Escuela Politécnica Superior
Universidad de Huelva

TERCER CURSO. TECNOLOGÍA DE REDES

Tema 3: Protocolos del Nivel de Aplicación

Manuel Sánchez Raya
Versión 0.1
5 de Febrero de 2004

ÍNDICE

BIBLIOGRAFÍA

Apuntes año 2002-2003. Estefanía Cortés Ancos.
Apuntes Universidad de Oviedo. J.A.Sirgo, Rafael C. González
Academia de Networking de Cisco Systems. Guía del Primer Año.
Internetworking with TCP/IP. Vol. I. D.E. Comer.

1.- Nivel de Aplicación. Introducción.

El nivel de aplicación es la capa 7 del modelo OSI. Es el más cercano al usuario final al actuar recíprocamente con las aplicaciones de software, como enviar y recibir correo electrónico, transferencia de archivos y emulación de terminal. Incluye a todos aquellos programas que se sirven de la capa de transporte para distribuir datos:

- Sobre TCP:
  - **TELNET** (Terminal Network)
  - **FTP** (File Transfer Protocol)
  - **SMTP** (Sample Mail Transfer Protocol).
- Sobre UDP:
  - **DNS** (Domain Name Service). Asocia nombres a direcciones IP.
  - **RIP** (Routing Information Protocol). Se usa para intercambiar información de enrutado.
  - **NFS** (Network File System). Permite compartir datos entre varios host de la red.
  - **SNMP** (Simple Network Management Protocol)

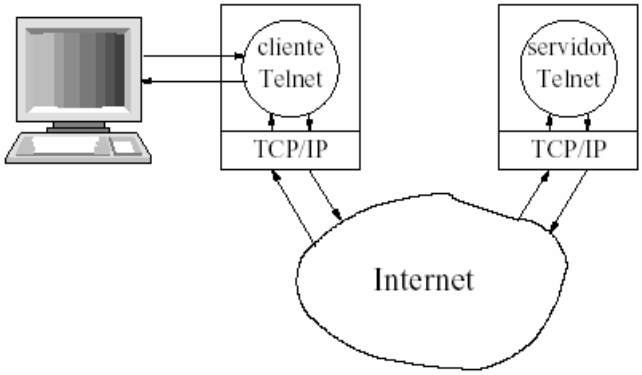
La mayoría de las aplicaciones que funcionan en un entorno de red se llaman aplicaciones cliente/servidor. Dichas aplicaciones tienen dos componentes: el lado cliente (ubicado en el host local) y el lado servidor (ubicado en un host remoto que proporciona servicios en respuesta a las peticiones del cliente).

2.- El protocolo TELNET.

TELNET permite a un usuario establecer una conexión TCP a un servidor de "login" en un lugar alejado. Aunque TELNET no es tan sofisticado como algunos protocolos de terminal remota, está disponible a lo largo de casi toda la red Internet.

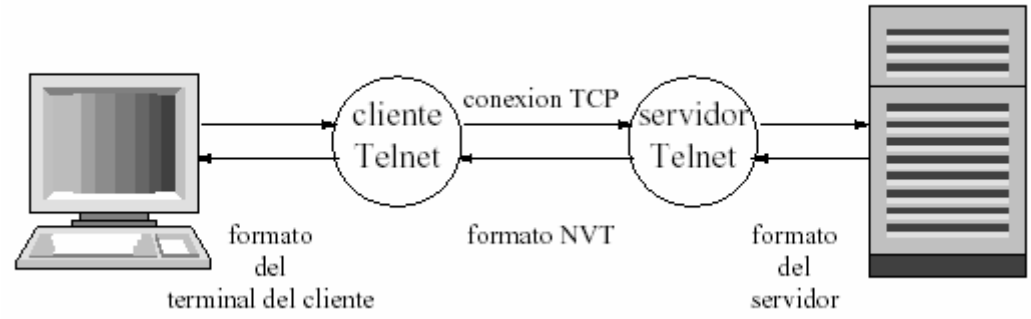
TELNET ofrece tres servicios básicos. En primer lugar, define un terminal virtual que proporciona una interfase con sistemas remotos. En segundo, incluye un mecanismo que permite al cliente y servidor negociar opciones y proporciona una serie de opciones estándar. Por último, TELNET trata a ambos lados de la conexión simétricamente. Así, en vez de forzar a un lado a conectarse a un terminal de usuario, permite a ambos lados de la conexión ser un programa.

Cuando un usuario invoca TELNET, un programa de aplicación en la máquina se convierte en cliente y contacta con un servidor en uno de los puertos TCP reservados, estableciendo una conexión sobre la que se comunicarán. El cliente acepta pulsaciones de tecla del terminal del usuario y las envía al servidor, a la vez que acepta caracteres que envía el servidor y los imprime en la pantalla del terminal.
El servicio es transparente, porque el terminal parece conectado directamente al servidor. Usa TCP, puerto 23.



Define un interfaz normalizado entre sistemas remotos: *network virtual terminal*, NVT. Al inicio, existe un mecanismo de negociación de opciones. Se tiene simetría en la comunicación. Ambos: cliente y servidor, son tratados como iguales.

2.1.- NVT. Network Virtual Terminal.



El Modo de funcionamiento por defecto es el siguiente:

- Caracteres de datos y control con 7 bits, comandos con el octavo bit a 1.
- Caracteres de control:
    - BEL (7): suena un pitido.
    - BS (8): Mueve un carácter a la izquierda.
    - HT (9): Mueve al siguiente tabulador horizontal.
    - LF (10): Mueve una línea abajo.
    - VT (11): Mueve al siguiente tabulador vertical.
    - FF (12): Mueve al principio de la siguiente página.
    - CR (13): Mueve al principio de la línea.
    - Resto (0-32) no hacen nada.

Modo de funcionamiento, por defecto:

- Líneas acaban con CR-LF.
- Se hace edición en local y se envían líneas completas.
- Se hace echo en local.
- Funcionamiento semiduplex. El turno se cede con el comando Go Ahead GA (249).

Los comandos sirven para controlar el proceso del servidor. Comandos básicos:

- BRK (243): Señal “break".
- IP (244): Interrumpe el proceso.
- AO (245): Aborta la salida.
- AYT (246): “Are you there?"
- EC (247): Borra carácter.
- EL (248): Borra línea.

Van precedidos de un carácter de escape *interpret as command* IAC (255).

2.2.- La señal SYNCH

Sirve para garantizar que los comandos enviados han sido procesados incluso si los buffers estaban llenos. Se genera como una notificación urgente TCP y el envío de un comando *data mark* DMARK (242).

La notificación urgente no es bloqueada y hace que se vacíen del buffer de entrada todos los caracteres hasta llegar al DMARK. Los datos se tiran, los comandos se procesan.

2.3.- Opciones.

Se puede negociar un comportamiento distinto del que se tiene por defecto con varias opciones. Ejemplos de opciones:

- Transmit Binary (0): Pasa a usar 8 bits para datos.
- Echo (1): Activa el eco remoto.
- Suppress-GA (3): Elimina la transmisión de GAs.
- Status (5): Pide el estado de una opción.
- Timing-Mark (6): Pide que inserte una marca de tiempo en el flujo en el otro sentido para sincronización.
- Terminal-Type (24): Pide información del terminal (marca y modelo).
- Linemode (34): Usa buffer local de líneas.
- y muchas mas.....

El Negociado de opciones se hace con los comandos WILL (251), WON'T (252), DO (253) y DON'T (254).

- WILL pide permiso para empezar a usar la opción. Se responde con DO o DON'T.
- DO pide que el receptor empiece a usar la opción. Se responde con WILL o WON'T.

2.4.- Comandos de acceso remoto.

Hay otras utilidades para acceso remoto (algunas con capacidades gráficas). En BSD UNIX, cuando se integra TCP/IP, se escriben una serie de utilidades:

- rlogin es una alternativa a telnet introducida en Unix BSD. Permite autorizar el acceso sin contraseña y rechazar el acceso de forma selectiva. Se agrupa con otros comandos con interfaz similar:

- rsh: permite ejecutar comandos de manera remota.

- rcp: permite copiar ficheros entre ordenadores remotos.

- ssh y scp son comandos similares a los comandos-r, pero cifran toda la comunicación. Disponibles para bastantes plataformas.

- X en un protocolo completo para conexión a la interfaz gráfica de los sistemas UNIX que es XWindows.

- Vnc es otro programa para el acceso en modo gráfico a sistemas UNIX que ejecutan el entorno XWindow.

3.- Transferencia de ficheros: FTP.

FTP es un servicio de transferencia de ficheros, muy utilizado en Internet, construido sobre un protocolo del mismo nombre: FTP, File Transfer Protocol. Permite transferir ficheros de todo tipo: imágenes, texto, programas. . . entre clientes y servidores de FTP.

El protocolo FTP (*File Transfer Protocol*) permite, a usuarios autorizados, entrar en un sistema remoto, identificarse y listar directorios remotos, copiar ficheros desde ó a la máquina remota y ejecutar algunos comandos remotos. Además, FTP maneja varios formatos de ficheros y puede hacer conversiones entre las representaciones más utilizadas (por ejemplo, entre EBCDIC y ASCII). FTP puede ser usado por usuarios interactivos así como por programas.

El protocolo FTP permite al usuario acceder a varias máquinas en una misma sesión. Mantiene dos conexiones TCP independientes para control y transferencia de datos. Usa el protocolo TELNET para el control de la conexión. La implementación del protocolo FTP depende del sistema operativo usado, aunque casi todas siguen el mismo patrón. En el lado del servidor, un proceso de aplicación, S, corre esperando por una conexión en el puerto asignado para TCP. Cuando un cliente abre una conexión con ese puerto, el proceso S lanza un nuevo proceso de control, N, para manejar la conexión, y vuelve a esperar por otro cliente. El proceso N se comunica con el cliente por medio de la llamada "conexión de control" y cuando recibe una petición de transferencia inicia otro proceso adicional, D, que abre otra conexión con el cliente que solamente se usa para la transmisión de datos. Una vez que la transferencia de datos ha concluido, el proceso D cierra la conexión y termina. El cliente vuelve a su interactividad con el proceso N y puede solicitar otra transferencia de datos.

Un servidor de FTP permite 2 tipos de acceso:

- **Usuario FTP**: para usuarios con cuenta en esa máquina. Hay que especificar login y password.
- **Anonymous**: para cualquier usuario de la red.

El usuario *anonymous* (abreviado como *ftp*) permite la entrada a cualquier usuario, tenga o no cuenta en la máquina. El usuario *anonymous* está restringido sólo a una parte del sistema de ficheros en la que está instalado el servidor de FTP. Si queremos permitir el acceso de usuarios (*anonymous*), hay que crear un usuario ftp. El servidor ftpd chequeara la existencia del usuario ftp antes de permitir un acceso mediante el usuario anonymous. La línea del /etc/passwd en UNIX para el usuario *anonymous* debe ser similar a:

```
ftp:*:11:50:Anonymous FTP:/usr/local/share/infosys/ftp:
```

Existen otras aplicaciones más simples como rcp (remote copy) o tftp (sobre UDP).

3.1.- Servidor FTP: el demonio ftpd.

En Unix el servidor es llamado ftpd o in.ftpd. Se encuentra en /usr/sbin. El servidor de FTP funciona normalmente bajo el control del demonio *inetd*, y utiliza el puerto TCP número 21. Cada vez que un cliente realiza una petición sobre ese puerto, inetd se encarga de arrancar un ftpd.

El demonio inetd se encarga de estar escuchando en los puertos TCP o UDP y de arrancar los servicios necesarios bajo demanda. La línea del fichero de configuración del *inetd* (/etc/inetd.conf) debe ser similar a:

```
# <service_name> <sock_type> <proto> <flags> <user> <server_path> <args>
      ftp          stream      tcp    nowait  root  /usr/sbin/ftpd /usr/sbin/ftpd
```

3.2.- Permisos de acceso al sistema de ficheros

Los usuarios con cuenta en el sistema, pueden acceder a todos los ficheros que tienen accesibles mediante telnet. Los usuarios *anonymous* están restringidos a aquello que se encuentra bajo el directorio HOME del usuario ftp del sistema. El sistema de ficheros para los usuarios *anonymous* debe tener una estructura determinada, y contener ciertos ficheros y ejecutables.

La estructura básica de subdirectorios que debe existir es:

- bin: Contiene todos los ejecutables necesarios; Normalmente sólo ls.
- etc: Contiene ficheros de configuración. Por ejemplo: copias reducidas del /etc/passwd y /etc/group.
- pub: Este directorio contiene la información servida.
- pub/incoming: directorio donde se permite que se depositen y se borren ficheros.

Los ficheros /etc/passwd y /etc/group que utiliza el servidor de FTP nunca deben ser copias idénticas de los reales de la máquina. Puede ser muy peligroso para la seguridad de la máquina si están accesibles mediante FTP.

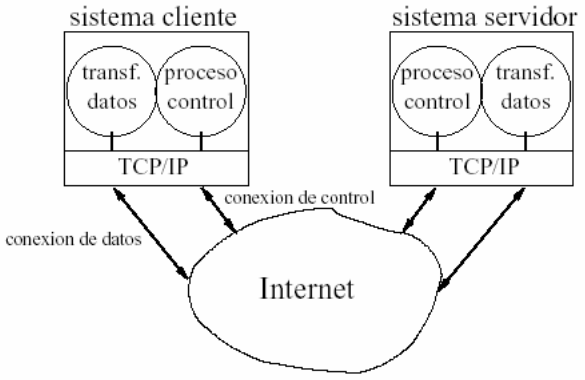
El Protocolo FTP (RFC 959) se apoya en TCP, puertos 20 y 21 y ha sido diseñado para uso interactivo o por programas. Permite definir formatos de ficheros (binario, ASCII, EDCDIC) y controla el acceso (autenticación).

3.3.- Conexiones.

Por cada transferencia de archivo se crean dos conexiones:

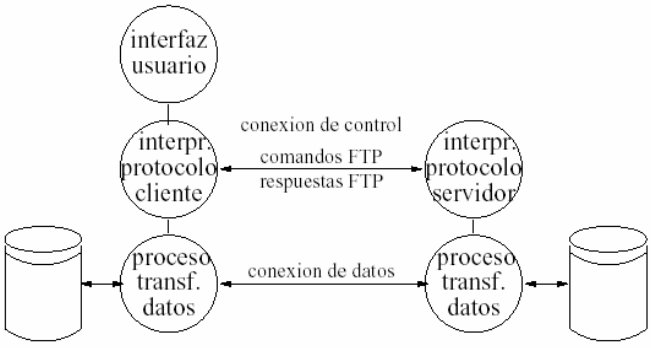
- **Conexión de control**: Un servidor FTP espera conexiones en el puerto 21 (puerto de control bien conocido).

- **Conexión de datos**: Para cada transferencia de datos se crea una conexión de datos. Las conexiones de datos usan el puerto 20 del servidor. El cliente reserva un puerto para ello y se lo comunica al servidor mediante la conexión de control.



3.4.- Establecimiento sesión FTP.

El intérprete del cliente (IC) inicia la conexión de control (Telnet NVT). IC envía comandos al intérprete del servidor (IS). Los dos primeros identifican al usuario y dan su contraseña. IS envía respuestas adecuadas al IC. Si es necesario, se establecen y usan conexiones de datos.



3.5.- Modos de transmisión y comandos.

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.

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.

3.6.- Sesión de ejemplo.

Una session de ejemplo puede ser algo parecido a lo siguiente:

- "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.

3.7.- El protocolo TFTP.

El protocolo TFTP (*Trivial File Transfer Protocol*) proporciona un servicio barato y poco sofisticado de transferencia de ficheros.

Al contrario que FTP, el protocolo TFTP no utiliza un servicio fiable de transmisión. No utiliza el protocolo TCP, sino que se basa en el protocolo UDP que corre por encima del IP, pero no es tan complejo como el TCP. TFTP utiliza temporización y retransmisión para asegurar que los datos llagan a su destino. La máquina fuente transmite un fichero en bloques de tamaño fijo (512 bytes) y espera por un acuse de recibo para cada bloque antes de enviar el siguiente. La máquina destino reconoce cada bloque que le llega.

Aunque TFTP contiene poco más que lo mínimo necesario para la transmisión, soporta varios tipos de formato de ficheros. Es muy empleado para que sistemas sin disco puedan obtener los ficheros de arranque del sistema operativo a través de la red (UNIX).

4.- Sistema de Nombres de Dominio: DNS.

DNS (Domain Name System), RFC 1034 y 1035 se basa en un esquema de nombres jerárquico y una base de datos distribuida para implementar dicho esquema de nombres. Se usa para relacionar los nombres de host y destino de correos electrónicos con direcciones IP. Los humanos preferimos nombres a direcciones IP (ej: cacharro.escet.urjc.es frente a 212.128.1.44)

Los números IP están ligados a la estructura de la red, pero eso no tiene porqué reflejarse en el nombrado de máquinas (ej: www.debian.org, www.de.bian.org). Los números IP están ligados a máquinas concretas, puede ser conveniente un nivel de abstracción no ligado a máquinas (ej: www.urjc.es puede cambiar de máquina, y de IP, pero no de nombre).

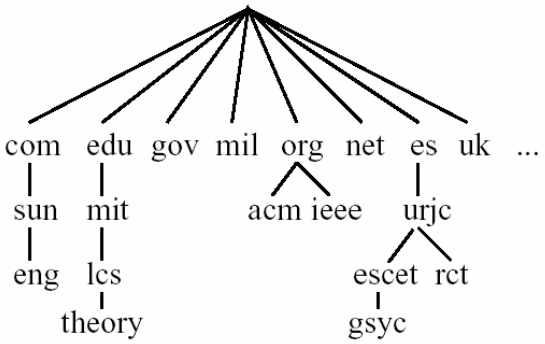
Es necesario establecer una correspondencia entre nombres y direcciones IP. Para relacionar un nombre con una dirección IP, un programa de aplicación llama a un procedimiento de biblioteca llamado resolver, pasándole el nombre como parámetro. El resolver envía un paquete UDP a un servidor DNS local que busca el nombre y devuelve la dirección IP al resolver quien lo devuelve al solicitante. Con la dirección IP, el programa puede realizar una conexión TCP con el destino o enviarle paquetes UDP.

4.1.- Estructura de nombrado.

Se descentraliza el control consiguiéndose una estructura jerárquica y fácilmente ampliable. De esta forma se consigue una jerarquía de dominios:

- Dominio raíz (root o "."). Gestionado por ICANN (Internet Corporation for Assigned Names and Numbers).
- Dominios de nivel máximo:
  - Tradicionales:
    - .com
    - .edu
    - .gov
    - .mil
    - .net
    - .org
    - códigos ISO de paises (uk, ar, es, . . .)
  - En negociación por ICANN en noviembre de 2000: biz, info, name, pro, aero, coop, museum
- Dominios secundarios, terciarios, . . .

La siguiente figura muestra un árbol de dominios:



El acceso al dominio puede ser de dos tipos:

- Dominio directo: proporciona para cada nombre una dirección IP.
- Dominio inverso: proporciona para cada dirección IP un nombre.

El dominio inverso también se conoce como dominio in-addr.arpa. Los elementos del dominio inverso son las direcciones de red construidas invirtiendo los números que la componen, y terminando en in-addr.arpa Ejemplo: La red 138.117.0.0 es el dominio inverso 117.138.inaddr. arpa
Los servidores WWW lo emplean para obtener sus estadísticas de accesos a sus páginas. Así pueden saber de qué nacionalidad son sus clientes.

4.2.- Estructura de gestion.

Cada vez que se delega un subdominio se delega también su gestión (incluyendo su posible subdivisión sucesiva).

- Si el gestor del dominio **es** delega un subdominio **urjc**, el responsable de **urjc** manejaría la correspondencia de nombres y direcciones de todas las máquinas de su dominio.

- Si el responsable de **urjc** lo cree conveniente, puede delegar un subdominio **escet**, sin que por ello tenga que informar al gestor de **es**.

Hay dominios (ejemplo: com, org) gestionados por varios “registrars" en régimen de competencia.

Cada aplicación va enlazada con una biblioteca de consulta al DNS (*resolver*), con llamadas como gethostbyname(). La consulta puede hacerse:

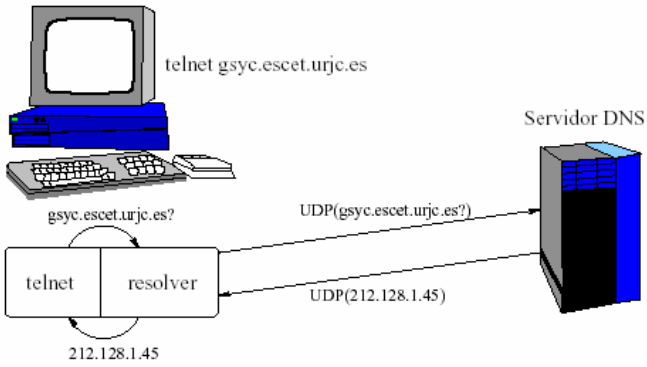
- Mirando en un fichero (/etc/hosts).
- Consultando a un servidor local (normalmente sólo caché).
- Consultando a un servidor en otra máquina (lista previamente especificada).

4.3.- Esquema de funcionamiento del DNS.

Se trata de mantener la información como una base de datos distribuida. Las consultas al DNS se realizan en modo cliente-servidor:

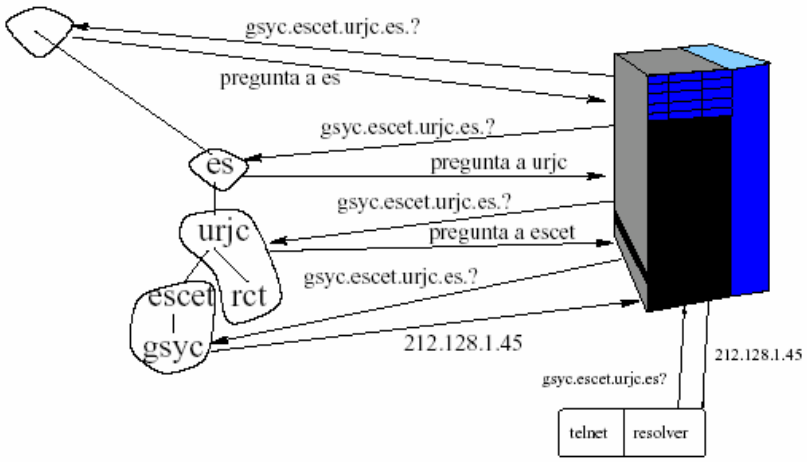
- Cuando una aplicación (cliente) quiere resolver un nombre, pregunta a un servidor de DNS.
- El servidor investiga por su cuenta y devuelve la dirección IP pedida.

Hay que configurar cada máquina con la lista de servidores de DNS que puede usar para resolver nombres, usualmente dos o tres (a veces se hace automáticamente). A un servidor siempre hay que decirle qué máquinas sirven el dominio raíz (“."). La siguiente figura muestra como se realizaría la consulta a un servidor DNS.



Cuando un servidor recibe una consulta para resolver un nombre (ejemplo: nic.funet.fi):

1. Comprueba si el nombre pertenece a alguno de los dominios que sirve (si es que sirve alguno). Si lo encuentra busca en su “mapa" y devuelve la dirección IP correspondiente.
2. En caso contrario pregunta a un servidor del dominio raíz, que le contestaría con la dirección IP de un servidor del dominio “fi".
3. Luego pregunta a ese, obteniendo la dirección IP de un servidor de “funet.fi".
4. Ahora se pregunta a este último, que ya tiene en sus mapas la dirección IP pedida.
5. (Cada servidor puede servir uno o varios dominios, o ninguno)



Los servidores de DNS responden dos tipos de consultas:

- Recursivas: Las que le hace un cliente, que obligan al servidor a hacer las consultas necesarias para encontrar la dirección pedida.

- Iterativas: Las que le hace otro servidor, a las que responden con la dirección IP del servidor del siguiente dominio en la jerarquía.

Siempre que pueden, los servidores usan datos de su cache o tabla de accesos anteriores.

4.4.- Tipos de servidores.

Según como son utilizados:

- **Reenviador** ("forwarder"). Los servidores lo usan antes de consultar al resto del DNS. Utilizados por servidores para centralizar las consultas. Evitan, por ejemplo, el acceso directo a servidores raíz.
- **Esclavo**. Utilizados por servidores en lugar del resto del DNS (por ejemplo, si hay cortafuegos).

Según como reciben los datos:

- **Primario**. Tiene la información actualizada.
- **Secundario**. Copia del primario.
- **Cache**. Guardan datos sobre los que han consultado.

Según el lugar de procedencia del dato:

- **Con autoridad** (*authoritative*). Tiene el mapa "original" para el dominio consultado (primario o secundario).
- **Sin autoridad**. Tiene el dato en su cache.

La información en DNS se almacena en *Registros de Recursos*, registros normalizados según la RFC 1033. Cada mapa de dominio incluye un conjunto de registros de recurso (RR):

- Son la unidad de consulta.
- Cada registro de recurso tiene 5 campos:
  [nombre] [ttl] IN tipo datos
  - Nombre de dominio.
  - Tiempo de vida. Tiempo de validez del registro en las caches.
  - Clase. En Internet siempre IN.
  - Tipo. Define el tipo del registro.
  - Valor. Contenido que depende del campo tipo.

Los tipos de registro de recursos pueden ser:

- **SOA**: Da información de gestión una zona del dominio (servidor de nombres primario, administrador, etc.).
- **NS**: Identifica al servidor de nombres.
- **A**: Define una dirección IP de la estación (puede tener varias).
- **MX**: Define el servidor de correo del dominio.
- **CNAME**: Permite asociar un alias a un nombre de dominio.
- **HINFO**: Da información del tipo de máquina y sistema operativo.
- **TXT**: Da información del dominio.

4.5.- Mapas de dominio.

BIND (*Berkeley Internet Name Domain*) es la implementación más extendida en Unix. Puede encontrarse en www.isc.org, siendo la versión actual la 8.1.2 (Berkeley hasta la 4.8.3). El cliente se llama resolver, y es una librería que se enlaza en el sistema operativo y captura las llamadas a gethostbyname() traduciendolas a consultas DNS en lugar de a búsquedas en el fichero hosts.

El servidor *named* tiene varios ficheros de configuración, donde se especifica de qué dominios es primario o secundario y en qué ficheros debe organizar la información. Los ficheros que contienen la configuración tienen un formato similar, en forma de lista de Registros de Recursos escritos como tablas editables (internamente los codifica en binario por eficiencia)

/etc/named.boot: Parámetros generales del servidor: nombres del resto de ficheros (En nuevas versiones puede llamarse named.conf, y su formato interno puede haber cambiado).

Y colgando de /etc/named (Salvo que se haya especificado otro nombre) :

named.db	Los datos del dominio (nombre genérico de una db)
named.ca	Cache. Inicialmente direcciones de servidores del dominio raíz.
named.hosts	Fichero de zona con las direcciones IP de nombres de host.
named.local	Resolución local de la dirección loopback.
named.rev	Fichero de zona del dominio inverso.

En definitiva, hay un fichero por cada dominio que se sirve (ya sea primario, creado en esta máquina con un editor, ya sea secundario, y DNS se lo trae al arrancar).

A continuación se lista un ejemplo de fichero de definición de dominio:

```
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
```

4.6.- Formato de mensaje DNS.

El mensaje DNS está formado por paquetes iguales para consultas y respuestas con una cabecera fija de 12 bytes:

- Identificación (2 bytes). Correspondencia consultas-respuestas.
- Banderas (2 bytes). Consulta/respuesta, truncamiento, tipo de petición (directa, inversa, estado), respuesta con autoridad, petición recursiva, error.
- Número de consultas (2 bytes).
- Número de RRs de respuesta (2 bytes).
- Número de RRs de autoridad (2 bytes).
- Número de otros RRs (2 bytes).

A continuación los datos, con tamaño variable:

- Consultas.
- Respuestas.
- Autoridad.
- Más información.

Cada consulta de un mensaje DNS (normalmente sólo una) tiene:

- Nombre de la petición: secuencia de etiquetas (tamaño, 1 byte, cadena, hasta 63 bytes), terminada por la etiqueta \root" (byte de tamaño a 0).
- Tipo de consulta (2 bytes). A, NS, PTR, etc.
- Clase de consulta (2 bytes). Normalmente IN.

Se usa el puerto 53 de TCP y UDP para el servidor de DNS.

- Normalmente el resolver hace consultas usando UDP.
- Normalmente el servidor responde usando el protocolo de la consulta.
- Si la respuesta UDP es de mas de 512 bytes (truncada), el resolver al repite usando TCP.
- Las transferencias de zona de primario a secundario usan TCP.

5.- Correo Electrónico.

Es una de las aplicaciones más comúnmente usadas en la red Internet ya que ofrece una forma sencilla de transmitir información. El correo electrónico es distinto a las otras aplicaciones de la red, pues no es necesario esperar a que la máquina remota reciba el mensaje para continuar trabajando. Cada vez que se quiere enviar un mensaje, el sistema crea una copia del mismo junto con la identificación del destino y se realiza una transferencia en modo "background". Posteriormente, el proceso de envío de paquetes tratará de entregar el mismo contactando con el servidor de mensajes en la máquina destino.

Las ventajas de usar el correo electrónico en la red Internet es que ésta proporciona un servicio universal y además fiable, ya que, al usar una comunicación extremo a extremo, se garantiza que el mensaje permanece en la máquina fuente hasta que ha sido copiado satisfactoriamente en la destino.

El estándar que se utiliza para el envío de mensajes es el SMTP (Simple Mail Transfer Protocol). Este protocolo se centra en cómo el sistema de distribución de mensajes pasa los datos a través de una unión de una máquina con la otra. Inicialmente, el cliente establece una conexión TCP con el servidor y se intercambian una serie de comandos para establecer la unión. Una vez la conexión está establecida, el cliente puede enviar uno o más mensajes, terminar la conexión o pedir al servidor intercambiar los papeles de emisor y receptor para que los mensajes puedan fluir en la dirección contraria. El receptor debe reconocer cada mensaje y puede abortar la conexión entera o la transferencia del mensaje actual.

Las principales RFCs relacionadas son las siguientes:

- RFC821 protocolo SMTP.
- RFC822 formato de mensajes e interpretación de cabeceras.
- RFC974 gestion de clausulas MX del DNS.
- RFC1425 protocolo ESMTP (Extended SMTP).
- RFC1341 MIME: « Multipurpose Internet Mail Extensions ».

Todo mensaje consta de 2 partes: **Sobre** y **Contenido**. El *sobre* consta de unas cabeceras estandarizadas para identificar el/los destinatarios/remitentes del mensaje que permiten que el mensaje llegue. El *contenido* está formado por el mensaje propiamente dicho. La RFC822 sólo admite texto NVT-ASCII (Network Virtual Terminal-ASCII). Para otros tipos se emplean codificadores de binarios (uuencode) y extensiones al mensaje (MIME, RFC1341).

La cabecera proporciona información al gestor de correo (MTA, *Mail Transfer Agent*) qué debe hacer con el correo. Dispone de los siguientes campos:

- From: Dirección de origen.
- To: Dirección/direcciones de destino.
- CC: Con copia a. . .
- BCC: Con copia ciega a. . .
- Subject: Tema sobre el que versa la carta.

- Date: Fecha en que fue enviada.
- Received: Por dónde ha pasado la carta.
- Message-id: Identificador del mensaje.

Las direcciones de correo ofrecen la información necesaria para enviar/recibir un mensaje. Indican a donde va y de donde viene, pero no por dónde ha de pasar. Se sigue el formato Internet:

id_usuario@maquina.dominio O bien,: id_usuario@dominio

Existen pasarelas con otros sistemas de correo, aunque el correo de Internet es el mas extendido en el planeta, existiendo otros, estándares: X.400, cc:mail; y muchos mas estándares propietarios. Para enviar mensajes desde uno a otro se emplean pasarelas conectadas a ambos traduzcan los formatos.

La RFC822 sólo admite texto NVT-ASCII. Para el envío de información no textual se ha recurrido a codificaciones de binarios en ASCII (7 bits). El método tradicional en UNIX es el empleo de los programas uuencode/uudecode y otros algo más elaborados: btoa (tarmail), BinHex (Macintosh) y MIME.

Las RFC 2045 - 2049 extienden el contenido de los mensajes de correo para que puedan adjuntarse datos genéricos (attachments). Multimedia Internet Mail Extensions define 5 cabeceras :

- MIME-version:
- Content-Description:
- Content-Id:
- Content-Transfer-Encoding
- Content-Type:

Hay 8 tipos: Text, Image, Audio, Video, Application, Message, Model y Multipart.

- varios subtipos: Text: html, plain o richtext; Image: gif, jpeg; . . .

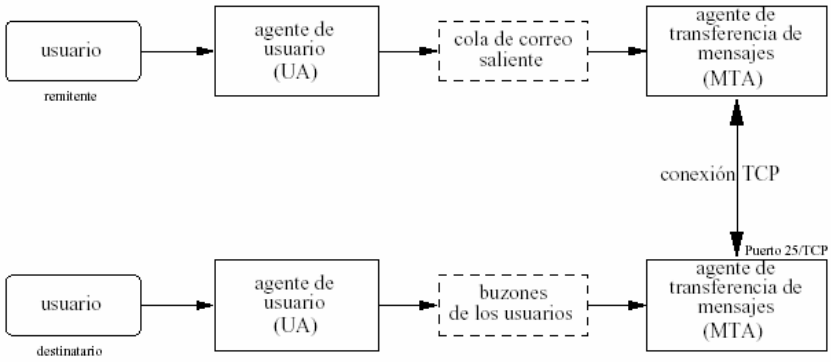
El *Content-Type* se utiliza para saber como tratar en el destino los datos (con qué aplicación). El *Content-Transfer-Encoding* se utiliza para saber como codificar para transmitir los datos por la red y descodificar los datos transmitidos.

- base 64 (ASCII armor) para binarios:
- grupos de 24 bits se rompen en 4x6 bits.
- Cada carácter de 6 bits se cuantifica en binario y se manda en ASCII
- (el 0 va como una A, el 1 como B. . .). Se rellena con = al final.
- quoted-printable-encoding para textos con acentos: codifica los caracteres por encima del 127 con un caracter “=" y dos dígitos hexadecimales.
- 7bit, 8bit, binary. . .

5.1.- El protocolo SMTP.

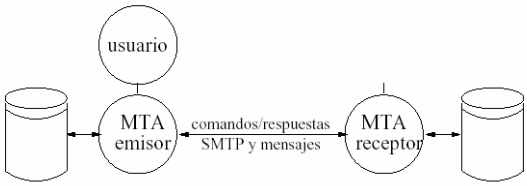
El protocolo SMTP (*Simple Mail Transfer Protocol*) define los comandos que se emplean para comunicarse con/desde un MTA. Se realiza mediante la conexión al puerto tcp/25 y el envío de comandos. Los mensajes estaban limitados a 64K y había problemas de timeouts, por lo que la RFC 1425 define otras técnicas ESMTP (EHLO...) En una comunicación tiene lugar la siguiente secuencia de eventos:

1. El programa empleado para la composición de mensajes pasa el mensaje al Agente de Usuario (UA).
2. Los mensajes se almacenan temporalmente en una cola (/var/spool/mqueue) en los sistemas UNIX.
3. La cola se procesa periódicamente, moviendo el mensaje desde el MTA emisor al MTA destinatario final (puede haber algún MTA Relay).
4. Al llegar al MTA destino (sistema UNIX destino del mensaje) el mensaje queda almacenado en el buzón de un usuario final (/var/mail/usuario) o se entrega directamente al UA para que lo procese automáticamente (procmail, vacation, ftpmail, impresora, fax...).




```
(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
```

El modelo de comunicación SMTP se recoge en la RFC 821. La siguiente figura muestra otro esquema de comunicación.



Los comandos tienen 4 caracteres y sintaxis fija. No se distinguen mayúsculas y minúsculas (excepto en nombres de buzones). La transferencia se realiza con codificación ASCII de 7 bits. Líneas finalizan con CR-LF.

5.1.1.- Procedimiento de envío.

El procedimiento de conexión es el siguiente:

- Al establecerse la **conexión TCP** (puerto 25), el receptor debe saludar con una respuesta 220 Ready.
- **HELO dominio** : El emisor se identifica. Se acepta con una respuesta 250.
- **QUIT** : El emisor no tiene nada mas que enviar. Se acepta con una respuesta 221.

El procedimiento de envio de mensajes es el siguiente:

1. *MAIL FROM:<camino de vuelta>* : Comienza una nueva transacción de envío de un mensaje. El camino de vuelta sirve para notificar errores. Es una lista de máquinas y el buzón origen. Se acepta con una respuesta 250 OK.

2. *RCPT TO:<camino de ida>* : Da un camino que identifica uno de los receptores del mensaje. Puede haber varios comandos RCPT. Se aceptan con una respuesta 250 OK. Si el <camino de ida> es incorrecto, hay algunas opciones:
   251 User not local; will forward to <camino de ida>
   551 User not local; please try <camino de ida>

3. *DATA* : Detrás va el contenido del mensaje, con todas las cabeceras. Se acepta con respuesta de aceptación temporal 354, el contenido completo se acepta con respuesta 250 OK. El final del contenido se marca con un punto solo en una línea. Para lograr transparencia en el contenido se duplican los puntos a principio de línea.

4. Al aceptar un mensaje, el MTA receptor inserta una línea (time stamp) al principio del contenido del mensaje. El MTA final inserta una línea de “camino de vuelta" al principio del contenido del mensaje.

5. El camino de ida puede ser una ruta de la forma: @UNO,@DOS:PEPE@TRES y define como llegar a la máquina donde está el buzón de entrega del mensaje. La primera máquina debe ser la receptora. Cuando el mensaje llega a una MTA, esta pone su identificador como el primero del camino de vuelta. Estos caminos forman parte del sobre, no aparecerán en el contenido. Un MTA que no puede

entregar un mensaje genera un mensaje de error. Para evitar bucles, el camino de vuelta (en MAIL) se puede dar vacío.

5.1.2.- Contenido del mensaje.

El contenido del mensaje (RFC 822) está formado por cabeceras, una línea en blanco y el cuerpo. Hay un conjunto de cabeceras predefinidas. Ejemplos:

- From: Dice quien escribió el mensaje.
- Sender: Dice quien envio realmente el mensaje.
- Reply-To: Dice a quien se debe responder.

La sintaxis es: "Nombre: valor" . Si el valor debe ocupar varias líneas basta con empezar con un espacio a partir de la segunda. Las definidas por el usuario deben empezar con X-.

5.2.- Protocolo POP.

Permite traer los mensajes de un buzón en una máquina remota. Usa TCP, puerto 110. Funcionamiento parecido a SMTP. Comandos varios, pero sólo dos respuestas (+OK, -ERR). Al conectarse se entra en modo autorización. El servidor acepta con +OK. En modo autorización se puede usar comandos:

- *USER nombre*. Identifica al usuario.
- *PASS contra*. Da la contraseña.

En modo transacción. Se pueden emitir comandos:

- *STAT*. Devuelve número de mensajes y tamaño total.
- *LIST*. Devuelve el número de mensaje y su tamaño. Puede llevar argumento.
- *RETR msj*. Devuelve el mensaje. Acaba con una línea con un punto.
- *DELE msj*. Se etiqueta el mensaje para borrar.
- *NOOP*. No hace nada.
- *RSET*. Reset de la conexión. Se quitan las etiquetas de borrado.
- *QUIT*. Para salir. Pasa a modo actualización y borra los mensajes marcados.

Hay varios comandos más opcionales. El protocolo funciona en modo cliente/servidor siguiendo una operativa que permite recuperar los mensajes del buzón (y borrarlos o dejarlos) para que el UA pueda presentarlos en pantalla. Hay varios protocolos para realizar esto:

- POP2, POP3: Post O_ce Protocol. (RFC-1725,1734,1082)
- IMAP: Internet Message Access Protocol (RFC-1203; 1730 al 1733)

5.3.- Protocolo IMAP.

El protocolo IMAP (Internet Message Access Protocol), permite manejar buzones en una máquina remota. Usa TCP, puerto 143. El esquema de comandos y respuestas es similar a SMTP y POP, integrando mecanismos de seguridad. También incluye comandos para crear, borrar y manipular buzones.