

# Introducción a NetGUI

## ARO/AI

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

Enero de 2018



(cc) 2017-2018 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/3.0/es>

- 1 NetGUI
- 2 Las máquinas virtuales dentro de NetGUI

# Contenidos

## 1 NetGUI

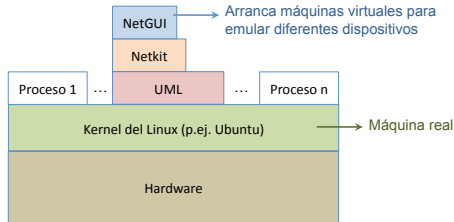
## 2 Las máquinas virtuales dentro de NetGUI

- Interfaces de red
- Captura de tráfico de red: tcpdump y wireshark

# NetGUI

- **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.gsyc.es/netgui>

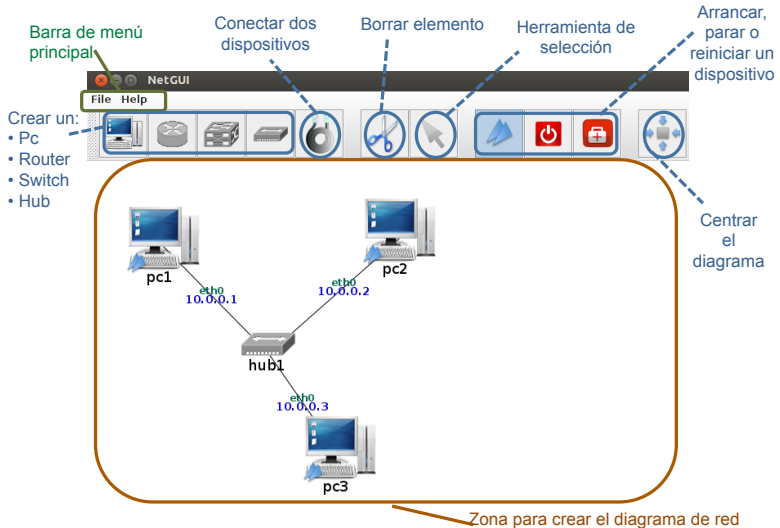
# NetGUI, Netkit y UML



- **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:



NOTA: En este curso no utilizaremos el dispositivo *switch*.

- 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.
- Si aún así la máquina sigue sin responder, podemos seleccionar el botón reiniciar y pulsar sobre la máquina en cuestión, que se reiniciará a los 5 segundos:



# 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

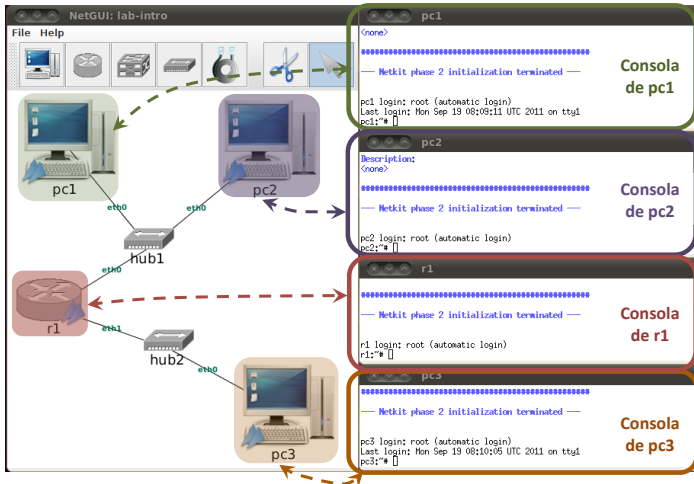
- **Mover toda la figura:** pulsando y arrastrando con el botón **izquierdo** del ratón sobre el fondo de la ventana (en un lugar en el que no haya ningún elemento).
- **Zoom:** pulsando y arrastrando con el botón **derecho** del ratón sobre el fondo de la ventana:
  - arrastrando hacia la derecha: aumentar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





- 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

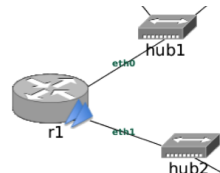
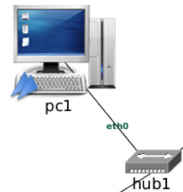
## 2 Las máquinas virtuales dentro de NetGUI

- Interfaces de red
- Captura de tráfico de red: tcpdump y wireshark



# 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, esta interfaz no se muestra en las figuras del escenario.
- Una máquina Linux que tenga una tarjeta Ethernet tiene, además de la interfaz **lo**, la interfaz **eth0**. En la figura **eth0** queda representada con la tarjeta de red que conecta pc1 y hub1.
- Un *router* Linux que tenga dos tarjetas Ethernet tendrá, además de la interfaz **lo**, dos interfaces eth: **eth0** y **eth1**. En la figura **eth0** queda representada con la tarjeta de red que conecta r1 y hub1 y **eth1** queda representada con la tarjeta de red que conecta r1 y hub2.



# 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      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)
```

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

```
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 212.128.4.100/24brd 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)

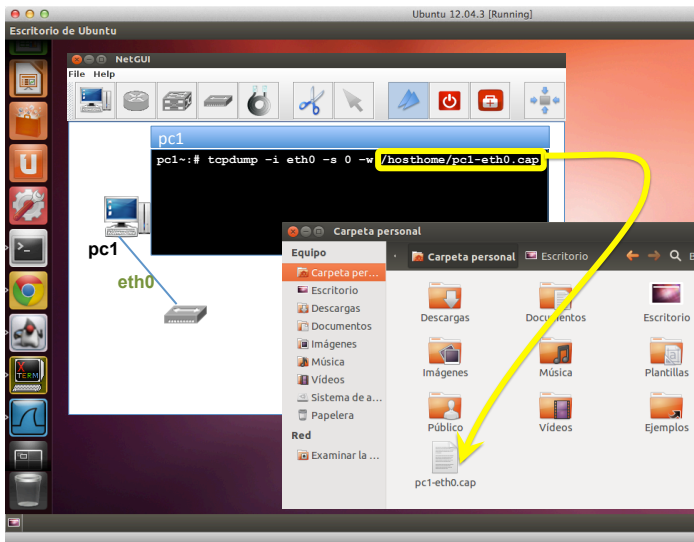
- Para interrumpir `tcpdump` es necesario pulsar `Ctrl+C`.

# 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 /hosthome/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 arrcarlo 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 pci-eth0.cap
```

## wireshark

PANEL 1:  
Lista de los  
paquetes  
capturados, y  
resumen de lo  
que contiene cada  
uno

PANEL 2:  
Detalles de todos  
los protocolos y  
cabeceras del  
paquete  
seleccionado en el  
panel 1

PANEL 3:  
Contenidos en  
hexadecimal y texto  
del paquete  
seleccionado en el  
panel 1 (resaltando el  
campo seleccionado  
en el panel 2)

