

Introducción a NetGUI

Redes I

Departamento de Sistemas Telemáticos y Computación (GSyC)

Septiembre de 2011



©2011 Grupo de Sistemas y Comunicaciones.
Algunos derechos reservados.
Este trabajo se distribuye bajo la licencia
Creative Commons Attribution Share-Alike
disponible en <http://creativecommons.org/licenses/by-sa/2.1/es>

- 1 NetGUI
- 2 Herramientas de configuración de la red: `ifconfig`, `ip`, `route`
- 3 Configuración de red mediante ficheros de configuración
- 4 Herramientas de diagnóstico de red: `arp`, `ping`, `tcpdump`, `wireshark`

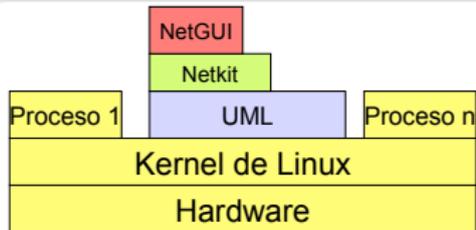
Contenidos

- 1 NetGUI
- 2 Herramientas de configuración de la red: `ifconfig`, `ip`, `route`
- 3 Configuración de red mediante ficheros de configuración
- 4 Herramientas de diagnóstico de red: `arp`, `ping`, `tcpdump`, `wireshark`

NetGUI

- **NetGUI** es herramienta construida sobre el software Netkit, que a su vez se apoya en *User-mode Linux* (UML).
- Funcionalidad:
 - Creación a través de una interfaz gráfica de un escenario de red mediante selección/arrastre de routers, concentradores y estaciones finales.
 - Almacenamiento y recuperación de escenarios de red previamente creados.
 - Interconexión de elementos de red
 - Arranque del HW emulado: cada estación final y cada router puede configurarse a través de una consola Linux.
 - Operación de la red a través de las consolas Linux.
- Es Software Libre que puede instalarse en Linux:
<http://mobiquo.gsync.es/netgui>

NetGUI, Netkit y UML



- **NetGUI:**
 - Interfaz gráfica para Netkit.
- **Netkit:**
 - Entorno software que permite realizar experimentos con redes de ordenadores virtuales sin necesidad de disponer de dispositivos de comunicaciones ni de ordenadores reales.
 - Permite arrancar varios nodos virtuales (ordenadores, hubs, routers) que ejecutan el kernel y las aplicaciones de GNU/Linux.
 - Utiliza máquinas virtuales UML.
- **UML (*User-mode Linux*):**
 - Es un kernel de Linux que puede ser arrancado como un proceso de usuario en una máquina real que tenga instalado Linux.
 - Llamaremos **máquinas virtuales** a cada uno de los procesos UML que emula un ordenador o un router, y **máquina real** a aquella en la que se están ejecutando los procesos UML.

La interfaz gráfica

- NetGUI se arranca con la orden `netgui.sh`

Menú para cargar/guardar un diagrama y para ayuda

Crear una máquina

Crear un router

Crear un switch

Crear un hub

Conectar dos dispositivos

Borrar

Herramienta de selección

Arrancar dispositivos

Parar dispositivos

Centrar el diagrama en la parte visible de la ventana

Zona para crear el diagrama de red

The screenshot shows a window titled "NetGUI: lab-RIP" with a "File Help" menu. The toolbar contains icons for creating a PC, router, switch, hub, connecting devices, deleting, selecting, starting, and stopping devices. The main area displays a network diagram with five routers (r1-r5), two hubs (hub1, hub2), and two PCs (pc1, pc2). Connections are labeled with interface names like eth0, eth1, eth2. A dashed red box highlights the toolbar and diagram area, and an orange box highlights the diagram area.

La herramienta de selección

- La herramienta de selección permite la siguiente funcionalidad:



- **Seleccionar un elemento:** haciendo clic con el botón izquierdo del ratón se selecciona un elemento del escenario de red.
- **Mover un elemento:** arrastrando con el botón izquierdo del ratón se mueve un elemento dentro del escenario de red.
- **Arrancar/Parar un dispositivo (máquina, router o switch):** haciendo clic con el botón derecho sobre un dispositivo si está parado se arranca, y si está arrancado se para.
IMPORTANTE: Hay que esperar unos segundos para que el dispositivo arranque o se detenga completamente. Cuando un nodo está arrancado aparecen dos flechas azules sobre su icono.
- **Mostrar la consola de un nodo arrancado (ordenador o encaminador):** haciendo un doble clic con el botón izquierdo del ratón sobre un dispositivo, su ventana de terminal pasa a primer plano.

Acciones sobre toda la figura

- **Mover toda la figura**: pulsando y arrastrando con el botón **izquierdo** del ratón sobre el fondo de la ventana (en un lugar en el que no haya ningún elemento).
- **Zoom**: pulsando y arrastrando con el botón **derecho** del ratón:
 - arrastrando hacia la derecha: aumentar el zoom
 - arrastrando hacia la izquierda: disminuir el zoom
- **Centrar**: El botón “Centrar” permite centrar la figura en la ventana:



El Menú *File*

- El menú *File* permite guardar escenarios de red y cargar escenarios guardados previamente.
- A la hora de guardar con *Save*, la primera vez hay que elegir un nombre de carpeta. En esa carpeta se almacenarán todos los ficheros asociados al escenario:
 - `netgui.nkp`: contiene la información del dibujo del escenario.
 - `*.disk`: contiene el sistema de ficheros de cada máquina virtual, con las modificaciones que se hayan hecho en cada una después de arrancarlas.
- Es conveniente guardar nada más terminar de dibujar el escenario, antes de arrancar las máquinas virtuales.

Nombres de las carpetas para NetGUI

- **MUY IMPORTANTE:**
 - No se pueden guardar escenarios en un *path* que incluya **ningún directorio en cuyo nombre haya algún espacio en blanco**.
 - No sólo el nombre de la carpeta del escenario debe estar libre de espacios, sino que todas las carpetas desde el *HOME* hasta la del escenario deben tener **NOMBRES SIN ESPACIOS**.

Consolas

The screenshot displays the NetGUI interface for a lab titled 'lab-intro'. The network topology includes:

- pc1** (green) connected to **hub1** via **eth0**.
- pc2** (blue) connected to **hub1** via **eth0**.
- hub1** connected to **r1** (red) via **eth0**.
- r1** connected to **hub2** via **eth1**.
- hub2** connected to **pc3** (orange) via **eth0**.

Four terminal consoles are shown on the right, each connected to a device in the network via dashed arrows:

- Consola de pc1** (green border): Shows the output of `<none>` and `Netkit phase 2 initialization terminated`. It also displays the login sequence for **pc1**: `pc1 login: root (automatic login)`, `Last login: Mon Sep 19 08:09:11 UTC 2011 on tty1`, and the prompt `pc1:~#`.
- Consola de pc2** (blue border): Shows the output of `<none>` and `Netkit phase 2 initialization terminated`. It also displays the login sequence for **pc2**: `pc2 login: root (automatic login)`, `Last login: Mon Sep 19 08:09:11 UTC 2011 on tty1`, and the prompt `pc2:~#`.
- Consola de r1** (red border): Shows the output of `<none>` and `Netkit phase 2 initialization terminated`. It also displays the login sequence for **r1**: `r1 login: root (automatic login)`, `Last login: Mon Sep 19 08:09:11 UTC 2011 on tty1`, and the prompt `r1:~#`.
- Consola de pc3** (orange border): Shows the output of `<none>` and `Netkit phase 2 initialization terminated`. It also displays the login sequence for **pc3**: `pc3 login: root (automatic login)`, `Last login: Mon Sep 19 08:10:05 UTC 2011 on tty1`, and the prompt `pc3:~#`.

Arrancar NetGUI

- NetGUI se arranca escribiendo en un terminal la orden `netgui.sh`
- Si ha habido ejecuciones previas de NetGUI, resulta conveniente ejecutar ANTES la orden `clean-netgui.sh`
- Cuando la anterior ejecución de NetGUI ha terminado de forma incorrecta, se hace imprescindible utilizar `clean-netgui.sh` antes de volver a arrancar NetGUI
- Por lo tanto, el procedimiento adecuado para arrancar NetGUI es:
 - ① Ejecutar en un terminal la orden: `clean-netgui.sh`
 - ② Ejecutar en un terminal la orden: `netgui.sh`

Cerrar NetGUI

- NUNCA debe cerrarse NetGUI sin apagar ANTES todas las máquinas virtuales.
- NUNCA debe cerrarse la ventana de una máquina virtual pulsando la **X** del marco de la ventana. Si se realiza esta acción, el sistema de ficheros de la máquina virtual quedará inconsistente, y aparecerán errores al reiniciar la máquina.
- Para apagar una máquina virtual debe usarse el botón rojo de la interfaz. Si al hacerlo la máquina virtual no se apagase, puede escribirse en su terminal la orden `halt` y esperar a que la ventana se cierre sola.
- NUNCA debe cerrarse NetGUI pulsando la **X** del marco de la ventana principal del escenario. Si se hiciera, ya no se podrían apagar las máquinas virtuales a través de NetGUI y habría que hacerlo escribiendo `halt` en sus ventanas de terminal.
- Por lo tanto, el procedimiento adecuado para salir de NetGUI es:
 - 1 Apagar una a una las máquinas virtuales mediante la interfaz de NetGUI.
 - 2 Si alguna máquina virtual no pudiera apagarse mediante la interfaz, apagarla escribiendo `halt` en su ventana de terminal
 - 3 Si ha habido cambios en el dibujo del escenario que se quieran guardar, elegir en el menú `File -> Save`.
 - 4 Elegir en el menú `File -> Exit`.

Contenidos

- 1 NetGUI
- 2 Herramientas de configuración de la red: `ifconfig`, `ip`, `route`
- 3 Configuración de red mediante ficheros de configuración
- 4 Herramientas de diagnóstico de red: `arp`, `ping`, `tcpdump`, `wireshark`

Configuración de red

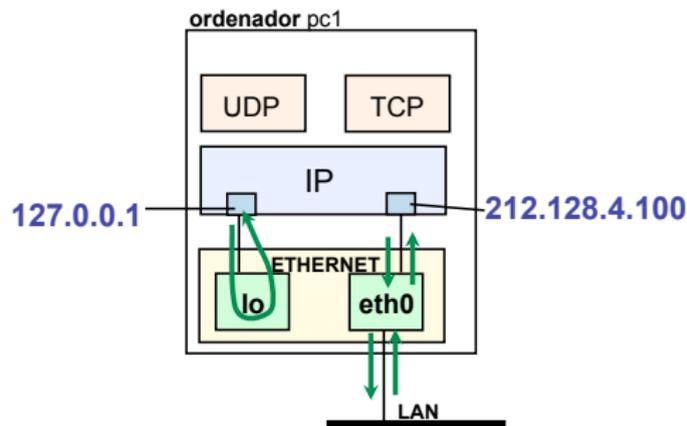
- **Configuración de red:** Añadir/eliminar/modificar direcciones IP y/o rutas en las tablas de encaminamiento
- Órdenes que se utilizan:
 - `ifconfig`
 - `ip`
 - `route`

Interfaces de red de una máquina Linux

- Todas las máquinas Linux tienen siempre la interfaz de red `lo` (**interfaz de loopback**), que es una interfaz de autoenvío.
- Una máquina Linux que tenga una tarjeta Ethernet tiene, además de la interfaz `lo`, la interfaz `eth0`.
- Un *router* Linux que tenga dos tarjetas Ethernet tendrá dos interfaces `eth`: `eth0` y `eth1`.

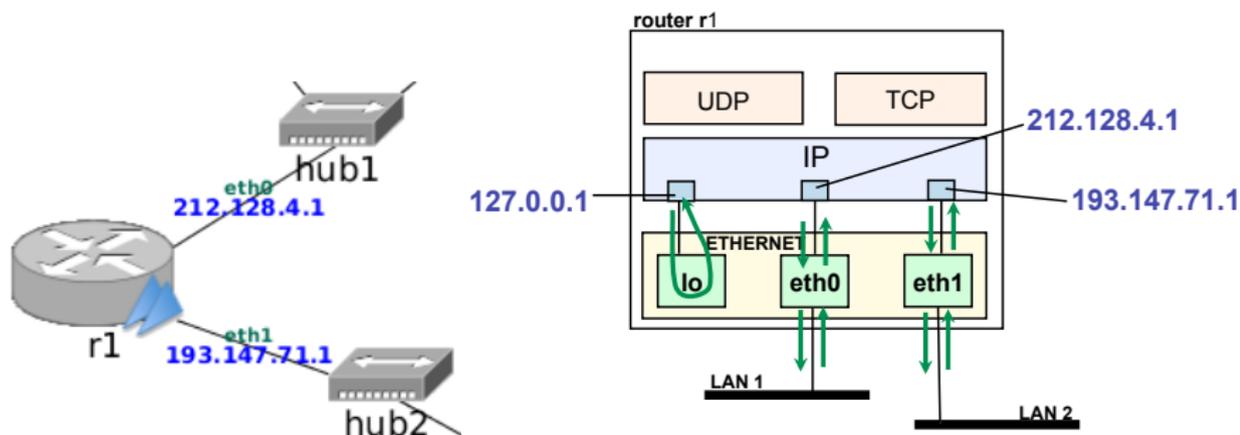
Interfaces de red y direcciones IP (I)

- A cada interfaz de red se le asigna una dirección IP
- A la interfaz de *loopback* se le suele asignar siempre la dirección IP 127.0.0.1
- Ejemplo de un PC de NetGUI:



Interfaces de red y direcciones IP (II)

- Ejemplo de un *router* de NetGUI:



Mostrar información de las interfaces de red

- Esta información incluye direcciones, Ethernet, IP, máscaras de red, etc.
- Con `ifconfig`:

```
pc1:~# ifconfig
eth0: Link encap:Ethernet Hwaddr 0A:29:92:55:93:70
inet addr:212.128.4.100 Bcast:212.128.4.255 Mask: 255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:4 errors:0 dropped:0 overruns:0 frame:0
TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:224 (224.0 b) TX bytes:280 (280.0 b)
Interrupt:5
lo: Link encap:Local Loopback
inet addr:127.0.0.1 Bcast:255.0.0.0
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:6 errors:0 dropped:0 overruns:0 frame:0
TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:504 (504.0 b) TX bytes:504 (504.0 b)
```

- Con `ip`:

```
pc1:~# ip address show
0: lo: <LOOPBACK,UP,10000> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
1: eth0: <BROADCAST,MULTICAST,UP,10000> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 0A:29:92:55:93:70 brd ff:ff:ff:ff:ff:ff
    inet 212.128.4.100/24 brd 212.128.4.255 scope global eth0
```

Añadir/eliminar una dirección IP

- Para configurar una dirección IP es necesario conocer: la interfaz donde la vamos a configurar, la dirección IP, y la máscara (o los bits que se corresponden con el prefijo de máscara).

- **Añadir una dirección IP:** Puede hacerse con `ifconfig` o con `ip`

- `ifconfig <interfaz> <dirIP> netmask <máscara>`

```
pc1:~# ifconfig eth0 10.0.0.1 netmask 255.255.255.0
```

- `ip address add dev <interfaz> <dirIP/prefijoMáscara> broadcast +`

```
pc1:~# ip link set eth0 up
pc1:~# ip address add dev eth0 10.0.0.1/24 broadcast +
```

- **Eliminar una dirección IP:** Puede hacerse con `ifconfig` o con `ip`

- Con `ifconfig` sólo se puede “apagar” la interfaz, que no es exactamente lo mismo que eliminar la dirección IP:

`ifconfig <interfaz> down`

```
pc1:~# ifconfig eth0 down
```

- `ip address del dev <interfaz> <dirIP/prefijoMáscara>`

```
pc1:~# ip address del dev eth0 10.0.0.1/24
```

- Después de añadir/eliminar una dirección IP es conveniente comprobar que la configuración se ha realizado correctamente (con `ip` o `ifconfig`).
- Los cambios realizados con estas órdenes no se conservan al reiniciar la máquina.

Mostrar la tabla de encaminamiento

- La información de la tabla de encaminamiento de una máquina se puede obtener con la orden `route` o con `ip` o con `netstat`.

- Con `route`:

```
pc1:~# route
Kernel IP routing table
Destination Gateway Genmask      Flags  Metric  Ref  Use Iface
10.0.0.0   *          255.255.255.0  U      0        0   0  eth0
```

- Con `ip`:

```
pc1:~# ip route show
10.0.0.0/24 dev eth0      proto kernel    scope  link    src 10.0.0.1
```

- Con `netstat`:

```
pc1:~# netstat -r
Kernel IP routing table
Destination Gateway Genmask      Flags  MSS  Window  irtt  Iface
10.0.0.0   *          255.255.255.0  U      0    0        0     eth0
```

- En cada ruta de la tabla, la interfaz (iface) que aparece se refiere a la interfaz de la máquina en la que se ejecuta la orden (pc1) por la que **saldrán** los paquetes que utilicen esa ruta.

Añadir una ruta en la tabla de encaminamiento

- Con `route`:

- Ruta a una máquina:

```
route add -host <máquinaDestino> gw <gateway>
```

```
pc1:~# route add -host 11.0.0.1 gw 10.0.0.1
```

- Ruta a una subred

```
route add -net <subredDestino> netmask <máscara> gw <gateway>
```

```
pc1:~# route add -net 12.0.0.0 netmask 255.255.255.0 gw 10.0.0.1
```

- Ruta por defecto

```
route add default gw <gateway>
```

```
pc1:~# route add default gw 10.0.0.2
```

- Con `ip`:

- Ruta a una máquina o a una subred:

```
ip route add <dirIP/máscara> via <gateway>
```

```
pc1:~# ip route add 12.0.0.0/24 via 10.0.0.1
```

- Ruta por defecto `ip route add default via <gateway>`

```
pc1:~# ip route add default via 10.0.0.2
```

- Los cambios realizados con estas órdenes no se conservan al reiniciar la máquina.

Borrar una ruta en la tabla de encaminamiento

- Con `route`:

- Ruta a una máquina:

```
route del -host <máquinaDestino>
```

```
pc1:~# route del -host 11.0.0.1
```

- Ruta a una subred

```
route del -net <subredDestino> netmask <máscara>
```

```
pc1:~# route del -net 12.0.0.0 netmask 255.255.255.0
```

- Ruta por defecto

```
route del default
```

```
pc1:~# route del default
```

- Con `ip`:

- Ruta a una máquina o a una subred:

```
ip route del <dirIP/máscara> via <gateway>
```

```
pc1:~# ip route del 12.0.0.0/24 via 10.0.0.1
```

- Ruta por defecto

```
ip route del default via <gateway>
```

```
pc1:~# ip route del default via 10.0.0.2
```

- Los cambios realizados con estas órdenes no se conservan al reiniciar la máquina.

Contenidos

- 1 NetGUI
- 2 Herramientas de configuración de la red: `ifconfig`, `ip`, `route`
- 3 Configuración de red mediante ficheros de configuración
- 4 Herramientas de diagnóstico de red: `arp`, `ping`, `tcpdump`, `wireshark`

Fichero de configuración de red

- Los cambios en la configuración de red realizados en el terminal con `ifconfig/ip/route` no se mantienen si se apaga y se vuelve a encender la máquina.
- Al arrancar una máquina su configuración de red por defecto se lee de un fichero de configuración.
- Dependiendo de la distribución de Linux, la configuración de red puede estar en un fichero o conjunto de ficheros diferentes.
 - En Debian y derivados (como Ubuntu) la configuración de red está en el fichero `/etc/network/interfaces`

Configuración de direcciones IP a través de /etc/network/interfaces

- Ejemplo de configuración de red en el fichero `/etc/network/interfaces`:

<code>auto lo</code>	→ la interfaz <code>lo</code> se configurará automáticamente al activar la red
<code>iface lo inet loopback</code>	→ la interfaz <code>lo</code> tendrá la dirección predefinida para la interfaz de loopback (<code>127.0.0.1</code>)
<code>auto eth0</code>	→ la interfaz <code>eth0</code> se configurará automáticamente al activar la red
<code>iface eth0 inet static</code>	→ la interfaz <code>eth0</code> tendrá una IP estática
<code> address 10.0.0.10</code>	→ dirección IP de <code>eth0</code>
<code> network 10.0.0.0</code>	→ dirección de la subred a la que pertenece <code>eth0</code> (opcional)
<code> netmask 255.255.255.0</code>	→ máscara de la subred a la que pertenece <code>eth0</code>
<code> broadcast 10.0.0.255</code>	→ dirección de broadcast de la subred a la que pertenece <code>eth0</code> (opcional)
<code> gateway 10.0.0.1</code>	→ ruta por defecto a través de <code>10.0.0.1</code> (opcional)

- Cuando se modifica este fichero es necesario reiniciar las interfaces de red para que la nueva configuración surta efecto, mediante la orden:
`/etc/init.d/networking restart`
- Puedes ver otros ejemplos de configuración de interfaces de red con:
`zless /usr/share/doc/ifupdown/examples/network-interfaces.gz`
- Puedes consultar el manual: `man interfaces`

Configuración de direcciones IP a través de `/etc/network/interfaces` en NetGUI

- Cuando se crea un escenario de red nuevo en NetGUI, la primera vez que se arranca una máquina virtual sólo tiene configurado el interfaz de loopback (`lo`).
- Para asignar en la máquina virtual direcciones IP a sus interfaces `eth0`, `eth1`... de forma que se conserven después de apagarla y volverla a encender, es necesario editar el fichero `/etc/network/interfaces` para añadirle las líneas que sean necesarias.
- No hay que olvidar reiniciar las interfaces de red cada vez que se modifica el fichero para que la nueva configuración tenga efecto:

```
pc1:~# /etc/init.d/networking restart
```

- Esta orden es equivalente a detener las interfaces de red y volver a activarlas:

```
pc1:~# /etc/init.d/networking stop  
pc1:~# /etc/init.d/networking start
```

Configuración de rutas a través de /etc/network/interfaces: Ejemplo

- Fichero `/etc/network/interfaces` incluyendo rutas:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.0.0.10
    network 10.0.0.0
    netmask 255.255.255.0
    broadcast 10.0.0.255
    up route add -net 12.0.0.0 netmask 255.255.255.0 gw 10.0.0.2
    up route add default gw 10.0.0.1
```

- Es equivalente poner:

```
up route add default gw 10.0.0.1
```

a poner:

```
gateway 10.0.0.1
```

- En la sección de una interfaz puede ponerse cualquier orden precedida por `up`: cuando se active esa interfaz se ejecutará la orden.
- También pueden ponerse órdenes precedidas por `down`: cuando se apague esa interfaz se ejecutará la orden.

Editar el fichero `/etc/network/interfaces` en NetGUI

- Dentro de las máquinas virtuales de NetGUI, puede usarse como editor `mcedit` o `vi`. Si no se conoce ninguno de los dos resulta más sencillo utilizar `mcedit`.
- Uso básico de `mcedit`:
 - La línea inferior muestra para qué sirve pulsar las teclas de función `F1` a `F10`.
 - `F2`: Guardar el fichero
 - `F10`: Salir del editor: si no se ha guardado el fichero, permite hacerlo en ese momento
 - En vez de pulsar una tecla de función, puede usarse el ratón sobre los atajos escritos en línea inferior.
- En el terminal de NetGUI para editar el fichero de configuración de la red escribe:

```
mcedit /etc/network/interfaces
```

Contenidos

- 1 NetGUI
- 2 Herramientas de configuración de la red: `ifconfig`, `ip`, `route`
- 3 Configuración de red mediante ficheros de configuración
- 4 Herramientas de diagnóstico de red: `arp`, `ping`, `tcpdump`, `wireshark`

Herramientas de diagnóstico de red

- **Diagnóstico de red:** Monitorizar el estado de conectividad a la red de las máquinas
- Herramientas que veremos en este tema:
 - `arp`
 - `ping`
 - `tcpdump`
 - `wireshark`

Cachés de ARP

- Para consultar la caché de ARP en una máquina se utiliza la orden `arp`:

```
pc2:~# arp -a
? (10.0.0.1) at 0A:29:92:55:93:70 [ether] on eth0
```

- Para añadir una entrada manual en una caché de ARP se puede usar la orden `arp` con la opción `-s` que asocia una dirección IP con una dirección Ethernet:

```
pc2:~# arp -s 10.0.0.2 01:02:03:04:05:06
? (10.0.0.1) at 0A:29:92:55:93:70 [ether] on eth0
? (10.0.0.2) at 01:02:03:04:05:06 [ether] PERM on eth0
```

- Para borrar una entrada de la caché de ARP se utiliza la orden `arp` con la opción `-d`:

```
pc2:~# arp -d 10.0.0.2
```

Comprobar la conectividad entre dos dispositivos: ping

- La orden `ping` permite comprobar si se puede alcanzar una máquina, y el tiempo que se tarda en ir y volver a ella (*round-trip time*, RTT).
- Envía un paquete cada segundo. La máquina destino contestará a cada uno de ellos con un paquete de respuesta.
- Por defecto `ping` se ejecuta indefinidamente. Hay que utilizar `Ctrl+C` para interrumpirlo.
- Tiene múltiples opciones, las más habituales son:
 - `-c <númeroPaquetes>`: número de paquetes que se envían.
 - `-s <tamañoPaquete>`: tamaño de los paquetes que se envían.
 - `-t <TTL>`: TTL de los paquetes que se envían (por defecto, 64).

ping: Ejemplo

```
pc2:~# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1): 56(84) bytes of data
64 bytes from 10.0.0.1: icmp_seq=0 ttl=64 time=1.896 ms
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=2.110 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=2.125 ms
^C
--- 10.0.0.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 1.896/2.044/2.125/0.105 ms
```

- Cuando se interrumpe el `ping`, aparece un resumen estadístico que contiene:
 - porcentaje de pérdidas
 - RTT mínimo, medio y máximo, y desviación media

Captura de tráfico de red: tcpdump

- Para capturar tráfico en una interfaz de red se puede utilizar la orden `tcpdump`.
- El tráfico que se captura puede verse directamente en el terminal mientras se va capturando, o puede guardarse en un fichero para analizarlo más tarde.
- `tcpdump` tiene varias opciones (véase `man tcpdump`).

Normalmente usaremos las siguientes opciones en las prácticas:

- i <dev> Interfaz en la que se quiere capturar tráfico
 - w <file> Fichero donde se guardarán los paquetes capturados, en vez de mostrarlos en pantalla
 - s <tamaño> Número de bytes que se capturan de cada paquete (por defecto 68 bytes, 0 para capturar paquetes enteros)
- En varias ocasiones resultará conveniente poner `-s 0` para asegurarse de que se capturan los paquetes enteros, aunque con esta opción el proceso de captura se ralentiza un poco más.

Captura de tráfico de red: tcpdump

- Arrancando `tcpdump` en modo normal la captura se realiza indefinidamente, y para interrumpirla es necesario usar `Ctrl+C`.
- En ocasiones resulta más conveniente arrancar `tcpdump` en segundo plano (*background*), lo que se hace añadiendo `&` al final de la orden:

```
pc1:~# tcpdump -i eth0 -s 0 &
```

- Esto permite escribir otras órdenes en la terminal después del `tcpdump`.
- Para interrumpir la captura cuando se está realizando en *background* es necesario:
 - pasar `tcpdump` a primer plano (*foreground*) con la orden `fg`:

```
pc1:~# fg
```

- usar `Ctrl+C`

Captura de tráfico en NetGUI: acceso al sistema de ficheros de la máquina real

- Una vez realizada una captura de tráfico redirigida a un fichero con `-w` se pueden visualizar los paquetes capturados con la herramienta `wireshark`.
- En las aulas de prácticas, `wireshark` está instalado fuera de las máquinas virtuales, es decir, en la máquina real.
- Dentro de una máquina virtual de NetGUI, escribir en el directorio `/hosthome` permite guardar ficheros en la máquina real: todos los ficheros grabados en el directorio `/hosthome` estarán en realidad en el `$HOME` del usuario en la máquina real.
- Las capturas realizadas en las máquinas virtuales conviene guardarlas en `/hosthome` para que sean accesibles desde la máquina real.
- Ejemplo:

```
pc1:~# tcpdump -i eth0 -s 0 -w /hosthome/pc1.eth0.cap
```

- Ejemplo lanzando `tcpdump` en *background*:

```
pc1:~# tcpdump -i eth0 -s 0 -w /hosthome/pc1.eth0.cap &
```

wireshark

- **wireshark** es una herramienta gráfica que permite visualizar paquetes capturados, navegando a través de los campos de cabecera y datos de cada uno de los protocolos utilizados.
- Puede arrancarse **wireshark** desde un terminal de la máquina real (por ejemplo *beta25*) de la siguiente forma:

```
usuario@beta25:~$ wireshark pc1.eth0.cap
```

wireshark

Resumen de los paquetes capturados

Detalle de las cabeceras del paquete seleccionado

Contenido del paquete seleccionado en hexadecimal y ASCII

The screenshot displays the Wireshark interface for a file named 'ftp.cap'. The main packet list pane shows the following data:

No.	Time	Source	Destination	Protocol	Info
6	0.012926	127.0.0.1	127.0.0.1	FTP	Response: 220 ProFTPD 1.3.0 Ser
7	0.012944	127.0.0.1	127.0.0.1	TCP	1900 > ftp [ACK] Seq=1 Ack=55 Win=
8	3.075341	127.0.0.1	127.0.0.1	FTP	Request: USER prueba
9	3.075382	127.0.0.1	127.0.0.1	TCP	ftp > 1900 [ACK] Seq=55 Ack=14 Win=
10	3.092710	127.0.0.1	127.0.0.1	FTP	Response: 331 Password required for
11	3.092904	127.0.0.1	127.0.0.1	TCP	1900 > ftp [ACK] Seq=14 Ack=90 Win=
12	6.957003	127.0.0.1	127.0.0.1	FTP	Request: PASS prueba-pass
13	6.994821	127.0.0.1	127.0.0.1	TCP	ftp > 1900 [ACK] Seq=90 Ack=32 Win=

The details pane for the selected packet (Frame 6) shows the following structure:

- Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
- Internet Protocol, Src: 127.0.0.1 (127.0.0.1), Dst: 127.0.0.1 (127.0.0.1)
- Transmission Control Protocol, Src Port: ftp (21), Dst Port: 1900 (1900), Seq: 1, Ack: 1, Len: 54
 - Source port: ftp (21)
 - Destination port: 1900 (1900)
 - Sequence number: 1 (relative sequence number)
 - [Next sequence number: 55 (relative sequence number)]
 - Acknowledgement number: 1 (relative ack number)
 - Header Length: 20 bytes

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```

0000  00 00 00 00 00 00 00 00 00 00 00 08 00 45 00  .....E.
0010  00 6a 60 29 40 00 40 06 dc 62 7f 00 00 01 7f 00  .j')@. .b.....
0020  00 01 00 15 07 6c 47 50 9b a2 48 21 3a 06 80 18  .....LGP ..H!...
0030  20 00 fe 5e 00 00 01 01 08 0a 06 41 fb 8d 06 41  ^.....A...A
0040  fb 8b 32 32 32 30 50 72 6f 46 54 50 44 20 31 2e  ..220 Pr ofTPD 1.
0050  32 32 30 50 72 6f 46 54 50 44 20 31 2e 30 50 72 6f  2.0 Serv es (Doh
File: /home... P: 42 D: 42 M: 0
  
```