

Repaso NetGUI Arquitectura de Internet

Sistemas Telemáticos para Medios Audiovisuales

Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación

Universidad Rey Juan Carlos

Septiembre de 2021



©2021 GSyC
Algunos derechos reservados.
Este trabajo se distribuye bajo la licencia
Creative Commons Attribution Share-Alike
disponible en <http://creativecommons.org/licenses/by-sa/4.0/>

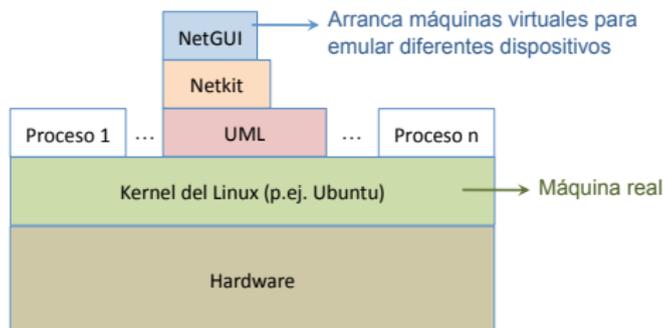
- 1 NetGUI
- 2 Las máquinas virtuales dentro de NetGUI
- 3 Herramientas de configuración de direcciones IP en Linux:
`ifconfig`, `ip`
- 4 Configuración de red mediante ficheros de configuración
- 5 Herramientas de configuración de la red: `route`
- 6 Configuración de rutas mediante ficheros de configuración
- 7 Herramientas de diagnóstico de red: `arp`, `ping`, `tracert`

Contenidos

- 1 NetGUI
- 2 Las máquinas virtuales dentro de NetGUI
 - Interfaces de red
 - Captura de tráfico de red: tcpdump y wireshark
- 3 Herramientas de configuración de direcciones IP en Linux: ifconfig, ip
- 4 Configuración de red mediante ficheros de configuración
- 5 Herramientas de configuración de la red: route
- 6 Configuración de rutas mediante ficheros de configuración
- 7 Herramientas de diagnóstico de red: arp, ping, traceroute

- **NetGUI** es una 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 (hubs) 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>

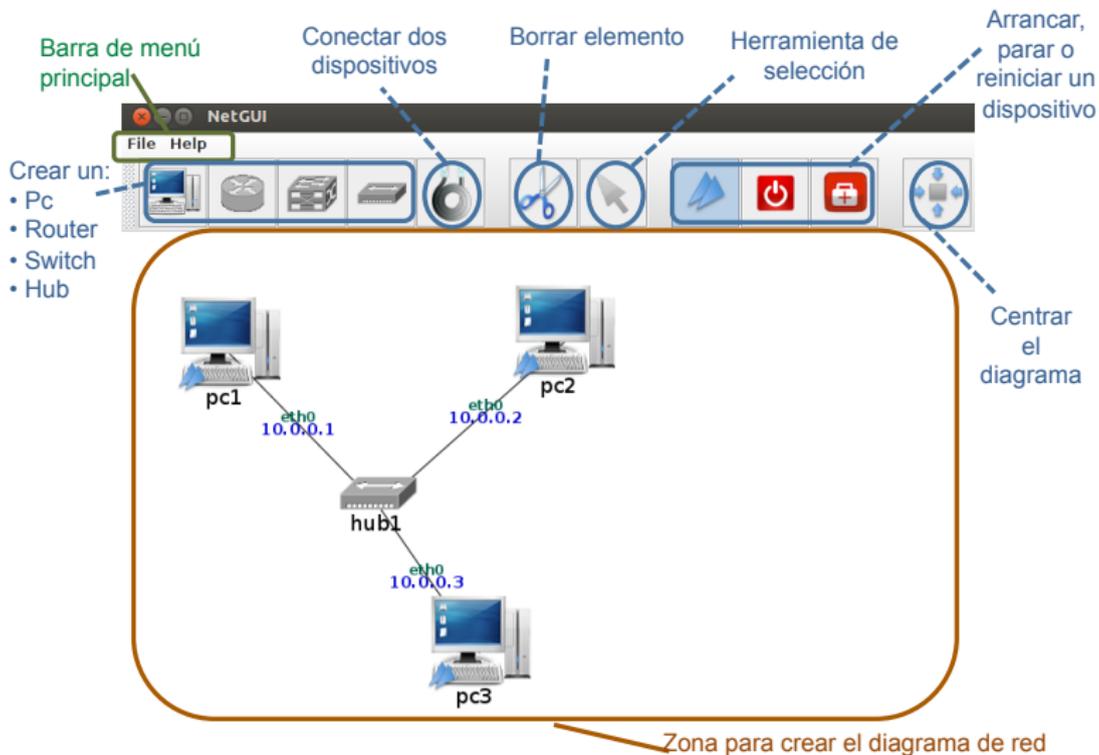
NetGUI, Netkit y UML



- **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 aquélla en la que se están ejecutando los procesos UML.

La interfaz gráfica

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



Creación/Borrado de dispositivos y su interconexión

- Los dispositivos con los que se puede trabajar en los escenarios de NetGUI son los siguientes: PC o máquina final, router, switch y hub. Para dibujarlos hay que pulsar sobre el botón que queramos utilizar y pinchar en el fondo de la zona de dibujo.



- Para conectar estos dispositivos utilizaremos el botón que representa el cable. Una vez seleccionado este botón, pulsaremos una vez sobre el primer dispositivo que queremos conectar y una segunda vez sobre el segundo dispositivo:



- Para borrar cualquier elemento que hayamos dibujado seleccionaremos el botón que muestra las tijeras y a continuación pulsaremos sobre el elemento a borrar, ya sea dispositivo o cable.



Iniciar/Para/Reiniciar la ejecución de los dispositivos

- Los hubs **no hay que arrancarlos ni pararlos**, se encuentran arrancados siempre.
- Para arrancar los dispositivos: PC (máquina final que no es un *router*), *router* o *switch*, es necesario seleccionar el botón de arranque y pulsar sobre el dispositivo concreto a arrancar. Al iniciarlo, aparecerá una ventana que muestra la consola para poder ejecutar comandos dentro de dicho dispositivo:



- Para interrumpir la ejecución de un dispositivo es necesario seleccionar el botón de parada y pulsar sobre el dispositivo concreto que deseamos parar:



- Si alguna máquina no ha arrancado bien y/o comienzan a salir de forma continuada mensajes de error en su consola, primero conviene intentar pararla y luego volverla a arrancar con los dos botones anteriores.

La herramienta de selección

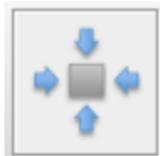
- La herramienta de selección permite la siguiente funcionalidad:



- **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.
- **Poner en primer plano la consola de un dispositivo arrancado:** 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 sobre el fondo de la ventana:
 - 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.
- Para guardar con **File->Save**, la primera vez hay que elegir un **nombre de carpeta que no exista**. 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.
- **No se pueden guardar escenarios en un *path* que incluya un directorio en cuyo nombre haya algún espacio en blanco**. Todas las carpetas desde el *HOME* hasta la del escenario deben tener **NOMBRES SIN ESPACIOS**.
- Al guardar un escenario simplemente se guardan los cambios de la figura en el archivo **netgui.nkp**. El estado de los ficheros de cada máquina virtual se va guardando automáticamente en los ficheros **.disk**.

Consolas de pcs/routers/switches

The image shows a NetGUI interface for a lab named 'lab-intro'. On the left, a network topology is displayed with nodes: pc1, pc2, hub1, r1, hub2, and pc3. Connections are shown between pc1 and hub1 (eth0), hub1 and pc2 (eth0), hub1 and r1 (eth0), r1 and hub2 (eth1), and hub2 and pc3 (eth0). On the right, four terminal windows are shown, each connected to a specific node in the topology by dashed arrows:

- Consola de pc1:** Shows the output of 'Netkit phase 2 initialization terminated' and an automatic login for root on tty1.
- Consola de pc2:** Shows the output of 'Netkit phase 2 initialization terminated' and an automatic login for root on tty2.
- Consola de r1:** Shows the output of 'Netkit phase 2 initialization terminated' and an automatic login for root on tty1.
- Consola de pc3:** Shows the output of 'Netkit phase 2 initialization terminated' and an automatic login for root on tty1.

- No hay una consola para los hubs, se encuentran siempre arrancados y configurados.

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 utilizando el botón  sobre cada una de ellas.
 - 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.
- Por lo tanto, el procedimiento adecuado para salir de NetGUI es:
 - 1 Apagar una a una las máquinas virtuales mediante el botón  sobre cada una de ellas.
 - 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 Las máquinas virtuales dentro de NetGUI
 - Interfaces de red
 - Captura de tráfico de red: tcpdump y wireshark
- 3 Herramientas de configuración de direcciones IP en Linux: `ifconfig`, `ip`
- 4 Configuración de red mediante ficheros de configuración
- 5 Herramientas de configuración de la red: `route`
- 6 Configuración de rutas mediante ficheros de configuración
- 7 Herramientas de diagnóstico de red: `arp`, `ping`, `traceroute`

Interfaces de red de una máquina Linux

- Una máquina Linux (pc) que tenga una tarjeta Ethernet tiene definida la interfaz `eth0`. En la figura `eth0` queda representada con la tarjeta de red que conecta pc1 y hub1.



Ejecución de comandos

- Para ejecutar un comando en una máquina virtual, escribimos dicho comando sobre la consola de esa determinada máquina. Por ejemplo:
 - El comando `ifconfig` o el comando `ip` permiten ver información relacionada con las interfaces de red una máquina.
 - Con `ifconfig` (se ha coloreado la información importante relativa a Ethernet) en `pc1`:

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

- También con el comando `ip` en `pc1`:

```
pc1:~# ip address show eth0
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 212.128.4.100/24brd 212.128.4.255 scope global eth0
```

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:

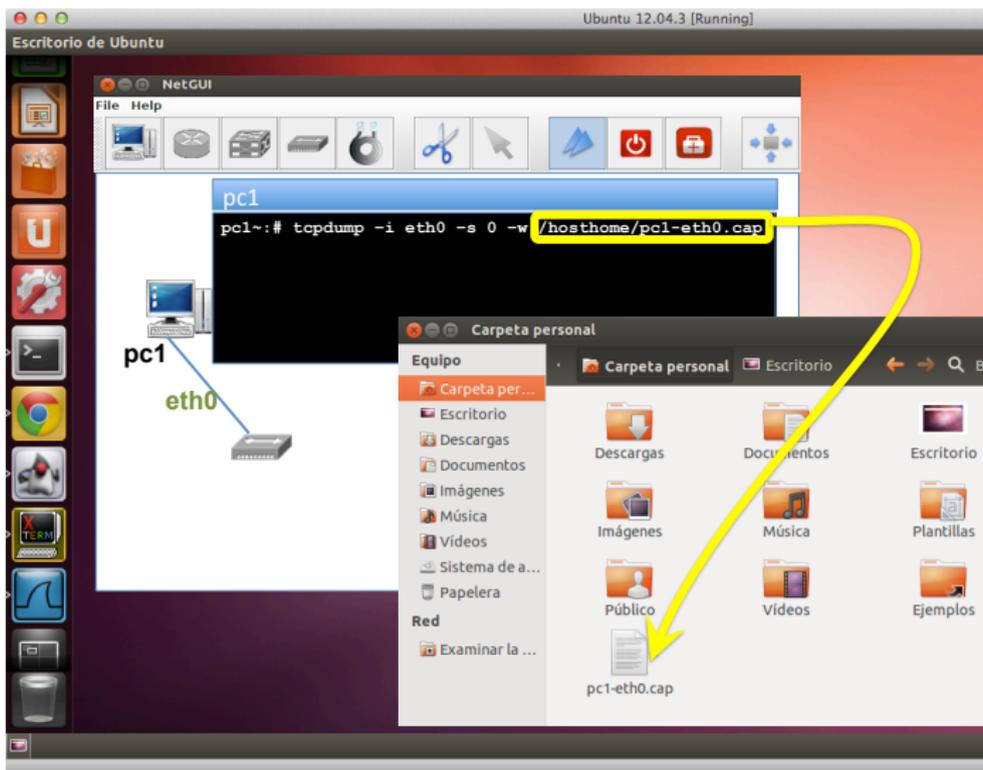
- i <dev> Interfaz en la que se quiere capturar tráfico
 - w <file> Fichero donde se guardarán los paquetes capturados, en vez de mostrarlos en pantalla
 - s <tamaño> Número de bytes que se capturan de cada paquete (por defecto 68 bytes, `-s 0` para capturar paquetes enteros)
- Para interrumpir `tcpdump` es necesario pulsar `Ctrl+C`.

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

- 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` en la máquina virtual estarán en realidad en la **Carpeta personal** 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
```

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



Captura de tráfico de red: tcpdump en *background*

- Si arrancamos tcpdump como hemos descrito previamente, la consola donde arrancamos tcpdump se queda ocupada con dicho programa y no podremos utilizarla para ejecutar otros comandos hasta que no interrumpamos tcpdump con Ctrl+C.
- En ocasiones queremos ejecutar otros comandos en una consola a la vez que realizamos una captura de tráfico. En estos casos 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 -w /home/pc1-eth0.cap &
```

- De esta forma tcpdump se ejecuta, pero además es posible escribir otras órdenes en la consola después de tcpdump.
- Para interrumpir la captura cuando se está realizando en *background* es necesario:

- 1 pasar tcpdump a primer plano (*foreground*) con la orden fg:

```
pc1:~# fg
```

- 2 pulsar Ctrl+C

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.
 - Debido a que las máquinas de NetGUI no tienen entorno gráfico instalado, no es posible arrancar **wireshark** dentro de las máquinas virtuales y es necesario arrancarlo desde la máquina real.
- Puede arrancarse **wireshark** desde un terminal de la máquina real (por ejemplo en la máquina *zeta25*) de la siguiente forma:

```
usuario@zeta25:~$ 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

ftp.cap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: + Expression... Limpicar Aplicar

| 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= |

Frame 6 (120 bytes on wire, 96 bytes captured)

- 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

```

0000  00 00 00 00 00 00 00 00 00 00 00 08 00 45 00  .....E.
0010  00 6a 60 29 40 00 04 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 20 50 72 6f 46 54 50 44 20 31 2e  ..220 Pr ofTPD 1.
0050  32 32 30 20 53 65 73 76 65 73 30 20 44 65 63 60  2.0 Serv es (Doh
File: /home... P: 42 D: 42 M: 0

```

Contenidos

- 1 NetGUI
- 2 Las máquinas virtuales dentro de NetGUI
 - Interfaces de red
 - Captura de tráfico de red: tcpdump y wireshark
- 3 Herramientas de configuración de direcciones IP en Linux:
`ifconfig`, `ip`
- 4 Configuración de red mediante ficheros de configuración
- 5 Herramientas de configuración de la red: `route`
- 6 Configuración de rutas mediante ficheros de configuración
- 7 Herramientas de diagnóstico de red: `arp`, `ping`, `traceroute`

Configuración de direcciones IP

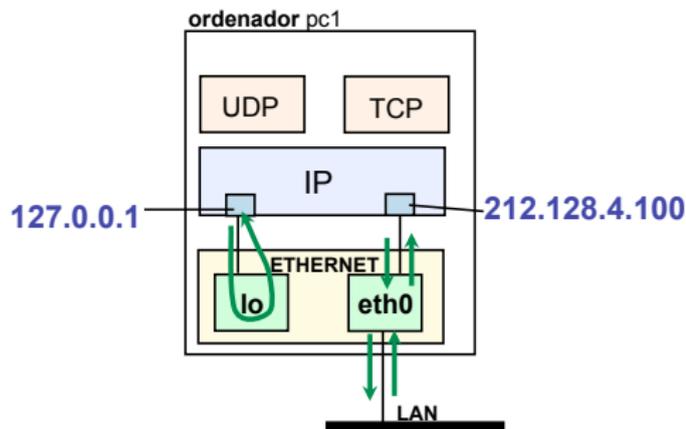
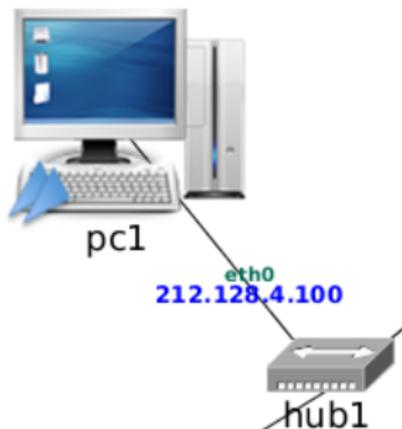
- **Configuración de red:** Añadir/eliminar/modificar direcciones IP.
- Órdenes que se utilizan:
 - `ifconfig`
 - `ip`

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`.

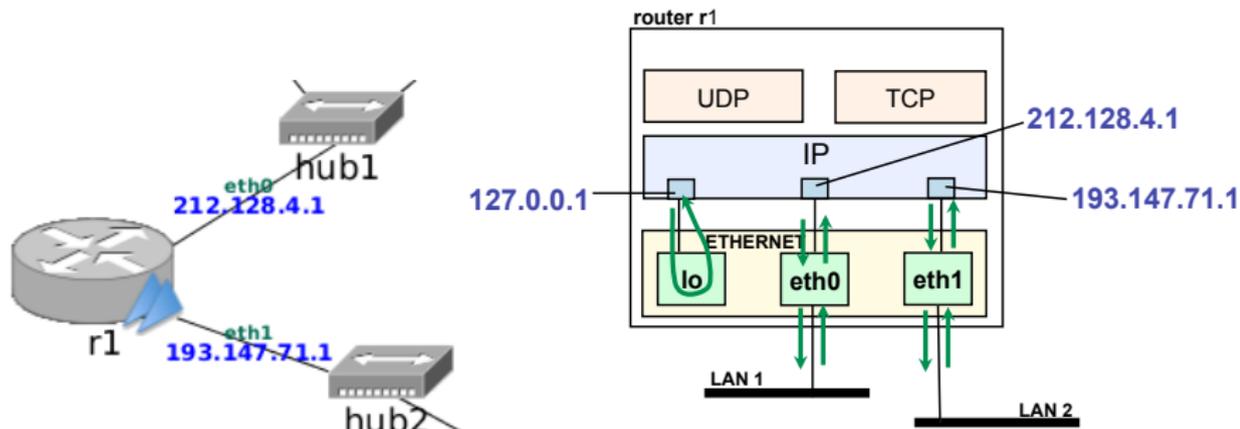
Interfaces de red y direcciones IP (I)

- A cada interfaz de red se le asigna una dirección IP
- A la interfaz de *loopback* se le suele asignar la dirección IP 127.0.0.1 o la 127.0.1.1
- 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:Locap Loopback
          inet addr:127.0.0.1 Mask: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 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 11.0.0.1 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 11.0.0.1/24 broadcast +
```

- Después de añadir 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.**

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).
- **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 11.0.0.1/24
```

- Después de 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.**

Contenidos

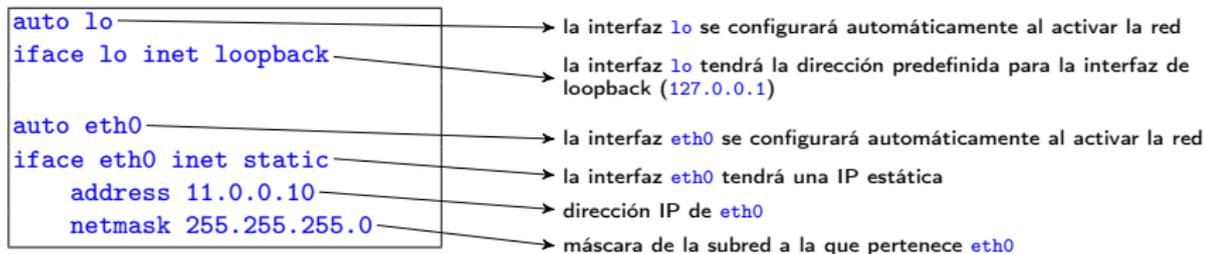
- 1 NetGUI
- 2 Las máquinas virtuales dentro de NetGUI
 - Interfaces de red
 - Captura de tráfico de red: tcpdump y wireshark
- 3 Herramientas de configuración de direcciones IP en Linux: `ifconfig`, `ip`
- 4 Configuración de red mediante ficheros de configuración**
- 5 Herramientas de configuración de la red: `route`
- 6 Configuración de rutas mediante ficheros de configuración
- 7 Herramientas de diagnóstico de red: `arp`, `ping`, `traceroute`

Fichero de configuración de red

- Los cambios en la configuración de red realizados en el terminal con `ifconfig/ip` 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

- Para editar el fichero de configuración de red escribe en la máquina virtual: `nano /etc/network/interfaces`
- Ejemplo de configuración del fichero:



- 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 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
```

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

- Dentro de las máquinas virtuales de NetGUI, puede usarse como editor `nano`, `mcedit` o `vi`. Si no se conoce ninguno de ellos, quizá el más sencillo es `nano`.
- En el terminal de NetGUI para editar el fichero de configuración de la red escribe:

```
nano /etc/network/interfaces
```

- Uso básico de `nano`:
 - La línea inferior muestra algunos atajos de teclado útiles, el símbolo `^` representa a la tecla CTRL.
 - Atajos más importantes:
 - **F2** o **Ctrl-X**: Salir del editor. Preguntará si se quiere guardar los cambios, contestar con **Y** y confirmar el nombre de fichero con **INTRO**.
 - **F3** o **Ctrl-O**: Guardar los cambios del fichero. Hay que confirmar el nombre del fichero con **INTRO**.
 - **F9**: Cortar la línea del cursor
 - **F10**: Pegar la última línea cortada
 - Además, puedes hacer **clic en el botón central del ratón** para que se pegue en el lugar del cursor el texto seleccionado en cualquier ventana: Esta atajo es global en Linux y suele funcionar casi en cualquier situación.

Contenidos

- 1 NetGUI
- 2 Las máquinas virtuales dentro de NetGUI
 - Interfaces de red
 - Captura de tráfico de red: tcpdump y wireshark
- 3 Herramientas de configuración de direcciones IP en Linux: `ifconfig`, `ip`
- 4 Configuración de red mediante ficheros de configuración
- 5 Herramientas de configuración de la red: `route`**
- 6 Configuración de rutas mediante ficheros de configuración
- 7 Herramientas de diagnóstico de red: `arp`, `ping`, `traceroute`

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
11.0.0.0    *           255.255.255.0  U      0      0   0   eth0
```

- Con `ip`:

```
pc1:~# ip route show
11.0.0.0/24 dev eth0      proto kernel    scope link    src 11.0.0.1
```

- Con `netstat`:

```
pc1:~# netstat -r
Kernel IP routing table
Destination Gateway Genmask          Flags MSS Window  irtt  Iface
11.0.0.0    *           255.255.255.0  U      0      0      0     eth0
```

Configuración por defecto de la tabla de encaminamiento (I)

- Si una máquina no tiene asignada ninguna dirección IP, la tabla de encaminamiento estará vacía.
- Al asignar una dirección IP a una máquina, **automáticamente** se añade una entrada en la tabla de encaminamiento para que dicha máquina se pueda comunicar con las máquinas que están directamente conectadas a dicha subred.

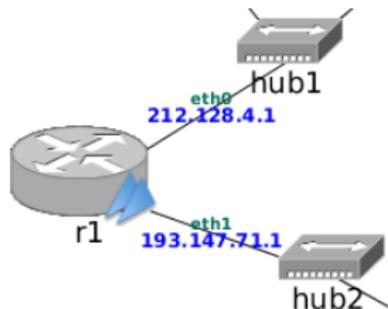


```
pc1:~# route
Kernel IP routing table
Destination Gateway Genmask      Flags Metric Ref Use Iface
212.128.4.0 *          255.255.255.0 U        0      0   0 eth0
```

* : Es equivalente a 0.0.0.0

Configuración por defecto de la tabla de encaminamiento (II)

- Si una máquina tiene asignadas varias direcciones IP, **automáticamente** tendrá configuradas en su tabla de encaminamiento tantas entradas como subredes a las que esté conectada directamente dicha máquina.



```
pc1:~# route
Kernel IP routing table
Destination Gateway Genmask          Flags Metric Ref Use Iface
212.128.4.0 *          255.255.255.0 U        0      0  0 eth0
193.147.71.0 *          255.255.255.0 U        0      0  0 eth1
```

- * : Es equivalente a 0.0.0.0
- 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 (r1) 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 12.0.0.1 gw 11.0.0.1
```

- 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 11.0.0.1
```

- Ruta por defecto

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

```
pc1:~# route add default gw 11.0.0.2
```

- 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 11.0.0.1
```

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

```
pc1:~# ip route add default via 11.0.0.2
```

- 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 12.0.0.1
```

- 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 11.0.0.1
```

- Ruta por defecto

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

```
pc1:~# ip route del default via 11.0.0.2
```

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

Contenidos

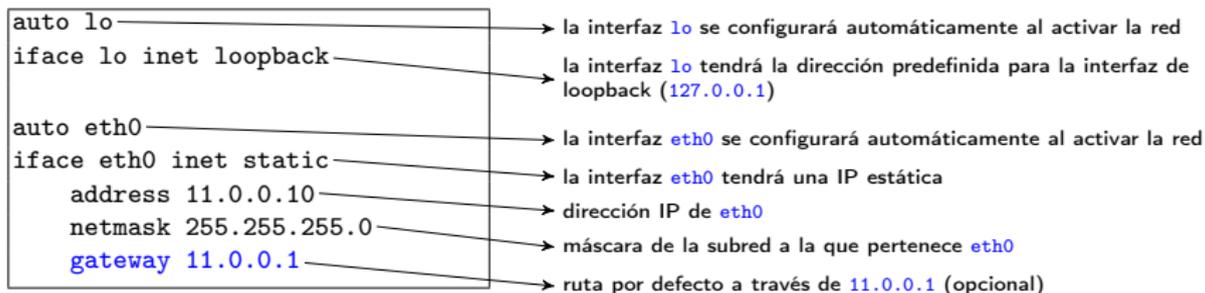
- 1 NetGUI
- 2 Las máquinas virtuales dentro de NetGUI
 - Interfaces de red
 - Captura de tráfico de red: tcpdump y wireshark
- 3 Herramientas de configuración de direcciones IP en Linux: `ifconfig`, `ip`
- 4 Configuración de red mediante ficheros de configuración
- 5 Herramientas de configuración de la red: `route`
- 6 Configuración de rutas mediante ficheros de configuración
- 7 Herramientas de diagnóstico de red: `arp`, `ping`, `traceroute`

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`

Ruta por defecto en `/etc/network/interfaces`

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



- 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 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 11.0.0.10
    netmask 255.255.255.0
    up route add -net 12.0.0.0 netmask 255.255.255.0 gw 11.0.0.2
    up route add default gw 11.0.0.1
```

- Es equivalente poner:
`up route add default gw 11.0.0.1`
a poner:
`gateway 11.0.0.1`
- 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.

Contenidos

- 1 NetGUI
- 2 Las máquinas virtuales dentro de NetGUI
 - Interfaces de red
 - Captura de tráfico de red: tcpdump y wireshark
- 3 Herramientas de configuración de direcciones IP en Linux: `ifconfig`, `ip`
- 4 Configuración de red mediante ficheros de configuración
- 5 Herramientas de configuración de la red: `route`
- 6 Configuración de rutas mediante ficheros de configuración
- 7 Herramientas de diagnóstico de red: `arp`, `ping`, `traceroute`

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`
 - `tracert`

Cachés de ARP

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

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

- Para borrar la caché de ARP:
 - Pasados unos 10 minutos de la última vez que se consultó una entrada, esta entrada se borra
 - Si se apaga y enciende una interfaz de red, se borran todas las entradas aprendidas por esa interfaz:

```
pc2:~# ifconfig eth0 down
pc2:~# ifconfig eth0 up
```

- Puede borrarse manualmente una entrada concreta con la orden `arp` mediante la opción `-d`:

```
pc2:~# arp -d 11.0.0.2
```

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 muchas opciones, las más habituales son:
 - `-c <númeroPaquetes>`: número de paquetes a enviar en vez de ejecutarse indefinidamente. que se envían. que se envían.
 - `-t <TTL>`: TTL inicial de los paquetes que se envían (por defecto, 64).

ping: Ejemplo

```
pc2:~# ping 11.0.0.1
PING 11.0.0.1 (11.0.0.1): 56(84) bytes of data
64 bytes from 11.0.0.1: icmp_seq=0 ttl=64 time=1.896 ms
64 bytes from 11.0.0.1: icmp_seq=1 ttl=64 time=2.110 ms
64 bytes from 11.0.0.1: icmp_seq=2 ttl=64 time=2.125 ms
^C
--- 11.0.0.1 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 (*round-trip time*, “tiempo de ronda”, es decir, tiempo en ir y volver al destino) mínimo, medio y máximo, y desviación media

Comprobar la ruta desde un origen a un destino: traceroute

- Envía paquetes UDP con puerto destino 33435, variando el TTL.
- Comienza enviando 3 paquetes con TTL=1, cuando obtiene alguna respuesta (ICMP: `time exceeded`) aumenta a TTL=2 y así sucesivamente hasta que obtiene la respuesta `UDP port unreachable`
- Cada vez que se obtiene una respuesta se imprime información de la máquina que envió dicha respuesta y el RTT con dicha máquina.
- NOTA: Consulta los detalles completos del funcionamiento del traceroute en las transparencias de teoría.

traceroute: Ejemplo

```
pc4:~# traceroute -n 11.0.0.1
traceroute to 11.0.0.1 (11.0.0.1), 64 hops max, 40 byte packets
 1 14.0.0.1 (14.0.0.1) 2.3 ms 3.3 ms 1.8 ms
 2 13.0.0.1 (13.0.0.1) 4.7 ms 5.6 ms 4.8 ms
 3 12.0.0.1 (12.0.0.1) 6.3 ms 8.3 ms 7.6 ms
 4 15.0.0.1 (15.0.0.1) 8.9 ms 10.5 ms 9.8 ms
 5 11.0.0.1 (11.0.0.1) 11.3 ms 10.3 ms 11.7 ms
```