

Módulo 02
La Capa de Aplicaciones
(Pt. 4)

Redes de Computadoras
Depto. de Cs. e Ing. de la Comp.
Universidad Nacional del Sur



Copyright

- Copyright © 2010-2024 A. G. Stankevicius
- Se asegura la libertad para copiar, distribuir y modificar este documento de acuerdo a los términos de la **GNU Free Documentation License**, versión 1.2 o cualquiera posterior publicada por la Free Software Foundation, sin secciones invariantes ni textos de cubierta delantera o trasera
- Una copia de esta licencia está siempre disponible en la página <http://www.gnu.org/copyleft/fdl.html>
- La versión transparente de este documento puede ser obtenida de la siguiente dirección:
<http://cs.uns.edu.ar/~ags/teaching>

Redes de Computadoras - Mg. A. G. Stankevicius 2

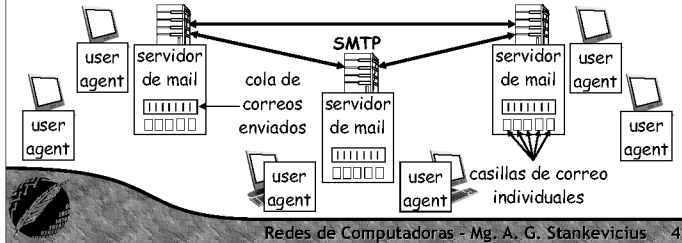
Contenidos

- Servicios que requiere la capa de aplicaciones
- Protocolos de la capa de aplicaciones
 - HTTP
 - DNS
 - SMTP, POP e IMAP
- Arquitectura de las aplicaciones P2P
- Programación basada en sockets

Redes de Computadoras - Mg. A. G. Stankevicius 3

Correo electrónico

- La infraestructura de correo electrónico convencional se compone de tres actores:
 - Los user-agents (esto es, los clientes)
 - Los servidores de correo electrónico
 - El protocolo **SMTP** (Simple Mail Transfer Protocol)



Correo electrónico

- Responsabilidad del user-agent:
 - Usualmente se lo denomina cliente de email
 - Compone, edita y visualiza correos
 - Los mensajes entrantes y salientes se almacenan en el servidor
- Responsabilidad del servidor:
 - Mantiene una casilla de correo (mailbox) independiente para cada usuario
 - Mantiene una cola de mensajes salientes en tránsito

El protocolo SMTP

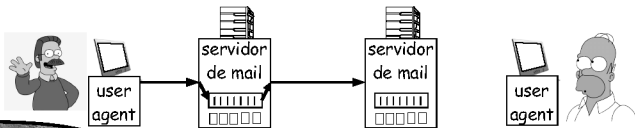
- Los servidores intercambian mensajes entre sí de forma directa, usando el protocolo **SMTP**
 - Se define formalmente en el **RFC 5321**
 - Usa **TCP** como protocolo de transporte
- Adoptan una arquitectura cliente-servidor
 - El servidor que envía un mensaje es el cliente
 - El servidor que recibe el mensaje es el servidor
 - El servidor está a la espera de nuevas conexiones en el puerto 25

El protocolo SMTP

- La transferencia de mensaje involucra tres etapas:
 - La inicialización (handshaking)
 - La transferencia del mensaje
 - La finalización
- Se basa en el intercambio de mensajes:
 - Comandos, codificados en **ASCII**
 - Respuestas, compuestas de un código y de una frase
- No usa **ASCII** extendido, usa **ASCII** de 7 bits

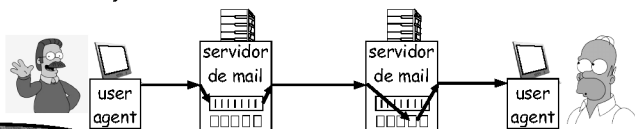
Esquema de interacción

- Flanders le quiere mandar un mail a Homero:
 - Flanders usa su user-agent para componer un nuevo mail destinado a Homero (**homero@springfield.com**)
 - El user-agent manda el nuevo correo al servidor de mail de Flanders, donde es encolado
 - El servidor de mail, actuando como cliente **SMTP**, se comunica con el servidor de mail de Homero



Esquema de interacción

- Continúa:
 - Se transfiere el nuevo correo electrónico entre los servidores de mail
 - El servidor de mail deposita en la casilla de correo de Homero el nuevo correo que acaba de recibir
 - El user-agent de Homero al verificar si llegaron nuevos mensajes finalmente recibe el correo de Flanders



Traza SMTP

```
S: 220 springfield.com
C: HELO hotmail.com
S: 250 Hello hotmail.com, pleased to meet you
C: MAIL FROM: <flanders@hotmail.com>
S: 250 flanders@hotmail.com ... Sender ok
C: RCPT TO: <homero@springfield.com>
S: 250 homero@springfield.com ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Homero, cuando me vas a devolver
C: la cortadora de cesped?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 springfield.com closing connection
```

Chateando con un servidor

- Una vez más, podemos intentar chatear con un servidor, en este caso **SMTP**:

```
$ telnet 127.0.0.1 25
```

...esperar a recibir la respuesta 220

...y probar qué pasa al ingresar los comandos

HELO, MAIL FROM, RCPT TO, DATA y **QUIT**,
en ese orden

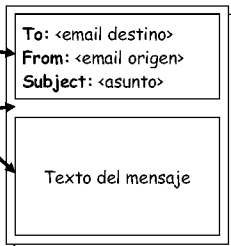
- Resulta cómodo usar herramientas como **FakeSMTP** para depurar el desarrollo de clientes (<http://nilhcem.com/FakeSMTP>)

Formato de los emails

- El formato del correo electrónico está especificado formalmente en el **RFC 822**

- Se compone de dos partes:

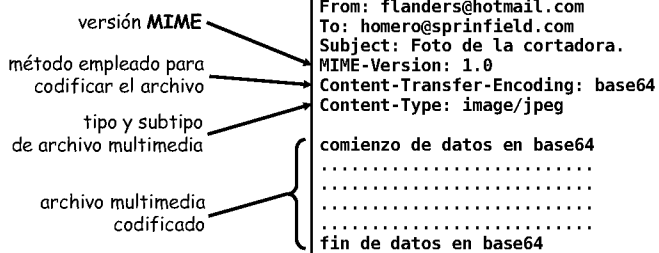
- Un encabezado
- El cuerpo del mensaje
- Se separan uno del otro por una línea en blanco
- No confundir los campos **To:** y **From:** con los del protocolo **SMTP**



Extensiones multimedia

- Recordemos que el cuerpo de un correo sólo puede contener caracteres **ASCII** de 7 bits
- Para poder enviar y recibir documentos de otros tipos (por caso, audio y video) se hace uso del formato **MIME** (Multipurpose Internet Mail Extension)
 - Definido formalmente en los **RFC 2045/6**
 - Líneas adicionales en el encabezamiento declaran qué tipo de archivo **MIME** aparece en el cuerpo del mensaje

Extensiones multimedia



Tipos y subtipos MIME

- El tipo y subtipo de archivo **MIME** se indica en la línea **Content-Type:** del encabezado
 - text para texto común (text/plain, text/html)
 - image para imágenes (image/jpeg, image/png)
 - audio para sonido (audio/basic, audio/mid)
 - video para video (video/mpeg)
 - application para formatos que requieran de un programa externo aparte del cliente de mail (application/msword)

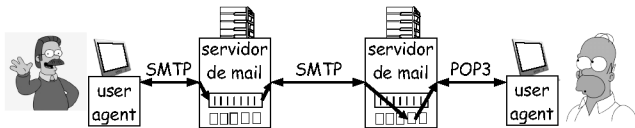
Mensajes multiparte

```
From: flanders@hotmail.com
To: homero@springfield.com
Subject: Yo de nuevo...
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=StartOfNextPart
```

```
--StartOfNextPart
Homero, no has visto mi cortadora de setos?
--StartOfNextPart
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 .....
..... base64
--StartOfNextPart
Te adjunto una foto de ella para refrescar tu memoria.
```

Protocolo de acceso al mail

- ¿Por qué hace falta un protocolo distinto para acceder a la casilla de correo personal?
 - El server **SMTP** tiene que estar siempre disponible, la computadora del usuario puede no estarlo
 - **SMTP** usualmente transmite un único mensaje, pero en la casilla de correo puede haber más de uno



Protocolo de acceso al mail

- Existen varias alternativas a la hora de acceder a los mails almacenados en la casilla de correo.
- **POP** (Post-Office Protocol):
 - Definido en el **RFC 1939** (versión 3)
 - Maneja las credenciales y la descarga de mails
- **IMAP** (Internet Mail Access Protocol):
 - Definido en el **RFC 9051** (versión 4, revisión 2)
 - Más complejo, pero con más variedad de funciones.
- **HTTP** (hotmail, gmail, etc.)

Traza POP3

● Fase de autorización:

- El cliente presenta sus credenciales
- El server contesta +OK o bien -ERR

● Fase de transacción:

- El cliente accede a los mensajes almacenados

```
S: +OK POP3 server ready
C: USER homero
S: +OK
C: PASS hamburguesa
S: +OK user successfully logged on
C: LIST
S: 1 498
S: 2 912
S: .
C: RETR 1
S: <contenido del mensaje 1>
S: .
C: DELE 1
C: RETR 2
S: <contenido del mensaje 2>
S: .
C: DELE 2
C: QUIT
S: +OK POP3 server signing off
```

POP3 vs. IMAP

- La traza anterior de **POP3** adopta una modalidad "descargo y borro"
 - Si cambio de user-agent, en el nuevo pierdo acceso a los mensajes descargados con el user-agent anterior
- También se puede hacer uso de **POP3** en una modalidad "descargo y guardo"
 - De esta forma, múltiples user-agents pueden tener acceso a la totalidad de los correos
- **POP3** no preserva el estado entre sesiones

POP3 vs. IMAP

- **IMAP**, en contraste, mantiene todos los correos en un único lugar: el servidor
- Permite que los usuarios organicen sus mensajes en distintas carpetas
- **IMAP** preserva el estado entre sesiones
 - Los nombres de las carpetas así como la distribución de mensajes en carpetas se conserva entre las distintas sesiones del usuario

¿Preguntas?
