Práctica 6

Herramientas básicas de red

6.1. Objetivos

Uso de herramientas y comandos para visualizar y configurar conexiones a Internet.

6.2. Introducción

Para conectarse a Internet, un equipo necesita los siguientes datos: dirección IP de su interfaz de red (podría tener más de uno), dirección IP de su encaminador por defecto y dirección IP de su servidor de nombres de dominio (DNS). Esta información la puede obtener automáticamente a través del protocolo Dynamic Host Configuration Protocol (DHCP) 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 esto es lo habitual en redes domésticas y en redes inalámbricas. En otros entornos es necesario configurar manualmente estos parámetros. A continuación se describe el uso de algunas herramientas para consultar y modificar la configuración de red y para comprobar su correcto funcionamiento.

6.3. Interfaces de red

Un 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 Internet 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 `ifconfig`, mientras que en Windows se usa `ipconfig`. Para interfaces inalámbricos, `iwconfig` permite visualizar y configurar sus características específicas. Las distribuciones modernas de GNU/Linux están en proceso de sustituir los comandos de configuración originarios de BSD por el comando `ip`. Este comando engloba los anteriores y además permite configurar elementos como multicast, túneles, control de tráfico, IPsec, etc. Puedes encontrar ejemplos de uso de `ip` en Internet[1]. En general, cada sistema puede tener un comando específico, aunque su funcionamiento suele ser similar.

Conéctate por `ssh` a `lab000.cps.unizar.es` y ejecuta el comando `ifconfig` (situado en `/sbin/`). Cada interfaz (a la izquierda) muestra detalles sobre su configuración, incluyendo direcciones, datos transmitidos (TX) y datos recibidos (RX) hasta el momento. En un equipo de usuario, los interfaces tienen nombres como `eno0` o `enp2s0` para Ethernet (anteriormente `eth0`[2]), `wls1` para wlan (anteriormente `wlan1`), o tener etiquetas dependientes de la tarjeta de red.

1. A partir de la configuración, ¿qué interfaz tiene una dirección IPv4 asociada? ¿Cuál es su MTU?

[1] `http://dougvitale.wordpress.com/2011/12/21/deprecated-linux-networking-commands-and-their-replacements/`

[2] Desde la versión v197 de `systemd/udev` se asignan nombres predecibles y persistentes a los interfaces de red. Más información en `https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/`

2. ¿Qué es el interfaz `lo`? ¿Cuál es su MTU? ¿Qué direcciones IPv4 e IPv6 tiene asignadas? ¿Por qué no tiene asociada una dirección hardware (MAC) como otros interfaces?

3. Ejecuta el comando `ip addr` y comprueba que esencialmente aparece la misma información.

4. ¿Cuál de los dos comandos muestra el tiempo de validez de las direcciones IP asignadas? Observa que se va decrementando con el tiempo.

Los equipos del laboratorio 1.02, además, tienen configurados puentes (*bridges*) virtuales (`br0`, `virbr0`). Estos conmutadores virtuales se configuran con el comando `brctl` y permiten conectar máquinas virtuales de forma equivalente a como se conectarían si fueran máquinas reales. A partir de ahora vamos a trabajar en la máquina local (lab102-yyy, siendo yyy un número entre 191 y 210).

5. Ejecuta `ifconfig` y observa la salida. ¿Cuántas tarjetas de red hay?

6. Ejecuta `brctl show br0` y observa la salida. ¿A qué interfaz de red está conectado el puerto `br0`?

7. Ejecuta `brctl showmacs br0` y observa la salida. ¿Qué información se está mostrando?

8. Ejecuta `brctl showstp br0` y observa la salida. ¿A qué te suenan los parámetros mostrados?

6.4. Conectividad local

Si estuviéramos en una red aislada (sin encaminador hacia Internet) tendríamos conectividad local, es decir, podríamos comunicarnos con el resto de equipos conectados a esa misma red local. Para ello, el ordenador construirá paquetes cuya dirección IP destino conoce (es con quien se quiere comunicar), pero posiblemente no conozca su identificador MAC, que debe especificar como destino en la cabecera ethernet (o el protocolo de interfaz de red que corresponda), así que necesita algún medio para obtener esa información. 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 `arp` permite ver y manipular esta tabla de asociaciones en IPv4, mientras que el comando `ip neigh` funciona tanto para IPv4 como para IPv6.

9. Comprueba qué asociaciones tienes en este momento.

10. Haz un `ping` a algún equipo del laboratorio cuya dirección no esté entre las asociaciones anteriores y posteriormente vuelve a comprobar la tabla de vecinos. ¿Qué información nueva se ha añadido?

6.5. Tablas de reexpedición/encaminamiento

Una función importante del nivel de red es la reexpedición del tráfico. La tabla de reexpedición/encaminamiento especifica hacia dónde hay que mandar un paquete dependiendo de su dirección destino.

6.5.1. IPv4

El comando `route` (actualizado por 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 óptimos. Lanzado sin argumentos, el comando `route` muestra la tabla de reexpedición/encaminamiento de tu equipo. Observa también la información mostrada por el comando `ip route`.

11. ¿Cuál es el encaminador por defecto de tu equipo?

12. ¿A qué redes está conectado tu equipo? ¿Cuál es la máscara de cada una de ellas en formato CIDR? ¿Se corresponden los bits a los mostrados en *Genmask*?

13. ¿Qué indica el que para ciertos destino no haya *vía* para llegar a ellos?

14. ¿Cuál sería el comando para añadir una ruta a la red 145.145.20.0 con máscara 255.255.255.0, pasando por el encaminador (*gateway*/siguiente salto) 145.145.20.1 a través del interfaz eth2?

15. ¿Qué indica *Metric*?

En GNU/Linux, uno de los métodos de comunicación entre el núcleo y el resto del sistema es a través del sistema de ficheros virtual montado en `/proc`, donde cada fichero es una especie de variable con cierto valor (o valores). Por ejemplo, el contenido del fichero `/proc/sys/net/ipv4/ip_forward` indica si el equipo está actuando como encaminador (1) o no (0). Si el equipo no actúa como encaminador, cuando recibe un paquete con una dirección IP que no le pertenece, directamente lo descarta. En cambio, si está actuando como encaminador, reexpedirá ese paquete como corresponda según su tabla.

16. ¿Está actuando tu equipo como encaminador?

Por otro lado, 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` (en `/sbin`). 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.

17. ¿Coincide el valor de `net.ipv4.ip_forward` con el de la pregunta anterior?

18. ¿En qué se diferencia la tabla de un ordenador normal de la de un encaminador?

6.5.2. IPv6

La configuración usando IPv6 es muy similar a la de IPv4. La diferencia más importante es que en IPv6 existe la autoconfiguración *sin estado*, en la que los equipos rellenan automáticamente los campos de sus direcciones IP con la información que conocen. De esta forma, al arrancar un equipo se autoconfigura como mínimo con una dirección localmente válida.

Entra en `http://test-ipv6.com/` para comprobar tu conectividad IPv6 y observa los resultados. Revisa las *Pruebas ejecutadas* y la *Información técnica*.

19. ¿Qué puedes deducir?

Revisa las direcciones IPv6 en las transparencias de clase. Busca en el manual de `route` cómo mostrar las rutas del protocolo IPv6 y muéstralas. Haz lo mismo para el nuevo comando `ip route`.

20. ¿Hay alguna red con dirección globalmente única en los destinos? ¿Hay encaminador por defecto? ¿Qué implica eso?

21. ¿Qué simboliza el prefijo `fe80`? ¿Qué implica lo que aparece en sus campos *Next Hop*?

22. ¿Qué simboliza el prefijo `ff`?

6.6. Servidores de nombres de dominio

Con todo lo anterior correctamente configurado, el equipo ya tiene conectividad a Internet. Aún así, existe un servicio adicional que se considera básico. Ese servicio es el servidor de nombres, que realiza traducciones de nombres *fácilmente usables por personas* a direcciones IP. 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.

23. ¿Cuáles son los servidores de nombres de tu máquina?
24. ¿Por qué están especificados con su dirección IP y no con su nombre?
25. ¿Qué pasaría si fallaran los dos?
26. Si tienes una conexión TCP activa con *www.google.com*, ¿qué pasaría con esa conexión si todos los servidores de nombres fallaran?

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:

27. *hendrix*
28. *moodle* (¿por qué *hendrix* funciona y *moodle* no?)
29. *moodle.unizar.es*
30. *hendrix.* (no te dejes el . final)
31. *moodle.unizar.es.* (no te dejes el . final) (¿por qué *moodle.unizar.es.* funciona y *hendrix.* no?)
32. *unizar.es.* (no te dejes el . final)
33. *unizar.es* (¿por qué *unizar.es* funciona con y sin . final?)
34. *www.unizar.es.*
35. Además de traducción de direcciones, ¿qué otras dos informaciones te han aparecido en algunas de las consultas anteriores?

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

36. Ejecuta `dig ANY unizar.es` y compara el resultado con la información obtenida con `host`.

En cuanto a los servidores de nombres de dominio, el uso de IPv6 simplemente implica manejar un nuevo *tipo* de información: las direcciones IPv6. Si las direcciones de IPv4 (32 bits) se simbolizan con tipo A (*address*), las direcciones IPv6 (4 veces mayores) se simbolizan con tipo AAAA.

37. Pregunta por tipos AAAA en *google* (`dig AAAA google.com`). ¿Responde con una dirección IPv6?

Obtén la dirección IP de los siguientes nombres:

38. *ipv6.google.com.*

39. *www.v6.facebook.com.*

40. ¿Qué obtienes si haces `ping` de *ipv6.google.com*? ¿Por qué?

41. ¿Y si haces `ping6`? ¿Por qué?

6.7. Estado de puertos

La herramienta `netstat` permite visualizar múltiples datos de red del ordenador en el que nos encontramos, incluyendo información de rutas e interfaces. No obstante, se usa particularmente para mostrar el estado de los puertos TCP y UDP. Al igual que otros comandos que has visto en la asignatura, esta herramienta está siendo sustituida, en este caso por el comando `ss` (*socket statistics*) en GNU/Linux.

42. ¿Qué hace `netstat` con el argumento `-t`? ¿Qué muestran las columnas *Local Address* y *Foreign Address*?

43. Prueba ahora `ss -t` ¿Se parece?

44. ¿Qué hace el argumento `-l`? ¿Por qué se muestran asteriscos en la columna *Foreign Address* (*Peer Address* en `ss`)?

45. ¿Qué hace el argumento `-a`? Observa que además de sockets TCP/UDP aparecen también los *UNIX domain sockets* asociados a un fichero, que habrás estudiado en Sistemas Operativos.

46. Lanza un `netcat` como servidor TCP y en otro terminal obtén un listado de los sockets TCP que estén en modo *listen* (e.g. `ss -l -t`). ¿Puedes localizar el socket del `netcat` en el listado? ¿Qué aparece en la columna *state*?

47. Lanza ahora un `netcat` que se conecte con el `netcat` anterior. Localiza su entrada con netstat o ss (e.g. `ss -t -a`). ¿Cuántas veces aparece? ¿En qué estado está ahora el socket? ¿Qué puertos se están utilizando en esa conexión?

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

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

50. 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?

51. 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?

52. Escribe algo en el `netcat` *servidor*. ¿Se transmite al cliente? ¿Por qué? ¿Ha cambiado la información que muestra `ss`?

53. Escribe algo *distinto* ahora en el `netcat` *cliente*. ¿Qué ha pasado? ¿Ha cambiado la información que muestra `ss`?

54. Pulsa Ctrl+C para finalizar el servidor. ¿Ha finalizado el cliente automáticamente? ¿Cuántas entradas aparecen ahora en el listado? ¿En qué estado?

55. 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?

Al igual que con los parámetros de la capa de red anteriores, puedes mostrar también los valores de parámetros de capa de transporte mediante el comando `sysctl`.

56. ¿Qué valor tiene el factor de escalado (opcional) de la ventana anunciada del protocolo TCP (`sysctl net.ipv4.tcp_window_scaling`)?

57. ¿Qué valores (mínimo, por defecto y máximo) tiene la ventana de recepción del protocolo TCP (`sysctl net.ipv4.tcp_rmem`)?

58. ¿Qué valores (mínimo, por defecto y máximo) tiene la ventana de emisión del protocolo TCP (`sysctl net.ipv4.tcp_wmem`)?

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.

59. 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`?

60. Usa ahora netcat como cliente para realizar una petición web (`nc -C wwww.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. 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.

61. Revisa el manual del comando `nmap` y haz una exploración de la red del laboratorio mediante `ping`.

6.10. ¿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`).