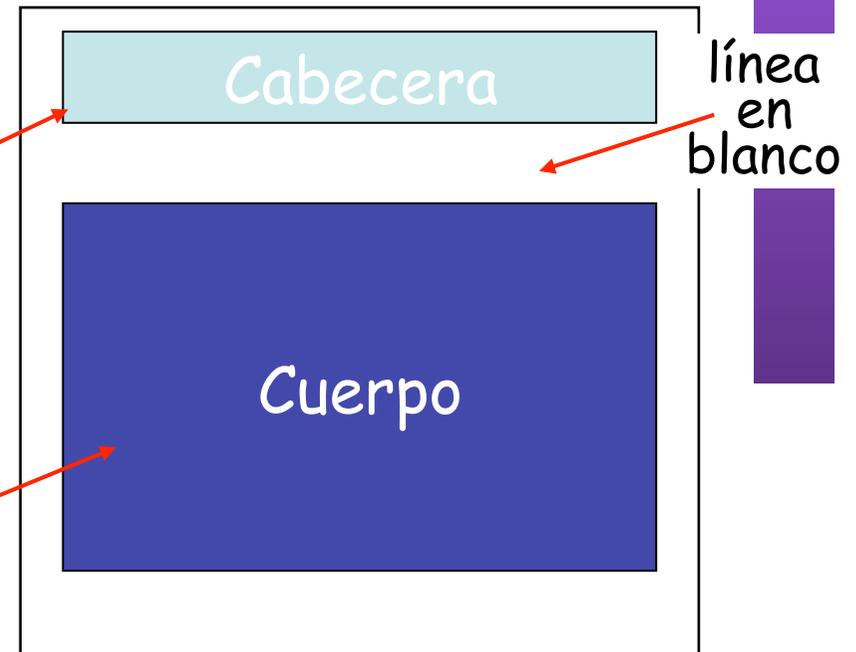
- HTTP: pull
- SMTP: push
- Ambos emplean comandos y respuestas en ASCII

Contenido

- Arquitectura
- SMTP
- **Formato de los mensajes**
- Protocolos de acceso al mail

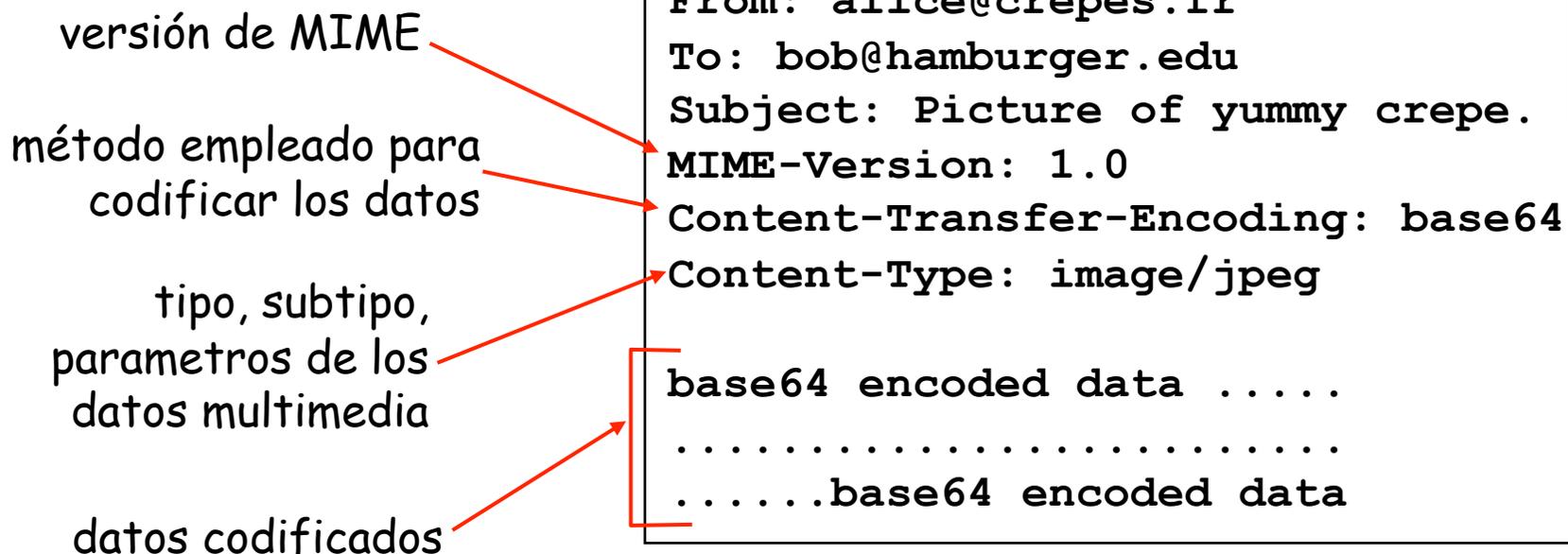
Formato del mensaje de email

- SMTP: protocolo para intercambiar mensajes de email (RFC 2821)
- RFC 822: estándar para el formato del mensaje:
- Líneas de cabecera, ej.,
 - **To:**
 - **From:**
 - **Subject:**Diferentes de los comandos de SMTP
- Cuerpo
 - el “mensaje”, solo caracteres ASCII



Formato del mensaje: Multimedia Extensions

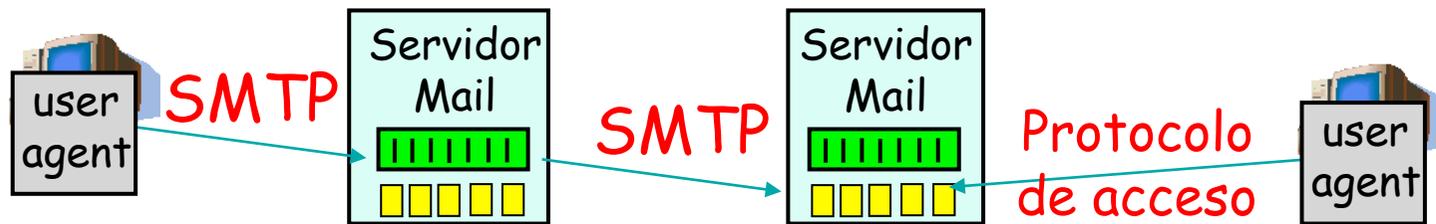
- MIME: Multimedia Mail Extension, RFC 2045, 2056
- Permite mandar contenido que no sea texto ASCII
- Líneas adicionales en la cabecera del mensaje para declarar el tipo del contenido



Contenido

- Arquitectura
- SMTP
- Formato de los mensajes
- **Protocolos de acceso al mail**

Protocolos de acceso al Mail



- SMTP: entrega/almacena en el servidor del receptor
- Protocolo de acceso al Mail: obtención de mensajes del servidor
 - POP: Post Office Protocol [RFC 2821]
 - Autorización (agente ↔ servidor) y descarga
 - IMAP: Internet Message Access Protocol [RFC 3501]
 - Más funcionalidades (más complejo)
 - Manipulación de mensajes almacenados en el servidor
 - HTTP: Hotmail , Yahoo! Mail, etc.

Protocolo POP3

Autorización

- Comandos del cliente:
 - **user** : declara el nombre de usuario
 - **pass** : clave
- Respuestas del servidor:
 - **+OK**
 - **-ERR**

Fase de transacción, cliente:

- **list** : lista números de mensajes
- **retr** : descarga mensaje por número
- **dele** : borrar
- **quit**

```
+OK Qpopper (version 4.0.5) at si starting.  
user daniel.morato  
+OK  
pass hungry  
+OK daniel.morato has 412 visible messages (0 hidden) in  
35020509 octets.  
list  
1 498  
2 912  
.  
retr 1  
<contenido mensaje 1>  
.  
dele 1  
retr 2  
<contenido mensaje 2>  
.  
dele 2  
quit  
+OK POP3 at si signing off
```

Más sobre POP3 e IMAP

Más sobre POP3

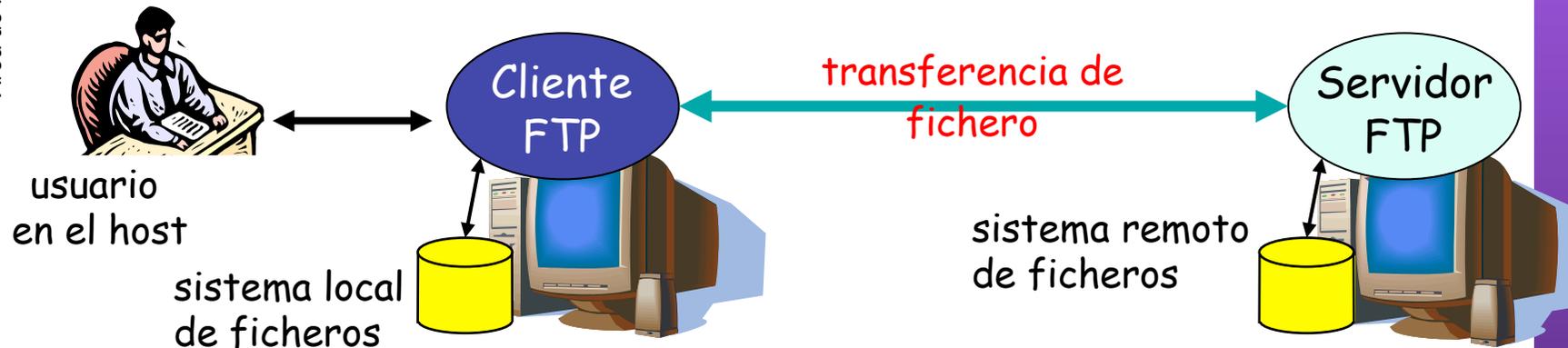
- El ejemplo anterior era “descargar y borrar”
- Bob no puede volver a leer los mensajes si cambia de cliente
- “Descargar y mantener”: copia el mensaje pero no lo borra. Permite descargarlos en otro cliente
- POP3 es sin estado entre sesiones
- Puerto 110

IMAP

- Mantiene todos los mensajes en un lugar: el servidor
- Permite al usuario organizar los mensajes en carpetas
- IMAP mantiene el estado entre sesiones:
 - Nombres de carpetas y relación entre ID de mensaje y carpeta en la que está

Servicios: FTP y Telnet

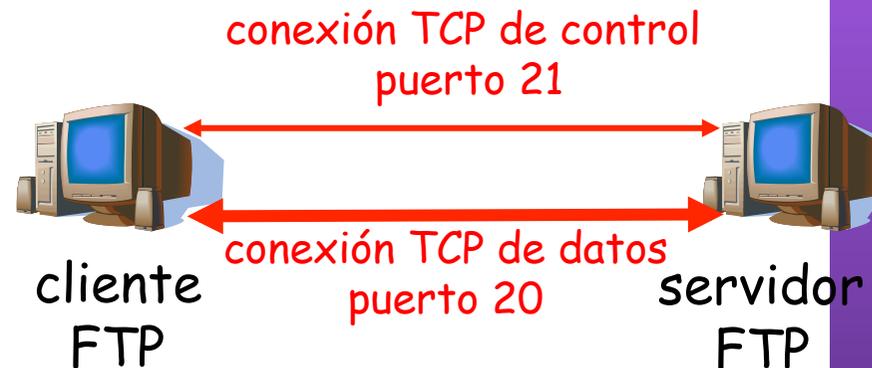
FTP: File Transfer Protocol



- Transferencia de fichero hacia/desde host remoto
- modelo cliente-servidor
 - **cliente**: extremo que inicia la transferencia (bien sea desde o hacia el extremo remoto)
 - **servidor**: host remoto
- FTP: RFC 959
- Servidor FTP: TCP puerto 21

FTP: conexiones de datos y control separadas

- El **cliente FTP** **contacta** con el servidor en el puerto 21
- **Se autentifica** a través de esta **conexión de control**
- Puede explorar los directorios remotos enviando comandos por la conexión de control
- Conexión de control “out of band”
- Cuando el **servidor** recibe un comando para una transferencia de fichero **abre una conexión TCP con el cliente**



- Servidor emplea el puerto 20 en esa conexión
- Tras transferir el fichero cierra esa conexión de datos
- El servidor FTP mantiene el “estado”: directorio actual, autenticación

Comandos y respuestas FTP

Comandos de ejemplo:

- Enviados como texto ASCII por el canal de control
- **USER username**
- **PASS password**
- **LIST** devuelve una lista de los ficheros en el directorio actual
- **RETR filename** Obtiene el fichero
- **STOR filename** Almacena el fichero en el host remoto

Códigos de respuesta:

- Código de estado y frase (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**

Login remoto (Telnet)

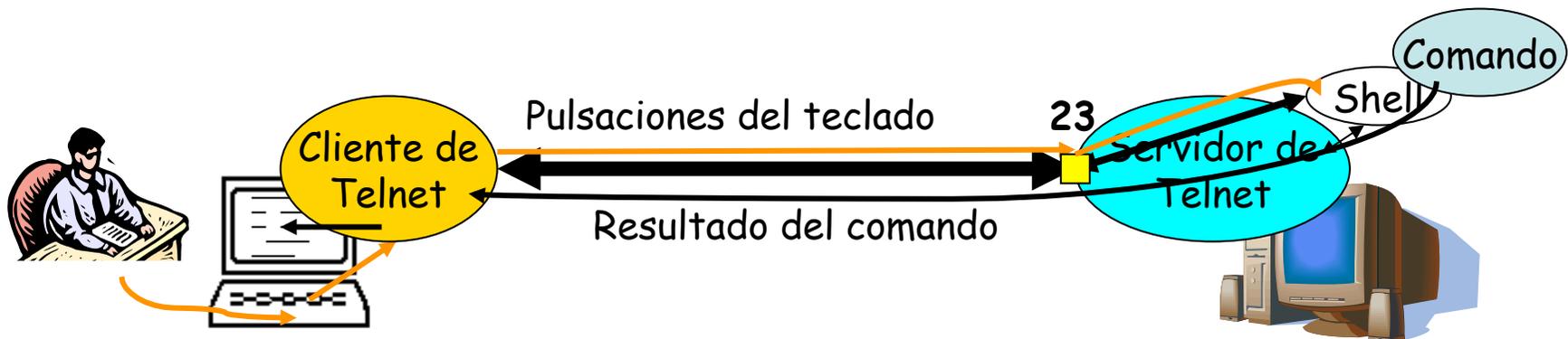
Uso interactivo remoto de una computadora

Funcionamiento:

- Usuario ejecuta un **cliente de Telnet** especificando la máquina servidor (...)
- Conexión TCP con el servidor (**puerto servidor 23**) (...)
- El servidor crea una **Shell** que queda conectada a la conexión TCP (...)

- Las pulsaciones del teclado del usuario se transmiten por la conexión a la Shell (...)
- La **shell ejecuta los comandos** que escribe el usuario (...)
- Resultado del comando a pantalla vuelve por la conexión TCP y sale en la pantalla del cliente (...)

Otros servicios similares:
rlogin, rsh, ssh



Ejemplo de Telnet

```
$ telnet 10.1.11.1
Trying 10.1.11.1...
Connected to 10.1.11.1.
Escape character is '^]'.

Red Hat Linux release 6.0 (Hedwig)
Kernel 2.2.5-15 on an i586
login: ro
Password:
Last login: Fri Nov  9 09:30:27 from lucas.net.tlm.unavarra.es
[ro@pclr11 ro]$ ls -al
total 3
drwxr-xr-x  2 ro      users      1024 Oct 31 20:10 .
drwxr-xr-x  5 root    root       1024 Sep 25 19:25 ..
-rw-----  1 ro      users      482  Nov  9 09:30 .bash_history
[ro@pclr11 ro]$ date
Fri Nov  9 09:50:57 CET 2001
[ro@pclr11 ro]$ ls
[ro@pclr11 ro]$ exit
logout
Connection closed by foreign host.
```



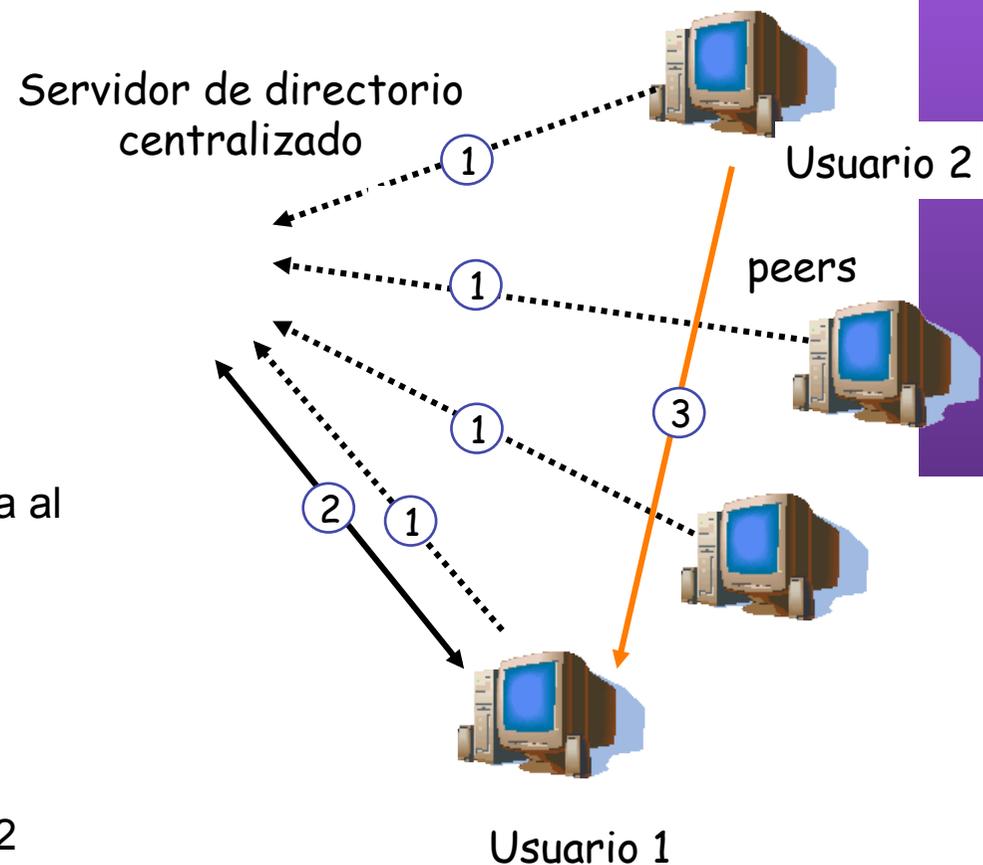
Servicios: intercambio P2P y mensajería



P2P: directorio centralizado

Diseño original de "Napster"

- 1) Cuando un peer se conecta, informa al servidor central:
 - Dirección IP
 - contenido
- 2) Usuario 1 hace una búsqueda de "Requiem"
- 3) Usuario 1 pide el fichero a Usuario 2



Ventajas e inconvenientes

Ventajas

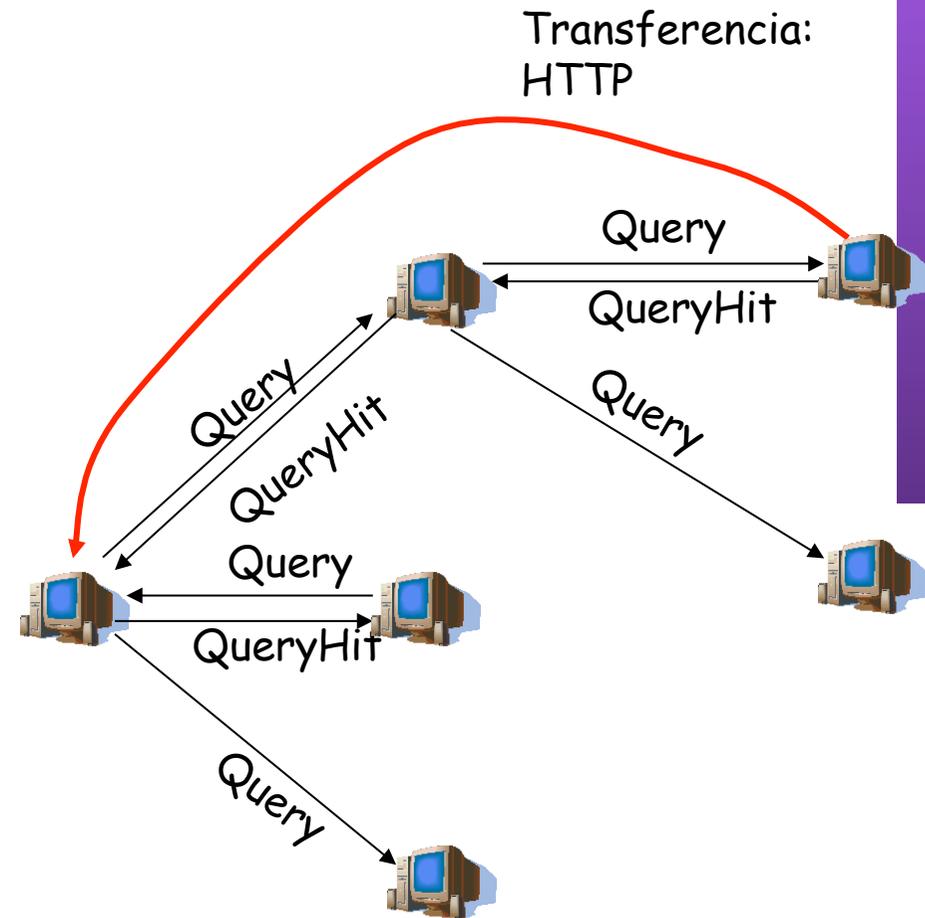
- Todos los peers son servidores
- **Altamente escalable**

Inconvenientes

- **Un punto de fallo central**
- Impone un límite de prestaciones
- Infracción de copyrights!

Gnutella

- Completamente distribuido
- Dominio público
- Overlay network
 - Grafo
 - Cada conexión un enlace
- Petición de búsqueda enviada sobre las conexiones TCP
- peers reenvían la petición
- Respuesta enviada por el camino inverso
- Escalabilidad: limitar el alcance de la inundación



Contenido

- DNS
- P2P
- **Mensajería**

Servicios de conversación

- Conversación a líneas en máquinas UNIX: **write**

```
[daniel@t1m13 daniel]$ write lir
Hola tu
[daniel@t1m13 daniel]$
```

```
[lir@t1m13 lir]$
Message from daniel@t1m13.net.t1m.unavarra.es on
pts/0 at 18:39 ...
Hola tu
EOF
```

- Conversación en terminal UNIX completo: **talk**

```
[daniel@t1m13 daniel]$ talk lir@t1m21
```

```
[Connection established]
Hola
```

```
[-----]
Pues hola
```

```
Message from Talk_Daemon@t1m13.net.t1m.unavarra.es
at 18:30 ...
talk: connection requested by daniel@t1m13.
talk: respond with: talk daniel@t1m13.
[lir@t1m13 lir]$ talk daniel@t1m13
```

```
[Connection established]
Pues hola
```

```
[-----]
Hola
```

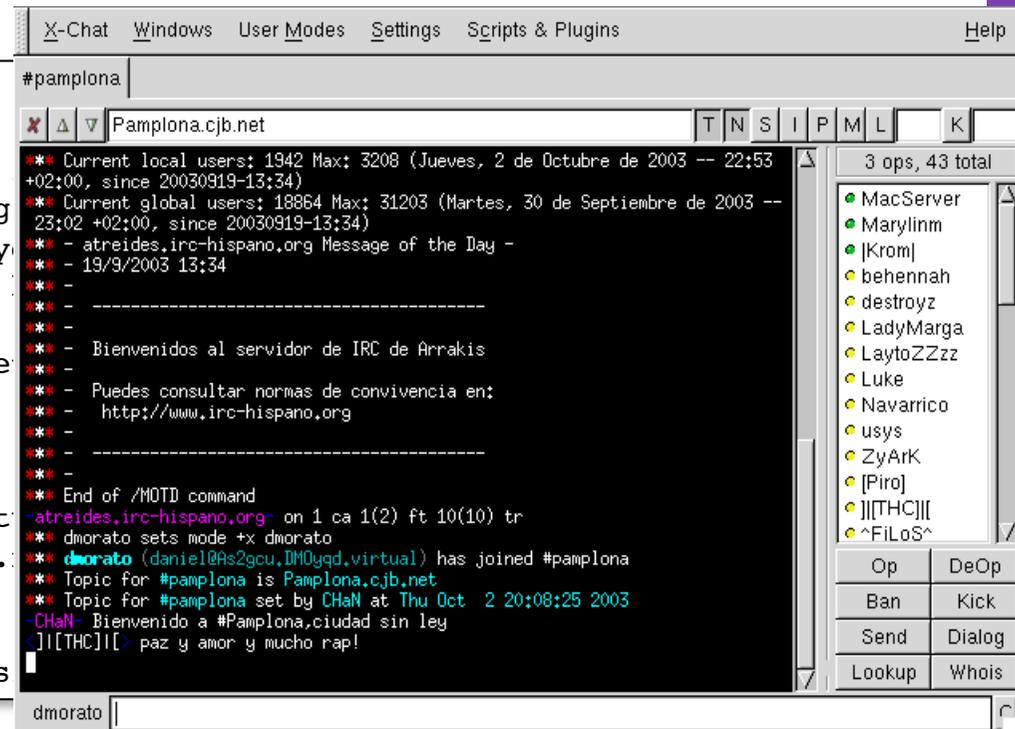
Evolución de los servicios clásicos de conversación

- Internet Relay Chat (IRC):
 - Los clientes se conectan a un servidor central
 - Existen “habitaciones”. Todos los usuarios que ejecuten el comando para “unirse” a una habitación podrán leer lo que cualquier otro en esa habitación escriba (... ..)

```
otilio[9]% irc dmorato irc.arrakis.es

*** Connecting to port 6667 of server
-atreides.irc-hispano.org- *** Looking
-atreides.irc-hispano.org- *** Found y
*** As2gcu.DMOyqd.virtual is now your
+atreides.irc-hispano.org)
*** Welcome to the IRC-Hispano Interne
+atreides.irc-hispano.org)

% /join #pamplona
*** dmorato (daniel@As2gcu.DMOyqd.virt
*** Topic for #pamplona: Pamplona.cjb.
*** #pamplona CHaN 1065118105
*** #pamplona 1006939646
-CHaN- Bienvenido a #Pamplona,ciudad s
```



Evolución de los servicios clásicos de conversación

Messengers (...)

