Prácticas con NetGUI
Práctica 4

Arquitectura de Redes de Ordenadores
Arquitectura de Internet

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

Abril de 2020

Resumen

En la primera parte de esta práctica se afianzan los siguientes conceptos y capacidades:

- trabajar con escenarios de red preconfigurados,
- estudiar las direcciones IP y las tablas de encaminamiento existentes en máquinas preconfiguradasm,
- realizar cambios a tablas de encaminamiento para cumplir las condiciones pedidas,
- elegir direcciones IP y máscaras de subred apropiadas,
- estudiar la salida del comando `traceroute` para deducir las rutas entre máquinas.

En la segunda parte de la práctica Se cubre el funcionamiento básico de los protocolos de nivel de transporte UDP y TCP.

Nota: Al cargar capturas en *Wireshark* es necesario ordenar los paquetes por su marca de tiempo, pulsando en la pestaña `Time`, de esta forma podremos analizar lo que ha ocurrido ordenadamente siguiendo el eje temporal.

1. Escenario A: Configuración enrutamiento IP

Descarga de la página de la asignatura el fichero `lab-p4-a.tgz`, que contiene un escenario de red. Descomprímelo de la misma manera que hiciste en la práctica anterior.

Lanza ahora NetGUI. En el menú, elige File → Open y selecciona la carpeta `lab-p4-a` en la que está el escenario. Verás aparecer la red de la figura 1.

Arranca todas las máquinas de una en una, esperando que una máquina haya terminado su arranque antes de arrancar la siguiente.

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.

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.

3. La máquina `r3` no tiene configuradas sus interfaces de red. Configura 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`.

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.

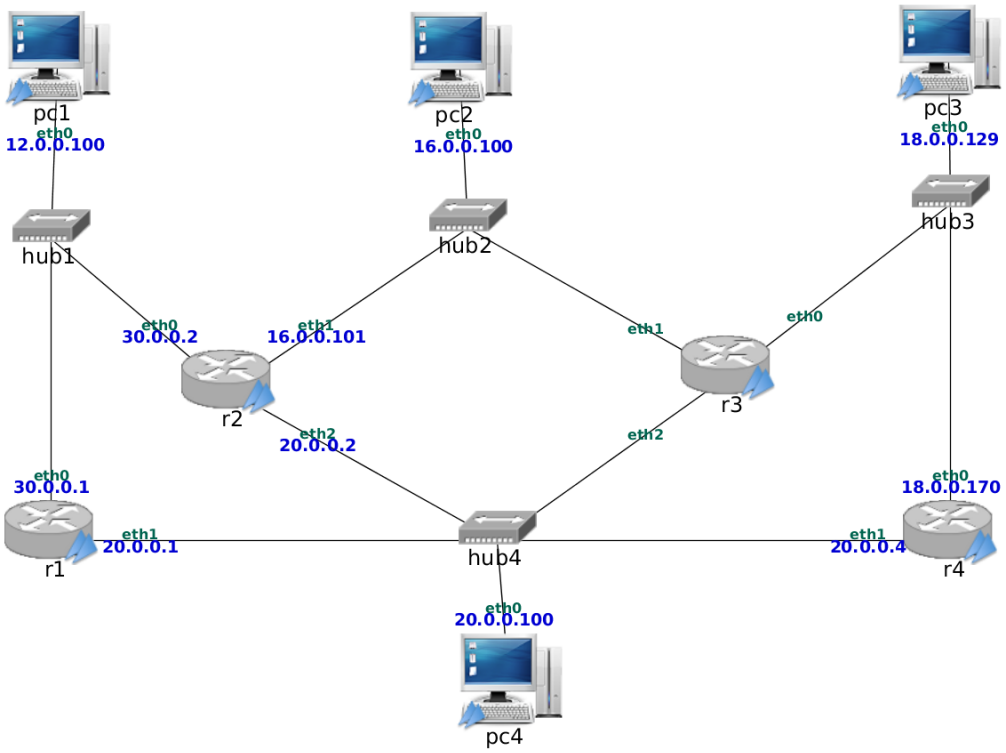

Figura 1: Escenario A

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.

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.

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.

1.1. Capturas de tráfico (ARP)

Antes de comenzar a realizar los siguientes ejercicios, espera al menos 10 minutos después de haber ejecutado el último `ping` del apartado anterior. **Importante**: asegúrate también que los PCs `pc1`, `pc4` y el *router* `r2` tienen sus tablas de enrutamiento respectivas configuradas de forma que `r2` es el *router* que encamina el tráfico de ida y vuelta entre `pc1` y `pc4`.

1. Consulta el estado de las cachés de ARP en los pcs y en los *routers*. Explica su contenido.

2. Arranca en `r2` y en `pc4` un `tcpdump` para capturar tráfico en su interfaz `eth0`, guardando la captura en un fichero (tal y como lo hiciste en la práctica 0).

3. Ejecuta en `pc1` un `ping` a `pc4` que envíe sólo 1 paquete ICMP Echo Request (`ping -c 1 <máquinaDestino>`).

4. Interrumpe las capturas en `r2` y `pc4` (`Ctrl+C`).

5. Comprueba el estado de las cachés de ARP en `pc1`, `pc4`, y `r2`. Explica su contenido.

6. Arranca en un terminal de la máquina real la aplicación `wireshark` para cargar el fichero de captura que has obtenido. Observa y anota el valor de los siguientes campos en los mensajes de la captura:

   - Mensaje de solicitud de ARP que envía `pc1` a `r2`.

        - Dirección Ethernet destino
        - Dirección Ethernet origen
        - Tipo en la cabecera Ethernet
        - Contenido del mensaje de solicitud de ARP: localiza el campo que contiene la dirección IP de la máquina sobre la que se está preguntando su dirección Ethernet.
    - Mensaje de solicitud de ARP que envía `r2` a `pc4`.
        - Dirección Ethernet destino
        - Dirección Ethernet origen
        - Tipo en la cabecera Ethernet
        - Contenido del mensaje de solicitud de ARP: localiza el campo que contiene la dirección IP de la máquina sobre la que se está preguntando su dirección Ethernet.
    - Mensaje de respuesta de ARP que envía `pc4` a `r2`.
        - Dirección Ethernet destino
        - Dirección Ethernet origen
        - Tipo en la cabecera Ethernet
        - Contenido del mensaje de respuesta de ARP: localiza el campo que contiene la dirección Ethernet solicitada.
    - Mensaje de respuesta de ARP que envía `r2` a `pc1`.
        - Dirección Ethernet destino
        - Dirección Ethernet origen
        - Tipo en la cabecera Ethernet
        - Contenido del mensaje de respuesta de ARP: localiza el campo que contiene la dirección Ethernet solicitada.
    - Datagrama IP que envía `pc1` a `pc4`.
        - Dirección Ethernet destino
        - Dirección Ethernet origen
        - Tipo en la cabecera Ethernet
        - Dirección IP origen
        - Dirección IP destino
        - Campo TTL
    - Datagrama IP que envía `pc4` a `pc1`.
        - Dirección Ethernet destino
        - Dirección Ethernet origen
        - Tipo en la cabecera Ethernet
        - Dirección IP origen
        - Dirección IP destino
        - Campo TTL

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?
2. ¿Qué información puedes extraer de la captura sobre la red en la que se ha realizado la captura?
3. ¿Cuál es el número de paquetes UDP y número de bytes de datos intercambiados?

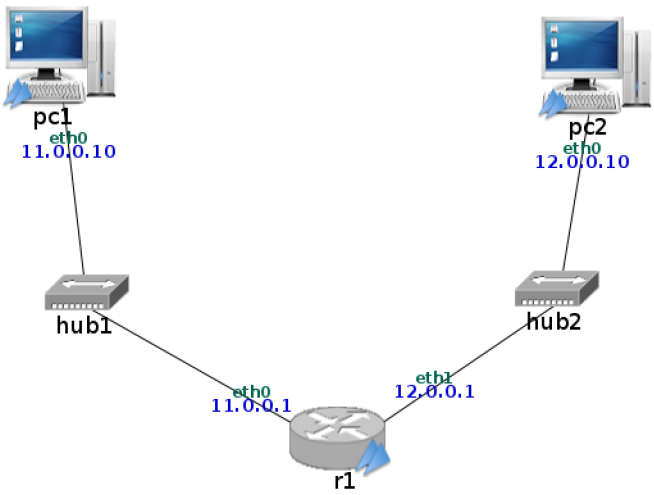

Figura 2: Escenario lab-p4-b

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

Descarga de la página de la asignatura el fichero `lab-p4-b.tgz`, que contiene un escenario de red. Descomprímelo de la misma manera que hiciste en prácticas anteriores.

Lanza ahora NetGUI. En el menú, elige File → Open y selecciona la carpeta `lab-p4-b` en la que está el escenario. Verás aparecer la red de la Figura 2.

Arranca las máquinas de una en una, esperando que una máquina haya terminado su arranque antes de arrancar la siguiente.

En este apartado utilizarás la orden `nc` para observar el funcionamiento de UDP en diversas situaciones. Ve ahora al anexo de esta práctica en el que se explica cómo utilizar nc para arrancar clientes y servidores UDP, y vuelve aquí después para continuar.

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

1. Inicia una captura en el router `r1`[1].
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.

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 fichero.
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 carácteres, cada uno de ellos ocupando 1 byte).
3. A continuación pulsa la tecla `INTRO` o `ENTER` (véase la figura 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?
2. ¿cuántos datagramas IP crees que se han enviado, y por qué?
3. ¿cuántos bytes de datos irán en cada datagrama UDP?

[1] Esta captura puedes realizarla sin necesidad de guardar el contenido en un fichero. La utilizaremos para ver los paquetes que se generan. Para ello ejecuta en r1: tcpdump -i eth0.

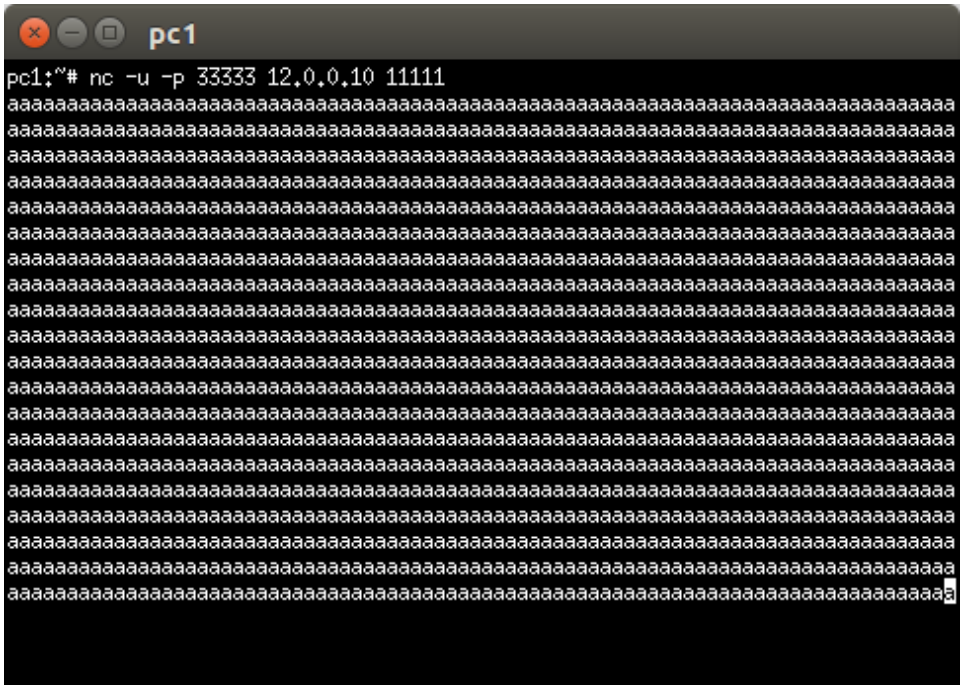

Figura 3: Tráfico UDP

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.

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

En este apartado utilizarás la orden `nc` para observar el funcionamiento de TCP en diversas situaciones. Ve ahora al anexo de esta práctica en el que se explica cómo utilizar nc para arrancar clientes y servidores TCP, y vuelve aquí después para continuar.

Utilizaremos el mismo escenario NetGUI del apartado anterior.

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.

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 pulsando `Ctrl+C` en el cliente [2]. 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

Vamos a visualizar los bufferes in/out (recepción/emisión) 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.
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.

3.3. Errores en las comunicaciones TCP

Provoca las siguientes situaciones de error:

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.
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.

[2] Con nc, una vez que se interrumple la conexión desde el cliente, el servidor también cierra la conexión.
Con otras aplicaciones podría mantenerse abierta la conexión desde el servidor si éste tuviera más datos que enviar

4. Anexo: Funcionamiento de nc

Las aplicaciones que se utilizarán en esta práctica para generar tráfico TCP y UDP siguen el modelo de comunicaciones cliente/servidor. En este modelo, cuando dos aplicaciones se comunican una de ellas funcionará como servidor y la otra funcionará como cliente.

Siempre es necesario lanzar primero la aplicación que funciona como servidor, que quedará a la espera de recibir tráfico procedente de la aplicación cliente, que se deberá lanzar después.

En esta práctica utilizaremos la aplicación `nc` para arrancar aplicaciones que utilizan TCP o UDP. `nc` puede arrancarse como cliente o como servidor, utilizando TCP o UDP como protocolo de transporte. Una aplicación `nc` lanzada como cliente se comunicará con otra lanzada como servidor y viceversa. 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`.

4.1. Tráfico UDP

4.1.1. Aplicación servidor UDP

Para arrancar una aplicación que funciona como servidor utilizando el protocolo UDP ejecutaremos la siguiente orden:

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

Donde:

- `<Pto-Loc>` es el 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
```

4.1.2. Aplicación cliente UDP

Para arrancar una aplicación que funciona como cliente utilizando el protocolo UDP ejecutaremos la siguiente orden:

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

Donde:

- `<Pto-Loc>` es el número de puerto local UDP en el que la aplicación cliente esperará recibir los datagramas UDP que vengan del servidor.
- `<IP-dest>` es la 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 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
```

4.1.3. 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.

Para interrumpir la ejecución de estas aplicaciones se debe utilizar `Ctrl+C`.

4.2. Tráfico TCP

4.2.1. Aplicación servidor TCP

Para arrancar una aplicación que funciona como servidor utilizando el protocolo TCP ejecutaremos la siguiente orden:

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

Donde:

- `<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
```

4.2.2. Aplicación cliente TCP

Para arrancar una aplicación que funciona como cliente utilizando el protocolo TCP ejecutaremos la siguiente orden:

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

Donde:

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

Por ejemplo, si queremos arrancar una aplicación cliente TCP 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
```

4.2.3. 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`.