Table 7-1 Binary Octets and Their Decimal Equivalents

Binary Code	"On" Bit Value	Decimal Value
00000000	n/a	0
00000001	1	1
00000011	2+1	3
00000111	4+2+1	7
00001111	8+4+2+1	15
00011111	16+8+4+2+1	31
00111111	32+16+8+4+2+1	63
01111111	64+32+16+8+4+2+1	127
11111111	128+64+32+16+8+4+2+1	255

As you can see, if all the bits are set to on, the decimal equivalent is 255. That's as high a number as you'll see in any octet. Conversely, if all the bits are set to off, the decimal equivalent is 0. Therefore, the range of numbers in any given octet in an IP address (or a subnet mask or default gateway; more on that later) is 0–255. But before we do that, let's delve further into the IP address.

TCP/IP's Big Three

When configuring TCP/IP, the first order of business is to assign an IP address and a subnet mask. There are no exceptions to this rule when setting up TCP/IP parameters, and in a minute you'll learn why. There is also another parameter, the default gateway, that's essential for communicating on larger networks, such as the Internet. You can think of each of these parameters as TCP/IP's Big Three, and they each get special attention in the following sections.

The IP Address

This is the number associated with your computer, analogous to the address of a house or a number of a telephone line. Every IP address is composed of two parts: the network ID and the host ID.

The network ID represents the network the computer belongs to, while the host ID uniquely identifies the computer on that network. Every network ID on a TCP/IP network (such as the Internet) must be unique, in the same way that every zip code in the United States must be unique. Further, every host ID on each network must also be unique to that network, in the same way that every house on a street must have a different number. If you accept the foregoing as true, then you must also posit that there must be a way to separate the IP address into a network portion and a host portion. But how does IP do this? How does the computer know which part of the IP address is the network identifier and what part is the host identifier? Read on.

The Subnet Mask

The subnet mask is used to divide the IP address into the network portion and the host portion. It is a required TCP/IP setting. Without the subnet mask, the computer has no idea what network it belongs to. The subnet mask is also a 32-bit binary number broken into four octets for easy human consumption, like this: 255.255.0.0.

The interesting thing to note about the subnet mask is how it looks to the computer, as shown in Figure 7-5.

Notice anything peculiar about the binary number shown in Figure 7-5? That's right; it's a string of *contiguous* 1s followed by another string of *contiguous* 0s. The contiguous nature of the binary notation makes the subnet mask easy to spot.

Note also that the decimal number 255 octet is a string of 8 binary 1s. The decimal 0 octet is a string of 8 binary 0s. The job of the 1s, then, is to mask out the network number—they identify which of the 32 1s and 0s in the IP address are used as the network ID. Everything else—everything that lines up with the 0s in the subnet mask—denotes the host ID.

If you use the IP address with the subnet mask here, you are able to determine which 1s and 0s of the IP address are the network ID, and which 1s and 0s are the host ID. The subnet mask tells you that the first 16 binary numbers are the network number, which when you convert back to decimal is 192.168. The next 16 binary numbers are the host ID, which translate into 2.200.

If any of this is confusing, think back to the example of the telephone protocol. Part of the telephone number you dial represents a big grouping of phone lines (the first three numbers), and part represents an individual line within that larger grouping (the last four numbers). You can think of this as the network ID and the host ID in IP communications.

So why is this determination of network ID and host ID important? Why is a subnet mask required for a valid IP address? It is needed for the successful delivery of information to the proper network, and it helps make TCP/IP a routable protocol.

Each and every packet of TCP/IP communications has a source address and a destination address. And once TCP/IP determines the network ID of the source and destination computers, an important decision is made about how to deliver the packet to its destination. If the network IDs match, the packet is delivered to the local segment of computers. But if the network IDs of source and destination do not match, the packet must be routed to a remote network via a default gateway.

The Default Gateway

Here's another key ingredient in IP configuration, which is why it's included in the Big Three. Although it is not technically required, your network communication would be very limited without the default gateway parameter.

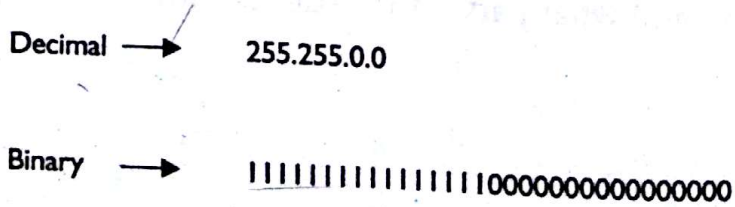
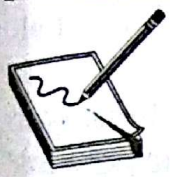


Figure 7-5 The subnet mask displayed in decimal and in binary

A default gateway is the IP address of the router, which is the pathway to any and all remote networks. (Why the default gateway isn't called the default router, especially since there are devices called gateways whose purpose differs from routers, is a matter of late-night conjecture.) To get a packet of information from one network to another, the packet is sent to the IP address of the default gateway, which then helps forward the packet to its destination network. In fact, computers that live on the other side of routers are said to be on remote networks. Without default gateways, Internet communication is not possible, because your computer doesn't have a way to send a packet destined for any other network.



NOTE We tend to think of a router as a big box with the letters C-I-S-C-O etched on the front, but it can be any device that simply passes information from one network to another. Almost any Windows machine with multiple network cards can be configured to act as a router, as you'll learn in Chapter 9.

Address Classes

You should also know when dealing with IP addresses that they are broken up in to different categories or classes. Each of these categories help define a range of networks in each class, and also help define how many hosts can exist on every network. Also, these address classes have a default subnet mask associated with them, which actually does the dirty work of defining the number of networks and the number of hosts per network.

The different classes are defined by the first few bits in the first octet of the IP address, as you will see. Keep in mind that these IP addresses are seen only as 1s and 0s, and this concept should fall into place.

Let's look at each of these address classes now.

Class A Addresses

The first bit of the first octet in a Class A IP address is always set to 0. If you refer to the binary-to-decimal chart in Table 7-1, you can see this puts a ceiling on the decimal equivalent of that first octet. If the first bit is set to off (0), the highest number you can make in the first octet is reached by turning all the other seven bits to on (1). When you do, the decimal equivalent is 127, which represents the upper range of a Class A address. Actually, it's 126, because the 127 network is a reserved address, as will be explained next.

A Class A network has a default subnet mask of 255.0.0.0. That means that, by default, the first octet of a Class A address is the network identifier, and the last three octets will be the host identifier. Because of this, there are only 126 Class A networks available for use, with a range in the first octet from 1–126 (the 0 network cannot be used; that, too will be explained below).

To test this, open your TCP/IP Properties page (right-click My Network Places, select Properties | TCP/IP, right-click and choose Properties). Enter a Class A address and then tab down to the subnet mask. The mask that Windows Server 2003 offers you is a Class A mask of 255.0.0.0.

So how many hosts can you create on a Class A network? Well, since the subnet mask tells you the first 8 bits of your 32 total bits will define the network ID, that means you can mix and match the rest of the 24 bits in the remaining three octets to define the hosts in a given Class A network. If you have 24 bits left to define hosts, there are 2^{24} such combinations of 1s and 0s. To save you a trip to the calculator, 2^{24} comes out to 16,777,214 hosts per network. (2^{24} is actually 16,777,216, but you can't use the first and last host IDs. Again, this will be explained in the following section.)

Class B Addresses

In a Class B address, the first 2 bits of the first octet are always set to 10. This means that the first octet's decimal value will always be at least 128. And if the second bit will always be set to 0, the highest the first octet will be is 191.

The Class B addresses have a default subnet mask of 255.255.0.0. These addresses are assigned to medium to relatively large companies and organizations, with the network numbers ranging from 128.0.*x*.*y* to 191.255.*x*.*y*, where *x* and *y* are the host values.

There are 2^{14} Class B networks (remember that the two bits are locked), or 16,384. Since you have 16 bits to mix and match to define the hosts on each of these networks, you end up with a possibility of 2^{16} or 65,534 hosts (when you remember to subtract 2).

Class C Addresses

In a class C address, the first 3 bits of the first octet are always set to 110. Those three bits are set in stone. That means the smallest decimal number the first octet can be is 192. And since the highest binary you can produce on that octet is 11011111, the upper range for the first octet of a Class C address is 223.

Class C addresses have a default subnet mask of 255.255.255.0, and are assigned to small companies, who sometimes get several Class C addresses at a time. The network numbers range from 192.0.0.*x* to 223.255.255.*x*, where *x* is the host value for the network.

There are 2^{21} possible Class C network addresses, or 2,097,152. Each Class C address has one octet's worth of hosts, which is 2^{8} less 2, or 254.

These address class differences are summarized for your review in Table 7-2.

Address Class	Default Subnet Mask	Range	Number of Networks	Hosts per Network
A	255.0.0.0	1.0.0.0–126.0.0.0	126	Approximately 16 million
B	255.255.0.0	128.0.0.0–191.255.0.0	16 large	64 large
C	255.255.255.0	192.0.0.0–223.255.255.0	2 million	254

Table 7-2 The Address Class Differences

The AND Operation

When a computer sends a packet of IP traffic to another computer, there must be a way to evaluate that packet and determine if the packet will be delivered to the local network, or to a remote network, and thus to the default gateway. This evaluation is performed using an AND operation.

It works like this: when a computer builds an IP packet for delivery, that packet will contain both a source IP address and subnet mask, as well as a destination IP address. Just like when you wrap and address a mail package, you include the address of where the package is going and an address of where it's being sent from.

Once the packet is composed, the computer will then AND the subnet mask to both source and destination IP addresses. ANDing combines the IP address and subnet mask, spawning a third number as a result. Where there are 1s and 1s, a 1 is produced once the AND operator does its evaluation. Where there is a 0 and any other number, a 0 results.

It's best to look at an example of the AND operator at work for full understanding. Suppose your system, whose IP address was 192.168.2.200 and subnet mask was 255.255.255.0, composed a TCP/IP packet whose destination IP address was 192.168.8.100. That information would be part of the header information in the packet.

Now, your network card needs to know where that packet will be sent. Will it be delivered locally or remotely? Here's where the AND operator steps in.

When the computer takes the source IP address and ANDs it to the subnet mask, the result is a binary number (see Figure 7-6). Note that this number matches the network identifier, which you have already determined by looking at the subnet mask.

The destination IP address is ANDed to the system's subnet mask (see Figure 7-7). The results of the source ANDing and the destination ANDing are then compared. If the number is a match, the packet is sent only to the local network. In our example, however, the ANDing results do not match, and therefore the computer determines that the packet needs to be routed to another network. It is then sent to the computer's default gateway for routing.

When it's all said and done, ANDing is just the computer's way of determining the source and destination network numbers, and then forwarding the packet as necessary based on that determination. The computer just uses binary when evaluating these numbers.

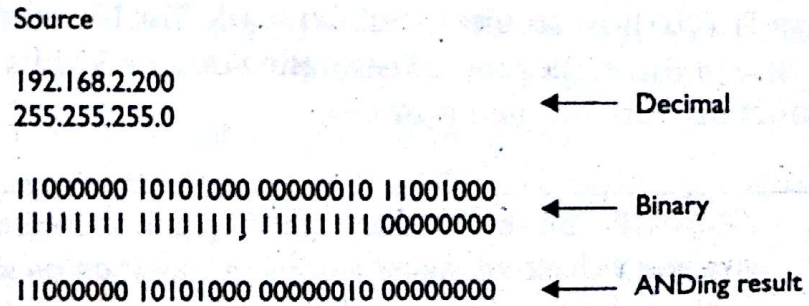


Figure 7-6 The result of the source ANDing process

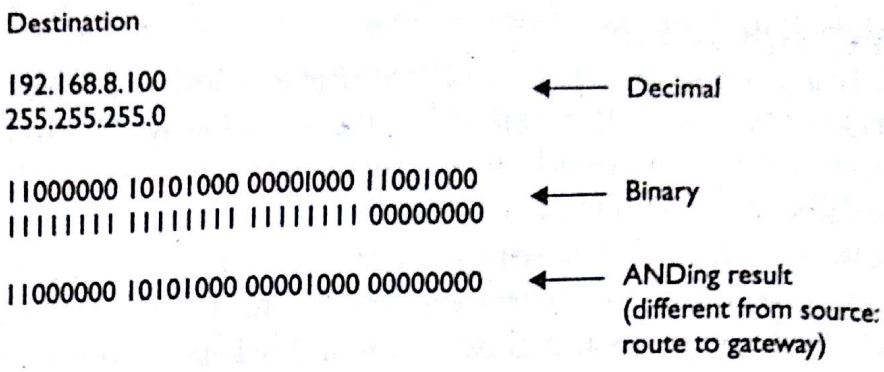


Figure 7-7 Result of the destination ANDing process

Now let's talk more about the IP addressing rules referred to in the preceding sections.

IP Addressing Rules

There are a few rules you should be aware of when assigning IP addresses to computers:

- The network ID cannot be set to 127. This address is reserved for loopback and diagnostic purposes.
- The network ID and host IDs cannot be all 1s. If all bits are set to 1, the address is interpreted as a broadcast address and will not be routed under normal circumstances. A Class B address of 151.255.0.0 is a perfectly good network ID. The network ID is 16 bits long, and only the last 8 bits in the network number are all 1s.
- Neither the network ID nor the host ID can have the bits be all 0s. If all of the host bits are set to 0, for example, that is interpreted as the same as the network number. In the example above, the network number is 151.255.0.0. The first host on that network, then, is 151.255.0.1.
- The host ID must be unique to the local network ID. There cannot be two 0.1 hosts on the 151.255 network.
- A unique network ID is needed for each network connected to a wide area network, which is usually the public Internet. (This network ID is typically provided by your ISP.) On the Internet, for example, there is only one 151.255 network.
- Every TCP/IP host requires a subnet mask. The IP address is useless without one because of the mask's role of separating out the 32 bits of the IP address into distinct network bits and host bits.

Classless Internet Domain Routing (CIDR) Notation

There is another way of expressing the IP address without having to spell out the subnet mask in octets of 255s and 0s. Instead, you can count out the number of 1s in the subnet mask and then represent the IP address and subnet mask with a slash followed by the number of 1 bits in the subnet mask, like 192.168.2.200/16.

This notation is called classless inter-domain routing (CIDR, pronounced "cedar") notation. The notation here specifies the same IP address as used in the previous example. The IP address part is very straightforward. The /16 tells you that this IP address has a subnet mask with the first 16 bits set to 1. And 16 1s gives you a subnet mask of 255.255.0.0, same as before.

CIDR notation is used to simplify entries on routing tables and is the notation used by a majority of backbone Internet routers. It is also used to cut down on the number of wasted IP addresses in the Class B address space. You should know how to recognize the CIDR notation when you see it and to convert the CIDR subnet mask notation into a decimal equivalent.

3. From the General tab of the TCP/IP Properties dialog box, click the Obtain An IP Address Automatically option.
4. On the Alternate Configuration tab, click the User Configured option, and then type values for the following TCP/IP parameters:
   - IP address
   - Subnet mask
   - Default gateway
   - Preferred and alternate DNS server
   - Preferred and alternate WINS server

Subnetting and Supernetting

If you stopped after reading about the different classes of IP addresses, you might get the impression that subnet masks will always be a set of 255s followed by a set of 0s. But remember that the rule of subnet masks deals with how these values are expressed in *binary* notation: a string of contiguous 1s followed by contiguous 0s. The 255 and 0 values are just decimal equivalents of an entire octet of 1s or 0s. As long as the 1s and 0s are clearly demarcated, other decimal values may be present in the subnet mask.

But you won't see just any decimal value. For example, you won't see the decimal value of 131 in a subnet mask. Why? Because 131 in binary looks like this: 10000011. That doesn't follow the subnet make rule of all 1s followed by all 0s. Table 7-3 shows you the other values you might see in a subnet mask besides 255, along with their binary equivalent.

So why might you see other subnet masks when configuring IP addresses?

Let's say you were given a Class C network number of 192.168.2.0. Recall from earlier that you therefore have 254 possible hosts you could add to that network. But what if you wanted to split that single network into two networks? Into ten networks? Could you? Yes, if you altered the default subnet mask.

The default subnet mask for a Class C network (255.255.255.0) allows for a single network. To make more subnets from this single network, you need to "borrow" bits from the default host portion of the IP address, and mix and match these bits to define additional networks. By borrowing bits from the host portion of the IP address, you increase the number of potential networks you can create. The more bits you borrow, the more possible networks.

Table 7-3 Other Possible Subnet Mask Values

Decimal Value	Binary Equivalent
128	10000000
192	11000000
224	11100000
240	11110000
248	11111000
252	11111100
254	11111110

You can think this subnetting as "sliding" the subnet mask bits to the right. In this example, let's slide the subnet mask 5 bits to the right. In binary, the new make would be

255.255.255.248

or

11111111 11111111 11111111 11111000

if you were to see it as the computer does. You now have masked 5 more bits to define more network numbers. How many networks can you generate? If you recall the shortcut from earlier, that leaves you with 2^5 possible combinations of 5 bits, or 32 possible subnets.

When you borrow 5 bits from your default Class C host portion, you are left with 3 bits you can mix and match to define the individual hosts on the subnets. How many possible combinations are there of 3 bits? If you remember the shortcut from earlier, you know there are 2^3 combinations, for a total of 8, as shown in Table 7-4.

Remember our previously iterated IP addressing rules, however: the first value and the last value cannot be used to define the hosts on a network. The all-0s value and the all-1s value are not valid host numbers.

So with that in mind, here's what the first IP address on the second subnet (the first subnet would be the 192.168.2.0 subnet) would look like when expressed in binary:

11000000 10101000.00000010 00001 001

with the subnet mask of

11111111 11111111 11111111 11111 000

This host's IP address, if you were to look it up using ipconfig, would be 192.168.2.9, with a subnet mask of 255.255.255.248. (Again, the periods show the separation of the network portion of the IP address from the host portion; the computer doesn't do it this way.) But the host, because of that administrator-configured subnet mask, would live on the network of 192.168.2.8. When you use custom subnet masks, the network number is not as easily identifiable as when using the default subnet masks. As you can see, the network number is hidden in the IP address and is only visible when you look at the IP address the way a computer does.

Table 7-4 Potential Combinations of 3 Bits

Binary	Decimal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Recall that you are just as likely to see this expressed in CIDR notation, which in this case would look like this: 192.168.2.9/29.

Supernetting just reverses the direction of the subnet mask "slide." Instead of moving the subnet mask to the right, thus dividing up a given network, supernetting moves the subnet mask to the left, combining many networks into one. This is significant because it can simplify entries on routing tables. Suppose that a single large company was using several consecutive Class C network numbers. Rather than having a routing table entry for each network, supernetting allows the combination of all these networks into a single supernet, and one routing entry can route traffic to all networks.

If this is the first time you've seen this, you will probably have to repeat the process a few times in order to understand it. But once you see it, you see it. It's very much like riding a bike or doing an algebra equation correctly for the first time.

For exam purposes (and this comes in quite handy in the real world), learn the shortcut tricks that quickly tell you how many subnets can be made with a custom subnet mask, and how many hosts can be created on a given subnet. It would not be surprising if you were asked questions that made sure you know how to define custom subnet masks for a given network situation.

For example, you might have a question that asks you to pretend you have a Class B address, and you need to create 95 subnets. That question might ask what the subnet mask should be, and how many hosts you will be able to create per subnet.

How to proceed? Here's one of the shortcuts: convert 95 to binary. You get 1011111. That's 7 bits needed to represent the decimal of 95. You should slide the default Class B subnet mask of 255.255.0.0 (11111111 11111111 0000000 0000000) 7 bits to the right, giving you a new subnet mask of 255.255.254.0 (11111111 11111111 11111110 0000000). And again, remember your CIDR. You could also express this new subnet mask by changing the default /16 mask to the new value of /21.

How many subnets can you create with this new custom subnet mask? 2^7, or 128. That's more subnets than are absolutely necessary, but if you use fewer bits—6, for example—you only can create 2^6, or 64 subnets, which won't meet your requirements.

So, how many hosts per subnet? Add the 1 bit remaining in the third octet with the 8 in the fourth octet, and that gives you a total of 9 bits that the subnet mask leaves unmasked for defining the hosts. 2^9 is 512. Remember to remove the first and last combinations of 9 bits, and you can define 510 hosts for each of your 128 networks created from a single Class B network ID.

Make sense now? It will, and once you do one all by yourself, you'll be able to do 100 without any problem. Again, it helps when you can see the IP address the same way the computer does.

Because Windows Server 2003 systems, along with all computers on the Internet, use TCP/IP to talk to one another, knowledge of IP is essential to administrating any modern network. You'll need to have a firm grasp of IP configuration especially when installing and configuring a DHCP server, because the job of the DHCP server is to automate IP address assignment. So just how do you set up and manage this vital service? Microsoft expects you to know, and the next section gives you the details.