Prácticas de laboratorio de Telemática II

Práctica 2

Departamento de Ingeniería Telemática

(ENTEL)

Mónica Aguilar
Juanjo Alins
Oscar Esparza
Jose L. Muñóz
Marcos Postigo
Antoni X. Valverde

La composición de este manual de prácticas ha sido realizada con el programa LaTeX 2ε. Agradecer a DONALD KNUTH la idea y el programa TeX y a LESLIE LAMPORT sus magníficas macros que han derivado en LaTeX.

Barcelona a 5 de Febrero de 2003

• Revision 1.4 comunix.lyx

Edición electrónica de este documento. Este manual de prácticas se puede conseguir en http://labtel2.upc.es

Índice de las prácticas

Índice de figuras

PRÁCTICA 2

Comunicación entre sistemas UNIX

2.1. Introducción

La familia TCP/IP es un conjunto de protocolos desarrollados para permitir la cooperación entre ordenadores con la finalidad de compartir recursos. Este nombre representa a dos de los protocolos más utilizados dentro de la familia, el IP (*Internet protocol*) y el TCP (*Transmission Control Protocol*). Todo el conjunto de protocolos se desarrolla en *Internet*, que no es más que la interconexión de multitud de redes, entre ellas la ARPANET, NSFnet, etc., donde participan casi la totalidad de universidades, centros de investigación y algunas empresas de todo el mundo.

Al final de los años sesenta había una gran necesidad de interconectar los centros de investigación y las universidades norteamericanas, para compartir los recursos informáticos e intercambiar información mediante una red de conmutación de paquetes. En 1968 la Universidad de California Los Ángeles (UCLA) juntamente con tres universidades más, crearon el embrión de la ARPANET interconectando los cuatro centros. En 1972 la agencia gubernamental ARPA (*Advanced Research Projects Agency*) fundó la red ARPANET potenciando proyectos militares. Inicialmente los nodos de esta red estaban conectados con líneas punto apunto empleando el protocolo NCP (*Network Control Protocol*). Más tarde en 1974 apareció la familia TCP/IP bajo la propuesta de CERF y KAHN, de hacer una red independiente de la tecnología del medio de comunicación y de la arquitectura del ordenador, con conectividad universal y con protocolos de aplicación estandarizados.

En el año 1983 se produjo una transición definitiva del NCP al TCP al estandarizarse el TCP/IP en ARPANET gracias a las contribuciones de diversos investigadores mediante los RFC (*Request for Comments*), e integrando la familia dentro del sistema operativo UNIX. El pionero de esta integración fue la Universidad de Berkeley, que, financiada por DARPA (*Defense Advanced Research Projects Agency*), hizo el código fuente de TCP para UNIX 4.2 BSD (*Berkeley System Distribution*) de dominio público. Otro acontecimiento importante fue el apoyo de las compañías operadoras al UNIX, en particular AT&T, al sistema V en 1983.

Cuando finalmente ARPANET creció y se convirtió en *Internet* (integrándose ella misma a *Internet* en 1990) el uso de TCP/IP se propagó incluso a las redes ajenas a ella. Hoy, muchas compañías construyen redes TCP/IP e *Internet* ha crecido tanto que se la puede considerar como la corriente principal de consumo tecnológico.

2.2. La pila de protocolos y su interrelación.

La arquitectura TCP/IP consta de cuatro niveles. Un primer nivel de acceso a la red, que englobaría el nivel físico, el de enlace de datos y parte del nivel de red del modelo de referencia OSI[1] de la ISO[2]. Un segundo nivel de inter-red ofrece un servicio de transmisión de la información entre dos puntos remotos de la red, no orientado a conexión y sin fiabilidad. Este mecanismo es controlado por el protocolo IP, ofreciendo unos servicios similares al nivel de red de la OSI. La unidad básica de información que maneja este nivel se denomina datagrama, según el argot de Internet. El protocolo IP no es fiable porque no asegura la entrega de datagramas. No está orientado a conexión porque el protocolo

[1] OSI es el acrónimo de *Open System Interconnection* (Interconexión de sistemas abiertos)
[2] ISO es el acrónimo de *International Organization of Standardization* (Organismo Internacional de Estandarización)

IP no mantiene la situación de los datagramas sucesivos enviados, de forma que los datagramas pueden llegar al destino duplicados, en orden incorrecto, etc.

Un tercer nivel de transporte suministra un transporte de información entre procesos funcionando en estaciones remotas, con o sin fiabilidad. En este nivel la unidad de información del nivel de transporte es el paquete. En el nivel de transporte, y ofreciendo sus servicios directamente a las aplicaciones, tenemos el protocolo TCP (*Transmission Control Protocol*) que da un servicio orientado a conexión para el transporte fiable de datos extremo a extremo, es decir, capaz de asegurar la entrega de información sin errores. Esto se consigue haciendo un control de errores y pidiendo al receptor las retransmisiones que sean necesarias al emisor. Se dice que es orientado a conexión, puesto que los dos procesos involucrados en la comunicación establecen una conexión antes de iniciar la comunicación, y se hace una reordenación de los paquetes recibidos. Por otro lado, en este nivel también está el protocolo UDP (*User Datagram Protocol*) que da un servicio no orientado a conexión, muy similar al que ofrece IP. En este sentido, y debido a que no hace retransmisiones, no verifica la entrega ni la corrección de datos. UDP permite el envío de información de forma más eficiente y rápida. Así se puede decir que TCP es más conveniente para la transferencia de ficheros, el acceso vía terminal remoto o la descarga de páginas web, mientras que UDP es más adecuado para servicios en tiempo real y que toleran algunos errores o pérdidas, como el envío de audio o vídeo (telefonía, videoconferencia, etc.).

Finalmente el cuarto nivel o nivel de aplicación, nos da la posibilidad de abrir y controlar una sesión con un nodo remoto para transferir información formateada (servicio de transferencia de ficheros), para establecer un diálogo interactivo remoto (servicio de terminal virtual), o para enviar mensajes textuales electrónicos en diferido (servicio de correo electrónico). Este nivel de la arquitectura TCP/IP engloba aproximadamente los servicios de sesión, presentación y aplicación de la OSI.

En el entorno *Internet* se habla de que la red está formada por redes o subredes, que conectan localmente los *hosts* (sistemas informáticos anfitriones de aplicaciones y usuarios), interconectadas a la vez por *gateways* (pasarelas). En la arquitectura TCP/IP habrá entonces otros protocolos para el diálogo *host-gateway* y *gateway-gateway* con objeto de gestionar el encaminamiento, controlar los flujos de datos, notificar errores, etc. Algunos de estos protocolos, digamos "auxiliares", son el ICMP (*Internet Control Message Protocol*), el RIP (*Routing Information Protocol*), el ARP (*Address Resolution Protocol*), el RARP (*Reverse Address Routing Protocol*), OSPF (*Open Shortest Path First*), etc.

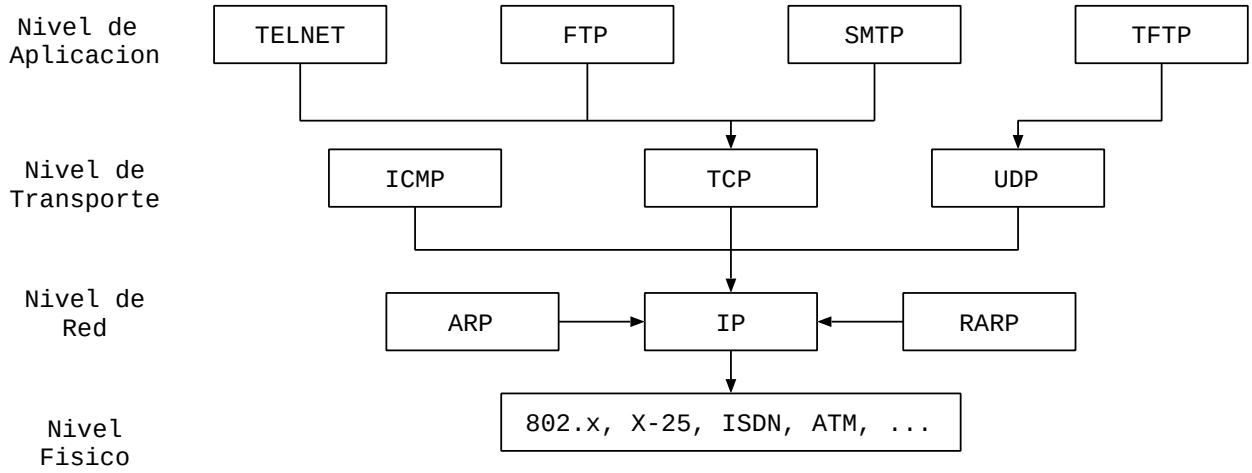

Figura 2.1: Clasificación de los protocolos más significativos de la arquitectura TCP/IP.

El hecho de que TCP/IP sea una arquitectura universal hace que los datagramas puedan viajar por cualquier medio físico, topología, o protocolo de enlace, (*Ethernet*, *Token Ring*, FDDI, X-25, ATM, etc.). En nuestro caso las estaciones de trabajo están interconectadas mediante una topología en estrella empleando el protocolo *ethernet*.

Se observa que dentro del campo de datos de la trama *ethernet*, está el datagrama, mientras que dentro del campo de datos del datagrama IP está el paquete TCP.

La comunicación entre dos nodos de nuestra red se hace mediante datagramas que son transportados dentro del campo de datos de la trama *ethernet*.

Una trama enviada por un *host* es detectada por todas las demás estaciones conectadas, pero sólo el nodo destinatario la recoge y la procesa. Si dos *hosts* intentan emitir a la vez, se produce lo que se denomina una colisión. En esta situación los dos *hosts* abortan la transmisión y realizan un reintento al cabo de un intervalo aleatorio de tiempo. Dentro de una red *ethernet*, las colisiones son un fenómeno natural. En un sistema con actividad elevada, niveles de colisión que ocupen el 30 % del ancho de banda de la red pueden ser habituales.

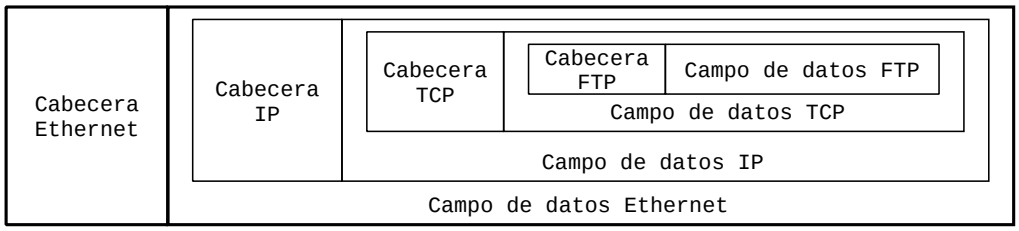

Figura 2.2: Encapsulamiento del servicio FTP empleando una trama ethernet.

2.2.1. La capa de inter-red: el protocolo IP

El objetivo de el protocolo IP es convertir redes físicamente (como pueden ser *Ethernet*, *Token Ring*, X.25, *Frame Relay*, ATM...) en una red aparentemente homogénea, lo que se conoce como interconexión de redes. A la red resultante se la puede denominar *internet* (observar la diferencia con *la Internet*), dónde podemos destacar que:

- Hay un esquema de identificación (o direccionamiento) de todos los sistemas, uniforme y universal. Este esquema de direccionamiento tiene que ser independiente del *hardware*. Esto se consigue asignando a cada nodo un número único de 32 bits (normalmente, puesto que suelen ser IPv4, *Internet Protocol version 4*) denominado dirección IP.
- Las comunicaciones entre usuarios siguen un método uniforme denominado encaminamiento de datagramas, independiente de la red en particular dónde residen.

El protocolo IP es un protocolo de red no orientado a conexión que proporciona un servicio de entrega de datagramas no fiable.

2.2.1.1. Direcciones IP

Tal y como se ha comentado, el protocolo de red IP utiliza direcciones formadas por números de 32 bits, siendo necesario asignar un número único a cada máquina de la red.

Para facilitar la lectura, las direcciones IP se separan en cuatro números de ocho bits denominados octetos, y se representan en formato decimal separados por un punto. Este formateo se denomina notación decimal de puntos (*dotted-decimal* en inglés), y así una dirección IP 0x9353280D se escribiría como 147.83.40.13. Para facilitar la identificación de direcciones, se utilizan nombres. Existen servidores DNS (*Domain Name System*) para resolver la equivalencia entre nombres y direcciones numéricas.

2.2.2. *Domain Name System* (DNS)

Las redes TCP/IP tienen diferentes maneras para traducir nombres de máquinas a direcciones IP, lo que se conoce como resolución de direcciones.

2.2.2.1. El fichero `/etc/hosts`

El mecanismo más simple de resolución de nombres, consiste en almacenar los nombres en el fichero `/etc/hosts`. Aunque se utilicen servidores de nombres DNS externos para resolver direcciones, debemos tener algún tipo de resolución de nombres, para incluso cuando no haya servicios de red ejecutándose, como es el caso del arranque de la máquina. También es interesante en el caso de pequeñas redes de área local que sólo requieran la administración de una persona, y que no tengan tráfico IP con el mundo exterior.

El fichero `/etc/hosts` contiene un registro por línea, consistente en una dirección IP en la primera columna, un nombre de máquina y de forma opcional, una lista de alias para esa máquina. Los campos se separan por tabuladores o espacios. Este es un ejemplo del aspecto del fichero `/etc/hosts`, correspondientes a la máquina `telem2-8` del laboratorio.

```
127.0.0.1       localhost
147.83.40.28    telem2-8.upc.es     telem2-8
192.168.2.28    telem2-8.upc.es
```

Del mismo modo que con las direcciones IP, a veces también puede interesarle usar nombres simbólicos para los direcciones numéricas de red. Con este objeto, el fichero `/etc/hosts` tiene un compañero llamado `/etc/networks`, que asocia nombres de red con los números correspondientes y viceversa. Por ejemplo, el contenido del fichero `/etc/networks` podría definir estas dos subredes que utilizamos en el Laboratorio:

```
Red_LT2         192.168.2.0
Red_Publica     147.83.40.0
```

2.2.2.2. Cómo funciona el DNS

Generalmente no trabajamos con direcciones IP sino con nombres de dominio del estilo de `www.red.net`. Para que esto pueda ser posible es necesario un proceso previo de conversión de nombres de dominio a direcciones IP, ya que el protocolo IP requiere direcciones IP para crear y enviar sus datagramas. Este proceso se conoce como resolución de nombres.

Necesidad del DNS

En los orígenes de Internet, cuando sólo había unos cientos de ordenadores conectados, la tabla con los nombres de dominio y direcciones IP se encontraba almacenada en un único ordenador con el nombre de HOSTS.TXT. El resto de ordenadores debían consultarle a éste cada vez que tenían que resolver un nombre. Este fichero contenía una estructura plana de nombres y funcionaba bien, ya que la lista sólo se actualizaba una o dos veces por semana.

Sin embargo, a medida que se fueron conectando más ordenadores a la red, el fichero HOSTS.TXT comenzó a ser demasiado extenso, el mantenimiento se hizo difícil ya que requería más de una actualización diaria y el tráfico de la red hacia este ordenador llegó a saturarla.

Es por ello que fue necesario diseñar un nuevo sistema de resolución de nombres que distribuyese el trabajo entre distintos servidores. Se ideó un sistema jerárquico de resolución conocido como DNS (*Domain Name System*, sistema de resolución de nombres).

Un servidor DNS mantiene una base de datos de direcciones IP y nombres de dominio y además organiza los nombres de máquina (*hostname*) en una jerarquía de dominios. Un dominio es una colección de nodos relacionados de alguna forma, por ejemplo porque estén en la misma red, tal como los nodos de una universidad.

En figura 2.3 vemos una parte del espacio de nombres. La raíz del árbol, que se identifica con un punto sencillo, es lo que se denomina dominio raíz y es el origen de todos los dominios.

El punto más alto de la jerarquía es el dominio raíz. Los dominios de primer nivel (*es*, *edu*, *com*...) parten del dominio raíz y los dominios de segundo nivel (*upm*, *ucm*, *ibm*...), de un dominio de primer nivel; y así sucesivamente. Cada uno de los dominios puede contener tanto *hosts* como más subdominios.

Un nombre de dominio es una secuencia de nombres separados por el carácter delimitador punto. Por ejemplo, `www.etsetb.upc.es`. Esta máquina pertenece al dominio `etsetb` (Escola Tècnica Superior d'Enginyeria de Tele-

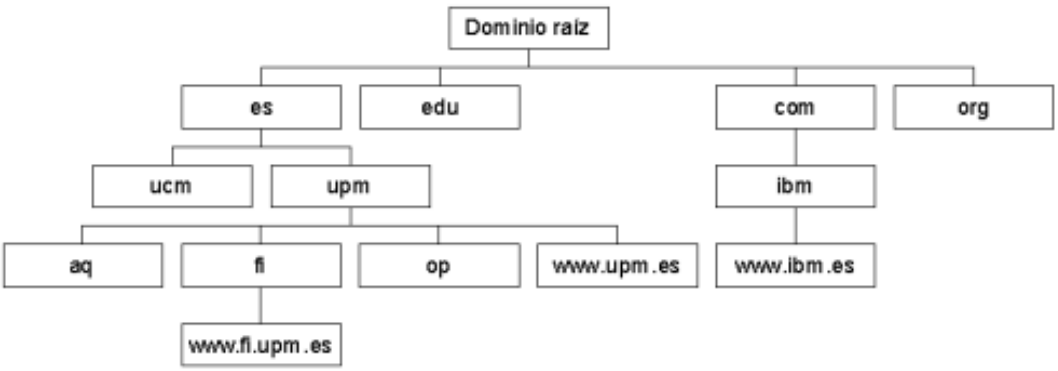

Figura 2.3: Una parte del dominio de la red donde se destaca el dominio de una universidad.

comunicacions de Barcelona) que a su vez pertenece al dominio `upc` (Universitat Politècnica de Catalunya) y éste a su vez, al dominio `es` (España).

Generalmente cada uno de los dominios es gestionado por un servidor distinto; es decir, tendremos un servidor para el dominio `fib.upc.es` (Facultad d'Informàtica de Barcelona), otro para `eupb.upc.es` (Escola Universitària Politècnica de Barcelona) y así sucesivamente.

Los dominios de primer nivel (*Top-Level Domains*) han sido clasificados tanto en función de su estructura organizativa como geográficamente. Los que siguen son algunos dominios de primer nivel que veremos con más frecuencia:

```
Dominio                 Descripción
.edu            Instituciones universitarias.
.com            Organizaciones comerciales.
.org            Organizaciones no comerciales.
.net            Pasarelas y otras redes administrativas.
```

Además, cada país suele tener su propio dominio de primer nivel codificado con las dos letras del país definidas en la tabla ISO-3166. Finlandia, por ejemplo, usa el dominio `fi`, en España se usa el dominio `es`, en México se usa `mx`, en Argentina `ar`, etc. Por debajo de cada dominio de primer nivel, cada país organiza los dominios a su manera.

La organización del espacio de nombres en una jerarquía de nombres de dominio sirve para resolver fácilmente el problema de la unicidad de los nombres y además sirve para que los nombres sean fáciles de recordar. Por ello es conveniente dividir un dominio con gran número de máquinas en subdominios. Además, el sistema DNS permite delegar la autoridad de un subdominio a sus administradores. Así, por ejemplo cada departamento de la UPC puede definir libremente todos los nodos que quiera dentro de su subdominio e incluso crear nuevos subdominios y delegarlos.

Tipos de servidores DNS

El DNS es como una gigantesca base de datos distribuida. Está realizada a través de los llamados servidores de nombres, que proporcionan la información de uno o varios dominios. Para cada zona, debe haber dos o más servidores de nombres capaces de responder por ella. Para obtener la dirección IP de un determinado *host*, todo lo que necesitamos es contactar con el servidor de nombres de la zona y solicitársela. Debemos caer en la cuenta de que cuando configuramos una máquina nueva y nos pide cuál es el servidor de nombres que tiene asociado, debemos introducirle su IP y no su nombre, dado que como aún no está conectado no podrá resolver la IP de dicho servidor de nombres.

Cuando consultamos al servidor de nombres local por una IP de un *host* lejano, la resolución de direcciones generará que el servidor de nombres local lleve a cabo una secuencia de peticiones jerárquica hasta finalmente obtener la dirección deseada. Aparentemente la búsqueda de una dirección IP supone mucho tráfico, sin embargo es minúsculo si lo comparamos con la consulta de un único gigantesco fichero HOSTS.TXT. Aun así hay técnicas para mejorar el rendimiento. Por ejemplo, se dispone de diversos servidores para distribuir la carga y dar cierta tolerancia a fallos. Además, dependiendo de la configuración del servidor, éste puede desempeñar distintos papeles:

- Servidores primarios (*primary name servers*). Estos servidores almacenan la información de su zona en una base de datos local. Son los responsables de mantener la información actualizada y cualquier cambio debe ser notificado a este servidor.

- Servidores secundarios (*secondary name servers*). Son aquellos que obtienen los datos de su zona desde otro servidor que tenga autoridad para esa zona. El proceso de copia de la información se denomina transferencia de zona.

- Servidores maestros (*master name servers*). Los servidores maestros son los que transfieren las zonas a los servidores secundarios. Cuando un servidor secundario arranca busca un servidor maestro y realiza la transferencia de zona. Un servidor maestro para una zona puede ser a la vez un servidor primario o secundario de esa zona. Estos servidores extraen la información desde el servidor primario de la zona. Así se evita que los servidores secundarios sobrecarguen al servidor primario con transferencias de zonas.

- Servidores locales (*caching-only servers*). Los servidores locales no tienen autoridad sobre ningún dominio: se limitan a contactar con otros servidores para resolver las peticiones de los clientes DNS. Estos servidores mantienen una memoria caché con las últimas preguntas contestadas. Cada vez que un cliente DNS le formula una pregunta, primero consulta en su memoria caché. Si encuentra la dirección IP solicitada, se la devuelve al cliente; si no, consulta a otros servidores, apunta la respuesta en su memoria caché y le comunica la respuesta al cliente.

Los servidores secundarios son importantes por varios motivos. En primer lugar, por seguridad debido a que la información se mantiene de forma redundante en varios servidores a la vez. Si un servidor tiene problemas, la información se podrá recuperar desde otro. Y en segundo lugar, por velocidad porque evita la sobrecarga del servidor principal distribuyendo el trabajo entre distintos servidores situados estratégicamente (por zonas geográficas, por ejemplo).

Resolución de nombres de dominio

La resolución de un nombre de dominio es la traducción del nombre a su correspondiente dirección IP. Para este proceso de traducción los *resolvers* pueden formular dos tipos de preguntas: recursivas e iterativas.

- Preguntas recursivas. Si un cliente formula una pregunta recursiva a un servidor DNS, éste debe intentar por todos los medios resolverla aunque para ello tenga que preguntar a otros servidores.

- Preguntas iterativas. Si, en cambio, el cliente formula una pregunta iterativa a un servidor DNS, este servidor devolverá o bien la dirección IP si la conoce o si no, la dirección de otro servidor que sea capaz de resolver el nombre.

Resolución inversa

La operación más habitual con el DNS es obtener la dirección IP correspondiente a un nombre de nodo. Sin embargo, a veces queremos hacer la operación opuesta: encontrar el nombre a partir de la dirección IP. Esto se conoce como resolución inversa, y la usan diversas aplicaciones para comprobación de identidad del cliente. Cuando se utiliza el fichero `hosts`, la resolución se realiza mediante una búsqueda simple en el fichero. Con el DNS, una búsqueda exhaustiva en el espacio de nombres carece de sentido. En su lugar, existe un dominio especial, el `in-addr.arpa`, que contiene las direcciones IP de todos los sistemas en una notación de puntos invertida. La inversión de los bytes es necesaria debido a que los nombres de dominio son más genéricos por la derecha, al contrario que ocurre con las direcciones. Por ejemplo, a la dirección `1.2.3.4` le corresponde el nombre `4.3.2.1.in-addr.arpa`.

Comandos usuales para consultas al DNS

`host [`*`nombre | direcciónIP`*`]`: Resuelve el nombre (o dirección IP) a una dirección IP (o nombre) dada.

`nslookup [`*`nombre | direcciónIP`*`]`: Lo mismo que *host*, pero indicando el servidor DNS utilizado.

`whois `*`direcciónIP`*: Da información sobre la organización propietaria de la dirección IP.

2.2.3. El Protocolo TCP

Como hemos comentado, TCP es un protocolo de transporte orientado a conexión que construye un servicio fiable encima del protocolo IP. La principal ventaja de TCP es que hace servir IP para dar al usuario la sensación de una conexión simple entre los procesos en su equipo y la máquina remota, de forma que no tiene que preocuparse de los datos, ni de cómo viajan, ni de la ruta que utilizan. Es como una tubería de doble sentido entre los dos procesos en la cual ambos pueden leer y escribir.

Se dice que es orientado a conexión porque los dos procesos remotos involucrados en la comunicación deben establecer una conexión antes de empezar la transferencia de información. La información se divide en paquetes en el nodo origen y se reensambla en el nodo receptor, realizando reordenación de paquetes si fuese necesario. Esto se debe a que pueden llegar al destino paquetes por diferentes caminos, con diferente retraso, y por lo tanto, desordenados. TCP tiene la capacidad descartar paquetes que lleguen duplicados.

Se dice que es fiable, ya que realiza control de errores, pidiendo la retransmisión de paquetes si llegan con error o no llegan.

2.2.4. El Protocolo UDP

TCP no siempre es adecuado para todo tipo de aplicaciones. En particular, existen algunas para las que el uso de TCP (conexión, retransmisiones, reordenación de paquetes) puede resultar no adecuado, generalmente porque el sistema requiere la información muy rápidamente.

Igual que TCP, el protocolo UDP permite contactar con un servicio remoto sin establecer conexión (no fiable). La información se envía en paquetes UDP. Como no necesita realizar conexión ni desconexión y al no haber retransmisiones, la sobrecarga de red es menor y la información se recibe con cada paquete UDP. Como contrapartida, UDP no garantiza que la información se recibirá en el extremo remoto por lo que los servicios deben ser tolerantes a pérdidas, o existir mecanismos para garantizar la fiabilidad a nivel de aplicación.

2.2.5. El concepto de puerto

Los *sockets* se pueden ver como "enchufes", como puntos para engancharse las conexiones de red. Si una aplicación quiere ofrecer un servicio, se engancha ella misma a un puerto y espera a los posibles clientes. A esto también se le dice “escuchar el puerto”. Un cliente que quiera hacer servir este servicio, asigna un puerto libre en su nodo local y se conecta al puerto del servidor remoto. El puerto es un número entero de 16 bits e identifica en última instancia el proceso a quién se debe entregar la información recibida.

Una conexión entre dos procesos se identifica por 5 parámetros: la dirección IP y puerto de origen, y la dirección IP y puerto destino, y el protocolo. Si una máquina realiza dos conexiones mediante *ftp* a un mismo servidor de ficheros (*ftp server*), el primer cliente utilizará un puerto local, el 1022 por ejemplo, y el segundo cliente utilizará un puerto local diferente, el 1023 por ejemplo. Ambos procesos se conectarán al mismo puerto destino del *ftp* (21) diferenciándose ambas conexiones por los *sockets* de origen.

Es importante destacar que el mismo número de puerto puede ser utilizado simultáneamente por diferentes protocolos (TCP y UDP). Por ejemplo, es diferente de puerto UDP 513 del TCP 513. Este puerto en concreto ofrece el servicio *rlogin* para TCP y el servicio *rwho* para UDP.

Para los servicios más usados, el IETF (*Internet Engineering Task Force*) publica regularmente el RFC-1700 (*Request For Comment*) que describe entre otras cosas los números de puertos asignados a los servicios ampliamente conocidos. En Unix se puede encontrar la correspondencia entre el nombre del servicio (*ftp*) y el número de puerto (21) , en el fichero */etc/services*.

2.3. Configuración de las comunicaciones

Unix sitúa mayoritariamente los ficheros para configurar las comunicaciones en el directorio `/etc`. Los más relevantes son:

`/etc/hosts` El fichero `/etc/hosts` contiene un registro por línea, consistente en una dirección IP en la primera columna, un nombre de máquina y de forma opcional, una lista de alias para esa máquina. Los campos se separan por tabuladores o espacios.

`/etc/protocols` El fichero `/etc/protocols` describe los distintos protocolos DARPA para *Internet* que están disponibles en el subsistema TCP/IP. Este fichero debería ser consultado en vez de usar los números en los ficheros de encabezamiento ARPA, o, peor aún, adivinarlos. Estos números se incluyen en el campo de protocolo de cualquier encabezamiento IP.
Este fichero no se debe modificar porque los cambios pueden producir paquetes IP incorrectos. Cada línea se compone del nombre del protocolo, número oficial para el protocolo, y de posibles alias para ese protocolo.

`/etc/services` El fichero `/etc/services` relaciona cada servicio de red estandarizado con el puerto y protocolo que utiliza.

`/etc/inetd.conf` El fichero `/etc/inetd.conf` permite especificar los servicios de red que estarán disponibles en el *host*. Así, el demonio `inetd` consulta el fichero cuando arranca y activa todos los servicios especificados. En este fichero se encuentran entre otros parámetros: el nombre del servicio, el tipo de *socket*, el protocolo utilizado, el usuario que ejecuta el demonio y el directorio donde se encuentra el servidor al que se dirige la petición que llega al demonio `inetd`. Las líneas con '#' no se consideran (son comentarios).

2.4. Comandos UNIX para la gestión de las comunicaciones

Dos de los comandos más importantes para la gestión de las comunicaciones en UNIX son `netstat` e `ifconfig`.

`netstat` Se usa para consultar el estado en el que se hallan las comunicaciones en el *host* local. En general, `netstat` proporciona información sobre:

- Las conexiones TCP activas en el *host* local
- El estado de todos los servicios TCP/IP del servidor local y de los puertos que usan
- Dispositivos y enlaces usados por TCP/IP
- Las tablas de encaminamiento IP usadas en el *host* local.

`ifconfig` Se usa para dar acceso a una interfaz. Esto incluye la asignación de una dirección IP y otros parámetros, así como la activación (se envían y reciben datagramas IP) de la interfaz.

2.5. Servicios ARPA (`telnet` y `ftp`).

Los servicios más importantes del TCP/IP son el terminal virtual, la transferencia de ficheros y el correo electrónico.

El servicio de terminal virtual o *remote login* permite a un usuario entrar remotamente a cualquier ordenador conectado a la red *Internet*. La sesión se inicia invocando el comando `telnet` (*network terminal protocol*) proporcionándole el nombre o la dirección del nodo remoto dónde nos queremos conectar. Mientras se mantiene la comunicación, cualquier secuencia de caracteres que tecleemos en nuestro terminal será enviado al nodo remoto, como si estuviéramos directamente conectados con los recursos remotos.

El comando `ftp` (*file transfer protocol*) proporciona un servicio de transferencia de ficheros, que permite que cualquier usuario conectado a la red pueda obtener o enviar ficheros de un nodo remoto. Para acceder al nodo hace falta tener la correspondiente autorización (*password*).

2.5.1. `telnet`

El comando `telnet` es la interfaz de usuario del protocolo TELNET de INTERNET. En los sistemas UNIX, el comando `telnet` se utiliza por el cliente mientras que en el servidor (el ordenador remoto invocado por el `telnet`) hay un *daemon* (proceso en *background*) denominado `telnetd`. Cuando se lanza el `telnet` se abre una conexión TCP y se activa el proceso `telnetd` en el servidor remoto. Este último se encarga de interconectar el pseudoterminal con el protocolo TCP de manera que el terminal local aparezca como terminal remoto.

El protocolo TELNET define lo que se denomina Terminal Virtual de Red (NVT) que delimita las capacidades de la fuente y el destino. La fuente se define como un teclado virtual que maneja un juego específico de caracteres, y el destinatario remoto como una impresora virtual que muestra los caracteres generados por el teclado.

La sintaxis del comando es `telnet [`*`host`*`]`, donde *`host`* es el identificador o dirección del nodo remoto. Una vez el ordenador remoto nos haya permitido entrar en la sesión, después de introducir el nombre de usuario y el *password*, todo carácter tecleado se transmitirá, excepto el carácter de control que generalmente es `ctrl ]`.

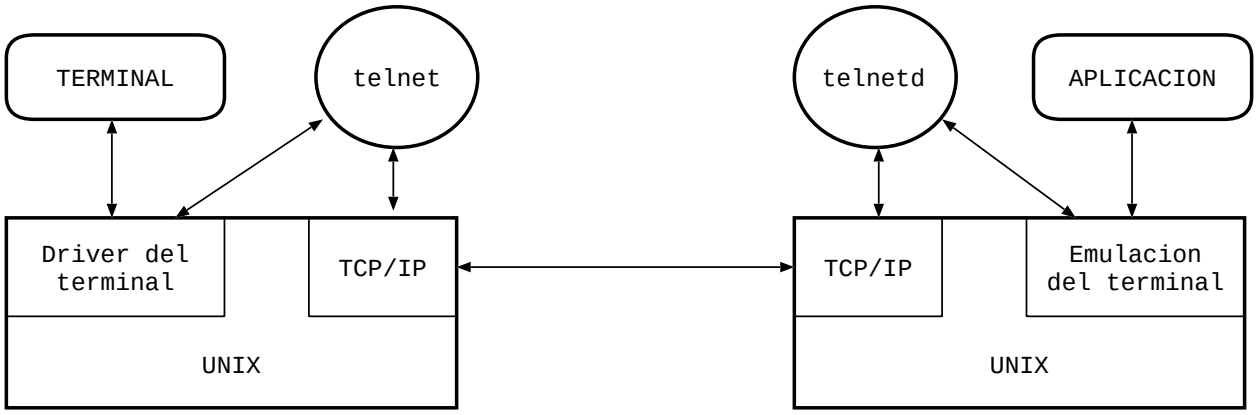

Figura 2.4: Elementos que intervienen en una sesión telnet entre dos nodos remotos.

En caso de no introducir un nombre de *host*, entramos en modo comando. En este modo aparece un *prompt* indicando que estamos en modo comando (`telnet>`). También se puede pasar de modo normal a modo comando pulsando el carácter de escape `ctrl ]` (el carácter de escape se puede cambiar previamente mediante `telnet>set escape`). En modo comando las conexiones pueden ser creadas o canceladas y se pueden modificar los principales parámetros de operación. Cuando tenemos una sesión abierta después de ejecutar un comando el sistema vuelve automáticamente a modo normal.

Los comandos de `telnet` más utilizados son:

`close` cierra la sesión que tenemos abierta.

`open` abre una sesión con un *host* remoto.

`quit` sale del `telnet`.

`mode` indica si la transmisión se hace carácter a carácter (*character*) o se envía una línea de caracteres (*line*) cuando se genere un EOL.

`status` da información del estado de la conexión, el modo, el eco y el carácter de escape.

`set` activa parámetros de operación (*echo*, *escape*, *erase*, *kill*, *eof*, *quit*).

`send` envía caracteres especiales al servidor.

`?` petición de información de ayuda de las acciones de los comandos.

2.5.2. `ftp`

El comando `ftp` es la interfaz del protocolo de ARPANET FTP. Este servicio proporciona la posibilidad de enviar, recibir, borrar o renombrar ficheros, crear, borrar o renombrar directorios, enviar mensajes, o añadir datos a los ficheros.

En los sistemas UNIX el FTP utiliza dos modos de transmisión: modo binario y modo texto.

En modo texto los ficheros son enviados mediante líneas compuestas de caracteres ASCII, separadas mediante retornos de carro y símbolos de nueva línea. En modo binario los ficheros son manipulados como una secuencia de bytes sin ningún tipo de conversión. También es necesario mencionar que utiliza el formato NVT. Los códigos de retorno generados están formados mediante un número de tres cifras, que empieza con 1, 2, 3, 4, ó 5, seguido de un mensaje textual explicando una acción. El mensaje indica un error si el código empieza con 4 ó 5.

Para abrir una sesión FTP se hace con el comando `ftp` pasando a modo comando, indicado por el *prompt* `ftp>`. Para ejecutar una transferencia de ficheros es necesario tener permisos sobre el fichero; para ello previo a la transferencia, el servidor nos pedirá un *username* y un *password*. Una vez establecida la conexión se está en el cliente, no en el servidor como es el caso del TELNET.

Los comandos más usuales son:

`open` abre una sesión.

`close` cierra una sesión.

`quit` sale del `ftp`.

`rmdir` elimina directorio en el servidor

`mkdir` crea directorio en el servidor

`delete` borra ficheros en el servidor

`!` volvemos a la shell y podemos ejecutar un comando de UNIX.

`get (o mget)` captura un fichero o varios del servidor.

`put (o mput)` envía un fichero o varios al servidor.

`type ascii` cambia el modo de transmisión a modo texto.

`type binary` cambia el modo de transmisión a modo binario.

`cd` cambia de directorio en el servidor.

`lcd` cambia de directorio en el cliente.

`dir` muestra el contenido del directorio servidor.

`pwd` lista el camino del directorio en el servidor.

`?` menu de ayuda.

`verbose` presenta los mensajes del servidor.

`hash` muestra el número de carácter de cada bloque del *buffer* transmitido.

`status` da el estado de los principales parámetros del servicio.

2.6. Ejercicios

Ejercicio 1

Configuración de las Comunicaciones UNIX

1. ¿Cuál es el número de protocolo oficial para los protocolos IP y TCP?
2. Visualice el fichero `/etc/services`. ¿Qué puertos utilizan los servicios *ftp* y el *telnet*?
3. ¿Por qué aparece dos veces el servicio *telnet* en el fichero?
4. En Windows 95, ¿qué puerto cree que utiliza el servicio del *telnet*? ¿Y en Windows XP?
5. Visualice el fichero `/etc/inetd.conf` y diga qué servicios están activados en su máquina.

Ejercicio 2

Gestión de las Comunicaciones UNIX

1. Utilizando el comando `ifconfig` identifique cuántos interfaces tiene su máquina.
2. Con el mismo comando detalle para cada interfaz:
   a) Dirección física Hardware (Ethernet)
   b) Dirección IP
3. Describa el interfaz de *loopback* (`lo`). ¿Para qué cree que sirve?
4. ¿A qué interfaz corresponde el nombre de su máquina (`telem2-x`)?
5. Ejecute los comandos `netstat -s | more` y `netstat -a | more` y comente los resultados.
6. Ejecute el comando `netstat -n` y `netstat -nr`. ¿Qué información le proporciona?

Ejercicio 3

Utilización del servicio ARPA/Berkeley: `telnet`, terminal virtual.

1. Desde el entorno de ventanas, abra un terminal e invoque a `telnet`. Abra una sesión remota a `telem2-10.upc.es` con el *login* `alumno` y el *password* `nolose00`.
2. Observe la estructura de directorios, del terminal remoto, abriendo un *File Manager* (mediante el comando `konqueror`) redireccionando la pantalla de la máquina remota hacia la suya local (comandos `xhost` y `export DISPLAY`). Compare la estructura de directorios con la de su máquina local. ¿Al invocar el *telnet* trabajamos sobre el cliente o sobre el servidor?
3. Cierre la sesión y el servicio.

Ejercicio 4

Utilización del servicio ARPA/Berkeley: `ftp`, file transfer protocol.

1. Mediante un `ftp` a `telem2-10.upc.es`, entre como el usuario `alumno`. Transfiera a su máquina local el fichero `hola.txt`. Simultáneamente con el *File Manager* (o mediante `ls -l`) monitorice el directorio de destino y la transferencia del fichero.
2. ¿Se mantienen los permisos originales de los ficheros en una transferencia *ftp*?

Ejercicio 5

DNS

1. Averigüe cuál es el servidor DNS al que su máquina realiza consultas de resolución de direcciones.

2. Edite el contenido del fichero `/etc/resolv.conf`. Se trata de los servidores de nombres que se indica que hay que consultar para resolver direcciones. Compruebe que las direcciones IP coinciden con las que consulta el programa `nslookup`.
3. Averigüe el nombre de las máquinas cuyas IP son `147.83.40.24` y `147.83.163.130`
4. Averigüe las IP de las máquinas cuyos nombres son `wamba.upc.es`, `xaloc.upc.es` y `yogui.upc.es`
5. Averigüe los datos de la organización propietaria de la dirección IP `147.83.39.1`.