# Protocolos del nivel de aplicación. Índice

5. Introducción.
6. Protocolo TELNET.
    a. NVT. Network Virtual Terminal.
    b. Funcionamiento.
    c. Opciones.
    d. Comandos de acceso remoto.
7. Protocolo FTP.
    a. Servidor FTP.
    b. Permisos de acceso.
    c. Establecimiento conexión
    a. Modos de transmisión y comandos.
2. Protocolo TFTP.
3. Sistema DNS.
    a. Nombrado.
    b. Gestión.
    c. Funcionamiento.
    d. Tipos de servidores.
    e. Formato de mensajes.
4. Correo electrónico.
    a. Protocolo SMTP
    b. Protocolo POP.

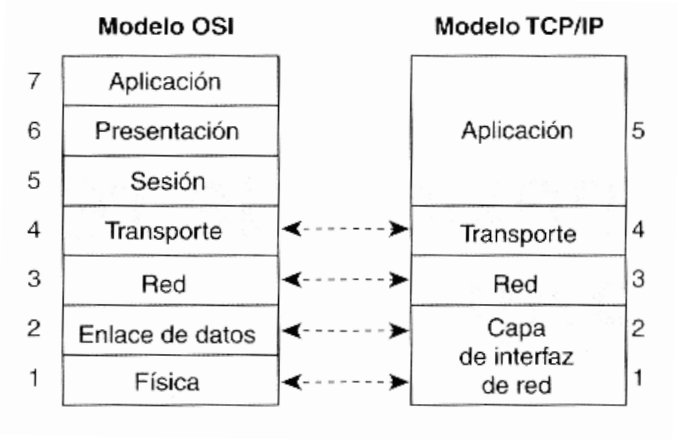
# Introducción

## Finalidad de la Capa 7: Capa de Aplicación

Interfaz con el usuario, aplicación final de comunicaciones.

Servicios sobre TCP:

- **TELNET** (Terminal Network).
- **SSH** (Secure Shell).
- **VNC** (Virtual Network Computing).
- **X** (XWindows Protocol).
- **FTP** (File Transfer Protocol).
- **SMTP** (Simple Mail Transfer Protocol).
- **POP** (Post Office Protocol).
- **LDAP** (Lightweight Directory Access Protocol).

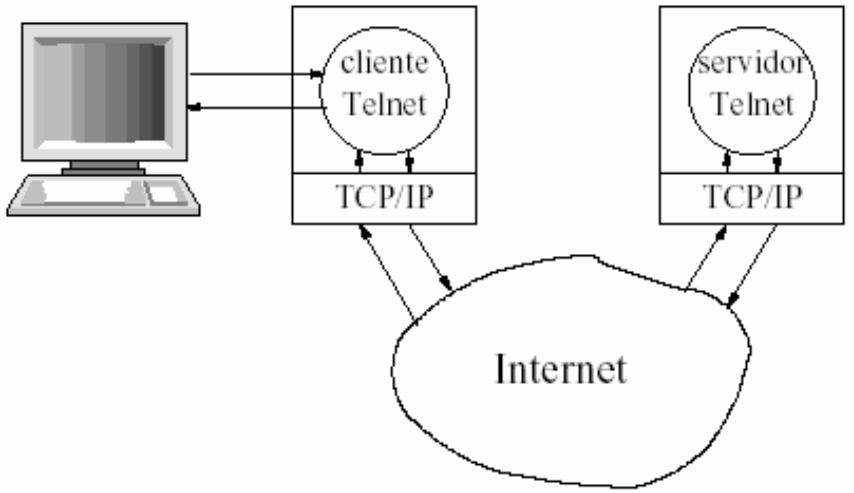



Servicios sobre UDP:

- **DNS** (Domain Name Service).
- **RIP** (Routing Information Protocol).
- **NFS** (Network File System).
- **SNMP** (Simple Network Management Protocol).

# Protocolo TELNET

- Permite conectarse a otro ordenador de forma remota.
- TELNET funciona de forma simétrica, pemitiendo a dos programas intercambiar información.
- El servicio es transparente porque el terminal parece conectado directamente al servidor.
- Define un lenguaje estándar intermedio, NVT para interconexión de sistemas heterogeneos.
- Al inicio se produce una negociación de opciones.
- Se envian caracteres de 7 bits y se emplea el octavo bit para marcar los caracteres de control.
- Por defecto:
  - Las líneas acaban en CR-LF.
  - Se hace eco en local y se envian lineas completas.
  - Funcionamiento semiduplex.



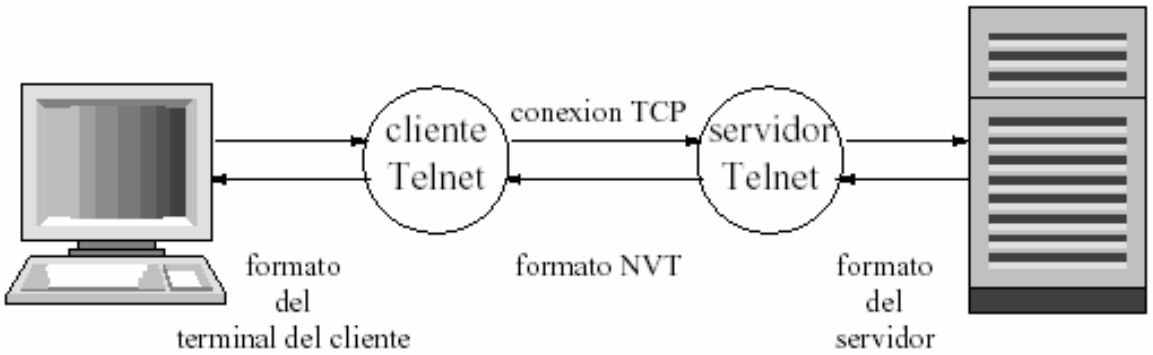
# NVT. Network Virtual Terminal

- El servicio es autenticado, requiriendose como paso previo que el usuario se identifique ante el servidor mediante una palabra de paso.

- NVT consta de un conjunto de 95 caracteres más 33 de control.

- Si bien el servicio es identificado, la password se transfiere en texto plano hacia el servidor. Esta password se podría interceptar y ser usada sin consentimiento. Para evitar este problema se emplea SSH.

- SSH proporciona autenticación y privacidad en las comunicaciones.



# Opciones de negociación

- El control de la conexión se lleva a cabo mediante “secuencias de escape”, en las que todo carácter de control va precedido de IAC (interpretar como comando) de código 255.
- La negociación de opciones es simetrica: se envia IAC WILL X o IAC W’ONT X, el otro extremo responde IAC DO X o IAC DO’NT X.

**Tabla 11.1** Ejemplos de comandos y opciones TELNET

| Comando NVT | Opción NVT |
|---|---|
| IAC *(interpret as command)* | |
| IP *(interrupt process)* | Transmit-Binary |
| AO *(abort output)* | |
| BRK *(break)* | |
| EL *(erase line)* | Echo |
| AYT *(are you there)* | |
| DM *(data mark)* | |
| WILL | Terminal-Type |
| WON’T | |
| DO | Linemode |
| DON’T | |

# Comandos de acceso remoto UNIX

Hay otras utilidades para acceso remoto en UNIX.

- **rlogin**. Autoriza el acceso sin contraseña de forma selectiva.
- **rsh**. Permite ejecutar comandos de forma remota.
- **rcp**. Permite realizar copia de ficheros entre ordenadores remotos.
- **ssh y scp**. Comandos similares a rsh y rcp que cifran la información para evitar que se filtre.
- **X**. Protocolo completo de comunicaciones para conexión de interfaz gráfica a sistemas unix, denominado XWindows.
- **vnc**. Acceso en modo gráfico a sistemas heterogeneos (unix, windows, mac).

# Transferencia de ficheros con FTP

FTP es el servicio de transferencia de ficheros entre hosts remotos.

- Permite a **usuarios autorizados** listar ficheros de un sistema remoto e intercambiar ficheros de un sistema a otro.
- FTP maneja **varios formatos de fichero** y puede hacer conversiones entre ellos, por ejemplo con ficheros ASCII.
- FTP puede usarse por programas o por usuarios interactivos.
- FTP emplea **dos conexiones TCP** independientes para datos y para control.
- FTP **emplea TELNET** para la conexión de control.
- El servidor FTP permite dos tipos de acceso:
  - **Usuario FTP**. Para usuarios con cuenta en esa máquina.
  - **Anonymous**. Para cualquier usuario de la red.

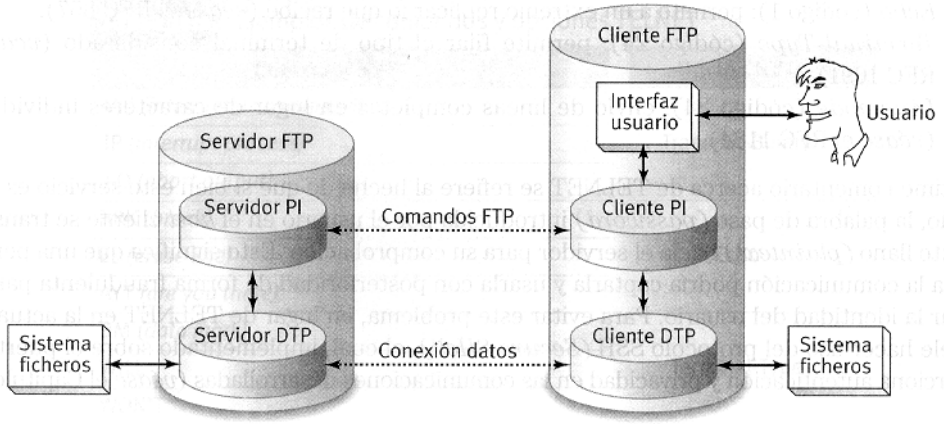
- El servidor FTP está implementado en unix como el programa **ftpd**. Se suele controlar desde el superserver inetd bajo demanda.

# Acceso al Público FTP

- Los usuarios con cuenta pueden acceder a todos los ficheros según el permiso establecido en unix.

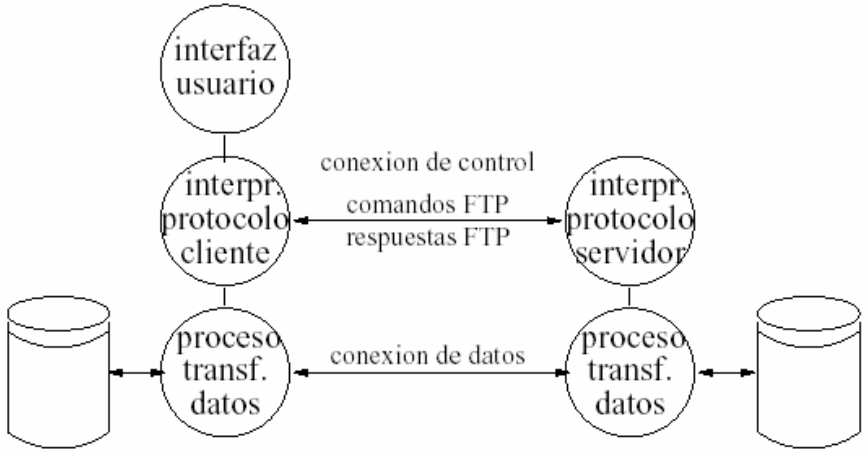
- Los usuarios ftp están restringidos al “home” del usuario ftp. Debe tener una estructura similar a:

  - bin. Ejecutables necesarios.
  - etc. Ficheros de configuración.
  - pub. Información servida.
  - pub/incoming. Directorio donde se permite que se depositen y borren ficheros.



# Establecimiento de Conexión

- Para cada transferencia se crean dos conexiones:
  - **Conexión de control**. Puerto 21.
  - **Conexión de datos**. Para cada transferencia se crea una conexión de datos por el puerto 20.




- El cliente inicia la conexión de control TELNET NVT y envia comandos al servidor. Si es necesario se establecen conexiones de datos.

# Modos de transmisión y comandos (I)

Disponemos de los siguientes modos de transmisión:

- **Modo “stream"**: Fichero como un flujo de octetos.
  - EOR y EOF con códigos de control.
  - No hay posibilidad de rearranque.
- **Modo bloque**: Bloques de datos con cabeceras.
- **Modo comprimido**: Compresión muy primitiva.

Disponemos de los siguientes comandos:

| | |
|---|---|
| USER | MODE |
| PASS | RETR |
| CWD | STOR |
| CDUP | DELE |
| QUIT | MKD |
| PORT 212,128,1,45,10,50 | PWD |
| PASV | LIST |
| TYPE | |

La respuesta está constituida por los siguientes elementos:

- 3 dígitos decimales (código de respuesta).
- 1 espacio.
- Mensaje en inglés.
- CR-LF.

# Modos de transmisión y comandos (II)

Los códigos de respuesta están formados por un número de tres cifras que puede ser alguno de los siguientes:

1XY Respuesta positiva preliminar.
2XY Respuesta positiva completa.
3XY Respuesta positiva intermedia.
4XY Respuesta negativa temporal.
5XY Respuesta negativa permanente.
X0Y Sintaxis.
X1Y Información.
X2Y Conexiones.
X3Y Cuentas y autenticación.
X4Y No especificada.
X5Y Sistema de ficheros.

A continuación se dan algunos códigos de respuesta:

200 OK
500 Syntax error.
230 User logged in.
331 User name OK, need password.
150 File status OK, about to open data connection.

# Sesión de Ejemplo

**“ftp S” : Conexión de control al servidor S en el puerto 21 desde el puerto P**

*<---220 Service ready*
*Login: nombre*
*---> USER nombre*
*<---331 User name ok, need password*
*Password: contra*
*---> PASS contra*
*<---230 User logged in*
*get test mitest*

**El Cliente abre el fichero *mitest* para escritura.**
**El Cliente espera conexiones en el puerto P.**

---> RETR test
<---150 File status OK, about to open data connection

**El Servidor abre conexión de datos (puerto 20 a puerto P).**

<---226 Closing data connection, file transfer successful
ascii ---> TYPE A
<---200 Command OK
put mitexto texto

**El Cliente abre el fichero mitexto para lectura.**
**El Cliente espera conexiones en el puerto P.**

---> STOR texto
<---550 Access denied
quit
---> QUIT

**El Servidor cierra todas las conexiones.**

# Protocolo TFTP

- Proporciona un servicio poco sofisticado para transferencia de ficheros.
- Es similar a FTP pero en el mensaje de petición se envía de manera directa datos del host remoto, del fichero fuente y destino y del sentido de la transferencia.
- TFTP emplea UDP con retransmisión en bloques de 512 bytes con reconocimiento por bloque.
- TFTP es un protocolo simple que soporta varios tipos de archivo y se emplea para el arranque del sistema operativo de sistemas sin disco a través de la red.

# Sistema de Nombres de Dominio: DNS

- Permite nombrar los host y destino de correo electrónico en Internet **empleando un nombre en lugar de su dirección IP**.

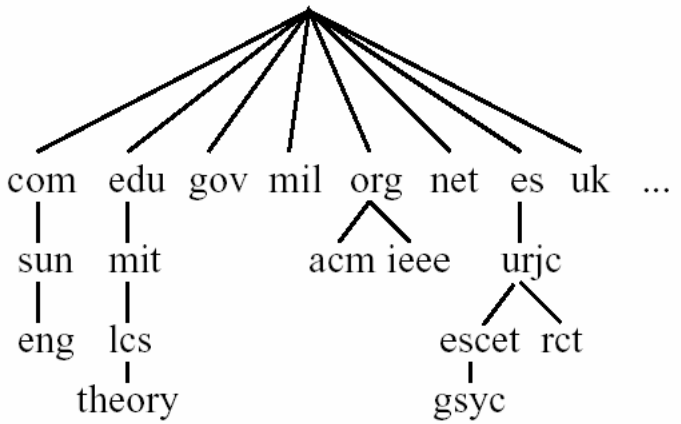
- DNS se basa en un **esquema jerárquico** y una **base de datos distribuida**.

- El procedimiento para convertir un nombre en dirección IP es el "**resolver**", este según su configuración en resolv.conf envía una petición DNS a un servidor DNS para obtener la dirección IP a partir del nombre.

- Es un **proceso transparente** al usuario. Se lleva a cabo de forma automática.

# Estructura de Nombrado

- El control de los nombres se descentraliza, consiguiendo una estructura jerárquica de dominios:
  - Dominio raiz.
  - Dominios de nivel máximo.
  - Dominios secundarios…

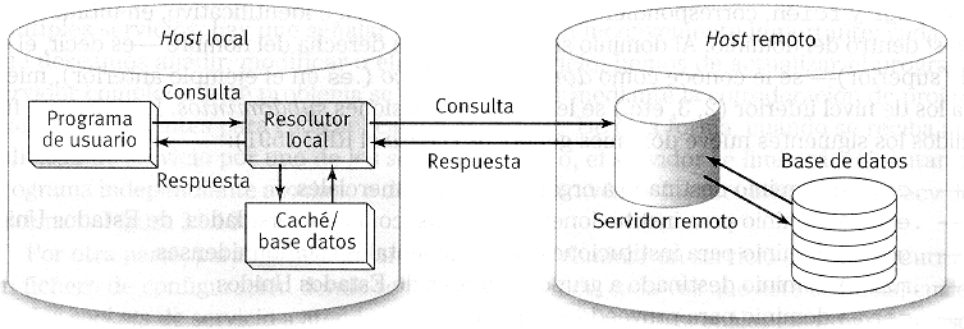



- El acceso al servidor puede ser:
  - Dominio directo: proporciona IP
  - Dominio inverso: proporciona nombre.

- Dominio inverso se indica con el formato x.y.in-addr.arpa, donde x e y forman parte de la dirección ip y.x.0.0

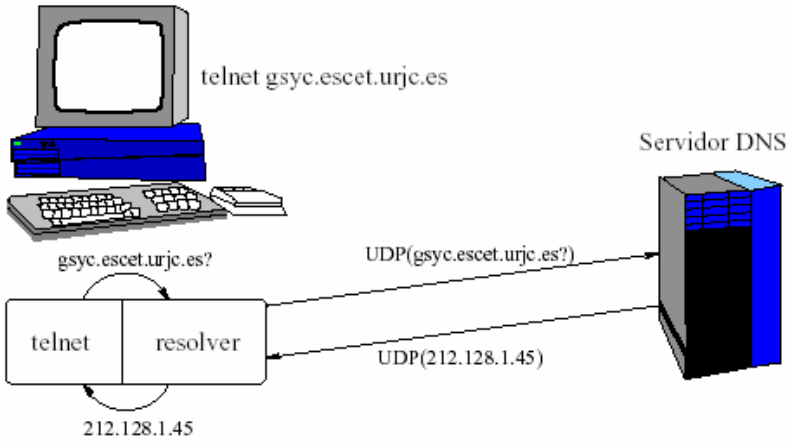
# Estructura de Gestión

- Cada vez que se delega un dominio se delega también su gestión y su subdivisión.
- Si el gestor de dominio es delega un subdominio uhu, el administrador del subdominio puede a su vez crear más subdominios y delegar a otros su gestión.
- Hay dominios (com, org) gestionados por varios registradores de dominio en regimen de competencia.
- La aplicación para traducir nombres a IPs es el “resolver”. La librería del DNS emplea una llamada gethostbyname() que devuelve la dirección ip de un host:
  - Mirando /etc/hosts
  - Consultando un servidor local
  - Consultando un servidor en otra máquina



# Esquema de Funcionamiento DNS (I)

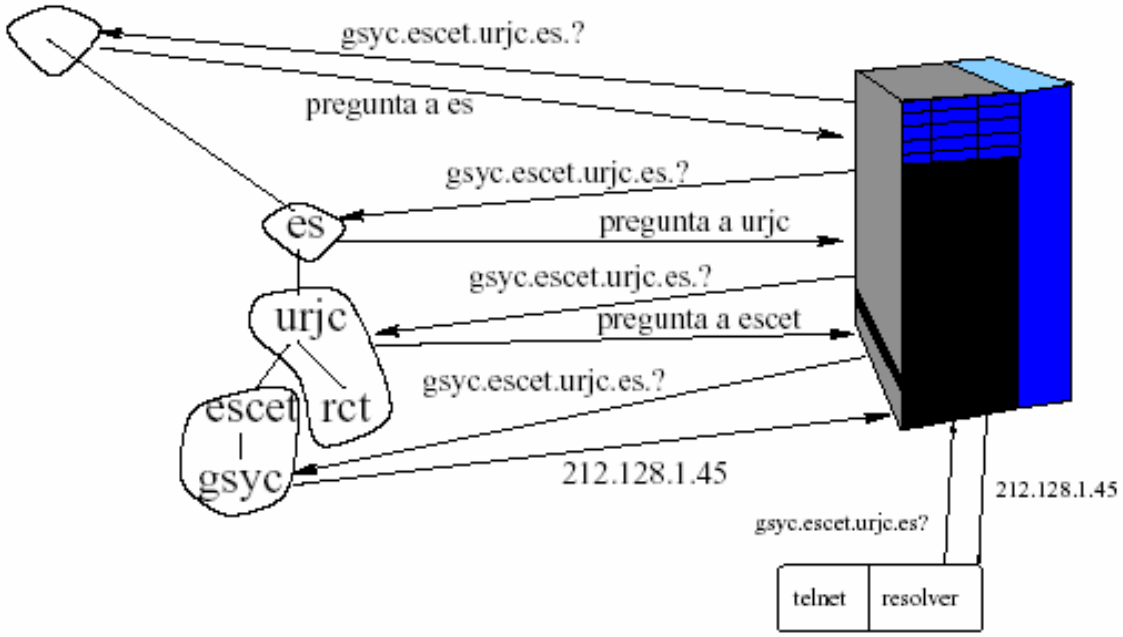
- Cuando un cliente quiere resolver un nombre pregunta al servidor DNS.
- El servidor investiga por su cuenta y devuelve la dirección pedida.




- Responden dos tipos de consultas:
  - **Recursivas**: obligan al servidor a hacer varias consultas.
  - **Iterativas**: peticiones de otro servidor, a las que responden con la IP de otro servidor de la jerarquia.

# Esquema de Funcionamiento DNS (II)

- Ejemplo de consulta DNS sobre gsyc.escet.urjc.es.
  - o Comprueba si está en su mapa local.
  - o Pregunta a servidor dominio raiz, contesta IP del siguiente dominio
  - o Pregunta a ese, obteniendo dirección del servidor del siguiente nivel.
  - o Pregunta al último que la tiene en sus tablas.



# Tipos de Servidores DNS

- Según como son utilizados:
  - **Reenviador**. Lo usan antes de consultar a los demás para centralizar las consultas.
  - **Esclavo**. Usados por servidores (por ejemplo si hay cortafuegos)

- Según como reciben los datos:
  - **Primario**. Tiene la información actualizada.
  - **Secundario**. Copia del primario.
  - **Cache**. Guardan datos de las consultas.

- Según el lugar de procedencia del dato:
  - **Con autoridad** (*authoritative*). Tiene el mapa original para el dominio consultado.
  - **Sin autoridad**. Tiene el dato en su cache.

# Mapas de Dominio

- **BIND** es la implementación del servidor en UNIX. El cliente es una librería que se enlaza al sistema operativo y traduce las consultas al DNS.

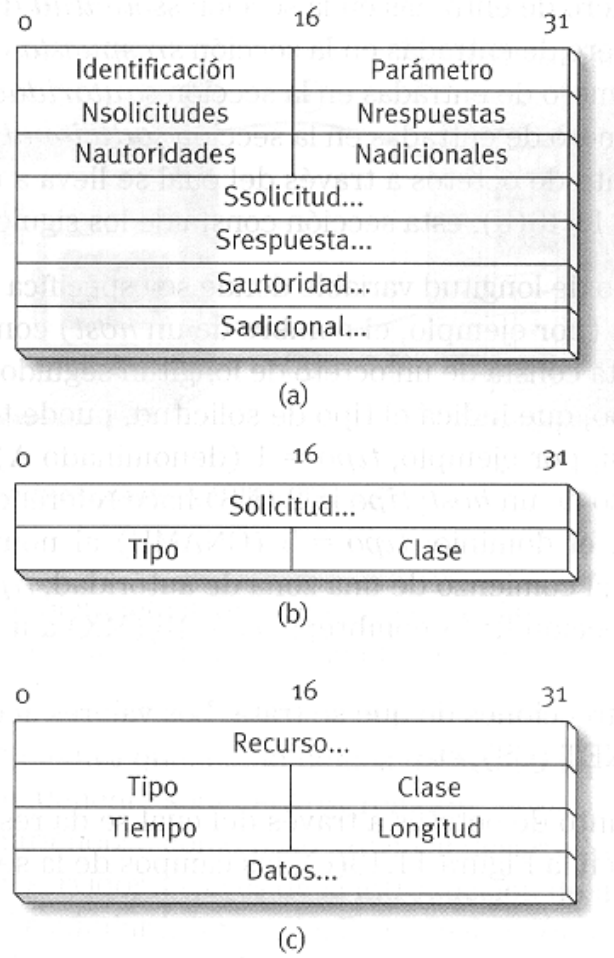
- El servidor named tiene varios ficheros de configuración, donde se especifican los dominos primarios y secundarios.

- Hay un fichero para cada dominio servido.

```
urjc.es.    172800  IN  SOA    venus.urjc.es.
                               root.venus.urjc.es. (
                               2000030702 ; Número de serie
                               86400      ; Refresco
                               7200       ; Reintento
                               2592000    ; Expiración
                               172800 )   ; Ttl
            172800  IN  NS     venus.urjc.es.
            172800  IN  MX     venus.urjc.es.
www         172800  IN  CNAME  venus.urjc.es.
venus       172800  IN  A      193.147.184.8
escet       172800  IN  NS     gsyc.escet.urjc.es
gsyc.escet  172800  IN  A      212.128.1.45
```

# Formato de Mensaje DNS

- El **mensaje DNS** está formado por paquetes con una cabecera de 12 bytes:
  - Identificación
  - Banderas
  - Número de consultas
  - Número de RRs de respuesta
  - Número de RRs de autoridad
  - Número de otros RRs
- **Datos** variable: Consultas, respuestas, etc.
- La **consulta** consta de:
  - Nombre de la petición.
  - Tipo de consulta.
  - Clase de consulta.
- Se usa el **puerto 53 TCP y UDP**:
  - El resolver hace consultas UDP.
  - El servidor responde con el mismo tipo.
  - Si la respuesta es mayor de 512 bytes, resolver repite con TCP.
  - Transferencias primario a secundario con TCP.



# Correo Electrónico. Introducción (I)

- **Muy utilizado** en Internet. No es necesario esperar a que la otra máquina esté en linea porque se emplea un servidor de mensajes.

- Se emplea una **comunicación extremo a extremo** que garantiza que el mensaje permanece en la máquina fuente hasta que ha sido copiado al destino.

- **SMTP** e emplea para el envio de mensajes. No autenticación, no encriptado.

- Todo mensaje consta de dos partes:
  - **Sobre**. Consta de cabeceras estandar para identificar los destinatarios y remitentes.
  - **Contenido**. Formado por el propio mensaje. Solo admite NVT-ASCII, para otras codificaciones se emplea MIME.

# Correo Electrónico. MIME

- Tipos de datos aceptados por el estándar MIME:

**Tabla 11.2** Tipos y subtipos aceptados para la cabecera `Content-Type` de MIME

| Tipo | Subtipo |
|---|---|
| text | plain |
| | richtext |
| image | gif |
| | jpeg |
| audio | basic |
| video | mpeg |
| application | octet-stream |
| | postscript |
| message | rfc822 |
| | partial |
| | external-body |
| multipart | mixed |
| | alternative |
| | parallel |
| | digest |
| x-*nuevotipo* | — |

# Correo Electrónico. Introducción (II)

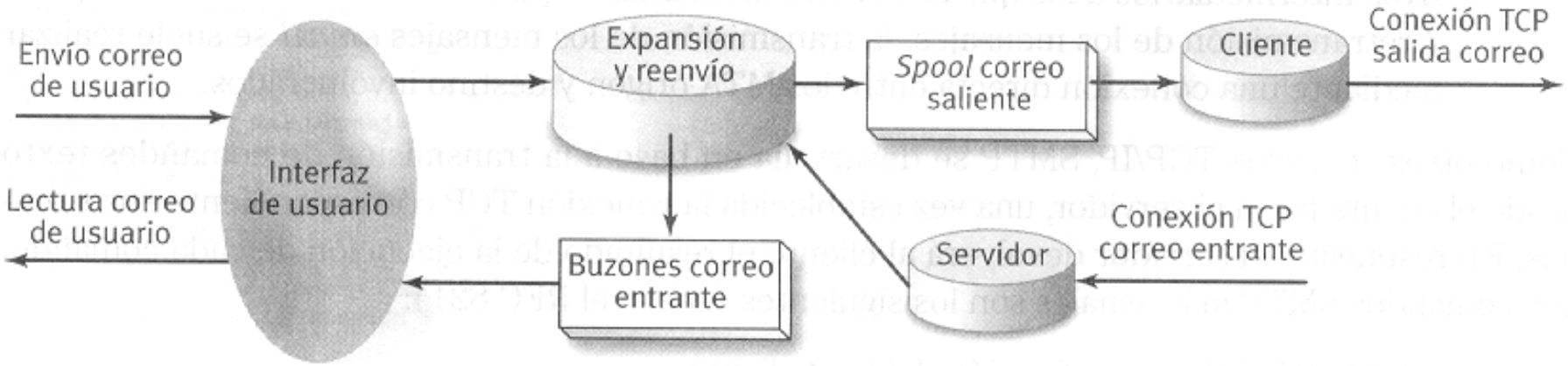
- La cabecera proporciona información al gestor de correo (MTA):
  - From: dirección origen.
  - To: dirección destino.
  - CC: direccion copia.
  - BCC: copia ciega.
  - Subject: tema.
  - Date: fecha
  - Received: por donde ha pasado la carta.
  - Message-id: número identificador de mensaje.

- Existen pasarelas a otros sistemas de correo y estándares como X400, cc:mail, etc.

- Se pueden adjuntar datos genericos (attachments).

# Protocolo SMTP

- SMTP define los **comandos** para comunicarse con el MTA.
- Secuencia de eventos:
  - o El programa transfiere mensaje a UA.
  - o Mensajes se almacenan temporalmente en cola.
  - o La cola se procesa periodicamente transfiriendo desde el MTA emisor al MTA destinatario. Puede haber algún MTA Relay.
  - o Al llegar al MTA destino el mensaje queda almacenado en el buzón o se entrega al UA.



# Procedimiento de envío (I)

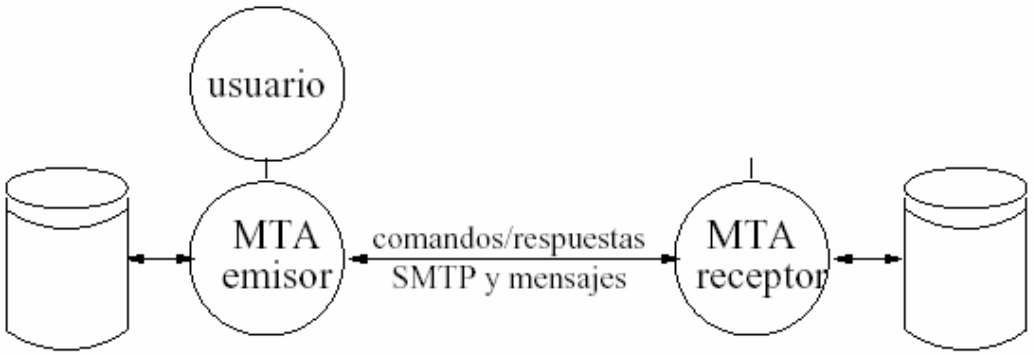
- El procedimiento de conexión es el siguiente:
  - o Al establecer conexión TCP:25 receptor saluda.
  - o **HELO dominio**. El emisor se identifica.
  - o **QUIT**. El emisor no tiene más que enviar.

```
(Telnet pantuflo.escet.urjc.es 25)
Connecting to pantuflo.escet.urjc.es (ether)...
220 pantuflo.escet.urjc.es
Sendmail SMI-8.6/SMI-SVR4 ready at Mon, 7 Sept 1998
>>> HELO a202e12.escet.urjc.es
250 pantuflo.escet.urjc.es
Hello a01-unix [192.2.3.14], pleased to meet you
>>> MAIL From:<alumno@a202e12.escet.urjc.es>
250 <alumno@a202e12.escet.urjc.es>... Sender ok
>>> RCPT To:<jcenteno@pantuflo.escet.urjc.es>
250 <jcenteno@pantuflo.escet.urjc.es>... Recipient ok
(Pueden ir varias RCPT seguidas)
>>> DATA
354 Enter mail, end with "." on a line by itself
Subject: Ejemplo
Texto del mail
.
250 MAA29247 Message accepted for delivery
>>> QUIT
221 pantuflo.escet.urjc.es closing connection
```

# Procedimiento de envío (II)

- El procedimiento de envio es el siguiente:
  - **MAIL FROM: <>** comienza transacción de envio, sirve para errores.
  - **RCPT TO: <>** camino que identifica uno de los receptores del mensaje.
  - **DATA**: contenido del mensaje con todas las cabeceras.Finaliza con un punto en linea vacia.

- El camino de ida se puede especificar en forma de ruta. Cuando el mensaje llega a una MTA, esta pone su identificador en el sobre, que no aparecerá en el contenido.



# Protocolo POP (I)

- Permite **traer los mensajes de un buzón** en una máquina remota.
- Usa TCP y tiene un funcionamiento parecido a SMTP.
- Emplea varios comandos pero solo **dos respuestas**: OK y ERR.
- Permite la **autenticación de los usuarios**, protocolo no encriptado.
- Comandos de **autorización**:
  - **USER** <>. Identifica al usuario.
  - **PASS** <>. Da la contraseña.

# Protocolo POP (II)

Comandos en modo transacción:

- **STAT**. Número y tamaño total de mensajes.
- **LIST**. Numero de mensaje y tamaño.
- **RETR** <>. Recupera el mensaje.
- **DELE** <>. Se etiqueta para borrar el mensaje.
- **NOOP**. No hace nada.
- **RSET**. Se quitan las etiquetas de borrado.
- **QUIT**. Salir y borra mensajes marcados.

```
#> telnet goliat.ugr.es 110
+OK QPOP (version 2.4) at goliat starting.
AUTH KERBEROS_V4
-ERR This command is not supported yet
USER usuario
+OK Password required for usuario.
PASS clave
+OK usuario has 2 messages (320 octets).
STAT
+OK 2 320
LIST
+OK 2 messages (320 octets)
1 120
2 200
.
RETR 3
-ERR Message 3 does not exist.
RETR 2
+OK 120 octets
<mensaje1> ...
.
DELE 2
+OK Message 2 has been deleted.
QUIT
+OK Pop server at goliat signing off.
#> ___
```

# Protocolo IMAP

- IMAP permite **manejar buzones** en una máquina remota.
- Emplea TCP puerto 143.
- El esquema de comandos y respuestas es **similar a SMTP y POP**.
- Integra **mecanismos de seguridad** más avanzados.
- POP es más sencillo que IMAP.
- IMAP facilita una gestión de buzones y mensajes más completa que POP, permitiendo que estos permanezcan en el servidor.