

La capa de aplicación

Protocolos FTP, SMTP, POP3 e IMAP

Programación y administración de redes - Semana 3

Grado en *Ingeniería Informática*

Departamento de Informática. Universidad de Jaén

Objetivos

General

Situar la capa de aplicación en la arquitectura de redes de Internet e identificar sus funciones, entre las que se encuentran protocolos como FTP, SMTP, POP3 e IMAP

Específicos

- Determinar cómo se lleva a cabo la **comunicación al nivel de capa** de aplicación
- Analizar la necesidad de contar con mecanismos de **multiplexación**
- Conocer el funcionamiento del **protocolo FTP**
- Conocer el funcionamiento de los protocolos relacionados con el envío y recepción de **correo electrónico**: SMTP, POP3 e IMAP

Ubicación y finalidad de la capa de aplicación



Antes de empezar ...

Visión general

Visualiza este corto vídeo (2 min.) para tener una visión general sobre cómo funciona la transferencia de datos en Internet.

<https://www.youtube.com/watch?app=desktop&v=9ndoxFjYqPs>



Conceptos

- Las aplicaciones, como son un navegador web y el servidor con el que se comunica, operan en la **capa de aplicación**
- Para realizar su trabajo, los programas que funcionan en la capa de aplicación necesitan **servicios de transporte** de datos
- Dependiendo del tipo de aplicación se recurrirá a unos servicios u otros de los ofrecidos por la **capa de transporte** (a estudiar en la quinta semana)

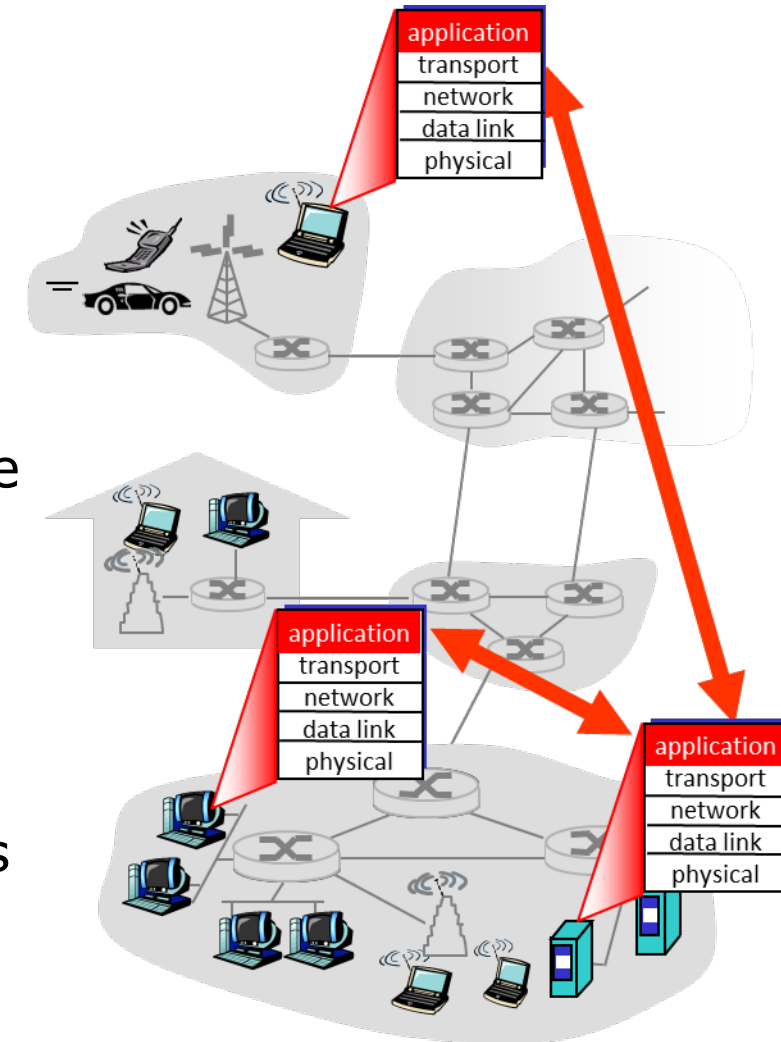
Introducción

Finalidad de la capa de aplicación

En esta capa se programan las aplicaciones y servicios que utiliza el usuario final y que involucran transmisión de información entre distintos equipos

Cómo funciona

- La capa de aplicación es la de **más alto nivel**, por lo que se sitúa en la parte superior de la pila TCP/IP
- Las aplicaciones ponen en marcha uno o varios procesos que se comunican **intercambiando mensajes** a través de una red de ordenadores
- Los mensajes que pueden utilizarse vienen determinados por **los protocolos** de la capa de aplicación
- Estos procesos usarán, a su vez, los **servicios** ofrecidos por la capa de **transporte**



Protocolos de la capa de aplicación

Finalidad

Establecer los tipos de mensajes, su formato, semántica y las reglas de sincronización entre los host que los usan para comunicarse

Detalles

- Tipos de mensajes
 - Mensajes de petición y de respuesta intercambiados por los *host*
- Sintaxis de cada tipo de mensaje
 - Formatos de los campos existentes en la cabecera
- Semántica de los campos
 - Significado de cada uno de los campos que conforman el mensaje
- Sincronía
 - Reglas sobre cuándo y cómo cada proceso envía y responde a los mensajes

DNS
Domain Name System

HTTP
Hypertext Transfer Protocol

FTP
File Transfer Protocol

DHCP
Dynamic Host Configuration Protocol

SMTP
Simple Mail Transfer Protocol

NNTP
Network News Transfer Protocol

Servicios de transporte demandados por las aplicaciones



Servicios ofrecidos a la capa de aplicación

Servicios de transporte en TCP/IP

Al crear una aplicación se deben tener en cuenta las características de los servicios de transporte disponibles que, en Internet, son esencialmente dos: TCP (orientado a conexión y fiable) y UDP (no orientado a conexión y no fiable)

Características

Las fundamentales a considerar son las siguientes:

– Pérdida de datos

- Ciertas aplicaciones (p.e., audio) pueden tolerar algunas pérdidas
- Otras aplicaciones (p.e., transferencia de archivos, SSH) requieren el **100% de transferencia fiable** de datos

– Temporización

- Algunas aplicaciones (pe., telefonía de Internet, juegos interactivos) requieren un **retardo bajo** para ser “efectivas”

– Ancho de banda

- Algunas aplicaciones (p.e., multimedia) requieren un **mínimo de ancho** de banda constante para ser “efectivas”
- Otras aplicaciones (“aplicaciones flexibles”) hacen uso de cualquier ancho de banda que tengan a su disposición

Características de aplicaciones en Internet

Aplicación	Pérdida de datos	Ancho de banda	Sensible al retardo	Protocolo de aplicación	Servicio de transporte
Transferencia de archivos	No pérdida	Flexible	No	FTP [RFC 959]	TCP
Correo electrónico	No pérdida	Flexible	No	SMTP [RFC 2821]	TCP
Documentos web	No pérdida	Flexible	No	HTTP [RFC 2616]	TCP
Audio/Vídeo en tiempo real	Tolerante	Audio: 5Kbps-1Mbps Vídeo: 10Kbps-5Mbps	Sí, 100ms	Propietario	TCP/UDP
Audio/Vídeo almacenado	Tolerante	Igual al anterior	Sí, pocos segundos	Propietario	TCP/UDP
Juegos interactivos	Tolerante	Decenas de Kbps	Sí, 100ms	Propietario	UDP (típicamente)
Mensajería instantánea	No pérdida	Flexible	Sí/No	Propietario	UDP (típicamente)
Acceso a terminales remotos	No pérdida	Flexible	Sí, pocos segundos	Telnet [RFC 854]	TCP

Actividad

Cómo funcionan tus aplicaciones

A diario usas diferentes aplicaciones para comunicarte con tu entorno, ¿sabes cómo funcionan?

Determina para las aplicaciones que más usas

- **Qué tipo de protocolo** de la capa de aplicación usa: HTTP, FTP, SMTP, ..., otro de tipo propietario
- Si es tolerante a la **pérdida de datos** o no
- Si precisa o no un **ancho de banda** y si es sensible o no al **retardo**
- Qué **servicio de la capa de transporte** emplea para enviar/recibir los mensajes que se comunican con otros *host*

Qué es y por qué es necesaria la multiplexación



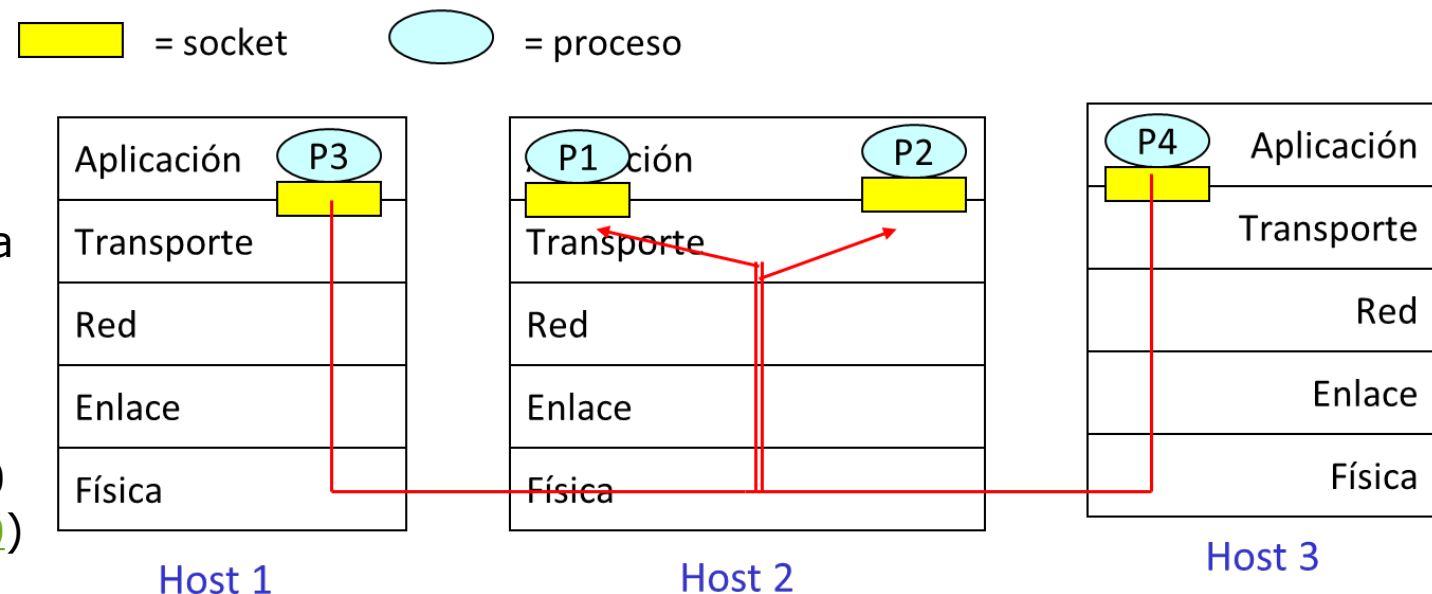
Mecanismo de multiplexación

Problemática

En un host múltiples procesos de la capa de aplicación demandan servicios de comunicación, pero existe una única conexión de red física

Solución:

- Mecanismo de **multiplexación/demultiplexación** basado en direcciones de la capa de transporte (**puertos**) que permita la entrega correcta de datos al proceso de la capa de aplicación que corresponda
- Un **socket** es el punto de acceso a los servicios de la capa de transporte, usados por los procesos de la capa de aplicación
- Al estudiar la capa de transporte, y en la práctica de programación con *sockets*, profundizaremos en este mecanismo de multiplexación
- En los apartados 2.3 y 2.4 del RFC 1180 (<https://datatracker.ietf.org/doc/html/rfc1180>) se describe la multiplexación en detalle



El protocolo FTP



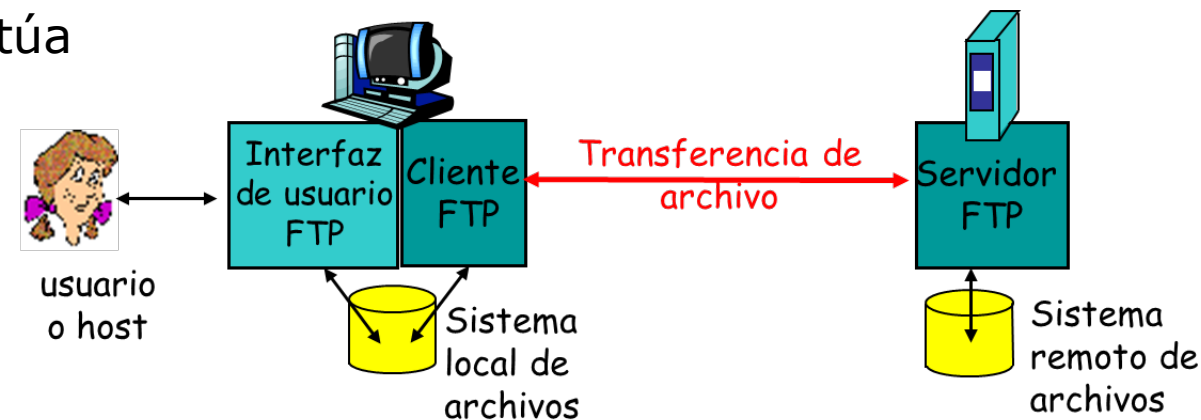
Protocolos de la capa de aplicación - FTP

Definición

FTP (File Transfer Protocol, RFC 959) es el protocolo que permite la transferencia de archivos entre dos host

Detalles esenciales

- Como la mayoría de protocolos, FTP está implementado según la arquitectura **cliente/servidor**:
 - **Cliente**: proceso que inicia la transferencia p.e. FileZilla, Navegador (ftp://)
 - **Servidor**: proceso en *host* remoto que actúa como servidor FTP
- FTP usa el protocolo de transporte TCP
- Servidor FTP: puertos TCP 20 y 21



Funcionamiento del protocolo FTP

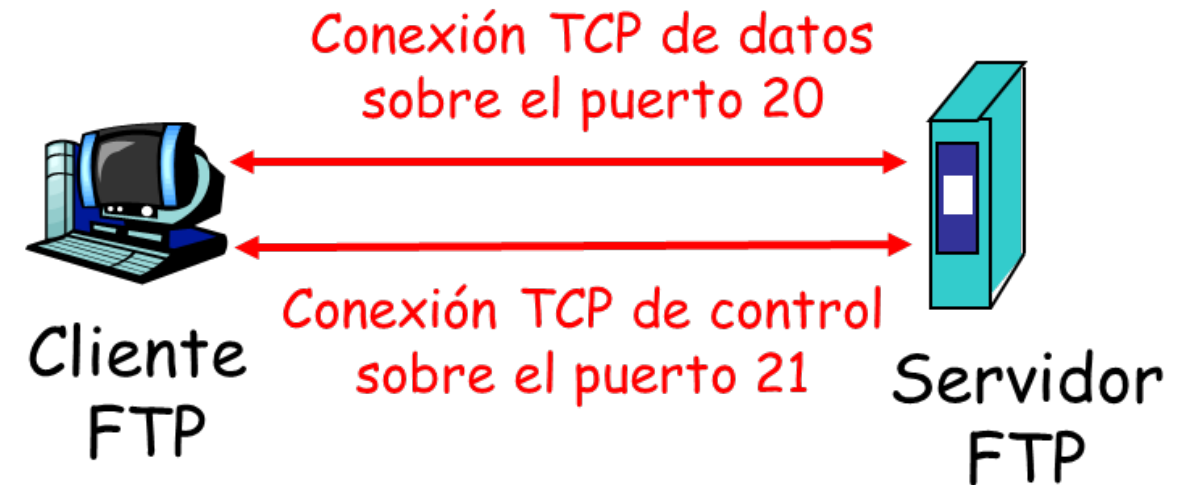
Se utilizan dos conexiones

– Conexión de **control**

- Puerto 21 → **Envío de órdenes**
- Se abre al establecer conexión entre cliente y servidor
- Se mantiene abierta durante **toda la sesión**

– Conexión de **datos**

- Puerto 20 → Transferencia de datos
- Se **establece y libera** por cada archivo que se envía



Funcionamiento del protocolo FTP

Pasos del protocolo

1. El cliente FTP **solicita establecer una conexión** TCP con el servidor FTP en el puerto 21
2. El cliente consigue la **autorización** sobre la conexión de control
3. El cliente navega por el directorio remoto enviando comandos sobre la **conexión de control**
4. Cuando el servidor recibe un comando para la transferencia de archivos, el servidor abre la **conexión TCP de datos** con el cliente
5. Después de transferir un archivo, el servidor **cierra** la conexión
6. Si se envía otro archivo se establecerá la conexión correspondiente, liberándola cuando concluya el envío
7. Cuando se decida terminar la sesión se liberará la conexión de control

Conceptos

- Al envío de información de control por otra conexión se denomina **envío de información fuera de banda**
- El servidor FTP mantiene **información de estado** que almacena las características de la conexión y su situación en cada momento

Mensajes de solicitud y respuesta de FTP

Mensajes de petición

Enviados como texto ASCII por el canal de control

- **USER usuario**
- **PASS contraseña**
- **LIST** devuelve lista de archivos del directorio en curso
- **RETR nombre de archivo** recupera el archivo
- **STOR nombre de archivo** almacena el archivo en el host remoto

Mensajes de respuesta

Códigos de estado y frase de respuesta

- **331 Username OK, password required**
- **125 data connection already open; transfer starting**
- **425 Can't open data connection**
- **452 Error writing file**

Actividad - Usar un cliente FTP

Acceso desde el terminal

- Usa el comando `ftp -p ftp.ujaen.es` para **establecer la conexión**
- Introduce `anonymous` como **nombre de usuario** y tu dirección de correo como **contraseña**
- El servidor FTP te dará la **bienvenida**, como se aprecia en la imagen mostrada a la derecha
- Usa el comando `help` para obtener una **lista de comandos**
- Introduce la orden `cd pub/doc/Manuales` para **cambiar el directorio actual**
- Usa el comando `dir` para obtener una **lista de archivos**
- Con el comando `get index.txt` **descarga** el índice
- Mediante la orden `bye` **cierra la conexión** con el servidor
- En tu directorio local encontrarás el archivo descargado

```
[usuario@servidor ~]$ ftp ftp.ujaen.es
Connected to ftp.ujaen.es (150.214.170.29).
220 Servidor FTP anonimo de la Universidad de Jaen
Name (ftp.ujaen.es:usuario): anonymous
331 Anonymous login ok, send your complete email address as your password
Password:
230- ////////////////////////////////////////////////////////////////////
//
//      Bienvenido al servidor FTP anonimo de la
//
//              UNIVERSIDAD DE JAEN
//
//      ftp://ftp.ujaen.es
//
//      Eres el usuario 1 de un maximo de 20
//
//      Para cualquier problema o duda, por favor, contacte con:
//
//      FTP Manager
//      Universidad de Jaen.
//      Campus de las Lagunillas, s/n
//      Jaen, Espana
//      Tfno: 34 953 212000
//      E-mail: ftp@ujaen.es
//
//      Este es un servidor anonimo.
//      Puede conectarse como usuario 'anonymous'
//      Cualquier password es valida. Se aconseja introducir
//      la direccion de correo electronico como password.
//
//      //////////////////////////////////////////////////////////////////////
230 Acceso permitido para anonymous
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> _
```

Acceso desde el navegador

- Introduce <ftp://ftp.ujaen.es> en la barra de direcciones

Protocolos de transferencia de correo electrónico - SMTP



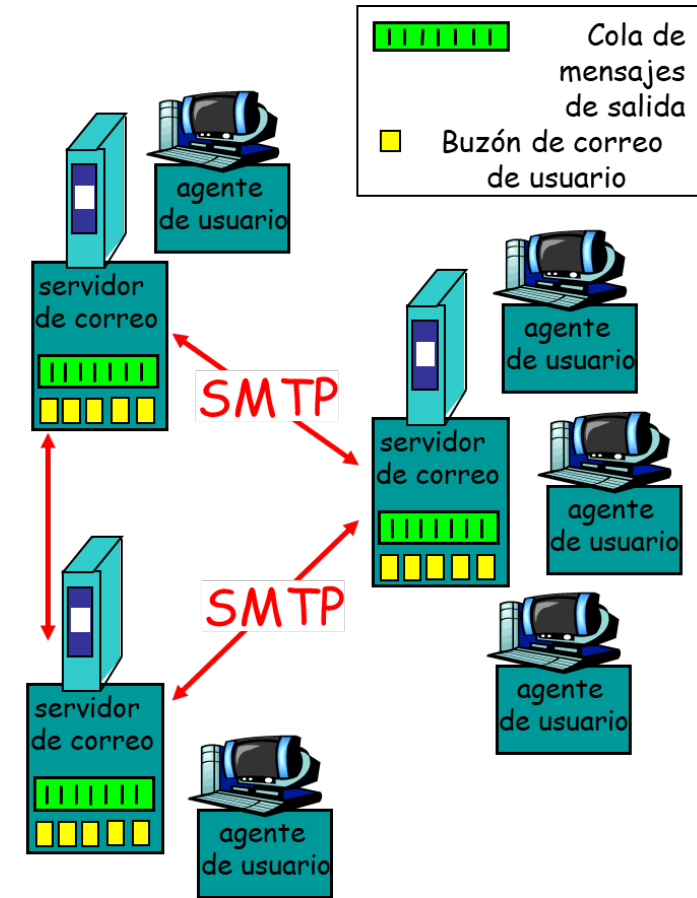
Correo electrónico - Introducción

Definición

Fue uno de los primeros servicios de Internet y continúa siendo uno de los más populares

Consta de tres componentes principales

- **Agente de usuario:**
 - Es el cliente de correo, facilita la **composición**, edición y **lectura** de los mensajes de correo, así como de la **salida y entrada**
 - Hay aplicaciones de gestión de correo, como Thunderbird y Outlook, así como acceso a través de navegador como GMail
- **Servidores de correo:**
 - Mantienen un **buzón de correo**, con los mensajes de entrada al usuario, y una **cola de mensajes** de salida (para ser enviados)
- **Protocolo SMTP:**
 - Usado tanto entre cliente y servidor, para realizar el envío de correo, como entre servidores, para transferir los mensajes de un o a otro



Correo electrónico - SMTP

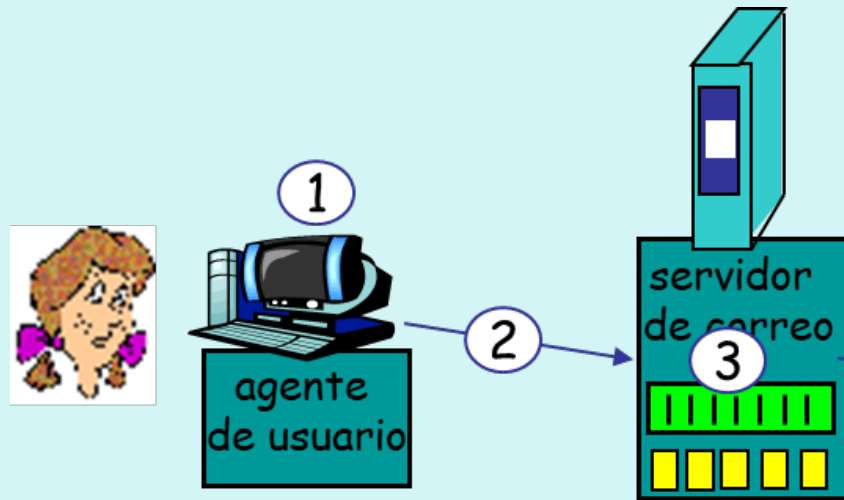
Definición

Protocolo simple para la transferencia de correo (Simple Mail Transfer Protocol, RFC 5321)

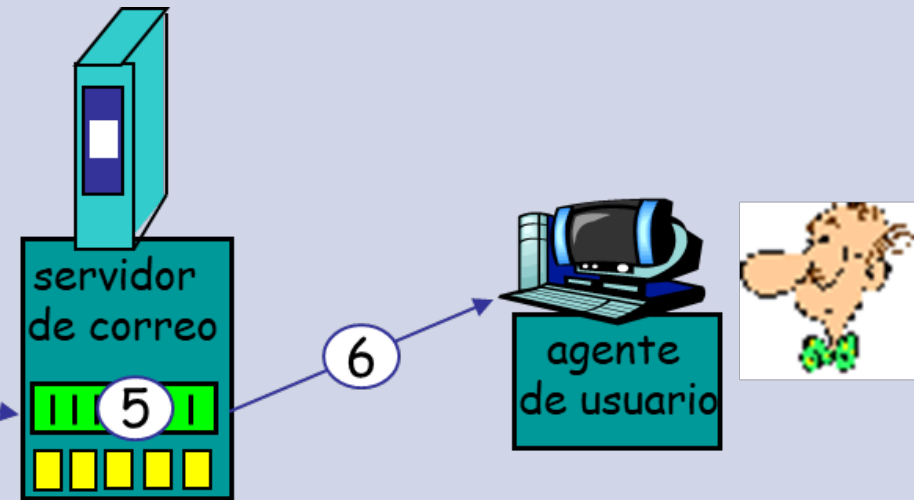
- Su funcionamiento se basa en establecer una **conexión TCP** por el puerto 25
- La transmisión entre servidores SMTP suele ser directa aunque se contempla la posibilidad de **pasarelas de correo**
- Utiliza **conexiones persistentes** (se mantiene la conexión mientras se envían los mensajes)
- Las **tres fases** de la transferencia son:
 - “Acuerdo” (saludo).
 - Transferencia de mensajes.
 - Cierre.
- Como en otros protocolos, la interacción entre cliente y servidor es del tipo `comando-respuesta`
 - El cliente envía comandos en codificación ASCII
 - El servidor responde con códigos **estado frase**
- Por su antigüedad, tanto los bytes de cabecera como los del cuerpo de un mensaje **deben ir codificados con el estándar ASCII** de 7 bits

Correo electrónico - SMTP - Ejemplo de envío

- 1) Alicia utiliza su **agente usuario** para componer el mensaje "a" Roberto (bob@escuela.edu)
- 2) El agente de usuario de Alicia envía un mensaje a su **servidor de correo (A)**, que ubica el mensaje en la cola de mensajes
- 3) El lado cliente de SMTP abre una conexión TCP con el **servidor de correo (B)** de Roberto



- 4) El **cliente SMTP** envía el mensaje de Alicia sobre la conexión TCP
- 5) El servidor de correo de Roberto deposita el mensaje en el **buzón de correo** de Roberto
- 6) Roberto recurre a su **agente de usuario** para leer el mensaje



Protocolo SMTP - Comandos más usuales

Comandos SMTP del cliente

El lado cliente SMTP usará habitualmente los siguientes comandos:

- **HELLO (HELO)**: identificación del host cliente
- **MAIL FROM**: especificación del remitente
- **RECIPIENT (RCPT)**: buzón destinatario de mensaje
- **DATA**: datos del mensaje (cuerpo y cabecera)
- **QUIT**: solicitud de cierre de la conexión
- **HELP**: comando de ayuda
- **EXPAND (EXPN)**: expande una lista de correo
- **VRFY**: permite verificar la existencia de un buzón
- **NOOP**: para solicitar eco
- **RESET (RSET)**: reinicio de transacción actual

Protocolo SMTP - Ejemplo de interacción

Conversación SMTP entre cliente y servidor

S: identifica al servidor (en verde), C: identifica al cliente (en gris)

S: 220 hamburger.edu

C: HELO crepes.fr

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

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

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

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

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

C: DATA

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

C: ¿Te gusta el ketchup?

C: ¿Y los encurtidos?

C: .

S: 250 Message accepted for delivery

C: QUIT

S: 221 hamburger.edu closing connection

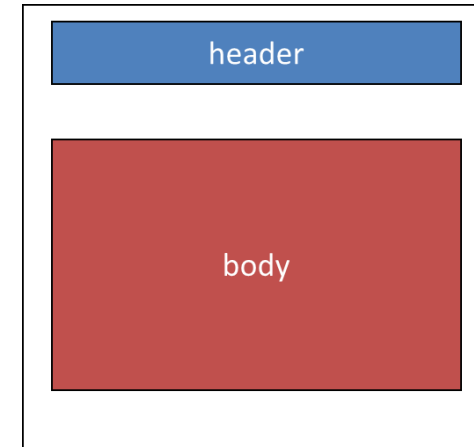
Formato del mensaje del protocolo SMTP

Definición

SMTP es un protocolo de la capa de aplicación y, como tal, emplea mensajes que se ajustan a un formato específico

Detalles esenciales

- Un mensaje SMTP se estructura en dos partes:
 - **Cabecera:** con los datos administrativos para el envío
 - Datos de transporte: destinatario, remitente, etc.
 - Otros datos: fecha, asunto, identificador, etc.
 - **Cuerpo:** contenido propiamente dicho del mensaje
- Algunos campos de la cabecera son:
 - **From:** remitente
 - **To:** destinatario
 - **Cc:** otros destinatarios
 - **Subject:** asunto
 - **Date:** fecha de envío
 - **In-Reply-To:** si es en respuesta a un mensaje
 - **Message-Id:** identificador del mensaje
 - **Return-path:** ruta de regreso



Extensiones MIME para correo electrónico



Extensiones MIME - Introducción

Problema

*Inicialmente el formato de los mensajes de correo únicamente permite transmitir **mensajes de texto**, sin otros elementos*

Solución

Ante esta situación se definen en los RFC 2045 a 2049 las extensiones multipropósito de correo Internet (MIME, *Multipurpose Internet Mail Extensions*). Con ellas se permite:

- Incluir en los mensajes caracteres que no forman parte de la **codificación ASCII**
- Definir un conjunto **extensible de formatos** para los mensajes
- Que los mensajes puedan estar constituidos de **varias partes** independientes, cada una con su formato

Nuevos campos en la cabecera de los mensajes para usar MIME:

- **MIME-Version**: versión MIME utilizada
- **Content-Type**: formato de los datos del mensaje
- **Content-Transfer-Encoding**: codificación utilizada para convertir los datos del mensaje ya que estos datos deben ser caracteres ASCII

Así, cuando llega un nuevo mensaje con estos campos MIME el cliente debe:

- **Descodificarlo** obteniendo el contenido original según el valor de Content-Transfer-Encoding
- En función del Content-Type **procesarlo** de forma adecuada, según sea imagen, audio, etc.

Extensiones y tipos MIME - Ejemplo de uso

Envío de una imagen por correo electrónico:

- Se ha enviado una imagen en **formato JPEG**
- Los bytes del archivo de imagen se han codificado en **base64** para que se puedan transportar por correo

- El tipo MIME se especifica con la directiva **Content-Type**
- Sintaxis: **Tipo/Subtipo; [parámetros]**

- Algunos tipos MIME usuales:

- **Text/[plain, html, ...]**
- **Image/[jpeg, gif, ...]**
- **Audio/[basic, 32kadpcm, ...]**
- **Video/[mpeg, quicktime, ...]**
- **Aplication/[mword, octet-stream, ...]** datos que deben ser procesados por el lector antes de ser “visionados”
- **Multipart/[mixed, paralell, ...]** para crear correos con distintos tipos de objetos/ficheros (textos, imágenes, audio, ...)

Versión MIME
Método utilizado
de datos codificados
Datos multimedia
tipo, subtipo,
declaración de parámetros
Datos codificados

```
From: alicia@crepes.fr
To: roberto@hamburger.edu
Subject: Imagen de mi piso
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

datos codificados base64.....
.....
.... datos codificados base64
```

MIME - Correos con varios objetos/archivos

Objetivo

Enviar varios **objetos de distintos tipos** (imágenes, textos, etc.) en un mensaje

Procedimiento

- Se introduce la cabecera
Content-Type: multipart/mixed
- El mecanismo define:
 - Dónde comienza y termina cada objeto con el campo **boundary**
 - Cómo está codificado cada objeto que no es texto ASCII, con el campo **Content-Transfer-Encoding**
 - El tipo de contenido de cada objeto con el campo **Content-Type**

```
From: alicia@crepes.fr
To: roberto@hamburger.edu
Subject: Imagen de un delicioso crepe.
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=StartOfNextPart
```

```
--StartOfNextPart
Querido Roberto, Te envío una imagen de un crepe.
--StartOfNextPart
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
datos codificados base64 .....
.....datos codificados base64
--StartOfNextPart
Dime si quieres tener la receta
```

Cabecera

Datos

Texto

Imagen

Texto

Protocolos de entrega de correo al usuario final



Protocolos de entrega de correo - Introducción

Necesidad

SMTP asume que el agente de usuario (cliente de correo) está en el mismo host que el servidor de correo, pero actualmente ya no suele ser el caso y los usuarios descargan en su ordenador los mensajes del servidor

Aspectos generales

- Existen dos protocolos que permiten realizar la descarga de correo:
 - **POP3**: Protocolo de Oficina Postal v.3 [RFC 1939]
 - **IMAP**: Protocolo de Acceso al Correo Internet [RFC 1730]
- Estos protocolos evitan el tener un servidor de correo local en la máquina del usuario y centralizar los buzones en máquinas servidoras destinadas para ello
- El envío desde una máquina local a un servidor sigue haciéndose con SMTP

Protocolos de entrega de correo - POP3

Características

Se trata de un protocolo muy simple, pero también limitado

Funcionamiento

- Los clientes conectan con el servidor a través del puerto TCP 110
- Consta de tres fases:
 - **Autorización:** mediante usuario/clave
 - **Transacción:** recuperación de mensajes que pueden ser borrados del servidor o no
 - **Finalización:** envío del commando **quit** por parte del cliente
- POP3 mantiene información de estado de la sesión activa, pero no conserva el estado entre sesiones

Protocolos de entrega de correo - POP3 - Ejemplo

Fase de autorización

- Comandos del cliente:
 - **user**: identificación del usuario por su nombre
 - **pass**: contraseña
- Respuestas del servidor:
 - **+OK**
 - **-ERR**

Fase de transacción, cliente:

- **list**: enumera los números de los mensajes que hay en el buzón
- **retr**: recupera un mensaje determinado por su número
- **dele**: elimina un mensaje
- **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
```

Protocolos de entrega de correo - IMAP

Características

Ofrece mayor funcionalidad que POP3, pero también es un protocolo mucho más complejo

Funcionalidad

- Permite definir **buzones** (carpetas) en el servidor para clasificar el correo
- Mantiene información de **estado** entre sesiones
- Permite obtener solo **partes de un mensaje**, algo útil si se tiene una conexión con poco ancho de banda ya que se pueden descargar únicamente los encabezados de los mensajes

Actividad - Acceso interactivo a servidor POP3

Acceso desde el terminal

- **Instala** si es necesario el programa Telnet en tu sistema, usando para ello el comando **sudo apt install telnet**
- **Conecta** con el servidor POP3 de tu proveedor de correo a través del puerto 110 con la orden **telnet servidor.correo 110**
- **Identifícate** con el comando **USER** y tu nombre y facilita la contraseña con el comando **PASSWD**
- Mediante el comando **LIST** obtén la **lista de mensajes** existentes en el buzón
- Usa el comando **RETR** para **leer** uno de los mensajes
- **Cierra la conexión** con el servidor mediante el comando **QUIT**

```
[usuario@servidor ~]$  
[usuario@servidor ~]$ telnet pop3.ujaen.es 110  
Trying 150.214.170.44...  
Connected to pop3.ujaen.es.  
Escape character is '^I'.  
+OK Hello there.  
USER [REDACTED]  
+OK Password required.  
PASS [REDACTED]  
+OK logged in.  
LIST  
+OK POP3 clients that break here, they violate STD53.  
1 2814  
2 6354  
3 4304  
4 18758  
.  
RETR 1
```

Cuestiones clave de este tema



Cuestiones clave

Qué deberías saber

*Al inicio de este tema se planteaban unos objetivos específicos que deberían permitirte **responder a las siguientes cuestiones clave***

Cuestiones

- ¿Qué lugar ocupa la capa de aplicación en la arquitectura TCP/IP?
- ¿Cuáles son los servicios que demanda de la capa de transporte?
- ¿Por qué se precisa un mecanismo de multiplexación y en qué consiste?
- ¿Cómo se comunican los protocolos de la capa de aplicación?
- ¿Cuál es la finalidad y funcionamiento del protocolo FTP?
- ¿Cuál es la finalidad y funcionamiento de los protocolos SMTP, POP3 y SMTP?
- ¿Cómo usar MIME para superar las limitaciones originales en la transferencia de mensajes de correo?

Material adicional

Descripción

Para ampliar tus conocimientos sobre los contenidos de esta semana te recomendamos que consultes los recursos indicados a continuación.

Recursos

- **Capítulo 2** La capa de aplicación, del libro Redes de computadoras 7ED disponible en [formato digital](#) en la BUJA (recuerda identificarte para poder acceder a leerlo desde tu navegador)
- **TCP/IP Key Applications** en el recurso electrónico [The TCP/IP Guide](#), donde encontrarars todos los detalles sobre FTP, SMTP, POP3 e IMAP
- **IETF Tools** (<https://tools.ietf.org>) te permite examinar los RFC de todos los protocolos de Internet, por ejemplo en <https://tools.ietf.org/html/rfc1939> se detalla el protocolo POP3