

Detección de problemas. Monitorización

Estos materiales se licencian bajo la «Creative Commons Reconocimiento-CompartirIgual License España». Para ver una copia de esta licencia, se puede visitar <http://creativecommons.org/licenses/by-sa/3.0/es/>

Autores:

- Pablo Boronat Pérez (Universitat Jaume I)
- Miguel Pérez Francisco (Universitat Jaume I)
- David Rubert Viana (Universitat Jaume I)

Introducción

La red libre evoluciona continuamente, cada día que pasa nacen nuevos nodos o supernodos, y ante un sistema tan dinámico como éste, disponer de herramientas para poder encontrar y diagnosticar posibles problemas es imprescindible.

La detección y resolución de problemas es una labor compleja, y que se aprende únicamente en base a la experiencia. Veremos en este tema cómo utilizar diferentes herramientas de diagnóstico de red para detectar problemas, o incluso preverlos antes de que sucedan.

Logs. Linux, AirOS i RouterOS

Los archivos de log de nuestro sistema nos proporcionan información histórica y en tiempo real sobre lo que está pasando. Es muy importante revisarlos en el momento en que sospechemos que algo no funciona bien, ya que cualquier cosa que pase en el sistema se debería registrar allí.

Veamos cómo activar y procesar los logs en función del sistema en el que estemos trabajando.

AirOS

AirOS es un sistema operativo que no engloba toda la complejidad de otros sistemas como puede ser Linux o RouterOS, por lo que los logs se activan y procesan de una manera sencilla, dandonos a su vez una información limitada al buen funcionamiento de la antena.

Activación de los logs

MAIN WIRELESS NETWORK ADVANCED SERVICES SYSTEM Tools: ▾ Logout

Ping Watchdog

Enable Ping Watchdog:

IP Address To Ping:

Ping Interval: seconds

Startup Delay: seconds

Failure Count To Reboot:

SNMP Agent

Enable SNMP Agent:

SNMP Community:

Contact:

Location:

Web Server

Use Secure Connection (HTTPS):

Secure Server Port:

Server Port:

Session Timeout: minutes

SSH Server

Enable SSH Server:

Server Port:

Enable Password Authentication:

Authorized Keys:

Telnet Server

Enable Telnet Server:

Server Port:

NTP Client

Enable NTP Client:

NTP Server:

Dynamic DNS

Enable Dynamic DNS:

Host Name:

Username:

Password:

System Log

Enable Log:

Enable Remote Log:

Remote Log IP Address:

Remote Log Port:

Procesado de los logs

RouterOS

En RouterOS se pueden consultar los logs del sistema en el apartado `/log`

```
08:10:07 wireless,info wlan4: data from unknown device
00:0C:42:61:86:69, sent deauth
08:10:07 wireless,info wlan4: data from unknown device
00:0C:42:61:86:69, sent deauth
08:10:07 wireless,info wlan4: data from unknown device
00:0C:42:61:86:69, sent deauth
08:10:07 wireless,info 00:0C:42:61:86:69@wlan4: connected, is AP,
wants WDS
08:10:07 route,ospf,info Discarding packet: no neighbor with this
source address
08:10:07 route,ospf,info      RouterId=10.228.133.97
08:10:07 route,ospf,info      source=172.16.107.37
08:10:10 route,ospf,info Discarding packet: no neighbor with this
source address
08:10:10 route,ospf,info      RouterId=10.228.133.97
08:10:10 route,ospf,info      source=172.16.107.37
08:10:11 route,ospf,info Discarding packet: no neighbor with this
source address
08:10:11 route,ospf,info      RouterId=10.228.133.97
08:10:11 route,ospf,info      source=172.16.107.37
08:13:27 wireless,info 00:15:6D:D2:4F:33@wlan2: reassociating
08:13:27 wireless,info 00:15:6D:D2:4F:33@wlan2: disconnected, ok
08:13:27 wireless,info 00:15:6D:D2:4F:33@wlan2: connected
08:16:14 wireless,info 00:15:6D:D2:4F:33@wlan2: reassociating
08:16:14 wireless,info 00:15:6D:D2:4F:33@wlan2: disconnected, ok
...
...
```

Se pueden consultar las acciones que se están registrando en el apartado /system logging

```
[admin@CS-UJI-BiblAP] /system logging> /system logging print
Flags: X - disabled, I - invalid
#   TOPICS                      ACTION
PREFIX
0   info                         memory
1   error                         memory
2   warning                        memory
3   critical                        echo
```

Para añadir que se registren nuevas acciones se puede ejecutar la orden

```
[admin@CS-UJI-BiblAP] /system logging> /system logging add
action=memory topics=ospf
[admin@CS-UJI-BiblAP] /system logging> /system logging print
Flags: X - disabled, I - invalid
#   TOPICS                      ACTION
PREFIX
0   info                         memory
1   error                         memory
```

```
2    warning           memory
3    critical          echo
4    ospf              memory
y para eliminarlas
/system logging remove 4
```

Linux

Como se ha comentado en el tema 5 GNU/Linux dispone de un potente sistema de logs que almacena los mensajes generados por las aplicaciones. Con cada mensaje se almacena qué programa lo generó, la prioridad y la fecha y hora en que se produjo.

Los ficheros de log en un sistema linux, se encuentran habitualmente en el directorio /var/log o en algún directorio dentro de éste.

El sistema de logs arranca con el script /etc/init.d/sysklogd, el duende que realiza los registros es syslogd que se configura mediante el fichero /etc/syslog.conf, donde se indica qué se quiere registrar y dónde se deben enviar los logs.

Los archivos más importantes son:

- /var/log/messages: donde se almacenan todos los mensajes con prioridad info (información), notice (notificación) o warn (aviso). Es uno de los ficheros en los que primero se mira cuando hay algún problema.
- /var/log/kern.log: almacena los logs del kernel.
- /var/log/dmesg: almacena la información que genera el kernel durante el arranque del sistema. Se puede ver su contenido con la orden dmesg.

Para ver el contenido (total o parcial) de alguno de estos ficheros se pueden utilizar alguna de las siguientes ordenes:

`cat /var/log/messages`

muestra el contenido completo del fichero /var/log/messages.

`less /var/log/messages`

muestra el contenido completo del fichero /var/log/messages paginando la salida (la tecla q permite finalizar sin mostrar todo el fichero).

`tail -50 /var/log/messages`

muestra las últimas 50 líneas del fichero /var/log/messages.

Los archivos de log suelen crecer mucho ya que en ellos se está guardando información continuamente. Por ello, existe una aplicación, logrotate (que se puede configurar a través del fichero en /etc/logrotate) que, si los ficheros de log son muy grandes, los comprime y aplica una rotación a los archivos (añadiéndoles la extensión .1.gz, .2.gz, etc.), volviendo a crear uno vacío (cuanto mayor es el número más antiguo es el log).

Existen aplicaciones gráficas para supervisar los logs, por ejemplo GNOME-System-Log, KSystemLog, Xlogmaster y Xwatch.

GNOME-System-Log

The screenshot shows the 'messages' tab of the GNOME System Log application. The left sidebar lists various log files: dmesg.0, dpkg.log, dpkg.log.1, fontconfig.log, kern.log, kern.log.1, lpr.log, lpr.log.1, mail.err, mail.info, mail.log, mail.warn, messages, messages.1, pm-powersave.log, pm-powersave.log.1, pm-suspend.log, pm-suspend.log.1, popularity-contest, and popularity-contest.0. The main pane displays a log entry from 'coscoll kernel' at 09:12:47 on July 5, 2011. The log message is: 'Jul 5 09:12:47 coscoll kernel: [1.002430] rtc_cmos 00:03: setting system clock'. The entire log window has a light blue background.

```
messages - Visor de sucesos del sistema (como superusuario)
Archivo Editar Ver Filtros Ayuda
dmesg.0
▷ dpkg.log
▷ dpkg.log.1
fontconfig.log
kern.log
kern.log.1
lpr.log
lpr.log.1
mail.err
mail.info
mail.log
mail.warn
▷ messages
▷ messages.1
pm-powersave.log
pm-powersave.log.1
pm-suspend.log
pm-suspend.log.1
popularity-contest
popularity-contest.0
Jul 5 09:12:47 coscoll kernel: [ 1.002430] rtc_cmos 00:03: setting system clock
Jul 5 09:12:47 coscoll kernel: [ 1.002472] Initializing network drop monitor serv
Jul 5 09:12:47 coscoll kernel: [ 1.002504] Freeing unused kernel memory: 380k fr
Jul 5 09:12:47 coscoll kernel: [ 1.002946] Write protecting the kernel text: 250
Jul 5 09:12:47 coscoll kernel: [ 1.002974] Write protecting the kernel read-only
Jul 5 09:12:47 coscoll kernel: [ 1.020884] input: AT Translated Set 2 keyboard a
Jul 5 09:12:47 coscoll kernel: [ 1.039419] udev[59]: starting version 164
Jul 5 09:12:47 coscoll kernel: [ 1.248335] SCSI subsystem initialized
Jul 5 09:12:47 coscoll kernel: [ 1.276866] usbcore: registered new interface dri
Jul 5 09:12:47 coscoll kernel: [ 1.276978] usbcore: registered new interface dri
Jul 5 09:12:47 coscoll kernel: [ 1.277093] usbcore: registered new device driver
Jul 5 09:12:47 coscoll kernel: [ 1.288298] via-rhine.c:v1.10-LK1.4.3 2007-03-06
Jul 5 09:12:47 coscoll kernel: [ 1.288376] via-rhine 0000:00:12.0: PCI INT A ->
Jul 5 09:12:47 coscoll kernel: [ 1.293769] eth0: VIA Rhine II at 0x1c000, 00:0b:0
Jul 5 09:12:47 coscoll kernel: [ 1.294495] eth0: MII PHY found at address 1, sta
Jul 5 09:12:47 coscoll kernel: [ 1.335741] ehci_hcd: USB 2.0 'Enhanced' Host Co
Jul 5 09:12:47 coscoll kernel: [ 1.335821] ehci_hcd 0000:00:10.4: PCI INT C -> C
Jul 5 09:12:47 coscoll kernel: [ 1.338803] ehci_hcd 0000:00:10.4: EHCI Host Cont
Jul 5 09:12:47 coscoll kernel: [ 1.338853] ehci_hcd 0000:00:10.4: new USB bus re
Jul 5 09:12:47 coscoll kernel: [ 1.338944] ehci_hcd 0000:00:10.4: irq 21, io mem
Jul 5 09:12:47 coscoll kernel: [ 1.340219] Floppy drive(s): fd0 is 1.44M
Jul 5 09:12:47 coscoll kernel: [ 1.351970] uhci_hcd: USB Universal Host Controll
Jul 5 09:12:47 coscoll kernel: [ 1.352054] ehci_hcd 0000:00:10.4: USB 2.0 starte
Jul 5 09:12:47 coscoll kernel: [ 1.352121] usb usb1: New USB device found, idVer
Jul 5 09:12:47 coscoll kernel: [ 1.352128] usb usb1: New USB device strings: Mfr
Jul 5 09:12:47 coscoll kernel: [ 1.352129] usb usb1: Product: EHCI Host Controll
511 líneas (48,2 KiB) - última actualización: mar jul 5 15:22:31 2011
```

KSystemLog

The screenshot shows the KSystemLog application window. The top menu bar includes File, Edit, Logs, Window, Settings, and Help. Below the menu is a toolbar with icons for Stop, Reload, Details, System Log, Kernel Log, Authentication Log, Daemons' Logs, and X.org Log. A search bar labeled 'Filter: Enter your search here...' is followed by a dropdown menu set to 'All'. The main area is a table with columns: Date, Host, Process, and Message. The table lists numerous log entries from July 5, 2011, primarily from the 'avahi-daemon[1511]' process. One specific entry is highlighted with a blue box and a tooltip providing details: 'Date: 2011-07-05 12:03:09', 'Level: Information', and 'Original file: /var/log/syslog'. At the bottom of the window, status bars show '190 lines.', '17:22:43: Log successfully loaded.', and 'Last updated: 17:22:43.'

Date	Host	Process	Message
2011-07-05 11:03:09	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::3e4a:92ff:fe43:b3dd.
2011-07-05 11:03:09	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef5:f42a.
2011-07-05 11:03:09	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef9:94a7.
2011-07-05 11:17:01	coscoll	/USR/SBIN/CRON[3122]	(root) CMD (cd / && run-parts -report /etc/cron.hourly)
2011-07-05 11:17:04	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef5:f42a.
2011-07-05 11:17:04	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::223:7dff:fe8e:548d.
2011-07-05 11:17:04	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef5:847b.
2011-07-05 11:17:04	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef9:94a7.
2011-07-05 11:17:04	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::6ab5:99ff:fea9:fc4d.
2011-07-05 11:25:05	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef5:847b.
2011-07-05 11:51:12	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef5:847b.
2011-07-05 11:51:12	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef9:94a7.
2011-07-05 11:51:12	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef5:f42a.
2011-07-05 11:51:12	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::6ab5:99ff:fea9:fc4d.
2011-07-05 11:51:12	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::223:7dff:fe8e:548d.
2011-07-05 12:03:08	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::223:7dff:fe8e:548d.
2011-07-05 12:03:08	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef9:94a7.
2011-07-05 12:03:08	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef5:f42a.
2011-07-05 12:03:09	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::223:7dff:fe8e:548d.
2011-07-05 12:03:09	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef9:94a7.
2011-07-05 12:03:09	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef5:f42a.
2011-07-05 12:17:01	coscoll	/USR/SBIN/CRON[3278]	(root) CMD (cd / && run-parts -report /etc/cron.hourly)
2011-07-05 12:31:42	coscoll	kernel	[11954.501838] [drm] nouveau 0000:01:00.0: Setting dpmis ...
2011-07-05 12:51:13	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef5:f42a.
2011-07-05 12:51:13	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::223:7dff:fe8e:548d.
2011-07-05 12:51:13	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::223:7dff:fe90:d7f5.
2011-07-05 12:51:13	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::225:b3ff:fef9:94a7.
2011-07-05 12:51:13	coscoll	avahi-daemon[1511]	Invalid response packet from host fe80::6ab5:99ff:fea9:fc4d.

190 lines. 17:22:43: Log successfully loaded. Last updated: 17:22:43.

Enlaces

<http://www.aboutdebian.com/syslog.htm>

TCPDump

- <http://es.wikipedia.org/wiki/Tcpdump>

TCPDump es una utilidad de análisis de tráfico de red en línea de comandos. Pese a su complejidad inicial, una vez aprendido su funcionamiento básico será la herramienta que nos resultará más útil para saber qué está pasando en la red. Nos va a permitir examinar en bruto el tráfico que está pasando por la red, así como filtrar por protocolo, IP, MAC, puerto, ect.

Mikrotik tiene también un *sniffer* de red, pero no es tan completo y cómodo como **tcpdump**.

Consideraciones iniciales

- TCPDump únicamente funciona cuando lo ejecutamos en modo **root**. Esto es así por motivos de seguridad.
- El concepto de TCPDump es lo que se muestra en esta imagen, poner en una red un cliente que “observa” el tráfico que se intercambia entre el resto de clientes.
- En las redes switcheadas actuales podemos observar el tráfico de los clientes directamente conectados al switch. En las redes inalámbricas podemos observar el tráfico que fluye por todo el enlace inalámbrico.
- En nuestras máquinas enrutadoras cobra sentido ya que hay mucho tráfico.
- Utilizarlo nos hará ser consciente de lo importante que es cuidar la seguridad en nuestras comunicaciones.



Server
10.0.0.1



Client
10.0.0.2



Manual y ejemplos

```
# man tcpdump
```

La mejor manera de entender tcpdump es ponerlo en práctica

Capturar tráfico cuya dirección IP de origen sea 10.228.130.1.

```
# tcpdump -i wlan0 src host 10.228.130.1
```

```
root@lateralus:~# tcpdump -n -i wlan0 src host 10.228.130.1
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on wlan0, link-type EN10MB (Ethernet), capture size
65535 bytes
14:56:06.291867 IP 10.228.130.1.80 > 192.168.1.222.37271: Flags
[S.], seq 2510715723, ack 98586484, win 5792, options [mss
1460,sackOK,TS val 119619640 ecr 146592,nop,wscale 2], length 0
14:56:06.294414 IP 10.228.130.1.80 > 192.168.1.222.37271: Flags
[.], ack 376, win 1716, options [nop,nop,TS val 119619640 ecr
146592], length 0
14:56:06.299367 IP 10.228.130.1.80 > 192.168.1.222.37271: Flags
[.], seq 1:1449, ack 376, win 1716, options [nop,nop,TS val
119619640 ecr 146592], length 1448
14:56:06.302374 IP 10.228.130.1.80 > 192.168.1.222.37271: Flags
[.], seq 1449:2897, ack 376, win 1716, options [nop,nop,TS val
119619640 ecr 146592], length 1448
```

Capturar tráfico con destino a la dirección MAC 50:43:A5:AE:69:55

```
# tcpdump -i eth0 ether dst 50:43:A5:AE:69:55
```

Capturar tráfico con red destino 10.228.0.0

```
# tcpdump -i wlan0 dst net 10.228.0.0/16
```

Capturar tráfico con puerto origen o destino el 53 (DNS)

```
# tcpdump -i wlan0 port 53
```

Capturar los paquetes de tipo ICMP

```
# tcpdump -i eth1 ip proto \icmp
```

Capturar los paquetes de tipo UDP

```
# tcpdump -i tunnel0 ip proto \udp
```

Capturar todo el tráfico TCP excepto el de SSH

```
# tcpdump -i tunnel0 ip proto \tcp and not port 22
```

Capturar tráfico para examinarlo posteriormente con wireshark:

```
# tcpdump -i <interface> -s 65535 -w <some-file>
```

Mikrotik. Packet sniffer

Muy similar a la funcionalidad de tcpdump, aunque un tanto incómodo y menos potente. Nos permite analizar paquetes que pasan por el router y filtrar por interfaz, ip origen/destino o protocolo.

```

#Para acceder:
/tool sniffer

# arrancarlo
/tool sniffer start

# pararlo
/tool sniffer stop

# Examinar el tráfico
/tool sniffer packet print
# TIME INTERFACE SRC-ADDRESS
DST-ADDRESS IP-
PROTOCOL SIZE
0 0.001 bridge1 10.228.144.161:22 (ssh)
10.228.144.170:48615 tcp
180
1 0.001 ether3 10.228.144.161:22 (ssh)
10.228.144.170:48615 tcp
180
2 0.027 ether3 10.228.144.163:3128 (squid)
10.228.145.58:50946 tcp
46
3 0.027 bridge1 10.228.144.163:3128 (squid)
10.228.145.58:50946 tcp
46
4 0.027 wlan2 10.228.144.163:3128 (squid)
10.228.145.58:50946 tcp
40
...
17 0.043 ether3 10.228.144.170:48615
10.228.144.161:22 (ssh) tcp
52
18 0.043 bridge1 10.228.144.170:48615
10.228.144.161:22 (ssh) tcp
52
19 0.077 wlan2 10.228.145.58:50994
10.228.144.163:3128 (squid) tcp
46
20 0.077 bridge1 10.228.145.58:50994
10.228.144.163:3128 (squid) tcp
40
21 0.077 ether3 10.228.145.58:50994
10.228.144.163:3128 (squid) tcp
40
22 0.249 wlan2 10.228.145.58:50946
10.228.144.163:3128 (squid) tcp
46
23 0.249 bridge1 10.228.145.58:50946
10.228.144.163:3128 (squid) tcp
40

```

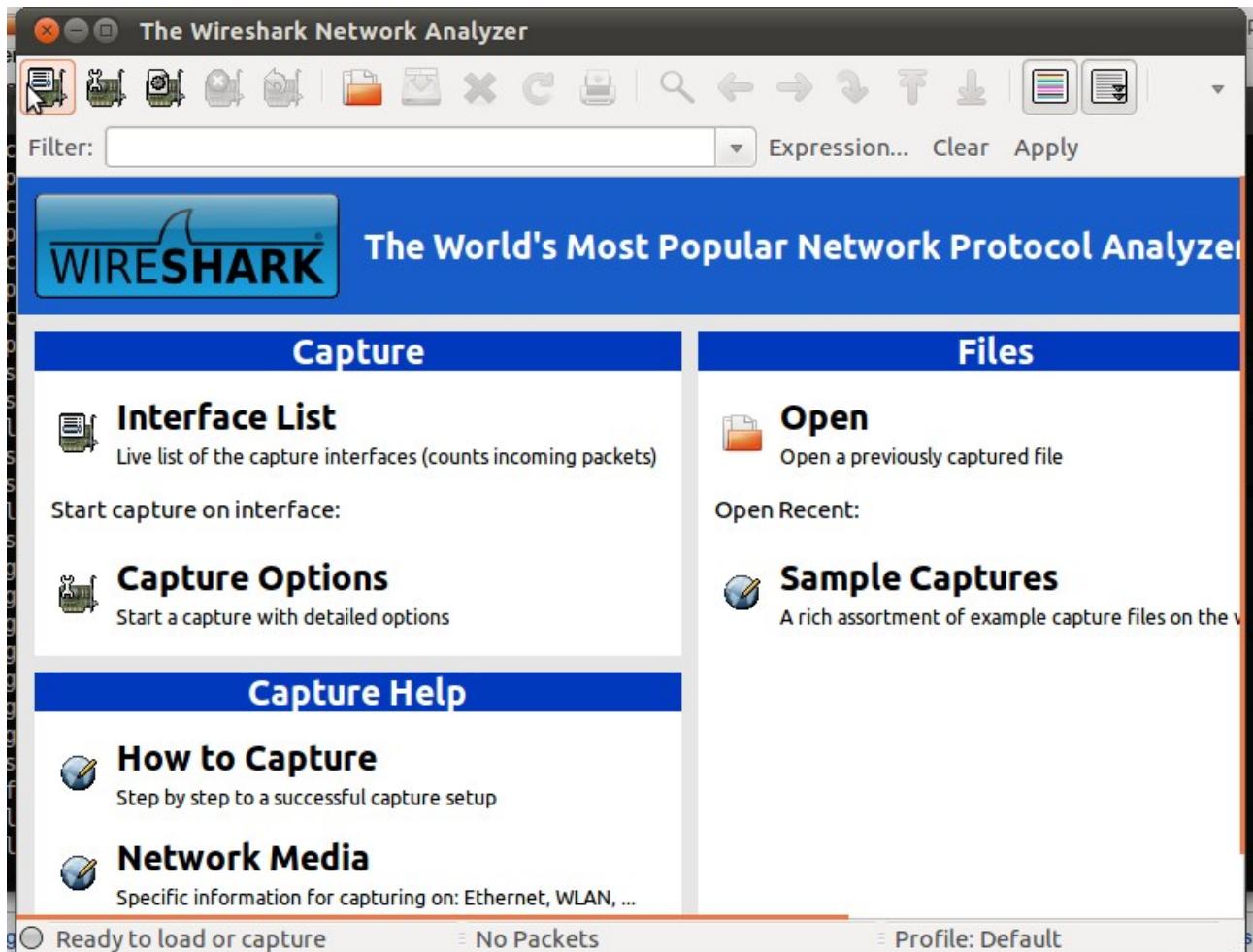
La documentación de esta herramienta la podemos ver aquí:

http://wiki.mikrotik.com/wiki/Manual:Tools/Packet_Sniffer

Wireshark

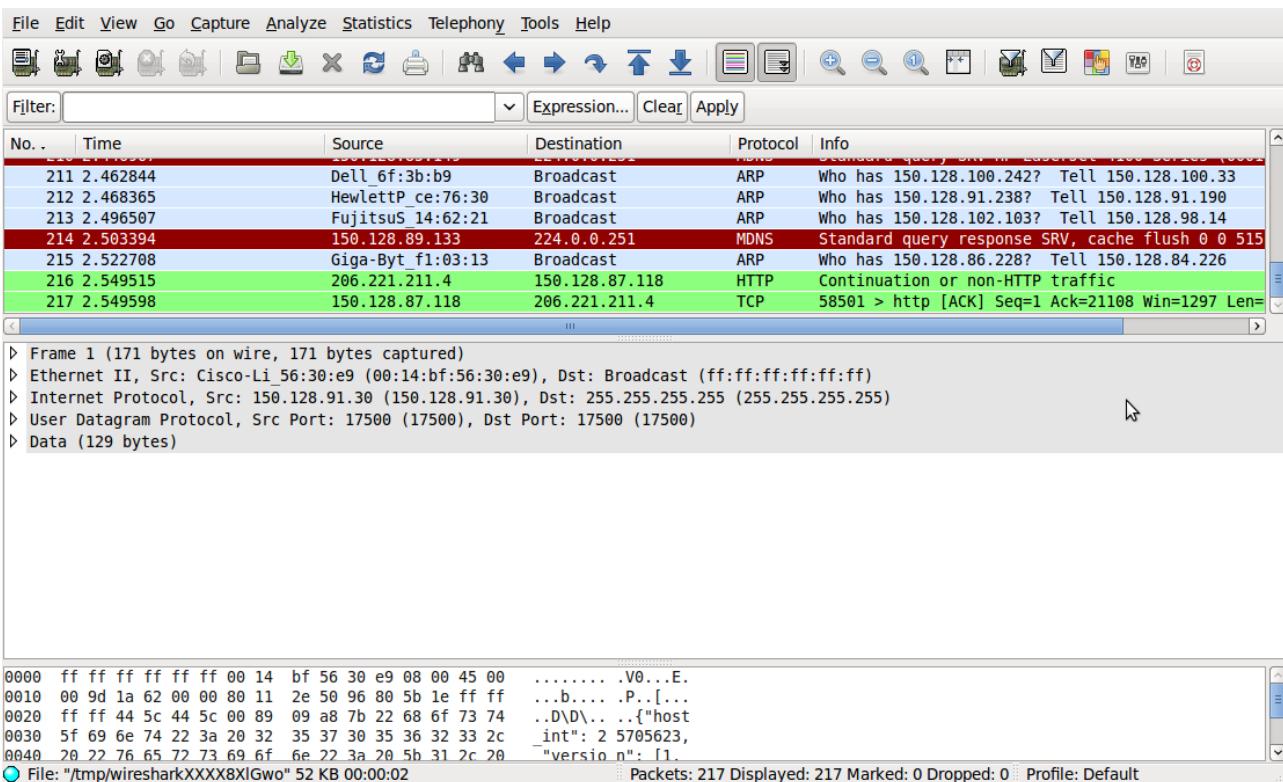
- <http://es.wikipedia.org/wiki/Wireshark>

Wireshark es, al igual que TCPDump, una herramienta de análisis del tráfico de red vía interfaz gráfica. Nos permite *escuchar* lo que está pasando en la red en un momento determinado, para posteriormente analizarlo gracias a funcionalidades como el agrupamiento, organización y filtrado de información.

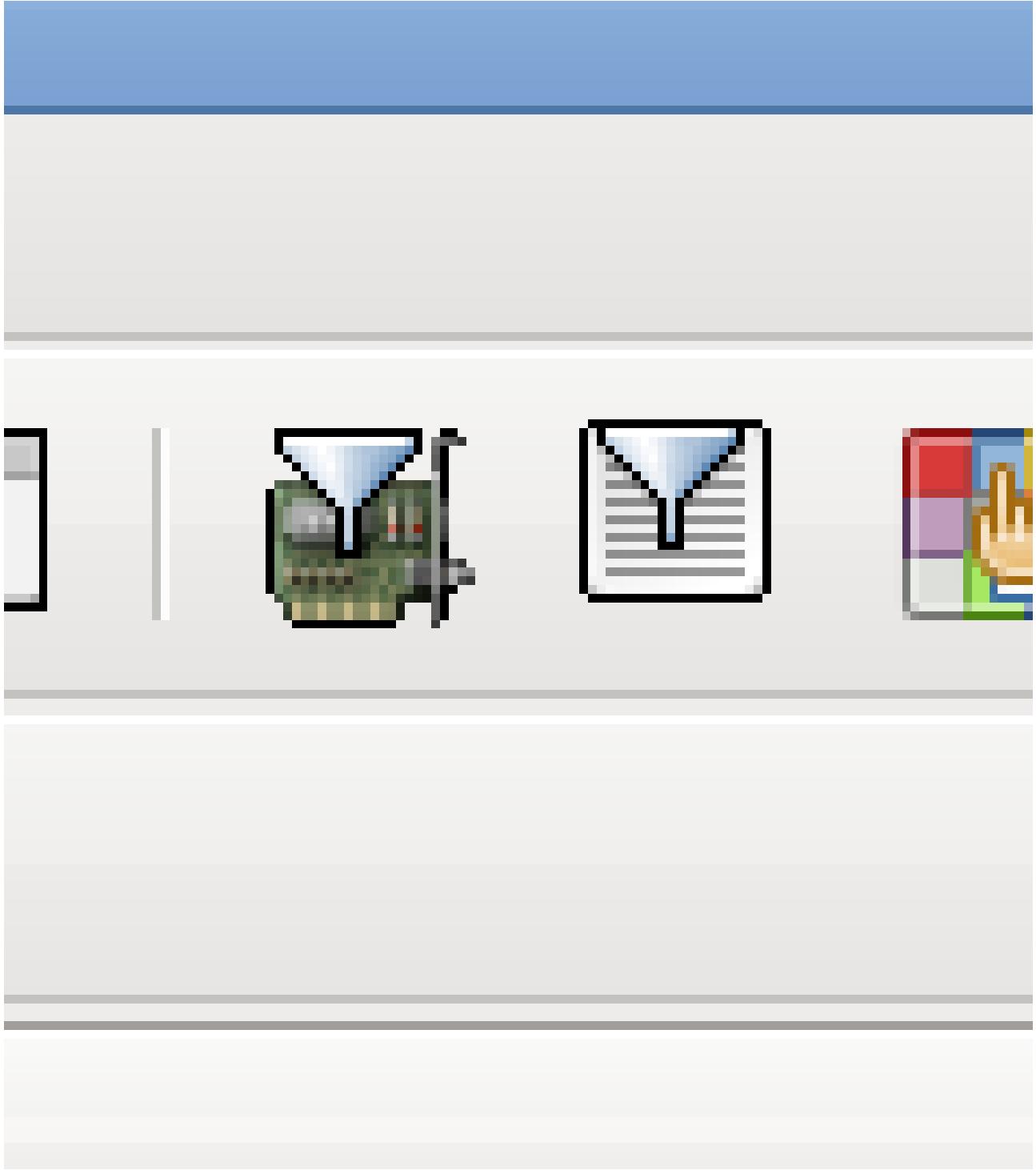


Debemos ejecutar Wireshark como administrador ya que se necesita acceso a los dispositivos de red. La captura realizada se puede guardar para analizarla posteriormente.

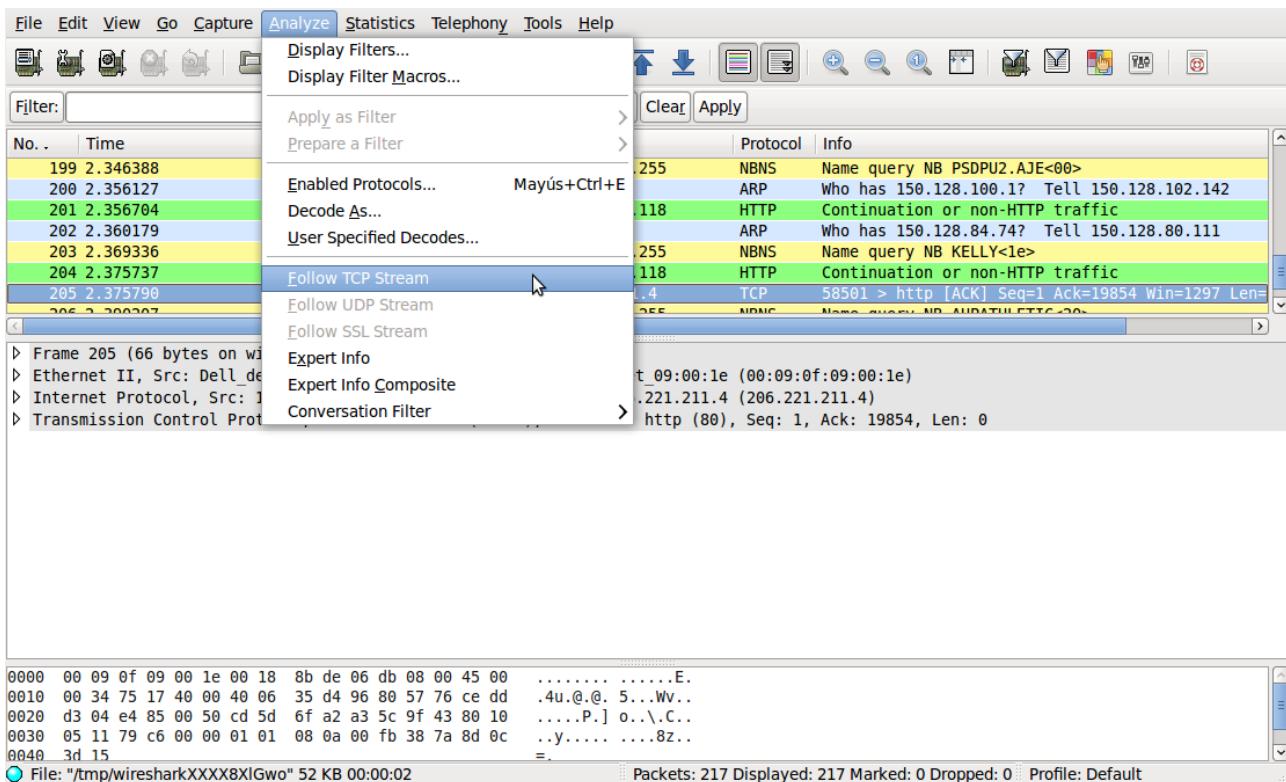
La ventana de la aplicación tiene tres partes. La primera nos muestra los mensajes capturados, la segunda, para el mensaje que seleccionemos, nos permite desplegar información sobre cada nivel de la pila de protocolos; esta parte es muy interesante porque podemos ver datos de la cabecera de cada protocolo. La tercera nos muestra en hexadecimal todos los bits del mensaje.



Sobre una captura se pueden aplicar filtros para focalizarnos en algún tipo de tráfico. Hay filtros de captura (para limitar el tamaño de los datos recogidos) y filtros que se aplican sobre una captura ya realizada (si quitamos el filtro volveremos a ver todos los mensajes de la captura).



Una utilidad interesante de Wireshark es *Follow tcp stream*. Con esta utilidad, si seleccionamos un mensaje de una conexión TCP, veremos los mensajes que se intercambian en los dos sentidos de esa conexión. El diálogo se ve con un color para interlocutor.



El resultado:

```

Stream Content
Accept: */*
Accept-Language: es,en-us;q=0.8,en;q=0.6,fr;q=0.4,ca;q=0.2
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
X-Requested-With: XMLHttpRequest
Referer: http://guifi.net/es
Cookie: SESS7fe44f77befd2c10b23dbc9ccc9371a5=d9b8kdapuk2ojjq53ip69aqne6; has_js=1;
_utma=99061256.899456046.1309685194.1309952722.1310036537.8; _utmc=99061256; _utmz=99061256.1310036537.8.3.utmcsr=google|utmccn=(organic)|utmcmd=organic|utmctr=172.16.1.189%20site%3Aguifi.net; _utmv=99061256.1=iptype=v4=1; Drupal_l10n_client=0;
_utmb=99061256.8.10.1310036537
If-Modified-Since: Thu, 07 Jul 2011 11:02:16 GMT

HTTP/1.1 200 OK
Date: Thu, 07 Jul 2011 11:03:24 GMT
Server: Apache
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Last-Modified: Thu, 07 Jul 2011 11:03:24 GMT
Cache-Control: store, no-cache, must-revalidate
Cache-Control: post-check=0, pre-check=0
Content-Length: 1973
Keep-Alive: timeout=3, max=100
Connection: Keep-Alive
Content-Type: application/xml; charset=utf-8

Buscar Guardar como Imprimir Entire conversation (3188 bytes) | ASCII EBCDIC Hex Dump C Arrays Raw
Ayuda Filter Out This Stream Cerrar

```

ipcalc

Aunque no se trata de una herramienta de monitorización, es muy útil para averiguar el rango de IPs de una subred.

`ipcalc 10.228.130.3/27`

devuelve

Address:	10.228.130.3	00001010.11100100.10000010.000
00011		
Netmask:	255.255.255.224 = 27	11111111.11111111.11111111.111

```

00000
Wildcard: 0.0.0.31          00000000.00000000.00000000.000
11111
=>
Network: 10.228.130.0/27    00001010.11100100.10000010.000
00000
HostMin: 10.228.130.1      00001010.11100100.10000010.000
00001
HostMax: 10.228.130.30    00001010.11100100.10000010.000
11110
Broadcast: 10.228.130.31   00001010.11100100.10000010.000
11111
Hosts/Net: 30               Class A, Private Internet

```

Netdiscover

Netdiscover busca ordenadores en una red de forma pasiva (por el tráfico que ve pasar) o mediante peticiones arp (activamente consultado con el protocolo ARP). También puede ser útil para inspeccionar el tráfico arp de una red o para encontrar direcciones de red (en el modo auto scan).

`netdiscover -i eth0`

busca ordenadores en la interfaz eth0, realiza un scan de las subredes privadas definidas en el RFC 1918 (192.168.0.0/16, 172.16.0.0/16, 172.26.0.0/16, ..., 10.0.0.0/8). Es decir, que si no indicamos el rango de IPs que debe buscar, irá probando activamente dentro de los rangos de IPs privadas.

`netdiscover -i ath0`

busca ordenadores en la interfaz ath0

`netdiscover -i eth0 -r 150.128.2.0/27`

busca ordenadores en la interfaz eth0 dentro del rango 150.128.2.0/27

`netdiscover -p`

busca ordenadores en modo pasivo, sólo «mira» la información que pasa por la red.

`netdiscover -P`

muestra la salida de forma que se pueda almacenar en un fichero.

Traceroute/ping/MTR

ping

- <http://es.wikipedia.org/wiki/Traceroute>
- <http://es.wikipedia.org/wiki/Ping>
- [http://en.wikipedia.org/wiki/MTR_\(Software\)](http://en.wikipedia.org/wiki/MTR_(Software))

Ping es una herramienta de diagnóstico que comprueba el estado de una conexión mandando y recibiendo paquetes ICMP. Útil para comprobar

rápidamente el estado de la red, pero parco en datos.

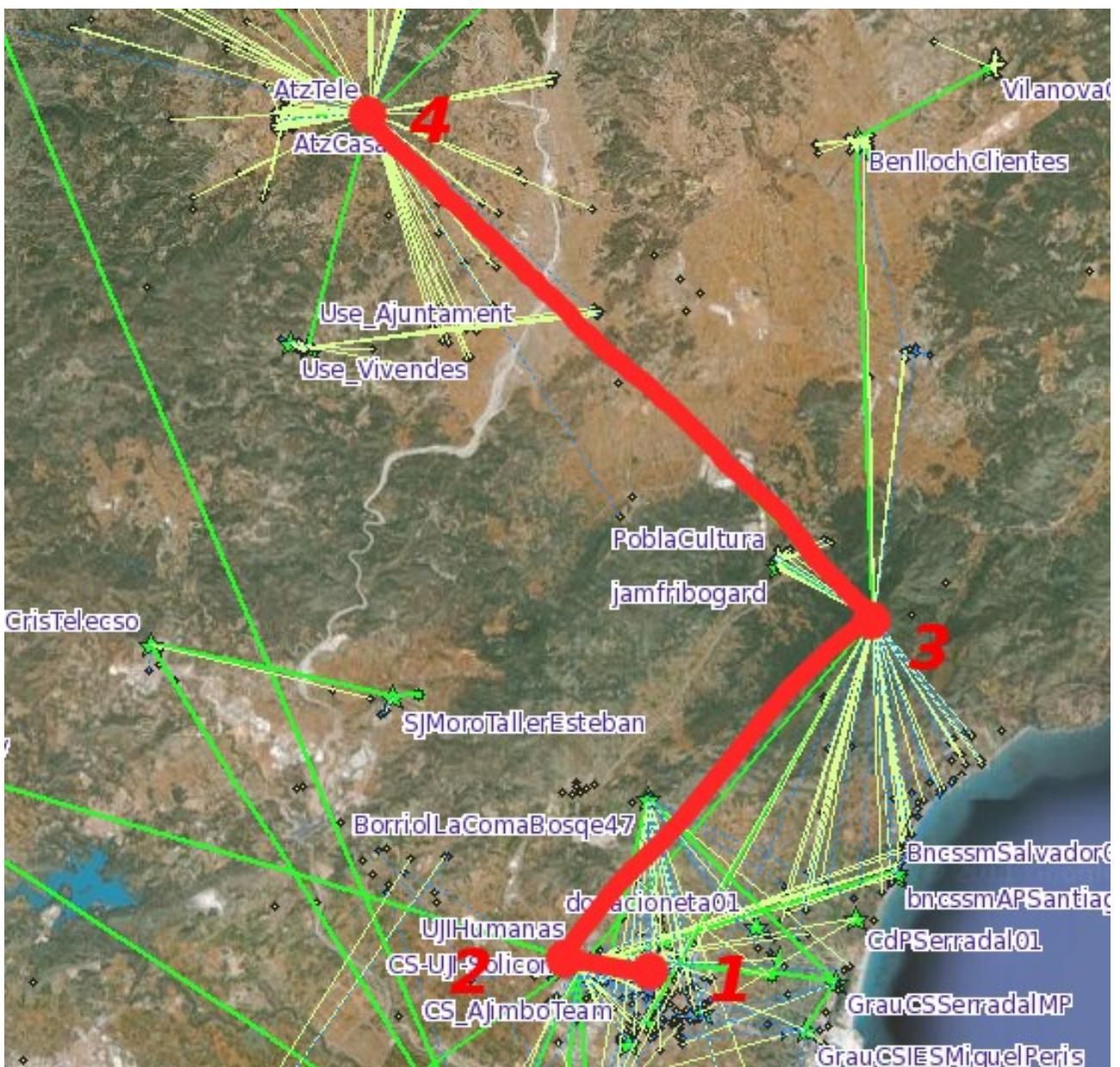
Podemos utilizar un par de parámetros útiles (sólo para Linux):

* -i time: podemos especificar el tiempo entre cada envío de trama ICMP.

* -t ttl: Especificamos el número máximo de hosts por los que puede pasar el paquete.

Ejemplo:

```
$ ping -i .2 -t 10 10.228.145.1
```



En routers que tengan varias IPs asignadas puede ser necesario especificar la IP origen del ping porque si no sabemos la que se envía puede que no obtengamos contestación. Por ejemplo, si hacemos ping desde un punto donde tenemos una ip 172.x.y.z y una 10.x.y.z si el ping da más de un salto y sale con la IP origen 172.x.y.z no recibiremos respuesta puesto que estas direcciones no se activan en el encaminamiento dinámico de guifi.net.

```
$ ping -I 10.228.130.1 10.228.132.33
```

MTR

Esta herramienta es la fusión de las dos anteriores, dándonos en un único comando lo mejor de **traceroute** y **ping**.

```
boots> mtr 10.228.131.1
HOST: boots                               Loss%   Snt    Last   Avg   Best
Wrst StDev
 1.|-- 10.228.144.189                     0.0%    10     0.2    0.2    0.2
0.3  0.0
 2.|-- 172.16.1.78                      0.0%    10     1.1    1.2    1.1
1.4  0.1
 3.|-- 172.16.1.189                     0.0%    10     1.3    1.5    1.3
2.1  0.3
 4.|-- 172.16.107.69                     0.0%    10     4.4    5.1    3.9
7.1  1.0
 5.|-- 10.228.131.1                      0.0%    10     4.3    6.6    4.0
16.9 3.9
```

Al igual que con ping, si se quiere forzar la IP origen, se debe utilizar la opción -a IP.

Pruebas de prestaciones

Para realizar pruebas de velocidad de uno o varios enlaces se puede utilizar la utilidad iperf. Está disponible tanto para GNU/Linux como para AirOs. Por el momento no está disponible en RouterOS que tiene sus propios test de velocidad.

La forma más sencilla de ejecutarlo es la siguiente: En uno de los extremos se ejecuta iperf en modo servidor

```
iperf -s
```

en el otro extremo se ejecuta el test contra el servidor

```
iperf -c 10.228.134.212
```

donde 10.228.134.212 es la IP del servidor.

En castello.guifi.net por ejemplo, hay un servidor iperf funcionando continuamente. De esta forma se pueden realizar tests de velocidad fácilmente

```
XM.v5.3# iperf -c 10.228.130.162 -P 5
```

```
-----
Client connecting to 10.228.130.162, TCP port 5001
TCP window size: 16.0 KByte (default)
-----
[ 5] local 10.228.170.3 port 2798 connected with 10.228.130.162
port 5001
[ 7] local 10.228.170.3 port 2799 connected with 10.228.130.162
port 5001
[10] local 10.228.170.3 port 2802 connected with 10.228.130.162
port 5001
[ 8] local 10.228.170.3 port 2800 connected with 10.228.130.162
port 5001
[ 9] local 10.228.170.3 port 2801 connected with 10.228.130.162
```

```

port 5001
[ ID] Interval Transfer Bandwidth
[  5] 0.0-10.0 sec 8.82 MBytes 7.40 Mbits/sec
[ ID] Interval Transfer Bandwidth
[  7] 0.0-10.0 sec 8.97 MBytes 7.52 Mbits/sec
[ ID] Interval Transfer Bandwidth
[ 10] 0.0-10.0 sec 8.66 MBytes 7.26 Mbits/sec
[ ID] Interval Transfer Bandwidth
[  8] 0.0-10.0 sec 8.88 MBytes 7.44 Mbits/sec
[ ID] Interval Transfer Bandwidth
[  9] 0.0-10.0 sec 8.90 MBytes 7.46 Mbits/sec
[SUM] 0.0-10.0 sec 44.2 MBytes 37.1 Mbits/sec

```

con la opción -P 5 indicamos que se realicen 5 pruebas de velocidad en paralelo de forma que el ancho de banda total es la suma de todos ellos.

Si se quiere realizar un test de velocidad entre dos dispositivos de ubiquity se puede utilizar el iperf tal y como se ha comentado anteriormente (entrando a los dispositivos por ssh), para ello deben tener IPs pertenecientes a la misma subred (o se debe poder llegar de una a la otra). La interfaz web del AirOS también dispone de una herramienta para medir la velocidad pero solo de otros dispositivos en la misma subred y se debe indicar el usuario y el password.

También cabe la posibilidad de hacer una descarga utilizando wget:

```

mperez@coscoll:~$ wget http://roure.act.uji.es/v
--2011-07-07 15:41:35--  http://roure.act.uji.es/v
Resolviendo rourre.act.uji.es... 150.128.97.53
Connecting to rourre.act.uji.es|150.128.97.53|:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 146276352 (140M) [text/plain]

```

```
Saving to: `v'
```

```
100%[=====] 146.276.352 11,1M/s  
in 13s
```

```
2011-07-07 15:41:48 (11,1 MB/s) - `v' saved [146276352/146276352]
```

Descarga el fichero v almacenado en el servidor web de roure.act.ubi.es. Muestra en pantalla la evolución de la descarga y al final nos muestra el tamaño del fichero y la velocidad a la que se ha descargado.

```
2011-07-07 15:41:48 (11,1 MB/s) - `v' saved [146276352/146276352]
```

También la orden scp se puede utilizar para averiguar la velocidad entre dos ordenadores (o nodos):

```
jlopez@coscoll:~$ scp jlopez@10.228.130.14:f.tgz .  
f.tgz 100% 140MB  
11.6MB/s 00:12
```

descarga el fichero f.tgz desde el ordenador 10.228.130.14 hasta el ordenador local (se solicitará la contraseña). Como se observa, muestra el tamaño del fichero, la velocidad de la descarga y el tiempo utilizado. Al tratarse de una copia segura (scp utiliza SSL) se añade una pequeña sobrecarga para el cifrado de los datos, aunque puede despreciarse.

Análogamente se puede realizar una transferencia del ordenador local a uno remoto mediante:

```
mperez@coscoll:~$ scp f.tgz jlopez@10.228.130.14:  
f.tgz 100% 140MB  
11.6MB/s 00:12
```

En RouterOS se puede utilizar bandwidth-server/bandwidth-test del apartado tools. Es similar al iperf pero sólo se puede utilizar para realizar pruebas entre dos mikrotiks

(http://wiki.mikrotik.com/wiki/Manual:Tools/Bandwidth_Test).

En uno de los mikrotiks se pone en marcha el servidor

```
/tool bandwidth-server set enables=yes
```

y se prueba en el otro mediante

```
/tool bandwidth-test protocol=tcp address=10.228.145.1
```

Tracedump:

```
newBaseSize: 12pt  
newBaseSizeInPt: 12
```