Capítulo 10

Configuración IP

Al terminar este capítulo, entenderás:

- Cuál es la información esencial para que un nodo pueda comunicarse en una red IP.
- Cómo se configura un nodo de forma manual.
- Cómo se configura un nodo de forma automática con DHCP.

Cuando un nodo arranca y el SO accede a las interfaces de red conectadas, puede determinar sus direcciones MAC[1], porque esas direcciones físicas están grabadas en una memoria ROM en el propio periférico.

Sin embargo, eso no ocurre con la dirección IP. La dirección IP es parte de la configuración del nodo, y como hemos visto, es totalmente dependiente de la red a la cada NIC esté conectada. Sin una dirección IP válida, el nodo no podrá comunicarse con otros nodos ni en la misma red ni fuera de ella.

10.1. Configuración manual

Si no hay disponible un servicio que ayude a realizar esta configuración automáticamente (que veremos más tarde), el usuario del nodo debe realizar la configuración de forma manual. En ese caso, el administrador de la red tiene que proporcionar al usuario la dirección IP para ese nodo y la máscara de subred. Probar asignando una dirección a ciegas es poco probable que funcione, y puede provocar problemas de conectividad a otros usuarios si se elige una dirección en uso.

Supón que el administrador de la red te ha asignado la dirección `20.3.4.13/24` para la interfaz Ethernet de tu computador. Lo primero que necesitas es saber cómo se llama la interfaz, algo que puedes averiguar con el comando `ip addr` (Figura 10.1).

[1] Este es el caso de Ethernet o WiFi, pero no todas las NIC tienen dirección MAC, solo las que utilizan una tecnología de medio compartido.

```
$ sudo ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue 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: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
    link/ether f8:5f:2a:c0:ff:ee brd ff:ff:ff:ff:ff:ff
```

Figura 10.1: Salida del comando `ip addr` para mostrar las interfaces de red.

Aquí puedes ver la interfaz loopback (`lo`) y la interfaz Ethernet (`link/ether`), que en este caso se llama `eno1`. Ahora puedes asignarle la nueva dirección y ver el resultado:

```
$ sudo ip addr add 20.3.4.13/24 dev eno1
$ sudo ip addr show eno1
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
    link/ether f8:5f:2a:c0:ff:ee brd ff:ff:ff:ff:ff:ff
    inet 20.3.4.13/24 scope global eno1
       valid_lft forever preferred_lft forever
```

Al hacer esto, el SO crea automáticamente una entrada en la tabla de encaminamiento que indica cómo llegar a los vecinos:

```
$ ip route
20.3.4.0/24 dev eno1 proto kernel scope link src 20.3.4.13
```

En esta situación el nodo tiene conectividad con cualquier vecino de la red local, aunque no puede llegar más allá. Como ya hemos visto, para salir de la red local, se necesita la IP del router local (también llamado «puerta de enlace»). Este dato también lo debe proporcionar el administrador de la red. Supongamos para este ejemplo que esa dirección es `20.3.4.1`. Puedes crear la entrada en la tabla de encaminamiento con el siguiente comando:

```
$ sudo ip route add default via 20.3.4.1
$ sudo ip route
default via 20.3.4.1 dev eno1
20.3.4.0/24 dev eno1 proto kernel scope link src 20.3.4.13
```

Esto corresponde con la tabla de encaminamiento básica que vimos en §8.3.2. La tabla dice cómo llegar a los vecinos y cómo llegar a cualquier otro nodo fuera de la red. Por eso se define como router por defecto.

Si no haces nada más, esta información se perderá cuando el nodo se reinicie. Para que sea persistente, tienes que escribirla en el archivo de configuración `/etc/network/interfaces`, que quedaría así:

```
auto eno1
iface eno1 inet static
    address 20.3.4.13
    netmask 255.255.255.0
    gateway 20.3.4.1
```

Este es el método clásico den Debian, pero existen otros métodos como `systemd-networkd`, `NetworkManager` o `netplan`, que aunque más versátiles, para un uso tan básico como este, resultan equivalentes.

Otra cuestión importante es el nombre del nodo. Se utiliza para identificar el nodo en la red y puede ser necesario para configurar ciertos servicios. El nombre del nodo se puede consultar y cambiar con el comando `hostname`:

```
$ hostname
maine
$ hostname atlantis
$ hostname
atlantis
```

Pero también es temporal. Si quieres que se mantenga después de reiniciar debes escribir ese nombre en el archivo `/etc/hostname`.

Este tipo de configuración se suele denominar «manual» o «estática» porque el usuario la tiene que definir explícitamente y no cambia a menos que lo haga el usuario. Cualquier SO (incluidos los móviles) tiene algún medio para configurar esta información de forma manual, ya sea con una shell o con una interfaz gráfica.

10.2. Configuración automática con DHCP

Pedir a un usuario sin conocimientos técnicos básicos que realice una configuración manual no es lo más conveniente. Además cada nuevo usuario para cada nuevo nodo u otro equipo como una impresora, debe solicitar la configuración al administrador de la red. El administrador a su vez debe llevar registro de las direcciones asignadas por cada subred, y cerciorarse de que se siguen usando, pues lo más probable es que un usuario que se va olvide notificar que ya no necesita esas direcciones.

Para resolver todos estos inconvenientes existe el protocolo DHCP (Dynamic Host Configuration Protocol) [22]. DHCP es un servicio que proporciona la configuración completa para un nodo, cuando éste lo solicita. DHCP se suele considerar un protocolo de aplicación que se encapsula sobre UDP y utiliza los puertos reservados 67 y 68 para servidor y cliente respectivamente.

Pero, ¿cómo puede un nodo sin dirección solicitar algo a un servidor? ¿Cómo sabe el nodo la dirección del servidor? Es posible porque el cliente

DHCP envía mensajes de difusión (dirigidos a `255.255.255.255`), algo que es posible incluso aunque la interfaz no tenga una dirección asignada.

En realidad, la conversación entre cliente y servidor DHCP es algo más compleja. Veamos el escenario típico con más detalle.

- El cliente, en el nodo que necesita configuración, envía un mensaje Discover a la dirección de difusión `255.255.255.255`. El propósito de este mensaje es «descubrir» servidores DHCP en la red.
- El servidor (o servidores) DHCP recibe esos mensajes y en función de su configuración responde con un mensaje Offer que contiene la dirección IP y otros parámetros de configuración. Este mensaje también va dirigido a la dirección de difusión.
- El cliente recibe estas «ofertas» y envía un mensaje Request para confirmar la que ha elegido.
- El servidor envía un mensaje Ack como confirmación. Este mensaje contiene la información de configuración completa, incluyendo la dirección IP, la máscara de subred, la puerta de enlace y también los servidores DNS. Desde este momento, el servidor considera que la dirección IP está asignada a ese cliente y no la ofrecerá a otros.
- Al recibir el mensaje Ack, el cliente aplica la configuración a la interfaz de red, y a partir de ese momento puede comunicarse con normalidad.
- Cuando el nodo se apaga o ya no necesita la dirección, envía un mensaje Release para liberarla. Así el servidor puede disponer de ella y asignarla a otro nodo.

Puedes hacer una captura de una conversación DHCP real. En este ejemplo se utiliza `eno1`, que es una interfaz Ethernet, pero lo puedes probar igualmente con una interfaz WiFi. En un terminal arranca una captura con `tshark -i eno1 -Y bootp`. El argumento `-Y bootp` es un filtro que le dice a `tshark` que te interesan solo los mensajes BOOTP, que incluye los mensajes DHCP, porque el segundo es una extensión del primero.

En otro terminal, ejecuta el cliente tal como aparece a continuación. Lo más probable es que ya tengas el programa `dhclient`, pero si no es así, instala el paquete `isc-dhcp-client`.

El siguiente comando libera la dirección que tienes asignada (envía el mensaje Release) para que así puedas capturar el proceso completo. Opcionalmente después de ese comando, puedes ver que ahora la interfaz no tiene dirección si ejecutas `ip a`.

```
$ sudo dhclient -v -r eno1
Killed old client process
Internet Systems Consortium DHCP Client 4.4.3-P1

Listening on LPF/eno1/f8:5f:2a:c0:ff:ee
Sending on   LPF/eno1/f8:5f:2a:c0:ff:ee
DHCPRELEASE of 172.24.27.53 on eno1 to 172.20.32.167 port 67
```

Este segundo comando es el que realiza la solicitud. Puedes ver en la propia salida del comando un resumen de los mensajes de los que hemos hablado.

```
$ sudo dhclient -v -i eno1
Internet Systems Consortium DHCP Client 4.4.3-P1

Listening on LPF/eno1/f8:5f:2a:c0:ff:ee
Sending on   LPF/eno1/f8:5f:2a:c0:ff:ee
DHCPDISCOVER on eno1 to 255.255.255.255 port 67 interval 5
DHCPDISCOVER on eno1 to 255.255.255.255 port 67 interval 10
DHCPOFFER of 172.24.27.53 from 172.20.32.167
DHCPREQUEST for 172.24.27.53 on eno1 to 255.255.255.255 port 67
DHCPACK of 172.24.27.53 from 172.20.32.167
bound to 172.24.27.53 -- renewal in 1571 seconds.
```

En la captura que tenías en macha en el otro terminal ha debido aparecer algo similar a esto:

```
$ sudo tshark -i eno1 -Y bootp
1        0.0.0.0 → 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0x9b6d7a5a
2  172.20.32.167 → 172.24.27.53    DHCP 342 DHCP Offer    - Transaction ID 0x9b6d7a5a
3        0.0.0.0 → 255.255.255.255 DHCP 342 DHCP Request  - Transaction ID 0x9b6d7a5a
4  172.20.32.167 → 172.24.27.53    DHCP 342 DHCP ACK      - Transaction ID 0x9b6d7a5a
```

Para ver más detalles, repite la captura añadiendo el argumento `-V` al comando `tshark`. Se muestran aquí los campos más interesantes de los mensajes Discover y Offer.

```
$ sudo tshark -i eno1 -Y bootp -v
Dynamic Host Configuration Protocol (Discover)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Transaction ID: 0x014ff92a
    Client IP address: 0.0.0.0
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: f8:5f:2a:c0:ff:ee
    Option: (50) Requested IP Address (0.0.0.0)
    Option: (12) Host Name
        Length: 5
        Host Name: maine
    Option: (55) Parameter Request List
        Length: 13
        Parameter Request List Item: (1) Subnet Mask
```

```
        Parameter Request List Item: (28) Broadcast Address
        Parameter Request List Item: (2) Time Offset
        Parameter Request List Item: (3) Router
        Parameter Request List Item: (15) Domain Name
        Parameter Request List Item: (6) Domain Name Server
        Parameter Request List Item: (119) Domain Search
        Parameter Request List Item: (12) Host Name
        Parameter Request List Item: (26) Interface MTU
        Parameter Request List Item: (121) Classless Static Route
        Parameter Request List Item: (42) Network Time Protocol Servers
    Option: (61) Client identifier
        Length: 19
        IAID: 96a502be
        DUID Type: link-layer address plus time (1)
        Hardware type: Ethernet (1)
        Time: 799853663
        Link layer address: f8:5f:2a:c0:ff:ee

Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Transaction ID: 0x014ff92a
    Client IP address: 0.0.0.0
    Your (client) IP address: 172.24.27.53
    Next server IP address: 172.20.32.167
    Client MAC address: f8:5f:2a:c0:ff:ee
    Option: (1) Subnet Mask (255.255.255.0)
    Option: (58) Renewal Time Value: 30 minutes (1800)
    Option: (59) Rebinding Time Value: 52 minutes, 30 seconds (3150)
    Option: (51) IP Address Lease Time: 1 hour (3600)
    Option: (54) DHCP Server Identifier (172.20.32.167)
    Option: (3) Router: 172.24.27.1
    Option: (15) Domain Name: uclm.es
    Option: (6) Domain Name Server:
        Domain Name Server: 172.20.32.3
        Domain Name Server: 172.20.32.4
```

En el mensaje Discover el cliente solicita diversa información. Lo puedes ver en la sección *Parameter Request List*. El servidor responde a estas solicitudes como opciones al final del mensaje Offer y Ack. No tiene por qué responder a todo, pero a parte por supuesto de la dirección del nodo (campo «Your (client) IP address»), normalmente siempre aparecerán los ya nombrados:

- Subnet Mask
- Router
- Domain Name Server

El servidor asigna una dirección al cliente por un período limitado: el «tiempo de concesión» (*lease time*), que es configurable en el servidor. En la captura esto aparece en el campo `Rebinding Time Value`. Durante ese plazo, el servidor asegura que la dirección no se va a asignar a otro cliente. Si el cliente quiere seguir utilizando la misma dirección puede enviar un

mensaje Renew para renovar la concesión antes de que termine el plazo de renovación (campo `Renewal Time Value` en la captura). En la captura puedes comprobar que el tiempo de concesión es de 52 min 30 seg y el de renovación es de 30 min. Estos valores son bastante bajos porque hemos tomado la captura en un entorno empresarial. En el entorno doméstico el tiempo de concesión suele ser de 24 horas o incluso más. Comprueba qué valores aparecen en tu caso analizando tu captura.

10.3. Servidor DHCP

El programa `dnsmasq` es un servidor DHCP sencillo, ligero y fácil de configurar, ideal para redes domésticas, despliegues pequeños o para realizar pruebas de laboratorio. También funciona como servidor DNS aunque para este ejemplo lo vamos a desactivar.

El servidor se configura mediante un archivo de texto que suele estar en `/etc/dnsmasq.conf`. El siguiente listado es un ejemplo perfectamente funcional de ese fichero que sirve para asignar direcciones en la red `192.168.2.0/24` incluyendo la información básica necesaria.

```
port=0
listen-address=192.168.2.1
dhcp-range=192.168.2.2,192.168.2.100,12h
dhcp-option=3,192.168.2.1
dhcp-option=6,8.8.8.8,1.1.1.1
```

Veámos en detalle su sintaxis y significado:

`port` es el puerto en el que debe escuchar el servidor DNS. Al asignar un valor de 0, el servidor DNS queda inoperante.

`listen-address` indica la dirección de la interfaz de red en la que el servidor DHCP atenderá peticiones.

`dhcp-range` es el rango de direcciones que puede asignar a los clientes. Desde `192.168.2.2` hasta `192.168.2.100`, con una concesión de 12 horas. El nodo que ejecuta este servidor DHCP probablemente actúa de router de la LAN (aunque no es obligatorio) y tendría la dirección `192.168.2.1`.

`dhcp-option` indica distintas opciones con información adicional que se puede ofrecer a los clientes. Los números de opción son los de la especificación de DHCP y corresponden con la captura que has visto en la sección anterior: 3 para indicar la dirección del router y 6 para las direcciones IP de dos servidores DNS, que en este caso son servidores públicos de Google y Cloudflare.

Puedes verlo en funcionamiento aplicando este mismo fichero de configuración con el ejemplo /dnsmasq utilizando un contenedor appdocker como puedes ver en la siguiente consola:

```
dnsmasq$ docker compose up
[+] Running 1/1
 - Container dnsmasq  Create
Attaching to dnsmasq
dnsmasq  | dnsmasq[1]: started, version 2.90 DNS disabled
dnsmasq  | dnsmasq-dhcp[1]: DHCP, IP range 192.168.2.2 -- 192.168.2.100, lease time 12h
```

En este ejemplo `docker` crea una red llamada `dnsmasq_net` en la que el host (tu PC) y el contenedor son vecinos. Eso significa que puedes solicitar una dirección a este DHCP. Debes listar las interfaces para averiguar cuál tiene una IP en el bloque `192.168.2.0/24`. Sí, la interfaz ya tiene una IP asignada, pero no importa, aún así puedes solicitar otra al servidor DHCP y comprobar su funcionamiento. La interfaz en cuestión tendrá un nombre similar a `br-f215a045cd10`. Por tanto, en otro terminal, puedes solicitar una dirección con:

```
$ ip a
[...]
54: br-f215a045cd10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 5a:47:ab:01:41:42 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.254/24 brd 192.168.2.255 scope global br-f215a045cd10
       valid_lft forever preferred_lft forever

$ sudo dhclient -v -i br-f215a045cd10
```

Al ejecutar esto deberías ver un registro detallado de las solicitudes y respuestas en la salida de `docker compose` con información similar a la captura `tshark` de la sección anterior. Después puede comprobar que la interfaz tiene ahora una dirección del rango (`192.168.2.85`).

```
$ ip addr show br-f215a045cd10
54: br-f215a045cd10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 5a:47:ab:01:41:42 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.254/24 brd 192.168.2.255 scope global br-f215a045cd10
       valid_lft forever preferred_lft forever
    inet 192.168.2.85/24 brd 192.168.2.255 scope global secondary dynamic br-f215a045cd10
       valid_lft 43196sec preferred_lft 43196sec
```

Y ¿qué más?

La dirección IP del nodo, la del router y las de los servidores DNS constituyen la configuración mínima esencial para que un nodo pueda conectarse a la red, sin embargo hay mucho más. Conforme una red y un despliegue TIC se vuelve más y más complejo es necesario administrar muchos otros

parámetros como inventarios de equipos, usuarios, programas instalados, versiones y un largo etcetera.

Esto da lugar a toda una disciplina que es la «Gestión y administración de redes y sistemas» en sus muchas variantes. En esta disciplina se utilizan protocolos y herramientas específicas. Y muy relacionado con todo esto quedan las tareas de monitorización y seguridad, que suponen otra gran área de conocimiento y estudio. Este capítulo solo ha sido el primer paso hacia ese mundo.