Configuring and Troubleshooting TCP/IP

CERTIFICATION OBJECTIVES

1.01 Configure TCP/IP on Servers and Clients
1.02 Determine Valid IP Addresses
1.03 Configure Routing
1.04 Troubleshoot TCP/IP and Routing
✓ Two-Minute Drill
Q&A Self Test
Chapter 1: Configuring and Troubleshooting TCP/IP

The goal of Exam 70-218 is to test your ability to implement, manage, and troubleshoot an existing Windows 2000 network environment. It's a very broad exam, covering skills and concepts “from the ground up,” so to speak. As such, it helps to understand how certain features are dependent on other features working properly. For example, a successful deployment of an application across multiple computers requires that Active Directory be working properly. A successful Active Directory implementation, in turn, requires a successful DNS installation. And DNS is part of a valid, working network infrastructure.

In this book, we’ll take it from the ground up, starting with basic TCP/IP addressing in this chapter. We’ll look at the “rhyme and reason” behind TCP/IP, as well as specific techniques for setting up, maintaining, and troubleshooting TCP/IP networks. As with all chapters in this book, we’ll focus on the skills and concepts you’re most likely to need to succeed on exam 70-218.

CERTIFICATION OBJECTIVE 1.01

Configure TCP/IP on Servers and Clients

TCP/IP (Transmission Control Protocol/Internet Protocol) is a set of protocols that enable computers to communicate with one another. It has been in use for over 20 years, and is the set of protocols used by the Internet, as well as countless smaller networks.

The protocols were developed by the Internet Engineering Task Force (IETF) using a system based on Requests for Comments (RFCs). The RFC system allows engineers to post technical papers describing new technologies to an electronic bulletin board for review and comments by peers. Today, there are over 3,000 RFCs published on IETF’s web site at www.ietf.org.

A TCP/IP network is composed of hosts. A host, in turn, is any device or service that’s connected to the network. The hosts use a couple of different addresses to identify and communicate with one another, a hardware address and an IP address.

Hardware Addresses

To connect to a network, a host must have a network interface card (NIC) installed. Every NIC that’s manufactured is given a unique 48-bit hardware address. The hardware address is literally “burned into” the card during the manufacturing process, and as a rule cannot be changed by the user. (Actually, some devices do
allow you to change a card’s hardware address, though it’s unlikely you’d ever want
to do this.)

The terms “Ethernet board” and “Ethernet card” are often used as synonyms
for “network interface card.”

Before we go any further, I need to point out that the term “hardware address”
could probably win some kind of award for having the most synonyms on
the planet. While I’ll stick to the term “hardware address” in this book, you may come
across any of the following terms used as a synonym:

- Media Access Control (MAC) address
- Physical address
- Ethernet address
- Token Ring address
- NIC address

As mentioned, the hardware address is a 48-bit number, something along the
lines of 0000000100000000101011011011011100000101101111, although
it’s far more common to see it expressed as six hexadecimal numbers separated by
hyphens or periods, as in 00-80-AD-7B-E0-B7 or 00.80.AD.7B.E0.B7. You can
view a machine’s hardware address by entering the `ipconfig /all` command at a
command prompt. (To get to the command prompt in Windows 2000, click the
Start button and choose Programs | Accessories | Command Prompt.) The hardware
address is next to Physical Address in the display, as in the example shown in
Figure 1-1. (To close the Command Prompt window, type `exit` and press ENTER.)

### IP Addresses

In addition to the hardware address that’s physically burned into each NIC, each
host on a TCP/IP network also has an IP address (sometimes called an Internet address).
Unlike the hardware address, the IP address is a logical address that’s assigned by
a network administrator, or by DHCP (Dynamic Host Configuration Protocol),
which can automatically assign an IP address when the host first connects to the
network. We’ll get into DHCP in detail in Chapter 4. For now, it’s sufficient to
keep in mind that the IP address is flexible in that it can be assigned or changed
at any time.
Each TCP/IP address is a 32-bit number, as in 111111111111111101001101110. You’ll rarely see a TCP/IP address expressed in that binary notation. Instead, you’ll see them expressed in dotted quad format (also called dotted decimal notation), where the address is divided into four octets. Each octet represents 8 bits of the address, and is expressed as a decimal number in the range of 0 to 255. Dots are used to separate the octets, as in the example 192.168.1.1.

Subnet Masks

Every host that has an IP address also has a subnet mask. The name “subnet mask” is a good one, because it “masks” the portion of the IP address that identifies the network to which a host belongs. Like IP addresses, a subnet mask is a 32-bit number. A series of 1’s are used to identify the network portion of the address. The 0’s are used to represent the host portion of the address. For example, Figure 1-2 shows an IP address expressed in binary format (1’s and 0’s). Beneath that is a subnet mask, also expressed in binary. The 1’s “mask off” those digits in the IP address that identify the network as a whole. The 0’s represent the portion of the address that identifies the host.

It’s customary to display the subnet mask in dotted quad format, just as we usually do with IP addresses. The binary octet 11111111, when converted to
decimal, is 255. The binary octet 00000000 is, of course, just 0 in decimal. Thus, we can display an IP address/subnet mask pair in the following more “human readable” format:

192.168.221.204
255.255.255.0

In English, we can say the preceding IP address/subnet mask combination identifies “host number 204 on network number 192.168.221.” However, it would be more correct to say that it identifies host number 204 on the network 192.168.221.0 because host number 0 on a network isn’t really a host at all. Rather, a 0 in the host portion of the address is the address of the network as a whole.

Here’s another way to view it. Think of the network ID as the area code, and the host ID as the specific phone number. But, unlike telephone numbers, where all area codes are three digits, the number of digits used for the area code can vary. The subnet mask “masks” the bits that represent the area code (network ID). The unmasked bits are the telephone number (host ID).

**Getting IP Addresses**

Just as a person’s Social Security Number uniquely identifies them among all the millions of U.S. citizens, an IP address uniquely identifies each host on the Internet. Which perhaps brings up the question, “With millions of IP addresses already taken, how do I know what IP addresses I can use for my network?” The answer to that question is a resounding “It depends.” Every single computer that can access the Internet doesn’t necessarily have its own unique IP address. However, the hosts that serve the Internet—that is, the hosts that can be reached from other computers on the Internet—all do have unique IP addresses. Each of those servers also has a unique fully qualified domain name (FQDN). For example, the FQDN www.microsoft.com uniquely identifies the web site host, www, on the unique domain name microsoft.com.
Chapter 1: Configuring and Troubleshooting TCP/IP

FROM THE CLASSROOM

Who Controls IP Addresses?

When reading about the various agencies involved in doling out globally unique IP addresses and domain names, you’ll probably come across many agency names and acronyms including the Internet Corporation for Assigned Names and Numbers (ICANN), the Internet Assigned Numbers Authority (IANA), the American Registry for Internet Numbers (ARIN), Asia Pacific Network Information Center (APNIC), European IP Networks (RIPE), InterNIC, and others. To avoid confusing matters, I’ll generally refer to ICANN as “the” organization in charge of allocating IP addresses, simply because they’re at the top of the heap, so to speak. If you’re interested in learning more about how it all works and the organizations involved, you can start by visiting ICANN’s web site at www.icann.org. The first step involves finding and registering a unique domain name through an InterNIC accredited registrar. You can find a list of those at www.internic.net/regist.html.

—Alan Simpson, MA, MCSA

Class A, B and C Addresses

There was a time when IP addresses were assigned to organizations based on their size—or roughly the number of computers that would be connected to the network. The largest organizations, such as IBM, General Electric, MIT, and Xerox, were assigned Class A addresses. Slightly smaller organizations received Class B addresses, and the smallest organizations got Class C addresses. Class A IP addresses all start with the number in the range of 1 to 126 and, by default, have a subnet mask of 255.0.0.0. Class B addresses have starting numbers in the range of 128 to 191, and use a standard subnet mask of 255.255.0.0. Class C network addresses all start with a number between 192 and 223, and have a default subnet mask of 255.255.255.0.

Some ranges of IP addresses, such as those starting with 127 and those starting with 224 through 255, aren’t classified as A, B, or C. These addresses are reserved as follows:

- 127.x.y.z (reserved *loopback* address)
- 224.x.y.z through 239.x.y.z (Class D reserved *multicast* addresses)
- 240.x.y.z through 254.x.y.z (Class E reserved experimental addresses)
There are also ranges of private IP addresses, which can assigned to hosts that are clients to, but not servers on, the Web, as listed here:

- 10.0.0.0 through 10.255.255.255 (subnet mask 255.0.0.0)
- 172.16.0.0 through 172.31.255.255 (subnet mask 255.255.0.0)
- 192.168.0.0 through 192.168.255.255 (subnet mask 255.255.255.0)

While the private IP addresses can’t be used for servers on the Internet, they can access the Internet through a proxy server or Network Address Translation (NAT).

Table 1-1 summarizes what you’ve just learned about the classed (also called classful) IP addresses. Each class is defined by a certain range of IP addresses. Each class also has “set aside” some private addresses that can be used on a local network without approval from a governing body that assigns globally unique IP addresses.

Recall that the subnet mask identifies which portion of an IP address represents the address of the network as a whole versus the address of an individual host. In a subnet mask, each 255 value indicates 8 bits. The more bits there are in the host portion of the IP addresses, the more unique hosts you can identify on that network. Table 1-2 illustrates this by comparing Class A, B, and C networks. As you increase the number of bits used to identify the network, you increase the number of networks you can have within the class. But, at the same time, since you’re taking away bits for identifying individual hosts, you decrease the maximum number of hosts a given network could contain.

### TABLE 1-1  Ranges of Public and Private IP Addresses

<table>
<thead>
<tr>
<th>Class</th>
<th>From…</th>
<th>To…</th>
<th>Default Subnet Mask</th>
<th>Private</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>1.x.y.z</td>
<td>126.x.y.z</td>
<td>255.0.0.0</td>
<td>10.x.y.z</td>
</tr>
<tr>
<td>Loopback</td>
<td>127.x.y.z</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class B</td>
<td>128.x.y.z</td>
<td>191.x.y.z</td>
<td>255.255.0.0</td>
<td>172.16.y.z through 172.31.y.z</td>
</tr>
<tr>
<td>Class C</td>
<td>192.x.y.z</td>
<td>223.x.y.z</td>
<td>255.255.255.0</td>
<td>192.168.y.z</td>
</tr>
<tr>
<td>Class D</td>
<td>224.x.y.z</td>
<td>239.x.y.z</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Class E</td>
<td>240.x.y.z</td>
<td>254.x.y.z</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Subnet and Broadcast Addresses

As previously stated in our discussions of the network and host portions of IP addresses, you can use the bits to the right of the network portion of the address to identify individual hosts on the network. While that’s true, there are a couple of exceptions. As mentioned, the lowest possible number is reserved as the network ID (also called the subnet address, the subnet ID, or IP network address). For example, if the IP address is 169.254.1.x with a subnet mask of 255.255.255.0, you cannot assign the address 169.254.1.0 to any specific host, because 169.254.1.0 is reserved as the network ID.

The highest possible address in the range of available addresses is reserved as the broadcast address. The broadcast address is used when a host needs to send a message to all other hosts on the network. Using the example 169.254.1.x with a subnet mask of 255.255.255.0, the highest possible host ID is 11111111, or 255 in binary. Hence, you cannot assign the address 169.254.1.255 to a host because that address is reserved for broadcasting. (We’ll discuss broadcasting in depth a little later in this chapter.) So, when you break it all down, here’s what you end up with, given 169.254.1.x with the subnet mask 255.255.255.0:

169.254.1.0      subnet address (network ID)
169.254.1.255    broadcast address

That leaves the following host addresses remaining, which you can assign to hosts:

169.254.1.1 to 169.254.1.254

It’s not always quite as simple as that because subnetting would allow you to break that network into smaller subnets, as we’ll discuss later in the chapter. But before we complicate matters, let’s look at another address you’re likely to assign to hosts in your network, the default gateway.

<table>
<thead>
<tr>
<th>Class</th>
<th>Subnet</th>
<th>Network Bits</th>
<th>Possible Networks</th>
<th>Host Bits</th>
<th>Hosts per Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>255.0.0</td>
<td>8</td>
<td>126</td>
<td>24</td>
<td>16,777,214</td>
</tr>
<tr>
<td>B</td>
<td>255.255.0.0</td>
<td>16</td>
<td>16,384</td>
<td>16</td>
<td>65,534</td>
</tr>
<tr>
<td>C</td>
<td>255.255.255.0</td>
<td>24</td>
<td>2,097,152</td>
<td>8</td>
<td>254</td>
</tr>
</tbody>
</table>
The Default Gateway

Every computer in a network is likely to have a default gateway address. This address represents an interface to computers outside the local subnet. The most common example is a NIC that connects the subnet to the Internet, as in the example shown in Figure 1-3. There, the address 192.168.100.2 identifies the NIC that connects the computer to other computers in the local subnet. The IP address 192.168.100.1 identifies the NIC that connects that computer to the Internet (in other words, computers not within the local subnet).

To understand how it works, you first need to be aware that all information sent across the network is divided into packets (also called frames), each of which contains the data to be sent, as well as the IP address of the destination. When the NIC is handed a packet, it compares the network portion of the destination address to the network portion of its own address. If it determines that the destination address is not the same as its own subnet address, it just sends the packet to the default gateway instead.

In the example shown in Figure 1-3, the server at the top of the subnet is playing the role of a router, in that it accepts messages that are intended for a host that’s not in the current subnet and sends them out through the default gateway. Any Windows 2000 Server computer can play the role of a router, as you’ll learn later. The important point to remember for now is that the default gateway address
represents the place to which all “foreign” packets are sent. If any host in the subnet needs to send a packet to some host that’s not in its own subnet, the packet gets shipped straight to the default gateway.

EXERCISE 1-1

Configuring TCP/IP on Servers

In this exercise, we’ll look at the specific steps required to assign an IP address to a Windows computer on a LAN. We’ll use an example of assigning a static IP address to a computer running Windows 2000 Server. A static IP address is one that’s assigned by the administrator and never changes. Windows 2000 also supports dynamic IP addressing, where a host gets its IP address automatically from a DHCP server. We’ll discuss all of that in Chapter 4. But since Microsoft recommends that all servers in a network use static IP addresses, we’ll assign a static IP address to a server here. Here are the steps involved:

1. Open the Network and Dial-Up Connections window, either from the Settings menu on the Start menu, or by right-clicking My Network Places on the desktop and choosing Properties.
2. Right-click the icon for your Local Area Connection and choose Properties.
3. In the dialog box that opens, click Internet Protocol (TCP/IP) and then click the Properties button.
4. Choose Use The Following IP Address to set a static IP address.
5. Fill in this computer’s IP address and subnet mask as in the example shown in Figure 1-4. Of course, you’ll want to use an IP address and subnet mask appropriate for your own network.
6. If you already know the IP addresses of the default gateway and DNS servers for this network, you can fill those in as well. Otherwise, you can leave those options blank for now.
7. Click the OK button in the current dialog box, and then click the OK button in the remaining dialog box to close that. You can also close the Network and Dial-Up Connections window if you like.
If you have access to two or more computers, and they’re connected right now, you can repeat these steps on any other computers in the network. Generally, Microsoft recommends using dynamic IP addressing in client computers. But until you have a DHCP server set up to assign IP addresses automatically, you can just assign static IP addresses to all of your computers. For the purposes of the exercises in this book, be sure to make all the computers part of the same subnet.

There's no rule that says you must use dynamic addresses on hosts and static IP addresses on servers. But since Microsoft recommends that approach, you should keep it in mind when answering any questions about assigning IP addresses to hosts.

Classless Inter-Domain Routing (CIDR)
The classed A, B, and C networks were fine in the early days of the Internet, when there were relatively few networks connected. But, as time went by and the Internet
As the demand for IP addresses grew, it became clear that the powers that be were going to run out of globally unique IP Class A, B, and C addresses.

To gain some flexibility in doling out ranges of globally unique IP addresses, the registrars came up with Classless Inter-Domain Routing (CIDR, pronounced cider) addresses. CIDR addresses don’t use traditional subnet masks to identify the network and host portions of an IP address. Rather, they use a /x at the end of the IP address, where x is the number of bits used to indicate the network portion of the address.

For example, the address 199.199.199.123/26 would be called a slash 26 address. When viewing the address in binary format, the top (leftmost) 26 bits would be the ones assigned by the registrar, leaving the remaining 6 bits for the administrators assigned to hosts. Referring back to our discussion of subnet masks, if we write the address 199.199.199.5 in binary, and then use corresponding 1’s and 0’s to mask the network portion of the address, we end up with the address and mask shown in Figure 1-5.

You can easily convert a /x to a more traditional subnet mask, though you’ll need to convert binary numbers to their decimal equivalents. You just have to jot down the 32-bit mask with x number of 1’s, followed by enough 0’s to make the number 36 bits in length. Divide that 32-bit number into four octets. Then convert each octet to a decimal number. For example, let’s take the /26 designation. We jot down 26 ones, followed by 6 zeros:

\[
11111111111111111111111111000000
\]

Use dots to separate that into four 8-bit octets:

\[
11111111.11111111.11111111.11000000
\]

Now convert each binary octet to a decimal number, and you get the following:

\[
255.255.255.192
\]
Thus, 255.255.255.192 and /26 are just two different ways of expressing the same thing—a subnet mask that, in binary, has 26 network bits and 6 host bits. If you take a look at Table 1-3, where I’ve converted a series of /x designations to binary and to subnet masks, you’ll see the progression. The last number in the subnet mask is just the last octet converted from binary to decimal.

Before we get any deeper into this business of working with binary numbers, let’s take a moment to look at some strategies you can use to convert decimal to binary, and vice versa.

**Converting Between Binary and Decimal**

The easiest way to convert a binary number to decimal, or a decimal number to binary, is to use the Windows Calculator. Click the Start button, choose Programs | Accessories | Calculator. From the Calculator’s menu bar, choose View | Scientific to get to the view shown in Figure 1-6. Notice the Hex (hexadecimal), Dec (decimal), Oct (octal), and Bin (binary) options.

To convert from decimal to binary, first click the Dec option button to let the Calculator know that you’re about to enter a decimal number. Then enter your decimal number and click the Bin option button to view that value in binary. For example, if you punch in the number 18 in decimal, and then click the Bin option button, the calculator displays 10010, which is the number 18 in binary. To convert that to the 8-bit chunk typically used in TCP/IP addressing, just pad the left side with leading 0’s, such as 00010010.

### TABLE 1-3

<table>
<thead>
<tr>
<th>/x Designation</th>
<th>Subnet Mask in Binary Format</th>
<th>Subnet Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>/24</td>
<td>11111111.11111111.11111111.00000000</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>/25</td>
<td>11111111.11111111.11111111.10000000</td>
<td>255.255.255.128</td>
</tr>
<tr>
<td>/26</td>
<td>11111111.11111111.11111111.11000000</td>
<td>255.255.255.192</td>
</tr>
<tr>
<td>/27</td>
<td>11111111.11111111.11111111.11100000</td>
<td>255.255.255.224</td>
</tr>
<tr>
<td>/28</td>
<td>11111111.11111111.11111111.11110000</td>
<td>255.255.255.240</td>
</tr>
<tr>
<td>/29</td>
<td>11111111.11111111.11111111.11111000</td>
<td>255.255.255.248</td>
</tr>
<tr>
<td>/30</td>
<td>11111111.11111111.11111111.11111100</td>
<td>255.255.255.252</td>
</tr>
</tbody>
</table>
To convert decimal to binary, start by clicking the Bin option button. Then type in the binary number. Leading 0’s will be ignored because they have no value, so just start typing at the first 1. For example, to convert 00110011 to decimal, you’d type or punch in 110011. Then click the Dec option button to see the result, 51.

### SCENARIO & SOLUTION

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>My work PC’s IP address is 24.0.2.5 (Class A) and has a subnet mask of 255.255.255.0 (Class C). What’s up with that?</td>
<td>Your company has a Class A address. It’s using the subnet mask 255.255.255.0 to break it down into smaller subnets. This illustrates the fact that it’s really the IP address, and not the subnet mask, that defines Class A, B, or C.</td>
</tr>
<tr>
<td>Are hardware addresses used at all in TCP/IP?</td>
<td>Ethernet cards don’t understand TCP/IP, they only understand hardware addresses. TCP/IP messages contain both the IP and hardware addresses of the sending and receiving machines. The protocol that resolves IP addresses to hardware addresses is called Address Resolution Protocol (ARP).</td>
</tr>
<tr>
<td>Do I have to get a globally unique domain name and IP address to set up my network?</td>
<td>Not at all. You can use a private address to set up your network. For example, if your subnet will have no more that 254 hosts, you can use 192.168.0.0 as your network address and 255.255.255.0 as your subnet mask. You can then assign addresses from 192.168.0.1 to 192.168.0.254 to hosts in your network. You’ll still have basic Internet connectivity.</td>
</tr>
</tbody>
</table>
Now that we’ve been through some of the basics of TCP/IP addressing, let’s take a look at some scenario questions and answers that might come up.

CERTIFICATION OBJECTIVE 1.02

Determine Valid IP Addresses

When you’re working with a standard Class C address, it’s easy to figure out the subnet address, broadcast address, and remaining addresses that are available to assign to hosts within the network. Recall that the subnet address is the last octet set to its lowest possible value, 0 in a Class C address; and the broadcast address is the last octet set to its highest possible value, 255 when you’re talking about an octet (because 11111111 = 255). You can assign all the addresses between those two extremes to hosts within your network. Thus, we end up with this:

Subnet mask: 255.255.255.0
Subnet address: 192.168.0.0
Broadcast address: 192.168.0.255
Remaining addresses for hosts 192.168.0.1 to 192.168.0.254

Working with these same numbers in binary shows why this all makes sense. For example, using those same values, 192.168.1.0 and a subnet mask of 255.255.255.0 in binary, we can use the letter \( n \) to identify network bits, and \( h \) to identify host bits:

\[
\begin{array}{cccc}
192 & 168 & 0 & 0 \\
11000000 & 10101000 & 00000000 & 00000000 \\
nnnnnnn & nnnnnnnn & nnnnnnn & hhhhhhh \\
11111111 & 11111111 & 11111111 & 00000000 \\
255 & 255 & 255 & 0 \\
\end{array}
\]

Subnet mask (binary)

Subnet mask (decimal)

\[255.255.255.0\]

Subnetting

When we subnet a Class C address (in other words, break it down into two or more subnets), we’re “swiping” host bits and making them into subnet bits. Let’s see what happens when we change the subnet mask in the preceding example from 255.255.255.0 to 255.255.255.224. The bits that are affected by the change are indicated next by the letter \( s \), to indicate that they’re “swiped” bits now used to identify the subnet:
It’s important to keep in mind that the subnet is a mask, not a number per se, and as such you must have a series of contiguous 1’s for the network/subnet, followed by contiguous 0’s for the host portion. Thus, only values that have leading 1’s, like 10000000, 11000000, 11100000, and so forth, are valid. There are only nine possibilities, as summarized in Table 1-4.

It wouldn’t actually make sense to have a /32 designation (for example, 255.255.255.255 subnet mask), since there wouldn’t be any bits left in the host portion of the address. I’ve only included that in the table to show the progression.

Finding Valid IP Addresses

You can determine how many subnets, and how many hosts per subnet, you’ll get from each /x designation as follows:

Number of subnets = \(2^n\)

Number of hosts = \(2^h - 2\)

where \(n\) is the number of network bits, and \(h\) is the number of host bits. Since there are always 32 bits in the mask, and we’re given the number of network bits
by the /x designation, we know there will always be $32 - n$ host bits available. (Incidentally, the reason you have to subtract 2 from the hosts calculation is because the highest and lowest addresses are reserved for the subnet address and broadcast address, respectively, so you can’t assign those two addresses to hosts.)

With that in mind, Table 1-5 lists some /x designations, the number of subnets each provides, and the number of hosts you’ll get per subnet. I’ve included /31 and /32 in the table just to illustrate the progression. These actually are invalid for practical use, though, because they don’t leave a sufficient number of bits for addressing hosts.

Suppose you want to work with the IP address 192.168.0.1/25. You know you have two subnets to work with. But what are the ranges of available IP addresses to work with? Well, we know the network ID of the first subnet is

192.168.0.0

We know we have 126 host addresses to work with, and the first possible host address is 192.168.0.1. Therefore, the range of available addresses must be

192.168.0.1 to 192.168.0.126

TABLE 1-5

<table>
<thead>
<tr>
<th>/x Designation</th>
<th>n Bits in Last Octet (x-24)</th>
<th>Available Subnets ($2^x$)</th>
<th>$h$ Bits ($32 - n$)</th>
<th>Hosts per Subnet ($2^h - 2$)</th>
<th>Subnet Mask (Decimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>/24</td>
<td>0</td>
<td>1</td>
<td>8</td>
<td>254</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>/25</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>126</td>
<td>255.255.255.128</td>
</tr>
<tr>
<td>/26</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>62</td>
<td>255.255.255.192</td>
</tr>
<tr>
<td>/27</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>30</td>
<td>255.255.255.224</td>
</tr>
<tr>
<td>/28</td>
<td>4</td>
<td>16</td>
<td>4</td>
<td>14</td>
<td>255.255.255.240</td>
</tr>
<tr>
<td>/29</td>
<td>5</td>
<td>32</td>
<td>3</td>
<td>6</td>
<td>255.255.255.248</td>
</tr>
<tr>
<td>/30</td>
<td>6</td>
<td>64</td>
<td>2</td>
<td>2</td>
<td>255.255.255.252</td>
</tr>
<tr>
<td>/31*</td>
<td>7</td>
<td>128</td>
<td>1</td>
<td>0</td>
<td>Not valid</td>
</tr>
<tr>
<td>/32*</td>
<td>8</td>
<td>256</td>
<td>0</td>
<td>0</td>
<td>Not valid</td>
</tr>
</tbody>
</table>

*Invalid because they don’t leave a sufficient number of host bits.
The broadcast address would be one higher than the last host address, so it must be

192.168.0.127

That covers the first subnet. The second subnet then starts right after the first subnet’s broadcast address. Thus, the subnet address of the second subnet must be

192.168.0.128

Once again, we have 126 possible host addresses. The first valid host address would be one greater than the subnet address, so the range must be

192.168.0.129 to 192.168.0.254

because that’s the range of numbers needed to address 126 hosts. The second subnet’s broadcast address would be one greater than the last host address, so that address must be

192.168.0.129 to 192.168.0.254

Table 1-6 summarizes the preceding information. As you can see, we’ve actually taken a Class C address and split it right in half, making two equal-sized subnets. If you use a /26 designation with a class C address of 192.168.9.9, you end up with 2^2 or 4 subnet bits. To determine the remaining host bits, we subtract 26 from 32, which tells us we have 6 host bits to work with. Thus, the maximum number of hosts per subnet would be 2^6 – 2, or 62. Again, reserving the lowest and highest address within each of the four subnets for the network ID and broadcast address leaves us with the subnets and IP addresses listed in Table 1-7.

<table>
<thead>
<tr>
<th>Subnet</th>
<th>Subnet Address</th>
<th>First Host</th>
<th>Last Host</th>
<th>Broadcast Address</th>
<th>Subnet Mask</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>192.168.0.0</td>
<td>192.168.0.1</td>
<td>192.168.0.126</td>
<td>192.168.0.127</td>
<td>255.255.255.128</td>
</tr>
<tr>
<td>2</td>
<td>192.168.0.128</td>
<td>192.168.0.129</td>
<td>192.168.0.254</td>
<td>192.168.0.255</td>
<td>255.255.255.128</td>
</tr>
</tbody>
</table>
In a nutshell, we’ve taken the Class C address 192.168.0.0 and divided it into four separate, equal-sized chunks. The starting address of each subnet is exactly 64 greater than the previous subnet’s starting address, because we have 64 addresses per subnet (60 hosts plus the subnet and broadcast addresses).

Subnetting is simple to do with a good subnet calculator, like SolarWinds.Net Advanced Subnet Calculator, available from www.tucows.com—though, of course, you wouldn’t be able to use that during your exam. But even without a subnet calculator, you can figure out anything as long as you know the network address and subnet mask. For example, suppose a senior administrator asks you to configure some new network using 192.168.0.160 with a subnet mask 255.255.255.240. What IP addresses can you assign to your host? Right off the bat, we know our network address, since that’s a given:

192.168.0.160 subnet address (given)

So, how many hosts per subnet? First, we need to figure out how many subnet bits are available, so we convert the last octet in the subnet mask, 240, to binary, which yields 11110000. So, we have 4 host bits to work with, and hence $2^4 - 2$, or 14 hosts per subnet. We know that the IP address of the first host will be one greater than the subnet address, thus our range of IP addresses is

192.168.0.161 to 192.168.0.174 (14 hosts per subnet)

The broadcast address is one more than the last IP address, and thus is the following:

192.168.0.175 (broadcast address)
Broadcasting

Hosts on small subnets often use broadcasting to communicate with one another. Broadcasting is required when a given host doesn’t know the address of some host with which it needs to communicate. To illustrate how broadcasting works, let’s suppose a host named Igor needs to contact a host named Franz, but doesn’t know Franz’s hardware address or IP address. How’s Igor going to get his message across? Easy. Since he doesn’t know of a specific address to send the message to, he sends it to the broadcast address, which, in turn, automatically delivers the message to every host in the subnet. You might think of Igor sending his message to the subnet’s broadcast address as being the same as Igor shouting “Igor at 192.168.0.2 here. If there is a Franz out there, please send me your address.”

Every host on the network hears the broadcast message and checks to see who the message is intended for. Each host examines the message to see if the name Igor is looking for matches its own name. If the names don’t match, the message is just ignored and no reply is sent back to Igor. However, when Franz sees that the message is addressed to him, he replies, “Hey Igor at 192.168.0.2, Franz here, and my address is 192.168.0.5,” as illustrated in Figure 1-7. So now Igor and Franz know each other’s addresses, and can send messages directly back and forth.

That all works just fine and dandy, but there’s one big drawback. Igor has to pester every host in the LAN just to find the one host he’s really trying to communicate with. That’s not a big deal on a single small subnet. However, if you look at an extremely large network, like the Internet, you can see why broadcasting would create way too much traffic and take way too long. For example, suppose you type www.GeneralSpecificX.com into your web browser, which knows nothing about that site’s IP address. If your browser had to go to every single host on the Internet asking “Are you www.GeneralSpecificX.com?,” it would be pestering literally hundreds of millions of computers with this stupid question. And those other hundreds of millions of hosts would be pestering each other, and your computer, with similar stupid questions. There’d be so much bandwidth eaten up by all these broadcasts, it would be impossible to get anything else done.

So what’s the solution to the broadcasting problem? In a word, routing. As you may recall, a router (or default gateway) connects a subnet to the “outside world.” One side of the router has an IP address that makes it a member of the subnet to which it’s connected. As such, the router “hears” all the broadcast messages going across the subnet. However, the one thing it won’t do is send those broadcast messages through to the outside world. In other words, when the router gets a broadcast
message, it “shuts down the gateway,” in essence saying “Broadcast messages stop here,” as illustrated in Figure 1-8.

CERTIFICATION OBJECTIVE 1.03

Configure Routing

So now, armed with this information, let’s look at the concept of routing, which is central to TCP/IP. The simplest form of routing is the network that’s attached to the Internet via a router or modem and an Internet service provider (ISP). Hosts within the local subnet can communicate by broadcasts, and such messages stay within the subnet. Messages that are destined for hosts outside the local subnet are sent to the default gateway, which is the IP address of the device that connects the subnet to the Internet, as illustrated in Figure 1-9. That device then forwards the message to the ISP, who handles it from there.
Small Business Routing Scenario

In a small business, we might find multiple routers connecting multiple departmental subnets, as shown in Figure 1-10. In that scenario, Router1 connects the Marketing and Sales subnets, Router2 connects Sales to Accounting, and Router3 connects Accounting to Marketing. Any host in the company can contact any other host, since all the subnets are connected by routers.

In this scenario, computers within any given department can communicate with one another by broadcasting. But, for a message to reach a remote network (some other department’s subnet), the message will have to cross one or two routers. For efficiency, we’d prefer messages to take the shortest path through one router. For example, we’d prefer a message being sent from Sales to Marketing to go through...
Router1. However, if Router1 were unavailable, Sales could still get its message across to Marketing by going through Router2 and Router 3. Hence, we get some fault tolerance here in that if one router goes down, messages can still get through.
Corporate Scenario

We can keep scaling up to larger, more complex scenarios. For example, Figure 1-11 shows a larger corporate or enterprise network with all kinds of networks and protocols joined together with a bunch of routers. In that example, Windows 2000 Server computers are used as routers (as opposed to “dedicated routers”). I’ll show you how to make a Windows 2000 Server computer into a router momentarily.

Obviously, we won’t get into all of the configuring needed to set up such a complex network right here. The point, though, is that a little bit of routing goes a long way in connecting all kinds of networks together, providing network communications across a wide variety of platforms. The Internet is exactly that,
a complex of computers, cables, networks, and routers connecting computers and networks around the globe.

Finally, bear in mind that there are no geographical boundaries here either. Any given subnet or network can be anywhere in the world. A large enterprise with offices in Europe, Asia, Mexico, and the United States would still use the same basic routing mechanisms to connect all these far-flung clients together into a (relatively) seamless network where any host could communicate with any other host, anywhere in the world.

**Building a Windows 2000 Router**

As you probably know, you can buy “dedicated” routers from manufacturers like Cisco and Lucent. But it’s not entirely necessary to do so, as any Windows 2000 Server computer can easily play the role of router. Configuration is easy as well. First, you need to have two separate subnets to connect, of course. On the

<table>
<thead>
<tr>
<th>SCENARIO &amp; SOLUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What’s the difference between a network, a subnet, a physical segment, and a segment?</strong></td>
</tr>
<tr>
<td>The terms are used interchangeably, just as the terms folder, subfolder, and directory are. Any time you have two or more computers connected, you essentially have a network. Large networks, however, are often divided into smaller subnets (also called segments), largely for efficiency. Technically, a subnet (or segment) is any group of computers that can communicate with one another without the aid of a router.</td>
</tr>
</tbody>
</table>

| **What’s the difference between a router and a gateway?** |
| The terms tend to be used interchangeably, as a synonym for “the way out of this local subnet.” However, the term gateway is a bit more generic, in that it can refer to a router; a computer that’s acting as a router; or even the software used to provide communications between disparate networks, such as a Gateway Services for NetWare, which provides connectivity between Windows and Novell networks. The default gateway is where all “foreign” packets are sent—all packets that aren’t addressed to a host that’s within the current subnet. |

| **What’s the difference between the Internet and an intranet?** |
| An intranet is composed of two or more subnets that use TCP/IP and routers to communicate with one another, though you’ll sometimes hear these referred to as “internets.” The Internet (with a capital I) is the big daddy of all internets that connects networks from all over the globe. |
Windows 2000 Server computer that will be playing the role of router, you need to make sure the Routing and Remote Access Server is configured. Here’s how:


2. Click the name of the server that will be acting as router. If you’ve never configured this service before, you’ll see a message prompting you to configure the service now, as in Figure 1-12. Click the Action button and choose Configure Routing and Remote Access.

3. A wizard opens up and takes you through the steps required for the basic configuration. In this scenario, you’d choose Network Router when prompted for the configuration type, and click the Next button.

4. Follow the wizard through until you get to the Finish page.

When you’ve completed the wizard, you’re ready to start the next phase, which involves installing the NICs. You would just go through the usual procedure. When both NICs are installed, each will have its own icon in Network and Dial-Up Connections, as in the example shown in Figure 1-13.

Since each NIC is a separate network interface, each can have its own unique TCP/IP settings. In this situation, you need to configure each NIC with a valid IP address for the subnet to which it connects. For example, take a look at Figure 1-14.
The subnet on the left has the address 192.168.0.0 subnet mask 255.255.255.0. The subnet on the right has the address 192.168.100.0 subnet mask 255.255.255.0. These are two separate subnets, since the network portions of their IP addresses clearly don’t match.

To get routing to work, each NIC needs to be connected to and configured as a host within its subnet. For example, in the example shown in Figure 1-14, I've given NIC1 the IP address 192.168.0.1, thereby making it a host on the 192.168.0.0
subnet. I gave NIC2 the IP address 192.168.100.1, making it a host on the 192.168.100.0 subnet. By the way, a computer that contains two or more NICs is called a multihomed computer. Server01 in this example is, obviously, a multihomed computer at this point.

To use the router, all the hosts on network 192.168.0.0 would need to be configured to use 192.168.0.1 as their default gateway. All the hosts on subnet 192.168.100.0 would use 192.168.0.1 as their default gateway. Thus, broadcasts and other communications within each subnet stay in their respective subnets. Messages intended for “some other subnet” are sent to the interface on the router.

Finally, you’ll want to make sure routing is enabled on the server. Typically, the wizard you ran earlier in this section would be sufficient to get that going. But just in case you have any problems with the router connection, you’ll want to make sure the service is enabled. Again, open the Routing and Remote Access Services administrative tool, right-click the server’s name in the console tree, and choose Properties. On the General tab, make sure routing is enabled. For this scenario, where you have only two subnets connected, you’ll also want to make sure the Local Area Network (LAN) Routing Only option is selected, as shown in Figure 1-15.

How Routing Works

To understand the basics of routing, start with a single source host that’s trying to get a message to some other destination host. The source host’s own IP address is 192.168.100.33 with a subnet mask of 255.255.255.0. The destination host’s IP address is 192.168.100.122 with a subnet mask of 255.255.255.0. If we stack the IP addresses and subnet mask one atop the other, we can see that the destination host is on the same network (or subnet) as the source host:

```
192.168.100.33     (source)
192.168.100.122    (destination)
255.255.255.0      (subnet mask)
```

The source host can “see” this same relationship. It “knows” that the destination host is on the same subnet. So it need not go through any routers to get to that host.

Viewing the Routing Table

Every computer on a TCP/IP network has a built-in routing table. The routing table is built automatically from known information. You can view the routing table by
entering the command `route print` at the command prompt. The results might look something like the example in Figure 1-16.

To interpret the command’s output in this example, you first need to know some things about the machine on which the command was entered. In this example, I entered the command at a Windows 2000 Professional computer that has the following TCP/IP configuration:

<table>
<thead>
<tr>
<th>TCP/IP Address:</th>
<th>10.10.1.31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subnet mask:</td>
<td>255.0.0.0</td>
</tr>
<tr>
<td>Default gateway:</td>
<td>10.10.1.1</td>
</tr>
</tbody>
</table>

The lines under Interface List indicate this computer’s network interfaces. The first item, `0x1`, is the TCP loopback interface used in conjunction with the loopback address for testing purposes. Every TCP/IP client has the same loopback address of 127.0.0.1; the loopback interface is just the address where loopback messages get sent. Later in this chapter, you’ll see how you can use that address for testing and
### Chapter 1: Configuring and Troubleshooting TCP/IP

#### FIGURE 1-16

Sample output from a `ROUTE PRINT` command

<table>
<thead>
<tr>
<th>Network Destination</th>
<th>Netmask</th>
<th>Gateway</th>
<th>Interface</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>10.10.1.1</td>
<td>10.10.1.31</td>
<td>1</td>
</tr>
<tr>
<td>10.10.1.1</td>
<td>255.255.255.255</td>
<td>10.10.1.1</td>
<td>10.10.1.31</td>
<td>1</td>
</tr>
<tr>
<td>10.10.1.1</td>
<td>255.255.255.255</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>1</td>
</tr>
<tr>
<td>10.255.255.255</td>
<td>255.255.255.255</td>
<td>10.10.1.1</td>
<td>10.10.1.31</td>
<td>1</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>255.255.255.255</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>1</td>
</tr>
<tr>
<td>224.0.0.0</td>
<td>224.0.0.0</td>
<td>10.10.1.1</td>
<td>10.10.1.31</td>
<td>1</td>
</tr>
<tr>
<td>255.255.255.255</td>
<td>255.255.255.255</td>
<td>10.10.1.1</td>
<td>10.10.1.31</td>
<td>1</td>
</tr>
</tbody>
</table>

### Troubleshooting

The second item in this example, 0x2, is this computer's NIC. You can see its hardware address, as well as the make and model of the card. This particular machine has only one NIC installed. If it were a multihomed machine with multiple NICs, those additional NICs would be listed as 0x3, 0x4, and so forth.

The next section of the display, titled Active Routes, lists routes that this machine knows about. Each row is divided into the following columns:

- **Network Destination**: A potential destination IP address, to which messages might be sent.
- **Netmask**: A subnet mask for the network destination, which further defines which addresses will be included in this route.
- **Gateway**: The IP address that provides access to the network destination addresses.
- **Interface**: The local IP address that leads to the gateway.
- **Metric**: The “cost” of a route in terms of “hops” across routers that will be required. The path from a host to its default gateway is also considered a hop, so there is always at least one hop, even when no routers are involved.

Now let's take a look at some of the routes listed in the sample output. The first line looks like this:

<table>
<thead>
<tr>
<th>Network Destination</th>
<th>Netmask</th>
<th>Gateway</th>
<th>Interface</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>10.10.1.1</td>
<td>10.10.1.31</td>
<td>1</td>
</tr>
</tbody>
</table>
The address 0.0.0.0 with the netmask 0.0.0.0 translates roughly to “the place you should go if none of the lines in the routing table apply.” In other words, this row defines the default gateway for all packets that aren’t within broadcast range, and that don’t meet any of the criteria in the other lines in the routing table.

Let’s take a look at the next line now:

<table>
<thead>
<tr>
<th>Network Destination</th>
<th>Netmask</th>
<th>Gateway</th>
<th>Interface</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.0</td>
<td>255.0.0.0</td>
<td>10.10.1.31</td>
<td>10.10.1.31</td>
<td>1</td>
</tr>
</tbody>
</table>

This line says that “To get to any address that starts with 10. (in other words, 10.0.0.0 subnet 255.0.0.0), use your own NIC at 10.10.1.31. There will not be any router hops to make.” This makes sense if you think about it for a minute. Recall that this is a machine on a Class A network. We know this because its own IP address and subnet mask are 10.10.1.31 and 255.0.0.0. So it stands to reason that in order to get a message out to another machine on this same network, the machine could use its own NIC (10.10.1.31) as the gateway to the local network, and there wouldn’t be any routers involved.

The third route network destination, 10.10.1.31 netmask 255.255.255.255, refers to the local computer. This line essentially says “To get to yourself, use the loopback address 127.0.0.1.” The next destination, 10.255.255.255 netmask 255.255.255.255, is the broadcast address. So this line says “To broadcast a message to all hosts on the 10.0.0.0 network, use your own 10.10.1.31 NIC.” The next destination address, 127.0.0.0 netmask 255.0.0.0, is the reserved loopback address. Because the netmask uses 255.0.0.0, this line says “Any message sent to 127.anything.anything.anything gets sent to the IP address 127.0.0.1.”

The network destination 224.0.0.0 netmask 224.0.0.0 is the reserved multicast address. We’ll discuss multicasting in depth later in the book. For now, it’s sufficient to know that multicasting is a means of sending a single stream of data to multiple IP addresses, sort of like a radio station that just sends out its show via an antenna, and any radio that happens to be tuned to that station hears the show. The 255.255.255.255 netmask 255.255.255.255 is the limited broadcast address, any alternative route used by some broadcasts to the local subnet.

The last section shown in Figure 1-16, titled Persistent Routes, lists static, permanent routes created by an administrator. In the sample output, there are none listed, simply because I haven’t created any. I’m relying on the default gateway address to handle all messages with destinations outside my local subnet. But you can’t always rely on that. We’ll discuss why, and how to get around it, in the next section.
EXERCISE 1-2

Viewing a Computer’s Routing Table

Like I said, every computer has a built-in routing table. So you can do this simple exercise on virtually any Windows machine. I didn’t create an exercise that actually lets you change the routing table, as you wouldn’t want to play around with that on a real, production network. Furthermore, you could only create a route to a viable network, and I don’t know what, if any, networks you’re connected to. But, anyway, to perform this simple exercise on a Windows 2000 machine, follow these steps:

1. Click the Start button and choose Programs | Accessories | Command Prompt.
2. Type `route print` and press ENTER.
3. That’s it. If you’d like a printed copy, type `route print >prn` and press ENTER.
4. Type `exit` and press ENTER if you want to close the Command Prompt window.

The output of your `route print` command may not match the example shown in this chapter, but you should see many of the same default routes.

Configuring Routing Tables

Suppose you work in a company that has an intranet composed of three networks and two routers, as shown in Figure 1-17. Notice some features of this scenario. We have three separate Class C networks here. Network A’s network address is 200.50.50.0, Network B’s network address is 199.150.150.0, and Network C’s address is 197.100.100.0. Networks A, B, and C could all be subnets of one Class C address, in which case you’d need a custom subnet mask. But for our current example, that wouldn’t matter. Either way, you’d still need routers to connect the various networks or subnets.

Notice that Network C contains two routers, one at the address 197.100.100.101 and the other at 197.100.100.102. There is no Internet connection in this example, which means there is no default gateway to which Network C can just send all messages intended for hosts outside itself. So how does a host on Network C
(say, 197.100.100.33) get a message to a host on Network A (say, 200.50.50.31)? The obvious answer is through Router1. But the machine can only know this if its routing table tells it to go through Router1. You can manually add a route to a machine’s routing table to handle a situation like this. In this example, in order for a host from Network C to get a message to Network A or B, it would need the following “instructions” placed in its routing table:

- To get a message to Network A (200.50.50.0), send it to Router1 at 197.100.100.101.
- To get a message to Network B (199.150.150.0), send it to Router2 at 197.100.100.102.
These “instructions,” which are formally called static routes, can be added to the routing table using the ROUTE command with the following syntax:

```
ROUTE [-p] ADD destination MASK subnet gateway METRIC m IF interface
```

where

- `-p` is an optional switch. If included, manually added routes are persistent, in that they exist from one reboot to the next. If omitted, the route exists only during the current session, and will cease to exist once the machine is rebooted.
- `destination` is the address or range of addresses that this routing table applies to.
- `subnet` is the subnet mask for the destination that identifies the network and host portions of the destination address.
- `gateway` is the IP address that provides access to the network.
- `m` is the number of router hops required to get to the destination.
- `interface` is a single-digit number identifying which NIC card interface to use. You can omit this to have the command locate the best interface automatically.

Let’s look at an example. Suppose I want to tell one of the workstations in Network C “When you get a message that’s addressed to any address starting with 200.50.50, send it to IP address 197.100.100.101.” To do that, I get to the command prompt on that computer and enter the following command:

```
route -p ADD 200.50.50.0 MASK 255.255.255.0 197.100.100.101 METRIC 2
```

We’d also want to tell that computer to send all messages destined for any address starting with 199.150.150 to 197.100.100.102, the near-side IP address of the router that connects network C to network B. So we’d also enter this command:

```
route -p ADD 199.150.150.0 MASK 255.255.255.0 197.100.100.102 METRIC 2
```

The command will check the specified route before adding it to the routing table. If for some reason the specified network cannot be reached, the entry will be rejected and you’ll see an error message to that effect. The problem could be a simple typo or a connection problem to the remote network.

Once you’ve successfully entered a route, it will appear in the output of the `route print` command. If you included the `-p` switch, the route will be listed under the
Persistent Routes heading. Otherwise, the new route just appears in the regular list of routes. In this example, the following lines would be added under Persistent Routes:

<table>
<thead>
<tr>
<th>Network Destination</th>
<th>Netmask</th>
<th>Gateway</th>
<th>Interface</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>200.50.50.0</td>
<td>255.255.255.0</td>
<td>197.100.100.101</td>
<td>197.100.100.33</td>
<td>2</td>
</tr>
<tr>
<td>199.150.150.0</td>
<td>255.255.255.0</td>
<td>197.100.100.102</td>
<td>197.100.100.33</td>
<td>2</td>
</tr>
</tbody>
</table>

Notice that in both `route add` commands, I omitted the IF `interface` parameter. Since this machine has only one NIC, the `route add` command can test the connection and figure this out on its own.

**How Routing Conflicts Are Handled**

Recall that our small sample network contained no default gateway to the Internet. As such, there’s no way for our sample host to communicate with the outside world beyond Networks A and B. But suppose we add another NIC to that machine, or an Internet connection through some other machine on the same subnet. For the sake of example, let’s say that the default gateway address to the Internet is at 197.100.100.1. When we do a `route print` command on that machine, the output might include the following routes:

<table>
<thead>
<tr>
<th>Network Destination</th>
<th>Netmask</th>
<th>Gateway</th>
<th>Interface</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>197.100.100.1</td>
<td>197.100.100.1</td>
<td>40</td>
</tr>
<tr>
<td>200.50.50.0</td>
<td>255.255.255.0</td>
<td>197.100.100.101</td>
<td>97.100.100.33</td>
<td>2</td>
</tr>
<tr>
<td>199.150.150.0</td>
<td>255.255.255.0</td>
<td>197.100.100.102</td>
<td>197.100.100.33</td>
<td>2</td>
</tr>
</tbody>
</table>

The large metric, 40, is somewhat typical of an Internet connection where many routers might have to be crossed to get to a specific destination on the Internet. But more importantly, there’s also a conflict here. The default gateway address 0.0.0.0 says “Use 197.100.100.1 for all communications outside this subnet.” But then, the next two lines say “Use 197.100.100.101 for communications to 200.50.50.0, and 197.100.100.102 for communications to 199.199.150.0.” So, which will it be when it comes time to send a message to Network A, the default gateway address or the specified route?

For example, let’s say some hypothetical routing table contains these two routes:

<table>
<thead>
<tr>
<th>Network Destination</th>
<th>Netmask</th>
<th>Gateway</th>
<th>Interface</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>200.50.50.0</td>
<td>255.255.255.0</td>
<td>197.100.100.101</td>
<td>197.100.100.33</td>
<td>2</td>
</tr>
<tr>
<td>200.50.50.200</td>
<td>255.255.255.255</td>
<td>197.100.100.102</td>
<td>197.100.100.34</td>
<td>2</td>
</tr>
</tbody>
</table>

The first route tells the machine to send anything destined for the network 200.50.50.0 through 197.100.100.101.” The second one tells the machine that a
message specifically destined for 200.50.50.200 netmask 255.255.255.255 goes through 197.100.100.102. Since the second entry has the more specific netmask, 255.255.255.255 (in other words, “this particular host”), as opposed to an entire network (255.255.255.0), the second item wins. In the event that two routes have identical subnet masks, the route with the smallest metric will be chosen first.

You can intentionally add some conflicting, static routes to a routing table for fault tolerance. Give the preferred route a low metric, such as 1; the “backup” route a higher metric, like 2; and so forth.

Managing the Routing Table

The route command offers two more options that you can use to manage a routing table. The route delete command allows you to delete static routes from the routing table. For example, entering the command

```
route delete 200.50.50 200
```

would delete the route that has the network destination address 200.50.50.200. If you have two routes with the same network destination address but different gateways, you can include the gateway address to specify the record you want to delete.

You can use the * wildcard character in both the print and delete versions of the command. For example route print 200* displays only routes whose network destination starts with 200. The route delete 200* command would delete all routes whose network destinations start with 200.

The route change command lets you change an existing static route, for example,

```
route change 199.150.150.0 MASK 255.255.255.0 197.100.100.102 METRIC 4
```

The route command alone on a line prints help for the command (same as entering the command route /?).

Now that I’ve told you all of this, let me first point out that it’s very unlikely that you’ll ever have to go from one machine to the next, setting up all these routes. Thanks to dynamic routing and modern routing protocols like RIP (Routing Information Protocol) and OSPF (Open Shortest Path First), routers can keep machines informed of available routes, and individual hosts can compare routing tables to one another and keep each other up to date. We’ll get to these protocols...
in a later chapter. For now, the important thing is to understand that all machines have a routing table. And even if you don’t specifically need to manually add static routes to a machine’s table, it’s good to be able to interpret the contents of the table for troubleshooting purposes.

Just because you might not have to use static routing much in the real world, that doesn’t mean you can just ignore all this stuff. You may well need to analyze some routing tables and understand how ROUTE ADD works to answer some questions on your certification exam!

CERTIFICATION OBJECTIVE 1.04

Troubleshoot TCP/IP and Routing

As you probably know, there’s a lot more to a TCP/IP network than just assigning IP addresses to machines. There are many things that can go wrong after the network is built, and even more things that can go wrong as you’re building the network. For the rest of this chapter, we’ll take a look at some diagnostic tools and troubleshooting techniques that you can use at any time to solve problems as they arise.

Troubleshooting with IPCONFIG

As the name implies, IPCONFIG is a command for checking a machine’s IP configuration. Checking a host’s IP configuration is always the best first step in troubleshooting connectivity problems. Entering just the command `ipconfig` at the command prompt displays basic IP configuration information (IP address, subnet mask, default gateway, and DNS suffix) for each network adapter in the computer. Settings that haven’t been configured yet are left blank.

For more detailed information, use the `/all` switch by entering the command `ipconfig /all`. This command will display general information about the current computer’s IP configuration, followed by detailed information about each installed NIC. Figure 1-18 shows an example of the display produced by an `ipconfig /all` command. Here’s a brief description of what each line is about. Some items describe settings that we haven’t discussed yet—but will over the next two chapters.

- **Host Name** This computer’s hostname, which might be a single name like server01 if DNS hasn’t been set up yet, or it might be a FQDN like server01.certifiable.net if DNS is set up.
Primary DNS Suffix  If DNS has been set up, the domain portion of the DNS name (for example, certifiable.net) appears here.

Node Type  Describes the method used to resolve NetBIOS-style hostnames, like server01, to IP addresses, as will be discussed in Chapter 2.

IP Routing Enabled  A simple Yes or No answer describing whether or not this machine is functioning as a router.

WINS Proxy Enabled  Specifies whether WINS name resolution is enabled, as described in Chapter 2.

Information that’s specific to network adapter cards is listed under the Ethernet Adapter heading. The name of the connection, as it appears in the Network and Dial-Up Connections window is followed by these lines:

Connection-Specific DNS Suffix  If DNS is enabled, shows the DNS domain name that’s specific to this network interface card.

Description  The make and model of the network interface card.

Physical Address  The hardware address of the network interface card.
DHCP Enabled Determines whether or not this card’s address can be assigned automatically by a DHCP server (Yes) or was manually entered (No). More information on DHCP is provided in Chapter 4.

IP Address The IP address of the network interface card.

Subnet Mask The subnet mask of the network interface card.

Default Gateway The IP address of the default gateway where messages outside the broadcast range will be sent.

DNS Servers If DNS is set up, lists the IP address of all available DNS servers.

In terms of what we’ve discussed so far in this chapter, what you’re mainly looking for in IPCONFIG’s output is to ensure that the computer has a valid IP address and subnet mask. If there is a gateway of some sort on the network, whether it be a dedicated router or just a computer that provides access to the Internet, the default gateway address for that router must be correct as well. If you find an error that needs correcting, you can make changes through the TCP/IP Properties dialog box, described previously in Exercise 1-1.

The IPCONFIG command works only on systems that have the TCP/IP networking protocols installed. If entering the ipconfig /all command returns an error message like “TCP/IP is not running on this system,” there’s a problem with the NIC or with the TCP/IP installation. To check to see if the NIC is working properly, open the Control Panel, open the System icon, and click the Hardware tab. Then click the Device Manager button and expand the Network Adapters category. Double-click the icon for your NIC to view its properties. If the dialog box doesn’t indicate any problems, you know the problem lies outside the NIC.

Windows 2000 doesn’t automatically install drivers for every NIC on the market, so it’s a good idea to check the card and TCP/IP right after you install Windows, or install a new card.

If the Properties dialog box indicates, instead, that there is a problem with the card, first check to make sure the card is on the Windows 2000 Hardware Compatibility List. Optionally, you can search for updated drivers via the Internet, and use the Update Driver button on the Drivers tab of the Properties dialog box to install the updated driver.
Chapter 1: Configuring and Troubleshooting TCP/IP

EXERCISE 1-3

Checking an IP Configuration
You’ll no doubt be using the `ipconfig /all` command often through this book, and in the real world as well. So, in this exercise, we’ll go through the simple steps necessary to use the command:

1. Click the Windows Start button and choose Programs | Accessories | Command Prompt.
2. Type `ipconfig /all` and press ENTER. You should see output similar to the example shown in Figure 1-18, earlier in this chapter, but with the data from the current machine.
3. After viewing the output, type `exit` and press ENTER to close the Command Prompt window.

Troubleshooting with PING
Whereas `ipconfig` is a good tool for checking a machine’s IP configuration, `ping` (Packet Internet Groper) is the preferred tool for checking to see if a NIC is working, and for checking connectivity between two machines. Basically, `ping` sends an `echo request` message that basically asks “Are you there?” If the machine being pinged can be reached from the current machine, `ping` displays that machine’s reply. Otherwise, it displays a list of “Request timed out” error messages.

A good strategy for troubleshooting TCP/IP problems with `ping` is to start with the local host, and then gradually work your way out to hosts that are increasingly distant from the local host, as discussed in the sections that follow.

Ping the Loopback Address
For testing and debugging purposes, you can start by pinging the loopback address, 127.0.0.1. That is, at the command prompt, type `ping 127.0.0.1` and press ENTER. You should get a reply, as in the example shown in Figure 1-19.
If pinging the loopback address results in an error message, there's likely a communication problem between Windows 2000 and your NIC. In that case, Microsoft recommends that you remove and reinstall TCP/IP.

**Ping Your Own IP Address**

If you can successfully ping the loopback address, try pinging the local PC’s IP address. Once again, you should see some sort of successful feedback. If you get an error message instead, there's likely a communication problem between your NIC and Windows 2000. In that case, Microsoft recommends that you remove and then reinstall your NIC’s driver.

**Ping the Default Gateway Address**

If your network has a functioning default gateway, you can ping its IP address to verify connectivity to the gateway. For example, if the default gateway address is 192.168.0.1, but pinging that address returns “Request timed out” errors, there's a problem with the address or the connection. If some other administrator already set up the default gateway, verify that you're using the correct default gateway address, and also have that administrator verify that the default gateway is properly connected to the network and functioning correctly.
Chapter 1: Configuring and Troubleshooting TCP/IP

Ping Nearby IP Addresses
Next, try pinging a host on the near side of the router (a computer on the same subnet) by its IP address. For example, let’s say you have a computer named server01 configured as IP 192.168.0.1 and another set up as client01, IP 192.168.0.2. If you’re sitting at 192.168.0.1 and want to check connectivity with the other machine, enter the ping command followed by that machine’s address; for example, ping 192.169.0.2. If the connection works, you’ll get a successful reply. If instead of a reply you get a “Destination host unreachable” error message, then obviously there’s some problem.

If you were able to successfully ping the loopback address, but can’t ping a separate machine, first check the network cabling. Many connectivity problems are nothing more than faulty cable connections. Another obvious but often overlooked potential cause should be checked—make sure the computer that you’re trying to ping is up and running and connected to the network! Finally, if you have basic connectivity but still can’t ping the other machine, the reason may lie in faulty ARP cache entries. You can use the ARP command, discussed in a moment, to view the contents of the cache, as well as to delete faulty entries.

Ping More Distant IP Addresses
If your network contains routers or you have an Internet connection, you can use the PING command to test connectivity to hosts on the far side of your router (outside your local subnet). The same basic syntax applies—ping ipaddress. For example, if you’re connected to the Internet, you could try pinging a web site of mine by entering ping 208.55.30.20.

Be forewarned that some web sites, including www.microsoft.com, are designed not to respond to ICMP Echo Requests, which is the official name of the type of packet a ping command sends. So, if a first attempt fails, try some other sites. If you can’t ping any web sites, trying pinging your default gateway address. If you can’t ping your default gateway, check to make sure its IP address and subnet mask are set up correctly, and that you’re ping the correct address of the default gateway.

Pinging Hostnames
You can also ping another computer by its hostname. For example, if you’re sitting at a computer named client01, which has a connection to a computer named server01, you can ping the server by entering the command ping server01. Once
again, if the connection works, you'll get a positive response. If the ping fails, it could be a name resolution problem—a topic we'll discuss at length in Chapter 2.

If you can ping a host by its IP address, but not its hostname, you should suspect a problem with your DNS configuration or name resolution.

**Troubleshooting with ARP**

The Address Resolution Protocol (ARP) maintains a cache of IP address to hardware address mappings. Entering the command `arp -g` or `arp -a` displays the current mappings. Faulty ARP entries can cause PING echo requests to other computers in the network to fail. For example, if you can ping both the loopback address and your own IP address, but not any other IP addresses, you might be able to fix the problem by clearing out the ARP cache. You can clear individual entries using the syntax `arp -d IPAddress`, where `IPAddress` represents the entry you want to remove. You can delete all ARP entries by using the * wildcard with `-d` (for example, `arp -d *`) or by entering the command `netsh interface ip delete arpcache`.

**Troubleshooting with Tracert**

If you cannot ping a host outside your subnet, you can use the `tracert` (Trace Route) command to locate where the problem might lie. `tracert` provides information about each router or gateway that a message crosses when trying to reach another host. Each router that the message crosses is considered a hop.

The basic syntax for the command is `tracert IPAddress`, where `IPAddress` is the IP address of the destination you’re trying to reach—just as in the PING command. For example, entering the command `tracert 208.55.30.20` would return a list of routers crossed on the way to that destination. The basic format of the display will look like this, where `name` represents the hostname of each router (as available) and `xxx.xxx.xxx.xxx` represents each router’s IP address:

```
Tracing route to www.coolnerds.com [208.55.30.20]
over a maximum of 30 hops:

1  <1 ms  <1 ms  <1 ms  name <xxx.xxx.xxx.xxx>
2  12 ms  19 ms  19 ms  name <xxx.xxx.xxx.xxx>
3  9 ms  15 ms  50 ms  name <xxx.xxx.xxx.xxx>

Trace complete.
```
By default, tracert is limited to testing 30 hops. But you can use the \-h switch to test more or fewer maximum hops. For example, if 30 hops weren’t enough to reach the destination host, you could try something like tracert \-h 40 208.55.30.20 to increase the maximum number of hops to 40.

**EXERCISE 1-4**

Tracing a Route

If you have Internet access from your current machine, you can try out the TRACERT command by following these simple steps:

1. Click the Windows Start button and choose Programs | Accessories | Command Prompt.
2. Type tracert 208.55.30.20 to ping my web site. You should see output similar to the example shown in Figure 1-20, though the names and IP addresses of routers crossed will be different.

**FIGURE 1-20** Results of a sample TRACERT command
3. After viewing the output, type `exit` and press ENTER to close the Command Prompt window.

Since you’re tracing the route to an IP address that’s on the Internet, the trace should complete successfully, provided your Internet connection is working.

If there is a problem with a router between your computer and the destination computer, you may receive feedback that looks more like this:

```
Tracing route to www.coolnerds.com [208.55.30.20]
over a maximum of 30 hops:

1   <10 ms  <10 ms  <10 ms  <xxx.xxx.xxx.xxx>
2    50 ms    50 ms    51 ms  <xxx.xxx.xxx.xxx>
3  <xxx.xxx.xxx.xxx> reports: Destination net unreachable.
```

Or perhaps like this:

```
Tracing route to www.coolnerds.com [208.55.30.20]
over a maximum of 30 hops:

1   <10 ms  <10 ms  <10 ms  <xxx.xxx.xxx.xxx>
2    *         *         *    Request timed out
3    *         *         *    Request timed out
4    *         *         *    Request timed out
```

If a router’s IP address appears repeatedly in the display, that’s called **looping**, and means the router is not forwarding to the next router. This is most often caused by an improper configuration at that specific router. Of course, whenever you encounter a problem tracing the route to an Internet address, it’s very likely that the faulty router will be outside your company’s internal network. The only thing to do, in that case, is to report the problem to your ISP. If the router is in-house, but outside your area of responsibility, you should report the problem to the administrator of that specific router.

Like PING, TRACERT will accept a hostname as well as an IP address. For example, you could enter the command `tracert www.coolnerds.com` to ping the host at 208.55.30.20. As with PING, if you’re able to get to the host by its IP address but not by its hostname, then you know you have a name resolution problem on your hands. As mentioned, we’ll start on name resolution in Chapter 2.
Finally, don’t forget that when it comes to troubleshooting routing problems, the ROUTE PRINT command can be an ideal resource for seeing where a machine “thinks” it’s supposed to route certain messages. Scan the table for conflicting routes, and remember that a route with a more specific netmask will take precedence over a conflicting route in the table that has a less specific netmask.

**Troubleshooting with PATHPING**

The PATHPING command combines features of PING and TRACERT, with some additional functionality. Whereas TRACERT can only point out places where there is no connectivity at all, PATHPING can point out routers that are slow or inconsistent in moving data along due to network congestion or dropped packets that need to be re-sent. To do this, PATHPING sends multiple PING echo requests to all the routers along a route for 25 seconds. Then it calculates the average time and percentage of lost packets encountered at each router. The resulting display helps you pinpoint which router along a path would be causing slow or inconsistent performance.

For example, suppose users are complaining the connection to a server named EgyptDC01 is slow or inconsistent. You enter the command `pathping EgyptDC01` from the source (any machine that’s experiencing problems), and get the following results (I’ve boldfaced the information that’s relevant to this example; your `pathping` command won’t do that):

```
Tracing route to egyptdc01 [10.10.1.2] 
over a maximum of 30 hops: 
0 myServer [172.16.87.35] 
1 aroute1 [180.10.20.22] 
2 aroute2 [192.168.52.1] 
3 aroute3 [192.168.80.1] 
4 aroute4 [10.10.20.22] 
5 egyptdc01 [10.10.1.2]

Computing statistics for 125 seconds...
Source to Here This Node/Link 
Hop RTT Lost/Sent = Pct Lost/Sent = Pct Address
0 0/100 = 0% myServer [172.16.87.35] 
1 46ms 0/100 = 0% 0/100 = 0% aroute1 [180.10.20.22]
```

[.GetType phenomenal C#]
Troubleshoot TCP/IP and Routing

21/100 = 21%

The **This Node/Link: Lost/Sent = Pct** and **Address** display the link between two router IP addresses. The value followed by the pipe character (|) is the loss rate for the specific link. In the example output, you can see that the link between 180.10.20.22 and 192.168.52.1 has a 21 percent loss rate. Dropped packets need to be retransmitted. So, with such a high drop rate, you can see that this link is the problem. So **pathping** has helped you locate the source of the problem. You could then go to that router, or contact its administrator, to try to resolve that problem. Most likely, the router is overloaded.

Now that you’ve learned about some basic network troubleshooting tools, let’s take a look at some possible problem scenarios that might come up, and the solutions to those problems.

### SCENARIO & SOLUTION

<table>
<thead>
<tr>
<th>I can’t get IPCONFIG, PING, or TRACERT to work at all. What should I do?</th>
<th>All three of those commands require a functioning TCP/IP stack. Check to make sure your NIC is properly installed (via Device Manager). If it is functioning, open the Dial-Up and Network Connections icon. Right-click the icon for your adapter and choose Properties. Make sure the Internet Protocol (TCP/IP) option under Components Checked Are Used By This Connection is selected.</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPCONFIG tells me that my computer already has an IP address of 169.254.0.10, but I’ve never assigned an IP address and I know there’s not a DHCP server on this network.</td>
<td>If a Windows 2000 computer is set up to obtain an IP address automatically, but the computer can’t find a DHCP server, Windows will automatically assign an IP address in the range of 169.254.0.1 through 169.254.255.254, with a subnet mask of 255.255.0.0. This capability is called <strong>Automatic Private IP Addressing (APIPA)</strong>, and will be discussed in Chapter 4.</td>
</tr>
</tbody>
</table>
CERTIFICATION SUMMARY

In this chapter, we’ve looked at the ground level of the network infrastructure, the TCP/IP addresses that uniquely identify hosts on a network. You can assign either static or dynamic IP addresses to hosts. A static IP address is one that’s assigned by an administrator, while a dynamic address is one that’s assigned to a host automatically by a DHCP server. Microsoft recommends that you assign static IP addresses to all the servers in your network. Use dynamic IP addressing for all the clients. Doing so minimizes network administration headaches.

A TCP/IP address identifies both the network that a host belongs to and the specific host. The subnet mask tells you which part of the address identifies the network, and which part identifies the host. Standard addresses and subnet masks are categorized as Class A (subnet 255.0.0.0), Class B (255.255.0.0), and Class C (255.255.255.0). Any network can be divided into smaller subnets by using a custom subnet mask, like 255.255.255.128, to split the host portion of that address into a subnet ID and host ID.

You can use the command-line utilities IPCONFIG, PING, ARP, TRACERT, and PATHPING at any stage of a network’s development to test and troubleshoot network connectivity problems. To check the IP configuration on the local host, use the `ipconfig /all` command. To test the connectivity between the local host and some other host, use `ping destination`. To check routing between the local host and a computer on the Internet or some other subnet, use `pathping destination` or `tracert destination`. Whereas `tracert` only lets you see “dead” connections, `pathping` lets you view packet-loss statistics, which can identify slow or inconsistent routers along a path. You can also use the `route print` command to view any computer’s routing table, which lets you see where the host is actually sending messages for a given IP address or range of addresses.
Configure TCP/IP on Servers and Clients

- TCP/IP is a suite of networking protocols, originally designed to solve problems on the Internet’s precursor, ARPANet.
- Don’t confuse IP addresses with hardware addresses. A hardware address is a unique 48-bit address that’s hardwired into every network interface card (NIC) and usually cannot be changed by an administrator.
- The terms Media Access Control (MAC) address, physical address, Ethernet address, Token Ring address, and NIC address are all synonymous with the term hardware address.
- A TCP/IP address is a logical 32-bit address that can be assigned by an administrator.
- A TCP/IP address’s subnet mask identifies which bits in the address represent the network, and which bits represent the host.
- TCP/IP addresses and subnet masks are usually displayed in dotted quad format, xxx.xxx.xxx.xxx, where xxx is any number from 0 to 255.
- The lowest host number in a range of IP addresses is reserved for the network ID, and cannot be assigned to a host. For example, the address 192.168.1.0 subnet mask 255.255.255.0 refers to the network 192.168.1.
- The highest available address in a range of IP addresses is reserved for the broadcast address. For example, 192.168.1.255 is the broadcast address for the network 192.168.1.x subnet mask 255.255.255.0.
- An IP address can be static (assigned by an administrator and permanent) or dynamic (assigned automatically by a DHCP server).

Determine Valid IP Addresses

- Subnetting allows you to subdivide a range of IP addresses into multiple subnets.
- To subnet, you “swipe” bits from the host portion of the subnet mask (the portion containing all 0’s) and make them into network bits (or subnet bits).
In binary, the subnet bits that define the network portion must be contiguous 1’s, and the address must provide for some hosts to be valid.

Valid, commonly used numbers for the host portion of a subnet mask include 128 (10000000), 192 (11000000), 224 (11100000), 240 (11110000), 248 (11111000), and 252 (11111100).

The number of subnets you can get from a subnetted octet is $2^n$ where $n$ is the number of network bits in the octet.

The number of hosts per subnet is equal to $2^h - 2$ where $h$ is the number of host bits the subnet mask provides.

**Configure Routing**

- Every computer on a TCP/IP network has a built-in routing table.
- Most entries in the routing table are dynamic, meaning they’re created automatically from known data.
- You can view the routing table on a machine by entering the `route print` command at the command prompt.
- The network destination 0.0.0.0 netmask 0.0.0.0 tells where all traffic not destined for the current network (or subnet) will be sent. Hence, it identifies the default gateway.
- The Gateway column of the `route print` display identifies the IP address of the NIC used to reach machines within a network destination IP range.
- The Interface column indicates which NIC in the local machine is used to reach the IP address specified in the Gateway column.
- The Metric column indicates the cost of using a route in terms of hops across routers. The trip to the default gateway also counts as a hop, so the metric will never be less than 1.
- If there are conflicting routes in the routing table, the route with the most specific subnet mask will be chosen. If there is a tie between routes, the route with the smallest metric will be used.
- You can manually add static routes to a routing table by using the `route add` command.
To create a persistent route—one that persists through reboots—use the \( -p \) switch in the `route add` command.

It’s rarely necessary to add static routes to a routing table, as the RIP and OSPF routing protocols do a good job of keeping the tables up-to-date automatically.

Troubleshoot TCP/IP and Routing

- The command `ipconfig /all` provides detailed information about the current machine’s IP address, subnet mask, default gateway address, and related information for each of its logical connections.
- If the `ipconfig` command fails, either the NIC or the TCP/IP protocol isn’t properly installed on the computer.
- To test the TCP/IP protocol on the current host, ping the loopback address by entering the command `ping 127.0.0.1`. If you get an error message, Microsoft recommends that you remove and reinstall the TCP/IP protocol.
- If you try to ping the local host’s own IP address and get an error message, Microsoft recommends that you remove and reinstall, or update, the NIC’s driver.
- To test connectivity between any two machines in a network, just ping the other machine’s IP address. For example, if you’re sitting at 192.168.0.1 and want to test connectivity to 192.168.0.2, enter the command `ping 192.168.0.2`.
- If you have trouble pinging another computer within your own subnet, always check the most obvious problems first. Are the cables connected correctly? Is the machine you’re pinging running and online? Did you correctly type the machine’s IP address in the PING command? If the physical connection is okay, try clearing the ARP cache by entering the command `arp –d *`.
- With the PING command, you can use a hostname, rather than an IP address, to identify the destination computer. If you’re able to PING a host by its IP address, but not its name, you’re probably looking at a name resolution problem.
If you try to PING an address that’s not on your subnet, and get an error message, you can use the TRACERT command to get information about each router that was contacted while trying to reach the destination address.

If poor or inconsistent connections are the problem, PATHPING would be the preferred troubleshooting command because it calculates dropped-packet statistics for each path in the route.
SELF TEST

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully because there might be more than one correct answer. Choose all correct answers for each question.

Configure TCP/IP on Servers and Clients

1. The 48-bit addresses expressed in the format xx-xx-xx-xx-xx-xx, as in 00-80-AD-7B-E0-B7, is referred to as which of the following? (Select three correct answers.)
   A. MAC address
   B. IP address
   C. Physical address
   D. Hardware address

2. You’re tasked with adding a new server to a Windows 2000 network. Which of the following would be the best approach to giving the server an IP address?
   A. Use Automatic Private IP Addressing (APIPA) to assign an address automatically.
   B. Go to the Ethernet card’s Internet Protocol (TCP/IP) Properties dialog box and choose Obtain An IP Address Automatically.
   C. Go to the Ethernet card’s Internet Protocol (TCP/IP) Properties dialog box, choose Use The Following IP Address, and manually assign a static IP address.
   D. Manually assign a dynamic IP address through the Ethernet card’s Internet Protocol (TCP/IP) Properties dialog box.

3. Which of the following would be the broadcast address for the network 192.168.0.1 with a subnet mask of 255.255.255.192?
   A. 192.168.1.0
   B. 192.168.1.1
   C. 192.168.1.62
   D. 192.168.1.63
Determine Valid IP Addresses

4. You want to subdivide the Class C network address 192.169.1.0 into four subnets. What would be the appropriate subnet mask?
   A. 192.255.255.255
   B. 255.192.0.0
   C. 255.255.192.0
   D. 255.255.255.192

5. You are the administrator for ABC Corp. One of the branch offices wants to create a subnet with the IP address 199.199.1.128 subnet 255.255.255.224. Which of the following would be the appropriate range of IP addresses that you could assign to hosts on the subnet?
   A. 199.199.1.128 to 199.199.1.159
   B. 199.199.1.128 to 199.199.1.191
   C. 199.199.1.129 to 199.199.1.158
   D. 199.199.1.129 to 199.199.1.255

Configure Routing

6. Client01 in Figure 1-21 cannot ping any sites on the Internet. However, it can ping other hosts within its own subnet. Which of the following would solve the problem? (Choose all that apply.)
   A. Change NIC1’s IP address to 69.81.8.9.
   B. Change NIC2’s IP address to 192.168.1.1.
   C. Change Client01’s default gateway to 192.168.1.1.
   D. Change Client01’s subnet mask to 255.0.0.0.

7. You are the administrator of one subnet in a large corporation. Another administrator asks you to configure a temporary static route to all hosts on the network 192.168.5.0 subnet 255.255.255.0 for testing purposes. Which of the following commands shows the appropriate network destination, subnet mask, and options for setting up such a route?
   A. route add 192.168.5.0 MASK 255.255.255.0
   B. route -p add 192.168.5.0 MASK 255.255.255.0
   C. route add 192.168.5.0 MASK 255.255.255.255
   D. route -p add 192.168.5.0 MASK 255.255.255.255
8. Which of the following routing table entries would be used first to get a message to the host at 201.202.203.101 subnet 255.255.255.0?
   A. Netmask 0.0.0.0 METRIC 30
   B. 201.202.203.0 Netmask 255.255.255.0 Metric 1
   C. 201.202.203.101 Netmask 255.255.255.255 Metric 1
   D. 201.202.203.101 Netmask 255.255.255.255 Metric 2

Troubleshoot TCP/IP and Routing

9. Which of the following commands would be best for testing connectivity to another computer within the same subnet?
   A. ipconfig
   B. tracert
   C. ping
   D. pathping

10. You are able to ping the local computer’s loopback address and IP address. However, you are unable to ping any other computers within the subnet. Which of the following would be the best first step to resolving the problem?
    A. Check the network cabling to ensure the computer is connected to the network.
    B. Remove and reinstall the TCP/IP software.
    C. Update the NIC’s driver.
    D. Assign a different IP address and subnet mask to the computer.
11. You are able to ping the loopback and a host’s own IP address, but you’re having problems pinging any other computers. You’ve checked all the connections and everything appears to be in place. What would be the appropriate next step?
   
   A. Try reaching the other computers with the PATHPING command.
   B. Clear the ARP cache using `arp -d *` and then try again.
   C. Use the `nbtstat` command to check the NetBIOS name cache.
   D. Replace the NIC.

LAB QUESTION

You are an administrator on the network shown in Figure 1-22. Users in the San Francisco office are complaining that they cannot reach DelDC01 from any clients. You run a TRACERT command to DelDC01 from a client in the San Francisco office. It gets as far as TXRouter and then times out. You run some additional PING tests and determine the following connections are valid:

- 176.17.1.1 (DelRouter) to 176.17.1.11 (DelDC01)
- 176.18.1.1 (TXRouter) to 176.17.1.1 (DelRouter)

What could you do at SFRouter to provide a consistent connection to DelDC01 for the San Francisco office?
SELF TEST ANSWERS

Configure TCP/IP on Servers and Clients

1. □ A, C, and D are all correct. The hardware address that’s burned into the Ethernet card goes by many different names.
□ B is incorrect because the logical, administrator-assigned TCP/IP address is always referred to as the IP address and (fortunately) doesn’t have any synonyms.

2. □ C is correct. Microsoft recommends assigning static IP addresses to all the servers in a network. The reason for this is that if you let DHCP assign addresses dynamically, the server’s IP address could change at some time, and then some clients might no longer be able to access the server.
□ A is wrong because APIPA assigns an IP address automatically at startup when there is no DHCP server available to assign an IP address from an acceptable pool.
□ B is wrong because Obtain An IP Address Automatically sets up a dynamic IP address that’s assigned and managed by DHCP.
□ D is wrong because a dynamic address is one that an administrator doesn’t assign manually. DHCP assigns dynamic IP addresses automatically without administrator control.

3. □ D is correct. The host octet in the mask, 192, is 11000000, which means we have 6 host bits, and thus $2^6 - 2$, or 62 host addresses. Since the subnet ID is 192.168.1.0, hosts would be numbered 192.168.1.1 to 192.168.1.62. The broadcast address would be the next value, 192.168.1.63.
□ A is incorrect because that’s the lowest available number, and hence would be used as the network ID.
□ B and C are incorrect because the highest address within the range is used as the broadcast address, and 192.168.1.63 is the highest available number in this range.

Determine Valid IP Addresses

4. □ D is correct. That’s the only subnet mask that would apply to a Class C address. The answer here would have to be 255.255.255.something.
□ A, B, and C are incorrect because they’re not valid subnet masks for a Class C address.

5. □ C is correct. You were given the subnet address 199.199.1.128 to work with, so right off the bat, you know that the first available host address will be one greater than that—199.199.1.129. Converting the host portion of the mask, 224, to binary gives you 11100000.
Chapter 1: Configuring and Troubleshooting TCP/IP

So you know you can have $2^5 - 2$, or 30 hosts on the network. Thus, the range of valid host addresses is 199.199.1.129 to 199.199.1.158.

- A and B are wrong because you can’t assign the network ID 199.199.1.128 to a host.
- C is wrong because 255 – 129 equals 126 possible host addresses. The host portion of the subnet mask only allows for 30 hosts per subnet.

Configure Routing

6. □ C is correct. The default gateway for all clients on that side of the router must match the connection that’s on the same side of the router (Server01 is playing the role of a router here).
   □ A is incorrect because NIC1 is properly addressed for its side of the router.
   □ B is wrong because NIC2 is on the far side of the router, so its IP address wouldn’t need to match the IP addresses on the other side of the router.
   □ D is wrong because Client01’s subnet mask is already appropriate for its subnet.

7. □ A is correct. 192.168.5.0 netmask 255.255.255.0 would encompass all hosts on the 192.168.5.0 network.
   □ B is incorrect because we’re looking to set up a temporary route. The -p switch would make this a persistent route.
   □ C and D are incorrect because the netmask is too specific. We want a route that will encompass all messages to 192.168.5.x.

8. □ C is correct. It has the most specific netmask and a lower metric than option D.
   □ A is incorrect because all the other options have a more specific netmask.
   □ B is incorrect because the netmask is less specific than those shown in C and D.
   □ D would not be chosen over C because D has a higher metric.

Troubleshoot TCP/IP and Routing

9. □ C is correct. ping is an easy tool for checking connectivity within the local subnet.
   □ A is wrong because ipconfig only tells you about the local host.
   □ B is wrong because tracert is for checking connectivity on the far side of the router (outside the local subnet).
   □ D is wrong because pathping is used for determining router statistics, not connectivity within a subnet.
10. ✓ A is correct. You know that the IP software on the local host is working correctly because you can ping the local host. Therefore, the first thing to check is the cable that connects the host to the network.
   ✗ B and C are wrong because you already know the NIC and IP software is functioning correctly.
   ✗ D is wrong because you'd only want to change the IP address and subnet mask after you've ascertained that the networking hardware is functioning, and have also determined that there's a problem with the local TCP/IP configuration.

11. ✓ B is correct. When connectivity between hosts within a subnet fails, even though everything appears to be in place, a faulty ARP cache could be the problem.
   ✗ A is incorrect because pathping tests across routers.
   ✗ C is incorrect because nbtstat and NetBIOS names aren't an issue here.
   ✗ D is incorrect because it's a lot more trouble to go to than entering arp -d * and trying again.

LAB ANSWER

This one takes some thinking, so let’s look at what you know. The route to DelDC01 is reachable from 176.18.1.1 on TXRouter, as indicated from your ping tests. DelRouter is able to reach TXRouter, as indicated by tracert. But the hop from TXRouter to DelRouter isn’t working. What’s the most likely scenario in a situation like this? Either the routing table on SFRouter is wrong and needs to be corrected, or there just is no persistent route in the table for this path.

So let’s say you check the routing table and there’s nothing in there to direct packages addressed to 176.17.1.x to any particular IP address. You can create a persistent connection for all packets addressed to 176.17.1.x to 176.18.1.1 by entering this command in SFRouter’s routing table:

`route -p add 176.17.1.0 MASK 255.255.255.0 176.18.1.1`