

Administración: TCP/IP

En este cuaderno encontrarás el material necesario para preparar el trabajo a realizar en la sesión de prácticas en la que configuraremos una red local (LAN, *Local Area Network*) con tecnología Ethernet. En el laboratorio emplearemos cableado (medios de transmisión) y *switches* (equipos de conmutación) para conectar varios *host* en una LAN. La preparación se puede llevar a cabo mediante varias máquinas virtuales, conectadas entre sí también por un hardware virtual.

Objetivo

La finalidad de esta práctica es administrar la configuración IP de las interfaces de un grupo de ordenadores para **configurar una red local**, así como **verificar su correcto funcionamiento**. Este objetivo general lo estructuraremos en los pasos siguientes:

1. **Interconexión física** de los *host* que formarán la LAN. Con este fin se usarán los medios hardware enumerados en el siguiente apartado y cuyas características se detallan más adelante.
2. Establecimiento de la **configuración IP** adecuada para que los *host* sean capaces de comunicarse entre sí. Esto implicará editar archivos de configuración y administrar el servicio de red de la distribución de GNU/Linux empleada.
3. **Verificación de la comunicación** entre los *host* que forman la LAN mediante las utilidades habituales para este tipo de tareas.

Medios

Además de los materiales de estudio, para completar esta práctica se emplearán los siguientes medios:

Hardware

- Dos o más ordenadores a comunicar entre sí a través de sus interfaces de red
- Un conmutador (*switch*) que permita interconectar las interfaces de los ordenadores
- Cables Ethernet para realizar las conexiones físicas

Software

- Consola de GNU/Linux para ejecutar comandos
- Editor de textos básico para editar archivos de configuración
- Utilidades de verificación de conexión

Nota

La preparación de esta práctica en tu ordenador requerirá el uso de al menos **dos máquinas virtuales**. El propio software de virtualización ofrece conmutación por software, por lo que no es preciso el *switch* ni los cables que sí usarás en el laboratorio de prácticas.

Descripción del hardware

En el montaje de una LAN es posible emplear distintas tecnologías: fibra óptica, WiFi y Ethernet son las más comunes. En nuestro caso recurriremos a esta última, por lo que precisaremos **interfaces de red** (NIC, *Network Interface Controller*) Ethernet, un **conmutador** Ethernet y los **cables** necesarios para establecer las conexiones, elementos representados en la siguiente figura.

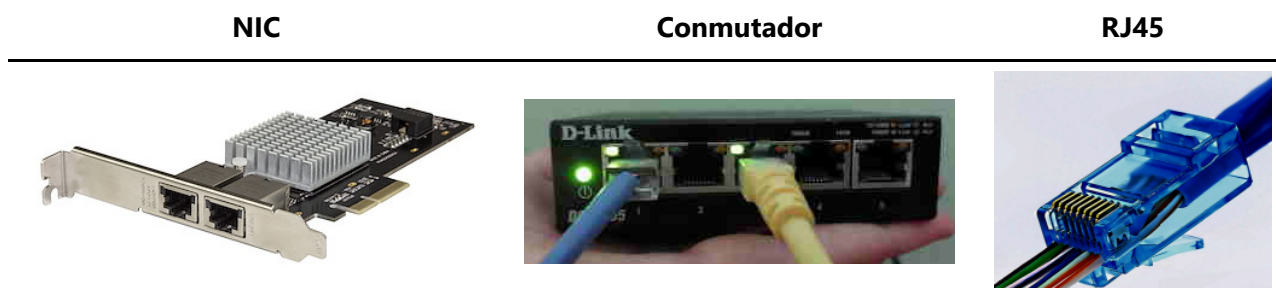


Figura 1. De izquierda a derecha: tarjeta de expansión PCI con dos interfaces de red Ethernet, conmutador Ethernet de cinco puertos y conector RJ45 con el cable de ocho hilos antes de cortar.

Interfaz de red Ethernet

La mayoría de ordenadores modernos, tanto de tipo sobremesa como portátiles, suelen incluir al menos una interfaz de red Ethernet como parte de su configuración básica. No obstante, en caso necesario es posible agregar un NIC o **controlador Ethernet** mediante la instalación de una tarjeta de expansión, similar a la de la figura siguiente, en el bus PCI de la placa base. Estas tarjetas, según su modelo, aportan una, dos o cuatro interfaces de red.

Una interfaz de red contemplará, según el estándar IEEE 802.3 que implemente, velocidades de transmisión de 10Mb/s, 100Mb/s, 1Gb/s, 10Gb/s o más. En la [página web del IEEE 802.3 Working Group](#) encontrarás información detallada de la evolución del estándar Ethernet y las velocidades de transmisión.

Nota

La notación Mb/s hace referencia a la unidad de medida **megabit por segundo** (un millón de bits por segundo). Es también común escribir **Mbit/s** y **Mbps**. No hay que confundirla con MB/s o MBps que denotan **megabytes por segundo**.

Conmutador Ethernet

La interconexión de varios *host* para formar una LAN implica contar con un **equipo de conmutación**. Lo habitual es usar un *switch* con tantos puertos como sea preciso. En la imagen anterior se aprecian las cinco interfaces con que cuenta un conmutador doméstico.

La gama del conmutador afecta no solo al número de puertos/interfaces con que cuenta, sino también a la **velocidad de transmisión máxima** a la que estas son capaces de operar. Como se aprecia en la anterior imagen, ese conmutador ofrece velocidades de 10Mb/s, 100Mb/s y 1000Mb/s. Los led asociados a cada puerto denotan la velocidad a la que está funcionando.

Cableado Ethernet

Para conectar las interfaces de red de los *host* a las del conmutador, de forma que pueda establecerse un intercambio de datos entre ellas, precisaremos un **cableado y conexionado** adecuados. Cada interfaz de red cuenta con un conector RJ45 *hembra*, por lo que los cables tendrán en sus extremos la terminación *macho* (véase parte central de la imagen inferior) de ese mismo tipo de conector.

Cada uno de los ocho hilos tiene asignada una señal determinada para la transmisión/recepción de datos. Es habitual emplear un **código de colores estándar** de forma que cada señal está vinculada al hilo que tiene un cierto color o pareja de colores.

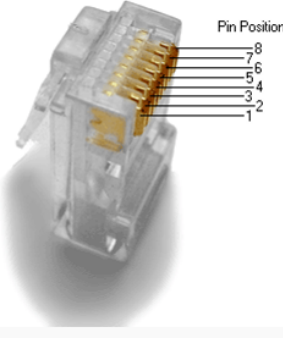
Pin	Función	568A	568B	Posición de los pines	Gigabit Ethernet (variante A)	Gigabit Ethernet (variante B)
1	Transceive data +	Blanco - Verde	Blanco - Naranja		Blanco - Naranja	Blanco - Verde
2	Transceive data -	Verde	Naranja		Naranja	Verde
3	Receive data +	Blanco - Naranja	Blanco - Verde		Blanco - Verde	Blanco - Naranja
4	Bi-directional data +	Azul	Azul		Azul	Blanco - Marrón
5	Bi-directional data -	Blanco - Azul	Blanco - Azul		Blanco - Azul	Marrón
6	Receive data -	Naranja	Verde		Verde	Naranja
7	Bi-directional data +	Blanco - Marrón	Blanco - Marrón		Blanco - Marrón	Azul
8	Bi-directional data -	Marrón	Marrón		Marrón	Blanco - Azul

Figura 2. Señales de cada uno de los ocho hilos y código de colores según distintos estándares.

Nota

El cableado Ethernet tendrá unas características u otras **según su categoría**. Esta determina, por ejemplo, el tipo de blindaje y trenzado, la existencia o no de separación física entre pares de cables, etc. De esta forma se consigue mayor resistencia a interferencias electromagnéticas y ampliar el ancho de banda que, a la postre, determinará la velocidad de transmisión máxima. En la actualidad son de uso común las categorías Cat6 (1Gb/s) y Cat6A/Cat7 (10Gb/s), pero en instalaciones antiguas se sigue usando cableado Cat5 (10Mb/s) e inferior.

Conexionado de la LAN

Una vez que disponemos de los elementos hardware ya descritos, el conexionado de los *host* al conmutador dará lugar a una **topología en estrella** como la representada en la Figura 3, en la que el *switch* actúa como concentrador/distribuidor de los paquetes de datos originados en los ordenadores.

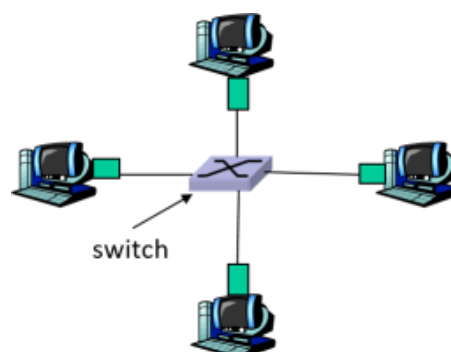


Figura 3. El conmutador actúa como nodo central de una LAN con topología en estrella.

EJERCICIO 1

Usando los materiales facilitados en el laboratorio, un conmutador por grupo de estudiantes y el número de cables necesario, lleva a cabo el montaje físico de tu LAN. Para ello sigue los pasos indicados a continuación:

1. Comienza por conectar la alimentación eléctrica del conmutador y, si es preciso, encenderlo.
2. Conecta el extremo de cada uno de los cables a la interfaz de red de cada uno de los ordenadores que se unirán a la red.
3. Conecta el otro extremo de cada cable a uno de los puertos del conmutador. En el momento en que se active dicha conexión, el led de la interfaz de red de ordenador debería mostrar actividad.

Para preparar esta parte de la práctica en casa, antes de hacerlo en el laboratorio, lo más cómodo es contar con **dos o más máquinas virtuales** que deberán compartir la configuración de red mostrada en la Figura 4. Esta consiste en tener un segundo adaptador de red habilitado, conectado a una **red interna** cuyo **nombre será común** a las máquinas virtuales. Para asignar una dirección MAC/Ethernet distinta a cada adaptador no olvides usar el botón que hay a la derecha de Dirección MAC.

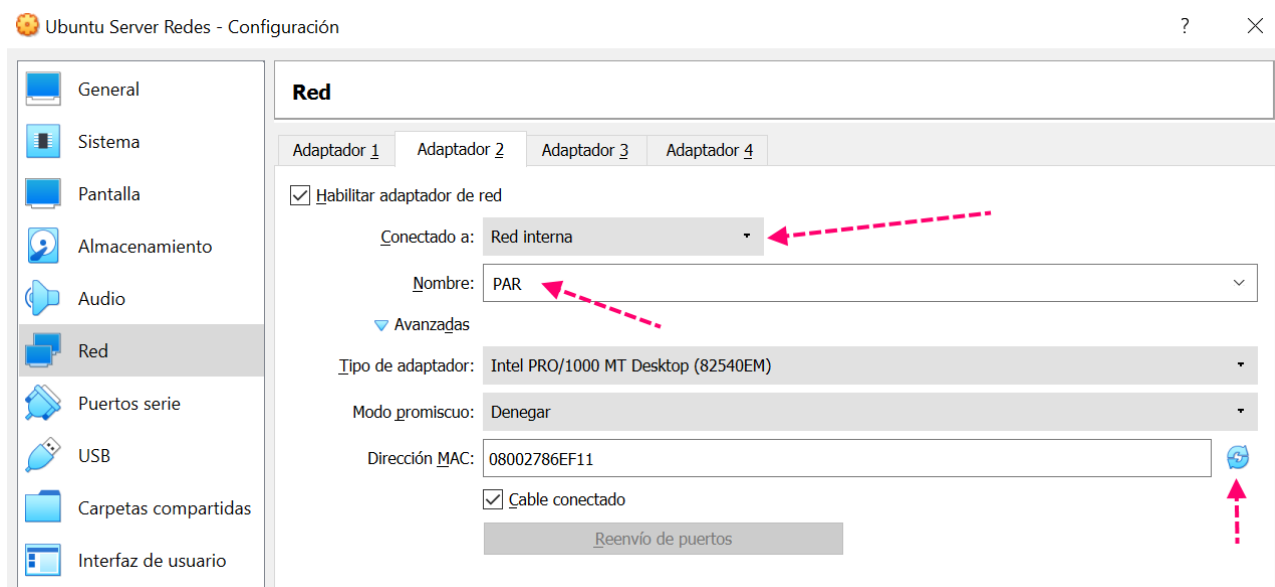


Figura 4. Las máquinas virtuales deberán tener configurada una interfaz conectada a una red interna.

Conceptos básicos sobre la configuración LAN

Antes de proceder a la configuración software de nuestra LAN y su posterior verificación, en este apartado se recopilan los conceptos sobre TCP/IP (ya abordados en las sesiones de teoría) que debemos tener presentes a la hora de completar esta práctica.

Direcciones IP en una LAN

Como ya sabemos, todo *host* conectado a una LAN que usa TCP/IP deberá tener asignada una dirección IP, de lo contrario no podrá comunicarse con los demás dispositivos. Al configurar estas direcciones IP hemos de tener en cuenta los siguientes aspectos:

- Todos los dispositivos conectados a un mismo *switch* Ethernet suelen pertenecer a la misma LAN y, por tanto, **comparten un mismo identificador de red**.
- Dentro de esa red, cada uno de los equipos tendrá asignado un **identificador de *host***
- La máscara de red o la notación CIDR indicarán qué parte de la dirección IP actúa como identificador de red y qué parte identifica al *host*. Por ejemplo:
 - IP: 150.214.178.112/24
 - Máscara: 255.255.255.0 (24 bits a 1 como ID de red)
 - ID de red: 150.214.178.0
 - ID de *host*: 112
- Al asignar identificadores a los *host* debe tenerse en cuenta que el primero, con todos los bits a 0, y el último, todos los bits a 1, no pueden usarse, ya que sirven como dirección de red y de multidifusión, respectivamente. Por ejemplo, para la red 150.214.178.X/24:
 - ID de red: 150.214.178.0 -- 8 bits del ID de *host* a 0
 - Dirección de *broadcast*: 150.214.178.255 -- 8 bits del ID de *host* a 1
- Para esta red habitualmente el *router* tendrá asignada la dirección 150.214.178.1, aunque esto no es obligatorio.

Otros conceptos sobre configuración TCP/IP

Además de las anteriores nociones sobre el uso de direcciones IPv4, la configuración de una red TCP/IP es habitual que también requiera algún conocimiento sobre los siguientes aspectos:

- **Puerta de enlace:** también conocida como *router* o *gateway* según el contexto. Es el equipo de interconexión que da salida a una red hacia otra, por ejemplo para formar una WAN con varias LAN o conectar una LAN a Internet.
- **Nombres en Internet:** a las personas no les es fácil, en general, recordar direcciones IP, razón por la que se asignan a los servidores nombres como www.ujaen.es o www.cnn.com, mucho más fáciles de recordar.
- **Archivo *hosts*:** mantiene una base de datos local con nombres de máquinas y las direcciones IP asociadas. En GNU/Linux se encuentra en `/etc/hosts`, mientras que en Windows suele estar en `C:\Windows\System32\Drivers\etc\hosts`.
- **Servidor de nombres (DNS, *Domain Name System*):** servidor que convierte nombres en direcciones IP o viceversa.
- **Servidor WINS (Windows):** servidor de nombres Windows, habitualmente asociado a servicios de red NetBIOS. Su ámbito suele restringirse al de la red local.
- **Servidor DHCP:** servidor que asigna la configuración TCP/IP a los equipos de una red en el momento en que se conectan a ella.

Nota

La LAN que montaremos en esta práctica no contará con un servidor DHCP, cuya configuración se aborda en una práctica posterior, y tampoco dispondrá de una puerta de enlace propia o un servidor DNS. Será, por tanto, una LAN que solo permitirá comunicarse a los equipos conectados físicamente a ella. El uso del archivo `hosts` permitiría a cada equipo de la LAN vincular las direcciones IP de los demás *host* con un nombre.

Direcciones MAC/Ethernet

En nuestra LAN, tal y como se ha indicado antes, cada interfaz de red de los *host* contará con una **dirección IP propia y única** en el ámbito de dicha red. Sin embargo, las interfaces de red del conmutador (los puertos del *switch*) no tendrán asignadas direcciones IP. La razón, como ya sabemos de sesiones previas de teoría, es que los conmutadores LAN operan en la **capa de enlace** y no en la de red.

Todas las interfaces de red, tanto las de los *host* como los puertos del conmutador o *switch*, tienen asignada desde fábrica una dirección única a la que se denomina MAC (*Media Access Control*), también conocida como **dirección física** o dirección Ethernet.

Las direcciones MAC tienen una longitud de 6 bytes (48 bits) con una estructura plana, lo que implica que hay 2^{48} direcciones posibles (más de 280 billones). Estas direcciones se expresan habitualmente en formato hexadecimal: `3F:5D:31:4A:CD:A1`. Recuerda que cada dígito hexadecimal toma un valor entre 0 y F (15 en decimal) para el cual se usan 4 bits, por lo que cada dos dígitos hexadecimales es un byte.

Nota

Para el envío físico de un datagrama entre los *host* de una LAN, hay que operar al nivel de la capa de enlace debido al encapsulamiento, lo que conlleva la necesidad de traducir la dirección IP (capa de red) del destino a su dirección física.

Desde la *shell* de GNU/Linux es posible obtener la dirección MAC de las interfaces de red por diferentes vías. El siguiente ejemplo muestra cómo obtener la lista de interfaces (sus nombres en nuestro sistema) y a partir del nombre de una de ellas recuperar su dirección MAC.

```
$ ls /sys/class/net/
eth0  lo  wifi0  wifi1  wifi2
$
$ cat /sys/class/net/eth0/address
38:d5:47:3d:e0:be
```

Ejemplo. Obtener la dirección MAC de la primera interfaz de red disponible.

Interfaces de red en GNU/Linux

Conceptualmente una interfaz de red es similar a la interfaz que ofrece la capa de enlace a la capa de red, de forma que **oculta todos los detalles específicos** de la tecnología subyacente (Ethernet, WiFi, etc.) para el transporte de datagramas.

En GNU/Linux era tradicional que las interfaces de red se llamasen `eth0`, `eth1`, ..., por cada una de las interfaces de red en el equipo. En la actualidad, sin embargo, el nombre cambia según el tipo de adaptador. Por ejemplo `wlan0`, `wlan1`, ... para las interfaces Wifi, o `enp0sN` para las de tipo Ethernet. Para obtener la lista de interfaces de red existentes en nuestro equipo usaremos el comando `ip l`.

Cada interfaz de red contará con su propia configuración TCP/IP, entre la que se incluirá la dirección IP asignada. Para establecer dicha configuración tenemos dos alternativas: recurrir a las utilidades de la interfaz gráfica de usuario de nuestro escritorio, vía habitual **para el usuario final**, o bien mediante la línea de comandos y la edición de archivos de configuración. Esta última será la vía que **usaremos de forma preferente** en las prácticas. Por ejemplo, con el comando `ip a` obtendremos la lista de interfaces y su configuración actual entre la que estarían las direcciones IPv4 e IPv6 asignadas.

Administrar la configuración TCP/IP - Utilidad `ip`

Una de las utilidades esenciales para la administración de la configuración de red en GNU/Linux es el comando `ip`. Este permite tanto obtener como modificar distintos aspectos. La sintaxis general de uso es la siguiente:

```
ip [opciones] elemento [acción]
```

Con `elemento` se selecciona el tipo de objeto sobre el que quiere actuarse. Los de uso más habitual son:

- `link`: interfaces de red
- `address`: direcciones IPv4 e IPv6 asociadas a las interfaces
- `route`: entradas en la tabla de enrutamiento
- `neighbour`: entradas en la caché ARP

El nombre del tipo de elemento se puede abreviar siempre que no haya confusiones, por lo que es posible usar `ip a` en lugar de `ip address` o `ip l` en lugar de `ip link`, por poner un par de ejemplos.

Con la `acción` se indicará la operación a realizar sobre el objeto. Las más usuales son las siguientes:

- `list`: enumerar las interfaces, direcciones o entradas
- `add`: agregar un elemento, por ejemplo una nueva entrada en la tabla de enrutamiento
- `delete`: eliminar un elemento, operación que complementa a la anterior
- `set`: modificar la configuración de un elemento
- `show`: mostrar detalles relativos a un elemento

Nota

Recuerda que siempre puedes usar el comando `man orden` para obtener información detallada de todas las opciones y formas de uso de un programa.

Enumeración de interfaces y sus configuraciones

En ausencia de una acción explícita, `ip` llevará a cabo la acción por defecto sobre el elemento u objeto que se seleccione. Esta es normalmente `list` o `show` según el tipo de elemento. Así, tiene el mismo resultado ejecutar el comando `ip link list` que simplemente `ip link`.

A continuación se ofrecen varios ejemplos de uso del comando **ip** con el objetivo de enumerar interfaces de red y sus configuraciones:

```
$ ip link
# Se obtiene la lista de interfaces de red
$ ip link show enp0s3
# Detalles sobre la interfaz de red enp0s3
$ ip a
# Se obtiene la lista de interfaces con las direcciones que tienen asignadas
$ ip a show enp0s3
# Datos sobre la configuración de la interfaz indicada
```

Cambio de la configuración de una interfaz de red

Los comandos **set**, **add** y **del** de la utilidad **ip**, entre otros, representan acciones que implican modificar la configuración de un objeto, por lo que **requieren privilegios de superusuario** y, en consecuencia, se preceden del comando **sudo**.

En los ejemplos siguientes se muestra cómo activar/desactivar una interfaz de red, cómo agregar una configuración de dirección IP o cómo eliminarla y, por último, cómo modificar el tamaño de la MTU:

```
# Desactivar la interfaz de red y eliminarla de la tabla de enrutamiento
$ sudo ip link set enp0s8 down
# Configurar la dirección de una interfaz de red
$ sudo ip addr add 192.168.16.12/24 dev enp0s8
# Activar una interfaz de red que se ha configurado antes
$ sudo ip link set enp0s8 up
# Eliminar configuración de una interfaz de red
$ sudo ip addr del 192.168.16.15/24 dev enp0s8
# Establecer el tamaño máximo del datagrama IP (MTU)
$ sudo ip link set dev enp0s8 mtu 400
```

Actividad de las interfaces de red

Además de la configuración asignada a cada interfaz, también es posible conocer la actividad que han tenido hasta el momento. Con este fin se usa la opción **-s** de forma conjunta con el elemento **link**. Por ejemplo:

```
$ ip -s link
# Enumera las interfaz existentes y, por cada una de ellas, sus estadísticas
$ ip -s link show enp0s3
# Muestra estadísticas de la interfaz de red enp0s3
```

Las estadísticas de actividad se componen de dos apartados: **RX** (recibido) y **TX** (enviado), detallándose en cada uno de ellos el número de bytes, paquetes, errores, descartes, etc.

EJERCICIO 2

Usa el comando `ip` para obtener la información necesaria para responder a las siguientes cuestiones:

1. ¿Cuántas interfaces de red hay en tu sistema? ¿Cuál es el nombre de cada una de ellas?
2. Por cada interfaz de red indica su dirección MAC, y sus direcciones IPv4 e IPv6 si la tienen asignada.
3. Por cada interfaz de red indica el número de paquetes y bytes tanto enviados como recibidos.
4. ¿Hay alguna interfaz de red que no tenga asignada dirección IPv4 ni muestre actividad? ¿A qué crees que se debe?

Nota

La utilidad `ip` forma parte de un paquete de herramientas para administración y supervisión de redes llamada `iproute2`. Este viene instalado por defecto en las distribuciones modernas de GNU/Linux, en detrimento del paquete de herramientas clásico al que pertenecen utilidades como `ifconfig`, `arp`, `route` o `netstat`. Puedes instalar dicho paquete con el comando `sudo apt install net-tools` y usar la orden `man` para obtener información sobre su uso.

Administrar la configuración TCP/IP - Utilidad `netplan`

Aunque el comando `ip` permite establecer la configuración IP de las interfaces de red, así como habilitarlas y deshabilitarlas cuando sea preciso, estas configuraciones son temporales ya que **se restablecerán** ante ciertos eventos, entre ellos el reinicio del sistema. Esta es la razón de la existencia de métodos alternativos y persistentes, basados en el uso de **archivos de configuración** que almacenan la configuración a aplicar siempre que sea preciso restablecerla.

En GNU/Linux existen dos servicios o *demonios* diseñados para administrar la configuración de red. Por una parte tenemos a `systemd-networkd`, cuyo uso es más habitual en servidores ya que se guía por una configuración estática alojada en archivos de configuración ubicados en `/etc/systemd/networkd`. Por otro lado se encuentra el servicio `NetworkManager`, más habitual en configuraciones cliente cuya conectividad a red cambia dinámicamente según la disponibilidad de redes.

Ubuntu nos ofrece un mecanismo de **más alto nivel**, a través del comando `netplan`, que se basa en definir la configuración de las interfaces mediante un archivo YAML (*Yet Another Markup Language*). La utilidad `netplan`, con las opciones adecuadas, se encarga de configurar `systemd-networkd` o `NetworkManager` según sea preciso.

Archivos de configuración `netplan`

Los archivos de configuración de `netplan` se alojan en el directorio `/etc/netplan` y tienen extensión `.yaml`. Es habitual que exista uno inicial, pero pueden añadirse tantos como se precise. El comando `netplan` los procesará **en orden según el nombre** que tengan asignado. Así, un archivo de nombre `00-XXX.yaml` se procesará antes que otro con nombre `01-XXX.yaml`.

La estructura habitual de un archivo de este tipo es la que se muestra a continuación. Ha de tenerse presente que **el sangrado de las líneas es importante** al ser el formato YAML. En este, como ocurre con el lenguaje Python, no se emplean llaves para delimitar bloques, por lo que es el sangrado el que determina el ámbito de las instrucciones/parámetros. Este sangrado se crea **con espacios, nunca con tabuladores**.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      addresses:
        - 192.168.36.2/24
    enp0s8:
      dhcp4: true
```

El parámetro `version` existirá siempre con el valor `2`. La clave `renderer` indica a `netplan` para qué servicio de red se generará la configuración, de los dos antes citados. Estos se indican con los valores `networkd` y `NetworkManager`. En ausencia de este parámetro se emitirá la configuración para el servicio que haya instalado en el sistema.

En la rama `ethernets` podrán existir tantas subramas como interfaces de red tenga el sistema, conteniendo cada una de ellas los parámetros para configurar dicha interfaz. En el anterior ejemplo se asigna una dirección IP estática a la interfaz `enp0s3`, indicada con la clave `addresses`, y una IP dinámica obtenida por DHCP a la interfaz `enp0s8`.

Nota

Utiliza `man netplan` para conocer las distintas claves y valores que podrían usarse en este archivo de configuración. En [Netplan configuration examples](#) se ofrecen múltiples ejemplos de archivos de configuración Netplan

Aplicar una configuración con `netplan`

Tras preparar el archivo de configuración y alojarlo en el directorio `/etc/netplan`, el siguiente paso será aplicar dicha configuración a las interfaces de red. Con este fin usaremos uno de dos comandos posibles:

- `sudo netplan try`: aplicará la configuración y esperará hasta dos minutos a que confirmemos que la queremos mantener. Durante ese tiempo es posible, desde otra terminal, examinar la configuración. Si no se pulsa `Intro` antes de ese tiempo (véase la Figura 5) la **configuración se restablecerá** a su estado anterior.

```

usuario@par:~$
usuario@par:~$ sudo netplan try
Warning: Stopping systemd-networkd.service, but it can still be activated by:
  systemd-networkd.socket
Do you want to keep these settings?

Press ENTER before the timeout to accept the new configuration

Changes will revert in 120 seconds
Configuration accepted.
usuario@par:~$
usuario@par:~$ _
    
```

Figura 5. La configuración se aplica de forma temporal, hasta que se confirme o transcurran los dos minutos.

- **sudo netplan apply**: aplica la configuración sin preservar la anterior ni ofrecer su establecimiento. Es el método más habitual de aplicar una configuración.

Nota

La ejecución de estos comandos únicamente es precisa mientras está en funcionamiento el sistema, para cambiar la configuración, pero no tras cada reinicio, momento durante el cual se leerá y aplicará de forma automática el contenido de los archivos alojados en **/etc/netplan**.

Otros parámetros de configuración con **netplan**

Más allá de indicar si una interfaz de red obtendrá su IP desde un servidor DHCP o se le asignará una IP fija, existen otros parámetros que permiten fijar otros aspectos de su funcionamiento:

- **nameservers**: establece las direcciones IP de los servidores de nombres a los que se consultará para resolver nombres en determinados dominios. A continuación se muestra un ejemplo de uso de este parámetro:

```

network:
  version: 2
  ethernets:
    enp0s3:
      addresses:
        - 192.168.36.2/24
      nameservers:
        addresses:
          - 150.214.170.139
    
```

- **routes**: define las rutas a incluir en la tabla de enrutamiento para una interfaz de red dada. En el siguiente ejemplo se usa este parámetro para definir la IP **192.168.36.1** como el *router* de salida por defecto:

```
network:
  version: 2
  ethernet:
    enp0s3:
      addresses:
        - 192.168.36.2/24
      routes:
        - to: 0.0.0.0/0
          via: 192.168.36.1
```

Nota

Hay parámetros de **netplan** obsoletos cuyo uso está desaconsejado en la actualidad. Entre ellos se encuentran **gateway4** y **gateway6**, cuya finalidad es fijar el *router* de salida por defecto para IPv4 e IPv6, respectivamente. En lugar de dichos parámetros ha de emplearse **routes** según el anterior ejemplo.

EJERCICIO 3

Mediante las herramientas que se han descrito en esta sección vamos a establecer una configuración de dirección IP para el adaptador de red secundario de cada *host* de nuestra LAN, aquel que no tiene asignada una IP y al que **previamente hemos conectado un cable** que va al conmutador Ethernet:

1. Cada grupo tendrá asignado un identificador de red del tipo **10.0.X.0/24**, siendo el valor **X** establecido **al inicio de la sesión de prácticas** con un valor diferente para cada grupo.
2. Editar el archivo de configuración **netplan** de cada equipo que se unirá a la red, asignándole una **dirección IP estática apropiada** asumiendo que la **10.0.X.1** estará reservada para el hipotético *router* que conectaría la red al resto del mundo.
3. **Agregar a la configuración** de la interfaz información sobre el *router* por defecto y el servidor de nombres.
4. Indica **cómo queda la entrada** correspondiente a la interfaz de red en el archivo de configuración.
5. **Aplicar la configuración** establecida en el archivo de configuración en cada equipo. ¿Qué orden has empleado? ¿Ha generado alguna salida por consola?
6. Comprobar que la interfaz de red tiene **asignada la dirección IP**. ¿Qué orden has empleado? ¿Cuál es la dirección de *broadcast* asignada a la interfaz?

Otros aspectos de la configuración de red

En el apartado *Otros conceptos sobre configuración TCP/IP* se introducían varios conceptos relacionados con la configuración TCP/IP, más allá de la propia dirección IP. Además de contar con una dirección IP

asignada, para que la comunicación a través de una interfaz de red sea posible lo normal es que haya que ajustar otros aspectos de la configuración de red. Por ejemplo:

- **Máscara de red:** ya sabemos que la dirección IP se divide en dos partes, el identificador de red y el de *host*, sirviendo la máscara para determinar la longitud de cada una de ellas. Al asignar una dirección IP estática con `netplan` hay que indicar la máscara, algo que hacemos implícitamente al emplear la notación CIDR: `10.0.X.0/24`, cuyo último número indica los **bits que estarían a 1** en la máscara siempre contando de izquierda a derecha.
- **Router por defecto:** si queremos que nuestra LAN se comunique con otras redes, ya sean otras LAN cercanas o a través de Internet, es preciso contar con un *router* "de salida" al que enviar los datagramas IP. La dirección IP de ese equipo pasará a formar parte de la tabla de enrutamiento.
- **Servidores de nombres:** contar con la dirección de uno o más servidores DNS es esencial a la hora de comunicarse con el exterior, ya que las personas solemos conocer los nombres de los servidores, no directamente sus direcciones IP.

En la sección anterior hemos aprendido a fijar algunas de estas configuraciones, por ejemplo con los parámetros `nameservers` y `routes` de `netplan`. En esta sección profundizamos en algunos de estos aspectos y conocemos también otras vías para fijar dichas configuraciones.

Nota

Cuando se solicita una dirección IP a un servidor DHCP, habitualmente también se obtienen otros parámetros como la máscara de red, servidores de nombres y *router* por defecto.

El nombre del *host*

Cada ordenador tiene asignado un nombre que le identifica. Es aconsejable que dicho nombre sea único en una LAN, de forma que no haya conflictos. Asimismo, el nombre de *host* permite saber en qué máquina se está trabajando cuando se realizan conexiones remotas (por ejemplo con SSH).

El comando `hostname` permite tanto obtener el nombre actual del *host* como cambiarlo:

```
$ hostname
par
$ sudo hostname servidor
$ hostname
servidor
```

El nombre de *host* se restablece en cada inicio del sistema, tomándolo del archivo de configuración `/etc/hostname`. Por tanto, para cambiar de manera persistente dicho identificador es preciso editar el mencionado archivo.

Nota

El comando `hostnamectl` es una alternativa para obtener y configurar el nombre de *host*. Este comando facilita información adicional sobre el ordenador, como su arquitectura de procesador, versión del núcleo de sistema, etc.

Traducción de nombres a direcciones IP

Ya sabemos que, en general, nos resulta más fácil recordar nombres que secuencias de números. Esa es la razón de que a los *host* de una LAN se les asignen nombres, como se ha explicado en el apartado previo, algo que se aplica a todos los equipos conectados a Internet.

Para acceder por su nombre a los equipos de nuestra LAN, o cualquier *host* cuyo nombre no esté registrado oficialmente en un servidor DNS, editaremos el archivo `/etc/hosts`. Este almacena por cada línea una dirección IP, ya sea IPv4 o IPv6, seguida de uno o más nombres. Esto facilita una resolución o traducción local: cualquier programa que use esos nombres obtendrá la IP correspondiente sin necesidad de realizar consultas externas.

```
# Dirección IP      Nombre      Alias
127.0.0.1          localhost   MiPortatil
150.214.178.18     p18.ujaen.es
```

Lógicamente no es posible introducir en `/etc/hosts` los nombres de todos los *host* conectados a Internet. Para obtener las direcciones IP de esos equipos se recurre a un servidor DNS. Esta es la razón de que se precise la IP de uno o más servidores DNS a los que poder realizar consultas para la traducción de nombres. Con este fin se emplea el archivo de configuración `/etc/resolv.conf`:

```
$ sudo nano /etc/resolv.conf # Editar el archivo
search ujaen.es             # Dominio de búsqueda
nameserver 150.214.170.15    # Servidor de nombres
```

La tabla de enrutamiento del *host*

Cada equipo conectado en una red, tanto los *host* como los *router*, cuentan con una **tabla de enrutamiento**. Esta contendrá las entradas necesarias para que el equipo sepa la interfaz por la que debe enviar un paquete.

El contenido de la tabla de enrutamiento en un *host* se completa de manera automática, aunque es posible introducir cambios por vía manual, a través de comandos (con la orden `ip route` según se explicó en un apartado previo) y archivos de configuración (según se ha indicado antes para `netplan` con el apartado `routes`).

Ya sabemos que el comando `ip route` nos facilita la tabla de enrutamiento que, como en el ejemplo siguiente, suele constar de múltiples entradas:

```
$ ip route
default via 10.0.2.2 dev enp0s3 proto dhcp metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
10.0.2.2 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 100
10.10.10.0/24 dev enp0s8 proto kernel scope link src 10.10.10.10 metric 101
```

En esta salida se distinguen dos tipos de entradas diferentes:

- `[destino] via [router] dev [interfaz] [proto_inicio] [métrica]`: indica que para llegar al `[destino]` el paquete se ha de enviar al `[router]` que está conectado a la interfaz `[interfaz]`. En la primera línea del ejemplo previo se indica que para llegar a cualquier red aparte de las enumeradas en la tabla, hay que usar como *router* de salida el que tiene IP `10.0.2.2`, accesible a través de la interfaz de red `enp0s3`.
- `[destino] dev [interfaz] proto kernel scope link src [dir_ip] [métrica]`: indica que para llegar al `[destino]` utilizaremos la interfaz `[interfaz]` con dirección `[dir_ip]` como IP de origen. En el anterior ejemplo hay varias entradas con este formato. La segunda y cuarta hacen referencia a redes accesibles a través de las interfaz `enp0s3` y `enp0s8`. La tercera determina cómo alcanzar el *router* con IP `10.0.2.2`.

Nota

El indicador `default` en una tabla de rutas funciona como un *comodín*: actúa como destino por defecto para todos aquellos paquetes de salida que no coinciden con ninguna entrada de la tabla de enrutamiento.

La caché ARP del *host*

En el apartado [Direcciones MAC/Ethernet](#) se explicó cómo las direcciones MAC, usadas al nivel de la capa de enlace, son las que en último término hacen posible la transmisión de tramas entre *host*, *switches* y *routers*. Ya sabemos que el protocolo ARP (*Address Resolution Protocol*) es el encargado de, dada una dirección IP, obtener la MAC que corresponda. Esa información se almacena temporalmente en la denominada caché ARP.

Esa tabla se consulta con el comando `ip neighbour` y su mantenimiento es automático aunque, como en el caso de la tabla de enrutamiento, se permite el uso de las acciones `add` y `delete` para agregar y eliminar entradas.

Comprobación de la configuración de red

Establecida la configuración de una red, y a pesar de las interfaces tengan asignadas sus direcciones IP, no estaremos seguros de que existe comunicación entre los *host* de la misma hasta que efectuemos algún tipo de comprobación. Con este fin recurriremos a dos comandos distintos: `ping` y `traceroute`.

La utilidad `ping`

Este comando usa el protocolo ICMP para enviar una solicitud de eco a un equipo remoto (*host* o *router*). Si dicho equipo devuelve la respuesta esperada es que existe conexión. La sintaxis de uso de este comando es la siguiente:

```
ping [opciones] host
```

La tabla siguiente enumera las opciones de uso más común y su finalidad.

Opción [parámetro]	Descripción
-c cantidad	Establece el número de paquetes a enviar
-i segundos	Espera el número de segundos indicados entre dos envíos consecutivos. Por defecto es 1
-I interfaz	Especifica la interfaz de red por la que se enviarán los paquetes
-l paquetes	Se envía el número de paquetes especificado tan rápido como se pueda
-n	No se resuelven nombres a direcciones IP
-p	Se envía un patrón de datos determinado que el otro equipo debe devolver. Se usa para pruebas
-R	Indica que se guarde la ruta seguida por el paquete. Muchos host ignoran esta opción
-s tamaño	Número de bytes de los paquetes a enviar
-t ttl	Tiempo de vida máximo del paquete

La utilidad **traceroute**

El programa **traceroute** es capaz de usar distintos protocolos (ICMP, UDP y TCP) con el objetivo de detectar todos los elementos de interconexión entre un **host** de origen y otro de destino, lo que genera la ruta que ha seguido un datagrama IP. La sintaxis de uso es la siguiente:

```
traceroute [opciones] host
```

La tabla siguiente enumera las opciones de uso más común y su finalidad.

Opción [parámetro]	Descripción
-i interfaz	Especifica la interfaz de red de salida
-I	Utilizar mensajes ICMP de solicitud de eco en lugar de UDP
-p puerto	Puerto a utilizar. Su significado varía según se use ICMP o UDP
-f ttl_ini	TTL inicial. Por defecto es 1
-m max_ttl	TTL máximo o longitud máxima de la ruta
-n	No resolver nombres a direcciones IP
-w segundos	Establece el número de segundos que se espera a una respuesta. Por defecto es 5

Nota

Para instalar esta utilidad ejecuta en tu sistema el comando **sudo apt install traceroute**.

EJERCICIO 4

Una vez configurada la LAN (según las indicaciones dadas en el Ejercicio 3) verifica su funcionamiento y comprueba asimismo que la conexión con el exterior, ya existente a través de la otra interfaz de red, no se ha visto afectada. Para ello:

1. Comprueba que los *host* que forman la LAN pueden enviarse eco entre ellos. ¿Qué orden has utilizado y con qué parámetros? Describe el resultado que has obtenido como salida.
2. Usa la misma orden anterior para verificar que puedes acceder a fcharte.com. ¿Qué diferencias hay respecto al resultado obtenido entre *host* de la LAN?
3. Determina qué elementos de interconexión existen entre tu ordenador y el *host* fcharte.com. ¿Qué orden has usado y con qué parámetros? Enumera las direcciones IP intermedias e intenta deducir a qué equipo corresponde cada una de ellas.
4. Usa la orden anterior con la opción **-I** primero y con la opción **-T** después. Necesitarás hacerlo con **sudo** porque se precisan privilegios de superusuario. ¿En qué cambian estas opciones el funcionamiento de la orden? ¿Qué diferencias hay en la salida obtenida?