Administración básica en red del S.O. Linux

Ingeniería de Tecnologías de Telecomunicación

Departamento de Ingeniería Telemática (DIT)

Universidad de Sevilla

Fco. Javier Muñoz Calle

Francisco José Fernández Jiménez

ÍNDICE

1. Objetivos y alcance (5 minutos)

1.1 Introducción

En una red de área local (LAN) los equipos están configurados de forma que sea posible la comunicación entre los mismos, el acceso a otros equipos que proporcionan servicios y el acceso al exterior de dicha red en el caso de que sea necesario solicitar servicios que no proporcionan los servidores locales.

El conocimiento de los parámetros y valores con los que están configurados los equipos de una red, así como de los mecanismos que permiten la modificación de dichos valores, es fundamental en las tareas de administración y mantenimiento de los ordenadores, equipos de redes, etc.

La monitorización de las condiciones en las que está operando la red local a la que un determinado equipo está conectado, es también de gran utilidad para determinar el correcto funcionamiento de dicha red.

1.2 Objetivo de la práctica

El primer objetivo de esta práctica es descubrir la configuración de la red en la que está operando su equipo. El alumno deberá descubrir la configuración IP de éste (dirección física, dirección IP, máscara de red, servidores, etc.), así como la de otros.

El segundo objetivo de esta práctica es que el alumno se familiarice con el uso de las herramientas de monitorización que existen para redes en las que todos los equipos comparten un medio común. El uso de un medio común, como sucede con las redes Ethernet, puede generar problemas bajo determinadas condiciones, los cuales pueden prevenirse o detectarse con la utilización de las herramientas aquí estudiadas.

Es importante que el alumno aprenda a administrar su equipo desde un entorno en modo texto, ejecutando directamente los comandos que le permitan descubrir cómo están configurados los equipos de la red, siendo esto relevante dada la inexistencia de entornos gráficos en multitud de servidores. Es importante entender el contenido de los archivos de configuración antes de manejar otras herramientas que los modifiquen de forma opaca.

1.3 Documentación de apoyo

- "The Debian system: concepts and techniques". Martin F. Krafft. 2005. ISBN: 1593270690.
- "Linux bible". Christopher Negus. Indianapolis, IN. Wiley, 2011. ISBN: 9780470929988.
- "Pro Linux System Administration". James Turnbull, Peter Lieverdink, Dennis Matotek. Berkeley, CA. Apress, 2009. ISBN: 978-1-4302-1912-5.
- "Linux Network Administrator's Guide (Openbook)". Olaf Kirch, Terry Dawson. O'Reilly, 2000. ISBN: 1-56592-400-2
- Página Web de tcpdump: http://www.tcpdump.org/
- Página Web de Wireshark: http://www.wireshark.org/

2. Configuración de red del sistema Linux (50 minutos)

En este apartado se presentan los ficheros que contienen la configuración más básica de la red del equipo. Estos ficheros son leídos por diversos scripts implicados en el proceso de arranque del equipo. Todos ellos son archivos de administración, por lo que su modificación requiere permisos de superusuario.

2.1 Configuración de las tarjetas de red (25 minutos)

Para ver el nombre y configuración de las distintas interfaces de red existentes en su equipo (activas o no) puede usar el comando "`ifconfig -a`". Al arrancar el equipo, la configuración de las distintas interfaces de red es leída del fichero "`/etc/network/interfaces`". A continuación se presentan, a modo ilustrativo, las líneas de dicho fichero empleadas para configurar las siguientes interfaces (mas información en "man interfaces"):

a) Interfaz local "lo" (utilizada para la comunicación IP interna y asociada a las IPs 127.x.x.x):

`/etc/network/interfaces`

```
auto lo
iface lo inet loopback
```

b) Interfaces correspondientes a tarjetas físicas Ethernet (las tarjetas WiFi emplearán algunos parámetros adicionales): pueden configurarse mediante una de las dos opciones siguientes (en el fichero "interfaces" NO deben aparecer comentarios tras las líneas de configuración):

- Configuración DHCP de la interfaz eth0:

`/etc/network/interfaces`

```
auto eth0                       <-- Activar la interfaz en el arranque
iface eth0 inet dhcp            <-- Configuración por DHCP
```

- Configuración estática de la interfaz eth0 (las direcciones IP indicadas son un ejemplo, no tienen que coincidir con las que debería usar en su equipo):

`/etc/network/interfaces`

```
auto eth0                                <-- Activar la interfaz en arranque

iface eth0 inet static                   <-- Configuración estática
address 172.16.4.50                      <-- IP interfaz eth0
   netmask 255.255.252.0                 <-- Máscara
   network 172.16.4.0                    <-- IP de subred
   broadcast 172.16.7.255                <-- IP de difusión
   gateway 172.16.4.13                   <-- Pasarela/Encaminador
   dns-nameservers 150.214.186.69        <-- DNSs por interfaz (si no,
                                            los de /etc/resolv.conf)
```

Para hacer efectiva la configuración de dichos ficheros se dispone de los siguientes comandos (empleados por el sistema en el proceso de arranque del equipo), que deben ser invocados como superusuario:

a) Desactivar y reactivar todas las interfaces de red con la configuración de indicada en "`/etc/network/interfaces`":

```
service networking restart
```

Este script sólo activará las interfaces que tengan configurado el arranque automático ("`auto interfaz`").

NOTA

En las distribuciones Linux actuales, tras ejecutar el comando "`service networking restart`" se muestra un aviso indicando que esta opción está "deprecated" (desaconsejada) al no reactivar algunos tipos de interfaces de red (lo que no afecta a las interfaces Ethernet). Alternativamente a este comando, para garantizar la reactivación de todas las interfaces de red, puede ejecutarse:

```
service networking stop && service networking start
```

o igualmente:

```
ifdown -a && ifup -a
```

b) Des/activar una interfaz de red concreta (y que, cuando se active, se le aplique la configuración del fichero "`/etc/network/interfaces`"):

Operación	Sintaxis	Ejemplo
Desactivar	`ifdown interfaz`	`ifdown eth0`
	`ifconfig interfaz down`	`ifconfig eth0 down`
Activar	`ifup interfaz`	`ifup eth0`
	`ifconfig interfaz up`	`ifconfig eth0 up`

Entre ambas opciones, debe aclararse que:

- "`ifconfig interfaz down`" desactiva la interfaz, pero no "limpia" procesos asociados (por ejemplo, no elimina de memoria el cliente DHCP "dhclient").
- "`ifdown eth0`" desactiva la interfaz y limpia los procesos asociados.
- "`ifconfig interfaz up`" sólo configura los parámetros explícitos de la red (IP y máscara), no los parámetros adicionales tales como la pasarela.
- "`ifup eth0`" aplica todos los parámetros recogidos en el fichero de configuración "`/etc/network/interfaces`", esto es, los parámetros propios de la tarjeta (IP y máscara) y todos los demás, incluida la pasarela. Si se invoca este comando sobre una interfaz que ya se encuentra activa ("up"), o de la que no se han limpiado sus procesos asociados, entonces no hace nada.

RECUERDE...

En varias operaciones de la práctica será necesario utilizar la dirección IP de otro equipo de la subred en la que se encuentra su máquina. Para obtenerla, ejecute el comando "`ifconfig`" (como root) en otro equipo de la sala que se encuentre disponible; si no hubiese ninguno libre, solicítele a un compañero que le indique la dirección IP de su equipo. A dicha dirección la llamaremos "IP_X". **Recuérdela porque se hará uso de ella** a lo largo de la práctica.

RECUERDE...

El objetivo de esta práctica es la Administración del sistema, tarea de la que se encarga el superusuario "root". Por ello, tenga en cuenta que para la edición de cualquiera de los ficheros de configuración referenciados en esta memoria, deberá usar "root".

TAREAS

1º Visualice el contenido del fichero "`/etc/network/interfaces`" y compruebe cómo su interfaz de red `eth0` se encuentra configurada para ser configurada mediante DHCP. Ejecute el comando "ifconfig" y compruebe cómo tiene una dirección IP, que ha sido asignada por el servidor DHCP.

2º Ejecute el comando "`ping 193.147.162.1`" para comprobar que su tarjeta de red funciona correctamente (recuerde que para detener el comando "ping" debe pulsar "Ctrl-C").

3º Edite el fichero "`/etc/network/interfaces`"[1] y configure su interfaz eth0 de forma estática con los mismos valores de red que actualmente tiene configurados (basta que configure los valores que muestran "ifconfig" y "route", no es necesario que configure los servidores DNS). Aplique los cambios sobre la tarjeta de red, vuelva a ejecutar el comando "ping" anterior y compruebe que efectivamente todo sigue funcionando igual.

4º Vuelva a editar el fichero "`/etc/network/interfaces`" para que la interfaz "eth0" siga configurándose por DHCP, y aplique los cambios.

5º Desde su ordenador, ejecute "ping IP_X" y compruebe que responde.

[1] Si desde una consola como "root" intenta arrancar un editor de texto en modo gráfico y obtiene un mensaje de error referido al servidor "X", pruebe a resetear el servidor gráfico ("Alt-Gr + Impr-Pant + K"). Este problema puede producirse al des/activar o reconfigurar las interfaces de red.

TAREAS

6º Ejecute los siguientes comandos y analice lo que va sucediendo:

```
ifconfig eth0
ping IP_X
ping 193.147.162.1
ifconfig eth0 down
ping IP_X
ping 193.147.162.1
ifconfig eth0 up
ping IP_X
ping 193.147.162.1
route
ifup eth0
ping IP_X
ping 193.147.162.1
route
ifconfig eth0 down
ifup eth0
ping IP_X
ping 193.147.162.1
route
ifconfig eth0 down
pkill -9 dhclient
ifup eth0
ping IP_X
ping 193.147.162.1
route
ifdown eth0
ifup eth0
ping IP_X
ping 193.147.162.1
route
```

La ejecución de estos comandos le debe permitir comprobar la diferencia entre los comandos "ifconfig eth0 up/down" e "ifup up/down".

2.1.1 Construcción inicial de la tabla de encaminamiento

La activación de las interfaces de red (no la local) lleva aparejada la inserción de las primeras líneas en la tabla de encaminamiento. Al activar una interfaz `ethX` se añade en la tabla de encaminamiento

- "1" entrada directa (o de subred): correspondiente al rango de direcciones IP de la subred a la que dicha interfaz permite acceso a nivel 2. Estas entradas son ordenadas de forma automática de menor a mayor tamaño de subred (de la máscara de más a la de menos "1").
- Si la interfaz contiene una entrada "gateway", se establece dicha pasarela como la asociada a la entrada "default" (entrada indirecta). Referente a la entrada por defecto ("`default`") de la tabla de encaminamiento, debe tenerse en cuenta que el script "`networkwing`" sólo activa una entrada por defecto, la del último parámetro "`gateway`" leído en el fichero

"`/etc/network/interfaces`". En caso de que manualmente (con el comando "route") se hayan configurado varias entradas default, al activar una interfaz con el comando "ifup interfaz", sólo se verá afectada la primera entrada default, que será sustituida por la pasarela configurada para dicha interfaz.

A modo de ejemplo, si la configuración de la interfaz "`eth0`" es:

`/etc/network/interfaces`

```
iface eth0 inet static
address 192.168.0.2
   netmask 255.255.255.0
   network 192.168.0.0
   broadcast 192.168.0.255
   gateway 192.168.0.1
```

se añadirán las siguientes entradas a la tabla de encaminamiento:

```
   Destination     Gateway         Genmask         Interface
   default         192.168.0.1     0.0.0.0         eth0
   192.168.0.0     *               255.255.255.0   eth0
```

Sobre esta situación inicial de la tabla de encaminamiento siempre podrán realizarse modificaciones mediante el comando "route".

Advierta cómo la tabla de encaminamiento de Linux no recoge la interfaz virtual "lo". Cuando se intenta enviar un paquete, antes de hacer uso de la tabla de encaminamiento Linux comprueba si el destinatario es el propio equipo, en cuyo caso lo envía por la interfaz local "lo" (no llegando a hacer uso de la tabla de encaminamiento).

TAREAS

1º Visualice la tabla de encaminamiento con el comando "route".

2º Desactive la interfaz eth0 y visualice la tabla de encaminamiento.

3º Vuelva a reactivar la interfaz "eth0" y vuelva a visualizar la tabla de encaminamiento. Compruebe que los valores de dirección IP de subred y máscara corresponden con los antes introducidos manualmente en el fichero de configuración de la interfaz.

2.2 Resolución de nombres (10 minutos)

Para configurar la resolución de nombres en Linux se emplean los siguientes archivos, de los que puede obtener más información en el manual "man nsswitch.conf":

Fichero	Utilidad	Líneas
/etc/hostname	Nombre de la máquina local (pasado al comando "hostname" y a la variable "$HOSTNAME")	nombre Ejemplo: FAST
/etc/hosts	Resolución local de nombres	IP nombre otros_alias Ejemplo: 127.0.0.1 localhost local
/etc/resolv.conf	IP servidores DNS	nameserver IP_servidor_DNS Ejemplo: nameserver 150.214.186.69
/etc/nsswitch.conf	Ordena el orden de consulta de las bases de datos a usar. Para la resolución de nombres ("hosts"), el valor "files" indica /etc/hosts)	hosts: files dns
/etc/host.conf	Prioridad de resolución de nombres ("hosts" indica /etc/hosts y "bind" servidor DNS). Sustituido por "nsswitch.conf" (se mantiene para aplicaciones antiguas)	order hosts,bind

TAREAS

1º Visualice su fichero "/etc/resolv.conf" para conocer cuáles son las direcciones IP de sus servidores DNS.

2º Ejecute el comando "ping trajano.us.es" y observe la dirección IP a la que se resuelve.

3º Edite el fichero "/etc/hosts" y asocie la dirección "127.0.0.1" al nombre "trajano.us.es".

TAREAS

4º Vuelva a ejecutar el comando "ping trajano.us.es" y compruebe cómo ahora se resuelve a la dirección IP local.

5º Modifique el fichero "/etc/nsswitch.conf" para que contenga el orden de resolución de nombres "hosts: dns files", comprobando cómo ahora al hacer "ping trajano.us.es", se vuelve a resolver el nombre “trajano.us.es” a su dirección IP real.

6º Mediante los comandos "hostname" y "echo $HOSTNAME", obtenga el nombre de su máquina local, y compruebe que es el mismo que el configurado en el fichero "/etc/hostname".

7º Ejecute el comando "hostname equipo" y salga del terminal de comandos en el que se encuentra trabajando. Vuelva a abrir un nuevo terminal de comandos y compruebe cómo el prompt del sistema recoge el nuevo nombre dado al equipo (al invocarse el intérprete de comandos, éste lee el nombre actual del equipo para configurar la variable de entorno $HOSTNAME, y la usa para mostrar el prompt). Ejecute "hostname" y compruebe cómo también se obtiene ese nombre.

2.3 Otros ficheros con la configuración básica de la red (10 minutos)

Además de los ficheros comentados, en Linux existen muchos otros ficheros asociados con las red. A continuación se resumen algunos de ellos (use "man" para obtener más información):

Fichero	**Utilidad**	**Líneas**

`/etc/protocols`	Alias IDs IP (recoge IANA)	`protocolo ID otros_alias`
		
		Ejemplo:
		`icmp 1 ICMP`

`/etc/services`	Alias puertos (recoge IANA)	`alias puerto/tcp_o_udp`
		
		Ejemplo:
		`domain 53/tcp`

`/etc/networks`	Nombres subredes (i.e. lo usa route)	`nombre IP_subred`
		
		Ejemplos:
		`loopback 127.0.0.0`
		`default 0.0.0.0`

Fichero	Utilidad	Líneas

`/etc/ethers`	Asignación de IP a MAC (puede usarse para crear entradas manuales en la caché ARP: "`arp -f /etc/ethers`")[2]	`dirección_MAC dirección_IP`
		
		Ejemplo:
		`00:01:02:03:04:05 80.10.1.4`

TAREAS

1º Visualice el contenido de los ficheros "/etc/protocols" y "/etc/services", y observe cómo recogen los valores estandarizados para los números de protocolo y números de puerto.

2º Edite el fichero "/etc/networks" para conseguir que al ejecutar el comando "route", en la línea que da acceso a su propia subred (entrada directa), en lugar de aparecer la IP de subred, aparezca el nombre “mi_subred”.

3º Compruebe cómo el fichero "/etc/ethers" no existe en su equipo. Cree dicho archivo con la entrada "00:01:02:03:04:05 IP" (siendo “IP” alguna dirección IP de su subred) y úselo para añadir una entrada manual a la caché ARP mediante el comando "`arp -f /etc/ethers`". Compruebe con el comando "arp" cómo dicha entrada manual (flag "M") se ha añadido correctamente a la caché ARP.

4º Elimine todas las entradas manuales de la caché ARP (sintaxis “`arp -d IP`”).

2.4 Parámetros de red del kernel (5 minutos)

Linux permite personalizar gran parte del comportamiento en red a través de la configuración del propio kernel. Para ello, el kernel dispone de múltiples parámetros modificables en tiempo de ejecución, accesibles a través de la carpeta virtual "`/proc/sys/`". De todos los parámetros del kernel, los que controlan el funcionamiento de la red se encuentran bajo la ruta "`/proc/sys/net/`". El valor de estos parámetros puede controlarse de dos formas (consulte la ayuda, “`man sysctl.conf`” y “`man sysctl`”, para obtener más información):

a) Configurando su valor para cada arranque del sistema mediante el fichero “`/etc/sysctl.conf`”: para configurar el parámetro ubicado en el fichero virtual "/proc/sys/dir1/dir2/parametro", se escribiría la línea:

[2] El script de arranque de la red “service networking start” no lee el fichero `/etc/ethers`. Esta lectura debe hacerse expresamente. La carga de este fichero dará error si alguna de las IPs indicadas es una IP propia del equipo o es una IP no perteneciente a ninguna de las subredes en las que se encuentra trabajando el equipo según la tabla de encaminamiento actual.

/etc/sysctl.conf

```
dir1.dir2.parametro=valor
```

La carga de este fichero de configuración puede realizarse mediante el comando "`sysctl -p`", el cual es invocado durante el arranque del sistema (y por el script "`service network restart`").

b) Modificar manualmente el valor de un parámetro en el kernel (al reiniciar el equipo este cambio se perderá): dos opciones

```
echo "valor"  >  /proc/sys/carpeta_segun_parámetro/parametro
```

o haciendo uso del comando "`sysctl`", el cual ya asume que el parámetro se encuentra dentro de la carpeta virtual "`/proc/sys/`":

```
sysctl -w carpeta_segun_parámetro="valor"
```

Entre los diversos parámetro de red del kernel, podemos citar los siguientes a modo de ejemplo:

- "`/net/ipv4/icmp_echo_ignore_all`" (booleano, esto es, "0" o distinto de "0"): si se activa, la máquina no enviará respuesta a las solicitudes de ping que reciba del exterior. El valor predeterminado de este parámetro es "0".
- "`/net/ipv4/icmp_echo_ignore_broadcasts`" (booleano): si se activa, la máquina no enviará respuesta a las solicitudes de ping recibidas que tengan como dirección destino una dirección de subred o difusión (pero si responderá a las demás, si el parámetro "`icmp_echo_ignore_all`" no ha sido activado). El valor predeterminado de este parámetro es "0".

TAREAS

1º Conéctese vía SSH a algún equipo de su sala que se encuentre encendido (recuerde que, previamente, debe activar el servidor SSH en el equipo al que se vaya a conectar).

2º Desde dicha conexión SSH, haga un "ping" a la dirección IP de su propio equipo, comprobando que funciona.

3º En una consola de comandos de su propio equipo (no en la conexión SSH que tiene abierta con el otro ordenador), desactive la respuesta a las consultas ICMP mediante el comando:

```
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

TAREAS

4º Desde la conexión SSH, vuelva a realizar el ping a la dirección IP de su propio equipo, comprobando que ahora no funciona.

5º En la consola de comandos de su propio equipo, reactive la respuesta a las consultas ICMP:

```
echo 0 > /proc/sys//net/ipv4/icmp_echo_ignore_all
```

6º Desde la conexión SSH, vuelva a realizar el ping a la dirección IP de su propio equipo, comprobando que vuelve a funcionar.

3. Comandos básicos para trabajo en red de Linux (80 minutos)

Se comentan aquellos comandos que son útiles para descubrir la configuración de la red, con una breve descripción de las principales opciones de los mismos, debiendo consultar los manuales para obtener detalles más precisos acerca del modo de operación de dichos comandos.

Recuerde que para cambiar la configuración de su equipo, será necesario que disponga de los permisos de "root".

3.1 Comando ifconfig (10 minutos)

Este comando (interface configuration) se utiliza para configurar las interfaces de red que residen en el kernel del sistema operativo. En el inicio del sistema puede emplearse para activar las interfaces, si es necesario. Además, resulta útil para depurar o mejorar el funcionamiento del equipo (consulte "`man ifconfig`"):

- Si no se le pasan argumentos, al invocar el comando `ifconfig` se mostrará el estado de las interfaces que estén en estado activo.
- Si sólo se le pasa como argumento el nombre de una interfaz, al invocarlo sólo presentará el estado de la misma.
- Si pasamos a este comando la opción "`-a`" (y sólo esta) mostrará el estado de todas las interfaces de red, incluso de aquellas que estén desactivadas (down).
- En cualquier otro caso, `ifconfig` servirá para configurar una interfaz concreta (activándola o inhabilitándola, gestionando el uso del protocolo ARP, ...).

El comando "ifconfig" permite modificar manualmente la configuración de las interfaces de red (cambio que se perderá al reiniciar el sistema).

Por último, recuerde la existencia de la interfaz virtual local "lo" y que se utiliza para la comunicación IP interna.

TAREAS

Ejecute los siguientes comandos y analice el resultado que obtiene:

```
ifconfig -a
ifconfig eth0
ping su_propia_dirección_IP
ping  193.147.162.1
ifconfig eth0 192.168.0.1 netmask 255.255.255.252
ifconfig
route
ping su_propia_dirección_IP
ping  193.147.162.1
service networking restart
ifconfig
route
ping  193.147.162.1
```

3.2 Comando ping (5 minutos)

Se utiliza para hacer pruebas de accesibilidad desde un equipo a otro (consulte "`man ping`"). Utiliza el mensaje de solicitud de eco del protocolo ICMP para instar a otro equipo a que le devuelva un mensaje de respuesta de eco del mismo protocolo (si lo requiere, consulte en la bibliografía el funcionamiento de este protocolo).

TAREAS

Realice y analice las siguientes pruebas:

```
ping su_propia_dirección_IP        (párelo con [Ctrl-C])
ping -c 4 su_propia_dirección_IP
ping -c 4 -i 3 su_propia_dirección_IP
ping -w 5 su_propia_dirección_IP
ping -w 8 -i 3 su_propia_dirección_IP
ping www.google.es
ping -r www.google.es
ping IP_X
ping -r IP_X
ping 193.147.162.1
ping -r 193.147.162.1
```

3.3 Comando route (10 minutos)

Este comando permite manipular las tablas de rutas IP que tiene el kernel (consulte "`man route`"). El comando `route` se usa principalmente para establecer rutas estáticas a máquinas o redes específicas, a través de una interfaz, después de que dicha interfaz haya sido configurada con `ifconfig`.

- Si se teclea route sin argumentos, se mostrará el contenido actual de la tabla de rutas.
- Las opciones "`add`" y "`del`" se utilizan para que el comando `route` añada o borre una determinada entrada, sea de directa o indirecta. La opción "-n" sirve para que se muestren las direcciones IP en formato numérico, sin traducirlas a nombres.

TAREAS

Ejecute las siguientes operaciones y analice los resultados (observe cómo al modificar la dirección IP de la interfaz con "ifconfig" se eliminan las entradas de la tabla de encaminamiento asociadas a esa IP y se añade una entrada directa para la nueva dirección):

```
/sbin/route
route
route -n
ping IP_X
ping 193.147.162.1
ping -r 193.147.162.1
route del default
route
ping IP_X
ping 193.147.162.1
ping -r 193.147.162.1
ifconfig eth0 10.0.0.1 netmask 255.255.255.128
route add -net 10.0.0.0 netmask 255.255.255.240 dev eth0
route add default gw 10.0.0.2 netmask 0.0.0.0 dev eth0
route
ping IP_X
ping 193.147.162.1
route del -net 10.0.0.0 netmask 255.255.255.128 dev eth0
route
ping IP_X
ping 193.147.162.1
```

Añada nuevas entradas en la tabla de encaminamiento, indicándoles distintas métricas. Observe cómo se ordenan las entradas. En último lugar, restaure la configuración de la tarjeta de red y tabla de encaminamiento ejecutando:

```
ifdown eth0; ifup eth0
```

3.4 Comando traceroute (5 minutos)

El comando `traceroute` se utiliza para realizar el seguimiento de la ruta que sigue un paquete hasta alcanzar su destino. El único parámetro obligatorio es la dirección IP o el nombre del equipo de destino (consulte "`man traceroute`").

TAREAS

Realice las siguientes operaciones para estudiar el funcionamiento de este comando:

```
traceroute su_propia_dirección_IP
traceroute ait08.us.es
traceroute trajano.us.es
traceroute www.google.es
traceroute IP_X
traceroute -r IP_X
traceroute 193.147.162.1
traceroute -r 193.147.162.1
traceroute -I 193.147.162.1
```

3.5 Comando netstat (10 minutos)

Este comando (network statistics) permite ver, entre otras cosas, las conexiones de red establecidas, tablas de rutas, estadísticas de las interfaces, etc. (consulte "`man netstat`"):

- Si se invoca sin argumentos imprimirá una lista de los sockets abiertos.
- El primer argumento que se le pasa al comando controla el tipo de información que dicho comando mostrará.

Si se desea obtener un listado de todos los sockets "a la escucha" actualmente puede usarse el comando "`netstat -l`" (esta información no es la misma que la que ofrece el comando "`service --status-all`", el cual indica los servicios que están configurados para que se activen al arrancar el sistema). Para obtener los sockets de conexiones actuales, pueden usarse los comandos "netstat -t" (sockets TCP), "netstat -u" (sockets UDP), "netstat -w" (sockets Raw) o "netstat -x" (sockets UNIX).

Para obtener los sockets abiertos y el proceso al que están asignados, puede usarse el comando "`netstat -l -p`".

Además de las conexiones, "netstat" permite mostrar mucha más información relativa a la red (consulte "man netstat").

TAREAS

1º Realice las siguientes operaciones relativas a conexiones, y analice el resultado:

```
netstat
netstat -route
netstat -l
netstat -t
netstat -u
netstat -w
netstat -x
netstat -l -p
service --status-all
```

2º Ejecute los siguientes comandos de "netstat" y evalúe que información están mostrando:

```
netstat -nr
netstat -i
```

3.6 Comando nmap (10 minutos)

Este comando (network mapper) representa una potente herramienta para el escaneo del estado de los puertos y servicios en una red, permitiendo analizar la seguridad de una red. Se basa en el envío de paquetes IP para determinar qué ordenadores están disponibles en la red, qué servicios (puertos) están ofreciendo, qué sistemas operativos (y que versión) están corriendo, qué tipo de filtros o muros de seguridad se está usando, y docenas de otras características (consulte "`man nmap`").

Uno de los usos más típicos del comando es comprobar qué máquinas se encuentran activas en una red (o, al menos, qué máquinas responden a solicitudes de eco ICMP, puesto que, como se vio anteriormente, puede configurarse un sistema para que no responda a estas solicitudes) o los puertos abiertos en una determinada máquina.

TAREAS

Realice las siguientes operaciones y analice los resultados:

```
nmap -sP 172.16.4.0/24
nmap -p 80,8080 su_propia_dirección_IP
nmap -sS -O su_propia_dirección_IP
nmap -p 80,8080 IP_X
nmap -sS -O IP_X
```

3.7 Comando arp (10 minutos)

El protocolo ARP (Address Resolution Protocol) permite la obtención de la dirección física (MAC) correspondiente a la dirección de red **de una máquina existente en la misma red local** (ARP es un protocolo de nivel de enlace), usando para ello el envío de un mensaje ARP Broadcast. La información obtenida a partir del protocolo ARP es almacenada dinámicamente por el kernel en la caché ARP.

El comando `arp` manipula la caché ARP (almacena las asociaciones MAC/IP) de varias formas (consulte "`man arp`"). Recuerde: **el comando arp sólo opera sobre la caché ARP, no tiene ninguna influencia sobre el protocolo ARP, el cual es controlado únicamente por el kernel.** La caché ARP se encuentra almacenada dentro del propio "kernel", pudiendo acceder a ella con el comando "`arp`" o, equivalentemente, mediante "`cat /proc/net/arp`".

Con objeto de acelerar la traducción entre direcciones físicas e IP, puede crearse el fichero "`/etc/ethers`", ya mencionado anteriormente, con las asociaciones que resulten más usuales, cargando su información mediante el comando "`arp -f /etc/ethers`".

TAREAS

1º Analice el resultado de la ejecución de los siguientes comandos

```
arp
arp -a
ping IP_X
arp IP_X
arp 193.147.162.1
ping 193.147.162.1
arp -d IP_X
arp IP_X
ping IP_X
arp
```

2º Cree un fichero "/etc/ethers" válido y cárguelo con el comando "arp".

3º La caché ARP puede ser consultada mediante múltiples comandos posibles. Analice y compare la información mostrada por los siguientes comandos:

```
arp –n
arp –nav
ip neighbor show
cat /proc/net/arp
```

3.8 Comandos host y nslookup (5 minutos)

Ambos comandos ("host" y "name server look up") corresponden a utilidades DNS para realizar conversiones entre nombres de red y direcciones IP. Mientras que "host" realiza la resolución mediante el fichero "`/etc/hosts`" y los servidores DNS, el comando "nslookup" no utiliza el fichero "`/etc/hosts`".

TAREAS

A modo de ejemplo ejecute lo siguiente y analice los resultados:

```
host www.google.es
host google.es
nslookup ait08.us.es
host 193.147.162.169
nslookup 193.147.162.130
```

3.9 Comando dig (10 minutos)

Este comando (domain information groper) ofrece información relacionada con los servidores DNS, estando determinado su funcionamiento por la RFC 1035 (consulte "man dig").

TAREAS

Ejecute los siguientes comandos y analice los resultados que aparecen:

```
dig www.google.es
dig www.google.es NS
dig us.es +trace
```

3.10 Comando ip (5 minutos)

Comando multifunción que permite una configuración avanzada de la red, incluyendo funciones realizadas mediante los comandos anteriormente analizados (consulte "man ip").

TAREAS

Ejecute los siguientes comandos y analice el resultado:

```
ip route
ip addr
ip link
ip neigh
ip maddr
```

4. Monitorización de tráfico de red en Linux (30 minutos)

Un sniffer, rastreador, monitor de red o analizador de protocolos es una herramienta para capturar e interpretar el tráfico que circule por el tramo de red en el que se encuentre el equipo. Mediante este método se pueden capturar claves de acceso, datos que se transmiten, números de secuencia, etc...

Estos programas suelen usar el modo promiscuo de las interfaces de red, bajo el cual la interfaz permite capturar toda la información que transcurra por el tramo de red al que esté conectada, y no solamente los paquetes dirigidos a ella.

En esta práctica se utilizarán los siguientes analizadores de protocolos sobre una red Ethernet:

- tcpdump: Analizador de protocolos en modo texto.
- wireshark: Analizador de protocolos en modo gráfico.

Ambas se basan en la librería de captura "pcap" (libpcap en sistemas Linux y winpcap en sistemas Windows). Esto lleva a que presenten muchas características comunes: misma sintaxis para definir filtros de capturas, similar información proporcionada,... En el **Anexo** de esta práctica se recoge la **Sintaxis de definición de filtros de captura** empleada por los dos analizadores que a continuación se estudian.

Antes de continuar, ejecute los siguientes comandos para regenerar el nombre correcto del equipo en la red guardado en el fichero "/etc/hostname"[3]:

```
ifdown eth0; ifup eth0
```

4.1 Analizador de protocolos TcpDump (10 minutos)

El programa se ejecuta del siguiente modo (como "root"):

```
tcpdump [parámetros] [filtro_de_captura]
```

escribiéndose el filtro de captura conforme a la sintaxis del Anexo (si no se indica ningún filtro de captura, se capturará todo el tráfico) y destacando como principales parámetros los siguientes:

Parámetros	Funcionalidad
-e	Incluye las direcciones de nivel de enlace en la información impresa.

[3] Si el nombre del equipo no se regenera, al intentar arrancar "wireshark" desde consola como superusuario, puede producirse un error de acceso al servidor X (el nombre se intenta traducir a IP, y no coincidirá con la IP real del equipo, fallando el proceso de autenticación).

Parámetros	Funcionalidad
-c	Con la opción `-c` se especifica el número de paquetes que se quiere capturar. Si no se indica esta opción, el programa se ejecuta hasta que se pulse Ctrl-C. Ejemplo: `tcpdump -c 20`
-i	Indica la interfaz de red a monitorizar. Si no se indica ninguna, se usa la primera en orden alfanumérico. Con "any" se captura de todas. Si se quiere monitorizar la interfaz `eth0`, se usa el comando: `tcpdump -i eth0`
-n	Cuando se está monitorizando la red puede que no interese que tcpdump intente resolver los nombres de las maquinas (por motivos de seguridad por ejemplo). Para ello se dispone de la opción `-n`.
-s	Para establecer la longitud de los datos que captura tcpdump usamos `-s len`, donde `len` es la longitud que nos interesa. Por defecto, tcpdump sólo captura los primeros 65535 bytes, lo cual es útil si lo único que se quiere son las cabeceras IP, TCP o UDP, pero en caso de estar analizando protocolos como NFS se truncan los datos. En ese caso se puede ajustar la longitud de la captura a la MTU (Maximun Transmission Unit) del medio que se está usando. Por ejemplo, para capturar toda la trama Ethernet se puede usar `-s 1500`.
-v	En función de la cantidad de información que se quiera que tcpdump capture y decodifique, se puede usar `-v`,`-vv`,`-vvv` aumentando el grado de información con cada una de las opciones.
-x	Para imprimir el contenido del paquete en formato hexadecimal, se puede usar la opción `-x`. La longitud que imprime viene determinada por la opción `-s` o los 68 bytes que muestra por defecto.
-r, -w	Se puede trabajar offline con tcpdump. Para grabar la captura para posteriormente leerla y analizarla se utiliza la opción `-w file` donde "`file`" es el nombre del fichero donde se grabará la captura de datos. Posteriormente se puede leer y analizar offline con `-r file`. Además este tipo de ficheros de captura lo pueden leer otros analizadores como, por ejemplo, wireshark.

Tal como se ha indicado, para que tcpdump sea capaz de capturar todos los paquetes que llegan al equipo (no sólo los destinado a él) resulta necesario que la interfaz de red sea configurada en modo promiscuo. Puede hacerse esto haciendo uso del comando `ifconfig`:

```
ifconfig eth0 promisc
```

Para eliminar el modo promiscuo bastaría usar el comando recíproco:

```
ifconfig eth0 -promisc
```

TAREAS

1º Haga pruebas con todas las anteriores opciones del comando tcpdump y analice los resultados.

2º Aplicando los filtros de captura necesarios, realice las capturas de paquetes necesarias para:

- Detectar los paquetes de su subred cuyo puerto origen o destino sea el 80 (si no detecta nada, pruebe con el puerto 8080). Mientras realiza la captura, intente acceder a una página Web que estuviese alojada en otro equipo distinto al suyo. Guarde la captura anterior en un fichero para posteriormente poder analizarla usando wireshark.
- Capture los paquetes que contengan una determinada dirección física.
- Detectar el ataque conocido como smurf (envío de solicitudes ping a la dirección de difusión de la red).

3º Ejecute como superusuario los siguiente comandos, y observe cómo el comando "ifconfig" indica cuándo una interfaz tiene el modo promiscuo activado:

```
ifconfig eth0 -promisc
ifconfig -a
ifconfig eth0 promisc
ifconfig -a
```

4.2 Analizador de protocolos Wireshark (20 minutos)

Esta aplicación permite analizar paquetes de forma interactiva o desde un archivo en el que previamente se haya realizado una captura de información. Para arrancarla basta ejecutar el comando "`wireshark &`" (siempre como "root").

TAREAS

Utilizando la herramienta "wireshark", realice las siguientes operaciones:

1º Aplicando el filtro de captura necesario, active la captura de paquetes para el puerto 22 en la interfaz local "eth0". Tras ello, conéctese mediante el servicio SSH al equipo con IP_X y ejecute en él el comando "ls". Tras ello, salga de la sesión SSH. Detenga la captura y analice la información.

2º Active la captura de paquetes en la interfaz local "eth0" con el siguiente filtro de captura:

```
host dirección_IP_interfaz_eth0 and (tcp and port 80)
```

Tras ello, conéctese mediante un navegador web a la página "http://www.google.es". Detenga la captura y analice la información. Seleccione una de las tramas recibidas (cualquiera), pulse con el botón derecho del ratón y seleccione "`Follow TCP Stream`". Comprobará que se puede ver todo lo que se ha transmitido y recibido, lo que indica la nula seguridad de este protocolo.

3º Repita lo mismo que en el apartado anterior pero cambiando en el filtro el puerto "80" por el "443", y accediendo a la página "https://ait08.us.es/". Comprobará que los datos aparecen encriptados y no son visibles.

4º Repita el paso anterior, pero en lugar de usar el filtro de captura use el filtro de visualización:

```
ip.addr == dirección_IP_interfaz_eth0 and tcp and tcp.port
== 443
```

5º Active la captura de paquetes con el siguiente filtro:

```
port 68
```

Compruebe que en el fichero "/etc/network/interfaces", su tarjeta "eth0" tiene configurado el valor "dhcp". Ejecute el siguiente comando para solicitar al servidor DHCP que vuelva a asignarle la dirección IP:

```
dhclient -v
```

Detenga la captura en wireshark y analice la información obtenida. Observe la información que el servidor DHCP le da al cliente y la MAC que manda el cliente al servidor.

TAREAS

6º Capture el tráfico IP entrante y saliente en la red del CdC en la que se encuentra.

7º Active la captura de paquetes en la interfaz local "eth0" con el filtro de captura "icmp". Ejecute los comandos:

```
ping 127.0.0.1
ping 127.4.5.128
ping IP_de_su_interfaz_eth0
```

Compruebe que no ha capturado nada a consecuencia de esos comandos, y razone el motivo de ello. Detenga la captura.

8º Vuelva a activar la captura de paquetes con el filtro de captura "icmp", pero ahora sobre la interfaz local "lo". Ejecute los comandos:

```
ping 127.0.0.1
ping 127.4.5.128
ping IP_de_su_interfaz_eth0
```

Realice las siguientes operaciones y justifique los resultados:

- Compruebe cómo los paquetes ICMP de todos esos comandos han sido cursados por dicha interfaz "lo".
- Analice las direcciones IP origen y destino capturadas para los distintos comandos anteriores, y compruebe cómo no coinciden.

9º Mediante el comando "arp -d IP_X", elimine en la caché ARP la entrada correspondiente al equipo con dirección IP_X. Ejecute "arp -n" para comprobar que se ha eliminado correctamente la MAC. Tras ello, ponga wireshark a capturar en la interfaz eth0 con el filtro de captura "ether proto \arp". Ejecute el comando "ping IP_X" y compruebe en wireshark cómo se obtienen mensajes ARP de solicitud y respuesta para obtener la dirección MAC_X del equipo IP_X. Detenga la captura.

10º Mediante el comando "arp -n", mire cuál es la dirección MAC_X asociada al equipo IP_X y guárdela en el fichero "/etc/ethers" añadiendo en éste la línea "MAC_X IP_X". Mediante el comando "arp -d IP_X", vuelva a eliminar en la caché ARP la entrada correspondiente al equipo con dirección IP_X. Ejecute "arp -n" para comprobar que se ha eliminado correctamente la MAC. Tras ello, ponga de nuevo wireshark a capturar en la interfaz eth0 con el filtro de captura "ether proto \arp". Ejecute el comando "ping IP_X" y compruebe en wireshark cómo ahora no se están enviando mensajes ARP de solicitud y respuesta para obtener la dirección MAC_X del equipo IP_X (al haber sido añadida manualmente en la caché ARP). Detenga la captura.

Dado que tanto TcpDump como Wireshark se basan en la misma librería de captura, una captura realizada con TcpDump puede ser visualizada posteriormente con Wireshark.

TAREAS

Realice las siguientes operaciones:

1º Capture con tcpdump el tráfico de la interfaz eth0, guardando la captura en el fichero "/tmp/captura.cap":

```
tcpdump -i eth0 -w /tmp/captura.cap
```

2º Visualice en Wireshark la captura realizada:

```
wireshark /tmp/captura.cap
```

5. Anexo (no evaluable): Sintaxis de expresiones para filtros de captura

PARA PROFUNDIZAR...

El contenido de este Anexo se ofrece para permitir que el alumno pueda profundizar en los conocimientos expuestos en la Práctica.

Este punto tiene como objeto describir la sintaxis de las expresiones de filtrado utilizadas en los programas basados en la librería "`pcap`" (consulte "man tcpdump" y "http://wiki.wireshark.org/CaptureFilters" para una información más detallada sobre la sintaxis de estas expresiones). Se harán pruebas con "`tcpdump`", pero todo es igualmente aplicable a "`wireshark`" o demás aplicaciones basadas en esa librería.

Un **filtro de captura o expresión de filtrado** tiene por objeto indicar las características de la captura a realizar. En ausencia de ésta se capturará todo el tráfico que vea el adaptador de red. La sintaxis de un filtro es un conjunto de primitivas enlazadas por operadores:

```
Primitiva operador Primitiva
```

donde cada **primitiva** está formada por un calificador y su valor asociado:

```
calificador valor
```

siendo:

a) **Modificador o calificador**: establece la propiedad que se desea especificar. Los 3 modificadores posibles son:

- Tipo: puede ser `host`, `net` o `port`. Indican, respectivamente, una maquina (por ejemplo `host 172.16.17.1`), una red completa (por ejemplo `net 172.16`), o un puerto concreto (por ejemplo `port 22`). Por defecto es `host`.

- Dirección: especifica desde o hacia donde se va a mirar el flujo de datos. Puede ser `src` (dirección fuente) o `dst` (dirección destino), y podemos combinarlos con `or` y `and`. Para el caso de protocolos punto a punto se puede sustituir por `inbound` (flujo entrante) u `outbound` (flujo saliente). Por ejemplo si se quiere dirección destino `172.16.17.2` y origen `193.147.162.169`, el filtro sería:

  ```
  dst 172.16.17.2 and src 193.147.162.169
  ```

Si se quiere que sea la dirección destino `172.16.17.2` o la dirección origen `193.147.162.169`, sería:

```
dst 172.16.17.2 or src 193.147.162.169
```

Si no existe este modificador se supone "`src or dst`".

- Protocolo: protocolo a capturar, pudiendo ser `tcp`, `udp`, `ip`, `ether` (en este caso captura tramas a nivel de enlace), `arp` (peticiones `arp`), `rarp` (peticiones reverse-arp), `fddi` (para redes FDDI, cuyo encapsulado es similar a `ether`). Existen otros protocolos más para nivel de enlace, pero su uso es escaso.

b) **Valor o ID de la primitiva**: valor numérico o nombre alfanumérico del calificador asociado. Pueden representar nombres de host, direcciónes IP, números de puertos y otros valores de filtrado.

Las primitivas completas que se pueden definir usando los anteriores calificadores son las siguientes (lo que aparece entre `[` y `]` es opcional, y el símbolo `|` significa "o"):

I) **<u>Calificador "Tipo"</u>**:

a) "`[dst|src] host maquina`": especifica la dirección destino u origen del paquete con el valor maquina el cual puede ser una dirección IPv4 (o IPv6 si se ha compilado soporte para el mismo), o un nombre DNS. Ejemplos (sustituya `x` para indicar la IP de su equipo en la subred del laboratorio):

- Capture el trafico cuya IP origen sea su máquina `172.16.17.x`:

```
tcpdump src host 172.16.17.x
```

- Capture todo el tráfico cuya dirección origen o destino sea su máquina:

```
tcpdump host 172.16.17.x
```

b) "`[dst|src] net red`": dirección de red destino, origen o ambas. El parámetro red puede ser una dirección numérica (por ejemplo `192.168.1.0`) o bien un alias que se resuelva a dirección (en los sistemas Unix se obtiene con ayuda del fichero "`/etc/networks`"). Decir que también se admite el clásico direccionamiento CIDR, que permite especificar una máscara usando "`net IP_red mask mascara`" o bien "`net red/bits`" (número de bits a 1 de la máscara); y hacer notar que el uso de "`net ... mask`" no es compatible con direcciones IPv6. Ejemplos:

- Capture todo el tráfico cuya red destino sea `172.16.17.x` (cambie "`x`" por "`0`" ó "`128`" según la subred del laboratorio en la que se encuentre su equipo):

```
tcpdump dst net 172.16.17.x
```

- Capture todo el tráfico cuya red origen sea `172.16.17.0/24`, esto es, capture todos los paquetes cuya dirección IP origen esté en el rango comprendido entre "`172.16.17.0`" y "`172.16.17.255`" (los dos comandos indicados son equivalentes):

```
tcpdump src net 172.16.17.0 mask 255.255.255.0
tcpdump src net 172.16.17.0/24
```

- Capture todo el tráfico con origen o destino `172.16.17.0/24`, esto es, capture todos los paquetes cuya dirección IP origen o destino esté en el rango comprendido entre "`172.16.17.0`" y "`172.16.17.255`" (los dos comandos indicados son equivalentes):

```
tcpdump net 172.16.17.0 mask 255.255.255.0
tcpdump net 172.16.17.0/24
```

c) "`[dst|src] port puerto`": puerto del paquete (ya sea udp o tcp). El puerto es un valor numérico entre 0-65535 o bien un nombre que en Unix se resuelve a través del `/etc/services` (de partida, el número que este fichero asocia a cada puerto es el establecido por la IANA para TCP/UDP, http://www.iana.org/assignments/service-names-port-numbers/). Ejemplos:

- Capture todo el tráfico con destino al puerto 22 (ssh): son equivalentes

```
tcpdump dst port 22
tcpdump dst port ssh
```

- Capture todo el tráfico con destino u origen puerto 80 (http): son equivalentes

```
tcpdump port 80
tcpdump port http
```

II) **Calificador "Protocolo"**:

a) "`ether src|dst|host dirección_ethernet`": especifica la dirección física del paquete. Este filtro es cierto si la dirección origen (`src`), la destino (`dst`) o cualquiera de las dos (`host`) coincide con dirección_ethernet. Ejemplos:

- Capture el tráfico con destino a la dirección Ethernet de la interfaz `eth0` de su máquina:

```
tcpdump ether dst xx:xx:xx:xx:xx:xx
```

b) "`ether proto \protocolo`": la condición impuesta por este filtro se cumple si el protocolo que contiene la trama es de tipo `protocolo`. Los protocolos de nivel de enlace y red admitidos son `ip`, `ip6`, `arp`, `rarp`, `atalk`, `aarp`, `decnet`, `sca`, `lat`, `mopdl`, `moprc` e `iso`. Esos nombres de protocolos de red corresponden en realidad

a un número establecido por la IANA para el campo Ethernet "Ethertype" (puede consultar la asignación en http://www.iana.org/assignments/ieee-802-numbers/).

Para facilitar la escritura de estas expresiones se encuentran definido un alias para cada protocolo: `ip`, `ip6`, `arp`, `rarp`, ..., que equivaldrían a "`ether proto \ip`", "`ether proto \ip6`", etc. (en general, "`protocolo` ⇒ `ether proto \protocolo`"). Obsérvese cómo el alias usa el mismo nombre que el protocolo; este es el motivo por el que "`protocolo`" en la expresión "`ether proto \protocolo`" va escapado (precedido por "\"), para diferénciarlo del alias.

RECUERDE...

En la consola de comandos, el shell usa diversos caracteres especiales como operadores. Por este motivo, cuando se desea escribir estas expresiones de filtrado en la consola (con tcpdump), si la expresión contiene alguno de esos caracteres especiales ("\", "(", ")", "&", "!", ...), debe indicarse al shell que corresponden a caracteres de la expresión de filtrado y no a operadores del shell. Para ello, pueden usarse dos alternativas:

- Escribir la expresión entre comillas: `tcpdump -n "ether proto \arp"`
- Escapar esos caracteres especiales: `tcpdump -n ether proto \\arp`

Ejemplos (se usa la opción `-n` de tcpdump para no realizar la conversión de IPs a nombres, pero no pertenece a la expresión del filtro):

- Capture todo el tráfico arp (ethertype "2054" en decimal o "0806" en hexadecimal): son equivalentes

```
tcpdump -n ether proto \\arp
tcpdump -n "ether proto \arp"
tcpdump -n ether proto 2054
tcpdump -n ether proto 0x0806
tcpdump -n arp
```

- Capture todo tráfico ip: son equivalentes

```
tcpdump -n ether proto \\ip
tcpdump -n "ether proto \ip"
tcpdump -n ether proto 2048
tcpdump -n ip
```

c) "`ether broadcast`": captura las tramas dirigidas hacia la dirección de difusión Ethernet (todos los bits a "1"). La palabra `ether` es opcional.

d) "`ether multicast`": captura las tramas dirigidas a una dirección multicast ethernet (primer bit a uno).

e) "`ip|ip6 proto \protocolo`": se captura el `protocolo` encapsulado en IP que se le indique. El protocolo puede ser `icmp`, `icmp6`, `igmp` (Internet Group Managent Protocol), `igrp` (Interior Gateway Routing Protocol), `pim` (Protocol Independent Multicast), `ah` (IP Authentication header), `esp` (Encapsulating Security Payload), `udp` o `tcp`. Esos nombres de protocolos sobre IP corresponden en realidad a un número que en los sistemas tipo Unix se encuentra registrado en el fichero `/etc/protocols` (de partida, el número que este fichero asocia a cada protocolo es el establecido por la IANA para el campo IP "identificador de protocolo", http://www.iana.org/assignments/protocol-numbers/).

Por comodidad existe un alias para cada protocolo, `tcp`, `udp`, `icmp, ...` que equivalen a "`ip proto \tcp or ip6 proto \tcp`", "`ip proto \udp or ip6 proto \udp`", etc. Por este motivo, al igual que en el caso de "ether", "`protocolo`" en la expresión "`ip proto \protocolo`" va escapado (precedido por "\"), para diferénciarlo del alias de igual nombre.

Ejemplos:

- Capturar todos los paquetes icmp: son equivalentes

```
tcpdump ip proto \\icmp
tcpdump "ip proto \icmp"
tcpdump ip proto 1
tcpdump icmp
```

- Capturar todo el tráfico udp: son equivalentes

```
tcpdump ip proto \\udp
tcpdump "ip proto \udp"
tcpdump ip proto 17
tcpdump udp
```

f) "`ip|ip6 protochain \protocolo`": en este caso lo que se busca es que dentro de las diferentes cabeceras encapsuladas en un paquete IP, una de ellas pertenezca al protocolo especificado.

g) "`ip|ip6 broadcast`": se capturan los paquetes dirigidos a la dirección de difusión de la red IP (las direcciones que son todo 0 o 1, o bien la dirección local de la subred).

h) "`ip|ip6 multicast`": se capturan los paquetes dirigidos a una dirección `multicast` IP.

III) Otras primitivas:

a) "`vlan [vlanid]`": se capturan paquetes 802.1Q VLAN. Esto modifica el resto de la interpretación del paquete capturado, en especial los desplazamientos a partir de los cuales se empieza a decodificar los protocolos, ya que se asume que estamos capturando

paquetes que viajan en tramas VLAN. Si está presente el parámetro `vlanid`, sólo se mostraran aquellos paquetes que vayan a la VLAN con identificador `vlanid`.

b) "`gateway maquina`": se capturan los paquetes que usen el equipo de dirección IP `maquina` como router. Los paquetes que cumplen con esa condición son aquellos que tienen como dirección Ethernet destino `maquina`, pero ni la dirección IP destino u origen es dicho equipo, usando así dicho equipo como nodo intermedio. `maquina` debe estar definida tanto en `/etc/ethers` como en `/etc/hosts`.

c) "`less|greater longitud`": captura los paquetes cuyo tamaño sea menor, mayor o igual a la longitud indicada.

5.1 Combinación de Filtros

Se pueden combinar las expresiones anteriores con los ayuda de los operadores `not`, `and` y `or` (corresponden a la "negación", el "y lógico" y el "o lógico"), dando lugar a filtros más complejos. Se pueden usar también los equivalentes del lenguaje C: "`!`", "`&&`" o "`||`".

Siempre se pueden combinar expresiones con ayuda de **paréntesis**. El uso de paréntesis permite que los grupos de expresiones de filtrado sean evaluados juntos como una sola primitiva virtual. En los shell de Unix, los paréntesis deben escaparse (anteponer el simbolo "\" antes del carácter especial para que no se interprete como tal carácter especial) porque son metacaracteres que se interpretan (esto es aplicable a "`tcpdump`", que se ejecuta bajo la Shell, pero no a "`wireshark`" pues éste usa interfaz X-Windows). Ejemplos:

- Capture todo el tráfico Web (TCP port 80):

```
tcpdump tcp and port 80
```

- Capture todas las peticiones DNS:

```
tcpdump udp and dst port 53
```

- Capture el tráfico al puerto telnet o ssh:

```
tcpdump tcp and \(port 22 or port 23\)
tcpdump tcp and "(port 22 or port 23)"
```

- Capture todo el tráfico excepto el Web:

```
tcpdump tcp and not port 80
```

5.2 Filtros avanzados

La sintaxis de los filtros de capturas permite hacer filtros a mano, indicando qué bytes de la trama se desean capturar e interpretarlos. Cuando queremos definir filtros de esta manera la expresión general es (en el manual `man tcpdump` se ofrece información detallada sobre la sintaxis de estos filtros avanzados):

```
expr relop expr
```

donde:

- "`relop`" puede ser cualquiera de las operaciones de relación de C: `>` ,`<` , `>=`, `<=`, `=` y `!=`.
- "`expr`" es una expresión aritmética compuesta por una serie de números enteros, los operadores binarios de C, (`+, -, *, /, & y |`), un operador de longitud "`len`", y una serie de palabras reservadas que nos permiten el acceso a los diferentes paquetes de datos (`ether, fddi, tr, ip, arp, rarp, tcp, udp, icmp, ip6`).

Resumen Sintaxis Filtros de captura (básica)

I) Filtrado a **nivel de Red** (IP) **y Transporte** (TCP/UDP).

A) Filtrado por direcciones IP o puertos TCP/UDP

Opcional
```
[ip][procedencia] tipo_filtrado  valor
```

procedencia	Significado
src	Origen
dst	Destino
Por omisión	"src or dst"

tipo_filtrado	valor
host	IP concreta (o **alias** de /etc/hosts y DNS, man hosts)
net	Subred, o **alias** en /etc/networks (Ej: 172.16.17, 172.16.17.0/24, 172.16.17.0 mask 255.255.255.0)
port portrange	Puerto TCP/UDP (o **alias** en /etc/services, como **ftp=21**, **http=80**,...,http://www.iana.org/assignments/port-numbers)

Filtros especiales	ip/ip6 broadcast	ip/ip6 multicast
Mensajes capturados	**dst** IP **subred/difusión** red o difusión general	**dst** IPv4/IPv6 **multicast**

B) Filtrado por protocolo de transporte sobre IP

```
[protocolo_red] proto \protocolo_transporte
```

protocolo_red	ip *(por omisión)*	ip6
protocolo_transporte	Requieren escapado: \tcp, \udp, \icmp, \igmp, \igrp, \pim, \ah, \asp, ... o **número equivalente** (/etc/protocols, como **\tcp = 6**, http://www.iana.org/assignments/protocol-numbers).	
Alias (para todos igual)	**tcp** = **ip/ip6 proto \tcp** **udp** = **ip/ip6 proto \udp**	**icmp** = **ip/ip6 proto \icmp** ...

II) Filtrado a **nivel de Enlace** (Ethernet).

A) Filtrado por direcciones MAC

```
ether  procedencia  valor
```

procedencia	src	dst	host
valor (ninguno por omisión)	**MAC** origen (o **alias** de /etc/ethers, man ethers)	**MAC** destino (o **alias** de /etc/ethers)	**MAC** src o dst

Filtros especiales	Mensajes capturados	Alias
ether broadcast	**dst MAC de difusión** (1...1)	broadcast = ether broadcast
ether multicast	**dst MAC multicast** (10...0)	multicast = ether multicast
gateway Name_router	Paquetes que han llegado **atravesando el router**; debe usar /etc/hosts y /etc/ethers (igual: *ether* host MAC_router and not *ip* host IP_router)	

B) Filtrado por protocolo de red sobre Ethernet

```
ether proto \protocolo_red
```

protocolo_red	Requieren escapado: \arp, \rarp, \ip, \ip6, ... o **número ethertype equivalente** (/etc/ethertypes, como **\ip = 0x0800** ó **\ip = 2048**, http://www.iana.org/assignments/ethernet-numbers).
Alias (para todos igual)	**arp** = **ether proto \arp** **rarp** = **ether proto \rarp** **ip** = **ether proto \ip** ...

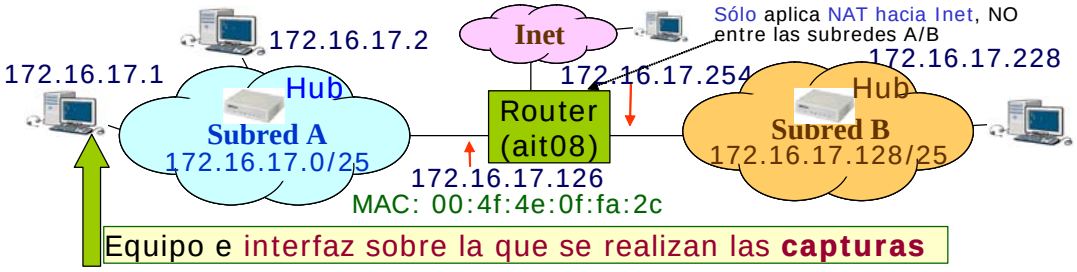
*) Operadores (negación y combinación de filtros): "**not**" ó "**!**", "**or**" ó "**||**", "**and**" ó "**&&**" (los operadores pueden usarse **antes y dentro de las expresiones**). Ej:

```
src or dst host not 172.16.17.1 \&\& \(port not 22 and not ftp-data\)
ether dst 01:02:03:04:05:06 or \(ip and tcp\) and \! \(ip proto \\udp\)
```

NOTA: Los escapados "\" están escritos pensando en la escritura del filtro en la línea de comandos sin comillas.

- "**&& !**" o "**|| !**" dan error de sintaxis, pero sí se admiten "**and !**" o "**or !**".
- Entre dos valores (ej. "**net (IP or IP2)**") sólo pueden usarse "**and**" y "**or**", no "**&&**" ni "**||**"
- Sólo se admiten paréntesis "**()**" antes de una expresión o del último campo (**valor/prot.**)

Ejemplos



Tráfico a capturar	Filtro de captura
Todo el tráfico (sólo útil para pequeñas capturas)	*No especificar* ***ninguno***
Tráfico arp procedente del router (ait08)	`ether src 00:4f:4e:0f:fa:2c && ether proto \arp`
Tráfico que llega de la Subred B, exceptuando al router	`src net 172.16.17.128/25 and ! src host 172.16.17.126`
Tráfico que llega de fuera de la Subred A, exceptuando al router	`ether src 00:4f:4e:0f:fa:2c and ! src host 172.16.17.126`
Captura todo el trafico excepto el de broadcast y multicast	`not broadcast and not multicast not (broadcast and multicast)`
Solicitudes http (puerto estándar) ó mensajes icmp procedentes de Internet y destinadas al equipo con IP 172.16.17.2	`ether src 00:4f:4e:0f:fa:2c and ! (src net (10 or 172.16 or 192.168)) && (icmp[icmptype]=icmp-echo or src port 80) && dst 172.16.17.2`