

Práctica 6

Configuración básica de red

6.1. Objetivos

Los objetivos de la práctica son:

1. Desarrollar habilidades prácticas para la configuración de red de un sistema informático.
2. Comprender los conceptos y tecnologías relacionados con la configuración de un sistema, incluyendo protocolos, direcciones IP, enrutamiento y seguridad.
3. Conocer la tecnología de virtualización y su aplicación práctica para la configuración de un sistema.

6.2. Introducción

Para conectarse a Internet, un equipo necesita tanto hardware específico como ciertos datos de configuración. La tarjeta de red constituye el interfaz que conectará físicamente el equipo a una red. Esta tarjeta puede tener un interfaz cableado (por ejemplo, Ethernet) o inalámbrico (por ejemplo, Wi-Fi). En cuanto a los datos de configuración, el equipo necesita:

1. Una dirección IP asignada a su interfaz de red
2. La dirección IP de su enrutador por defecto
3. La dirección IP de su servidor DNS (*Domain Name System*)

Esta información la puede obtener automáticamente a través del protocolo DHCP (*Dynamic Host Configuration Protocol*) si en la red hay un servidor DHCP activo. De esta forma, al arrancar, el ordenador solicita dicha información y el servidor se la proporciona. Actualmente este procedimiento es el habitual en redes domésticas y en redes inalámbricas. En otros entornos puede ser necesario configurar manualmente estos parámetros.

Por cuestiones de seguridad y estabilidad, en los equipos de los laboratorios no tenemos los privilegios de administración necesarios para modificar la configuración de red. Así se evitan configuraciones que podrían causar problemas de conectividad, vulnerabilidades de seguridad o interferencias con otros dispositivos en la red. Por este motivo, vamos a realizar la práctica sobre una máquina virtual, sobre la que sí tendremos permisos de administración.

6.2.1. Máquina virtual

Una máquina virtual es un sistema completo (hardware virtualizado, sistema operativo invitado y aplicaciones) que es emulado por un hipervisor en un sistema anfitrión. En nuestro caso vamos a trabajar con una máquina virtual con el sistema operativo Debian 12 ejecutada por el software de virtualización QEMU.

Lo primero que haremos será preparar el entorno de trabajo:

1. Abrir sesión en **central**, el sistema que da soporte a las prácticas relacionadas con máquinas virtuales:

```
$ ssh central
```

2. Copiar la imagen del sistema operativo invitado (*guest*) y su script de arranque:

```
$ cp /misc/usuarios/chus/rc/debian-12.5.qcow2.img  
  /misc/alumnos/rc/rcyyyy/$LOGNAME/  
$ cp /misc/usuarios/chus/rc/ArrancarDebianQemu.sh  
  /misc/alumnos/rc/rcyyyy/$LOGNAME/
```

yyyy es el año de comienzo del curso, por ejemplo, 2024 para el curso 2024-25. La copia también puede hacerse desde un terminal en la máquina local, pero consumiría ancho de banda de la red del laboratorio 1.02 al tener que viajar la imagen de **central** al equipo local y del equipo local de nuevo a **central**.

Para arrancar la máquina virtual hay que hacer lo siguiente:

1. Ejecutar en la máquina local (no en **central**) el script de arranque del sistema operativo invitado:

```
$ ./ArrancarDebianQemu.sh /misc/alumnos/rc/rc2024/$LOGNAME/debian-12.5.qcow2.img
```

Este script lanza el emulador **qemu** pasándole entre otros parámetros la imagen del sistema operativo invitado.

Por cuestiones de espacio en **central**, no se permite utilizar más de una máquina virtual por estudiante ni crear copias de respaldo de la máquina virtual.

2. Una vez iniciado el sistema invitado, se abrirá una ventana emulando la pantalla de la máquina virtual, desde la que podemos acceder al sistema como usuario **rc** y contraseña **rc**.
3. Cambiar a usuario administrador (**root**) mediante el siguiente comando, con contraseña **rc**:

```
$ su -
```

Por defecto, la línea de comandos del usuario **root** muestra el carácter **#** en lugar de **\$**.

6.3. Interfaces de red

Una interfaz de red es el punto de interconexión entre un equipo y una red. El interfaz de red más habitual es una tarjeta de red (*network interface card*, NIC). Existen también interfaces de red que no son dispositivos hardware, por ejemplo, el interfaz local (*loopback interface*).

Un sistema conectado a una red TCP/IP necesita una dirección IP asociada al interfaz con dicha red. En GNU/Linux, la consulta o modificación de la configuración de interfaces se realiza mediante el comando **ip**, que sustituye a los comandos de configuración originarios de BSD (**ifconfig**, **route**, etc.). En general, cada sistema puede tener un comando específico, pero su funcionamiento suele ser similar.

Ejecuta la orden **ip address** en la máquina virtual. Obtendrás una lista detallada de las interfaces de red disponibles y sus respectivas configuraciones. A continuación, se muestra un ejemplo típico de salida y su descripción:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
   inet 127.0.0.1/8 scope host lo  
       valid_lft forever preferred_lft forever  
   inet6 ::1/128 scope host noprefixroute  
       valid_lft forever preferred_lft forever  
2: ens3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000  
   link/ether 52:54:00:12:34:56 brd ff:ff:ff:ff:ff:ff  
   altname enp0s3
```

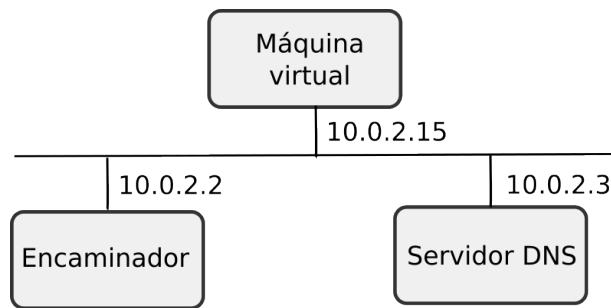


Figura 6.1: Configuración típica del modo de red de usuario QEMU¹.

1. `lo`: interfaz (*loopback*), utilizada para la comunicación interna del sistema:

- `<LOOPBACK,UP,LOWER_UP>`: indica que es una interfaz de loopback (LOOPBACK), que está activa (UP) y que está detectando una conexión correcta del cable, en este caso virtual, de red (LOWER_UP).
- Dirección IPv4: 127.0.0.1/8, que es la dirección estándar de loopback.
- Dirección IPv6: ::1/128, que es la dirección estándar de loopback en IPv6.

2. `ens3`, interfaz Ethernet virtual proporcionada por QEMU:

- Dirección de enlace: la dirección de enlace (MAC) de la tarjeta de red es 52:54:00:12:34:56, y la dirección de difusión (broadcast) es ff:ff:ff:ff:ff:ff.
- Estado: DOWN, indica que la interfaz está inactiva.

Observa que el interfaz `ens3`, que corresponde a la tarjeta de red, no tiene ninguna dirección IP asociada.

Vamos a configurar la red al igual que hace QEMU en el modo de red predeterminado mediante DHCP, pero de forma manual. En el modo de red predeterminado de QEMU, o modo *user*, QEMU actúa como un encaminador/NAT para la máquina virtual. Por defecto, la máquina virtual obtiene una dirección IP en el rango 10.0.2.x, con el anfitrión QEMU asignándose la dirección 10.0.2.2 para la puerta de enlace y 10.0.2.3 para el servidor DNS. La figura 6.1 muestra una configuración típica del modo de red de usuario. Este modo es adecuado para pruebas y desarrollo, ya que permite al sistema invitado comunicarse con los equipos de su red local y acceder a Internet sin necesidad de configuraciones adicionales de red en el sistema operativo anfitrión.

6.4. Conectividad local

La conectividad local en una red se refiere a la capacidad de los dispositivos dentro de dicha red local (LAN) para comunicarse entre sí sin necesidad de pasar por encaminadores, redes externas o Internet. A estas comunicaciones las hemos llamado en clase punto-a-punto. A continuación, se detallan las órdenes necesarias para dotar de conectividad local a nuestra máquina virtual. También ejecutaremos órdenes que nos permitan observar el efecto de las órdenes de configuración y verificar su correcto funcionamiento.

1. Asigna dirección IP y máscara de red al interfaz de red `ens3`:

```
# ip addr add 10.0.2.15/24 dev ens3
```

Esta orden asigna la dirección IP 10.0.2.15 con una máscara de subred de 24 bits a la interfaz `ens3`. Ejecuta `ip address` para comprobar el efecto de la orden ejecutada.

¹<https://wiki.qemu.org/Documentation/Networking>

2. Comprueba si tienes conectividad local. Para ello intenta conectarte con el encaminador de QEMU:
`ping 10.0.2.2`
3. Levanta la interfaz de red `ens3` (activa su uso):
`# ip link set ens3 up`
4. Ejecuta `ip address`. ¿Qué efecto ha tenido la orden ejecutada?
5. Comprueba de nuevo si tienes conectividad local: `ping 10.0.2.2`

Para ejecutar `ping`, nuestro sistema encapsula cada mensaje ICMP dentro de un paquete IP, especificando en la cabecera del paquete la dirección IP del sistema destino, que en este caso es 10.0.2.2. A su vez, encapsula cada paquete IP en una trama, especificando en la cabecera de la trama el identificador MAC del sistema destino. En clase hemos visto que en IPv4 las direcciones lógicas (IP) se asocian a los identificadores físicos (MAC) mediante el protocolo ARP (*Address Resolution Protocol*), mientras que en IPv6 esas asociaciones se gestionan a través de ICMPv6 (*Neighbour Discovery*). Es decir, cualquier equipo tiene una tabla con asociaciones entre direcciones IP y sus correspondientes identificadores MAC. Así, cuando se necesita un identificador MAC se genera un mensaje ARP (o ICMPv6) de difusión total (*broadcast*) del tipo «Quien tenga esta dirección IP, que me diga su identificador MAC». Ese mensaje llegará a todos los equipos de la red local, y el equipo aludido responderá. Con esa información se actualiza la tabla de asociaciones IP-MAC. El comando `ip neighbour` permite consultar y manipular esta tabla.

6. ¿Por qué en la salida del anterior `ping` ejecutado el tiempo del primer mensaje ICMP es superior a del resto?
7. Comprueba qué asociaciones tienes en este momento: `ip neighbour`
8. Haz un `ping` al servidor DNS de QEMU (figura 6.1).
9. Vuelve a comprobar la tabla de vecinos. ¿Qué información nueva se ha añadido?

Con lo anterior ya hemos configurado y verificado la conectividad local.

6.5. Conectividad con redes externas

La conectividad con redes externas se refiere a la capacidad de los dispositivos de una red local para comunicarse con redes fuera de su alcance directo. En nuestro caso, la máquina virtual tendrá conectividad externa si es capaz de comunicarse con equipos ubicados en otras redes, como por ejemplo, `lab000`, o servidores web alojados en Internet. A continuación, se detallan las órdenes necesarias para dotar de conectividad con redes externas a nuestra máquina virtual. También ejecutaremos órdenes que nos permitan observar el efecto de las órdenes de configuración y verificar su correcto funcionamiento.

10. Comprueba si tienes conectividad con un equipo que se encuentre en una red externa, por ejemplo 155.210.152.177 (`lab000`): `ping 155.210.152.177`

Para tener conectividad con redes externas es necesario establecer un encaminador por defecto. Esto implica configurar rutas en la tabla de reexpedición/encaminamiento. El comando `ip route` permite consultar las rutas actuales y configurar las rutas de forma estática, es decir, manualmente sin que intervenga ningún protocolo de búsqueda de caminos.

11. Lanza `ip route` para consultar las rutas actuales y comprueba que no se muestra ninguna información.
12. Especifica que la ruta por defecto es enviar los paquetes a 10.0.2.2; la puerta de enlace o encaminador por defecto que proporciona QEMU.

```
# ip route add default via 10.0.2.2
```

13. Ejecuta `ip route` para comprobar el efecto de la orden ejecutada.
14. Comprueba de nuevo si tienes conectividad externa: `ping 155.210.152.177`

Un encaminador usualmente tendrá varios interfaces de red y varias entradas en su tabla de reexpedición, pero su funcionamiento es idéntico al de un equipo como el que estamos configurando. La única diferencia es que, por defecto, cuando un ordenador recibe un mensaje del cual no es el destinatario lo descarta, mientras que un encaminador lo reexpide hacia donde corresponda según su tabla. El contenido del fichero `/proc/sys/net/ipv4/ip_forward` indica si el equipo está actuando como encaminador (1) o no (0). Si se desea cambiar este comportamiento, basta con escribir el valor correspondiente en ese fichero.

15. ¿Está actuando tu máquina virtual como encaminador?
16. ¿Y el ordenador físico sobre el que estás ejecutando QEMU?
17. Consulta la tabla de reexpedición del ordenador físico para ver que efectivamente es algo más compleja de lo habitual en un ordenador típico.

6.6. Sistema de nombres de dominio (DNS)

Con todo lo anterior correctamente configurado, el equipo ya tiene conectividad con la red local y con Internet. Aún así, existe un servicio adicional que se considera básico. Ese servicio es el sistema de nombres de dominio (DNS), que realiza traducciones de nombres *fácilmente usables por personas* a direcciones IP. Por ejemplo, este sistema permite traducir `lab000.cps.unizar.es` a `155.210.152.177`. Los nombres o *dominios* son jerárquicos. Por ejemplo, todos los nombres dentro del dominio de España acaban en `.es` y todos los equipos de la Universidad de Zaragoza acaban en `.unizar.es`. En general, como mínimo cada red dispone de dos equipos encargados de la traducción de nombres. Para que nuestro equipo los conozca, hay que especificar cómo llegar a ellos, es decir, su dirección IP. En GNU/Linux, esa especificación se encuentra en el fichero `/etc/resolv.conf`, que también incluye el dominio que añadirá por defecto a los nombres que vaya a traducir. Una forma sencilla de realizar consultas de nombres es a través del comando `host`. Revisa el manual del comando y observa la respuesta al preguntar por los siguientes nombres:

18. Comprueba si puedes utilizar nombres en lugar de direcciones IP: `host lab000.cps.unizar.es`
Esto es debido a que no se ha especificado un servidor DNS, `cat /etc/resolv.conf`

19. Añade el servidor de nombres con dirección 10.0.2.3 a tu lista de servidores:

```
# echo nameserver 10.0.2.3 > /etc/resolv.conf
```

20. Ejecuta `cat /etc/resolv.conf` para comprobar el efecto de la orden anterior.
21. Comprueba si ahora puedes traducir nombres: `host lab000.cps.unizar.es`

Por otra parte, resulta conveniente tener configurados ciertos dominios en los que buscar para traducir nombres con mayor facilidad.

22. ¿Funciona `host hendrix`?
23. En el mismo archivo de configuración de antes, *añade* un nombre de dominio en el que buscar:

```
# echo search cps.unizar.es >> /etc/resolv.conf
```

24. ¿Funciona `host hendrix`?

25. ¿Qué información se obtiene al preguntar por *unizar.es*?

El comando **dig** también permite realizar consultas de nombres de forma similar a **host**, pero proporcionando muchos más detalles. Los principales registros que DNS maneja son:

- A (Address), define la dirección IPv4
- AAAA (Address), define la dirección IPv6
- NS (Name Server), define los servidores DNS
- MX (Mail eXchanger), define los servidores de correo
- CNAME (Canonical Name), permite definir alias de otros nombres
- SOA (Start Of Authority), contiene información sobre el servidor DNS primario
- LOC (LOCation), define la localización
- TXT (TeXT): almacena cualquier información

26. Ejecuta **dig ANY unizar.es** y compara el resultado con la información obtenida con **host**.

Con todo lo anterior el equipo ya tiene su conexión de red totalmente configurada. Aún así, es necesario tener claro que lo que hemos abordado son los comandos de configuración y su uso, no la configuración permanente del sistema. Es decir, al reiniciar la máquina virtual todo volverá a su estado inicial salvo las modificaciones en el fichero **/etc/resolv.conf**. Para una configuración persistente no hay que configurar la red del sistema sino los ficheros de configuración de arranque, los cuales realizarán las acciones vistas en esta práctica. Dicha configuración es mucho más dependiente del sistema, y se aborda en asignaturas como «Administración de sistemas».

6.7. Estado de puertos

La herramienta **ss** (*socket state*) permite mostrar el estado de los puertos TCP y UDP.

27. ¿Qué hace **ss** con el argumento **-t**?

28. ¿Qué hace el argumento **-l**? ¿Por qué se muestran asteriscos en la columna *Peer Address*?

29. Lanza un **netcat** como servidor TCP y en otro terminal obtén un listado de los sockets TCP que estén en modo *listen*. ¿Puedes localizar el socket del **netcat** en el listado? ¿Qué aparece en la columna *state*?

30. Lanza ahora un **netcat** que se conecte con el **netcat** anterior. Localiza su entrada con **ss -t -a**. ¿Cuántas veces aparece? ¿En qué estado está ahora el socket? ¿Qué puertos se están utilizando en esa conexión?

31. Si pulsas Ctrl+C en uno de los **netcat**, ¿finaliza la conexión de ese **netcat** o de los dos? ¿Por qué?

32. Si inmediatamente después de cerrar la conexión ejecutas **ss -a -t** ¿en qué estado aparece la conexión?

33. Lanza ahora un **netcat** como servidor UDP y en otro terminal obtén un listado de los sockets UDP (**ss -u -l**). ¿Puedes localizar el socket del **netcat** en el listado?

34. Lanza ahora un **netcat** UDP para interactuar con el **netcat** anterior, pero sin enviar ningún texto entre ellos. ¿Cuántas entradas aparecen en el listado referidas a los sockets utilizados (**ss -u -l -a**)? ¿Cuál es su estado?

35. Escribe algo en el **netcat** *servidor*. ¿Se transmite al cliente? ¿Por qué? ¿Ha cambiado la información que muestra **ss**?
36. Escribe algo *distinto* ahora en el **netcat** *cliente*. ¿Qué ha pasado? ¿Ha cambiado la información que muestra **ss**?
37. Pulsa Ctrl+C para finalizar el servidor. ¿Ha finalizado el cliente automáticamente? ¿Cuántas entradas aparecen ahora en el listado? ¿En qué estado?
38. Sin cancelar el cliente anterior, lanza un nuevo servidor **netcat** UDP en el mismo puerto. ¿Si escribes algo en el cliente, lo recibe el nuevo servidor? ¿Por qué?

6.8. Interacción con protocolos de aplicación

En prácticas anteriores se ha utilizado la herramienta **netcat** para observar y verificar el comportamiento de aplicaciones como la de enviar vocales. Esto es generalizable para cualquier protocolo de aplicación cuyas comunicaciones estén basadas en texto, por ejemplo el protocolo HTTP.

39. Lanza el **netcat** como servidor TCP en tu equipo, por ejemplo en el puerto 32002. A continuación introduce `http://pon-aquí-tu-direccion-ip:32002/` en un navegador (usando la dirección IP de tu equipo.) ¿Qué mensaje ha recibido **netcat**?
40. Usa ahora **netcat** como cliente para realizar una petición web (`nc -C www.unizar.es 80`). Escribe exactamente lo siguiente, respetando mayúsculas y minúsculas, y sin olvidar la línea en blanco final:

```
GET / HTTP/1.1
Host: www.unizar.es
```

Observa el mensaje recibido y explica por qué es diferente al de la pregunta anterior.

6.9. Configuración avanzada

En la asignatura hemos visto muchos algoritmos que dependen de parámetros. En la mayoría de los casos estos parámetros se pueden cambiar, dependiendo del sistema operativo. En GNU/Linux se puede interactuar con el sistema mediante la función `setsockopt()` (específica para parámetros de red), el directorio `/proc/sys/` o el comando `sysctl`. Este comando proporciona y permite modificar la misma información que hay en el directorio `/proc/sys/`. Lanza `sysctl -a` para mostrar todos los parámetros (puedes filtrar los relativos a las redes con `grep net`). También puedes buscar el valor de un parámetro si conoces el nombre que tiene.

41. ¿Coincide el valor de `net.ipv4.ip_forward` con el del fichero `/proc/sys/net/ipv4/ip_forward`?
42. ¿Qué valor tiene el factor de escalado (opcional) de la ventana anunciada del protocolo TCP (`sysctl net.ipv4.tcp_window_scaling`)?
43. ¿Qué valores (mínimo, por defecto y máximo) tiene la ventana de recepción del protocolo TCP (`sysctl net.ipv4.tcp_rmem`)?
44. ¿Qué valores (mínimo, por defecto y máximo) tiene la ventana de emisión del protocolo TCP (`sysctl net.ipv4.tcp_wmem`)?

6.10. Herramienta Nmap

Otra herramienta muy interesante es **nmap**, que permite explorar y analizar muchos aspectos de las redes. Muchas exploraciones las realiza usando los sockets de forma «normal». En cambio, para hacer ciertas exploraciones menos convencionales, esta herramienta construye «manualmente» los paquetes,

es decir, rellenando los campos de las cabeceras con ciertos valores (correctos o no). En general los sistemas no permiten que cualquier usuario pueda hacer esto, así que para realizar ciertas exploraciones es necesario tener permisos de administrador. Eso sí, ten en cuenta que cualquier exploración implica que la máquina explorada debe responder, con lo que además de saber que está siendo explorada conoce la dirección de quien la está explorando.

45. Revisa el manual del comando `nmap` y haz una exploración de la red virtual de QEMU `ping`.

6.11. ¿Sabías que...?

- Aunque la mayoría de servicios y contenidos alojados en «Internet IPv6» también lo están en «Internet IPv4», hay algunos (cada vez más) que sólo están disponibles usando IPv6. Ciertas empresas proporcionan servicios gratuitos de túnel IPv6 sobre IPv4, para acceder a «Internet IPv6» desde proveedores de servicios que sólo proporcionan IPv4. Por ejemplo puedes configurar un túnel de estas características en <http://www.tunnelbroker.net/>
- La herramienta `nmap` es una de las preferidas por los hackers en el cine desde su aparición en *Matrix Reloaded* (<http://nmap.org/movies.html>).