

Práctica 3: Monitorización de la Red

Redes de Computadores – U.L.P.G.C.

Índice

Introducción.....	2
Netstat.....	3
Apartado a.....	3
Apartado b.....	9
Apartado c.....	10
Apartado d.....	11
Tcpdump.....	13
Apartado a.....	13
Apartado b.....	14
Apartado c.....	15
Apartado d.....	16
Traceroute.....	21
Conexión con www.ulpgc.es	21
Conexión con www.rediris.es	23
Ethereal.....	25
Manejo de la herramienta gráfica Ethereal.....	25
Modificadores a la hora de invocar Ethereal.....	28
Filtrado de Paquetes.....	28
Referencias Bibliográficas.....	30

Introducción

En esta práctica, con los equipos ya configurados en red y para el acceso a Internet, que es la tarea realizada en la práctica 2, anterior a ésta, se usará una serie de comandos útiles para conocer el estado de la red. Así, podremos ver el estado de las conexiones, desde el nivel de puertos e incluso los protocolos. A su vez, se puede observar el tráfico activo, donde destaca la herramienta tcpdump. De todo el tráfico que se vea se podrá filtrar el que nos interese y en función de todo ello y los resultados que se tengan se mostrarán las conclusiones oportunas.

Netstat

Con el comando *netstat* podemos ver las conexiones activas en nuestro equipo en un momento determinado, es decir, el tráfico de la red en dicho instante. Este comando admite varios modificadores, que le dan mayor utilidad, y serán usados para la resolución de los siguientes apartados del guión de prácticas.

Apartado a

En este punto nos limitamos a ejecutar directamente *netstat* sin modificadores, de forma que obtendremos un resultado como el siguiente, ejecutándolo desde el equipo pasarela:

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      1      0 localhost.localdo:32772 localhost.localdoma:ipp CLOSE_WAIT

Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags   Type       State      I-Node Path
unix  12    []     DGRAM          3309 /dev/log
unix  3     []     STREAM        CONNECTED  5346
unix  3     []     STREAM        CONNECTED  5345
unix  3     []     STREAM        CONNECTED  5337 /tmp/orbit-root/linc-bcd-0-4375f62112c4d
unix  3     []     STREAM        CONNECTED  5336
unix  3     []     STREAM        CONNECTED  5335 /tmp/orbit-root/linc-b32-0-77aa27522e803
unix  3     []     STREAM        CONNECTED  5334
unix  3     []     STREAM        CONNECTED  5333 /tmp/orbit-root/linc-bcd-0-4375f62112c4d
unix  3     []     STREAM        CONNECTED  5332
unix  3     []     STREAM        CONNECTED  5329 /tmp/orbit-root/linc-b2d-0-3746f073551e9
unix  3     []     STREAM        CONNECTED  5328
unix  3     []     STREAM        CONNECTED  5323 /tmp/.ICE-unix/2789
unix  3     []     STREAM        CONNECTED  5322
unix  3     []     STREAM        CONNECTED  5317 /tmp/.X11-unix/X0
unix  3     []     STREAM        CONNECTED  5316
unix  3     []     STREAM        CONNECTED  5242 /tmp/orbit-root/linc-b99-0-117f39b52a81b
unix  3     []     STREAM        CONNECTED  5241
unix  3     []     STREAM        CONNECTED  5240 /tmp/orbit-root/linc-bcb-0-329a70666ebc2
unix  3     []     STREAM        CONNECTED  5239
```

unix	3	[]	STREAM	CONNECTED	5234	/tmp/orbit-root/linc-bcb-0-329a70666ebc2
unix	3	[]	STREAM	CONNECTED	5233	
unix	3	[]	STREAM	CONNECTED	5232	/tmp/orbit-root/linc-b32-0-77aa27522e803
unix	3	[]	STREAM	CONNECTED	5231	
unix	3	[]	STREAM	CONNECTED	5230	/tmp/orbit-root/linc-bcb-0-329a70666ebc2
unix	3	[]	STREAM	CONNECTED	5229	
unix	3	[]	STREAM	CONNECTED	5226	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix	3	[]	STREAM	CONNECTED	5225	
unix	3	[]	STREAM	CONNECTED	5220	/tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	5219	
unix	3	[]	STREAM	CONNECTED	5209	/tmp/orbit-root/linc-b99-0-117f39b52a81b
unix	3	[]	STREAM	CONNECTED	5208	
unix	3	[]	STREAM	CONNECTED	5207	/tmp/orbit-root/linc-bc4-0-4c9717c9f020d
unix	3	[]	STREAM	CONNECTED	5206	
unix	3	[]	STREAM	CONNECTED	5201	/tmp/orbit-root/linc-bc4-0-4c9717c9f020d
unix	3	[]	STREAM	CONNECTED	5200	
unix	3	[]	STREAM	CONNECTED	5199	/tmp/orbit-root/linc-b32-0-77aa27522e803
unix	3	[]	STREAM	CONNECTED	5198	
unix	3	[]	STREAM	CONNECTED	5197	/tmp/orbit-root/linc-bc4-0-4c9717c9f020d
unix	3	[]	STREAM	CONNECTED	5196	
unix	3	[]	STREAM	CONNECTED	5193	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix	3	[]	STREAM	CONNECTED	5192	
unix	3	[]	STREAM	CONNECTED	5187	/tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	5186	
unix	3	[]	STREAM	CONNECTED	5183	/tmp/orbit-root/linc-b99-0-117f39b52a81b
unix	3	[]	STREAM	CONNECTED	5182	
unix	3	[]	STREAM	CONNECTED	5181	/tmp/orbit-root/linc-bc2-0-2100b1abc9e32
unix	3	[]	STREAM	CONNECTED	5180	
unix	3	[]	STREAM	CONNECTED	5175	/tmp/orbit-root/linc-bc2-0-2100b1abc9e32
unix	3	[]	STREAM	CONNECTED	5174	
unix	3	[]	STREAM	CONNECTED	5173	/tmp/orbit-root/linc-b32-0-77aa27522e803
unix	3	[]	STREAM	CONNECTED	5172	
unix	3	[]	STREAM	CONNECTED	5171	/tmp/orbit-root/linc-bc2-0-2100b1abc9e32
unix	3	[]	STREAM	CONNECTED	5170	
unix	3	[]	STREAM	CONNECTED	5167	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix	3	[]	STREAM	CONNECTED	5166	
unix	3	[]	STREAM	CONNECTED	5161	/tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	5160	

unix	3	[]	STREAM	CONNECTED	5155	/tmp/orbit-root/linc-b99-0-117f39b52a81b
unix	3	[]	STREAM	CONNECTED	5154	
unix	3	[]	STREAM	CONNECTED	5153	/tmp/orbit-root/linc-bc0-0-11256adf933e5
unix	3	[]	STREAM	CONNECTED	5152	
unix	3	[]	STREAM	CONNECTED	5147	/tmp/orbit-root/linc-bc0-0-11256adf933e5
unix	3	[]	STREAM	CONNECTED	5146	
unix	3	[]	STREAM	CONNECTED	5145	/tmp/orbit-root/linc-b32-0-77aa27522e803
unix	3	[]	STREAM	CONNECTED	5144	
unix	3	[]	STREAM	CONNECTED	5143	/tmp/orbit-root/linc-bc0-0-11256adf933e5
unix	3	[]	STREAM	CONNECTED	5142	
unix	3	[]	STREAM	CONNECTED	5139	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix	3	[]	STREAM	CONNECTED	5138	
unix	3	[]	STREAM	CONNECTED	5133	/tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	5132	
unix	3	[]	STREAM	CONNECTED	5126	/tmp/.fam4c7TG3
unix	3	[]	STREAM	CONNECTED	5125	
unix	3	[]	STREAM	CONNECTED	5114	/tmp/orbit-root/linc-ba3-0-40d53a85304dd
unix	3	[]	STREAM	CONNECTED	5113	
unix	3	[]	STREAM	CONNECTED	5110	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix	3	[]	STREAM	CONNECTED	5109	
unix	3	[]	STREAM	CONNECTED	5104	/tmp/.ICE-unix/2789
unix	3	[]	STREAM	CONNECTED	5103	
unix	3	[]	STREAM	CONNECTED	5096	/tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	5095	
unix	3	[]	STREAM	CONNECTED	5089	/tmp/mapping-root
unix	3	[]	STREAM	CONNECTED	5082	
unix	3	[]	STREAM	CONNECTED	5067	/tmp/orbit-root/linc-ba7-0-5307e37356934
unix	3	[]	STREAM	CONNECTED	5066	
unix	3	[]	STREAM	CONNECTED	5065	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix	3	[]	STREAM	CONNECTED	5064	
unix	3	[]	STREAM	CONNECTED	5063	/tmp/.famX0sOwR
unix	3	[]	STREAM	CONNECTED	5062	
unix	3	[]	STREAM	CONNECTED	5053	/tmp/orbit-root/linc-b9b-0-305f3391a425e
unix	3	[]	STREAM	CONNECTED	5052	
unix	3	[]	STREAM	CONNECTED	5051	/tmp/orbit-root/linc-ba7-0-5307e37356934
unix	3	[]	STREAM	CONNECTED	5050	
unix	3	[]	STREAM	CONNECTED	5047	/tmp/orbit-root/linc-ba7-0-5307e37356934
unix	3	[]	STREAM	CONNECTED	5046	

unix	3	[]	STREAM	CONNECTED	5043	/tmp/orbit-root/linc-b32-0-77aa27522e803
unix	3	[]	STREAM	CONNECTED	5042	
unix	3	[]	STREAM	CONNECTED	5033	/tmp/orbit-root/linc-b9b-0-305f3391a425e
unix	3	[]	STREAM	CONNECTED	5032	
unix	3	[]	STREAM	CONNECTED	5031	/tmp/orbit-root/linc-b32-0-77aa27522e803
unix	3	[]	STREAM	CONNECTED	5030	
unix	3	[]	STREAM	CONNECTED	5029	/tmp/.fameDPXFO
unix	3	[]	STREAM	CONNECTED	5028	
unix	3	[]	STREAM	CONNECTED	5018	/var/run/dbus/system_bus_socket
unix	3	[]	STREAM	CONNECTED	5017	
unix	3	[]	STREAM	CONNECTED	5006	/tmp/.ICE-unix/2789
unix	3	[]	STREAM	CONNECTED	5005	
unix	3	[]	STREAM	CONNECTED	5002	/tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	5001	
unix	3	[]	STREAM	CONNECTED	4995	/tmp/orbit-root/linc-b9f-0-162873ebcb1b4
unix	3	[]	STREAM	CONNECTED	4994	
unix	3	[]	STREAM	CONNECTED	4991	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix	3	[]	STREAM	CONNECTED	4990	
unix	3	[]	STREAM	CONNECTED	4985	/tmp/.ICE-unix/2789
unix	3	[]	STREAM	CONNECTED	4984	
unix	3	[]	STREAM	CONNECTED	4979	/tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	4978	
unix	3	[]	STREAM	CONNECTED	4974	/tmp/orbit-root/linc-b9b-0-305f3391a425e
unix	3	[]	STREAM	CONNECTED	4973	
unix	3	[]	STREAM	CONNECTED	4970	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix	3	[]	STREAM	CONNECTED	4969	
unix	3	[]	STREAM	CONNECTED	4964	/tmp/.ICE-unix/2789
unix	3	[]	STREAM	CONNECTED	4963	
unix	3	[]	STREAM	CONNECTED	4958	/tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	4957	
unix	3	[]	STREAM	CONNECTED	4955	/tmp/orbit-root/linc-b9d-0-a71fbb8608d8
unix	3	[]	STREAM	CONNECTED	4954	
unix	3	[]	STREAM	CONNECTED	4951	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix	3	[]	STREAM	CONNECTED	4950	
unix	3	[]	STREAM	CONNECTED	4945	/tmp/.ICE-unix/2789
unix	3	[]	STREAM	CONNECTED	4944	
unix	3	[]	STREAM	CONNECTED	4939	/tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	4938	

unix 3	[]	STREAM	CONNECTED	4936	/tmp/orbit-root/linc-b99-0-117f39b52a81b
unix 3	[]	STREAM	CONNECTED	4935	
unix 3	[]	STREAM	CONNECTED	4934	/tmp/orbit-root/linc-b32-0-77aa27522e803
unix 3	[]	STREAM	CONNECTED	4933	
unix 3	[]	STREAM	CONNECTED	4932	/tmp/orbit-root/linc-b99-0-117f39b52a81b
unix 3	[]	STREAM	CONNECTED	4931	
unix 3	[]	STREAM	CONNECTED	4928	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix 3	[]	STREAM	CONNECTED	4927	
unix 3	[]	STREAM	CONNECTED	4922	/tmp/.ICE-unix/2789
unix 3	[]	STREAM	CONNECTED	4921	
unix 3	[]	STREAM	CONNECTED	4916	/tmp/.X11-unix/X0
unix 3	[]	STREAM	CONNECTED	4915	
unix 3	[]	STREAM	CONNECTED	4903	/tmp/.ICE-unix/2789
unix 3	[]	STREAM	CONNECTED	4902	
unix 3	[]	STREAM	CONNECTED	4901	/tmp/.X11-unix/X0
unix 3	[]	STREAM	CONNECTED	4900	
unix 3	[]	STREAM	CONNECTED	4899	/tmp/orbit-root/linc-b95-0-4a34015bd3d1
unix 3	[]	STREAM	CONNECTED	4898	
unix 3	[]	STREAM	CONNECTED	4895	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix 3	[]	STREAM	CONNECTED	4894	
unix 3	[]	STREAM	CONNECTED	4841	/tmp/orbit-root/linc-b34-0-370fb37e72827
unix 3	[]	STREAM	CONNECTED	4835	
unix 3	[]	STREAM	CONNECTED	4817	/tmp/orbit-root/linc-b34-0-370fb37e72827
unix 3	[]	STREAM	CONNECTED	4816	
unix 3	[]	STREAM	CONNECTED	4815	/tmp/orbit-root/linc-b32-0-77aa27522e803
unix 3	[]	STREAM	CONNECTED	4814	
unix 3	[]	STREAM	CONNECTED	4709	/tmp/.famDjxHmM
unix 3	[]	STREAM	CONNECTED	4708	
unix 3	[]	STREAM	CONNECTED	4690	/tmp/orbit-root/linc-b34-0-370fb37e72827
unix 3	[]	STREAM	CONNECTED	4689	
unix 3	[]	STREAM	CONNECTED	4686	/tmp/orbit-root/linc-b2d-0-3746f073551e9
unix 3	[]	STREAM	CONNECTED	4685	
unix 3	[]	STREAM	CONNECTED	4680	/tmp/.X11-unix/X0
unix 3	[]	STREAM	CONNECTED	4679	
unix 3	[]	STREAM	CONNECTED	4672	/tmp/orbit-root/linc-ae5-0-248e2ca162ecb
unix 3	[]	STREAM	CONNECTED	4671	
unix 3	[]	STREAM	CONNECTED	4670	/tmp/orbit-root/linc-b32-0-77aa27522e803
unix 3	[]	STREAM	CONNECTED	4669	

unix	2	[]	DGRAM	4665	
unix	3	[]	STREAM	CONNECTED	4632 /tmp/orbit-root/linc-ae5-0-248e2ca162ecb
unix	3	[]	STREAM	CONNECTED	4631
unix	3	[]	STREAM	CONNECTED	4630 /tmp/orbit-root/linc-b2d-0-3746f073551e9
unix	3	[]	STREAM	CONNECTED	4507
unix	2	[]	DGRAM	4494	
unix	3	[]	STREAM	CONNECTED	4489 /tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	4488
unix	3	[]	STREAM	CONNECTED	4374 /tmp/.font-unix/fs7100
unix	3	[]	STREAM	CONNECTED	4373
unix	4	[]	STREAM	CONNECTED	4379 /tmp/.X11-unix/X0
unix	3	[]	STREAM	CONNECTED	4356
unix	2	[]	DGRAM	3912	
unix	2	[]	DGRAM	3857	
unix	2	[]	DGRAM	3818	
unix	2	[]	DGRAM	3783	
unix	2	[]	DGRAM	3766	
unix	2	[]	DGRAM	3706	
unix	2	[]	DGRAM	3371	
unix	2	[]	DGRAM	3320	

El resultado que se obtiene tiene una extensión bastante grande, por lo que se vuelca en un fichero por comodidad. Con este comando hemos conseguido ver las conexiones de red visivles desde nuestro equipo, teniendo la infomación más relevante de cada una, como puede ser el tipo (*DGRAM* --> Datagrama; *STREAM* --> Flujo de datos). Como parece obvio, no siempre nos interesa ver tanta cantidad de información, porque entre otras cosas es poco manejable, de forma que se usarán modificadores como los de los siguientes apartados.

Lo verdaderamente interesante es que nos muestra en las primeras líneas la información TCP y/o UDP, que en nuestro caso es la siguiente:

```
tcp    1    0 localhost.localdo:32772 localhost.localdoma:ipp CLOSE_WAIT
```

Se observa que se tiene TCP como portocolo de transporte, para nuestro equipo, que viene denominado por localhost.localdo.

El resto de líneas son los sockets de UNIX que para nosotros tienen menor importancia.

Apartado b

En este apartado usaremos netstat con los modificadores `-antu` para que sólo se nos muestren las conexiones IP. El significado de estos modificadores es el siguiente:

`-a` --> Muestra tanto los sockets que están escuchando como no (LISTEN y los que no).

`-n` --> Hace que las direcciones de las conexiones se muestren de forma numérica (como IPs), en lugar de como los nombres simbólicos de host, puertos o nombres de usuarios.

`-t` --> Muestra las conexiones TCP.

`-u` --> Muestra las conexiones UDP.

De este modo, lanzamos `netstat -antu` y obtenemos:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 0.0.0.0:32769          0.0.0.0:*              LISTEN
tcp      0      0 127.0.0.1:32770       0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:111           0.0.0.0:*              LISTEN
tcp      0      0 127.0.0.1:631         0.0.0.0:*              LISTEN
tcp      0      0 127.0.0.1:25          0.0.0.0:*              LISTEN
tcp      1      0 127.0.0.1:32772       127.0.0.1:631          CLOSE_WAIT
tcp      0      0 :::22                  :::*                    LISTEN
udp      0      0 0.0.0.0:32768          0.0.0.0:*
udp      0      0 0.0.0.0:111           0.0.0.0:*
udp      0      0 0.0.0.0:631           0.0.0.0:*
udp      0      0 0.0.0.0:890           0.0.0.0:*
```

En el texto anterior se observa claramente como se indica el protocolo de los puertos activos (TCP o UDP, que son lo únicos que hemos querido ver, según el comando indicado). Igualmente es posible ver el estado de las conexiones, donde LISTEN indica que se está escuchando por un puerto determinado, indicado en esa misma línea, en la columna Local Address (siguiendo la forma estandarizada de *DirecciónIP:Puerto*). A parte de este estado existen otros, como el CLOSE_WAIT, indicativo de que la conexión está cerrada o a la espera.

Apartado c

En este punto realizamos una conexión a una determinada web, en concreto www.google.com, cuya IP es la 66.102.11.99 (la cual se verá a través del propio netstat). El resultado que se tiene por el netstat al hacer dicha conexión, es el siguiente:

```
Active Internet connections (servers and established)
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:32769	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:32770	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:111	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:631	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN
tcp	0	0	172.16.1.5:32846	66.102.11.99:80	ESTABLISHED
tcp	0	0	172.16.1.5:32847	66.102.11.99:80	ESTABLISHED
tcp	1	0	127.0.0.1:32772	127.0.0.1:631	CLOSE_WAIT
tcp	0	0	:::22	:::*	LISTEN
udp	0	0	0.0.0.0:32768	0.0.0.0:*	
udp	0	0	0.0.0.0:111	0.0.0.0:*	
udp	0	0	0.0.0.0:631	0.0.0.0:*	
udp	0	0	0.0.0.0:890	0.0.0.0:*	

Se puede ver claramente como se han establecido dos conexiones (ESTABLISHED) entre nuestro equipo (IP 172.16.1.5) y el equipo remoto que alberga la página web de www.google.com (IP 66.102.11.99). El hecho de que hayan dos conexiones TCP establecidas es algo propio del funcionamiento de conexión web, es decir, se crea una conexión para el propio navegador en sí, al abrirlo y otra conexión más para cada ventana o pestaña del navegador, lo que quiere decir que el programa principal del navegador que abramos lanzará tantos hilos como ventanas de exploración web se abrán, usando cada una de ellas un puerto distinto; evidentemente, el límite será el número de puertos disponibles. Así, se han tomado los puertos 32846 y 32847 (consecutivos), pues se podrán tomar puertos consecutivos desde el 32846 hasta alguno determinado, y se irán usando todos ellos en función de las necesidades. Por otro lado, la conexión siempre se establece con el puerto 80 de la IP 66.102.11.99, que alberga la web de www.google.com. Esto es debido, a que es el puerto 80 el standard para la conexiones web, para la navegación, pues el puerto para HTTP.

Apartado d

En esta ocasión, sin cerrar la conexión web del apartado anterior, procederemos a realizar una conexión por ftp (que usará el puerto 21, como establece la funcionalidad standard de usos de cada puerto de los equipos). Cuando se haya realizado la conexión ftp tendremos la siguiente salida para el comando netstat:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp      0      0 0.0.0.0:32769          0.0.0.0:*               LISTEN
tcp      0      0 127.0.0.1:32770       0.0.0.0:*               LISTEN
tcp      0      0 0.0.0.0:111           0.0.0.0:*               LISTEN
tcp      0      0 127.0.0.1:631         0.0.0.0:*               LISTEN
tcp      0      0 127.0.0.1:25          0.0.0.0:*               LISTEN
tcp      0      0 172.16.1.5:32846      66.102.11.99:80        TIME_WAIT
tcp      0      0 172.16.1.5:32847      66.102.11.99:80        TIME_WAIT
tcp      0      0 172.16.1.5:32850      130.206.1.5:21         ESTABLISHED
tcp      1      0 127.0.0.1:32772       127.0.0.1:631          CLOSE_WAIT
tcp      0      0 :::22                  :::*                     LISTEN
udp      0      0 0.0.0.0:32768         0.0.0.0:*               *
udp      0      0 0.0.0.0:111           0.0.0.0:*               *
udp      0      0 0.0.0.0:631           0.0.0.0:*               *
udp      0      0 0.0.0.0:890           0.0.0.0:*               *
```

Ahora las conexiones que se establecieron en el punto anterior para el acceso a una página web, se ven en estado TIME_WAIT, pues estamos conectados, pero en un modo de espera.

Por otro lado, en lo referente a la conexión FTP que hemos realizado, se puede ver que se establece una conexión con el puerto 21 de una máquina remota, cuya IP es la 130.206.1.5, correspondiente al equipo que alberga el ftp de www.rediris.es; como se observa, es la IP de un equipo que sirve el ftp y parece lógico que si realizamos otras conexiones ftp con www.rediris.es podría ocurrir que nos conectáramos con otras direcciones IP de diferentes equipos que sirven al ftp de www.rediris.es (por ejemplo, 130.206.1.2, es decir, las que estén dentro del servidor 130.206.1.0). Para dicha conexión, en nuestro equipo (IP 172.16.1.5) se ha asignado el puerto 32850 para la escucha y comunicación con dicho ftp. Como es lógico su estado será el de conexión

establecida (ESTABLISHED).

Tcpdump

A continuación se resuelven los apartados relacionados con el comando tcpdump, según el guión de prácticas. La principal funcionalidad de este comando radica en el hecho de que nos muestra todo el tráfico que hay en la red, durante una sesión del mismo.

Apartado a

Para ver las entradas que tenemos almacenadas en la caché ARP, usamos el comando “arp”. Concretamente hemos usados dos opciones diferentes: “n” que sirve para no ver los nombres sino directamente las Ips (para verlo más claro) y “a” que nos permite ver el resultado en formato BSD y además podemos especificar un host determinado, aunque si se usa sin host muestra todas las entradas, que es lo que nos interesa:

```
[root@pasarela3 root]# arp -n
Address          HWtype  HWaddress          Flags Mask  Iface
172.16.1.1       ether   00:A0:C9:B2:30:20  C           eth1
[root@pasarela3 root]# arp -na
? (172.16.1.1) at 00:A0:C9:B2:30:20 [ether] on eth1
```

Para borrar una entrada, usamos el parámetro “d” seguido con el nombre del host o la IP que queremos borrar, así tenemos:

```
[root@pasarela3 root]# arp -n
Address          HWtype  HWaddress          Flags Mask  Iface
172.16.1.1       ether   00:A0:C9:B2:30:20  C           eth1
[root@pasarela3 root]# arp -d 172.16.1.1
[root@pasarela3 root]# arp -n
Address          HWtype  HWaddress          Flags Mask  Iface
172.16.1.1       ether   (incomplete)      C           eth1
```

Apartado b

Al hacer un ping a la dirección 172.16.1.1, el volcado del tcpdump nos muestra lo siguiente:

```
[root@localhost root]# tcpdump -n -i eth1 \
(src 172.16.1.1 and dst 172.16.1.5) or \
(src 172.16.1.5 and dst 172.16.1.1)
tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
12:14:35.774521 arp who-has 172.16.1.1 tell 172.16.1.5
12:14:35.774595 arp reply 172.16.1.1 is-at 00:a0:c9:b2:30:20
12:14:35.774606 IP 172.16.1.5 > 172.16.1.1: icmp 64: echo request seq 0
12:14:35.774684 IP 172.16.1.1 > 172.16.1.5: icmp 64: echo reply seq 0

4 packets captured
4 packets received by filter
0 packets dropped by kernel
```

Aquí la información obtenida es trivial, así que la desglosaremos:

- 12:14:35.774521 arp who-has 172.16.1.1 tell 172.16.1.5: petición de dirección física (el que tenga la ip 172.16.1.1 que se lo diga a 172.16.1.5).
- 12:14:35.774595 arp reply 172.16.1.1 is-at 00:a0:c9:b2:30:20: respuesta de dirección física (la dirección IP 172.16.1.1 tiene la MAC 00:a0:c9:b2:30:20).
- 12:14:35.774606 IP 172.16.1.5 > 172.16.1.1: icmp 64: echo request seq 0: Petición de ping de 172.16.1.5 a 172.16.1.1.
- 12:14:35.774684 IP 172.16.1.1 > 172.16.1.5: icmp 64: echo reply seq 0: Respuesta de ping de 172.16.1.1 a 172.16.1.5.

Apartado c

Si hacemos de nuevo un ping el volcado es:

```
[root@localhost root]# tcpdump -n -i eth1 \
(src 172.16.1.1 and dst 172.16.1.5) or \
(src 172.16.1.5 and dst 172.16.1.1)
tcpdump: verbose output suppressed, use -v or -vv for full protocol
decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
12:16:22.855328 IP 172.16.1.5 > 172.16.1.1: icmp 64: echo request seq 0
12:16:22.855416 IP 172.16.1.1 > 172.16.1.5: icmp 64: echo reply seq 0

2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

Evidentemente no es el mismo ya que que las dos primeras líneas que veíamos antes, correspondientes a la petición ARP, han desaparecido porque la MAC de la dirección 172.16.1.1 ya la tenemos en caché. Lo comprobamos:

```
[root@pasarela3 root]# arp -n
```

Address	HWtype	HWaddress	Flags Mask	Iface
172.16.1.1	ether	00:A0:C9:B2:30:20	C	eth1

Apartado d

En este apartado hicimos una conexión FTP al servidor del laboratorio (172.16.1.1) y comprobamos lo que salía por el tcpdump. La conexión FTP fue:

```
lftp :~> open 172.16.1.1
lftp 172.16.1.1:/> debug 9
lftp 172.16.1.1:/> ls
---- Conectándose a 172.16.1.1 (172.16.1.1) port 21
<--- 220 ProFTPD 1.2.8 Server (ProFTPD del Laboratorio de Redes) [triton.dis.ulpgc.es]
---> USER anonymous
<--- 331 Anonymous login ok, send your complete email address as your password.
---> PASS lftp@
<--- 230 Anonymous access granted, restrictions apply.
---> PRET LIST
<--- 500 PRET not understood
---> PASV
<--- 227 Entering Passive Mode (172,16,1,1,135,14).
---- Conectando socket de datos a (172.16.1.1) puerto 34574
---> LIST
<--- 150 Opening ASCII mode data connection for file list
---- Cerrando socket de datos
<--- 226 Transfer complete.
drwxrwx--- 6 ftp ftp 4096 Nov 9 10:21 FedoraCore2
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 ParcheBlaster
drwxrwx--- 3 ftp ftp 4096 Oct 29 11:10 Placa_Red
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 PureFTP
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 asad
drwxrwx--- 4 ftp ftp 4096 Oct 29 11:10 d-link
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 diald
drwxrwx--- 5 ftp ftp 4096 Oct 29 11:10 garc
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 libpcap
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 manuales
```

```
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 mgetty
drwxrwx--- 7 ftp ftp 4096 Oct 29 11:10 modems
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 postfix
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 ppp
drwxrwx--- 3 ftp ftp 4096 Oct 29 11:49 proftpd
drwxrwx--- 9 ftp ftp 4096 Oct 29 11:10 rfc
drwxrwx--- 2 ftp ftp 4096 Nov 10 11:02 seguridad
drwxrwx--- 3 ftp ftp 4096 Oct 29 11:10 software
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 ssh
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 windump
drwxrwx--- 2 ftp ftp 4096 Oct 29 11:10 wireless
lftp 172.16.1.1:/> close
---> QUIT
<--- 221 Goodbye.
---- Cerrando socket de control
lftp 172.16.1.1:/>
```

Por otro lado, la salida del tcpdump ya dividida fue:

Conexion

```
12:29:03.903439 IP 172.16.1.5.32865 > 172.16.1.1.ftp: S 2784293344:2784293344(0) win 5840
<mss 1460,sackOK,timestamp 3711120 0,nop,wscale 0>
12:29:03.903461 IP 172.16.1.1.ftp > 172.16.1.5.32865: S 275609097:275609097(0) ack
2784293345 win 5792 <mss 1460,sackOK,timestamp 530422183 3711120,nop,wscale 0>
12:29:03.903483 IP 172.16.1.5.32865 > 172.16.1.1.ftp: . ack 1 win 5840 <nop,nop,timestamp
3711121 530422183>
12:29:03.904652 IP 172.16.1.1.ftp > 172.16.1.5.32865: P 1:84(83) ack 1 win 5792
<nop,nop,timestamp 530422185 3711121>
12:29:03.904669 IP 172.16.1.5.32865 > 172.16.1.1.ftp: . ack 84 win 5840 <nop,nop,timestamp
3711122 530422185>
12:29:03.906555 IP 172.16.1.5.32865 > 172.16.1.1.ftp: P 1:17(16) ack 84 win 5840
<nop,nop,timestamp 3711124 530422185>
```

12:29:03.906629 IP 172.16.1.1.ftp > 172.16.1.5.32865: . ack 17 win 5792 <nop,nop,timestamp
530422187 3711124>

12:29:03.906835 IP 172.16.1.1.ftp > 172.16.1.5.32865: P 84:160(76) ack 17 win 5792
<nop,nop,timestamp 530422187 3711124>

12:29:03.908692 IP 172.16.1.5.32865 > 172.16.1.1.ftp: P 17:29(12) ack 160 win 5840
<nop,nop,timestamp 3711126 530422187>

12:29:03.909656 IP 172.16.1.1.ftp > 172.16.1.5.32865: P 160:211(51) ack 29 win 5792
<nop,nop,timestamp 530422190 3711126>

12:29:03.912563 IP 172.16.1.5.32865 > 172.16.1.1.ftp: P 29:41(12) ack 211 win 5840
<nop,nop,timestamp 3711130 530422190>

12:29:03.912689 IP 172.16.1.1.ftp > 172.16.1.5.32865: P 211:236(25) ack 41 win 5792
<nop,nop,timestamp 530422193 3711130>

12:29:03.922490 IP 172.16.1.5.32865 > 172.16.1.1.ftp: P 41:47(6) ack 236 win 5840
<nop,nop,timestamp 3711140 530422193>

12:29:03.922684 IP 172.16.1.1.ftp > 172.16.1.5.32865: P 236:284(48) ack 47 win 5792
<nop,nop,timestamp 530422203 3711140>

Respuesta LS

12:29:03.925178 IP 172.16.1.5.32866 > 172.16.1.1.34574: S 2786949295:2786949295(0) win 5840
<mss 1460,sackOK,timestamp 3711142 0,nop,wscale 0>

12:29:03.925231 IP 172.16.1.5.32865 > 172.16.1.1.ftp: P 47:53(6) ack 284 win 5840
<nop,nop,timestamp 3711142 530422203>

12:29:03.925267 IP 172.16.1.1.34574 > 172.16.1.5.32866: S 276088736:276088736(0) ack
2786949296 win 5792 <mss 1460,sackOK,timestamp 530422205 3711142,nop,wscale 0>

12:29:03.925287 IP 172.16.1.5.32866 > 172.16.1.1.34574: . ack 1 win 5840 <nop,nop,timestamp
3711142 530422205>

12:29:03.925477 IP 172.16.1.1.ftp > 172.16.1.5.32865: P 284:338(54) ack 53 win 5792
<nop,nop,timestamp 530422205 3711142>

12:29:03.927028 IP 172.16.1.1.34574 > 172.16.1.5.32866: P 1:1019(1018) ack 1 win 5792
<nop,nop,timestamp 530422207 3711142>

12:29:03.927046 IP 172.16.1.5.32866 > 172.16.1.1.34574: . ack 1019 win 7126
<nop,nop,timestamp 3711144 530422207>

```
12:29:03.927061 IP 172.16.1.1.34574 > 172.16.1.5.32866: FP 1019:1339(320) ack 1 win 5792
<nop,nop,timestamp 530422207 3711142>
12:29:03.927287 IP 172.16.1.5.32866 > 172.16.1.1.34574: F 1:1(0) ack 1340 win 9162
<nop,nop,timestamp 3711144 530422207>
12:29:03.927357 IP 172.16.1.1.34574 > 172.16.1.5.32866: . ack 2 win 5792 <nop,nop,timestamp
530422207 3711144>
12:29:03.927481 IP 172.16.1.1.ftp > 172.16.1.5.32865: P 338:362(24) ack 53 win 5792
<nop,nop,timestamp 530422207 3711142>
12:29:03.965428 IP 172.16.1.5.32865 > 172.16.1.1.ftp: . ack 362 win 5840 <nop,nop,timestamp
3711183 530422205>
```

Close

```
12:29:06.863054 IP 172.16.1.5.32865 > 172.16.1.1.ftp: P 53:59(6) ack 362 win 5840
<nop,nop,timestamp 3714081 530422205>
12:29:06.863263 IP 172.16.1.1.ftp > 172.16.1.5.32865: P 362:376(14) ack 59 win 5792
<nop,nop,timestamp 530425143 3714081>
12:29:06.863283 IP 172.16.1.5.32865 > 172.16.1.1.ftp: . ack 376 win 5840 <nop,nop,timestamp
3714081 530425143>
12:29:06.863336 IP 172.16.1.1.ftp > 172.16.1.5.32865: F 376:376(0) ack 59 win 5792
<nop,nop,timestamp 530425144 3714081>
12:29:06.863598 IP 172.16.1.5.32865 > 172.16.1.1.ftp: F 59:59(0) ack 377 win 5840
<nop,nop,timestamp 3714081 530425144>
12:29:06.863674 IP 172.16.1.1.ftp > 172.16.1.5.32865: . ack 60 win 5792 <nop,nop,timestamp
530425144 3714081>
```

Hemos dividido en esas tres partes los resultados debido a los siguientes conocimientos:

- Las transacciones de datos van por otro puerto que se negocia
- Entre el “ls” y el close hay aproximadamente 3 segundos de diferencia
- El “lftp” no hace login hasta que no se le hace la primera petición

Podemos ver claramente como la segunda parte es donde por primera vez se usan puertos

diferentes en el intercambio de paquetes, posiblemente son la respuesta a nuestra petición “ls”.

Podemos observar también como existe una diferencia de 3 segundos entre el último paquete de la segunda parte con el primer paquete de la tercera, lo que nos indica el inicio de la transacción para el “close”.

Traceroute

El comando traceroute nos permite observar la ruta que sigue un paquete IP desde la dirección origen desde donde se envía (nuestro equipo) hasta la dirección destino. De este modo, veremos todos los enrutadores por los que pasa, mostrándonos las Ip de los mismos y en su caso sus nombres.

A continuación realizaremos dos conexiones diferentes para ver los resultados que nos ofrece el comando traceroute, así como la salida de tcpdump durante la ejecución de un traceroute, con la finalidad de ver el tráfico que éste genera.

Por otro lado, se mencionarán algunos de los modificadores más útiles del comando traceroute, los cuales han sido utilizados en las pruebas que se verán a continuación.

Conexión con www.ulpgc.es

De este modo, empezaremos lanzando el siguiente comando:

```
traceroute -f 1 -m 120 www.ulpgc.es
```

Se han usado dos modificadores, cuya explicación es la siguiente (si accedemos al manual del comando traceroute, con man traceroute podremos ver el formato general de éste comando y todos sus modificadores):

-f --> Indica cual es el primer paquete a testear con el *ttl* (time to live), es decir, de los saltos que haga el paquete que mandemos, por los sucesivos routers que salte, sólo se mostrará a partir del que indiquemos para este comando; al poner *-f 1* se mostrará desde el número 1, es decir, desde el primero de los paquetes (en consecuencia, se mostrarán todos).

-m --> Establece el *ttl* (time to live) máximo. Su valor por defecto es de 64,

Así, tras lanzar este comando su resultado será:

```
traceroute to web2.ulpgc.es (193.145.138.27), 120 hops max, 38 byte packets
 1 servidor (172.16.1.1) 0.122 ms 0.103 ms 0.101 ms
 2 INF-INFORMATICA1PUB-RC1.red.ulpgc.es (193.145.147.2) 0.319 ms 5.021 ms 0.309 ms
```

```
3 web2.ulpgc.es (193.145.138.27) 0.476 ms 0.400 ms 0.400 ms
```

Por otro lado, lo que veremos como salida en el tcpdump será lo siguiente, al ejecutar el comando traceroute.

```
11:13:01.164313 IP 172.16.1.3.32794 > 172.16.1.1.domain: 50241+ A? www.ulpgc.es. (30)
11:13:01.164511 IP 172.16.1.1.domain > 172.16.1.3.32794: 50241 2/6/2 CNAME web2.ulpgc.es., (217)
11:13:01.164916 IP 172.16.1.1 > 172.16.1.3: icmp 46: time exceeded in-transit
11:13:01.165084 IP 172.16.1.3.32795 > 172.16.1.1.domain: 50242+ PTR? 1.1.16.172.in-addr.arpa. (41)
11:13:01.165253 IP 172.16.1.1.domain > 172.16.1.3.32795: 50242* 1/1/0 PTR[.domain]
11:13:01.165469 IP 172.16.1.1 > 172.16.1.3: icmp 46: time exceeded in-transit
11:13:01.165632 IP 172.16.1.1 > 172.16.1.3: icmp 46: time exceeded in-transit
11:13:01.188768 IP 172.16.1.3.32795 > 172.16.1.1.domain: 50243+ PTR? 2.147.145.193.in-addr.arpa.(44)
11:13:01.188990 IP 172.16.1.1.domain > 172.16.1.3.32795: 50243 1/6/2 (246)
11:13:01.190701 IP 172.16.1.3.32795 > 172.16.1.1.domain: 50244+ PTR? 27.138.145.193.in-addr.arpa.(45)
11:13:01.190925 IP 172.16.1.1.domain > 172.16.1.3.32795: 50244 1/6/2 (224)
```

En la salida del comando traceroute se pueden ver los saltos por routers antes de alcanzar la dirección IP destino; en este caso son 3 saltos:

1. Se manda el paquete IP al servidor del Laboratorio (*servidor (172.16.1.1)*).
2. Desde el servidor se manda a *INF-INFORMATICA1PUB-RC1.red.ulpgc.es (193.145.147.2)*, que sería uno de los equipos servidores para la red de la ulpgc.
3. Finalmente se salta a la IP de destino: *web2.ulpgc.es (193.145.138.27)*. Se observa que en realidad, para la web www.ulpgc.es, su IP es la *193.145.138.27*, con un nombre que realmente es *web2.ulpgc.es*, siendo el nombre www.ulpgc.es un alias suyo. De hecho en la cabecera del traceroute aparece la IP y nombre que son el verdadero destino (*traceroute to web2.ulpgc.es (193.145.138.27)*).

En cuanto a los tiempos que aparecen para cada router al que se salta, se tratan de tiempos de acceso a la IP del router en cuestión, relativos al salto. Se trata de un terceto de tres valores de tiempos (por lo general en milisegundos), que permiten ver la velocidad a la que se transmite el datagrama IP que hemos mandado, a través del enrutamiento que se le aplica para alcanzar el destino.

Finalmente, podemos analizar la salida que obtenemos por el tcpdump. Ésta nos da información relevante al tráfico que surge debido al trazo de la ruta. En dicha información podemos ver como la comunicación se produce realmente, siempre entre nuestro equipo (IP 172.16.1.3) y el servidor del Laboratorio (IP 172.16.1.1), pues el que es visible para nosotros. Será, entonces, el servidor quien se encarga de pasar (a modo de pasarela) por nosotros las IP que van a Internet, así como enrutar hacia nosotros los datagramas que se nos envían a nuestro equipo desde Internet (o simplemente fuera del Laboratorio). Igualmente se puede ver como se usan los puertos 32794 y 32795 para dicho trazo, para el envío del datagrama y obtención de información de la ruta, respectivamente.

Conexión con *www.rediris.es*

Ahora hacemos un trazo hacia una ruta más alejada que la anterior. De este modo, el resultado para el traceroute (*traceroute www.rediris.es*; en esta ocasión sin modificadores, si bien será muy similar, pues los valores por defecto coinciden aproximadamente para los indicados en el caso anterior) es el siguiente:

```
traceroute to sun.rediris.es (130.206.1.2), 30 hops max, 38 byte packets
 1 servidor (172.16.1.1) 0.154 ms 0.119 ms 0.105 ms
 2 INF-INFORMATICA1PUB-RC1.red.ulpgc.es (193.145.147.2) 0.547 ms 0.343 ms 0.448 ms
 3 172.20.0.253 (172.20.0.253) 1.498 ms 1.384 ms 2.107 ms
 4 130.206.198.137 (130.206.198.137) 26.239 ms 2.768 ms 2.700 ms
 5 CAN-L.AT0-0-0-2.EB-Sevilla0.red.rediris.es (130.206.240.121) 20.847 ms 19.506 ms 26.651 ms
 6 AND.SO4-1-0.EB-IRIS2.red.rediris.es (130.206.240.17) 30.370 ms 31.996 ms 30.416 ms
 7 130.206.220.66 (130.206.220.66) 30.093 ms 30.104 ms 30.058 ms
 8 sun.rediris.es (130.206.1.2) 30.411 ms 30.893 ms 29.974 ms
```

Por el tcpdump se observará los siguiente:

```
11:36:53.925265 IP 172.16.1.3.32801 > 172.16.1.1.domain: 23660+ A? www.rediris.es. (32)
11:36:53.925479 IP 172.16.1.1.domain > 172.16.1.3.32801: 23660 2/3/1 CNAME sun.rediris.es., (135)
11:36:53.952502 IP 172.16.1.1 > 172.16.1.3: icmp 46: time exceeded in-transit
11:36:53.952690 IP 172.16.1.3.32802 > 172.16.1.1.domain: 23661+ PTR? 1.1.16.172.in-addr.arpa. (41)
11:36:53.952854 IP 172.16.1.1.domain > 172.16.1.3.32802: 23661* 1/1/0 PTR[.domain]
11:36:53.953111 IP 172.16.1.1 > 172.16.1.3: icmp 46: time exceeded in-transit
11:36:53.953284 IP 172.16.1.1 > 172.16.1.3: icmp 46: time exceeded in-transit
11:36:53.983345 IP 172.16.1.3.32802 > 172.16.1.1.domain: 23662+ PTR? 2.147.145.193.in-addr.arpa. (44)
```

```
11:36:53.983572 IP 172.16.1.1.domain > 172.16.1.3.32802: 23662 1/6/2 (246)
11:36:54.042274 IP 172.16.1.3.32802 > 172.16.1.1.domain: 23663+ PTR? 253.0.20.172.in-addr.arpa. (43)
11:36:54.042417 IP 172.16.1.1.domain > 172.16.1.3.32802: 23663 NXDomain 0/1/0 (120)
11:36:54.073437 IP 172.16.1.3.32802 > 172.16.1.1.domain: 23664+ PTR? 137.198.206.130.in-addr.arpa. (46)
11:36:54.073578 IP 172.16.1.1.domain > 172.16.1.3.32802: 23664 NXDomain 0/1/0 (107)
11:36:54.100412 IP 172.16.1.3.32802 > 172.16.1.1.domain: 23665+ PTR? 121.240.206.130.in-addr.arpa. (46)
11:36:54.100590 IP 172.16.1.1.domain > 172.16.1.3.32802: 23665 1/2/2 (172)
11:36:54.185470 IP 172.16.1.3.32802 > 172.16.1.1.domain: 23666+ PTR? 17.240.206.130.in-addr.arpa. (45)
11:36:54.185645 IP 172.16.1.1.domain > 172.16.1.3.32802: 23666 1/2/2 (164)
11:36:54.278609 IP 172.16.1.3.32802 > 172.16.1.1.domain: 23667+ PTR? 66.220.206.130.in-addr.arpa. (45)
11:36:54.278750 IP 172.16.1.1.domain > 172.16.1.3.32802: 23667 NXDomain 0/1/0 (106)
11:36:54.369596 IP 172.16.1.3.32802 > 172.16.1.1.domain: 23668+ PTR? 2.1.206.130.in-addr.arpa. (42)
11:36:54.369810 IP 172.16.1.1.domain > 172.16.1.3.32802: 23668 1/3/2 PTR[domain]
```

Como la IP de este caso está más alejada, el número de saltos será lógicamente mayor (de 8), pero al igual que en el caso anterior, el primer salto será sobre el servidor del Laboratorio (*servidor* (172.16.1.1)). Lo mismo ocurre con el segundo salto, que nos conecta con el servidor del DIS (*INFORMATICAI*PUB-RCI.red.ulpgc.es (193.145.147.2)), lo cual nos permite el acceso al exterior/Internet. Desde ahí en adelante los saltos y enrutamiento se harán en función de las necesidades para alcanzar la IP indicada como destino, lo cual dependerá de las tablas de enrutamiento de los routers por los que salte el datagrama IP que hemos mandado.

Como curiosidad, puede verse que en el salto número 5 atravesamos el router de Sevilla, que se encuentra en *CAN-L.AT0-0-0-2.EB-Sevilla0.red.rediris.es* (130.206.240.121). Desde ahí se siguen produciendo saltos, hasta por fin llegar al destino: *sun.rediris.es* (130.206.1.2).

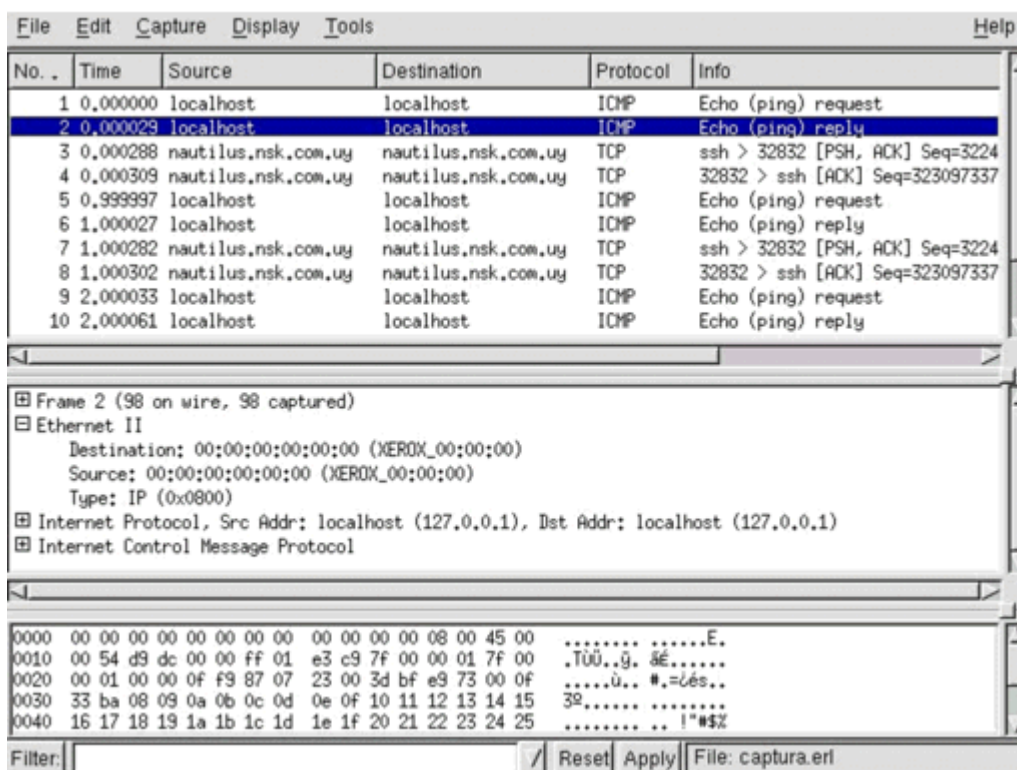
En cuanto a los datos que nos ofrece el *tcpdump*, podemos decir que se ve al igual que en el caso anterior que la comunicación es entre nuestro equipo (IP 172.16.1.3) y el servidor del Laboratorio (IP 172.16.1.1), aunque en este caso se han asignado los puertos 32801 y 32802, a diferencia del caso anterior.

Ethereal

Ethereal es un analizador de protocolos con interfaz gráfica capaz de reconocer muchos protocolos distintos. Permite tanto revisar los paquetes de datos en una red activa como desde un archivo de captura previamente generado; es capaz de comprender diversos formatos de archivo propios de otros programas de captura, en particular el clásico *tcpdump*.

Manejo de la herramienta gráfica Ethereal

El programa Ethereal puede hacerse a través del menú de invocación del entorno gráfico o desde una terminal Unix si no existe la opción en el menú. Si se hace a través de una terminal Unix, el comando *ethereal &* (ejecución en segundo plano) arranca el programa y devuelve el control de la terminal al usuario para poder continuar ingresando comandos. El símbolo *&* arranca el programa como proceso independiente de la terminal. Inicialmente, la ventana principal del programa aparece vacía, si bien luego mostrará las capturas de datos que circulen por la red.



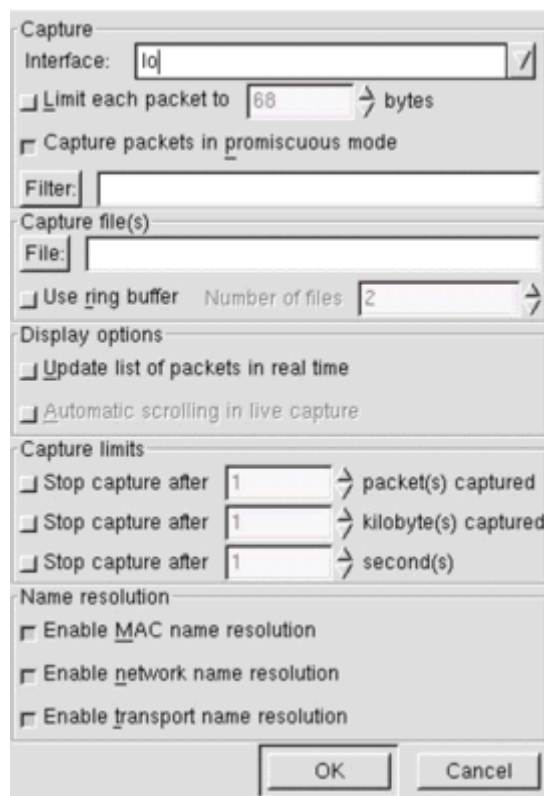
En la ventana principal de Ethereal se reconocen tres áreas de despliegue:

1. Resumen de paquetes capturados, un paquete por línea; uno de ellos ha sido seleccionado

como paquete actual (dando clic sobre la línea del paquete). Al desplazarse en la lista y cambiar el paquete actual se actualizan las otras dos ventanas, donde se despliega en dos formatos diferentes el contenido del paquete.

2. Detalles de encabezado de protocolos para el paquete seleccionado; los encabezados pueden abrirse (clic en +) para ver mayor detalle, o cerrarse (clic en -) para ocupar sólo una línea.
3. Datos crudos del paquete, representación hexadecimal y ASCII del encabezado del paquete seleccionado en el campo del medio.

Para iniciar la captura de datos, elegir las opciones de menú Capture: Start (Capturar, Comienzo). En la ventana de opciones de captura (ver figura), debe fijarse al menos la interfaz sobre la que se quiere realizar la captura. Los nombres varían según los sistemas operativos; la interfaz lo (loopback) permite enviar y recibir paquetes en la propia máquina.



Para capturar en un archivo debe indicarse su nombre en el cuadro "Capture file(s)" de la ventana de Opciones de Captura (Capture: Start abre esta ventana). Estos archivos pueden ser

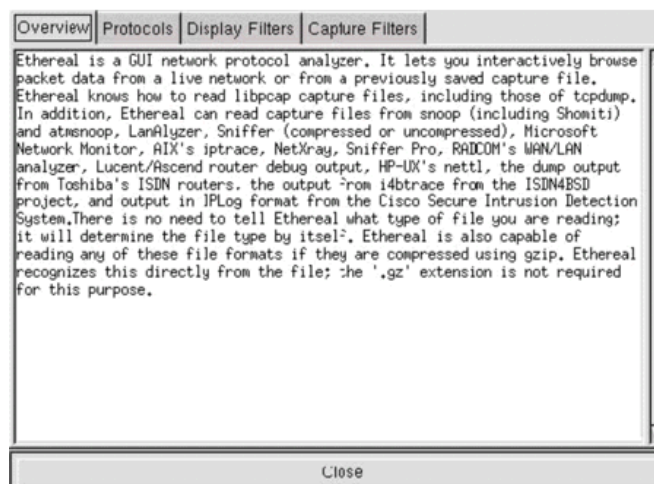
examinados luego con el propio Ethereal mediante la opción de menú File: Open. El tráfico ya capturado puede grabarse en un archivo eligiendo File: Print (Archivo: Imprimir); esta opción graba en formato legible (texto).

La ventana de estado muestra en tiempo real la cantidad de paquetes capturados, en total y de algunos tipos corrientes. La situación de captura se mantiene hasta que se presiona el botón Stop. Luego de unos instantes aparecen los paquetes capturados, tal cual se ve en la imagen de la ventana principal. Si se activó la opción de actualizar lista de paquetes en tiempo real ("Update list of packets in real time") estos se visualizan a medida que son capturados.

Total	8	(100,0%)
SCTP	0	(0,0%)
TCP	2	(25,0%)
UDP	0	(0,0%)
ICMP	6	(75,0%)
OSPF	0	(0,0%)
GRE	0	(0,0%)
NetBIOS	0	(0,0%)
IPX	0	(0,0%)
VINES	0	(0,0%)
Other	0	(0,0%)

Stop

Parte de la información hasta ahora vista puede verse y contrastarse con la documentación de Ethereal. La ventana de ayuda da una reseña del programa (Overview), lista los protocolos reconocidos, lista los nombres de los filtros posibles (Display filters) y refiere a la página man de tcpdump para la sintaxis de filtros de captura (Capture filters); la sintaxis de filtrado en la captura es diferente de la sintaxis de filtrado en el despliegue.



Modificadores a la hora de invocar Ethereal

En Unix, la página man de ethereal contiene sobre las opciones de invocación del programa cuando se lo arranca de la línea de comando. Algunas de las más usadas son:

-h --> muestra versión y opciones

-f <expresión> --> expresión filtro de captura (sintaxis de tcpdump)

-i <interfaz> --> nombre de interfaz de escucha, tal como es mostrada en la salida de los comandos Unix ifconfig -a o netstat -i.

-n --> deshabilita resolución de nombres (tales como nombres de máquinas y nombres de puertos TCP y UDP). Esta opción es útil para evitar ver el intercambio de paquetes originados en las consultas al servidor DNS para resolver nombres de máquinas.

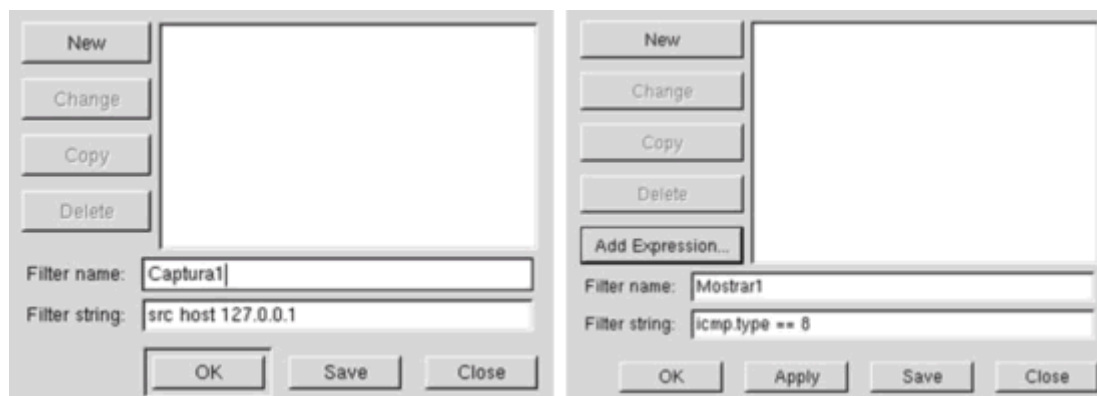
-r <archivo> --> lee los paquetes del archivo indicado en lugar de realizar una captura. El archivo debe haber sido generado con Ethereal, tcpdump o algún otro analizador que use el mismo formato.

-w <archivo> --> fija el nombre del archivo de captura.

Filtrado de Paquetes

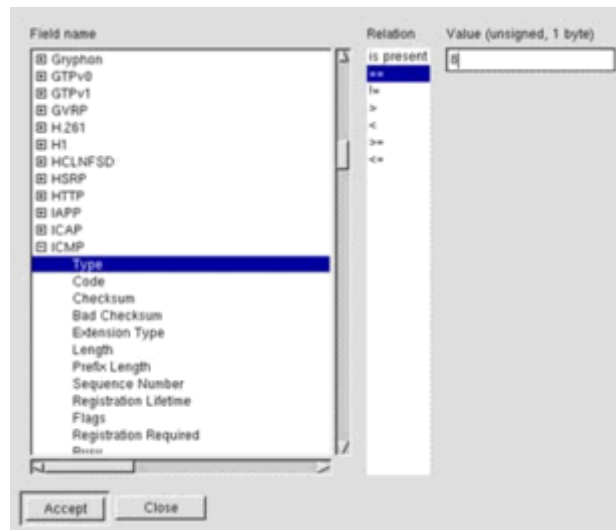
El filtrado de paquetes permite capturar o desplegar sólo aquellos paquetes de interés para el estudio en curso, desconociendo la existencia de otros. Ethereal tiene dos modos de filtro distintos:

1. **Filtro de captura:** sólo se retienen los paquetes que cumplen la expresión filtro. Define lo que se guarda.
2. **Filtro de despliegue:** de los paquetes capturados, sólo se muestran los paquetes que cumplen la expresión filtro. Define lo que se ve de lo que hay guardado.



La sintaxis de escritura de ambos tipos de filtro es diferente. Los filtros de captura siguen la sintaxis del comando tcpdump y deben ser escritos en el cuadro Filter de la ventana de opciones de captura, antes de iniciar la captura. Los filtros de despliegue se fijan en el cuadro File de la ventana principal de Ethereal. En ambos casos, presionando este botón File aparece un cuadro de diálogo que permite asignar un nombre a la expresión filtro construída.

Para los filtros de despliegue existe una ayuda adicional en el botón Add Expression, que permite construir la expresión eligiendo el protocolo, sus campos y operadores relacionales.



En cuanto a los filtros de captura puede decirse que un filtrado de paquetes por tipo puede lograrse en forma muy simple escribiendo el tipo en el cuadro Filter de la ventana principal: tcp, icmp, udp, etc. Para construcción de expresiones de filtro de captura se dispone de palabras claves (src, dst, host, net, len, ...), operadores lógicos (and, or, not) y paréntesis. Los paquetes capturados deberán cumplir con la expresión booleana construída. La página man de tcpdump muestra todas las opciones.

Referencias Bibliográficas

- Ethereal - Network protocol analyzer (<http://www.ethereal.com>), versión para Linux. Ayuda en línea y página man. Manual de usuario: <http://www.ethereal.com/docs/user-guide>
- tcpdump, página man.
- The Internet Lab Manual. Introducción:
http://www.cs.virginia.edu/~itlab/book/pdf/Introduction_v6c.pdf
- Guía de Administración de Redes de Linux (disponible en la web de la asignatura de RC).