IPv4 addressing

This chapter covers

- The fields of the IPv4 header
- The binary number system
- How to convert between decimal and binary
- The structure of IPv4 addresses
- How to configure IPv4 addresses on Cisco routers

In chapter 6, we focused on Layer 2 of the TCP/IP model: how switches use information in the Ethernet header to make forwarding decisions. In this chapter, we will move up a layer to Layer 3 and look at the contents of the *Internet Protocol version 4* (IPv4) header, focusing on IPv4 addressing.

We are now in the realm of routers, rather than switches. Whereas switches use information in the Layer 2 header to decide how to forward messages to their proper destinations, routers use information in the Layer 3 header to make their forwarding decisions. In this chapter, we won't yet focus on exactly how routers make those forwarding decisions; we will leave that for part 2 of this book. Instead, we will first focus on the contents of the IPv4 header and the addresses used in that header.

The specific exam topic we will cover is topic 1.6: Configure and verify IPv4 addressing and subnetting. However, IPv4 addressing is not only relevant to exam topic 1.6; it is a fundamental topic that is essential to understanding nearly any other CCNA exam topic. Also note that we will cover *subnetting*, the second half of topic 1.6, in part 2 of this book.

Given the name *IPv4*, you may wonder what happened to previous versions. The history and characteristics of IPv0, v1, v2, and v3, although important steps in the evolution toward IPv4, are not necessary to know for the CCNA exam, so we will not cover them. It is IPv4, officially defined in RFC 791 (simply titled "Internet Protocol"), which is the foundation of modern networks such as the Internet.

Note

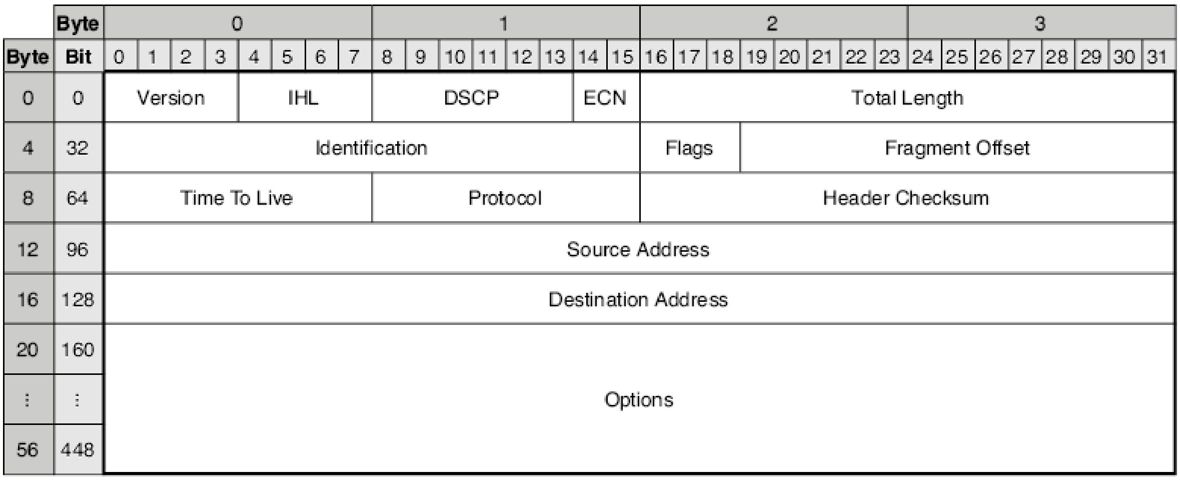
In addition to IPv4, IPv6 is another major exam topic which has its own part in this book. IPv6 was introduced in 1995 to replace IPv4, but its adoption has been slow. Although IPv6 adoption is accelerating as the available IPv4 address pool runs out, it seems that for the foreseeable future network engineers will have to be familiar with both IPv4 and IPv6.

7.1 The IPv4 header

Before looking at the details of IPv4 addressing, it's helpful to understand the header which contains those addresses. However, the IPv4 header doesn't just contain IPv4 addresses; it contains a variety of fields, each serving a different role in enabling the end-to-end delivery of packets (the role of layer 3).

The IPv4 header is more complex than the Ethernet header, as you'll probably notice when looking at figure 7.1. In total there are 14 fields (although the Options field is optional), whereas the Ethernet header and trailer only have four (six if you include the Preamble and SFD).

Figure 7.1 The format of the IPv4 header. The header is typically 20 bytes in size (the minimum size) but can be up to 60 bytes if the Options field is used.



Before we examine the purpose of each field of the header, I want to clarify how to read figure 7.1. The fields of the header are contained within the thick border and should be read from left to right, top to bottom; the first bit of the header is in the top-left position, and the last bit is in the bottom-right position. The numbers along the top indicate that each row is 4 bytes (32 bits) in length. The numbers on the left of each row indicate the starting byte/bit number of that row. For example, the second row starts at byte 4 (the fifth byte), which is bit 32 (the 33rd bit).

Note

In networking, you'll have to get used to counting from 0. For example, the range from 0 to 31 includes 32 bits in total: bit 0 is the first bit, bit 1 is the second bit, etc. Likewise, byte 0 is the first byte, byte 1 is the second byte, byte 2 is the third byte, byte 3 is the fourth byte, etc.

As stated above, the Options field is optional (and variable in size), so the length of the IPv4 header is variable. Without the Options field the header is 20 bytes in length, from the first bit of the Version field to the last bit of the Destination Address field. With the Options field at its maximum size (40 bytes), the IPv4 header is 60 bytes in length. However, the Options field is rarely used and is beyond the scope of the CCNA exam.

Exam Tip

For the purpose of the CCNA exam, don't worry about memorizing the length and position of each field of the IPv4 header. Questions on the CCNA exam are more substantial than trivia like "What's the length of field X?". For the purpose of this chapter, it's sufficient to have a basic understanding of the purpose of each field. This chapter focuses on the IPv4 addresses in the Source Address and Destination Address fields, and in later chapters we will look at other fields in greater detail as required.

7.1.1 The Version field

The first field of the IPv4 header is the *Version* field. It is four bits in length. As I mentioned previously, there are two versions of IP used in modern networks: IPv4 and IPv6. The purpose of this field is simple: to indicate which version of IP is being used. In modern networks you can expect to find one of two values in this field:

- A value of 0b0100 (0d4) indicates IPv4.
- A value of 0b0110 (0d6) indicates IPv6.

Note

As mentioned in chapter 6, the prefix *0b* indicates a binary number and the prefix *0d* indicates a decimal number. We will look at how to convert between the two number systems later in this chapter

7.1.2 The IHL field

The second field is the *Internet Header Length* (IHL) field, which is four bits in length. This field is used to indicate the length of the IPv4 header. The reason this field is necessary is because the IP header is variable in length, depending on whether the Options field is present or not (and the Options field itself is variable in length, too).

The IHL field indicates the length of the IPv4 header *in four-byte increments*. For example, if the value of this field is 5, it means the header is 20 bytes in length (the minimum length of the IPv4 header).

Note

A value less than 5 should not be used in this field, because the IPv4 header cannot be less than 20 bytes in length.

Any value greater than 5 in the IHL field indicates that the Options field is present in the header. The maximum value of the IHL field is 15, indicating that the header is 60 bytes in length (the maximum length of the IPv4 header). In that case the Options field is 40 bytes in length and the rest of the header is 20 bytes.

7.1.3 The DSCP and ECN fields

The next two fields are *Differentiated Services Code Point* (DSCP), which is six bits in length, and *Explicit Congestion Notification* (ECN), which is two bits in length. Together they form the *Type of Service* (ToS) byte. This byte of the IPv4 header has had multiple definitions over the history of IPv4, but DSCP+ECN is the current definition.

These fields are used for *Quality of Service* (QoS), which is a network feature used to prioritize specific types of network traffic over other types. A common use case for QoS is to prioritize delay-sensitive network traffic - network traffic for which it is very important to reach the destination as soon as possible, without delay. One example of this is voice and video traffic; I think most of us know how frustrating it can be to have a Zoom call (or a call using any similar application) with poor quality. QoS helps ensure that this traffic is forwarded with as little delay as possible.

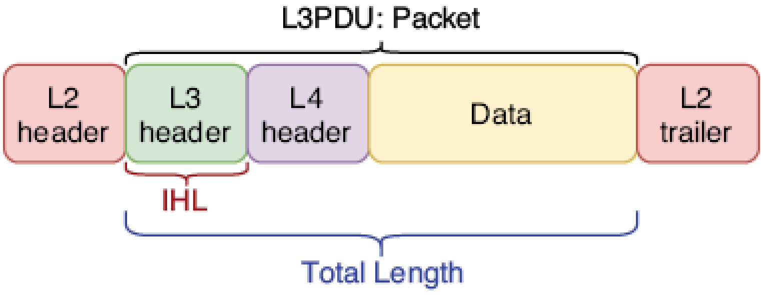
Note

QoS is another CCNA exam topic, and we will cover it in chapter 36 of this book.

7.1.4 The Total Length field

The *Total Length* field is a 16-bit field that indicates the total length of the packet - the IPv4 header and its payload. Don't confuse this with the IHL field, which indicates the length of the IPv4 header alone. Figure 7.2 illustrates the difference between the IHL and Total Length fields.

Figure 7.2 The difference between the IHL and Total Length fields. The IHL field indicates the length of the IPv4 header (Layer 3 header), whereas the Total Length field indicates the length of the entire packet. The Layer 2 header and trailer are shown to emphasize that a packet will always be encapsulated in a frame before being sent; a packet alone is not ready to be sent over the physical medium.



Another difference between the IHL and Total Length fields is that the value of the Total Length field indicates the length of the packet in bytes, rather than four-byte increments. A value of 100 in the Total Length field means the packet is 100 bytes in length. A value of 1000 in the Total Length field means the packet is 1000 bytes in length.

7.1.5 The Identification, Flags, and Fragment Offset fields

The *Identification*, *Flags*, and *Fragment Offset* fields, 32 bits in total, are used together to support packet *fragmentation* - when a packet is broken up into multiple smaller packets called *fragments*. IPv4 uses a concept called *Maximum Transmission Unit* (MTU) to indicate the maximum size a packet should be, and any packet larger than the MTU will be fragmented. Then, the final destination host of the packet reassembles the fragments to restore the original packet.

The typical MTU is 1500 bytes, and this should be supported on all modern devices. However, if for some reason a router in the packet's path to the destination has a lower MTU, it will fragment the packet. Another possibility is that a host sends packets larger than the standard 1500-byte size (sometimes packet sizes up to 9000 bytes are used). If a router in the path to the destination doesn't support those larger packets, it will fragment them. Let's briefly examine the role of each of these three fields.

Identification field

This field is 16 bits in length and is used to identify which original packet a fragment belongs to. When a packet is fragmented, all of its fragments must have the same value in this field.

Flags field

This field is three bits in length and is used to control and identify fragments. The three bits of this field (bit 0, bit 1, and bit 2) are defined as follows:

- Bit 0: *Reserved*. This bit's use hasn't been defined, so it is always set to 0.
- Bit 1: *Don't Fragment* (DF) bit. If this bit is set to 1, it means the packet should not be fragmented. In that case, if the packet's size is greater than the MTU it will be discarded.
- Bit 2: *More Fragments* (MF) bit. If this bit is set to 1, it means there are more fragments remaining - this one isn't the last. The final fragment of the packet will have a value of 0 in this field (indicating that there are no more fragments). An unfragmented packet will always have a value of 0 for this bit.

Fragment Offset field

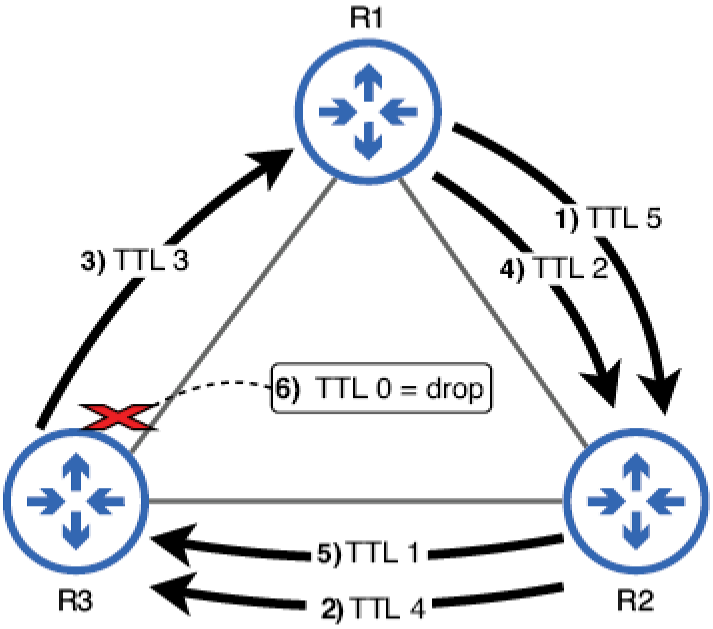
This field is 13 bits in length and is used to indicate the position of the fragment within the original packet. This allows fragmented packets to be reassembled even if the fragments arrive out of order. This is rare, but if there are multiple paths to a destination, different fragments might take different paths, in which case they may arrive at the destination out of order.

7.1.6 The TTL field

The *Time To Live* (TTL) field is an eight-bit field. When a host sends a packet, it will set a certain value in this field (a common value is 64), and then each router that forwards the packet will decrement the value in this field by 1. If the value reaches 0, the router will drop the packet.

The reason for this mechanism is to prevent packets from *looping* around the network infinitely. A *loop* is when a message travels around the network without being able to find its destination. For example, if there are three routers (R1, R2, and R3), a looping packet might be passed from R1 to R2, from R2 to R3, from R3 to R1, from R1 to R2, from R2 to R3, etc. in a loop.

Figure 7.3 A looped packet is dropped due to the TTL mechanism. (1) R1 forwards the packet to R2 with a TTL of 5. (2) R2 forwards the packet to R3 with a TTL of 4. (3) R3 forwards the packet to R1 with a TTL of 3. (4) R1 forwards the packet to R2 with a TTL of 2. (5) R2 forwards the packet to R3 with a TTL of 1. (6) R3 wants to forward the packet to R1 but drops the packet because it must decrement the TTL to 0.



7.1.7 The Protocol field

In the previous chapter we covered the ping utility, which is a component of ICMP. If a packet contains an ICMP message, that is indicated with a value of 1 in this field. Below are the Protocol field values of some protocols we will cover in this book:

1: ICMP

- 6: *Transmission Control Protocol* (TCP)
- 17: *User Datagram Protocol* (UDP)
- 89: *Open Shortest Path First* (OSPF)

7.1.8 The Header Checksum field

7.1.9 The Source Address and Destination Address fields

7.1.9 The Source Address and Destination Address fields

7.1.10 The Options field

7.1.10 The Options field

7.2 The binary number system

7.2.1 Decimal

After counting up to 0d99 (9 in the “tens” position and 9 in the “ones” position), we have to add a third digit; the number after 0d99 is 0d100 - a *1* in the “hundreds” position, and a 0 in both the “tens” and “ones” positions. Because decimal uses 10 digits, the value of each additional position increases tenfold as you add more digits: 1, then 10, then 100, then 1000, etc. That’s why, in the number 1009 (for example), the *1* on the left has a greater value than the *9* on the right, even though on their own the number nine has a greater value than the number one.

7.2.2 Binary

The value after 0b1 is 0b10 - a *1* in the "twos" position and a *0* in the "ones" position; this is equivalent to 0d2. After 0b10 is 0b11 (equivalent to 0d3), and then once again both positions have reached their maximum value, so a third digit is needed. This results in 0b100 (equivalent to 0d4). Whereas the value of each decimal position increases tenfold, the value of each binary position doubles, because binary uses two digits. Figure 7.3 shows an eight-digit binary number with the value of each position above each bit (binary digit).

Figure 7.3 An eight-bit (one-byte) number with the value of each bit written above. The decimal equivalent of 0b10101101 is 0d173. This can be calculated by adding the value of each bit that is set to "1".

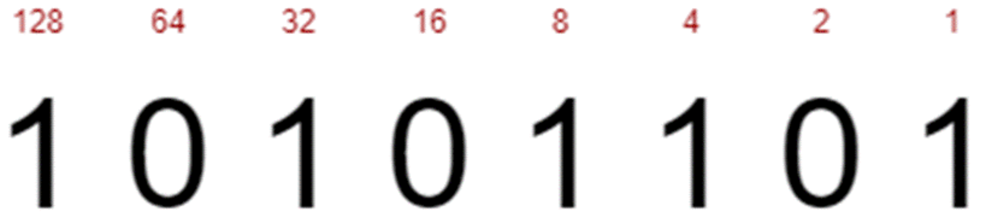
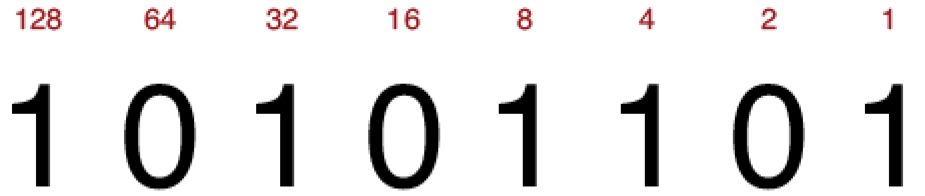


Figure 7.4 An 8-bit (1-byte) number with the value of each bit written above. The decimal equivalent of 0b10101101 is 0d173. This can be calculated by adding the value of each bit that is set to 1.



Note

Table 7.1 lists some decimal numbers and their binary equivalents. With only two digits available, binary numbers quickly grow in size (the value after 0b11111 would be 0b100000). That is why, although computers use binary, we convert those binary values to other number systems (decimal and hexadecimal) to make them more human-readable.

Table 7.1 Decimal numbers and their binary equivalents

Dec.	Bin.	Dec.	Bin.	Dec.	Bin.	Dec.	Bin.
0	0	8	1000	16	10000	24	11000
1	1	9	1001	17	10001	25	11001
2	10	10	1010	18	10010	26	11010

3	11	11	1011	19	10011	27	11011
4	100	12	1100	20	10100	28	11100
5	101	13	1101	21	10101	29	11101
6	110	14	1110	22	10110	30	11110
7	111	15	1111	23	10111	31	11111

Table 7.1 Decimal numbers and their binary equivalents (view table figure)

Dec.	Bin.		Dec.	Bin.		Dec.	Bin.		Dec.	Bin.
0	0		8	1000		16	10000		24	11000
1	1		9	1001		17	10001		25	11001
2	10		10	1010		18	10010		26	11010
3	11		11	1011		19	10011		27	11011
4	100		12	1100		20	10100		28	11100
5	101		13	1101		21	10101		29	11101
6	110		14	1110		22	10110		30	11110
7	111		15	1111		23	10111		31	11111

Converting binary numbers to decimal

Converting binary numbers to decimal

Figure 7.4 The binary number 00101111 is equal to 47 in decimal. To calculate this, add the value of each bit that is set to “1”: 32 + 8 + 4 + 2 + 1 = 47.

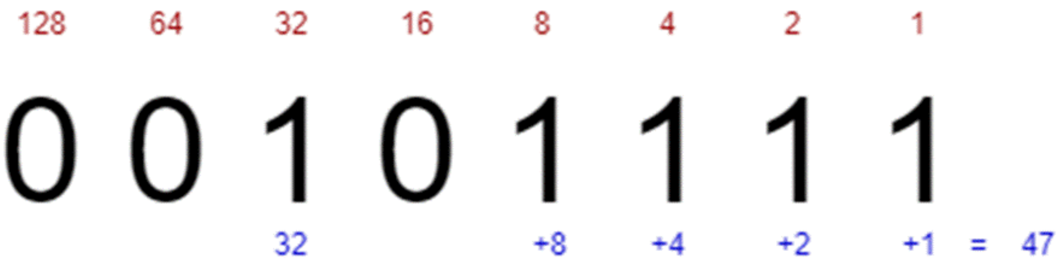
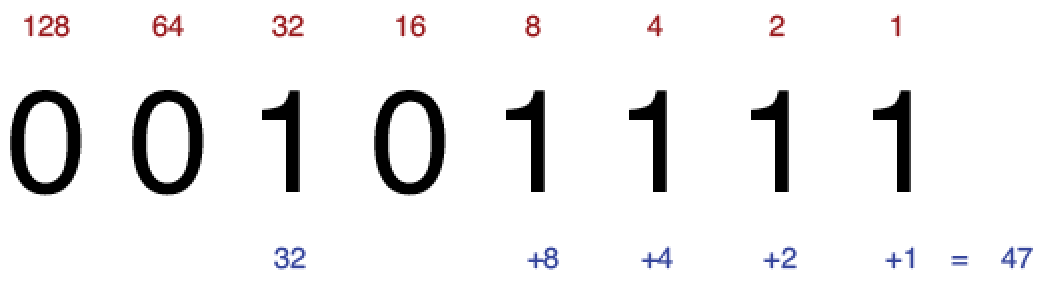


Figure 7.5 The binary number 00101111 is equal to 47 in decimal. To calculate this, add the value of each bit that is set to 1: 32 + 8 + 4 + 2 + 1 = 47.



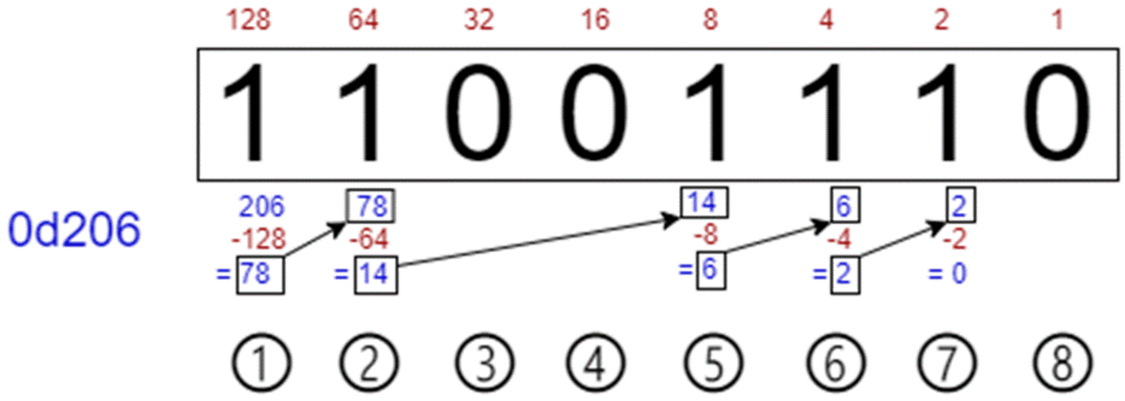
Note

Note

Converting decimal numbers to binary

Converting decimal numbers to binary

Figure 7.5 The process of converting a decimal number (206) to a binary number. ①Subtracting 128 from 206 gives a remainder of 78. Write a "1" in the 128 position. ②Subtracting 64 from 78 gives a remainder of 14. Write a "1" in the 64 position. ③32 cannot be subtracted from 14. Write a "0" in the 32 position. ④16 cannot be subtracted from 14. Write a "0" in the 16 position. ⑤Subtracting 8 from 14 gives a remainder of 6. Write a "1" in the 8 position. ⑥Subtracting 4 from 6 gives a remainder of 2. Write a "1" in the 4 position. ⑦Subtracting 2 from 2 gives a remainder of 0. Write a "1" in the 2 position. ⑧1 cannot be subtracted from 0. Write a "0" in the 1 position. We now have the answer: 0d206 is equivalent to 0b11001110.

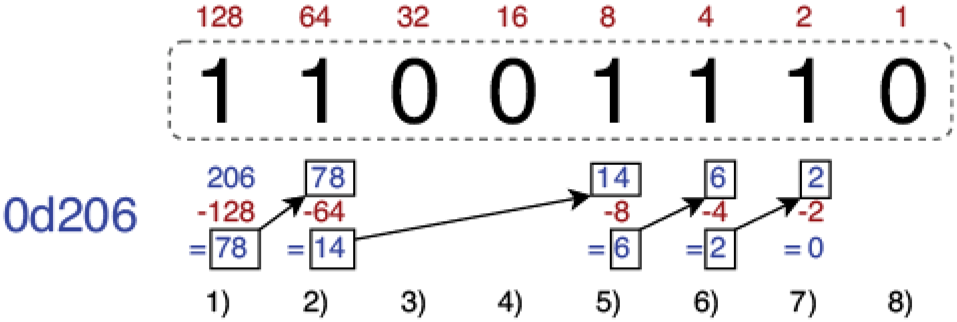


1. Like converting from binary to decimal, this process becomes much easier with practice, and eventually you should be able to do it in your head. To practice, write some random numbers from 0 to 255 (56, 127, 201, 199, etc.) and try converting them into binary.
2. For additional practice with converting between decimal and binary (in both directions), you can try the *Binary Game* on Cisco Learning Network: https://learningnetwork.cisco.com/s/binary-game. Try it a few times each day; as you practice and improve, your scores in the Binary Game should increase and you'll find yourself able to do the necessary calculations in your head.

3. **Exam Tip**
4. Being able to quickly convert between decimal and binary is a big help on the CCNA exam, especially when it comes to *subnetting* (the topic of chapter 12). The CCNA exam has a two hour time limit - don't unnecessarily spend time doing binary to decimal and decimal to binary conversions. A bit of practice with Cisco's Binary Game goes a long way.
5. **Exam Applications**
6. Binary is a fundamental topic with applications to various CCNA exam topics. In addition to IPv4 addressing (the topic of this chapter), below are some other topics that require you to be proficient with binary, including converting between binary and decimal:
7. **IPv4 subnetting**: *Subnetting* is the process of dividing networks into smaller networks, and is the second half of exam topic 1.6: *Configure and verify IPv4 addressing and subnetting*. To subnet IPv4 networks, you need to be able to convert IPv4 addresses from decimal to binary, and vice-versa. We will cover subnetting in chapter 12 of this book.
8. **IPv6 addressing**: This is exam topic 1.8: *Configure and verify IPv6 addressing and prefix*. To understand IPv6 addresses you need to be able to convert between binary, decimal, and hexadecimal (because IPv6 addresses are usually written in hexadecimal). We will cover IPv6 in part 6 of this book.

We now have the answer: 0d206 is equivalent to 0b11001110.

Figure 7.6 The process of converting a decimal number (206) to a binary number (11001110) by subtracting each bit's decimal value



7.3 IPv4 Addressing

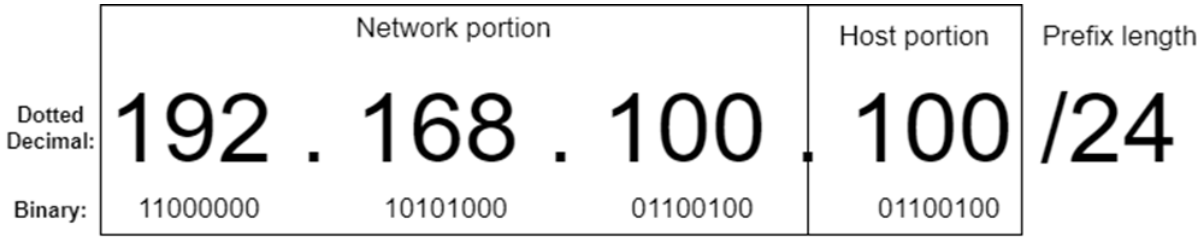
1. An IPv4 address is a 32-bit number that identifies a host at layer 3 of the TCP/IP Model. IP addresses (whether IPv4 or IPv6) are used to address a message to its final intended recipient, unlike MAC addresses which are used to address a message to the next hop. Whereas switches are said to be *layer 2 devices*, routers are said to be *layer 3 devices* or to *operate at layer 3*, because they make forwarding decisions based on the destination IP address of messages (located in the layer 3 header).
2. **Note**

3. In this chapter we will look at how to configure IPv4 addresses on routers, but we will cover how routers forward packets in part 2 of this book.

4. **7.3.1 The structure of an IPv4 address**

5. IPv4 addresses are 32 bits in length, but a 32-bit string of 1s and 0s isn't very human-readable or easy to remember. To make them easier to read, IPv4 addresses are represented using decimal instead of binary. To simplify it even further, we first split the 32-bit IPv4 address into four groups of eight bits called *octets*, separated by a period, and then convert each of the octets to decimal; this is called *dotted decimal notation*. This is why, for the purpose of the CCNA, you only need to be able to convert between binary and decimal for numbers of up to 8 bits. Figure 7.6 shows an IPv4 address written in dotted decimal as well as in binary.

6. **Figure 7.6 An IPv4 address written in both dotted decimal and binary. The 32-bit address is split into four octets consisting of eight bits each. The address is divided into two parts: the *network portion* and the *host portion*. The *prefix length* indicates the size of the network portion, in bits and the remainder is the host portion.**



7. **Note**

8. You may wonder what the difference is between an *octet* and a *byte*, both of which I have defined as a group of eight bits. An *octet* always means eight bits. However, a *byte* isn't necessarily eight bits; a byte is the minimum unit of data that a computer can read from or write to at one time. This is almost always eight bits, but in the past there have been computers that use six-, seven-, and nine-bit bytes. Therefore, the term *octet* is sometimes used instead to refer to a group of eight bits. In the context of IPv4 addresses, *octet* is preferred.

IPv4 addresses are divided into two parts: the *network portion* and the *host portion*. This can be likened to a home address; the network portion is like the street name, and the host portion is like the house number. All houses on a street share the same street name, but have a unique house number. Likewise, the IP addresses of all hosts connected to a LAN will share the same network portion, but have a unique host portion.

Prefix length

Exam Tip

Note

Exam applications

All hosts in the same LAN as the host with IPv4 address *192.168.100.100* will share the same network portion; the first three octets of their IPv4 addresses will be the same (192.168.100). However, each host will have a unique host portion; the final octet will be unique. Some possible addresses of other hosts in the LAN could be 192.168.100.1, 192.168.100.178, 192.168.100.234, etc.

- Figure 7.7 shows two networks: *LAN 1* and *LAN 2*. Notice that the IP address of each host in LAN 1 begins with *192.168.1*, and the IP address of each host in LAN 2 begins with *192.168.2*. The router (R1) serves to connect the two LANs together; its G0/0 interface has IP address 192.168.1.1/24, and its G0/1 interface has IP address 192.168.2.1/24. Hosts in the separate LANs can communicate with each other via R1. Notice that the switches do not have IP addresses - this is because switches are not *layer 3 aware*. Switches operate at layer 2 of the TCP/IP model and do not get involved with layer 3.
- **Note**
- When talking about routers, the term *interface* is typically used instead of *port*.

7.3 IPv4 addressing

Note

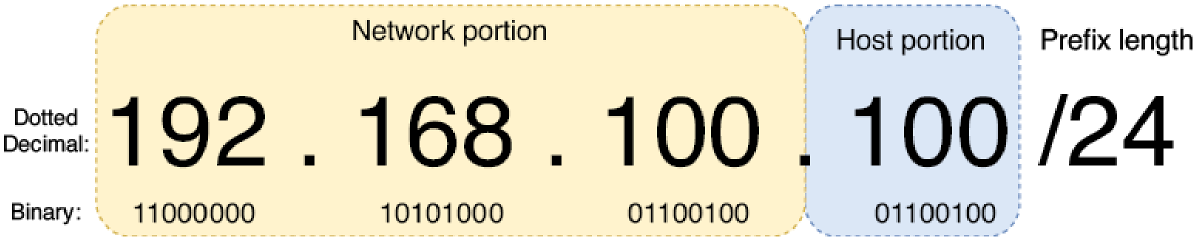
Note

Netmasks

7.3.1 The structure of an IPv4 address

Like IPv4 addresses, netmasks are usually written in dotted decimal notation. Figure 7.8 shows an IPv4 address (172.16.20.21) with a netmask (255.255.0.0). The first 16 bits of the netmask are "1", meaning the first 16 bits of the IPv4 address are the network portion.

Figure 7.7 An IPv4 address written in both dotted decimal and binary. The 32-bit address is split into four octets consisting of 8 bits each. The address is divided into two parts: the *network portion* and the *host portion.* The *prefix length* indicates the size of the network portion in bits, and the remainder is the host portion.



Octet and byte

A netmask is often called a *subnet mask*; we will cover the topic of *subnets* in chapter 12.

Prefix length

Prefix length: /8 = netmask: 255.0.0.0

Note

Prefix length: /24 = netmask: 255.255.255.0

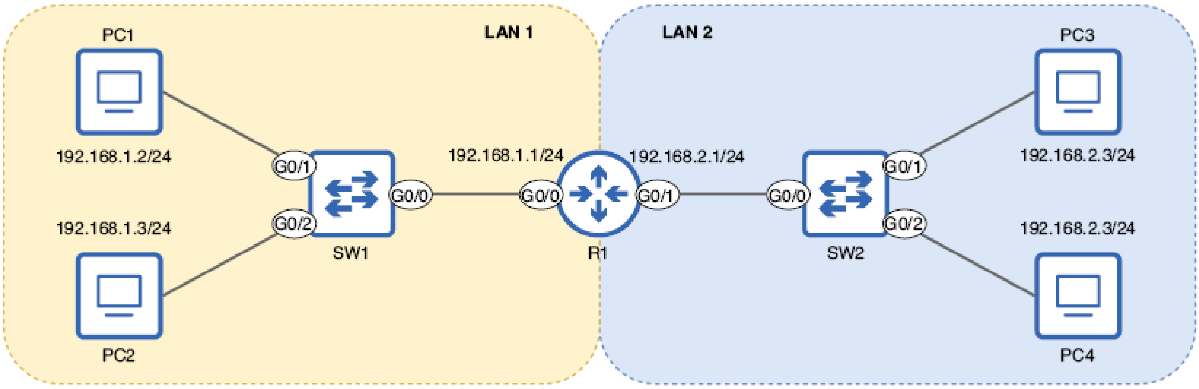
Note

A netmask is always a series of 1s followed by a series of 0s; this is because IPv4 addresses are always structured to have the network portion on the left (the most significant bits) and the host portion on the right (the least significant bits). Netmasks like *0.0.0.255* or *255.0.255.0* are not possible.

Note

Unlike MAC addresses, which are assigned to a device by its manufacturer, IP addresses must be assigned by the engineer or admin configuring the device. Let's look at how to configure IP addresses on a Cisco router.

Figure 7.8 Two networks (LAN 1 and LAN 2) connected via a router (R1). IP addresses of hosts in each LAN share the same network portion: 192.168.1 in LAN 1 and 192.168.2 in LAN 2.



Note

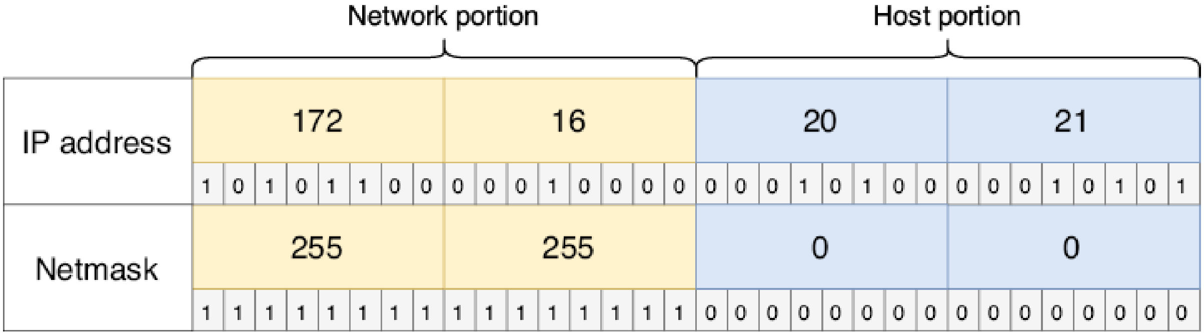
Figure 7.9 zooms in on R1 from figure 7.7, and shows how to configure IP addresses on and enable R1's G0/0 and G0/1 interfaces. In the rest of this section we will analyze these configurations and also use show commands to verify the status of R1's interfaces before and after configuration.

Netmasks

Note

The switches and PCs present in figure 7.7 have been replaced with perpendicular lines at the end of the connections in figure 7.9. This is a common technique in network diagrams to indicate that a LAN is connected to an interface, but its details are not important to the diagram. Figure 7.9 focuses on R1, so there is no need to show the switches and PCs.

Figure 7.9 An IPv4 address (top) and its netmask (bottom). The first 16 bits of the netmask are set to 1, indicating that the first 16 bits of the IPv4 address are the network portion. This is equivalent to a /16 prefix length.



	Network portion		Host portion	
IP address	172	16	20	21
	1 0 1 0 1 1 0 0	0 0 0 1 0 0 0 0	0 0 0 1 0 1 0 0	0 0 0 1 0 1 0 1
Netmask	255	255	0	0
	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Note

```
R1# show ip interface brief Interface IP-Address OK? Method Status Protocol
GigabitEthernet0/0 unassigned YES unset administratively down down GigabitEthernet0/1
unassigned YES unset administratively down down GigabitEthernet0/2 unassigned YES
unset administratively down down GigabitEthernet0/3 unassigned YES unset
administratively down down
```

The Interface column lists R1's interfaces - it has four, and we will configure two of them. The IP-Address column will list the IP address of each interface after we have configured them, but currently it just states unassigned.

- The Status column lists the physical status of each interface. If the interface is connected to another device the status will be up, if it isn't the Status will be down, and if the interface is manually disabled it will be administratively down (regardless of whether it is connected to another device or not). As shown above, the default state is administratively down - Cisco router interfaces are disabled by default and must be manually enabled.
- The Protocol column indicates whether the layer 2 protocol of the interface is functioning properly. For an Ethernet interface this is fairly simple - if the Status column says up, the Protocol should be up as well. If the Status column says down or administratively down, the Protocol should be down.
- **Configuration**

Note

```
R1# configure terminal Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface gigabitethernet0/0 R1(config-if)#
```

7.3.2 Configuring IPv4 addresses on a router

Note

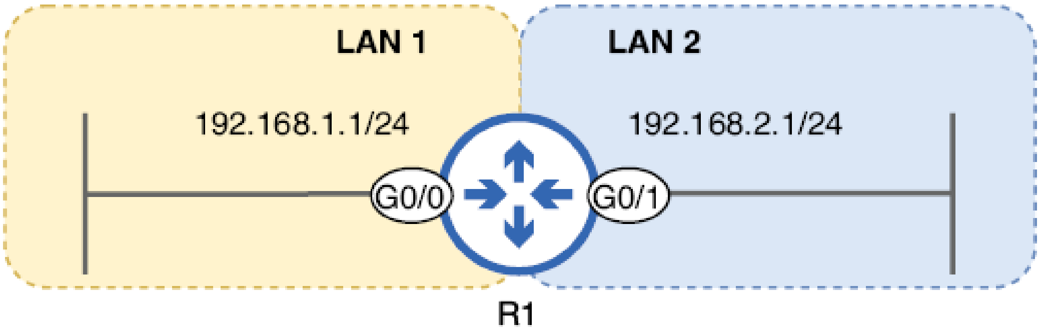
Note

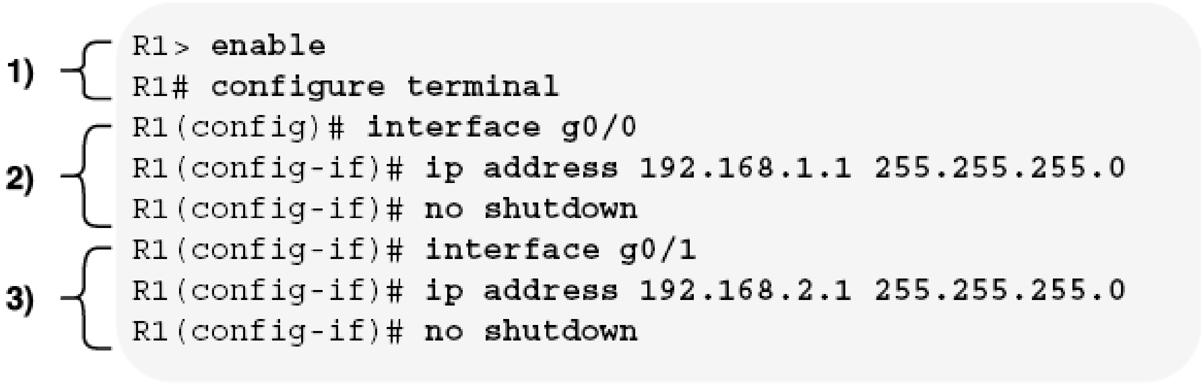
The command to configure an interface's IP address is ip address *ip-address netmask*; as I mentioned previously, you need to know netmasks when configuring IP addresses in Cisco IOS. In the following example I configure the IP address of R1's G0/0 interface. The netmask is 255.255.255.0 because the prefix length is /24 - the first 24 bits of the netmask are set to "1" and the last eight are set to "0".

R1(config-if)# ip address 192.168.1.1 255.255.255.0

1. However, G0/0 still isn't ready to forward traffic; the interface is still disabled. To change that you must use the no shutdown command, as shown in the example below. After issuing the command two messages are displayed indicating that the interface is up and running. The first message indicates that the interface is physically operational (the Status column of show ip interface brief), and the second message indicates that the layer 2 protocol is operational (the Protocol column of show ip interface brief).
2. R1(config-if)# no shutdown %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up
3. **Note**

Figure 7.10 How to configure IP addresses and enable router interfaces. R1 is connected to two LANs: 192.168.1.1/24 (G0/0) and 192.168.2.1/24 (G0/1).





```
1) { R1> enable
     R1# configure terminal
2) { R1(config)# interface g0/0
     R1(config-if)# ip address 192.168.1.1 255.255.255.0
     R1(config-if)# no shutdown
3) { R1(config-if)# interface g0/1
     R1(config-if)# ip address 192.168.2.1 255.255.255.0
     R1(config-if)# no shutdown
```

Note

R1(config-if)# interface g0/1 R1(config-if)# ip address 192.168.2.1 255.255.255.0 R1(config-if)# no shutdown %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

Preverification

To access interface configuration mode for another interface, you don't have to return to global configuration mode; you can do it directly from interface configuration mode. Note that there is no indication that I switched from configuring G0/0 to G0/1 - again, always double-check that you used the correct interface name after the interface command.

```
R1# show ip interface brief
Interface          IP-Address   OK? Method Status                Protocol
GigabitEthernet0/0  unassigned   YES unset  administratively down  down
GigabitEthernet0/1  unassigned   YES unset  administratively down  down
GigabitEthernet0/2  unassigned   YES unset  administratively down  down
GigabitEthernet0/3  unassigned   YES unset  administratively down  down
```

copy

R1's G0/0 and G0/1 are both configured and enabled. After configuration, it's always a good idea to verify that the configurations are correct. In the example below I once again used the show ip interface brief command to verify.

```
R1# show ip interface brief Interface IP-Address OK? Method Status Protocol
GigabitEthernet0/0 192.168.1.1 YES manual up up GigabitEthernet0/1 192.168.2.1 YES
manual up up GigabitEthernet0/2 unassigned YES unset administratively down down
GigabitEthernet0/3 unassigned YES unset administratively down down
```

Notice that G0/0 and G0/1 both have the correct IP addresses, and are up in both the Status and Protocol columns. However, show ip interface brief doesn't display the netmask. To double-check that the netmask is correct you can use the **show ip interface** [*interface-name*] command, as in the example below. Notice that, although you must use a netmask when configuring IP addresses, the prefix length is displayed as */X* in the output of this command. This command shows a lot of output, so I am only including the first few lines.

Configuration

Note

```
R1# configure terminal                              #1
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# interface gigabitethernet0/0            #2
```

R1(config-if)# #3

#1 Accesses global configuration mode

#2 Accesses interface configuration mode for G0/0

#3 The prompt changes to R1(config-if)#.

copy

In addition to the prefix length, there are two more things I would like to point out about the above output, related to other topics covered in this chapter: First, the line Broadcast address is 255.255.255.255 indicates the IP address that will be used to send a message to all hosts in the local network: 255.255.255.255. This is a specially reserved IP address for broadcast packets. If R1 wants to send a message to all hosts in LAN 1, it will send a packet addressed to 255.255.255.255 out of its G0/0 interface (encapsulated in a frame addressed to the MAC address ffff.ffff.ffff).

Note

Note

After verifying that the configurations are correct, it’s always a good idea to save the configuration with one of the commands covered in chapter 5: write, write memory, or copy running-config startup-config.

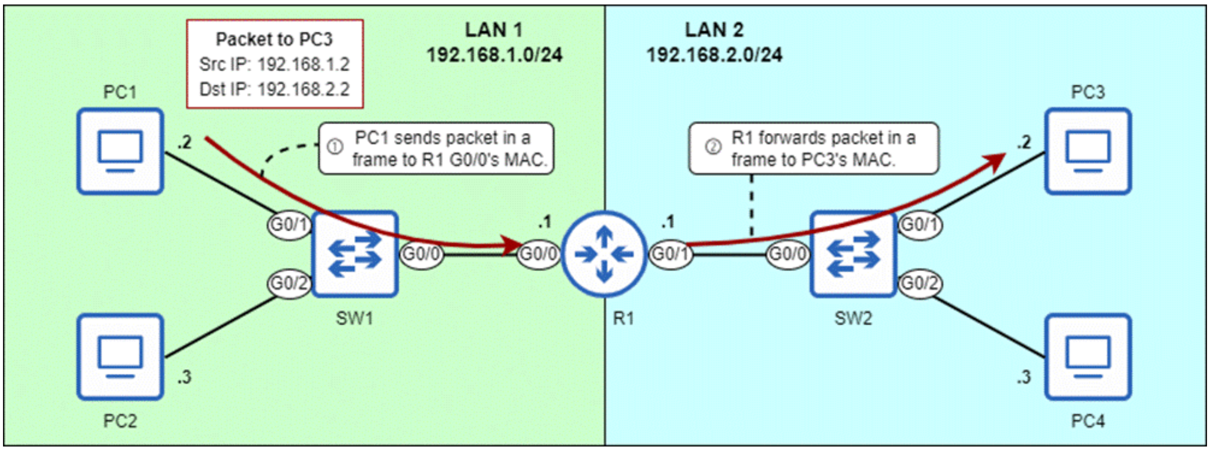
1

R1(config-if)# **ip address 192.168.1.1 255.255.255.0**

1

copy

Figure 7.10 PC1 sends a packet to PC3 via R1. The packet is addressed to PC3’s IP address. 1) PC1 sends the packet in a frame addressed to the MAC address of R1’s G0/0 interface. 2) R1 forwards the packet in a frame addressed to the MAC address of PC3. In order for R1 to serve its purpose of connecting the two networks together, it must have an appropriate IP address on each of its interfaces, which we configured in this section.



```
R1(config-if)# no shutdown                    #1
%LINK-3-UPDOWN: Interface GigabitEthernet0/0,
 changed state to up                          #2
%LINEPROTO-5-UPDOWN: Line protocol on Interface
 GigabitEthernet0/0, changed state to up      #2
```

#1 Enables the interface

#2 Messages indicate that the interface has been enabled.

copy

Note

For a router to forward packets to *remote* networks (that aren't directly connected to the router itself), additional configurations are required; we will cover those configurations in later chapters. However, in the example network shown in figure 7.10, LAN 1 and LAN 2 are both directly connected to R1 - no more configurations are needed. PC1 and PC2 in LAN 1 can now communicate with PC3 and PC4 in LAN 2 via R1.

7.3.3 Attributes of an IPv4 network

```
R1(config-if)# interface g0/1                        #1
R1(config-if)# ip address 192.168.2.1 255.255.255.0   #2
R1(config-if)# no shutdown                            #2
%LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state
to up
```

#1 Enters interface configuration mode for G0/1

#2 Configures G0/1's IP address and enables it

copy

Note

The *network address* is the first address of any network, and it is used to identify the network; it cannot be assigned to a host. An IPv4 address is a network address if all bits of its host portion are set to "0". Figure 7.11 shows an example of a network address: 192.168.100.0/24.

Final verification

Broadcast address

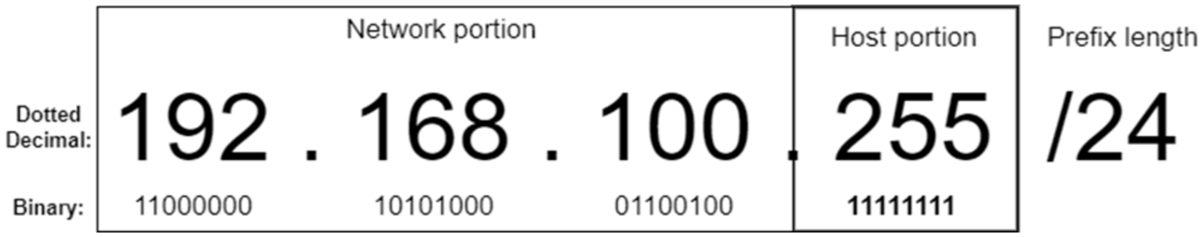
```
R1# show ip interface brief
Interface          IP-Address   OK? Method Status            Protocol
GigabitEthernet0/0  192.168.1.1  YES manual up                  up     #1
```

```
GigabitEthernet0/1  192.168.2.1  YES manual up                    up      #1
GigabitEthernet0/2  unassigned   YES unset  administratively down  down
GigabitEthernet0/3  unassigned   YES unset  administratively down  down
```

#1 G0/0 and G0/1 each have an IP address and are up/up.

copy

Figure 7.12 192.168.100.255 is a broadcast address, as indicated by the host portion of 11111111. This address can be used to address a message to all hosts in the 192.168.100.0/24 network.



```
R1# show ip interface                          #1
GigabitEthernet0/0 is up, line protocol is up       #2
  Internet address is 192.168.1.1/24              #2
  Broadcast address is 255.255.255.255            #2
  Address determined by setup command             #2
  MTU is 1500 bytes                          #2
. . .
GigabitEthernet0/1 is up, line protocol is up       #3
  Internet address is 192.168.2.1/24              #3
  Broadcast address is 255.255.255.255            #3
  Address determined by setup command             #3
  MTU is 1500 bytes                          #3
. . .
```

#1 Views more detailed information about R1's interfaces

#2 Information about G0/0

#3 Information about G0/1

copy

Note

Maximum number of hosts

The *maximum number of hosts* in a network is the number of IP addresses available to assign to hosts connected to the network. To calculate the total number of IP addresses in a network, the formula is 2^y, where *y* is the number of host bits. For example, with a /24 prefix length there are eight host bits; 2^8 is equal to 256, so there are 256 total IP addresses in a /24 network (such as 192.168.100.0/24).

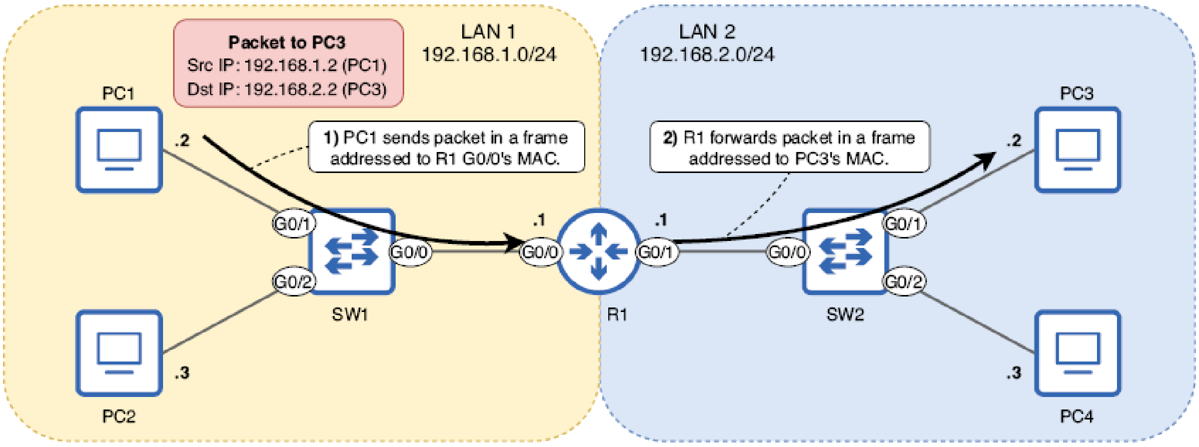
However, because the *network* and *broadcast* addresses of each network can't be assigned to hosts, we have to subtract two from the total number of addresses in the network to find the maximum number of hosts. Therefore the formula to determine the maximum number of hosts in a network is actually 2^y-2. Therefore the maximum number of hosts of a /24 network is 254 (2^8-2). Below are the maximum number of hosts in networks with /8, /16, and /24 prefix lengths.

Note

/16: 2^{16}-2 = 65,534 hosts

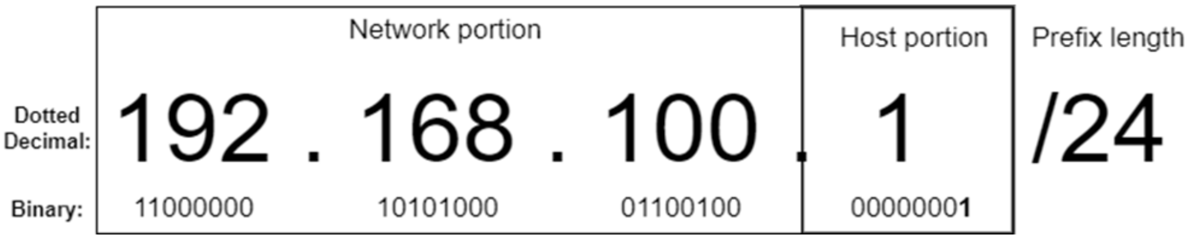
/24: 2^{24}-2 = 16,777,214 hosts

Figure 7.11 PC1 sends a packet to PC3 via R1. The packet is addressed to PC3's IP address. (1) PC1 sends the packet in a frame addressed to the MAC address of R1's G0/0 interface. (2) R1 forwards the packet in a frame addressed to the MAC address of PC3. For R1 to serve its purpose of connecting the two networks together, it must have an appropriate IP address on each of its interfaces, which we configured in this section.


Packet to PC3
Src IP: 192.168.1.2 (PC1)
Dst IP: 192.168.2.2 (PC3)
LAN 1
192.168.1.0/24
LAN 2
192.168.2.0/24
PC1
PC2
PC3
PC4
.2
.3
.2
.3
1) PC1 sends packet in a frame addressed to R1 G0/0's MAC.
2) R1 forwards packet in a frame addressed to PC3's MAC.
G0/1
G0/2
G0/0
SW1
.1
G0/0
R1
.1
G0/1
G0/0
G0/1
G0/2
SW2

Note

Figure 7.13 192.168.100.1 is the first usable address of the 192.168.100.0/24 network. It is the first address after the network address.



Note

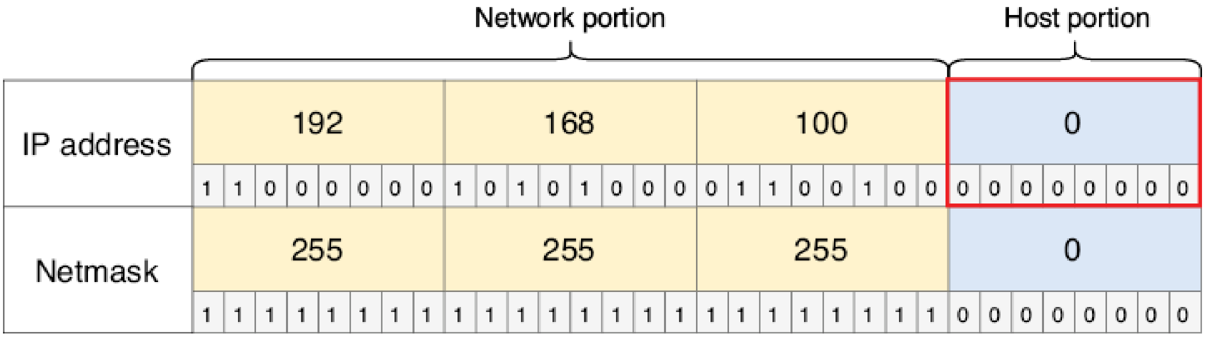
7.3.3 Attributes of an IPv4 network

The *last usable address* of a network is the last IP address that can be assigned to a host; it's the last IP address before the broadcast address. This address is also simple to find - subtract one from the broadcast address (change the least-significant bit to 0). Figure 7.14 shows the last usable address of the 192.168.100.0/24 network.

Network address

If you know the first and last usable addresses, you know the range of usable addresses: from the first usable address to the last usable address. For example, the range of usable addresses in the 192.168.100.0/24 network is from 192.168.100.1 to 192.168.100.254: 254 addresses in total.

Figure 7.12 192.168.100.0 is a network address, as indicated by the host portion of 00000000. This address is used to identify the 192.168.100.0/24 network as a whole and cannot be assigned to a host. 192.168.100.100 (used in figure 7.7) is a host address in the 192.168.100.0/24 network.

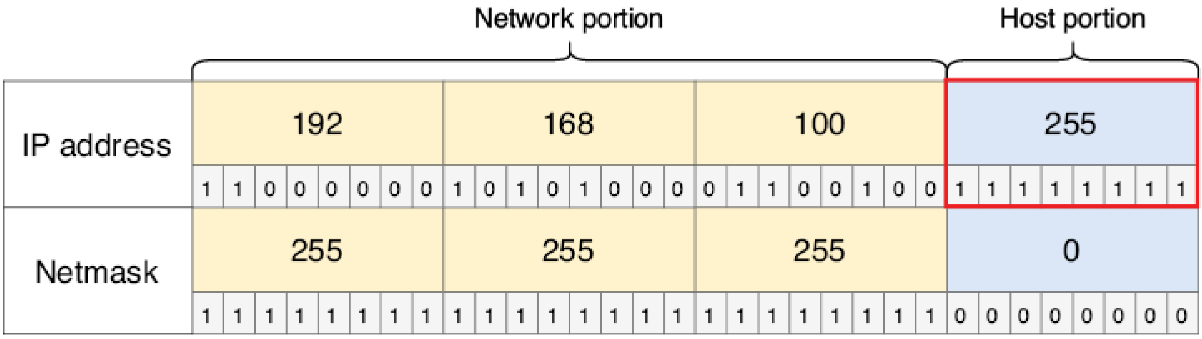


	Network portion			Host portion
IP address	192	168	100	0
	1 1 0 0 0 0 0 0	1 0 1 0 1 0 0 0	0 1 1 0 0 1 0 0	0 0 0 0 0 0 0 0
Netmask	255	255	255	0
	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0

Broadcast address

Q: PC1's IP address is 172.20.20.127/16. What is the usable address range of the network PC1 belongs to?

Figure 7.13 192.168.100.255 is a broadcast address, as indicated by the host portion of 11111111. This address can be used to address a message to all hosts in the 192.168.100.0/24 network.



	Network portion			Host portion
IP address	192	168	100	255
	1 1 0 0 0 0 0 0	1 0 1 0 1 0 0 0	0 1 1 0 0 1 0 0	1 1 1 1 1 1 1 1
Netmask	255	255	255	0
	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0

Note

172.20.0.1 - 172.20.255.254

Maximum number of hosts

To find the usable address range of a network, you need to know the first and last usable addresses of the network. This is fairly simple when using a prefix length of /8, /16, or /24; the division between the network portion and host portion is between octets (in this case between the second and third octets, because the prefix length is /16).

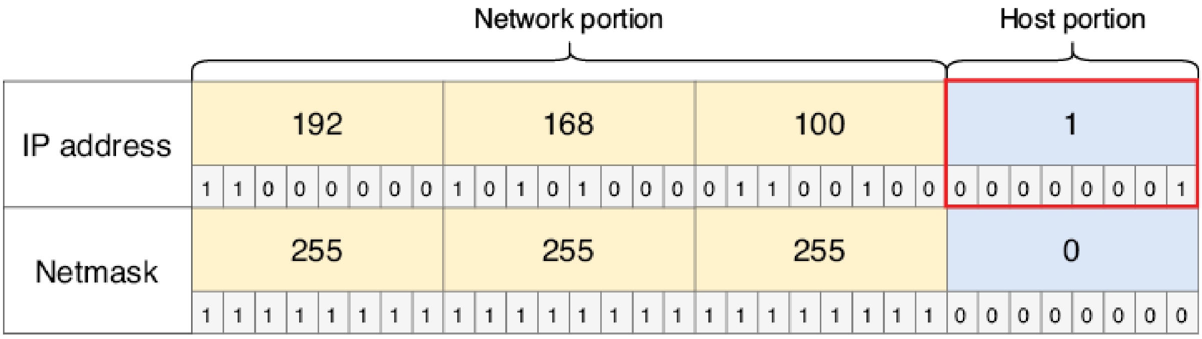
To find the first usable address, simply change the octet(s) of the host portion to 0 (this is the network address), and then add 1 to the last octet: PC1's address is 172.20.20.127, the network address is 172.20.0.0, and the first usable address is 172.20.0.1.

- To find the first usable address, change the octet(s) of the host portion to 255 (this is the broadcast address), and then subtract 1 from the last octet: PC1's address is 172.20.20.127, the broadcast address is 172.20.255.255, and the last usable address is 172.20.255.254. Now you know the usable address range: from 172.20.0.1 to 172.20.255.254. Therefore the answer to this question is C.
- This process will become more challenging when we cover *subnetting* in chapter 11 of this book. When subnetting, we use prefix lengths that do not fit neatly between octets of an IP address, such as /19, /23, /28, etc. In that case it is important to be proficient at converting between decimal and binary so you can identify the network and host bits, convert the host bits to 0 or 1 as necessary, convert them back to decimal, etc.
- **7.3.4 IPv4 address classes**

First and last usable addresses

Note

Figure 7.14 192.168.100.1 is the first usable address of the 192.168.100.0/24 network. It is the first address after the network address.



	Network portion			Host portion
IP address	192	168	100	1
	1 1 0 0 0 0 0 0	1 0 1 0 1 0 0 0	0 1 1 0 0 1 0 0	0 0 0 0 0 0 0 1
Netmask	255	255	255	0
	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0

Note

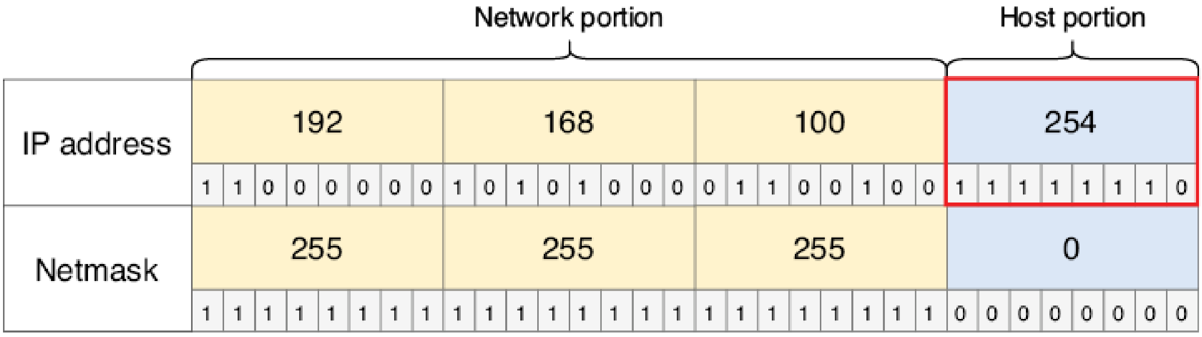
Table 7.1 IPv4 address classes

Class	First octet bit pattern	First octet decimal range	Prefix length	Note
A	0xxxxxxx	0 - 127	/8	Address range: 0.0.0.0 - 127.255.255.255

B	10xxxxxx	128 - 191	/16	Address range: 128.0.0.0 - 191.255.255.255
C	110xxxxx	192 - 223	/24	Address range: 192.0.0.0 to 223.255.255.255
D	1110xxxx	224 - 239		Reserved for multicast addresses
E	1111xxxx	240 - 255		Reserved for experimental purposes

The class of an IPv4 address is determined by the first one to four bits of the address; *class A* addresses begin with "0", *class B* addresses begin with "10", *class C* addresses begin with "110", *class D* addresses begin with "1110", and *class E* addresses begin with "1111". Classes A, B, and C are the ranges from which hosts are assigned IPv4 addresses. For example, the IP addresses we configured on R1 in this chapter are from the class C range. Classes D and E are reserved for particular purposes; we won't cover them in this book, except for a few mentions of multicast IP addresses (class D).

Figure 7.15 192.168.100.254 is the last usable address of the 192.168.100.0/24 network. It is the last address before the broadcast address.



Some addresses in each class are reserved for special purposes and can't be assigned to hosts. For example, class A addresses with a first octet of *0* or *127* are reserved.

Exam scenario

Few class A networks exist (128), but each class A network contains many addresses (16,777,216).

Class B networks are a middle ground. There are 16,384 class B networks, each containing 65,536 addresses.

A. 172.20.20.1–172.20.20.254

B. 172.20.20.0–172.20.20.255

C. 172.20.0.1–172.20.255.254

D. 172.20.0.0–172.20.255.255

Table 7.2 summarizes the characteristics of classes A, B, and C. You don’t have to memorize the number of networks and addresses per network for each address class; just understand that a smaller network portion means fewer networks with more hosts in each network, and a larger network portion means more networks with fewer hosts in each network.

Table 7.2 Characteristics of classes A, B, and C

Class	First octet	Size of network portion	Size of host portion	Number of networks	Addresses per network
A	0xxxxxxx	8 bits	24 bits	128 (27)	16,777,216 (224)
B	10xxxxxx	16 bits	16 bits	16,384 (214)	65,536 (216)
C	110xxxxx	24 bits	8 bits	2,097,152 (221)	256 (28)

Note

The reason there are only 2^{7} class A networks even though the network portion is 8 bits in length is that the first bit is fixed as “0” - only seven bits are available to change and make different networks. The same reasoning applies for why there are 2^{14} class B networks (not 2^{16}) and 2^{21} class C networks (not 2^{24}).

7.3.4 IPv4 address classes

7.4 Summary

Note

The *Version* field indicates the version of IP (IPv4 or IPv6).

The *Internet Header Length* (IHL) field indicates the length of the header in four-byte increments.

Table 7.2 IPv4 address classes (view table figure)

Class	**First octet bit pattern**	**First octet decimal range**	**Prefix length**	**Note**
A	0xxxxxxx	0–127	/8	Address range: 0.0.0.0–127.255.255.255
B	10xxxxxx	128–191	/16	Address range: 128.0.0.0–191.255.255.255
C	110xxxxx	192–223	/24	Address range: 192.0.0.0 to 223.255.255.255
D	1110xxxx	224–239		Reserved for multicast addresses

Class	First octet bit pattern	First octet decimal range	Prefix length	Note
E	1111xxxx	240–255		Reserved for experimental purposes

The *Total Length* field indicates the length of the entire packet in bytes.

Note

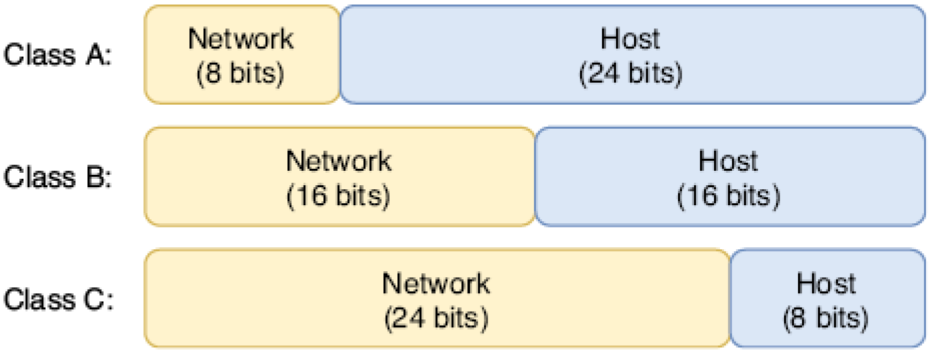
The *Time To Live* (TTL) field is used to prevent packets from looping indefinitely around the network. Each time a router forwards a packet its TTL is decremented by 1, and if it reaches 0 the packet is dropped.

The *Protocol* field indicates the type of message encapsulated inside of the packet, such as ICMP, TCP, UDP, or OSPF.

- The *Header Checksum* field is used to check for errors in the IPv4 header.
- The *Source Address* field contains the IPv4 address of the host that sent the packet.
- The *Destination Address* field contains the IPv4 address of the packet's intended recipient.

The *Options* field is optional and variable in length - from 0 bytes (if not used) to a maximum of 40 bytes in length. This field is rarely used.

Figure 7.16 The network portion and host portion sizes of class A, class B, and class C IPv4 addresses



The binary number system uses two digits: 0 and 1. It is also called *base 2*. The value of each digit position increases twofold: 1, 2, 4, 8, 16, 32, 64, 128, etc.

An eight-bit binary number provides 256 possible values: from 0d0 (00000000) to 0d255 (11111111).

Table 7.3 Characteristics of classes A, B, and C [(view table figure)]

Class	First octet	Size of network portion	Size of host portion	Number of networks	Addresses per network
A	0xxxxxxx	8 bits	24 bits	128 (2^7)	16,777,216 (2^{24})

Class	First octet	Size of network portion	Size of host portion	Number of networks	Addresses per network
B	10xxxxxx	16 bits	16 bits	16,384 (2^{14})	65,536 (2^{16})
C	110xxxxx	24 bits	8 bits	2,097,152 (2^{21})	256 (2^{8})

Note

IPv4 addresses are divided into two parts: the *network portion* and the *host portion*. All hosts within a LAN will have the same network portion, but a unique host portion.

The size of the network portion can be indicated with a *prefix length* in the format */X*, where *X* is the number of bits in the network portion. Any bits that are not part of the network portion are part of the host portion.

Reserved addresses

A “1” in the netmask means the bit in the same position of the IP address is part of the network portion. A “0” in the netmask means the bit in the same position of the IP address is part of the host portion.

- The show ip interface brief command lists a router’s interface and information about their IP addresses and status.
- The show ip interface [*interface-name*] command shows more detail about each interface.

Summary

- The IPv4 header is 20 to 60 bytes in length and contains 14 fields.
- The *Version* field indicates the version of IP (IPv4 or IPv6).
- The *Internet Header Length* (IHL) field indicates the length of the header in 4-byte increments.
- The *Differentiated Services Code Point* (DSCP) and *Explicit Congestion Notification* (ECN) fields are used to prioritize certain kinds of traffic. This is called *Quality of Service* (QoS).
- The *Total Length* field indicates the length of the entire packet in bytes.
- The *Identification*, *Flags*, and *Fragment Offset* fields support packet fragmentation. If a packet is larger than an interface’s *Maximum Transmission Unit* (MTU), the router will divide the packet into multiple smaller packets called *fragments*. The standard MTU is 1500 bytes.
- The *Time To Live* (TTL) field is used to prevent packets from looping indefinitely around the network. Each time a router forwards a packet, its TTL is decremented by 1, and if it reaches 0, the packet is dropped.
- The *Protocol* field indicates the type of message encapsulated inside of the packet, such as ICMP, TCP, UDP, or OSPF.

- The *Header Checksum* field is used to check for errors in the IPv4 header.
- The *Source Address* field contains the IPv4 address of the host that sent the packet.
- The *Destination Address* field contains the IPv4 address of the packet's intended recipient.
- The *Options* field is optional and variable in length—from 0 bytes (if not used) to a maximum of 40 bytes in length. This field is rarely used.
- The decimal number system uses 10 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. It is also called *base 10*. The value of each digit position increases tenfold: 1, 10, 100, 1000, etc.
- The binary number system uses two digits: 0 and 1. It is also called *base 2*. The value of each digit position increases twofold: 1, 2, 4, 8, 16, 32, 64, 128, etc.
- An 8-bit binary number provides 256 possible values: from 0d0 (00000000) to 0d255 (11111111).
- For the CCNA exam, you must be able to convert between binary and decimal for numbers of up to 8 bits in length. You can practice at https://learningnetwork.cisco.com/s/binary-game.
- An IPv4 address is a 32-bit number that identifies a host at Layer 3. It is divided into four groups of 8 bits called *octets* and written in *dotted decimal notation*.
- IPv4 addresses are divided into two parts: the *network portion* and the *host portion*. All hosts within a LAN will have the same network portion but a unique host portion.
- The size of the network portion can be indicated with a *prefix length* in the format /X, where X is the number of bits in the network portion. Any bits that are not part of the network portion are part of the host portion.
- The size of the network portion can also be indicated with a *netmask* (also called a *subnet mask*). A netmask is a string of 32 bits that is paired with an IP address to indicate which bits of the IP address are the network portion and which are the host portion.
- A 1 in the netmask means the bit in the same position as the IP address is part of the network portion. A 0 in the netmask means the bit in the same position as the IP address is part of the host portion.
- The **show ip interface brief** command lists a router's interface and information about their IP addresses and status.
- The **show ip interface** *[interface-name]* command shows more detail about each interface.
- Router interfaces are disabled by default and must be enabled with the **no shutdown** command.
- Interface configuration mode can be accessed with the **interface** *interface-name* command from global configuration mode.

- An interface's IPv4 address can be configured with the **ip address** *ip-address netmask* command in interface configuration mode.
- The network address of a network is the first address of the network, with a host portion of all 0s. It is used to identify the network and cannot be assigned to a host.
- The broadcast address of a network is the last address of the network, with a host portion of all 1s. It can be used to send a message to all hosts in the network. However, to send a message to all hosts on the local network, the address 255.255.255.255 is usually used.
- The maximum number of hosts of a network is the number of IP addresses that can be assigned to hosts. The formula is $2^y - 2$, where y is the number of bits in the host portion. Two is subtracted for the network and broadcast addresses.
- The *first usable address* of a network is the first address that can be assigned to a host. The *last usable address* is the last address that can be assigned to a host.
- IPv4 addresses can be organized into five classes: A, B, C, D, and E. Class D is reserved for multicast addresses, and Class E is reserved for experimental purposes. Addresses from classes A, B, and C are assigned to network hosts.
- Class A addresses have a first octet of 0–127 and use a /8 prefix length. Class B addresses have a first octet of 128–191 and use a /16 prefix length. Class C addresses have a first octet of 192–223 and use a /24 prefix length.
- Networks that follow class A, B, and C rules are called classful networks. This system is now obsolete and has been replaced with classless networking, which is more flexible.