Material Docente en abierto de la Universidad Rey Juan Carlos

Apuntes de Arquitectura de Redes de Ordenadores: Manual de NetGUI

1º Ingeniería Telemática, 1º Ingeniería en Tecnologías de la Telecomunicación, 1º Ingeniería en Sistemas de Telecomunicación

Material disponible en BURJC Digital: https://burjcdigital.urjc.es

Autores: Eva M. Castro Barbero, José Centeno González, Pedro de las Heras Quirós
{eva.castro, jose.centeno, pedro.delasheras}@urjc.es

Curso 2023/24

Universidad Rey Juan Carlos

Índice

Tema 1: Introducción a NetGUI

Arquitectura de Redes de Ordenadores

1º Ingeniería Telemática, 1º Ingeniería en Tecnologías de la Telecomunicación, 1º Ingeniería en Sistemas de Telecomunicación

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

Diciembre 2023

Contenidos

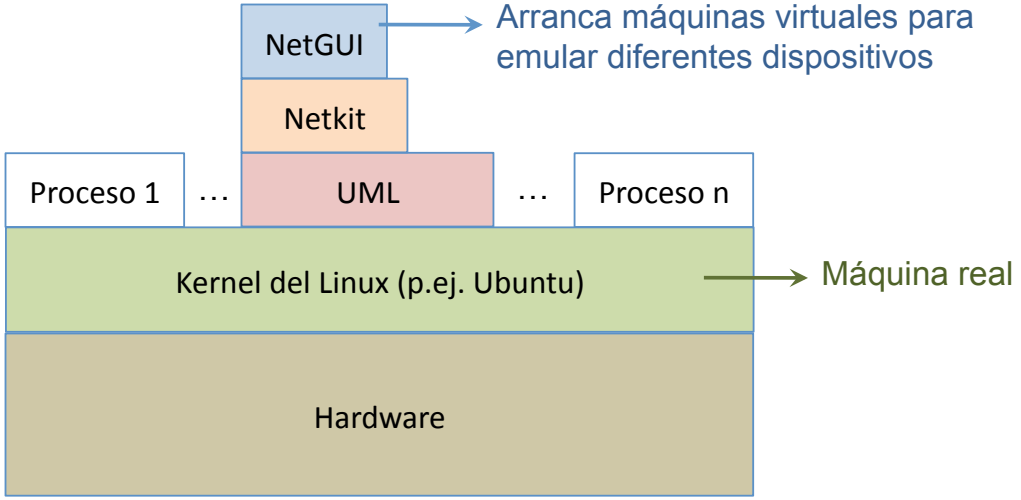
Contenidos

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:

  Página de 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.

Instalación

- En Ubuntu añade el repositorio de paquetes de los laboratorios Linux de la ETSIT:

```
wget -qO - https://labs.etsit.urjc.es/repo/3515B95F.key | sudo apt-key add -

echo deb https://labs.etsit.urjc.es/repo/ `lsb_release -c -s` main | \
sudo tee /etc/apt/sources.list.d/lablinux-etsit.list > /dev/null

sudo apt-get update
```

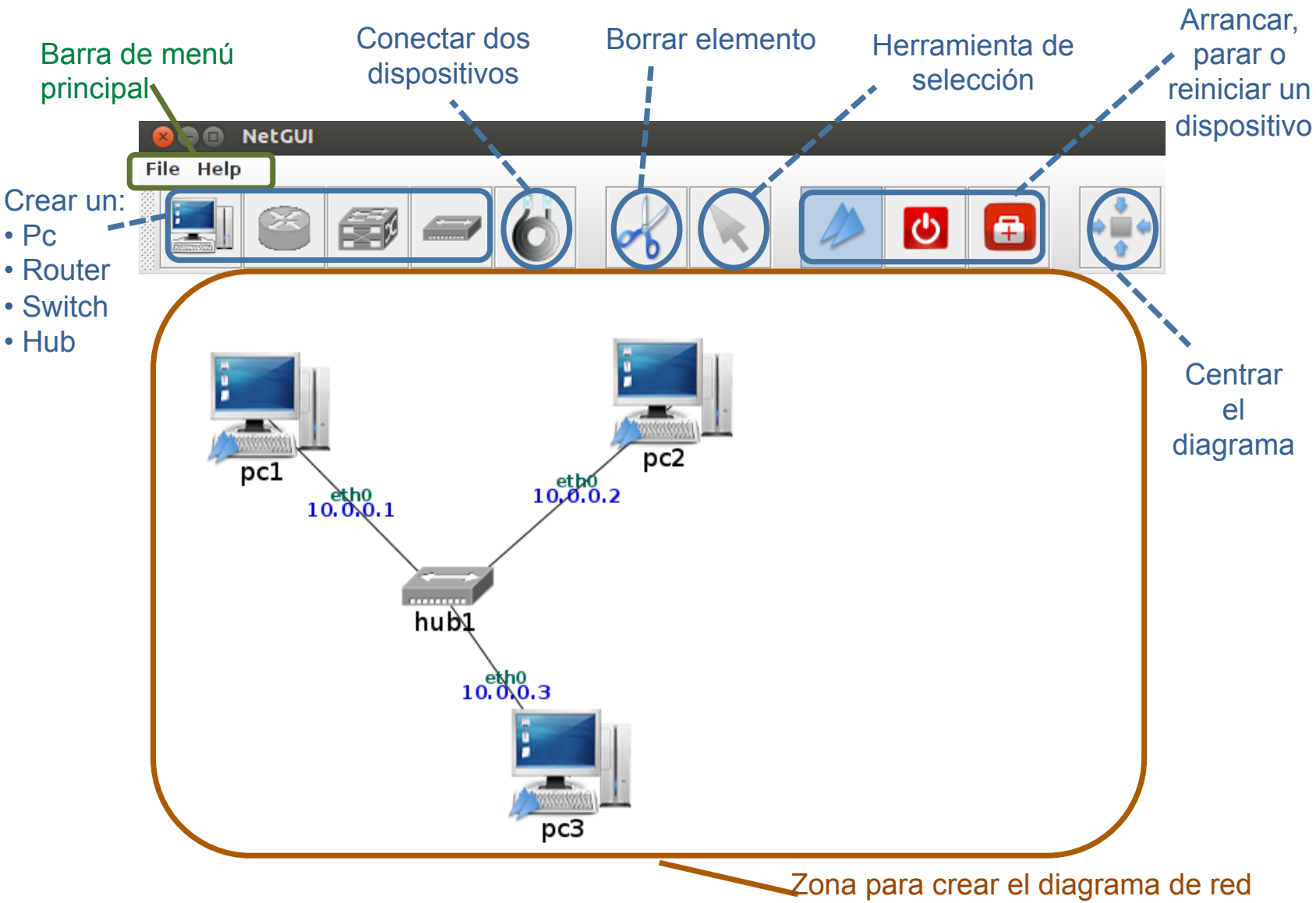
E instala el paquete de NetGUI:

```
sudo apt-get install netgui
```

- Descarga el fichero de VirtualBox que contiene una instalación de Ubuntu y NetGUI. Sigue las instrucciones:
  - Página de los laboratorios Linux para el uso de VirtualBox
- También puedes usar el acceso remoto por VNCweb para ejecutar NetGUI en las máquinas del laboratorio desde casa. Sigue las instrucciones:
  - Página de los laboratorios Linux para el uso de VNC Web

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

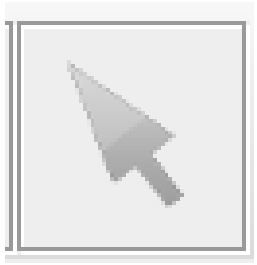
- Los hubs **no hay que arrancarlos ni pararlos**, se encuentran arrancados siempre.
- Para arrancar los dispositivos: PC (máquina final que no es un *router*), *router* o *switch*, es necesario seleccionar el botón de arranque y pulsar sobre el dispositivo concreto a arrancar. Al iniciarlo, aparecerá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

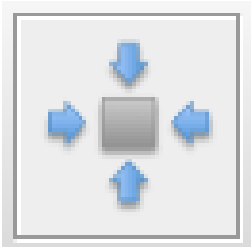
- La herramienta de selección permite la siguiente funcionalidad:



  - **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

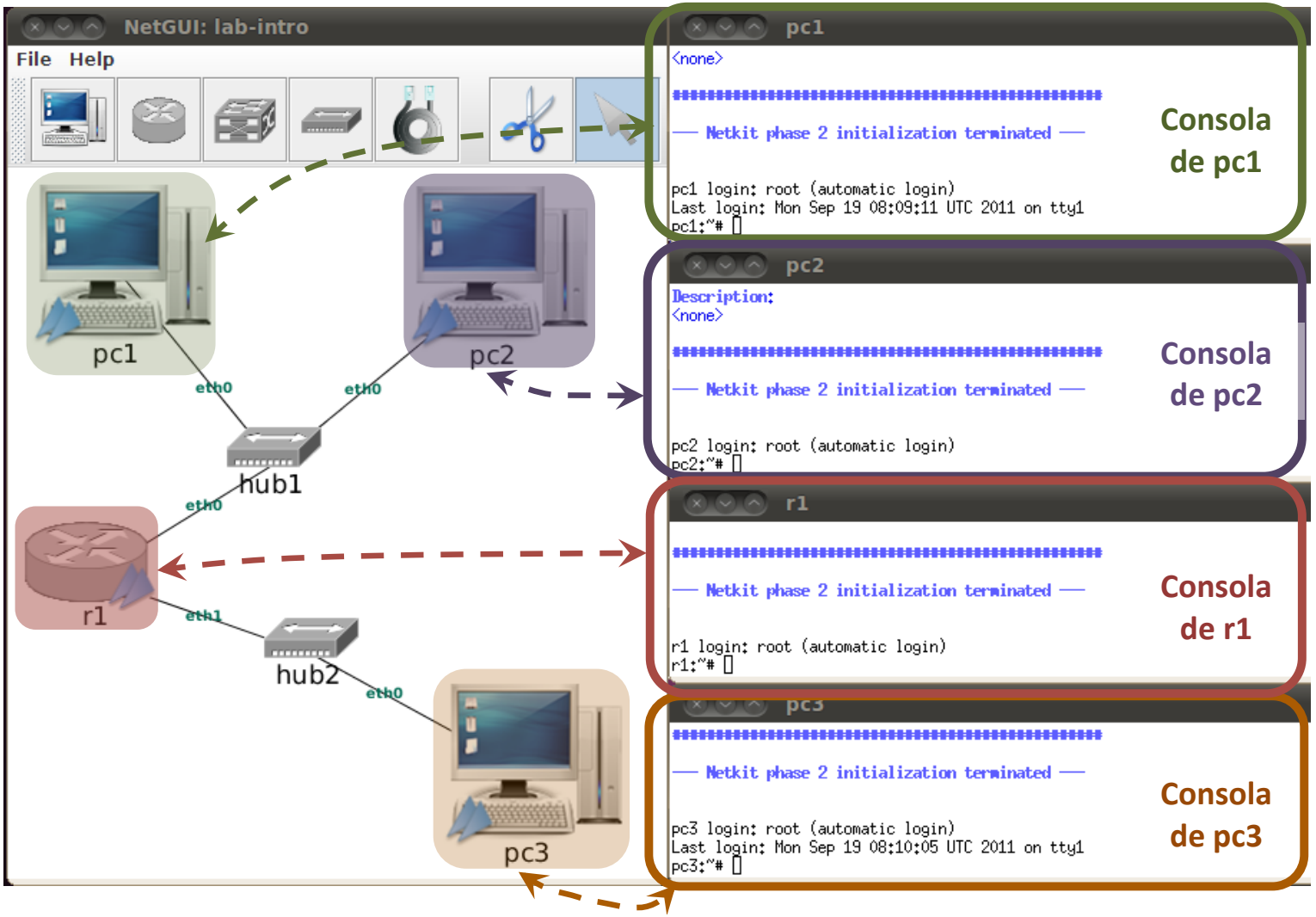
- **Mover toda la figura**: pulsando y arrastrando con el botón izquierdo del ratón sobre el fondo de la ventana (en un lugar en el que no haya ningún elemento).
- **Zoom**: pulsando y arrastrando con el botón derecho del ratón sobre el fondo de la ventana:
  - arrastrando hacia la derecha: aumentar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

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

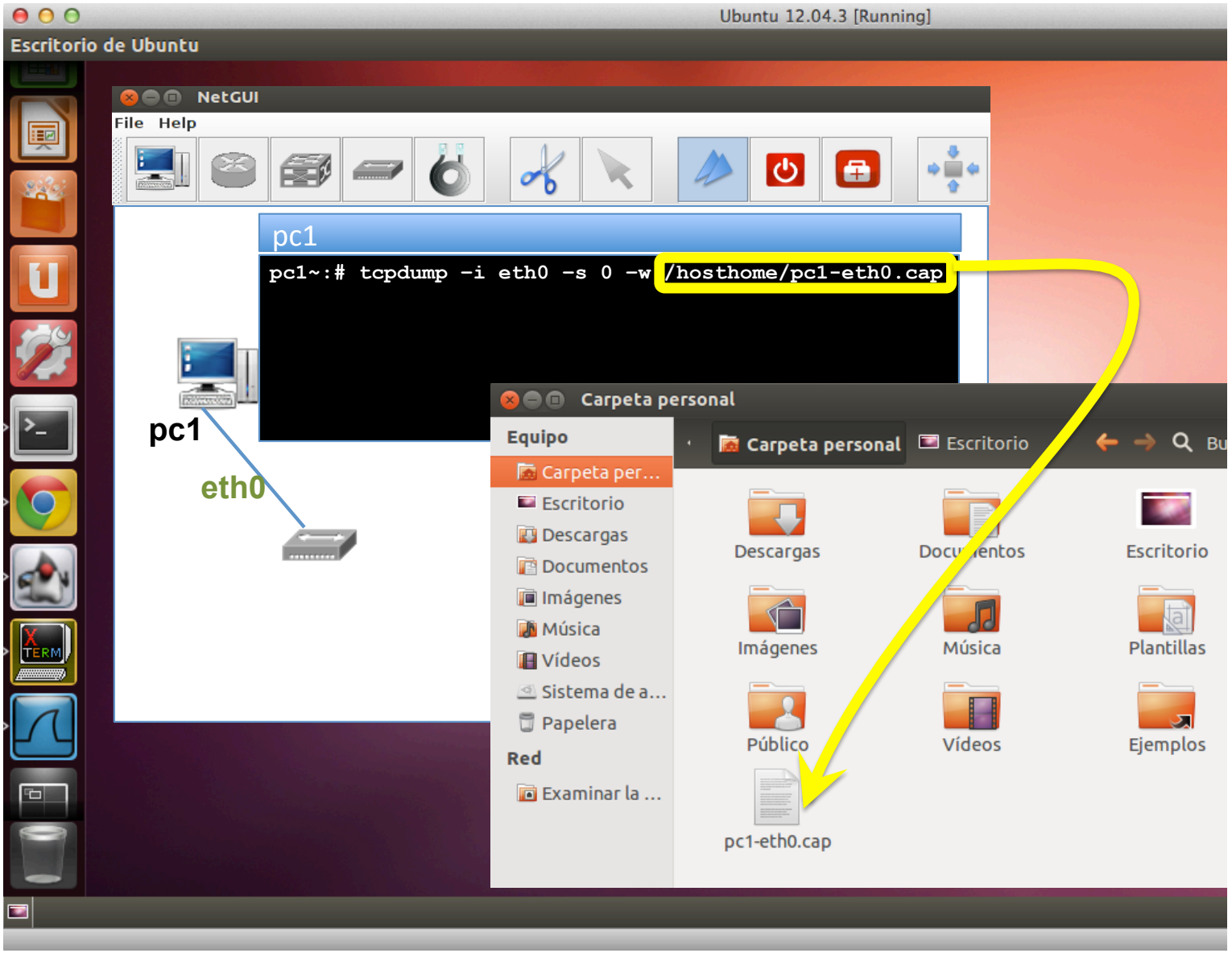
  `-s <tamaño>` Número de bytes que se capturan de cada paquete (por defecto 68 bytes, `-s 0` para capturar paquetes enteros)
- Para interrumpir `tcpdump` es necesario pulsar `Ctrl+C`.

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 /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 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:

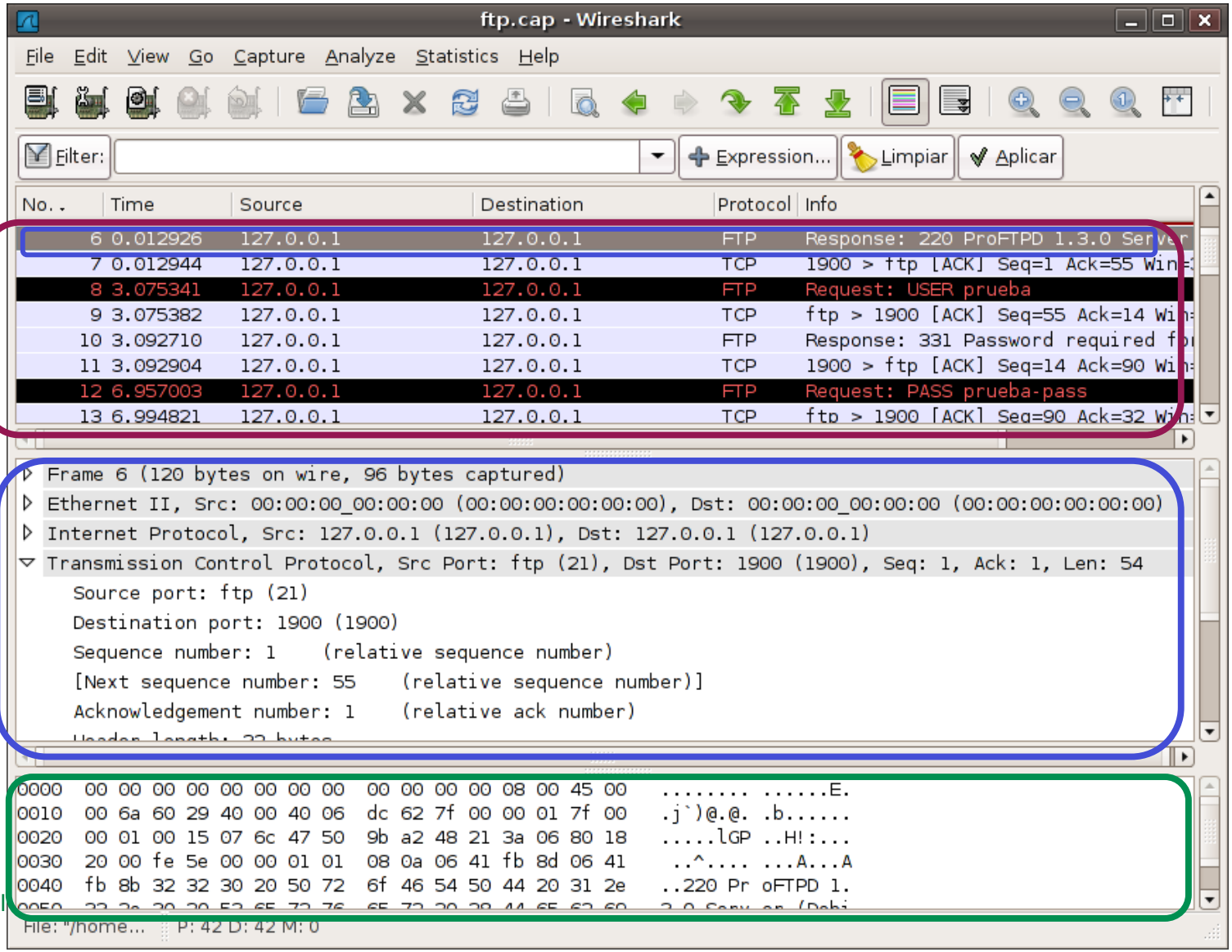
```
usuario@zeta25:~$ wireshark pc1-eth0.cap
```

wireshark

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)



Tema 2: Hubs, Switches

Arquitectura de Redes de Ordenadores

1º Ingeniería Telemática, 1º Ingeniería en Tecnologías de la Telecomunicación, 1º Ingeniería en Sistemas de Telecomunicación

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

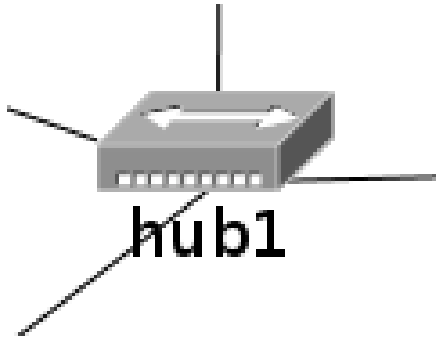
Diciembre 2023

Contenidos

Contenidos

Hubs

- El dispositivo hub en NetGUI **está siempre arrancado**, por tanto no hay que iniciarlo ni detenerlo.

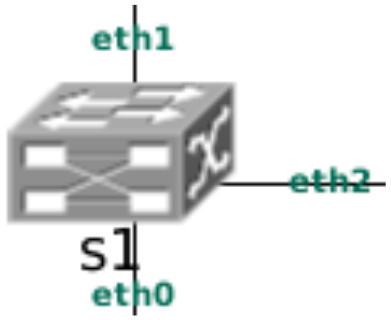


- No tiene consola para poder operar con él.
- El hub copia todos los bits recibidos por un puerto en el resto de los puertos. Por tanto, se capturarán los mismos en cualquiera de las interfaces de red que están conectadas al mismo hub.

Contenidos

Bridges/Switches en NetGUI

- La interfaz de NetGUI permite dibujar *bridges/switches* los cuáles están representados a través del siguiente icono:



- Estos *bridges/switches* se configuran a través del comando `brctl` que pertenece al paquete `bridge-utils` en Linux.
- Normalmente un switch tendrá más de una interfaz, que se nombran según el convenio: eth0, eth1, eth2, etc

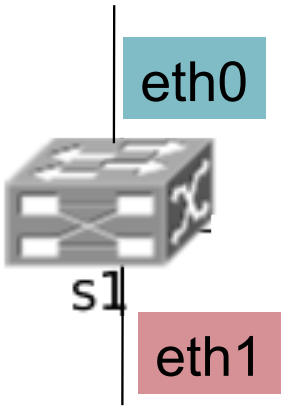
Aprendizaje y reenvío del switch

- Las entradas en **la tabla de direcciones aprendidas** caducan si no se utilizan durante un tiempo. El tiempo que una dirección Ethernet lleva almacenado en la tabla se guarda en un contador *ageing timer* (como máximo 5 minutos, valor por defecto).
- Funciones principales del switch:
  - **Aprendizaje**: el switch aprende las direcciones Ethernet origen en las tramas que recibe asociándolas a una de sus interfaces y guardando esta información en su tabla de direcciones aprendidas. Si esa dirección ya la tiene aprendida, reinicia el contador *ageing timer*.
  - **Reenvío**: el switch reenvía en función de la tabla de direcciones aprendidas. Si la dirección Ethernet destino de la trama es Broadcast o no la tiene en su tabla, reenvía la trama por todas las interfaces salvo por donde le ha llegado.

Consultar información sobre el *bridge*

Para consultar la información del switch s1:

```
s1:~# brctl show
bridge name          bridge id          STP enabled     interfaces
    s1           8000.3e323176a0de          no              eth0
                                                            eth1
```

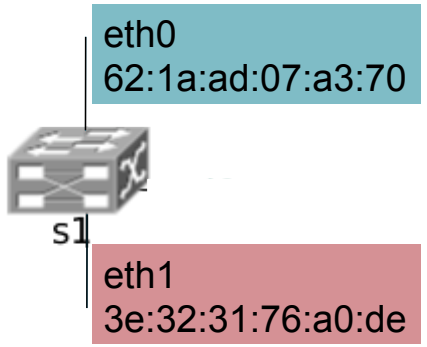


- La información del switch muestra que tiene 2 interfaces: eth0 y eth1.
- El switch no está utilizando STP (Spanning Tree Protocol), un protocolo para resolver problemas de explosión de tráfico al conectar switches que formen un bucle.

Tabla de direcciones Ethernet aprendidas (I)

- Para mostrar la tabla de direcciones aprendidas en s1:

```
s1:~# brctl showmacs s1
port no      mac addr            is local?   ageing timer
  2      3e:32:31:76:a0:de        yes            0.00
  1      62:1a:ad:07:a3:70        yes            0.00
```



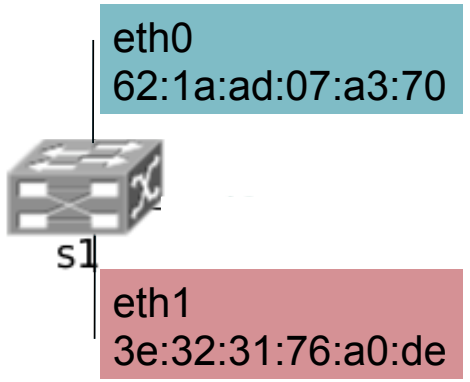
- La tabla anterior muestra sólo las interfaces locales del switch. Es decir, el switch está recién arrancado, todavía no ha aprendido ninguna dirección Ethernet.
- El `port no` enumera las interfaces empezando por 1:
  - port no=1 es la interfaz eth0
  - port no=2 es la interfaz eth1
  - ...
- La columna `ageing timer` indica el tiempo (en segundos) que ha pasado desde que se aprendió o refrescó cada entrada. Las interfaces locales del switch siempre muestran el valor 0 y no caducan nunca.

Tabla de direcciones Ethernet aprendidas (II)

- Para mostrar la tabla de direcciones aprendidas en s1:

```
s1:~# brctl showmacs s1
```

port no	mac addr	is local?	ageing timer
2	0a:29:6e:9a:3e:d4	no	11.77
2	3e:32:31:76:a0:de	yes	0.00
1	62:1a:ad:07:a3:70	yes	0.00
1	aa:da:5c:68:ed:27	no	11.77



- En este caso, la tabla anterior muestra además de las direcciones locales del switch, otras direcciones aprendidas:
  - El switch ha aprendido la dirección aa:da:5c:68:ed:27 a través de su interfaz eth0, en el puerto 1. Hace 11.77 segundos que ha aprendido dicha dirección Ethernet.
  - El switch ha aprendido la dirección 0a:29:6e:9a:3e:d4 a través de su interfaz eth1, en el puerto 2. Hace 11.77 segundos que ha aprendido dicha dirección Ethernet.
- Cada vez que un switch reenvía una trama Ethernet, aprende la dirección Ethernet origen de la trama con valor ageing timer=0. Si ya la tenía aprendida, actualiza su valor ageing timer a 0. Según avanza el tiempo, el valor ageing timer va aumentando.
- Una entrada se elimina de la tabla de direcciones aprendidas cuando su valor aging timer llegar al valor máximo permitido (por defecto, 300 seg).

Borrar la tabla de direcciones Ethernet aprendidas por el *bridge*

- Para borrar la tabla de direcciones aprendidas en s1:

```
s1:~# ifconfig s1 down
```

- Al habilitar de nuevo el *bridge*, éste no tendrá ninguna dirección Ethernet aprendida, salvo las de sus interfaces locales:

```
s1:~# ifconfig s1 up
s1:~# brctl showmacs s1
port no     mac addr       is local?  ageing timer
   2    3e:32:31:76:a0:de   yes         0.00
   1    62:1a:ad:07:a3:70   yes         0.00
```

Tema 3: Configuración de direcciones IP en Linux

Arquitectura de Redes de Ordenadores

1º Ingeniería Telemática, 1º Ingeniería en Tecnologías de la Telecomunicación, 1º Ingeniería en Sistemas de Telecomunicación

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

Diciembre 2023

Contenidos

Contenidos

Configuración de direcciones IP

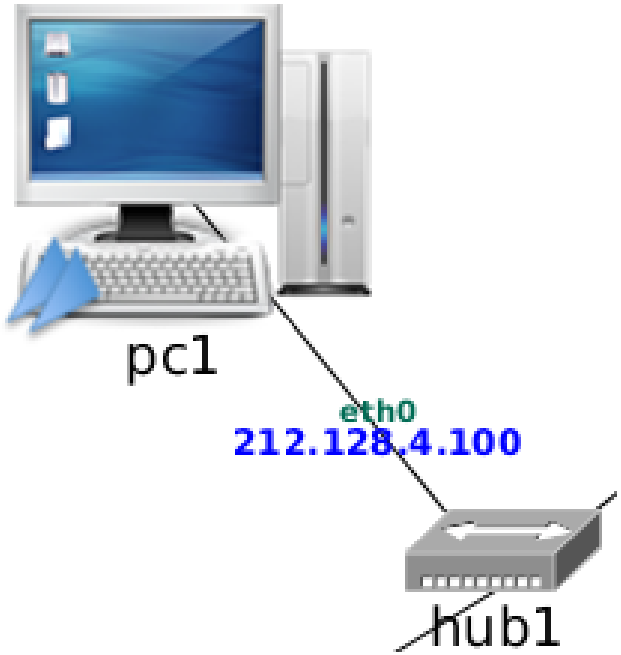
- Configuración de red: Añadir/eliminar/modificar direcciones IP.
- Órdenes que se utilizan:
  - `ifconfig`
  - `ip`

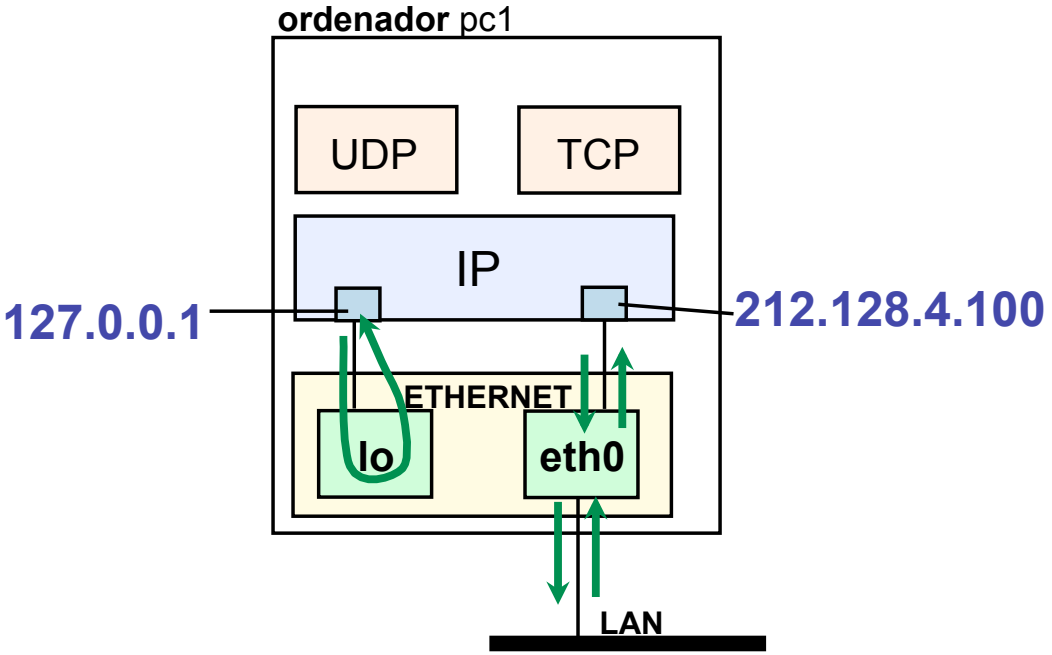
Interfaces de red de una máquina Linux

- 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)

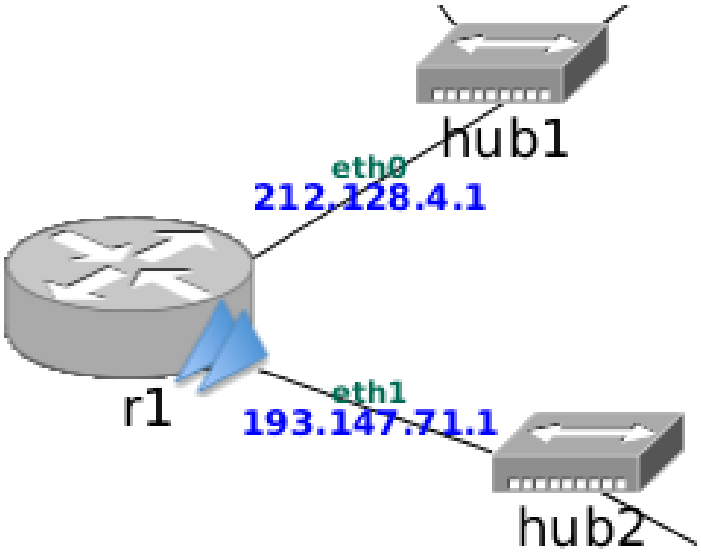
- A cada interfaz de red se le asigna una dirección IP
- A la interfaz de *loopback* se le suele asignar la dirección IP 127.0.0.1 o la 127.0.1.1
- Ejemplo de un PC de NetGUI:

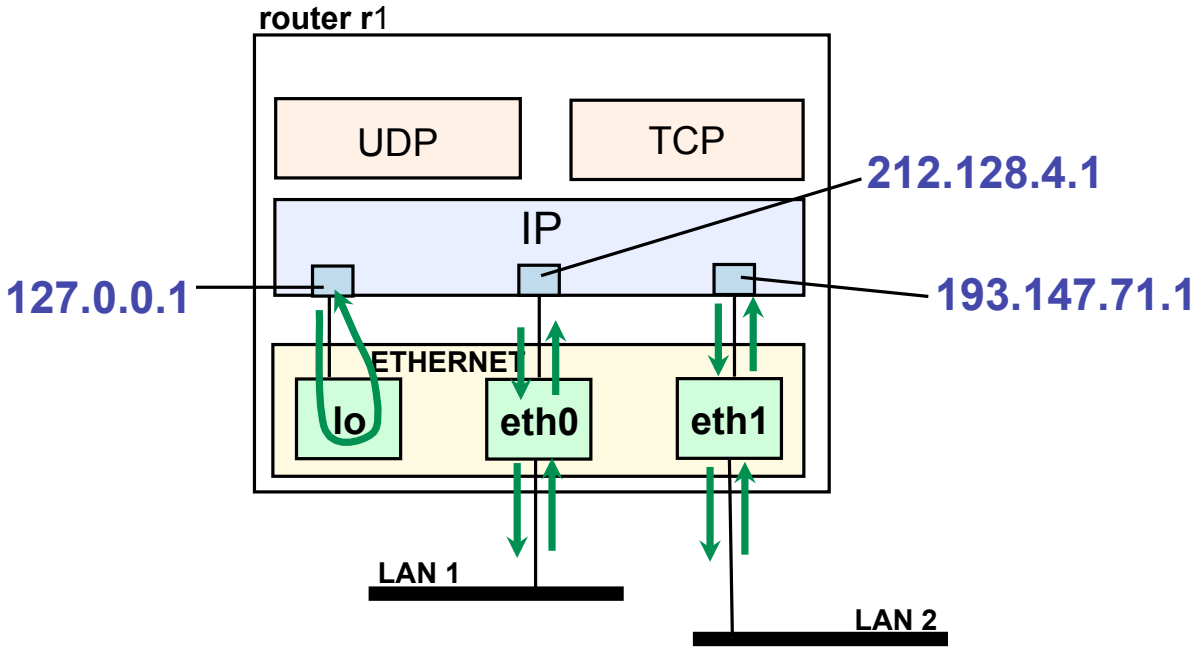




Interfaces de red y direcciones IP (II)

- 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.**

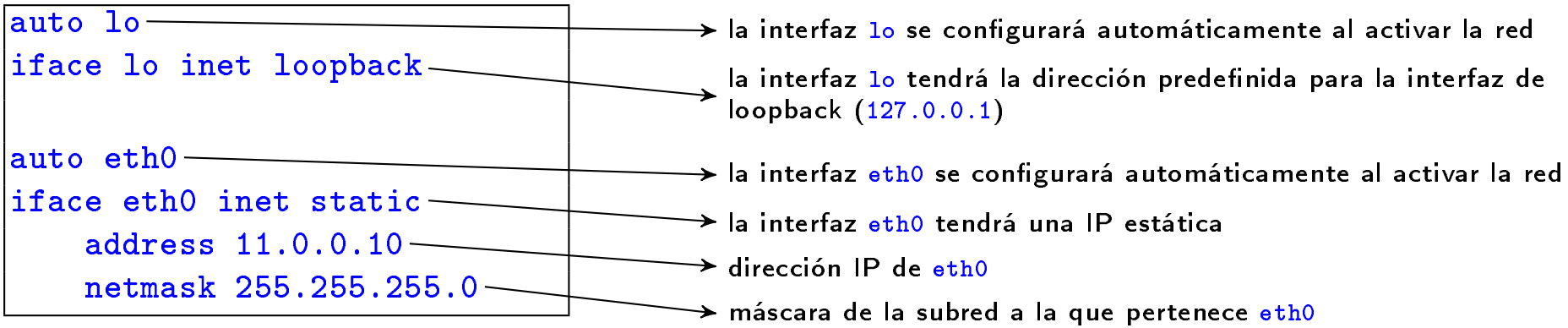
Contenidos

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.

Tema 4: Tabla de encaminamiento

Arquitectura de Redes de Ordenadores

1º Ingeniería Telemática, 1º Ingeniería en Tecnologías de la Telecomunicación, 1º Ingeniería en Sistemas de Telecomunicación

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

Diciembre 2023

Contenidos

Contenidos

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)

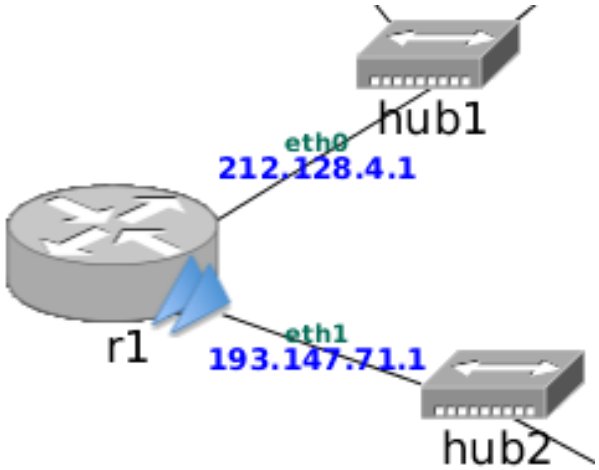
- Si una máquina no tiene asignada ninguna dirección IP, la tabla de encaminamiento estará vacía.
- Al asignar una dirección IP a una máquina, **automáticamente** se añade una entrada en la tabla de encaminamiento para que dicha máquina se pueda comunicar con las máquinas que están directamente conectadas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

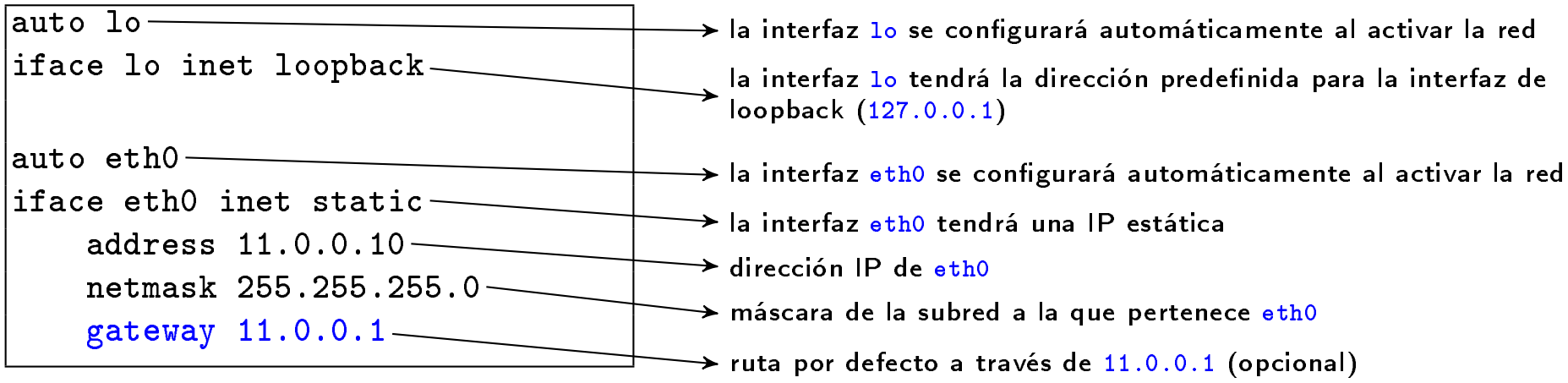
Contenidos

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 prececidas por down: cuando se apague esa interfaz se ejecutará la orden.

Contenidos

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`
  - `traceroute`

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 `Ctrt+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 inical 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
```

Tema 5: Tráfico TCP/UDP

Arquitectura de Redes de Ordenadores

1º Ingeniería Telemática, 1º Ingeniería en Tecnologías de la Telecomunicación, 1º Ingeniería en Sistemas de Telecomunicación

Eva M. Castro Barbero (eva.castro@urjc.es)
José Centeno González (jose.centeno@urjc.es)
Pedro de las Heras Quirós (pedro.delasheras@urjc.es)

Diciembre 2023

Contenidos

netstat

- La herramienta `netstat` permite obtener información sobre varios aspectos del estado de la red en un sistema Unix/Linux.
- Entre otros usos, permite ver el listado de comunicaciones activas en una máquina: detalles de las conexiones TCP y comunicaciones UDP que hay establecidas en ese momento.
- Sintaxis:

```
netstat -tna
```

```
netstat -una
```

  - la opción `-t` muestra información de las conexiones TCP
  - la opción `-u` muestra información de las comunicaciones UDP
  - la opción `-n` muestra direcciones IP (si se omite, se trata de mostrar nombres de máquinas por DNS en su lugar)
  - la opción `-a` muestra información de todas las comunicaciones, incluyendo aquellas en las que está máquina ha lanzado un servidor que está esperando recibir mensajes de clientes

netstat

- `netstat` mostrará la siguiente información para las comunicaciones activas:

```
pc1:~# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
```

  - La columna `Proto` indica el protocolo utilizado (UDP o TCP)
  - La columna `Local Address` muestra la dirección IP local de la máquina donde se esperan recibir datos y el número de puerto.
  - En la columna `Foreign Address` muestra la dirección IP y puerto de las máquinas remota con la que se ha establecido una comunicación.
  - Las columnas `Recv-Q` (`receiving queue`) y `Send-Q` (`sending queue`) muestran la cantidad de bytes que hay almacenados en los buffers locales reservados para la recepción de datos y emisión de datos de este servidor.
  - La columna `State` indicará el estado de la comunicación.

netstat: comunicaciones UDP (I)

- Para visualizar las comunicaciones UDP activas:

```
pc1:~# netstat -una
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 0.0.0.0:7777            0.0.0.0:*
```

  - El resultado de ejecutar este comando muestra un servidor UDP esperando recibir conexiones de clientes en el puerto 7777.
  - La columna `Local Address` muestra la dirección 0.0.0.0 que indica que se esperan recibir comunicaciones UDP en cualquiera de las direcciones IP configuradas actualmente en la máquina local.
  - En la columna `Foreign Address` se mostrarán las direcciones IP y puertos de las máquinas clientes remotos que se conecten con este servidor. Actualmente no hay ninguna.
  - Las columnas `Recv-Q` y `Send-Q` muestran que no hay datos almacenados en los buffers.

netstat: comunicaciones UDP (II)

- Para visualizar las comunicaciones UDP activas:

```
pc1:~# netstat -una
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 11.0.0.1:7777           11.0.0.2:32768          ESTABLISHED
```

  - El resultado de ejecutar este comando muestra una comunicación UDP entre la dirección IP local 11.0.0.1 y puerto 7777 y la dirección IP remota 11.0.0.2 y puerto 32768.
  - Las columnas `Recv-Q` y `Send-Q` muestran que no hay datos almacenados en los buffers.

netstat: comunicaciones TCP (I)

- Para visualizar las comunicaciones TCP activas:

```
pc1:~# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:7777            0.0.0.0:*               LISTEN
```

  - El resultado de ejecutar este comando muestra un servidor TCP esperando recibir conexiones de clientes en el puerto 7777.
  - La columna `Local Address` muestra la dirección 0.0.0.0 que indica que se esperan recibir comunicaciones UDP en cualquiera de las direcciones IP configuradas actualmente en la máquina local.
  - En la columna `Foreign Address` se mostrarán las direcciones IP y puertos de las máquinas clientes remotos que se conecten con este servidor. Actualmente no hay ninguna.
  - Las columnas `Recv-Q` y `Send-Q` muestran que no hay datos almacenados en los buffers.

netstat: comunicaciones TCP (II)

- Para visualizar las comunicaciones TCP activas:

```
pc1:~# netstat -tna
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 11.0.0.1:7777           11.0.0.2:33715          ESTABLISHED
```

  - El resultado de ejecutar este comando muestra una comunicación TCP entre la dirección IP local 11.0.0.1 y puerto 7777 y la dirección IP remota 11.0.0.2 y puerto 33715.
  - Las columnas `Recv-Q` y `Send-Q` muestran que no hay datos almacenados en los buffers.

Contenidos

Herramienta nc

- Usaremos `nc` para generar tráfico TCP y UDP según el modelo de comunicaciones cliente/servidor.
- Se arrancarán 2 aplicaciones: una funcionando con el rol cliente y la otra con el rol servidor.
- **Siempre es necesario lanzar primero la aplicación que funciona como servidor**. El servidor quedará a la espera de recibir tráfico procedente de la aplicación cliente, que se deberá lanzar después.
  - Una aplicación lanzada con `nc` como **cliente** lee de la entrada estándar (por omisión el teclado) los caracteres introducidos y al pulsar la tecla `INTRO` la línea de texto es enviada usando TCP o UDP a la aplicación servidor.
  - Al recibir la línea de texto, la aplicación **servidor** lanzada con `nc` mostrará en la pantalla los datos recibidos de la aplicación cliente lanzada con `nc`.

Contenidos

Aplicación servidor UDP

- Para arrancar `nc` como **servidor** utilizando el protocolo UDP ejecutaremos la siguiente orden:

```
nc -u -l -p <Pto-Loc>
```

  - `-u`: UDP
  - `-l`: *listen* = modo servidor
  - `-p <Pto-Loc>`: número de puerto local UDP en el que la aplicación servidor esperará recibir los datagramas UDP de una aplicación cliente.

- Por ejemplo, si queremos arrancar una aplicación servidor UDP en el puerto 7777 de la máquina `pc1` utilizaremos la siguiente orden:

```
pc1:~# nc -u -l -p 7777
```

Aplicación cliente UDP

- Para arrancar `nc` como **cliente** utilizando el protocolo UDP ejecutaremos la siguiente orden:

```
nc -u -p <Pto-Loc> <IP-dest> <Pto-dest>
```

  Donde:

  - `-u`: UDP
  - `-p <Pto-Loc>`: número de puerto local UDP en el que la aplicación cliente esperará recibir los datagramas UDP que vengan del servidor.
  - `<IP-dest>`: dirección IP de la máquina donde se está ejecutando la aplicación servidor UDP.
  - `<Pto-dest>` es el número de puerto UDP en el que escucha la aplicación servidor UDP.

- Por ejemplo, si queremos arrancar una aplicación cliente UDP en `pc2` que espere recibir datagramas UDP en el puerto 6666 y que envíe datagramas UDP a la dirección IP 200.0.0.1 y puerto 7777 (donde se encuentra esperando recibir datagramas UDP la aplicación servidor) utilizaremos la siguiente orden:

```
pc2:~# nc -u -p 6666 200.0.0.1 7777
```

Envío de datos UDP

- Una vez lanzadas las aplicaciones servidor UDP y cliente UDP, el cliente puede enviarle líneas de texto al servidor.
- Después de que el cliente haya enviado al menos una línea de texto al servidor, todo lo que escribamos a través de la entrada estándar de un extremo será enviado al otro extremo como datagramas UDP: si escribimos en el terminal de la aplicación cliente, esto será enviado a la aplicación servidor, y viceversa.
- Pasado un cierto tiempo desde el último mensaje del cliente, el servidor "olvida" al cliente (recuerda que en UDP no hay conexiones) y es necesario volver a enviar un mensaje desde el cliente para que el servidor pueda volver a enviarle mensajes.
- Para interrumpir la ejecución de estas aplicaciones se debe utilizar `Ctrl+C`.

Contenidos

Aplicación servidor TCP

- Para arrancar `nc` como **servidor** utilizando el protocolo TCP ejecutaremos la siguiente orden:

```
nc -l -p <Pto-Loc>
```

  Donde:

  - `-l`: *listen* = modo servidor
  - `-p <Pto-Loc>`: es el número de puerto local TCP en el que la aplicación servidor esperará recibir mensajes TCP de una aplicación cliente.

- Por ejemplo, si queremos arrancar una aplicación servidor TCP en el puerto 7777 de la máquina `pc1` utilizaremos la siguiente orden:

```
pc1:~# nc -l -p 7777
```

Aplicación cliente TCP

- Para arrancar `nc` como **cliente** utilizando el protocolo TCP ejecutaremos la siguiente orden:

```
nc -p <Pto-Loc> <IP-dest> <Pto-dest>
```

  Donde:

  - `-p <Pto-Loc>`: número de puerto local TCP en el que la aplicación cliente esperará recibir los mensajes de la aplicación servidor TCP.
  - `<IP-dest>`: dirección IP de la máquina donde se está ejecutando la aplicación servidor TCP.
  - `<Pto-dest>`: número de puerto TCP en el que escucha la aplicación servidor TCP.

- Por ejemplo, si queremos arrancar una aplicación cliente TCP en `pc2` que utilice el puerto origen 6666 para establecer una conexión TCP con un servidor TCP que escuche en el puerto destino 7777 de la máquina 200.0.0.1, utilizaremos la siguiente orden:

```
pc2:~# nc -p 6666 200.0.0.1 7777
```

Envío de datos TCP

- Una vez iniciada la aplicación servidor TCP, ésta se queda esperando recibir mensajes de una aplicación cliente TCP.
- Una vez iniciada la aplicación cliente TCP, ésta intercambiará unos mensajes de control (apertura de conexión) con la aplicación servidor, por lo que es imprescindible que dicha aplicación servidor haya sido lanzada antes.
- Si la comunicación entre ambas aplicaciones es posible, a partir de este momento todo lo que escribamos a través de la entrada estándar de una aplicación será enviada a la otra: si escribimos en el terminal de la aplicación cliente, esto será enviado a la aplicación servidor, y viceversa.
- Para interrumpir la ejecución de estas aplicaciones se debe utilizar `Ctrl+C`.

Contenidos

Gráfica de *tcptrace* dentro de Wireshark

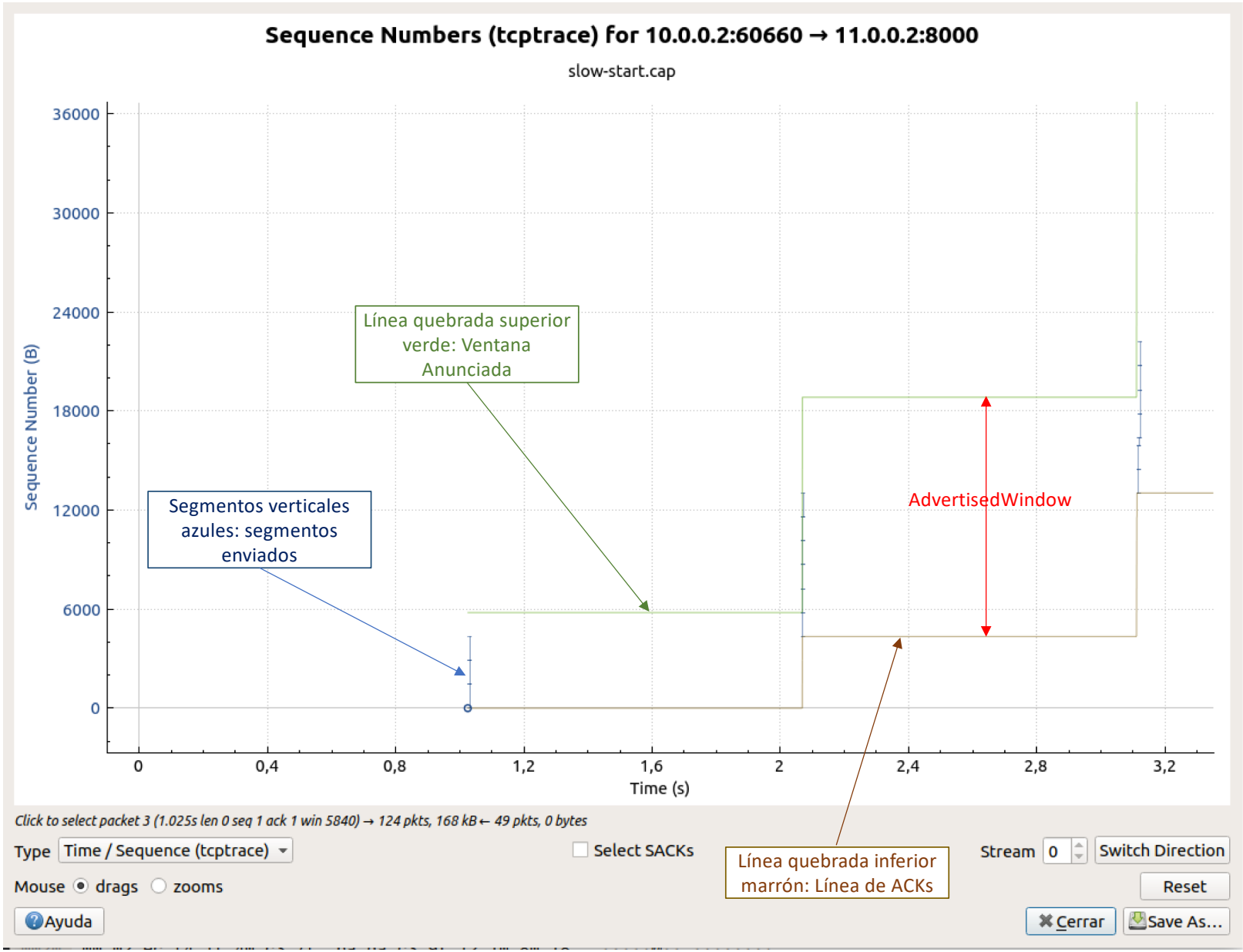
- En Wireshark, además de mirar el contenido de los paquetes de una conexión TCP, puede verse en una gráfica la **evolución del envío de datos y recepción de acks respecto al tiempo**.
- Wireshark permite mostrar varios tipos de gráficas de una conexión TCP: Nosotros utilizaremos la gráfica de *tcptrace*.
- Como una conexión TCP permite el envío de datos en ambos sentidos, se pueden visualizar 2 gráficas de *tcptrace* diferentes: las correspondientes a cada sentido de la comunicación.
- Para ver en Wireshark la gráfica de *tcptrace* de uno de los sentidos de una conexión TCP es necesario:
  - Cargar el fichero de una captura que contenga los paquetes de una conexión TCP.
  - Seleccionar un segmento de la conexión del sentido de la comunicación que queremos analizar (si el segmento seleccionado va del proceso A al proceso B, la gráfica que se mostrará será la correspondiente al envío de datos de A a B).
  - Seleccionar en el menú de Wireshark:
    Statistics→TCP Stream Graph→Time-Sequence Graph (tcptrace)

Versiones de Wireshark

- En las versiones recientes de Ubuntu hay dos versiones diferentes de Wireshark, en las que varía un poco la apariencia de las gráficas *tcptrace*:
  - `wireshark` (a veces queda instalado con nombre `wireshark-qt`)
  - `wireshark-gtk`
- Por defecto suele instalarse al versión "nueva" (`wireshark`). Si se quiere tener instalada también la versión "antigua":

  `sudo apt install wireshark-gtk`.

wireshark: Apariencia



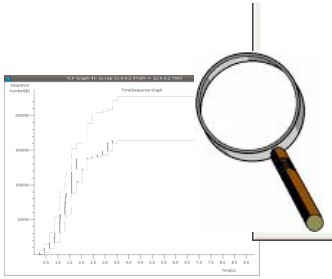
wireshark: Controles

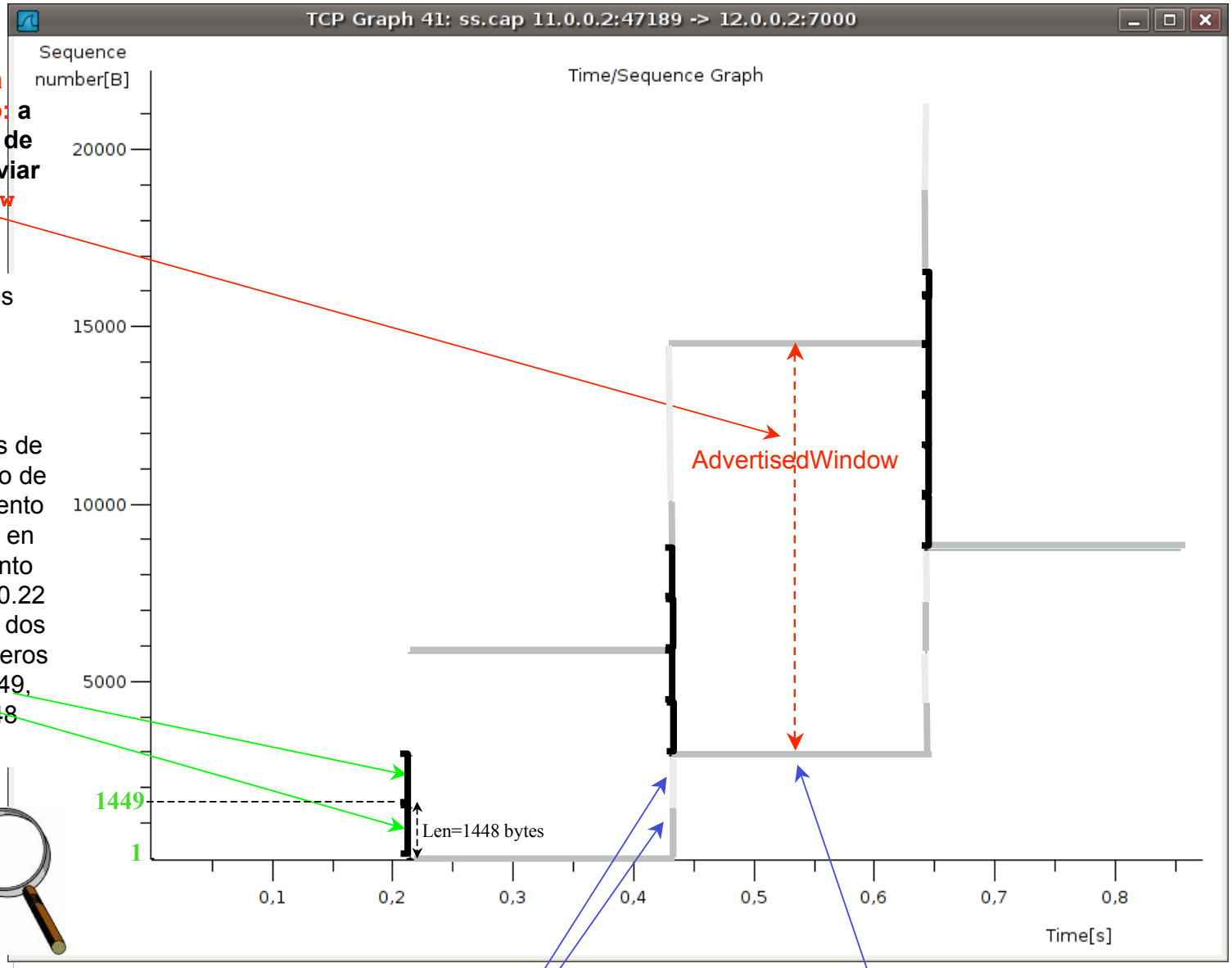
- **Rueda del ratón**: zoom in/out
- **Arrastrar con el botón izquierdo**: desplazar el gráfico (útil si se ha hecho "zoom in")
- **ESPACIO**: activa/desactiva una cruz para ayudar a ver sobre los ejes la posición del ratón.
- **Click izquierdo sobre un segmento**: seleccionar el paquete concreto en la lista de paquetes de Wireshark.
- **Botón 'Switch Direction'**: pasa a mostrar el otro sentido de la conexión.

wireshark-gtk: Apariencia

Ventana anunciada por el otro extremo: a partir del último nº de ACK se pueden enviar `AdvertisedWindow` bytes

Segmentos verticales negros: **segmentos TCP de datos enviados**. Cada segmento ocupa un conjunto de números de secuencia: el número de secuencia del segmento TCP más la longitud en bytes de ese segmento TCP. En el instante 0.22 segundos se envían dos segmentos con números de secuencia 1 y 1449, cuya longitud es 1448 bytes.





Segmentos verticales grises: **segmentos TCP de ACK** recibidos

Línea inferior gris claro: **último nº de ACK recibido**

wireshark-gtk: Controles

- **Click central**: zoom in
- **MAYS + Click central**: zoom out
- **Arrastrar con el botón derecho**: desplazar el gráfico (útil si se ha hecho "zoom in")
- **ESPACIO**: activa/desactiva una cruz para ayudar a ver sobre los ejes la posición del ratón.
- **Click izquierdo sobre un segmento**: seleccionar el paquete concreto en la lista de paquetes de Wireshark.
- **CTRL + arrastrar con el botón derecho**: lupa
- **s**: Alterna entre números de secuencia relativos y absolutos, sólo si está desactivada la opción Edit→Preferences→Protocols→TCP→Relative sequence numbers and window scaling.

Seguimiento de conexiones con tcpdump en el terminal