Prácticas con NetGUI

Práctica 3

Arquitectura de Redes de Ordenadores

Arquitectura de Internet

GSyC

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

Abril de 2018

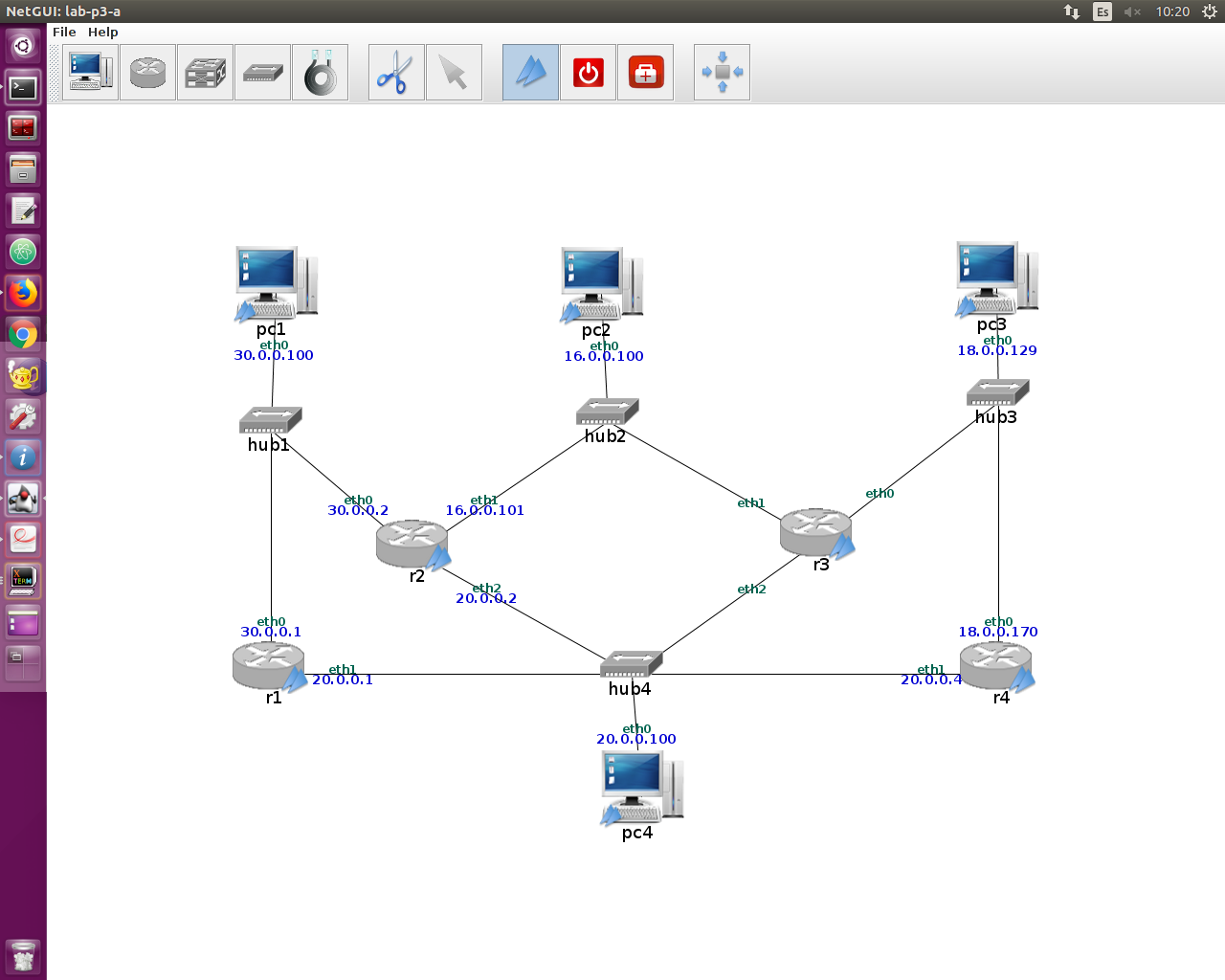
**1.Escenario A: Configuración enrutamiento IP.**

1. Observa las direcciones IP que aparecen configuradas en el escenario de red. Comprobarás que todas las máquinas excepto r3 tienen ya configurada su dirección IP. Comprueba qué rutas tienen configuradas las máquinas de la figura.

Para comprobar las rutas de las máquinas utilizamos el comando *route* en cada una de ellas.

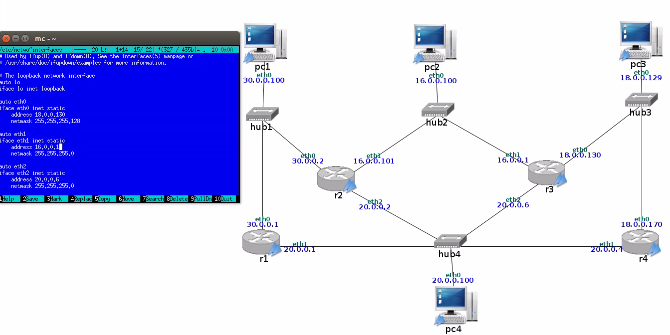
2. Comprueba que en pc1 no funciona un ping a la dirección 30.0.0.2 ¿Por qué? Realiza los cambios necesarios en la configuración de pc1 para que dicho ping funcione. Realiza los cambios de forma que pc1 mantenga su nueva configuración aunque se apague y vuelva a encenderse.

El ping no funciona porque no están en la misma subred entonces cambiamos la dirección de pc1 que es la forma más rápida de solucionar el problema y así ya funcionará.



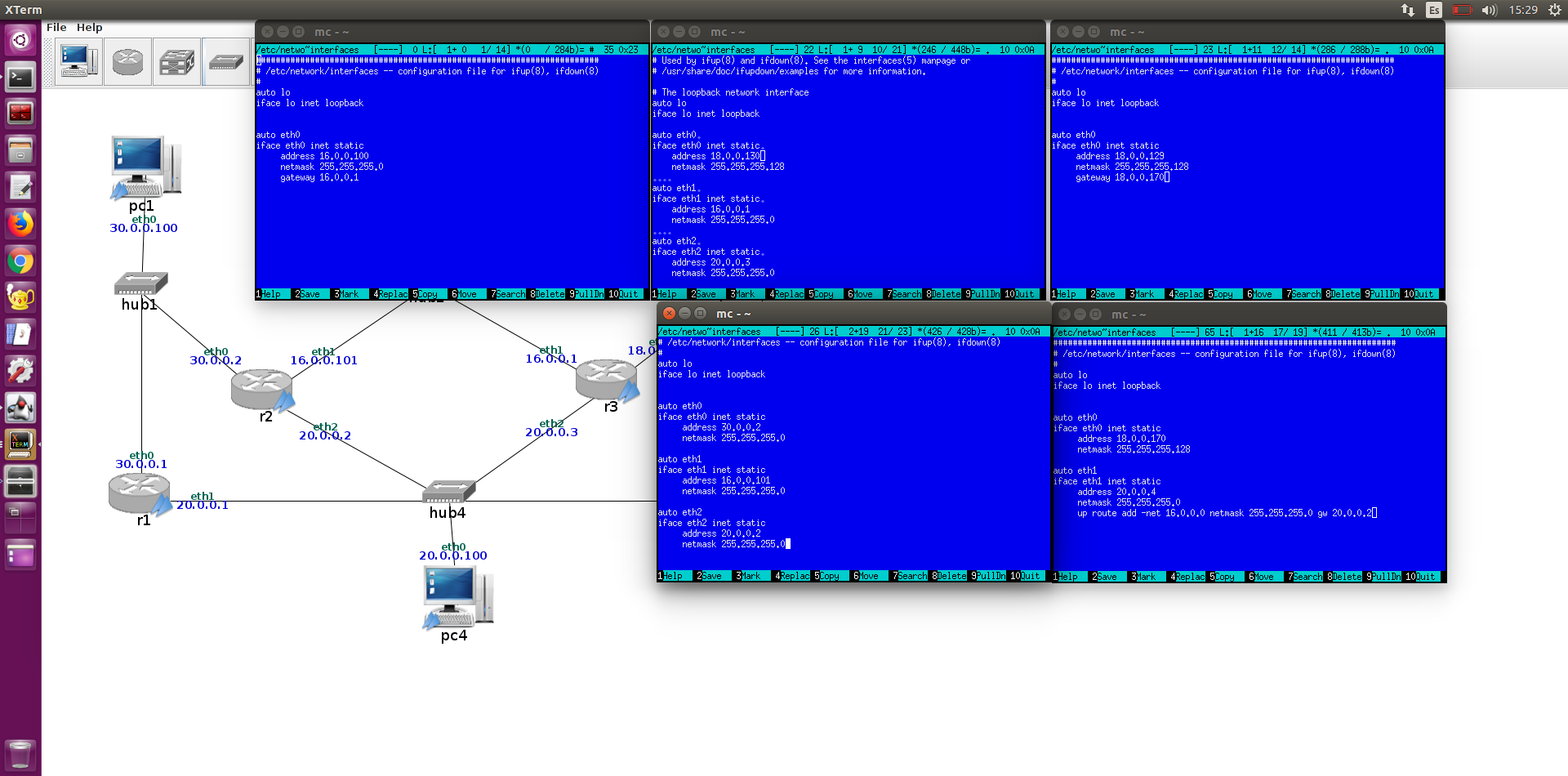
3. La máquina r3 no tiene configuradas sus interfaces de red. Configurar direcciones IP adecuadas para sus interfaces eth0 , eth1 y eth2 , de forma que dicha configuración se mantenga después de apagar y volver a encender r3 .

Adjuntamos la captura del escenario ya configurado y del interfaz de r3 con los cambios realizados.(KAROL DICE QUE POR TEORÍA NO SE PUEDE PONER A LA RUTA QUE SALE POR EHT0 DEL R3 18.0.0.128 PORQUE EL 128 ES EL INICIO DE LA MÁSCARA)



4. Realiza los cambios necesarios para que pc2 y pc3 puedan intercambiar datagramas IP y lo hagan por las siguientes rutas:

Desde pc2 a pc3 : pc2 => r3 => pc3

Desde pc3 a pc2 : pc3 => r4 => r2 => pc2

Intenta realizar los mínimos cambios posibles.Comprueba que las rutas seguidas son las pedidas ejecutando traceroute desde pc2 a pc3 y viceversa.

5. Realiza los cambios necesarios para que pc4 pueda intercambiar datagramas IP con pc1 , pc2 y pc3 , independientemente de la ruta por la que lo haga.

Intenta realizar los mínimos cambios posibles.

Pc4 puede intercambiar con pc1 con lo que ya tenemos configurado anteriormente.

Para que pc4 y pc2 intercambien datagramas IP añadimos la siguiente ruta en pc4 *up route add -net 16.0.0.0 netmask 255.255.255.0 gw 20.0.0.2.*

Para que pc4 y pc3 intercambien datagramas IP añadimos la siguiente ruta en pc4 sin borrar las anterior como bien sabemos *up route add -net 18.0.0.128 netmask 255.255.255.128 gw 20.0.0.4.*

6. Localiza qué máquinas de entre pc1 , pc2 , pc3 y pc4 no pueden intercambiar datagramas entre sí. Realiza los cambios necesarios para que puedan.

Intenta realizar los mínimos cambios posibles.

Pc1 no puede intercambiar datagramas con pc2 y pc3 pero si con pc4.

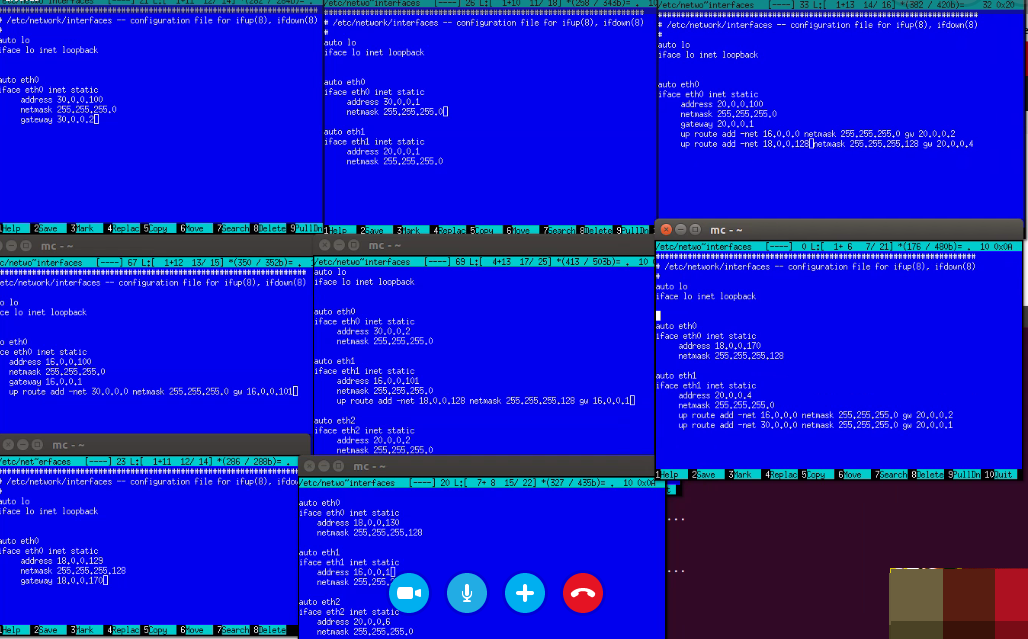
Pc2 no puede intercambiar datagramas con pc1 pero si con pc3 y pc4.

Para arreglarlo en pc2 hemos añadido *up route add -net 30.0.0.0 netmask 255.255.255.0 gw 16.0.0.101*. Con esto ya pueden intercambiar datagramas pc2 a pc1 y viceversa.

Pc3 no puede intercambiar datagramas con pc1 pero si con pc2 y pc4.

Para que pc3 pueda intercambiar datagramas con pc1 y viceversa añadimos en r4 un ruta *up route add -net 30.0.0.0 netmask 255.255.255.0 gw 20.0.0.1* y tambien en r2 eth1 *up route add -net 18.0.0.128 netmask 255.255.255.128 gw*

*16.0.0.1.*

Pc4 intercambia datagramas con todos los demás pcs.

7. Apaga r1 y r4 . Realiza los cambios necesarios para que pc1 , pc2 , pc3 y pc4 puedan seguir intercambiando datagramas IP entre sí.

Intenta realizar los mínimos cambios posibles.

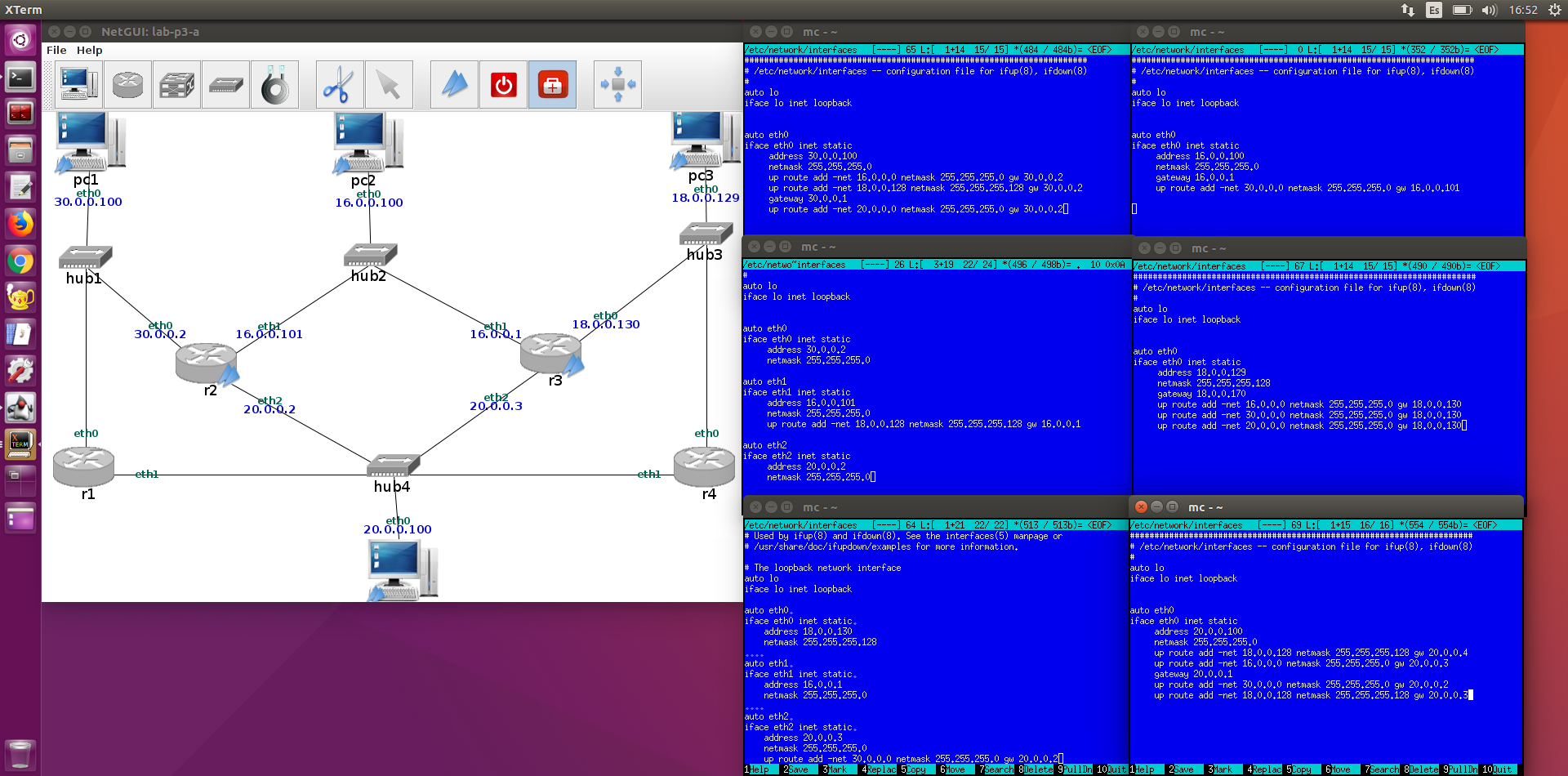
Comprobamos que pc1 intercambia datagramas con pc2 pero no con pc3 ni con pc4.

En pc2 se intercambia datagramas con pc1 pero no con pc3 ni con pc4.

En pc3 no se intercambia datagramas con ninguno de los pcs.

En pc4 tampoco se intercambia ningún datagrama.

Todo esto tras apagar r1 y r4 como nos indica el enunciado.



**2. Escenario B: Comunicación de aplicaciones usando el protocolo UDP**

**2.1. Análisis de captura de tráfico UDP**

En la captura udp.cap se muestra una comunicación UDP. Contesta a las siguientes preguntas:

1. ¿Cuáles son las direcciones IP y puertos involucrados en la comunicación?

Dirección IP origen: 10.0.0.2

Dirección IP destino: 11.0.0.2

Puerto de origen: 32768

Puerto de destino: 33000

2. ¿Qué información puedes extraer de la captura sobre la red en la que se ha realizado la captura?

Podemos ver que hay dos subredes distintas y dos máquinas intercambiando información. Además observamos un TTL = 64 que nos indica que no pasa por ningún router intermedio.

3. ¿Cuál es el número de paquetes UDP y número de bytes de datos intercambiados?

Observamos dos paquetes de UDP, los cuales tienen 6 y 5 bytes. Por lo que intercambian en total 11 bytes de datos.

**2.2. Estudio de UDP mediante aplicaciones cliente y servidor lanzadas con nc**

En esta parte mirar anexo funcionamiento nc para la teoría.

**2.2.1. UDP es un protocolo basado en datagramas: no hay establecimiento de conexión.**

1. Inicia una captura en el router r11

. tcpdump -i eth0 -s 0 -w /hosthome/r1.cap

2. Usando nc lanza una aplicación servidor UDP en la máquina 12.0.0.10 y puerto 11111: nc -u -l -p 11111.

3. Usando nc lanza una aplicación cliente UDP en la máquina 11.0.0.10 para que se comunique con el servidor.

(no envíes datos ni desde el cliente al servidor ni desde el servidor al cliente), desde el puerto local 33333:

nc -u -p 33333 12.0.0.10 11111

4. Interrumpe la captura.

Explica qué paquetes deberían haberse capturado. Observa la captura y comprueba tu suposición.

No se han capturado paquetes, ya que el protocolo UDP es “No orientado a conexión”. Por lo tanto, realizando estas tareas no envía bytes, tendrías que enviarlos para que capturase paquetes.

**2.2.2. Fragmentación IP con envíos UDP.**

1. Inicia una nueva captura en el router r1 para que guarde los paquetes capturados en un chero.

2. Escribe en el terminal donde tienes lanzado el cliente 20 líneas de texto, pulsando una letra cualquiera del teclado (con el tamaño por defecto del terminal de NetGUI, cada línea permite escribir 80 caracteres, así que estarás generando una línea de 80x20=1600 caracteres, cada uno de ellos ocupando 1 byte).

3. A continuación pulsa la tecla INTRO o ENTER (véase la gura 3)

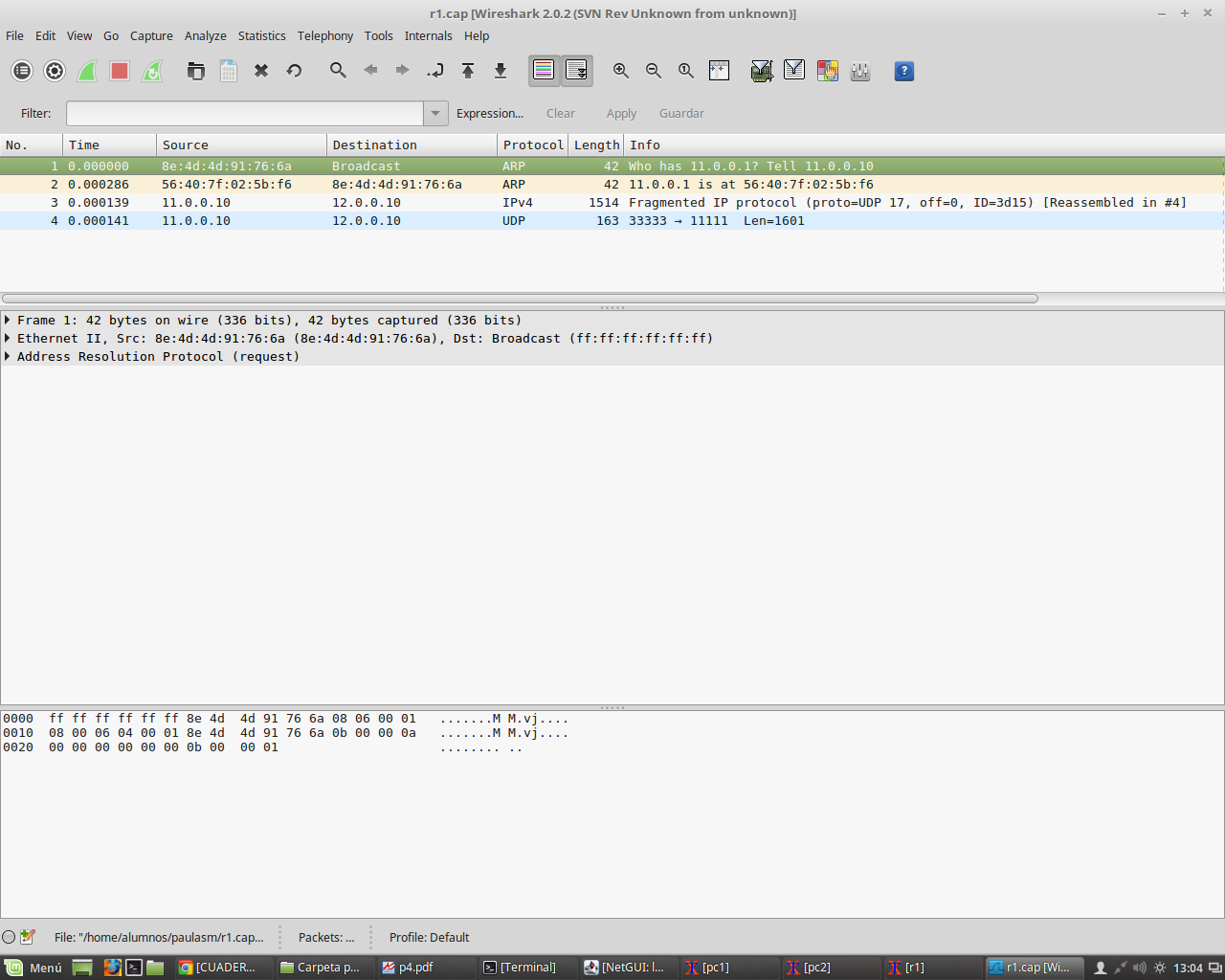
Antes de observar en la captura lo que ha ocurrido responde a estas preguntas:

1. ¿cuántos datagramas UDP crees que se han enviado, y por qué?

Se ha enviado un datagrama UDP, ya que en el protocolo UDP no se pueden fragmentar los datagramas. Su longitud será de 1601 bytes (letras a + tecla intro).

2. ¿cuántos datagramas IP crees que se han enviado, y por qué?

Se habrán enviado un datagramas IP fragmentado ya que, datagrama IP es 1480 (+cabecera 1500) bytes y aquí tenemos 1601. captura r1-eth0-1

3. ¿cuántos bytes de datos irán en cada datagrama UDP? Interrumpe ahora la captura y comprueba si tus suposiciones son correctas. Explica razonadamente el número de datagramas UDP, el número de datagramas IP, y el número de bytes de datos que va en cada uno de los datagramas UDP.

**2.2.3. UDP es un protocolo basado en datagramas: no hay cierre de conexión** 1. Inicia una captura en el router r1. Esta vez no es necesario guardar el contenido en un fichero.

2. Interrumpe la ejecución del cliente pulsando la tecla C mientras mantienes pulsada la tecla Ctrl (a partir de ahora diremos pulsa Ctrl+C).

3. Interrumpe la ejecución del servidor pulsando Ctrl+C. Explica cuántos paquetes deberían haberse capturado y por qué como consecuencia de terminar el cliente con Ctrl+C. Interrumpe la captura y comprueba tu suposición

Cero paquetes puesto que no hemos enviado nada simplemente hemos interrumpido las capturas anteriores.

**3. Escenario C: Comunicación de aplicaciones usando el protocolo TCP.**

**3.1. TCP es un protocolo orientado a conexión.**

**3.1.1. Establecimiento de conexión.**

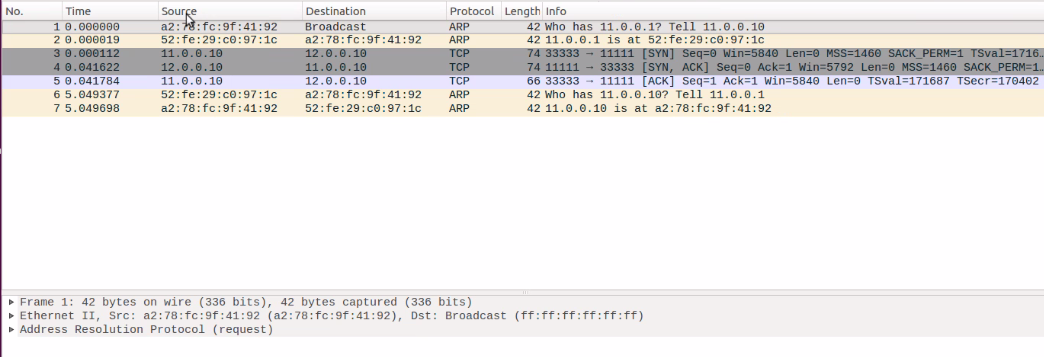
1. Inicia una captura en el router r1 y guarda su contenido en un fichero.

2. Usando nc, lanza una aplicación servidor en la máquina 12.0.0.10 que atienda peticiones de conexión destinadas al puerto TCP 11111: nc -l -p 11111.

3. Lanza una aplicación cliente con nc en la máquina 11.0.0.10 para que se establezca una conexión TCP con la aplicación servidor, usando como puerto local TCP el 33333: nc -p 33333 12.0.0.10 11111.

Explica cuántos paquetes deberían haberse capturado y por qué. Interrumpe la captura y comprueba si tus respuestas se corresponden con lo observado en la captura.

Como observamos en la captura se capturan tres paquetes TCP.

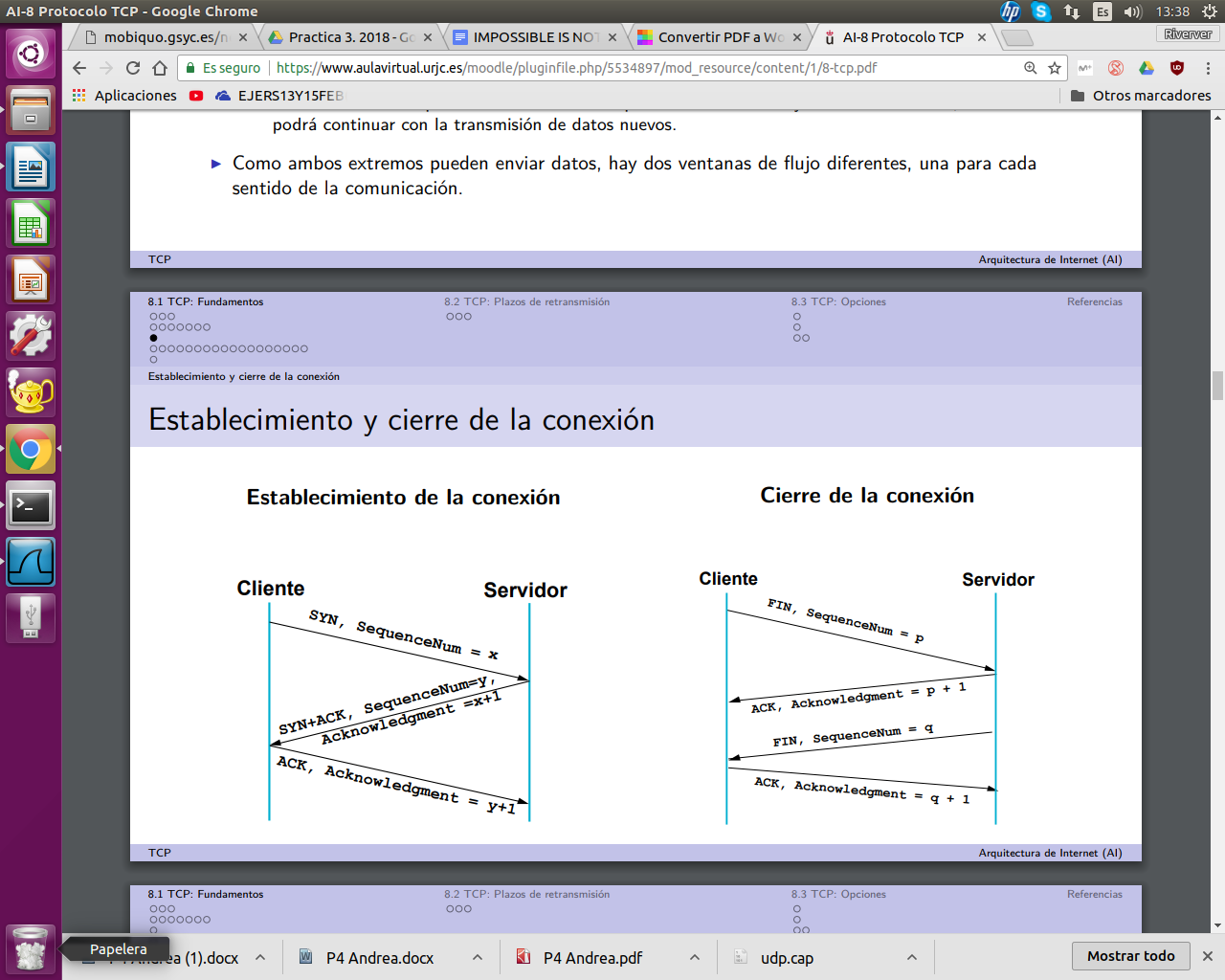


Adjuntamos explicación:

En cuanto al primer paquete TCP, en la información del mensaje vemos SYN que envía PC1 a PC2, y lleva un número de secuencia inicial asociado a la conexión del host destinatario o servidor (esto se utiliza al crear una conexión para indicar al otro extremo cual va a ser el primer número de secuencia con el que va a comenzar a transmitir (distinto de cero)). En el segundo mensaje vemos que PC2 responde con un paquete SYN + ACK (piggybacking)(acuse de recibo) confirmando la recepción del SYN inicial enviado por PC1), y enviándole a su vez su propio número de secuencia. Para finalizar, en el tercer paquete PC1 reconoce la recepción del SYN de PC2 (servidor) mediante el envío de un ACK. En este momento queda establecida la conexión y se puede iniciar la transferencia de datos entre los dos PCs.

3.1.2. Cierre de conexión.

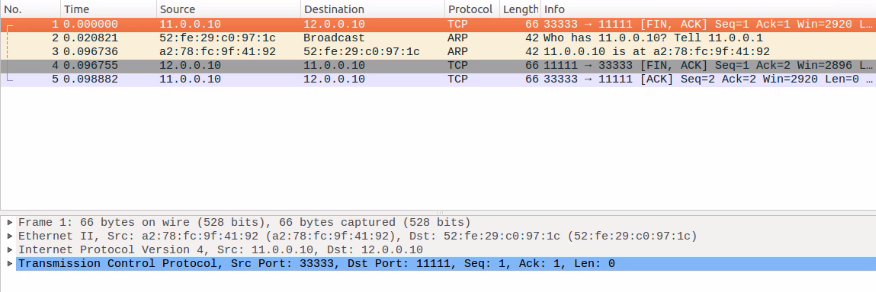
1. Inicia una captura en el router r1 y guarda el contenido en un fichero.

2. Interrumpe la ejecución del cliente(pc1) pulsando Ctrl+C en el cliente.Explica cuántos paquetes deberían haberse capturado y por qué. Interrumpe la captura y comprueba si tus respuestas se corresponden con lo observado en la captura.

**3.2. Buffer de recepción en TCP.**

Inicia una captura en el terminal de r1 guardando el contenido en un fichero.

1. Lanza una aplicación servidor utilizando nc en la máquina 12.0.0.10 que acepte conexiones en el puerto TCP 11111 (arráncala en segundo plano): nc -l -p 11111 &.

2. Lanza una aplicación cliente con nc en la máquina 11.0.0.10 para que se conecte al servidor, desde el puerto local TCP 33333: nc -p 33333 12.0.0.10 11111

3. Observa el estado que muestra netstat -tna en el servidor y sus buffers.

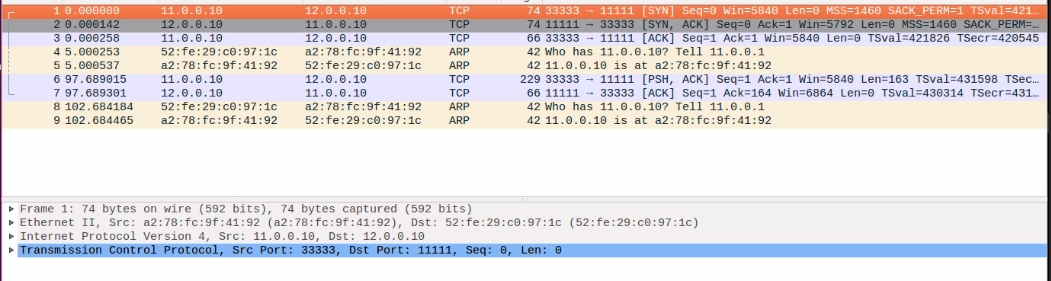
4. Trae a primer plano la ejecución de nc ejecutando en el terminal: fg

5. Pausa con Ctrl+Z la ejecución de nc en el servidor, para que la aplicación del servidor siga arrancada pero no se ejecute, por tanto, si TCP en el lado servidor recibe datos, estos no se se van a leer en nc quedarán almacenados en la cola de entrada de la implementación de TCP.

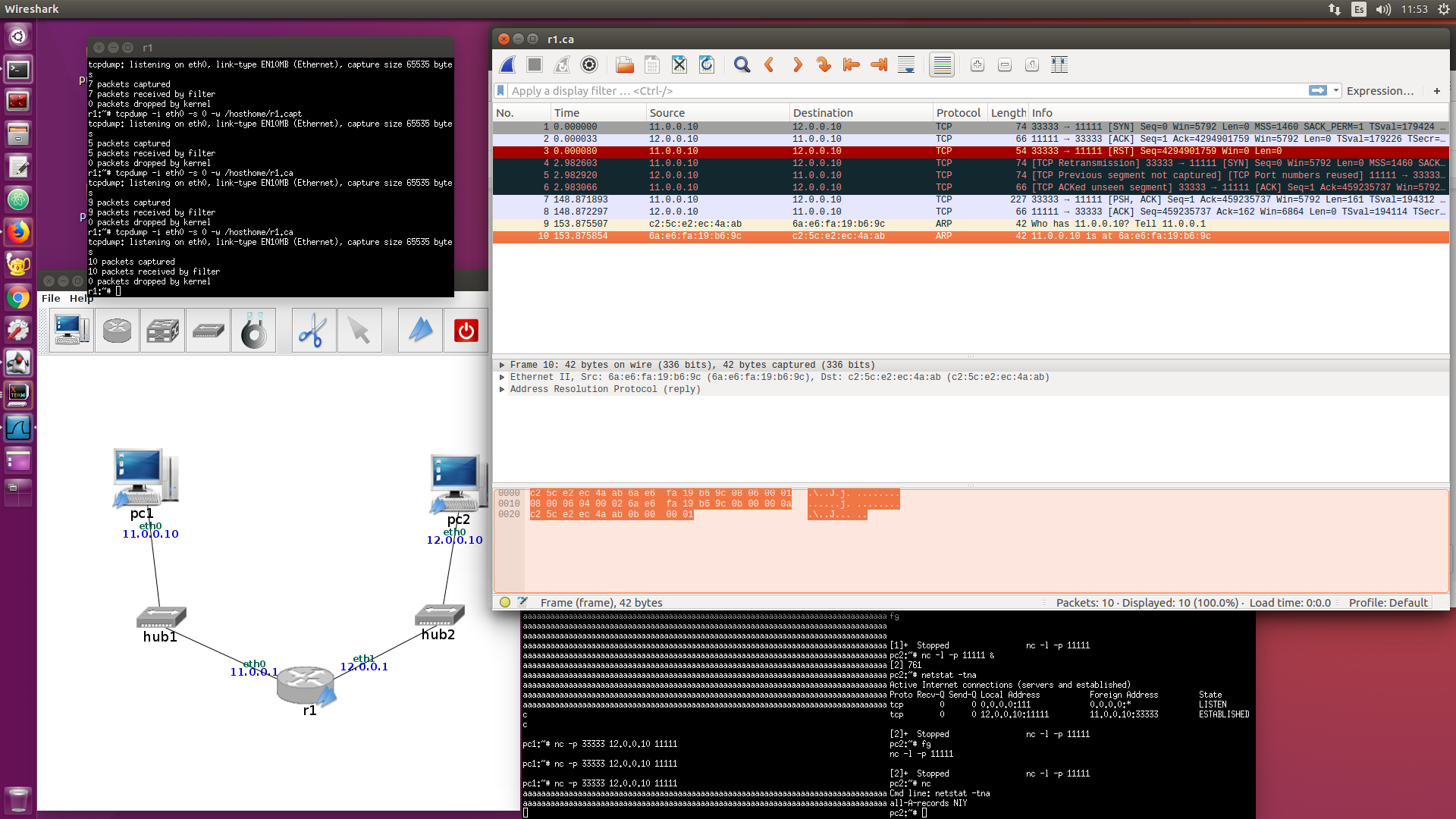
6. Para ver cómo los datos se quedan almacenados en el servidor, envía una cadena de caracteres desde el cliente y pulsa INTRO.

7. Ejecuta netstat -tna en el servidor para ver cómo esos datos se quedan en el buffer de recepción y no los lee la aplicación.

8. Interrumpe la captura y fíjate cómo hay un asentimiento que indica que todos los datos han sido recibidos. Dado que la aplicación servidor está suspendida, los datos se encuentran almacenados en el buffer de recepción de la implementación de TCP, en el kernel del sistema operativo, pero no los ha leído aún la aplicación servidora arrancada con nc.



chicas porque a mi me sale esto

TIA A MI ME SALEN 11 

9. Trae a primer plano la ejecución del servidor, para ello usa fg. Verás como los datos que habías enviado desde el cliente se muestran en la pantalla. El servidor los ha leído del buffer de recepción y el buffer está vacío. Una vez realizada la prueba puedes interrumpir la ejecución del cliente y el servidor.

Cuando ejecutamos el comando aparece la cadena de caracteres que habíamos puesto en pc1, la podemos ver en pc2.

**3.3. Errores en las comunicaciones TCP.**

1. Existe la máquina 12.0.0.10 pero no hay una aplicación escuchando en el puerto 11111. Prueba a lanzar el cliente y comprueba qué ocurre.



Al intentar hacer este envío, vemos que como estamos generando tráfico TCP y es está orientado a conexión y fiable, pues al no estar activado el servidor, no se podrá llevar a cabo este envío. Al hacer activar el comando nc –p 10000 12.0.0.10 11111, nos sale en el terminal de la máquina que desea hacer esa comunicación lo siguiente: CONECTION REFUSED, eso quiere decir que nuestra conexión ha sido rechazada.

2. Existe la red 12.0.0.0 y hay ruta para llegar hasta ella, pero no existe la máquina 12.0.0.10. (Para realizar este apartado apaga la máquina 12.0.0.10). Prueba a lanzar el cliente y comprueba qué ocurre.



