

## Capítulo 7

# Direccionamiento IP

Al terminar este capítulo, entenderás:

- Qué es una dirección IP y cuál es su formato.
- Para qué sirve la máscara de red.
- Qué es el direccionamiento con clases y sus limitaciones.
- Qué es el direccionamiento sin clases y la técnica VLSM
- Cómo calcular el direccionamiento para una topología dada.

## 7.1. Qué es una dirección IP y su formato

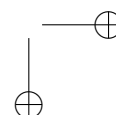
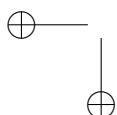
Una dirección IP es un identificador único asignado a cada dispositivo de la interred. Se trata de un número entero de 32 bits, por lo que teóricamente el espacio de direcciones es de  $2^{32}$  direcciones, o 4 «gibidirecciones»<sup>1</sup>, es decir, concretamente 4.294.967.296 direcciones. Aunque parezcan muchas (y ciertamente lo son) por varias razones que veremos, no todas las direcciones están disponibles para asignar a dispositivos, hasta el punto que desde hace unos años hay una apremiante escasez de direcciones IP.

A diferencia de las direcciones físicas, las direcciones IP tienen una estructura jerárquica. Como norma general podemos considerar dos partes<sup>2</sup>: la primera es el prefijo de red (*network ID*) e identifica la red a la que pertenece el dispositivo. La segunda es el identificador de nodo (*host ID*) y sirve para eso, indica qué nodo es dentro de esa red. La Figura 7.1 muestra el formato de la dirección IP.

El prefijo de red es imprescindible para que los routers puedan hacer su trabajo. Esencialmente la función de un router es decidir por cual de sus interfaces debe enviar cada paquete que recibe. Para eso, dispone de una tabla (la tabla de encaminamiento) que recorre buscando cuál de sus filas

<sup>1</sup>Nos tomamos la licencia de usar el prefijo «gibi» del SI que indica  $2^{30}$ , como en las unidades de memoria.

<sup>2</sup>Mas adelante veremos que puede tener más.



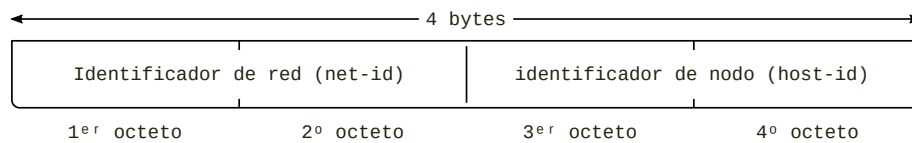


FIGURA 7.1: Formato de la dirección IP

encaja con la dirección IP destino que aparece en el paquete. El prefijo de red permite que la tabla pueda incluir las direcciones de las redes conocidas, en lugar de tener que incluir todos los posibles nodos destino. De ese modo las tablas pueden tener unas decenas de filas en lugar de millones.

## 7.2. Máscara de red

Para determinar qué parte de la dirección corresponde al prefijo de red y cuál al identificador de nodo se necesita la **máscara de red**. Es simplemente otro número de 32 bits que, en binario, se ve como una secuencia de unos seguida de una secuencia de ceros, aunque normalmente se representa en notación decimal, como una dirección IP. Así, una máscara formada por 16 unos seguidos de 16 ceros, en decimal sería 255.255.0.0.

Para obtener el prefijo de red se realiza una operación **AND** a nivel de bits entre la dirección y la máscara. En el ejemplo, dada la dirección 150.20.45.76, y la máscara 255.255.0.0, el resultado que se obtiene al hacer el **AND** es 150.20.0.0. El prefijo de red es 150.20 (la parte que ha «superado» la máscara), y el identificador de dispositivo es 45.76 (la parte eliminada por la máscara) (ver Cuadro 7.1). La dirección 150.20.0.0 (la que tiene todos ceros en la parte de identificador de nodo) representa a la red completa y por ello se conoce como **dirección de red**.

Relacionado con esto, también tenemos la **dirección broadcast**. Es la dirección que representa a todos los dispositivos de la red, y se obtiene también a partir del prefijo de red, pero con todos los bits restantes a 1, en lugar de 0. En el ejemplo, la dirección de broadcast de la red 150.20.0.0 para la máscara aplicada es 150.20.255.255.

	decimal	1 <sup>er</sup> octeto	2 <sup>o</sup> octeto	3 <sup>er</sup> octeto	4 <sup>o</sup> octeto
Dirección	150.20.45.76	10010110	00010100	00101101	01001100
Máscara	255.255.0.0	11111111	11111111	00000000	00000000
Dir. de red	150.20.0.0	10010110	00010100	00000000	00000000
Dir. broadcast	150.20.255.255	10010110	00010100	11111111	11111111

CUADRO 7.1: Ejemplo de uso de la máscara de red

Como en total tienen que ser 32 bits, cuando necesitamos una red grande con muchos nodos, el prefijo será corto, mientras que si es una red pequeña, el prefijo será largo. Esto también implica que si creamos demasiadas redes grandes, quedará poco espacio para las pequeñas, y viceversa.

En todo caso, una organización cualquiera no puede decidir por su cuenta qué prefijo usar para su red. Para hacer un reparto organizado y evitar conflictos, existen organizaciones específicas como la ICANN o sus delegados regionales (las RIR) que se encargan de asignar —previo pago, por supuesto— estos prefijos de red a las organizaciones que los soliciten.

Pero, ¿cómo se sabe cuál es la máscara de red que hay que aplicar? Hay dos respuestas a esta pregunta, la que se definió originalmente (que llamamos «direccionamiento con clases») y la que se utiliza en la actualidad, que llamamos «direccionamiento sin clases». Veremos ambas a continuación.

### 7.3. Direccionamiento con clases

Para hacer un reparto racional entre redes grandes y pequeñas, la IETF originalmente definió un sistema que llamamos direccionamiento con clases o *classfull addressing* [9], que divide el espacio de direcciones en cinco clases nombradas desde la A a la E, aunque únicamente las clases A, B y C se utilizan para direccionamiento unicast.

clase	prefijo (bits)	máscara	# redes	direcciones por red
A	0	255.0.0.0	128	16 777 216
B	10	255.255.0.0	16 384	65 536
C	110	255.255.255.0	2 097 152	256

CUADRO 7.2: Direccionamiento con clases

Con este sistema, dada una dirección IP se puede determinar directamente a qué clase pertenece, pero también mucha otra información útil. Veamos un ejemplo, con la dirección 161.67.21.13:

- Es una dirección de clase B porque los dos primeros bits de 161 en binario son 10.
- Por ser de clase B, le corresponde una máscara 255.255.0.0.
- Su dirección de red es 161.67.0.0.
- Su dirección de broadcast es 161.67.255.255.
- Su rango de direcciones asignables va de 161.67.0.1 a 161.67.255.254, es decir  $2^{16} - 2$  direcciones.

Hablamos de «direcciones asignables» precisamente porque, ni la dirección de red ni la de broadcast, se pueden asignar a una interfaz de red, y por eso se deben descontar del total.

### 7.3.1. Subnetting

El direccionamiento con clases tiene varios problemas. Hay pocas redes de clase A y B, porque son demasiado grandes (sobre todo las clase A), y las de clase C son demasiado pequeñas, más a día de hoy. Esto provocaba un gran desperdicio de direcciones porque muchas organizaciones que solicitaron bloques clase A y B hace años solo utilizaban una mínima parte, pero nadie más las puede aprovechar.

Para tratar de resolver esto, se desarrolló el concepto de *subnetting* o más formalmente FLSM. Consiste en dividir una red cualquiera en subredes más pequeñas. Estas nuevas subredes son útiles para la misma organización que ya poseía la red original. Desde un punto de vista administrativo, permite a la organización delegar la gestión de cada subred al departamento o unidad a la que se le asigna. Eso facilita la gestión de los servicios, visibilidad hacia el exterior y privilegios de acceso específicos de cada subred, adaptándose a las necesidades, restricciones y políticas de cada departamento.

Para crear subredes, se toman bits de la parte del identificador de host y se utilizan para direccionar las subredes. A este nuevo campo se le llama «prefijo de subred» o *subnet ID* (consulta la Figura 7.2).

Identificador de red (net-id)	Identificador de subred (subnet-id)	identificador de nodo (host-id)
----------------------------------	--	------------------------------------

FIGURA 7.2: Formato de las dirección IP con *subnetting*

Si se toman  $n$  bits para el prefijo de subred, se obtendrán  $2^n$  subredes. Siempre que se aplica *subnetting* se obtiene una potencia de dos de subredes. Eso significa que si, por ejemplo, se necesitan 3 subredes, habrá que crear al menos 4 y una de ellas quedará sin uso. Obviamente, tomar bits del identificador de host para el de subred, implica que las subredes resultantes serán más pequeñas, es decir, habrá menos direcciones para asignar a nodos.

Lo interesante del *subnetting* es que se aplica solo desde el punto de vista de la organización. Desde fuera, para el resto de Internet, el conjunto se sigue viendo como la red original completa. Nadie fuera de esa red necesita saber que se ha aplicado *subnetting* y mucho menos cómo se ha hecho, en cuantas subredes se ha dividido, etc.

Inicialmente el primer y último bloque que se obtiene al realizar la división causaba problemas en el encaminamiento porque esas direcciones se podían confundir con la dirección de red y broadcast de la red original. Para evitarlo, se prohibía el uso de estos dos bloques, llamados *subnet-zero* y *subnet-all-ones*. Es una limitación muy grave, ya que en el caso de usar pocos bits para el prefijo de subred, se podían perder muchas direcciones. En el ejemplo anterior implicaría perder la mitad del espacio de direcciones original. Más tarde la RFC 1878 [12] especificó la forma de evitar esta limitación. A día de hoy la mayoría de equipos de red y sistemas operativos lo soportan sin problema.

### 7.3.2. Ejemplo de *subnetting*

Veamos un ejemplo sencillo. Partimos de una red con dirección 150.20.0.0, que es de clase B (máscara 255.255.0.0), es decir, tenemos 16 bits para direccionar nodos. Vamos a dividirla en 4 subredes. Utilizaremos los 2 primeros bits del tercer octeto como prefijo de subred y los 14 restantes como identificador de nodo. La tabla 7.3 muestra las subredes resultantes. Para cada subred se muestra el tercer octeto en binario, la dirección de subred y la dirección de broadcast. Para todas ellas, la máscara de subred será 255.255.192.0.

subred	tercer octeto	dir. de subred	dir. broadcast
0	0000 0000	150.20.0.0	150.20.63.255
1	0100 0000	150.20.64.0	150.20.127.255
2	1000 0000	150.20.128.0	150.20.191.255
3	1100 0000	150.20.192.0	150.20.255.255

CUADRO 7.3: Ejemplo de *subnetting*

Existen muchas herramientas para realizar el cálculo de subredes con *subnetting*. Una de las más sencillas es *sipcalc*. En el Listado 7.1 se muestra un ejemplo para la misma división que hemos visto en el ejemplo anterior.

```
$ sipcalc -bi 150.20.0.0/16 -s /18
-[ipv4 : 150.20.0.0/16] - 0

[CIDR]
Host address^^I^^I- 150.20.0.0
Host address (decimal)^I- 2517893120
Host address (hex)^I- 96140000
Network address^^I^^I- 150.20.0.0
Network mask^^I^^I- 255.255.0.0
Network mask (bits)^I- 16
Network mask (hex)^I- FFFF0000
Broadcast address^^I- 150.20.255.255
```

```

Cisco wildcard- 0.0.255.255
Addresses in network- 65536
Network range- 150.20.0.0 - 150.20.255.255
Usable range- 150.20.0.1 - 150.20.255.254

[CIDR bitmaps]
Host address- 10010110.00010100.00000000.00000000
Network address- 10010110.00010100.00000000.00000000
Network mask- 11111111.11111111.00000000.00000000
Broadcast address- 10010110.00010100.11111111.11111111
Cisco wildcard- 00000000.00000000.11111111.11111111
Network range- 10010110.00010100.00000000.00000000 -
                10010110.00010100.11111111.11111111
Usable range- 10010110.00010100.00000000.00000001 -
                10010110.00010100.11111111.11111110

[Split network]
Network- 150.20.0.0      - 150.20.63.255
Network- 150.20.64.0   - 150.20.127.255
Network- 150.20.128.0  - 150.20.191.255
Network- 150.20.192.0 - 150.20.255.255

```

LISTADO 7.1: Ejemplo de uso de sipcalc

## 7.4. Direccionamiento sin clases

Aunque la técnica de *subnetting* facilitó un uso más eficiente, no era suficiente. Años más tarde se modificó el sistema de direccionamiento para proporcionar una mayor flexibilidad, sobre todo para el espacio que aún quedaba por asignar. Se llamó a esto «direccionamiento sin clases» o *classless addressing*, aunque la designación técnica es «encaminamiento inter-dominio sin clases» o CIDR (Classless Interdomain Routing) [13].

En el direccionamiento sin clases la máscara de red no está limitada a 8, 16 o 24 bits correspondientes a las anteriores clases A, B y C, sino que puede tener cualquier tamaño, de modo que, es posible tener prefijos de red de cualquier tamaño, por supuesto siempre dentro de los 32 bits del tamaño de la dirección IP.

Con la llegada del direccionamiento sin clases, los bits iniciales de la dirección IP ya no tienen el significado especial que tenían en A, B y C. Eso quiere decir que ya no se puede saber qué máscara aplicar si disponemos solo de la dirección. Y sin la máscara no se puede determinar la dirección de red o broadcast. Por ese motivo, es necesario conocer explícitamente la máscara junto con la dirección.

Para facilitar esta tarea, se introdujo una sintaxis más compacta: la llamada «notación CIDR». Consiste simplemente en añadir el carácter / seguido del número de bits que forman el prefijo de red. Por ejemplo, la

dirección 161.67.21.13 con máscara 255.255.0.0 se puede escribir como 161.67.21.13/16.

### 7.4.1. VLSM

VLSM es la generalización del concepto de *subnetting*. Permite dividir una red en subredes de diferentes tamaños —de ahí su nombre. Las máscaras varían de tamaño entre subredes. Esto permite un aprovechamiento mucho mayor del espacio de direccionamiento.

En un caso real, el objetivo de un reparto de direcciones es parte esencial del diseño de la infraestructura de comunicaciones de una organización. No se trata simplemente de dividir sin más el espacio de direcciones disponible en cierta cantidad de bloques, como hemos visto en el ejemplo de *subnetting*. Lo normal es que una organización que disponga de un determinado bloque deba proporcionar direcciones a distintos departamentos, o usos específicos, como servidores, impresoras, telefonía IP, etc.

Por eso, la organización necesita determinar primero cuáles son las necesidades de direccionamiento de la empresa y, a partir de ahí, diseñar un esquema de direccionamiento. En ese sentido, es importante tener en cuenta las expectativas de crecimiento de cada uno de los departamentos contando con el margen de direcciones libres requerido dentro de cada bloque, dado que una vez en explotación, cambiar el esquema de direccionamiento puede ser una tarea compleja.

Aunque todo esto es igualmente importante con el direccionamiento con clases y *subnetting*, VLSM resuelve mucho mejor este problema ya que ofrece mayor flexibilidad y reduce notablemente el desperdicio de direcciones.

### 7.4.2. Bloques /30

VLSM resulta especialmente importante cuando es necesario asignar direcciones a redes pequeñas. Y la red más pequeña posible es una necesidad habitual: direccionamiento para enlaces punto a punto, como los que se utilizan para interconectar dos routers. En ese caso, solo se necesita una dirección para cada extremo del enlace. Se utiliza en ese caso la subred de tamaño mínimo, que es la que tiene una máscara de 30 bits. Un bloque de este tipo solo tiene 2 bits para *host ID* y por tanto tiene 4 direcciones: la de red, la de broadcast y dos direcciones asignables. Por ejemplo, el bloque 170.10.20.160/30 aplicado a un enlace serie (ver Figura 7.3) tiene las siguientes direcciones:

Idealmente esta necesidad la deberían cubrir los bloques /31, pero siguiendo las reglas establecidas, las únicas dos direcciones que ofrece son la de

Dirección de red: 170.10.20.160/30  
 Dirección asignable 1: 170.10.20.161/30  
 Dirección asignable 2: 170.10.20.162/30  
 Dirección de broadcast: 170.10.20.163/30

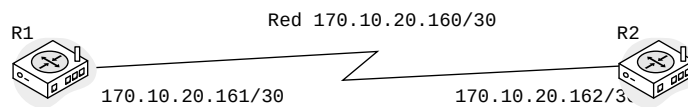


FIGURA 7.3: Direccionamiento IP de un enlace serie

red y la de broadcast, de modo que no quedaría ninguna para asignar a un nodo. Sin embargo, la [14] establece que es posible utilizar un bloque /31 precisamente para este propósito y solo para este propósito, bajo la premisa de que en un enlace punto a punto no se necesita una dirección de broadcast. Siguiendo el ejemplo anterior, el bloque 170.10.20.160/31 tendría las siguientes direcciones:

Dirección asignable 1: 170.10.20.160/31  
 Dirección asignable 2: 170.10.20.161/31

En todo caso, ten presente que aunque la especificación no es nueva, puede haber equipos de red y SO soportan que no soportan esta posibilidad.

Y por último, las direcciones /32 no definen una subred, sino que identifican exclusivamente un nodo individual. Precisamente se utiliza esa máscara para enfatizar que se trata de un nodo concreto y no de una red. Por tanto, usar direcciones como 170.10.20.160/32 y 170.10.20.161/32 en los extremos de un enlace punto a punto no sería correcto, ya que sin configuración específica adicional habría problemas de encaminamiento: por ejemplo, el sistema operativo no podría resolver direcciones mediante ARP ni determinar que hay un vecino conectado directamente.

### 7.4.3. Ejemplo de VLSM

Partiendo de la dirección 70.50.0.0/18, vamos a definir el esquema de direccionamiento para cubrir las necesidades de la topología de la Figura 7.4, incluyendo los enlaces serie que interconectan los routers.

Asumiendo que estas cantidades de nodos expresan las necesidades de cada red considerando el posible crecimiento futuro, el proceso de división debería minimizar la cantidad de direcciones que sobran (no se asignen a nodos) en cada bloque. Al mismo tiempo, es muy conveniente que los blo-

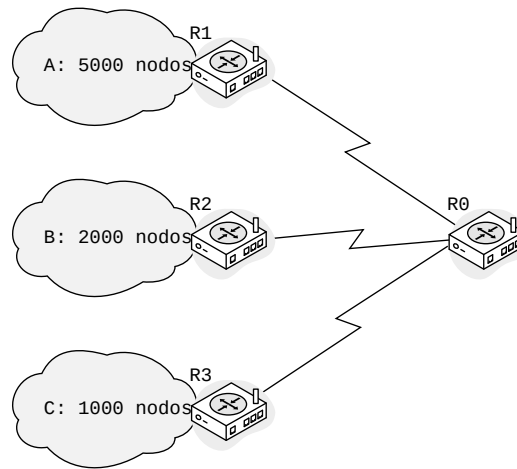


FIGURA 7.4: Topología de ejemplo para VLSM

ques sin asignar sean lo mayores posible, para que conseguir así un mejor aprovechamiento en el futuro.

Lo primero que haremos es determinar las necesidades de la topología. Veamos por ejemplo la red A que requiere 5 000 nodos. A eso tenemos que sumar 1 dirección más para la interfaz interna del router R1 además de las direcciones de red y broadcast. En total 5 003 direcciones. Ahora debemos calcular su logaritmo en base 2 para determinar cuántos bits de *host-ID* se necesitan:

$$\log_2(5\,003) = 12,28 \quad (7.1)$$

Obviamente no podemos tener bits decimales, de modo que necesitaremos 13 bits para conseguir las 5 003 direcciones. La máscara de esa subred será por tanto  $32 - 13 = /19$ . Ten en cuenta que con 13 bits tenemos realmente  $2^{13} = 8\,192$  direcciones, así que «sobran»  $8\,192 - 5\,003 = 3\,189$  direcciones.

Si aplicas estos mismos cálculos al resto de las subredes, obtendrás los resultados que aparecen en el Cuadro 7.4. Para cada red, se muestra el nombre, la cantidad de nodos que necesita (Necesidad), el tamaño en bits del *host-ID*, la máscara la subred resultante (Máscara), la cantidad total de direcciones (Total) y la cantidad de direcciones libres/desperdiciadas (Libres).

Una vez tenemos claras las necesidades podemos proceder al reparto. Por conveniencia se hace siempre el reparto desde la red más grande a la más

Red	Necesidad	host-ID	Máscara	Total	Libres
A	5 003	13	/19	8 192	3 189
B	2 003	11	/21	2 048	45
C	1 003	10	/22	1 024	21
R1-R0	2	2	/30	4	0
R2-R0	2	2	/30	4	0
R3-R0	2	2	/30	4	0

CUADRO 7.4: Ejemplo de VLSM: Necesidades de las subredes

pequeña. Veámoslo paso a paso. Como el bloque de partida es /18 y la red A necesita un bloque /19 debemos utilizar 1 bit. La tabla 7.5 muestra esta división. En la primera columna se muestra el tercer byte de la dirección que es dónde está la frontera entre *net-ID* y *host-id*. El bit marcado en negrita es el que se ha utilizado para dividir la red original.

3 <sup>er</sup> byte	Dirección de red	Red
0000 0000	70.50.0.0/19	Red A
0010 0000	70.50.32.0/19	Libre

CUADRO 7.5: Ejemplo de VLSM: Red A

El siguiente paso es asignar espacio para la red B. Como requiere un bloque /21 y tenemos un bloque **70.50.32.0/19**, significa que necesitamos solo una cuarta parte, de modo que utilizaremos 2 bits adicionales del *host-id*. La tabla 7.6 muestra la división.

3 <sup>er</sup> byte	Dirección de red	Red
0010 <b>0</b> 000	70.50.32.0/21	Red B
0010 <b>1</b> 000	70.50.40.0/21	Libre
0011 <b>0</b> 000	70.50.48.0/21	Libre
0011 <b>1</b> 000	70.50.56.0/21	Libre

CUADRO 7.6: Ejemplo de VLSM: Red B

La red C necesita un bloque /22, así que dividimos el primer bloque libre (70.50.40.0/21) con 1 bit adicional. La tabla 7.7 muestra la división.

Por último, utilizaremos el primer bloque libre (70.50.44.0/22) para asignar subredes a los 3 enlaces serie. En este caso se utilizan 8 bits adicionales para obtener máscaras /30. La tabla 7.8 muestra la división. Al realizar la división con 8 bits tendríamos 256 de estos bloques /30, pero como solo necesitamos 3. En la tabla se aparecen únicamente los 4 primeros.

3 <sup>er</sup> byte	Dirección de red	Red
0010 1000	70.50.40.0/22	Red C
0010 1100	70.50.44.0/22	Libre

CUADRO 7.7: Ejemplo de VLSM: Red C

3 <sup>er</sup> byte	4 <sup>o</sup> byte	Dirección de red	Red
0010 1100	0000 0000	70.50.44.0/30	R1-R0
0010 1100	0000 0100	70.50.44.4/30	R2-R0
0010 1100	0000 1000	70.50.44.8/30	R3-R0
0010 1100	0000 1100	70.50.44.12/30	Libre

CUADRO 7.8: Ejemplo de VLSM: Enlaces serie

Veamos ahora otros datos relevantes de cada bloque. En el Cuadro 7.9 se muestra para cada bloque, la dirección de red, la máscara en decimal, la primera y última dirección asignable y la dirección de broadcast.

Nombre	Dir. red	Máscara	1 <sup>a</sup> dir.	Última dir.	Broadcast
A	70.50.0.0/19	255.255.224.0	70.50.0.1	70.50.31.254	70.50.31.255
B	70.50.32.0/21	255.255.248.0	70.50.32.1	70.50.39.254	70.50.39.255
C	70.50.40.0/22	255.255.252.0	70.50.40.1	70.50.43.354	70.50.43.255
R1-R0	70.50.44.0/30	255.255.255.252	70.50.44.1	70.50.44.2	70.50.44.3
R2-R0	70.50.44.4/30	255.255.255.252	70.50.44.5	70.50.44.6	70.50.44.7
R2-R0	70.50.44.8/30	255.255.255.252	70.50.44.9	70.50.44.10	70.50.44.11

CUADRO 7.9: Ejemplo de VLSM: Resultado

Por último, en el Cuadro 7.10 aparecen todos los bloques que han quedado libres. La primera columna es solo un identificador para referirnos a ellos. Los bloques 1 a 7 corresponden con la fragmentación que hemos causado al asignar los enlaces serie. Hemos agregados en lo posible el espacio disponible para evitar esos 253 bloques /30 que no necesitamos. El bloque 9 también es una agregación de los bloques 70.50.48.0/21 y 70.50.56.0/21 que quedaron libres. Como son contiguos y difieren solo en el primer bit del *net-id* es posible expresarlos como un solo bloque /20.

Id	3 <sup>er</sup> byte	4 <sup>o</sup> byte	Dirección del bloque
1	0010 1100	0000 1100	70.50.44.12/30
2	0010 1100	0001 0000	70.50.44.16/28
3	0010 1100	0010 0000	70.50.44.32/27
4	0010 1100	0100 0000	70.50.44.64/26
5	0010 1100	1000 0000	70.50.44.128/25
6	0010 1101	0000 0000	70.50.45.0/24
7	0010 1110	0000 0000	70.50.46.128/23
8	0010 1100	0000 0000	70.50.44.0/22
9	0011 0000	0000 0000	70.50.48.0/20

CUADRO 7.10: Ejemplo de VLSM: Bloques libres

## 7.5. Direcciones especiales

Además de las direcciones de red y broadcast, hay otras direcciones especiales que tienen usos particulares o predefinidos. Veamos las más importantes:

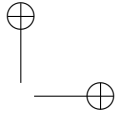
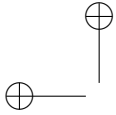
**Dirección nula** La dirección `0.0.0.0` —conocida como `INADDR_ANY`— se utiliza en varias situaciones diferentes cuando no se conoce, no se dispone o no tiene sentido utilizar una dirección IP concreta. Por ejemplo, en las peticiones ARP, en una tabla de encaminamiento para indicar entrega directa o router por defecto y otros casos que veremos a lo largo del libro.

**Direcciones *loopback*** Cualquier dirección del rango `127/8` solo tiene sentido en el propio nodo y no se puede asignar a ninguna interfaz de red que pueda ser accesible desde fuera. La dirección más común para este uso es `127.0.0.1`, de la que ya hemos hablado en capítulos anteriores.

**Direcciones privadas** Son direcciones que se pueden utilizar únicamente en redes privadas, y por ello no necesitan autorización ni asignación por parte de ninguna autoridad, pero son ignoradas por los routers fuera de la red privada. Las veremos en detalle en el capítulo 20.

**Direcciones de enlace local** Son direcciones que se utilizan para comunicar dispositivos en la misma red local, sin necesidad de un router. Hablaremos de ellas también en el capítulo 20.

**Direcciones multicast** Son direcciones que identifican a grupos. No están asignadas a ningún interfaz de red. Sirven para enviar un mensaje a todos los dispositivos que pertenezcan a ese grupo. Se utilizan en aplicaciones como la IPTV, radio por IP, distribución de contenidos, etc. Son las direcciones de clase D y empiezan por `1110`. El uso de



esta clase no fue descartado con la llegada del direccionamiento sin clases, y se sigue utilizando en la actualidad.

## Y ¿qué más?

En realidad no hay mucho más que saber sobre el direccionamiento IP. Con lo que has visto en este capítulo ya tienes una buena base para diseñar por ti mismo un esquema de direccionamiento de cualquier organización. El diseño puede ser arbitrariamente complejo y tendrás que considerar muchos factores, pero el proceso de diseño se basa en los conceptos que has visto aquí. Incluso con IPv6 la base es prácticamente la misma en lo que concierne a la subdivisión de bloques, ya que nunca existió direccionamiento con clase en IPv6. En el capítulo 10 veremos cómo se asignan estas direcciones a las interfaces de red, ya sea manual o automáticamente con DHCP, pero esa tarea en realidad es independiente del esquema de direccionamiento.

