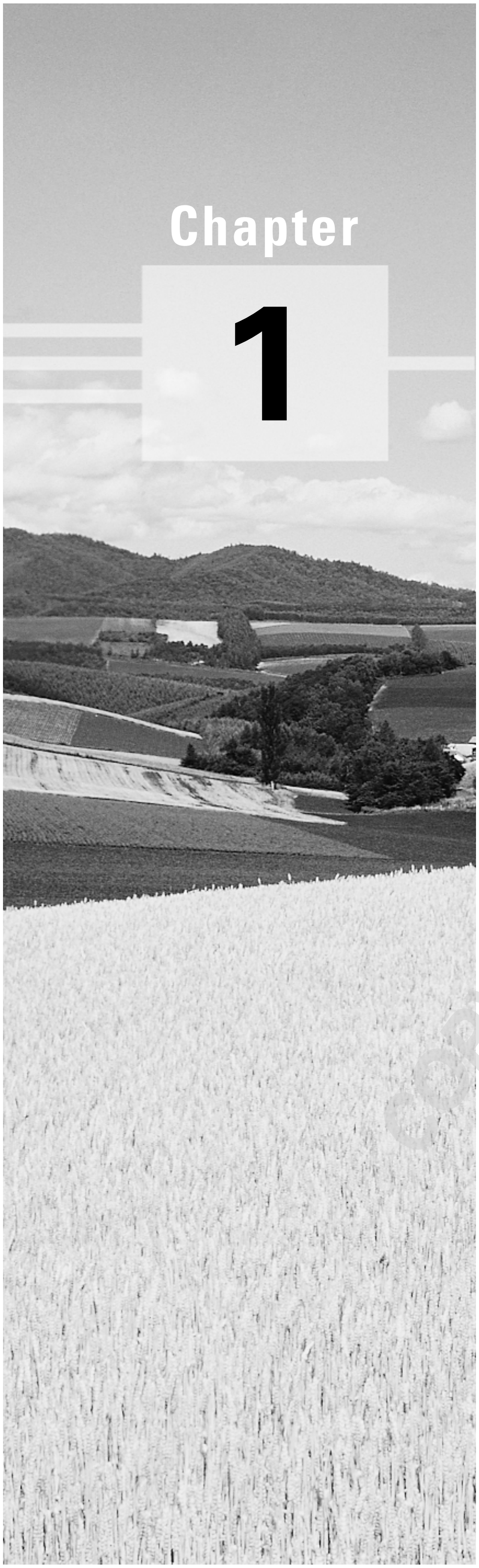
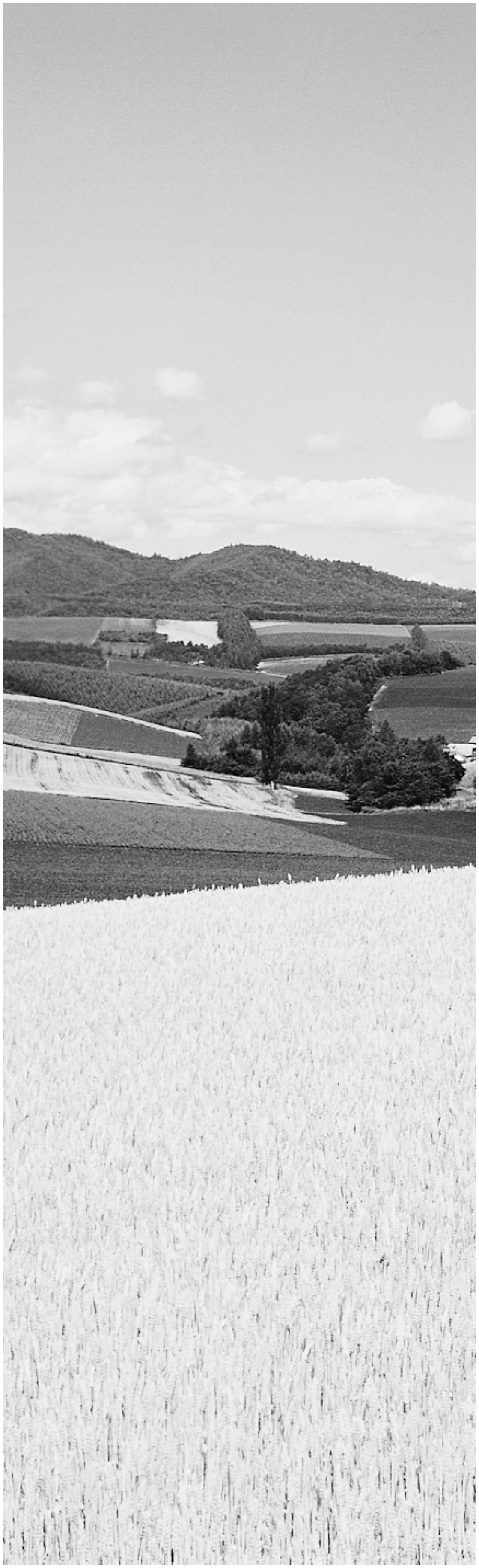


Installing and Configuring Network Protocols

MICROSOFT EXAM OBJECTIVES COVERED IN THIS CHAPTER:

- ✓ **Troubleshoot routing. Diagnostic utilities include the `tracert` command, the `ping` command, and the `ipconfig` command.**
  - Validate local computer configuration by using the `ipconfig`, `arp`, and `route` commands.
  - Validate network connectivity by using the `tracert`, `ping`, and `pathping` commands.
- ✓ **Configure and troubleshoot TCP/IP on servers and client computers. Considerations include subnet masks, default gateways, network IDs, and broadcast addresses.**
  - Configure client computer TCP/IP properties.
  - Validate client computer network configuration by using the `winipcfg`, `ipconfig`, and `arp` commands.
  - Validate client computer network connectivity by using the `ping` command.
- ✓ **Configure, administer, and troubleshoot DHCP on servers and client computers.**
  - Detect unauthorized DHCP servers on a network.
  - Configure authorization of DHCP servers.
  - Configure client computers to use dynamic IP addressing.
  - Configure DHCP server properties.
  - Create and configure a DHCP scope.



- ✓ **Troubleshoot name resolution on client computers. Considerations include WINS, DNS, NetBIOS, the Hosts file, and the Lmhosts file.**
  - Configure client computer name resolution properties.
  - Troubleshoot name resolution problems by using the `nbtstat`, `ipconfig`, `nslookup`, and `netdiag` commands.
  - Create and configure a Hosts file for troubleshooting name resolution problems.
  - Create and configure an Lmhosts file for troubleshooting name resolution problems.

NOTE

Only the "Configure client computers to use dynamic IP addressing" subobjective under the "Configure, administer, and troubleshoot DHCP on servers and client computers objective is covered in this chapter. The subobjectives related to DHCP are covered in Chapter 2, "Managing the Dynamic Host Configuration Protocol." In addition, the `nbtstat` and `nslookup` commands, as well as the Hosts and Lmhosts files, are covered in Chapter 3, "Windows 2000 Name Resolution."

Network protocols are the most fundamental part of the network that you will need to know for the exam. With the release of Windows 2000, Microsoft heavily endorsed the ubiquitous TCP/IP protocol. NetBIOS, implemented with NetBEUI in Windows 2000, is an aging protocol that is nonetheless still supported by Microsoft. You will need to understand both of these protocols in order to pass the exam. But before you dive into the specifics of network protocols, you should understand what protocols are and why you need to know how they work. The first section of this chapter gives you an overview of protocols in general. Then you will learn about the specific protocols that are typically used in Windows 2000. Finally, you will learn how to install, configure, and troubleshoot TCP/IP, which is the most commonly used and accepted network protocol today.

How Protocols Work

A *protocol* is a set of basic steps that both parties (or computers) must perform in the right order. For instance, for one computer to send a message to another computer, the first computer must perform the steps given in the following general example:

1. Break the data into small sections called packets.
2. Add addressing information to the packets, identifying the destination computer.
3. Deliver the data to the network card for transmission over the network.

The receiving computer must perform the inverse of these steps:

1. Accept the data from the network adapter card.
2. Remove the transmitting information that was added by the transmitting computer.
3. Reassemble the packets of data into the original message.

Each computer needs to perform these steps, in the same way and in the correct order, so that the data will arrive and be reassembled correctly. If one computer uses a protocol with different steps or even the same steps with different parameters (such as different sequencing, timing, or error correction), the two computers won't be able to communicate with each other.

Network Packets

Networks primarily send and receive small chunks of data called *packets*. Network protocols construct, modify, and disassemble packets as they move data across the network. Packets have the following components:

- A source address specifying the sending computer
- A destination address specifying where the packet is being sent
- Instructions that tell the computer how to pass the data along
- Reassembly information (if the packet is part of a longer message)
- The data to be transmitted to the remote computer (often called the *packet payload*)
- Error-checking information to ensure that the data arrives intact

These components are assembled into slightly larger chunks; each packet contains three distinct parts, listed below, and each part contains some of the components listed above.

Header A typical header includes an alert signal to indicate that the data is being transmitted, source and destination addresses, and clock information to synchronize the transmission.

Data This is the actual data being sent. It can vary (depending on the network type) from 48 bytes to 4 kilobytes.

Trailer The content of the trailer (or even the existence of a trailer) varies among network types, but it typically includes a cyclic redundancy check (CRC). The CRC helps the network determine whether a packet has been damaged in transmission.

Network Protocols and Windows 2000

Microsoft networking products come with four network transports, and each is intended for networks of different sizes with different requirements. Each network transport has various strengths and weaknesses. In general, *NetBEUI* is intended for small, single-server networks. *NWLink* is intended for networks that require access to Novell NetWare file servers. AppleTalk's primary use is interoperating with Macintosh computers (a topic that's too specialized to discuss further here). TCP/IP is a complex transport sufficient for globe-spanning networks such as the Internet, and Microsoft is doing everything possible to position TCP/IP as a one-size-fits-all network protocol. In Windows 2000, TCP/IP is required to use Active Directory and is the default protocol for Windows 2000.

Microsoft Exam Objective

Troubleshoot routing. Diagnostic utilities include the `tracert` command, the `ping` command, and the `ipconfig` command.

Configure and troubleshoot TCP/IP on servers and client computers. Considerations include subnet masks, default gateways, network IDs, and broadcast addresses.

NetBEUI

NetBEUI stands for NetBIOS Enhanced User Interface. (NetBIOS, in turn, stands for Network Basic Input Output System. NetBEUI implements the NetBIOS Frame (NBF) transport protocol, which was developed by IBM in the mid-1980s to support LAN workgroups under OS/2 and LAN Manager.

When IBM developed NetBEUI, they didn't intend for it to allow networked PCs to have enterprise-wide connectivity. Instead, NetBEUI was developed for workgroups of 2 to 200 computers. NetBEUI traffic can't be routed between networks, so it's constrained to small local area networks consisting of relatively small numbers of clients and servers.

NetBEUI has a number of advantages, including these:

- It's fast on small networks, because it has very low overhead.
- It's easy to set up and implement.
- It's largely self-tuning.

NetBEUI has some drawbacks, too:

- NetBEUI cannot be routed between networks. This makes it totally unsuitable for large-scale networks.
- There are few management or maintenance tools for NetBEUI, which makes it difficult to troubleshoot.
- NetBEUI offers very little cross-platform support.
- Microsoft is trying to do away with NetBEUI in favor of TCP/IP.
- NetBEUI consumes an inordinate amount of network bandwidth.

Because it's not widely used, there is—outside the realm of Microsoft operating systems—very little software available to help you analyze NetBEUI problems. However, there's an alternate flavor of NetBEUI called *NBT* (which stands for NetBIOS over TCP/IP). NBT is routable, and because it uses TCP/IP as its transport, it gains all the advantages of TCP/IP. However, Microsoft is trying to kill off NBT, too.

NWLink

NWLink is Microsoft's implementation of Novell's IPX/SPX protocol stack, which is used in Novell NetWare. In fact, it's fair to say that NWLink is nothing more than IPX for Windows NT. IPX is the protocol; NWLink is the networking component that implements it.

IPX is included with Windows 2000 primarily to allow Windows 2000 clients and servers to interconnect with older Novell NetWare servers and clients. Microsoft clients and servers can then be added to existing network installations, over time easing the migration between platforms and obviating the need for a complete cutover from one networking standard to another. (IPX is also a popular protocol for networked games, guaranteeing its appearance in future Microsoft operating systems for some years to come.)

The advantages of NWLink include the following:

- It's easy to set up and manage.
- It's routable.
- It's easy to connect to installed NetWare servers and clients.

NWLink provides a reasonable middle ground between the simple, nonroutable NetBEUI transport protocol and the complex, routable TCP/IP protocol. Like NetBEUI, IPX has many self-tuning characteristics, and it requires little administrative knowledge or skill to set up. However, NWLink has some disadvantages, such as these:

- It is difficult to exchange traffic with other organizations that aren't using IPX/SPX.
- It has limited support in Windows 2000.
- It doesn't support standard network management protocols.

Truly large networks (networks that connect many organizations) may find that NWLink is difficult to work over IPX, because there is no effective central IPX addressing scheme—as there is with TCP/IP—to ensure that two networks don't use the same address numbers. IPX doesn't support the wide range of network management tools available for TCP/IP.

NOTE

You do not need to know how to work with NWLink for the MCSA exam. However, it is useful to understand why Microsoft included NWLink with Windows 2000.

TCP/IP

TCP/IP is actually two sets of protocols bundled together: the *Transmission Control Protocol (TCP)* and the *Internet Protocol (IP)*. TCP/IP, and a suite of related protocols, was developed by the Department of Defense's Advanced Research Projects Agency (ARPA, or later DARPA) beginning in 1969. Their original goal was to develop network protocols that were robust enough to route around damage caused by nuclear war. Happily, that design goal was never tested, but some aspects of that design have led to the redundant, distributed whole we call the Internet.

TCP/IP is by far the most widely used protocol for interconnecting computers, and it is the protocol of the Internet. This is because, although ARPA originally created TCP/IP to connect military networks together, it provided the protocol standards to government agencies and universities free of charge. The academic world leapt at the chance to use a robust protocol to interconnect their networks, and the Internet was born. Many organizations and individuals collaborated to create higher-level protocols for everything from newsgroups, mail transfer, and file transfer to printing, remote booting, and even document browsing.

To support NetBIOS over TCP/IP, Microsoft has included NBT. If you're already using a TCP/IP network, supporting NBT allows older clients to use NetBIOS-based services without actually allowing any NetBEUI traffic across your network.

TCP/IP is currently the protocol king because of its rapid and widespread adoption. It also brings some significant advantages to the table, including the following:

- Broad connectivity among all types of computers and servers, including direct access to the Internet
- Strong support for routing, using a number of flexible routing protocols
- Support for advanced name and address resolution services (which will be covered in more depth in the next chapter): the Domain Name Service (DNS), the Dynamic Host Configuration Protocol (DHCP), and the Windows Internet Name Service (WINS)
- Support for a wide variety of Internet-standard protocols, including protocols for mail transport, web browsing, and file and print services
- Centralized network number and name assignment, which facilitates internetworking among organizations

If you have a network that spans more than one metropolitan area, or if you want to connect to (or over) the Internet, you'll need to use TCP/IP. It's not fast or easy to use, but it can carry an immense amount of payload and it's mechanically very robust.

TCP/IP also has some disadvantages:

- It's harder to set up than NetBEUI or IPX.

- Its routing and connectivity features impose relatively high overhead.
- It's slower than IPX and NetBEUI.

Even given these disadvantages, we'll all have to learn to live with TCP/IP, since it's the core protocol that Windows 2000 depends on for all its network services.

Understanding IP Addressing

Understanding IP addressing is critical to understanding how IP routing works. An IP address is a numeric identifier assigned to each machine on an IP network. It designates the location of the device it is assigned to on the network. This type of address is a software address, not a hardware address, which is hard-coded into the machine or network interface card.

We're going to assume you're comfortable with binary notation and math for the remainder of this section. You will need this knowledge for the exam.

The Hierarchical IP Addressing Scheme

An IP address is made up of 32 bits of information. These bits are divided into four sections (sometimes called octets or quads) containing one byte (8 bits) each. There are three methods for specifying an IP address:

- Dotted-decimal, as in `130.57.30.56`
- Binary, as in `10000010.00111001.00011110.00111000`
- Hexadecimal, as in `82 39 1E 38` (rarely used)

All of these examples represent the same IP address.

The 32-bit IP address is a structured or *hierarchical address,* as opposed to a flat or nonhierarchical one. Although IP could have used either flat or hierarchical addressing, its designers chose hierarchical addressing—for a very good reason, as it turns out.

The good news about flat addressing is that it can handle a large number of addresses, namely 4.3 billion (a 32-bit address space with two possible values for each position—either zero or one—giving you 2^{32}, which equals approximately 4.3 billion). The bad news—and the reason why flat addressing isn't used in IP—relates to routing. If every address were totally unique, every router on the Internet would need to store the address of each and every *other* machine on the Internet. It would be fair to say that this would make efficient routing impossible, even if only a fraction of the possible addresses were used.

The solution to this dilemma is to use a hierarchical addressing scheme that breaks the address space into ordered chunks. Instead of treating the entire 32 bits as a unique identifier, one part of the IP address is designated as the *network address* and the other part as a *node address*, giving it a layered, hierarchical structure.

The Network Address The network address uniquely identifies each network. Every machine on the same network shares that network address as part of its IP address. In the IP address `130.57.30.56`, for example, the `130.57` is the network address.

The Node Address The node address is assigned to, and uniquely identifies, each machine on a network. This part of the address must be unique because it identifies a particular machine—an individual, as opposed to a network that is a group. This number can also be referred to as a host address. In the sample IP address `130.57.30.56`, the `.30.56` is the node address.

The designers of the Internet decided to create classes of networks based on network size. For the small number of networks possessing a very large number of nodes, they created the Class A network. At the other extreme is the Class C network, reserved for the numerous networks with a small number of nodes. The class distinction for networks in between very large and very small is predictably called the Class B network. How you would subdivide an IP address into a network and node address is determined by the class designation of your network. Table 1.1 provides a summary of the three classes of networks, which will be described in more detail in the following sections.

TABLE 1.1 Network Address Classes

Class	Leading Bit Pattern	Decimal Range of First Byte of Network Address	Maximum Number of Networks	Maximum Nodes per Network
A	0	1–127	127	16,777,214
B	10	128–191	16,384	65,534
C	110	192–223	2,097,152	254

To ensure efficient routing, Internet designers defined a mandate for the leading bits section of the address for each different network class. For example, since a router knows that a Class A network address always starts with a zero, the router might be able to speed a packet on its way after reading only the first bit of its address. Table 1.1 illustrates how the leading bits of a network address are defined.

Some IP addresses are reserved for special purposes and shouldn't be assigned to nodes by network administrators. Table 1.2 lists the members of this exclusive little club, along with their reason for being included in it.

TABLE 1.2 Special Network Addresses

Address	Function
Network address of all zeros	Interpreted to mean "this network."
Network address of all ones	Interpreted to mean "all networks."
Network address 127	Reserved for loopback tests. Designates the local node and allows that node to send a test packet to itself without generating network traffic.
Node address of all zeros	Interpreted to mean "this node."
Node address of all ones	Interpreted to mean "all nodes" on the specified network; for example, `128.2.255.255` means "all nodes" on network `128.2` (Class B address).
Entire IP address set to all zeros	Used to designate the default route.
Entire IP address set to all ones (same as 255.255.255.255)	Broadcast to all nodes on the current network; sometimes called an "all ones broadcast."

Subnetting a Network

If an organization is large and has numerous computers, or if its computers are geographically dispersed, it makes good sense to divide a colossal network into smaller ones connected by routers. These smaller nets are called *subnets*. The benefits to using subnets include the following:

- Reduced network traffic: We all appreciate less traffic of any kind, and so do networks. Without routers, packet traffic could choke the entire network. With routers, most traffic stays on the local network—only packets destined for other networks pass through the router and over to another subnet. This traffic reduction also improves overall performance.
- Simplified management: It's easier to identify and isolate network problems in a group of smaller interconnected networks than within one gigantic one.

One problem with the original IP addressing scheme is that a single network address can be used to refer to multiple physical networks. An organization can request individual network addresses for each of its physical networks. If these requests were granted, there wouldn't be enough addresses to go around. Another problem relates to routers—if each router on the Internet needed to know about every physical network, routing tables would be impossibly huge. There would be an overwhelming amount of administrative overhead to maintain those tables, and the resulting physical overhead on the routers would be massive (CPU cycles, memory, disk space, and so on). Because routers exchange routing information with each other, an additional, related consequence is that a terrific overabundance of network traffic would result.

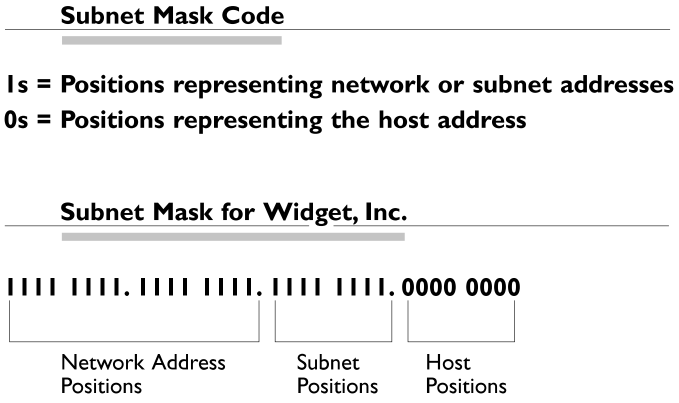
Although there's more than one way to approach this tangle, the principal solution is the one that we'll cover in this section—*subnetting*. As you might guess, *subnetting* is the process of carving a single IP network into smaller logical subnetworks. This trick is achieved by subdividing the host portion of an IP address to create something called a *subnet address*. The actual subdivision is accomplished through the use of a *subnet mask*.

SUBNET MASKS

For the subnet address scheme to work, every machine on the network must know which part of the host address will be used as the subnet address. This is accomplished by assigning each machine a subnet mask.

The network administrator creates a 32-bit subnet mask comprised of ones and zeros. The ones in the subnet mask represent the positions that refer to the network or subnet addresses. The zeros represent the positions that refer to the host part of the address. This combination is illustrated in Figure 1.1.

FIGURE 1.1 The subnet mask revealed


Subnet Mask Code
1s = Positions representing network or subnet addresses
0s = Positions representing the host address
Subnet Mask for Widget, Inc.
1111 1111. 1111 1111. 1111 1111. 0000 0000
Network Address Positions
Subnet Positions
Host Positions

In order to subnet a Class B network, for example, the first two bytes of the subnet mask are ones, formatted as `Net.Net.Node.Node`. The third byte, normally assigned as part of the host address, is now used to represent the subnet address. Hence, those bit positions are represented with ones in the subnet mask. The fourth byte is the only part in our example that represents the unique host address.

The subnet mask can also be expressed using the decimal equivalents of the binary patterns. The binary pattern of `1111 1111` is the same as decimal `255`. Consequently, the subnet mask in our example can be denoted in two ways, as shown in Figure 1.2.

FIGURE 1.2 Different ways to represent the same mask

Subnet Mask in Binary: 1111 1111. 1111 1111. 1111 1111. 0000 0000

Subnet Mask in Decimal: 255 . 255 . 255 . 0

(The spaces in the above example are only for illustrative purposes.
The subnet mask in decimal would actually appear as 255.255.255.0.)

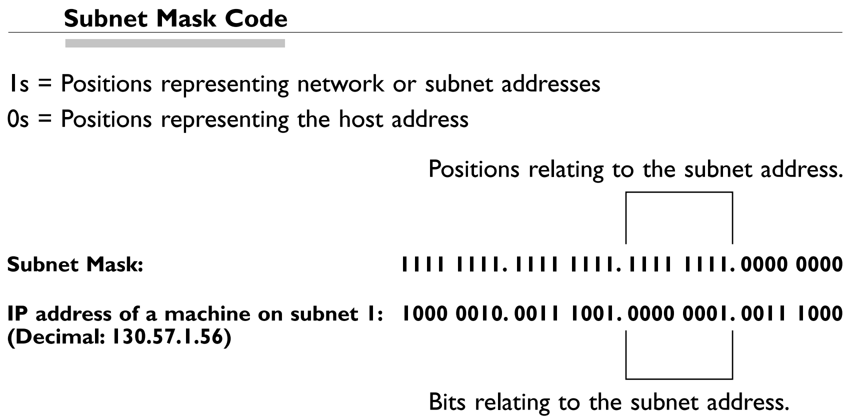
Not all networks need to have subnets and therefore don't need to use subnet masks. In this event, they are said to have a *default subnet mask*. This is basically the same as saying they don't have a subnet address. The default subnet masks for the different classes of networks are shown in Table 1.3. (Now you know where the familiar "`255.255.255.0`" comes from!)

TABLE 1.3 Special Network Addresses

Class	Format	Default Subnet Mask
A	Net.Node.Node.Node	255.0.0.0
B	Net.Net.Node.Node	255.255.0.0
C	Net.Net.Net.Node	255.255.255.0

Once the network administrator has created the subnet mask and assigned it to each machine, the IP software applies the subnet mask to the IP address to determine its subnet address. The word "mask" carries the implied meaning of "lens" in this case—the IP software looks at its IP address through the lens of its subnet mask to see its subnet address. An illustration of an IP address being viewed through a subnet mask is shown in Figure 1.3.

FIGURE 1.3 Applying the subnet mask


Subnet Mask Code
1s = Positions representing network or subnet addresses
0s = Positions representing the host address
Positions relating to the subnet address.
Subnet Mask:
1111 1111. 1111 1111. 1111 1111. 0000 0000
IP address of a machine on subnet 1:
(Decimal: 130.57.1.56)
1000 0010. 0011 1001. 0000 0001. 0011 1000
Bits relating to the subnet address.

In this example, the IP software learns through the subnet mask that, instead of being part of the host address, the third byte of its IP address is now going to be used as a subnet address. The IP software then looks at the bit positions in its IP address that correspond to the mask, which are 0000 0001.

The final step is for the subnet bit values to be matched up with the binary numbering convention and converted to decimal, as illustrated in Figure 1.4.

FIGURE 1.4 Converting the subnet mask to decimal

Binary Numbering Convention

Position / Value:	◀— (continued)	128	64	32	16	8	4	2	1
Widget third byte:		0	0	0	0	0	0	0	1
Decimal Equivalent:									0 + 1 = 1
Subnet Address:									1

By using the entire third byte of a Class B address as the subnet address, it is easy to set and determine the subnet address. If the Class B network in our example above wants to have a Subnet 6, the third byte of all machines on that subnet will be 0000 0110 (decimal 6 in binary).

Using the entire third byte of a Class B network address for the subnet allows for a fair number of available subnet addresses. One byte dedicated to the subnet provides eight bit positions. Each position can be either a one or a zero, so the calculation is 2^8, or 256. Because you cannot use the two patterns of all zeros and all ones, you must subtract two for a total of 254. Thus, our Class B network can have up to 254 total subnetworks, each with 254 hosts.

Although the official IP specification limits the use of zero as a subnet address, some products actually permit this usage. Microsoft's TCP/IP stack allows it, as does the software in most routers (provided you enable this feature). This gives you one additional subnet. However, you should not use a subnet of zero (all zeros) unless all of the software on your network recognizes this convention.

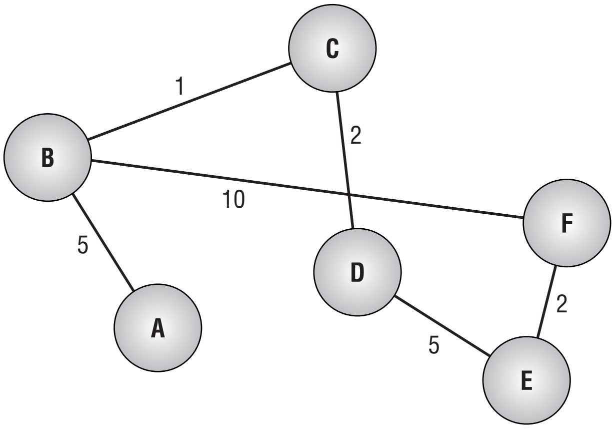
How Routing Works

In this section, we'll confine the discussion of routing theory and practice to IP routing, even though the same concepts apply to IPX/SPX and AppleTalk routing. The underlying idea is that each packet on a network has a source address and a destination address, which means that any device that receives the packet can inspect its headers to determine where it came from and where it's going. If such a device also has some information about the network's design and implementation—like how long it takes packets to travel over a particular link—it can intelligently change the routing to minimize the total cost, which refers to the time it takes a packet to travel from the source to the destination.

NOTE Although Windows 2000 supports routing IPX, AppleTalk, and IP, only IP routing will be covered in this section. It's the most widely used, and it's the one you're most likely to see on the test.

Figure 1.5 shows an imaginary network made up of six interconnected local networks. These networks, imaginatively named A through F, are connected by links of varying speeds and costs. This accurately mirrors what happens in the real world, where it's common for internal networks (or Internet providers) to have multiple ways to establish a link between two points.

FIGURE 1.5 An example network



Imagine that a client machine on network B wants to send traffic to a machine on network E. The most obvious route would probably be B to F to E, but you could also use B to C to D to E. Notice the costs: B-F-E has a total cost of 12, while the seemingly longer B-C-D-E actually has a lower cost of

8! That doesn't appear to make sense, since the latter route has a longer path. When you consider what "cost" really means, though, things get better. Assigning link costs is entirely up to you. Normally, you assign costs that reflect your preference for how you want traffic to flow. An expensive or slow link would probably deserve a higher cost than a cheaper or faster link; by assigning your most-expensive links (say, a metered ISDN connection) a high cost, you'd make them too expensive to use if there were less-expensive links available.

Now, revisit Figure 1.5 with the assumption that each circle is really a router. After all, you can hide all the complexity of the network behind a router, since only the router is in charge of moving packets. Call your client machine "X" and your server "Y." When X wants to send traffic to Y, it already knows the destination IP address of its target. X will build a packet, including its IP address as the source and Y's address as the destination. X will then use its default gateway setting to send that packet to router B.

NOTE Technically, a gateway and a router are two different things. However, Microsoft uses the terms interchangeably, and so will we.

Router B receives the packet and has both source and destination address information. By examining the IP addresses, it can determine that it doesn't "know" a direct route to the network where Y is located. However, there are two intermediate nodes that claim to know how to reach Y: C and F. Since C has the lowest link cost, the router at B will send the packet to C in a simple routing algorithm. When C receives it, it will go through the same process, forwarding the packet on to D, and so on. Eventually, the packet gets where it's going.

Static Routing

Static routing systems make no attempt to discover other routers or systems on their networks. Instead, you tell the routing engine how to get data to other networks; specifically, you tell it which other networks are reachable from your network by specifying their network addresses and subnet masks, along with a metric for that network. This information goes into the system's routing table, a big list of known routes to other networks. When an outgoing packet arrives at the routing engine, the engine can examine the routing table to select the lowest-cost route to the destination. If there's no explicit entry in the routing table for that network, the packet goes to the default gateway, which is then entrusted with getting the packet where it needs to go.

Static routing is faster and more efficient than dynamic routing. Static routing works well when your network doesn't change much. You can identify the remote networks to which you want to route and then add static routes to them to reflect the costs and topology of your network. In Windows 2000, you maintain static routes with the `route` command, which allows you to either see the contents of the routing table or modify it by adding and removing static routes to individual networks. The `route` command is explained in detail later in this chapter.

Dynamic Routing

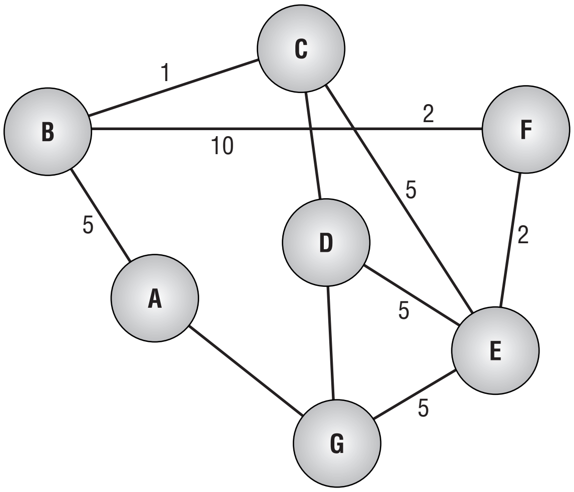
By contrast with static routing, *dynamic routing* doesn't depend on your adding fixed, unchangeable routes to remote networks. Instead, a dynamic routing engine can discover its surroundings by finding and communicating with other nearby routers in an internetwork.

This process, usually called *router discovery,* enables a newly added (or rebooted) router to configure itself. This is roughly equivalent to the process that happens when you move into a new neighborhood. Within a short time of your arrival, you'll probably meet most of the people who live nearby, either because they come to you or because you go to them. At that point, you have useful information about the surrounding environment that could come only from people who were already there.

The two major dynamic routing protocols in Windows 2000 are the Routing Information Protocol (RIP) and the Open Shortest Path First (OSPF) protocol. Each has its advantages and disadvantages, but they share some common features and functionality. Each router (whether a hardware device, a Windows 2000 machine, or whatever) is connected to at least two separate physical networks. When the router starts, the only information it has is drawn from its internal routing table. Normally, that means it knows about all the attached networks plus whatever static routes have been previously defined. The router then receives configuration information that tells it about the state and topology of the network.

As time goes on, the network's physical topology can change. For example, take a look at the network in Figure 1.6. If network G suddenly dropped out of the air, the routers in sites A, D, and E would need to readjust their routing tables since they could no longer route traffic directly to G. The process by which this adjustment happens is what makes routing dynamic, and it's also the largest area of difference between the two major dynamic routing protocols for IP.

FIGURE 1.6 A more complex, dynamically routed network



Installing Network Protocols

Windows 2000 supports a wide range of network protocols, both in the set provided with Windows 2000 itself and from third-party vendors. In brief, any vendor who wants to write an NDIS-compatible driver can do so; in theory, any network protocol could potentially have a Windows 2000 version. Microsoft ships protocol stacks for TCP/IP, NetBEUI, Novell's IPX/SPX (which Microsoft calls NWLink), AppleTalk, and DLC. You install or remove all of these protocols using the same interface; once you actually install the protocol, you will still have to configure it.

Installation Basics

You install network protocols through the Local Area Connection Properties dialog box, which lists all of the known protocols on your Windows 2000 machine. Protocols marked with a check indicate that they're bound to the adapter whose properties you're inspecting. Figure 1.7 shows an example of what this dialog box might look like on your machine.

Microsoft Exam Objective

Configure and troubleshoot TCP/IP on servers and client computers. Considerations include subnet masks, default gateways, network IDs, and broadcast addresses.

- Configure client computer TCP/IP properties.

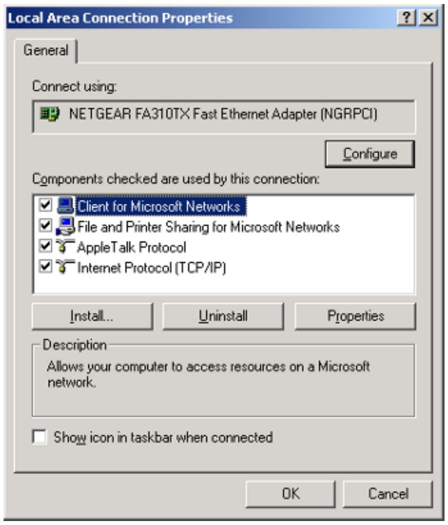
Configure, administer, and troubleshoot DHCP on servers and client computers.

- Configure client computers to use dynamic IP addressing.

Troubleshoot name resolution on client computers. Considerations include WINS, DNS, NetBIOS, the Hosts file, and the Lmhosts file.

- Configure client computer name resolution properties.

FIGURE 1.7 The Local Area Connection Properties dialog box shows you which protocols are already installed.



Connect using:
NETGEAR FA310TX Fast Ethernet Adapter (NGRPCI)

Components checked are used by this connection:
☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☑ AppleTalk Protocol
☑ Internet Protocol (TCP/IP)

Description
Allows your computer to access resources on a Microsoft network.

☐ Show icon in taskbar when connected

Installing and Configuring TCP/IP

TCP/IP is normally installed as part of the Windows 2000 setup process. This is no accident, since Microsoft would much rather have all its Windows 2000 customers use TCP/IP than NetBIOS. If you need to install TCP/IP manually, you still can. The process for installing it is very similar to the process required to install NWLink, as you can see in Exercise 1.1.

EXERCISE 1.1

Installing TCP/IP

Follow these steps to install the TCP/IP protocol:

1. Open the Network and Dial-Up Connections folder (Start ➢ Settings ➢ Network and Dial-Up Connections).
2. Right-click the Local Area Connection icon and choose Properties. The Local Area Connection Properties dialog box appears, as shown earlier in Figure 1.7.
3. Click the Install button. The Select Network Component Type dialog box appears. Select Protocol and click the Add button.
4. The Select Network Protocol dialog box appears. Choose Internet Protocol (TCP/IP); then click the OK button.
5. If prompted, insert your Windows 2000 CD and click OK.
6. Click the Close button in the Local Area Connection Properties dialog box.

When you install TCP/IP, it defaults to using DHCP for automatic configuration. If you want to use DHCP for automatic configuration you certainly can, but it's always useful to know how to manually configure a TCP/IP connection (especially since Microsoft will be asking you to prove you know how to as part of the exam!). Now you'll see what that configuration process entails.

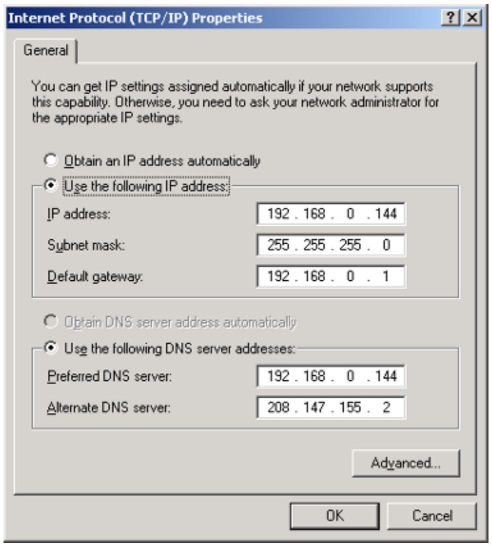
See Chapter 2 for more information on DHCP.

Configuring Basic TCP/IP Settings

If you've bought into the rap that TCP/IP is convoluted and difficult to configure, Windows 2000's basic TCP/IP Properties dialog box may surprise you. TCP/IP actually requires only two pieces of information to function: the IP address you want to use for this system and the subnet mask that corresponds to the network subnet the client is on. For example, in the Windows 2000 classes we teach, we set up an in-classroom network that doesn't connect to any outside networks, including the Internet. We don't need a *default gateway* since that's used for routing, and we don't set up DNS until later in the class. With only these two parameters, machines can communicate with each other (although not having DNS is inconvenient, since it means we have to enter IP addresses by hand).

Figure 1.8 shows the Internet Protocol (TCP/IP) Properties dialog box. You reach this dialog box by opening the Local Area Connection icon, selecting the Internet Protocol (TCP/IP), and clicking the Properties button. Of course, if you have multiple network adapters in a single computer, you can set independent TCP/IP properties for each adapter. Depending on what you want to do, you'll use either the automatic configuration buttons or the text fields.

FIGURE 1.8 The basic TCP/IP Properties dialog box



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 0 . 144

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 0 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: 192 . 168 . 0 . 144

Alternate DNS server: 208 . 147 . 155 . 2

If You Want to Use DHCP

If you're configuring a Windows 2000 Professional machine, chances are probably pretty good that you're using DHCP with it. In that case, the default TCP/IP settings will work fine for you, since they configure the TCP/IP stack to get configuration parameters from any available DHCP server. Remember that you can mix and match DHCP and non-DHCP machines; on a single client, you can use DHCP to get everything except DNS server addresses if you want to. You have two basic choices:

- To configure a client to get its TCP/IP configuration information from a DHCP server, leave the Obtain An IP Address Automatically radio button selected.
- If you're using DHCP for basic IP addressing and you want to accept DNS server addresses from the DHCP server, leave the Obtain DNS Server Address Automatically radio button selected.

If You Don't Want to Use DHCP

We recommend against using DHCP on servers, since they're not nearly as dynamic as clients. Ideally, you won't reboot servers unless they *need* it, and you won't be moving them around. Therefore, the "dynamic" in DHCP isn't really useful, and its other benefits are outweighed by the comfort that comes from knowing that your server has a correct and unchanging IP configuration. If you want to configure the TCP/IP settings yourself, start by selecting the Use The Following IP Address radio button, and then fill in the following fields:

- In the IP Address field, enter the IP address you want to use for this machine. Remember that Windows 2000 won't do any kind of sanity checking. The most common mistake people make with this field is to enter an address that doesn't match the address range they're using on their network.
- In the Subnet Mask field, enter the appropriate subnet mask for your network.
- If you want this machine to be able to route packets to other networks, enter the gateway or router address you want it to use in the Default Gateway field. Again, remember that Windows 2000 will slavishly use whatever address you enter here, so make sure it's right.
- If you're using DNS on your network, enter the first DNS server you want this client to talk to in the Preferred DNS Server field. It's critical to get this right on a Windows 2000 network since DNS is required for Active Directory services. If you want to specify another server to use

when the preferred server is unavailable or can’t resolve a DNS query, enter it in the Alternate DNS Server field. (You can also specify additional servers, as you’ll see in the following section on the Advanced TCP/IP Properties dialog box.)

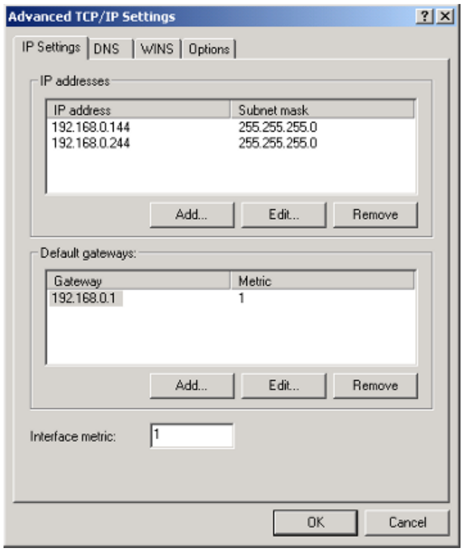
Configuring Advanced TCP/IP Settings

The Advanced button in the TCP/IP Properties dialog box brings up something that looks more like what you’d expect from TCP/IP. The Advanced TCP/IP Properties dialog box contains four tabs that let you extend and override the settings from the simpler dialog box shown in Figure 1.8.

Expanding the Basic Settings

The basic configuration dialog box you saw earlier lets you enter one IP address, one subnet mask, and one default gateway. For the majority of systems that’s enough, but what if you want to configure a machine that can communicate on multiple IP addresses? For example, if you’re setting up an Internet Information Services (IIS) server, you may want it to answer to multiple IP addresses on a single physical network connection (such as the connection that links your server to the Internet). Adding multiple IP addresses in this manner is called multi-homing. You may also want to specify multiple gateways so that an outbound packet sent by your systems can be sent to whichever gateway is “cheapest” (more on what “cheap” means in a minute). The IP Settings tab of the Advanced TCP/IP Properties dialog box allows you to do both of these things. Figure 1.9 shows what it looks like.

FIGURE 1.9 The IP Settings tab of the Advanced TCP/IP Settings dialog box



Advanced TCP/IP Settings

IP addresses

IP address	Subnet mask
192.168.0.144	255.255.255.0
192.168.0.244	255.255.255.0

Default gateways:

Gateway	Metric
192.168.0.1	1

Interface metric: 1

Your options on the IP Setting tab include the following:

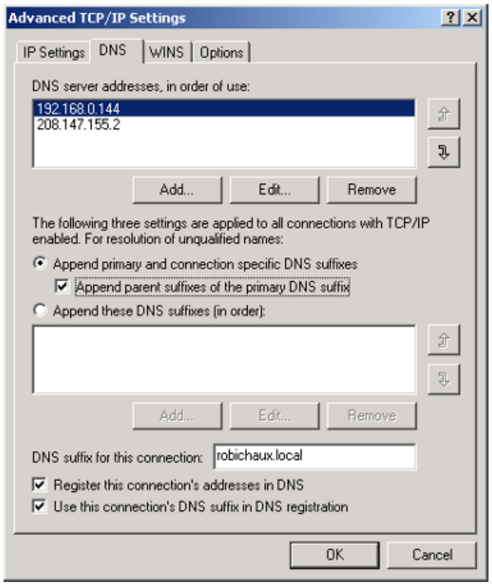
- The IP Addresses control group lists the IP addresses currently defined for this network adapter. You can add new address bindings, edit existing bindings, or remove an address with the buttons at the bottom of the control group. Once you add an address here and close all open network properties dialog boxes (including the Local Area Connection dialog box), any changes you make here take effect.
- The Default Gateways control group shows the routing gateways that are currently defined *for this computer only*. Each gateway has an IP address (to which the client sends outbound packets) and an associated metric, or cost. When deciding where to send packets bound for other networks, Windows 2000 examines its internal TCP/IP routing table to see whether it already "knows" how to get packets to the destination network. If so, it uses that route. If not, it uses the default gateway. If you specify more than one default gateway, the system chooses a gateway by selecting the one that has the lowest cost. If that gateway is down, or if it can't get packets to the destination system, Windows 2000 tries the next-most-expensive gateway. This process repeats until the packets arrive at their destination or until the system runs out of gateways to try.

Expanding DNS and WINS Settings

If all you want to do is configure your clients to use two DNS servers, you can use the Preferred and Alternate Server Configuration fields in the basic TCP/IP Properties dialog box. The DNS tab of the Advanced TCP/IP Properties dialog box allows you to specify more than two servers; in addition, you can control which DNS domain names are appended to search queries when you don't specify a fully qualified domain name. However, we want to point out the most salient feature here as well: The DNS Server Addresses, In Order Of Use field (and its associated buttons) lets you specify multiple DNS servers. When the client resolver needs an address looked up, it starts by querying the server at the top of this list and working down the list until it finds an answer or runs out of servers to query. Adding servers to this list is a quick way to improve your clients' fault tolerance, since losing the preferred and alternate DNS servers will otherwise result in a loss of DNS service to the clients. Figure 1.10 shows the DNS tab.

DNS and WINS are explained in detail in Chapter 3.

FIGURE 1.10 The DNS tab of the Advanced TCP/IP Properties dialog box



DNS server addresses, in order of use:

192.168.0.144
208.147.155.2

The following three settings are applied to all connections with TCP/IP enabled. For resolution of unqualified names:

● Append primary and connection specific DNS suffixes

☑ Append parent suffixes of the primary DNS suffix

○ Append these DNS suffixes (in order):

DNS suffix for this connection: robichaux.local

☑ Register this connection's addresses in DNS

☑ Use this connection's DNS suffix in DNS registration

Likewise, the WINS tab (not shown here) allows you to specify multiple WINS servers. In fact, the only place where you can specify which WINS servers to use is in this tab, because Microsoft is trying to move you away from using WINS (and NetBIOS!) to using pure TCP/IP and DNS. Finally, the Options tab lets you configure protocol-specific options, including whether the IP Security (IPSec) extensions are used and whether any type of packet filtering is enabled.

Before attempting Exercise 1.2 on your network, be sure to choose an IP address *not* in use by any other host or device on your network!

EXERCISE 1.2

Configuring TCP/IP Settings

Follow these steps to add a second IP address to your existing NIC. Note that this exercise assumes that you're not using DHCP on that NIC, since you can't assign additional addresses to a DHCP-enabled NIC.

1. Choose an IP address on your network that's not currently in use by another device (e.g., `169.254.0.202`). Make sure you know the correct subnet mask to use with that IP address (e.g., `255.255.255.0`).
2. Open the Network and Dial-Up Connections folder (Start ➢ Settings ➢ Network and Dial-Up Connections).
3. Right-click the Local Area Connection icon and choose Properties. The Local Area Connection Properties dialog box appears, as shown earlier in Figure 1.7.
4. Select Internet Protocol (TCP/IP) in the Components list, and then click the Properties button. The Internet Protocol (TCP/IP) Properties dialog box appears.
5. Click the Advanced button. The Advanced TCP/IP Properties dialog box appears.
6. Click the Add button in the IP Addresses control group. The TCP/IP Address dialog box appears.
7. Type in the IP address and subnet mask you chose in Step 1.
8. Click the OK button in the Advanced TCP/IP Settings dialog box.
9. Click the OK button in the Internet Protocol (TCP/IP) Properties dialog box.
10. Click the OK button in the Local Area Connection Properties dialog box.

Installing NetBEUI

What about NetBEUI? Up to now, you've been reading that Microsoft is trying its darnedest to drive a pointed wooden stake through NetBEUI and replace it with TCP/IP. Although that's a lofty goal, there are still lots of NetBEUI networks and seats out there in the world, and it's important to know how to install NetBEUI in case you ever need it on a network. The good news is that there's virtually nothing new to learn by this point, because you install NetBEUI using steps that are almost identical to what you've already done while installing NWLink and TCP/IP. In this case, there are actually three separate pieces you may need to install to use NetBEUI:

- You need the NetBEUI protocol itself, which you install using the steps outlined in Exercise 1.1. Note that NetBEUI isn't installed by default on any version of Windows 2000.
- The Client for Microsoft Networks client allows your client machine to attach to shares and printers on other servers, no matter which transport protocol you're using. This is actually the Workstation service, which you may recognize from Windows NT 4.0.
- The File and Print Sharing for Microsoft Networks service allows your machine to act as a server, sharing resources with other machines just like the Server service in Windows NT 4.0.

NetBEUI is designed to be self-tuning, so there are no properties to set for it. However, do you remember the WINS tab on the Advanced TCP/IP Properties dialog box? You can use a radio button there to turn off the use of NetBEUI over TCP/IP. As an alternative, you can remove the protocol or selectively unbind it from some or all of your network adapters.

Real World Scenario

Multiple Protocols Are Nice but Inefficient

Your company has been running Windows NT, Novell NetWare, and even some Banyan that has been floating around for years. There are also connections to old mainframe controllers that still use DLC. Over the years, the connections to these various operating systems have been accomplished piecemeal by adding the clients and protocols at each workstation. This is a common approach, particularly on networks that have grown over time as each special-interest group kept control over its piece of the network. The interoperability features of Windows 2000 (as well as Windows NT before it), specifically the ability to run multiple protocols, are phenomenal. With the NDIS (Network Driver Interface Specification) and TDI (Transport Data Interface), you can run just about as many protocols as you would like. But the ease of this functionality can also cause problems at the other end, because running multiple protocols creates multiple bandwidth consumption and multiple points of management.

Although you can't just throw out the other systems (well, maybe you can throw out the Banyan), there are other ways to approach this interoperability. With the global acceptance of TCP/IP, every major operating system today supports this protocol. This provides an opportunity to remove the other protocols on the network, such as IPX and NetBEUI, which are fading out of use. Each protocol stack brings its own overhead to the network. If you have Windows NT or Windows 9.*x* machines on your network with multiple protocols, you also have multiple instances of services that ride on top of those protocols. For example, if you're running NWLink and TCP/IP, there is a complete browser service (not the Internet kind of browser) that handles NetBIOS requests. This type of redundancy isn't efficient—it just provides another level of complexity where something can go wrong, thus adding to support efforts.

In the future, you'll see the maturity of protocol interoperability applied at the other end of the OSI stack. Using technologies such as XML and HTTP, the network client is becoming simply the browser (yes, the Internet kind) that can be used to access information across different underlying platforms. Until then, the best practice is to work toward the goal of a unified client and to minimize the number of protocols on your network, as well as the number of clients, if possible. Although the functionality is there to support more protocols and more clients, this is another case where more is not necessarily better.

The reality of today is that you will be supporting TCP/IP and for this reason you need to be able to support the basic configurations at the client. You also need to understand how a particular address relates to the network-addressing scheme for your network as a whole. You will not be able to troubleshoot an address problem if you do not understand how a particular address fits into the network and what will happen if it is not appropriate for the subnet.

Troubleshooting Network Protocols

In a perfect world, troubleshooting would never be necessary. However, in the real world, we troubleshoot things constantly to solve problems—such as, why the microwave isn't working (because lightning hit the power line last night) or why one workstation can't see others on the network (its network cable was unplugged).

Microsoft Exam Objective

Troubleshoot routing. Diagnostic utilities include the `tracert` command, the `ping` command, and the `ipconfig` command.

- Validate local computer configuration by using the `ipconfig`, `arp`, and `route` commands.
- Validate network connectivity by using the `tracert`, `ping`, and `pathping` commands.

Configure and troubleshoot TCP/IP on servers and client computers. Considerations include subnet masks, default gateways, network IDs, and broadcast addresses.

- Validate client computer network configuration by using the `winipcfg`, `ipconfig`, and `arp` commands.
- Validate client computer network connectivity by using the `ping` command.

Troubleshoot name resolution on client computers. Considerations include WINS, DNS, NetBIOS, the Hosts file, and the Lmhosts file.

- Troubleshoot name resolution problems by using the `nbstat`, `ipconfig`, `nslookup`, and `netdiag` commands.

Knowing how to effectively troubleshoot network problems is an essential part of managing even small networks, and Microsoft expects you to understand basic troubleshooting principles and how to apply them in Windows 2000 networking. Fortunately, you probably already know *what* to check; now you'll read about a set of tools that you can use to verify the proper functioning of your network. More important, you'll learn how to use those tools the right way at the right time.

Putting the Problem into Perspective

When someone complains that their network is broken, your first impulse should be to ask, "Well, what changed?" This might seem weird, but it's actually very practical. If a system is working and then it stops working, obviously something has changed somewhere—either as the result of an explicit change or by accident. Once you can identify what has, or has not, changed, you're ready to start looking for effects of the change and ways to fix whatever's gone wrong.

For example, one of the servers in our home office is a reliable old Intergraph TD-30. Even though it's been running Windows 2000 Advanced Server since very early in the beta cycle, it just keeps trucking along, never giving us any trouble. It happens to have a front-mounted power switch. When we can't contact it, our first suspicion is that it's been powered off.

The first sign of network trouble is usually pretty obvious, too: One machine can't talk to another. Using the above example, if we look at the back panel of *hawk*, our primary Windows 2000 Server, we can see that its NIC has some LEDs that indicate network activity. Just because we see those lights blinking, that doesn't tell us anything about what kind of network data is being carried by the Transport, Application, Session, or Presentation layers of the OSI model—all it tells us is that the Physical-layer components are sending and receiving *something*.

Breaking Down the Problem into Manageable Chunks

You can often save yourself a lot of unnecessary time and effort when troubleshooting a problem by doing something simple: stopping to think. It's hard to keep your wits about you when something's wrong with your network and end users are clamoring for your head on a stick, but if you can clearly identify the problem source, you're well on your way to being able to effectively resolve it without any time-wasting detours.

What Kind of Problem Is It?

Sometimes this can be the most frustrating part of troubleshooting. Getting a phone call or a pager message that says, "The network is down," doesn't tell you much. Is it your connection to the Internet? Your e-mail server? A file server somewhere on your LAN? Without knowing what specific service or connection is unavailable, you won't know what to start fixing.

Some types of problems immediately suggest a solution. For example, if a client calls us and says they get DNS errors when trying to connect to websites, our first two thoughts are that someone's changed their DNS settings or that their DNS servers are down. Likewise, if a user reports a problem reaching a particular share on a server, the problem may be on the client, the server, or the intervening network. If you can, arm yourself with as many details about how the problem is manifesting itself (including exact error messages), when it started, and whether or not it's consistent before you try to figure out what the problem is. Knowing these things beforehand can guide you to an easy, quick solution if it's a problem you've seen and fixed before—but only if you *know* you've seen it before!

Who's Having the Problem?

Knowing which users or computers are affected by a problem is very important since that gives you insight into possible causes (including user mistakes) and helps you select a course of action.

If One User Reports a Problem

If one user on our network has a problem, our experience has been that more often than not the problem stems from some change the user made. Windows contains a lot of interrelated components, and it's not evident to most people that a simple change they made in component A may have unexpected side effects on component B. When troubleshooting an end-user problem, your first question should always involve whether or not they've changed anything on the machine. This includes changing Control Panel settings, installing or removing software, rebooting, or any other action that might have directly or indirectly changed the state of the machine. If you can find out what's changed, that will give you a list of potential places to start looking.

For example, in a class we recently taught, one user complained that he could no longer see the network after doing one of the labs. As it turned out, he had turned his machine into a DHCP server while experimenting, which meant that his previously assigned DHCP address could no longer be used. He'd picked another IP address at random, which turned out not to work with our classroom network configuration. Knowing what changed let us pinpoint and fix the problem quickly.

If Several Users Report the Same Problem

Multiuser troubleshooting is, paradoxically, both easier and harder than single-user troubleshooting. Most of the time, one user can't change anything that will affect other users on the network, so you generally don't have to worry about that variable. On the other hand, the kinds of changes that can accidentally affect connectivity for many users at once are more likely to be things *you've* changed. The first step in fixing this kind of problem is identifying its scope. Is everyone on the network affected? Are only people in one workgroup or on one floor of a building affected? Is the problem limited to the lack of one key service (like DNS), or is all network traffic hosed? Answering this type of question helps you isolate where the problem's occurring so you can concentrate your efforts on that area.

Verifying Physical Connectivity

Physical-layer connectivity is absolutely critical. If you don't have a physical connection to the network you want to talk to, how can you send packets to it? This might seem like an obvious question to ask, but the number of times that we've seen people forget to ask it and look for a more complex—and ultimately nonexistent—problem would boggle your mind. So, when you first notice a network problem, be sure to verify that all of your network cables are correctly connected; that your hub, router, or switch has power; and so on. Take a look at the activity or "heartbeat" lights on your NIC, hub, or switch to see whether the Physical layer is reporting any type of activity.

Knowing something about the scope of the outage helps, too—if all your users begin complaining at once that the network is down, it's unlikely to be the fault of one user's network cable. Contrariwise, if a single user is having trouble with the network, it's unlikely that a router or switch is to blame. If you've properly identified exactly who's having trouble, that may suggest a cause based on your knowledge of the physical topology. When we worked at Intergraph, for example, we used wall-mounted routers that plugged into ordinary power outlets. Every so often, someone on the cleaning crew would unplug a router at night to plug in a floor polisher or vacuum cleaner and then forget to plug it back in. Result: No one in that area would have network access the next morning!

If you verify that all of the physical connections are okay, with power and cabling all in good order, and you *still* have no connectivity, it indicates that the problem probably resides within a higher layer.

Using *Ipconfig, Winipcfg, and Arp* to See What's What

If your problem persists even after you've verified that the physical aspects of the network are all in order, you will need to dig a little deeper using some of the tools that are built into Windows 2000. `Ipconfig`, `winipcfg`, and `arp` are tools you can use to verify that the local computer's configuration information is correct. You might spot a problem in the output of one of these commands and quickly resolve the issue with a simple configuration change.

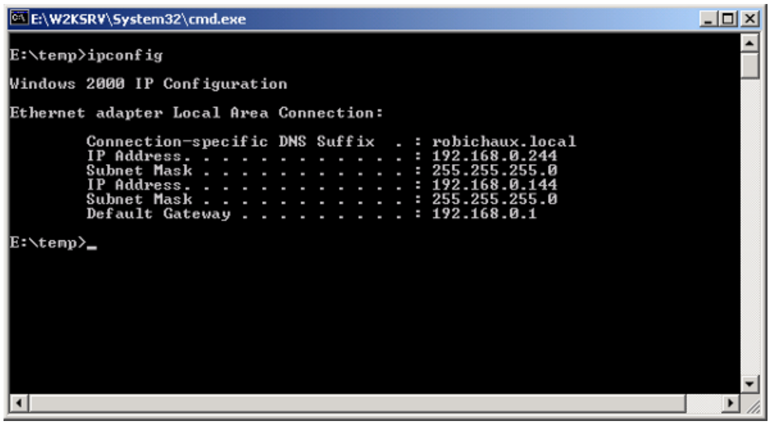
Ipconfig

Windows 2000 includes a useful tool called ipconfig. As its name implies, it's used to see the configuration of TCP/IP interfaces on your local machine. Typing **`ipconfig`** into a Windows 2000 command-prompt window presents you with a neat summary of your current IP configuration, including the local DNS name, the IP addresses, and the subnet masks configured for all adapters on the computer. Figure 1.11 shows an example of this output. You can use `ipconfig` in this mode to get a quick snapshot of its IP configuration, even if it's using DHCP. For example, if you see that the problem machine has no IP address assigned, even though there's a DNS server, it suggests the possibility that the DHCP server isn't authorized in Active Directory.

NOTE

The `ipconfig` command does not work on Windows 95/98/Me computers. Use the `winipcfg` command instead, with the same switches and parameters that you would use with `ipconfig`.

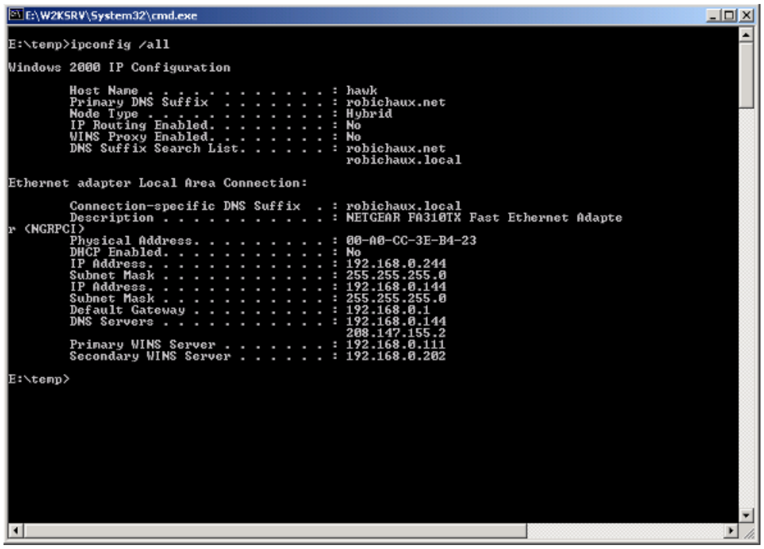
FIGURE 1.11 The output from *ipconfig*



```
E:\temp>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : robichaux.local
        IP Address. . . . . . . . . . . . : 192.168.0.244
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        IP Address. . . . . . . . . . . . : 192.168.0.144
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.0.1

E:\temp>_
```

In addition to the DHCP-related switches that you'll see in Chapter 2, there's another switch of interest to troubleshooters: `/all`. As you might expect, adding the `/all` switch causes `ipconfig` to spill its guts and display everything it knows about the current IP configuration on all installed adapters. In addition to the DNS information and IP address that it ordinarily displays, you'll also get the MAC address of each NIC, the present WINS configuration (if any), and the IP addresses being used for the preferred and alternate DNS servers. Figure 1.12 shows an example of the `ipconfig /all` output.

FIGURE 1.12 The output from *ipconfig /al*l



```
E:\temp>ipconfig /all

Windows 2000 IP Configuration

        Host Name . . . . . . . . . . . . : hawk
        Primary DNS Suffix  . . . . . . . : robichaux.net
        Node Type . . . . . . . . . . . . : Hybrid
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No
        DNS Suffix Search List. . . . . . : robichaux.net
                                            robichaux.local

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : robichaux.local
        Description . . . . . . . . . . . : NETGEAR FA310TX Fast Ethernet Adapte
r (NGRPCI)
        Physical Address. . . . . . . . . : 00-A0-CC-3E-B4-23
        DHCP Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 192.168.0.244
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        IP Address. . . . . . . . . . . . : 192.168.0.144
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.0.1
        DNS Servers . . . . . . . . . . . : 192.168.0.144
                                            208.147.155.2
        Primary WINS Server . . . . . . . : 192.168.0.111
        Secondary WINS Server . . . . . . : 192.168.0.202

E:\temp>
```

What can you do with all this information? It depends. If you're familiar enough with your network to know what IP address configurations should look like, often a quick check with `ipconfig` will tell you where the problem lies. For example, you might notice that an adapter that should be DHCP-enabled isn't, or vice versa. Even if you're not familiar with the details of your network, though, knowing how to find the IP addresses and subnet masks in use on your computers can be very valuable. Exercise 1.3 demonstrates how to check configurations with `ipconfig`.

EXERCISE 1.3

Checking Configurations with *Ipconfig*

Follow these steps to run `ipconfig` and analyze its output:

1. Open a command window (Start ➤ Run; then enter **`cmd`** in the Run dialog box and click OK).
2. At the command prompt, type **`ipconfig`**. Notice that you see an abbreviated display containing the machine's connection-specific DNS suffix, its IP address, subnet mask, and default gateway.
3. Type **`ipconfig /all`**. Note that a great deal more information is displayed, including information on multiple adapters (if you have more than one).

ARP

Before diving into the specifics of the `arp` command, you should understand what a MAC address and the Address Resolution Protocol (ARP) are.

Defining the MAC Address and the ARP

Every network adapter is given a unique 12-digit hexadecimal hardware address from the factory called the MAC address. The MAC address can never be changed, so no two network cards will ever have the same hardware identifier (people have reported duplicate addresses, but this is only due to errors on the part of the hardware manufacturer).

ARP maps IP addresses to node names (i.e., MAC addresses) to IP addresses. It equates logical and physical device addresses. ARP maintains tables of name-to-address mappings and can send out discovery packets if a desired name or address is not currently in its table. The discovery packet requests that the entity corresponding to the known name or address respond with the needed information. A copy of this table is stored on the local machine for easy access in the arp cache.

A related protocol, RARP (Reverse Address Resolution Protocol), performs the same functions in reverse; that is, given a node name, it determines the corresponding IP address.

The *Arp* Command

The arp command allows you to view and modify the arp cache. You might want to do this if two computers are unable to connect to each other. You can run the arp -a command on each machine to see if they have the correct MAC address listed in cache for each other. You can determine the correct MAC address using the ipconfig /all command, explained in the preceding section. A sample output of the arp -a command is shown here:

```
C:\> arp -a
Interface 66.127.67.40 on interface 0x1000002
Internet Address      Physical Address      Type
66.127.67.41          00-10-67-00-a2-93     dynamic
66.127.67.42          00-80-ad-88-6a-79     dynamic
```

If any of the physical addresses listed in the table are not correct for that IP address, you should run the arp -d *ipaddress* command, where *ipaddress* is the IP address of the offending entry. This command deletes the entry from the cache. The next time the local machine attempts to access that IP address, it won't find an entry in the cache and will broadcast a new arp request. The new IP-to-MAC mapping will then be placed in cache. Table 1.4 lists all of the ARP switches and their functions.

TABLE 1.4 ARP Switches

Switch	Name	Function
-d ipaddress	Delete	Deletes an entry in the arp cache.
-s macaddress	Static	Adds a new static mapping to the arp cache.
-N interface IP address	Interface	Displays the contents of the arp cache for the interface specified. This is useful if you have more than one network adapter installed on your machine.

TABLE 1.4 ARP Switches *(continued)*

Switch	Name	Function
-a	Display	Displays the contents of the arp cache for all interfaces.
-g	Display	Displays the contents of the arp cache for all interfaces (yes, exactly the same as -a).

Troubleshooting Routing with the *Route* Command

As you saw earlier in the chapter, the local host maintains a routing table of all its known routes. The `route add` command allows you to add new static routes; you can choose whether these routes remain in the routing table after the system reboots. Routes that stick around in this manner are called *persistent routes*. The command itself is simple:

```
route add <destination> mask <netMask> <gateway> <metric>
<interface>
```

You specify the destination, net mask, gateway, metric, and interface name on the command line. These parameters are all required, and `route add` does some basic sanity checking to make sure that the net mask and destination match and that you haven't left anything out. One speed bump: You have to specify the interface as a number, not as a name. However, the `route print` command lists its interfaces and the associated numbers.

The `route print` command can show you all or part of the routing table from the command line. Just typing **`route print`** into a command window will give you a complete dump of the entire routing table; adding a wildcard IP address (for example, **`route print 206.151.*`**) will display only routes that match 206.151.

You can also use the `route delete` and `route change` commands to make changes to the routing table. The official syntax for the route command is `route [-f] [-p] [`*`command`*` [`*`destination`*`]] [MASK `*`netmask`*`] [`*`gateway`*`] [metric `*`metric`*`] [if `*`interface`*`]`. The switches are processed in order. Table 1.5 lists the various switches that can be used with the `route` command.

TABLE 1.5 *Route* Switches

Switch	Function
`-f`	Clears the routing table of all gateway entries.
`-p`	When used with the `add` command, this switch adds the route to the routing table and to the Windows 2000 Registry. The route is automatically added to the routing table each time TCP/IP is initialized. By default, routes added without the `-p` switch are only stored in the RAM-based IP routing table and are not preserved when TCP/IP is restarted. This option is ignored for all other commands.
`Print <destination>`	Displays a route to the host specified by destination.
`Add <destination> Mask <netmask> <gateway> Metric <metric> if <interface>`	Adds a route for the specified destination using the forwarding IP address of the gateway. The `metric` and `if` options are optional.
`Delete <destination>`	Deletes a route for the specified destination.
`Change <destination> Mask <netmask> <gateway> Metric <metric> if <interface>`	Changes an existing route.
`Mask <netmask>`	Indicates the network mask value. If you leave `<netmask>` blank, the default is `255.255.255.255`.
`Metric <metric>`	Specifies the cost to reach the destination. Lower values prioritize the route over other similar routes with higher values.
`if <interface>`	Specifies the interface to use for the route.

Using *Ping, Tracert,* and *Pathping* to Check Connectivity between Network Hosts

The next step up from Physical layer troubleshooting is tracing the route that packets take, or are attempting to take, between the source and destination. Once you've verified that all of the physical connections are in good shape, you must see whether you can send *any* type of packet between points A and B.

TCP/IP includes a protocol called the *Internet Control Message Protocol (ICMP)*. ICMP is designed to pass control and status information between TCP/IP devices. One type of ICMP packet, popularly known as a *ping* packet, tells the receiving system to send back an ICMP response. This gives you confirmation of whether or not the ICMP ping packet reached the target, which in turn tells you whether or not you can get packets from place to place. Since name resolution and application services depend on lower-level protocols, this sort of "Is this thing on?" test is the next logical step after testing the underlying physical connection. The `ping` and `tracert` tools both use ICMP to help sniff out network problems.

The *Ping* Tool

When you ping a remote computer using the `ping` utility in its default mode, your computer sends out four ICMP ping packets and measures the time required before each packet's corresponding response arrives. When it finishes, `ping` gives you a helpful summary showing the number of packets sent and received; the minimum, maximum, and average round-trip times; and a percentage indicating how many ping packets got no response. The following is a sample session that pings the machine at IP address `206.151.234.1`:

```
F:\Shared\abi-0.7.8>ping 206.151.234.1

Pinging 206.151.234.1 with 32 bytes of data:

Reply from 206.151.234.1: bytes=32 time=125ms TTL=250
Reply from 206.151.234.1: bytes=32 time=110ms TTL=250
Reply from 206.151.234.1: bytes=32 time=110ms TTL=250
Reply from 206.151.234.1: bytes=32 time=110ms TTL=250

Ping statistics for 206.151.234.1:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 110ms, Maximum = 125ms, Average = 113ms
```

What does this tell you? First of all, you can see that all of the packets you sent arrived, and there are approximately five hops in between this machine and your target. You know the latter because the time to live, or TTL, value is 250. By default, the TTL on the packets that `ping` sends out is set to 255, and each routing device that routes the packets subtracts one from the TTL value. When a packet's TTL hits zero, it is dropped.

More importantly, this `ping` session shows that data is flowing normally between your machine and the target. Since all of the ping packets got there (notice the "`0% loss`" line near the bottom), you can comfortably say that any network problems on this link aren't because of a routing problem. Packets are flowing normally between here and there.

How would you identify a problem using this data? The most obvious way to tell is when `ping` times out without getting *any* packets back from the remote end. That's a big red flag indicating either that you typed the IP address incorrectly or that something is blocking traffic between the two ends of the connection. Likewise, high rates of packet loss signal that something may be wrong somewhere along the path between the machines.

The *Tracert* Tool

When your plumbing is stopped up, you can tell because your sink, toilet, or shower won't drain—but knowing that it won't drain doesn't tell you where the blockage is. Likewise, the `ping` utility can tell you whether packets are flowing, but it won't necessarily tell you where the problem is. Windows 2000 includes a tool called `tracert` (pronounced "traceroute" after the original Unix version) that takes advantage of the TTL in each IP packet to map out the path that the packets are taking as they flow to a remote system. Recall that each device that routes a packet decrements its TTL. `Tracert` begins by sending one ICMP ping packet with a TTL of 1. That means that the first router or gateway to encounter it sends an ICMP response, decrements the ping packet's TTL, notices that the TTL is now zero, and drops the packet. At that point, `tracert` sends a second packet with a TTL of 2. The first device responds, decrements the TTL, and then routes the packet to the next hop. The next device in the chain responds to the `ping`, decrements the TTL, and drops the original packet. This process continues with `tracert` gradually incrementing the TTL until the packet finally reaches the desired destination host.

As it sends these packets, `tracert` keeps a running log of which hosts along the route have responded and which ones haven't. You can use this information to figure out where the stoppage is. For example, take a look at this tracert session:

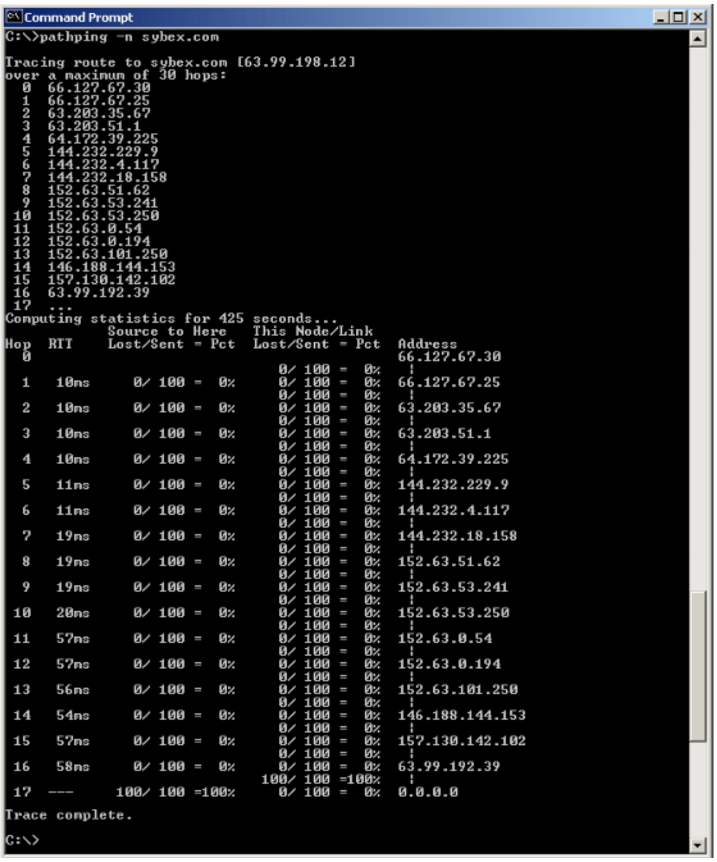
```
F:\>tracert www.microsoft.com

Tracing route to microsoft.com [207.46.131.137]
over a maximum of 30 hops:

1   <10 ms   <10 ms   <10 ms  ELGRANDE [192.168.0.1]
2     *        *        *     Request timed out.
3     *        *        *     Request timed out.
4     *        *        *     Request timed out.
5     *        *        *     Request timed out.
```

You can clearly see that the problem lies at the first hop away from your machine, a machine named ELGRANDE running the Routing and Remote Access Services (RRAS) package, or that the router at the second hop has gone down. You know this because the trace shows no response from any machine "downstream" of ELGRANDE. (Just for fun, while writing this section we did a second tracert on a Unix box outside our network at the same time and found that connectivity to Microsoft's website was blocked by a problem with an intermediate router in California!) In this case, it's easy to fix the problem on your end by restarting the RRAS service on ELGRANDE, but you wouldn't know that you needed to do that unless you did a tracert. However, if the router at hop 2 has failed, you might not be able to fix the problem.

The *Pathping* Tool

The pathping tool provides the functionality of both ping and tracert and adds some of its own features into the mix as well. A sample pathping output is shown in Figure 1.13. The first list in the output is the route that the packet takes to reach the destination. This is similar to the output of the tracert command. You will have to wait for several seconds (25 per hop, to be exact) until the next list appears. The two rightmost columns provide the most useful information. The Address column indicates the address of the node or link that the hop went to. The This Node/Link Lost/Sent% column indicates the packet loss that occurred at that point in the route. Typically the packet loss should be 0, but if you are having routing problems, you might spot a malfunctioning router by seeing where along the line you are losing packets.

FIGURE 1.13 Pathping output



```
C:\>pathping -n sybex.com

Tracing route to sybex.com [63.99.198.12]
over a maximum of 30 hops:
  0  66.127.67.30
  1  66.127.67.25
  2  63.203.35.67
  3  63.203.51.1
  4  64.172.39.225
  5  144.232.229.9
  6  144.232.4.117
  7  144.232.18.158
  8  152.63.51.62
  9  152.63.53.241
 10  152.63.53.250
 11  152.63.0.54
 12  152.63.0.194
 13  152.63.101.250
 14  146.188.144.153
 15  157.130.142.102
 16  63.99.192.39
 17  ...
Computing statistics for 425 seconds...
            Source to Here   This Node/Link
Hop  RTT    Lost/Sent = Pct  Lost/Sent = Pct  Address
  0                                           66.127.67.30
                                0/ 100 =  0%   |
  1   10ms     0/ 100 =  0%     0/ 100 =  0%  66.127.67.25
                                0/ 100 =  0%   |
  2   10ms     0/ 100 =  0%     0/ 100 =  0%  63.203.35.67
                                0/ 100 =  0%   |
  3   10ms     0/ 100 =  0%     0/ 100 =  0%  63.203.51.1
                                0/ 100 =  0%   |
  4   10ms     0/ 100 =  0%     0/ 100 =  0%  64.172.39.225
                                0/ 100 =  0%   |
  5   11ms     0/ 100 =  0%     0/ 100 =  0%  144.232.229.9
                                0/ 100 =  0%   |
  6   11ms     0/ 100 =  0%     0/ 100 =  0%  144.232.4.117
                                0/ 100 =  0%   |
  7   19ms     0/ 100 =  0%     0/ 100 =  0%  144.232.18.158
                                0/ 100 =  0%   |
  8   19ms     0/ 100 =  0%     0/ 100 =  0%  152.63.51.62
                                0/ 100 =  0%   |
  9   19ms     0/ 100 =  0%     0/ 100 =  0%  152.63.53.241
                                0/ 100 =  0%   |
 10   20ms     0/ 100 =  0%     0/ 100 =  0%  152.63.53.250
                                0/ 100 =  0%   |
 11   57ms     0/ 100 =  0%     0/ 100 =  0%  152.63.0.54
                                0/ 100 =  0%   |
 12   57ms     0/ 100 =  0%     0/ 100 =  0%  152.63.0.194
                                0/ 100 =  0%   |
 13   56ms     0/ 100 =  0%     0/ 100 =  0%  152.63.101.250
                                0/ 100 =  0%   |
 14   54ms     0/ 100 =  0%     0/ 100 =  0%  146.188.144.153
                                0/ 100 =  0%   |
 15   57ms     0/ 100 =  0%     0/ 100 =  0%  157.130.142.102
                                0/ 100 =  0%   |
 16   58ms     0/ 100 =  0%     0/ 100 =  0%  63.99.192.39
                              100/ 100 =100%   |
 17  ---     100/ 100 =100%     0/ 100 =  0%  0.0.0.0

Trace complete.

C:\>
```

The most useful switch to know is the `-n` switch, which only displays the IP address of each hop rather than resolving each name.

Using *Netdiag*

`Netdiag` is a tool that shows you just about everything about the state of the local computer's network configuration. The output that results from running the `netdiag` command can be quite daunting because of its sheer length, but if you break it down into manageable pieces, it can provide you with the information you need to fix connectivity problems. `Netdiag` requires no switches, but several optional switches are available to fine-tune the output. The most commonly used switch is the `/fix` switch, which will fix simple DNS client configuration errors. You must install the Windows 2000 Support Tools provided on the Windows 2000 CD to use `netdiag`. Table 1.6 explains the various tests that `netdiag` performs.

TABLE 1.6 Netdiag Tests

Test	Name	Explanation
NDIS	Network Adapter Status	Shows the details related to the network adapter such as the MAC address and the adapter name. If the adapter does not respond, the remaining tests will not run.
IPConfig	IP Configuration	This provides similar information to the `ipconfig /all` command. In addition, it pings the DHCP and WINS servers and verifies that the default gateway is on the same subnet as the local IP address.
Member	Domain Membership	Displays information regarding the local machine's domain membership.
NetBTTransports	Transports Test	Displays information about the NBT transport. If no NBT transport is found, it gives an error message.
Automatic Private IP Addressing (APIPA)	APIPA Address	Checks to see if any of the network adapters on the local machine are Automatic Private IP Addressing.

TABLE 1.6 Netdiag Tests *(continued)*

Test	Name	Explanation
IPLoopBk	IP Loopback Ping	Pings the IP loopback address of `127.0.0.1`.
DefGw	Default Gateway	Pings all the default gateways for each interface.
NbtNm	NetBT Name Test	This test is similar to the `nbtstat -n` command, discussed in Chapter 3.
WINS	WINS Service Test	Sends NetBT queries to all WINS servers.
Winsock	Winsock Test	Fetches available transport protocols for use with Windows Sockets.
DNS	DNS Test	Verifies the DNS configuration of the local computer. If the `/fix` option is used, the test tries to re-register with the DNS server.
Browser	Redirector and Browser Test	Checks the status of the Workstation service. Queries the redirector and the browser for transport lists. Checks the NBT transports test to see if the NBT transports are present. Verifies that the browser is bound to all of the NBT transports. Verifies that the local machine can send mailslot messages.

TABLE 1.6 Netdiag Tests *(continued)*

Test	Name	Explanation
DsGetDc	DC Discovery Test	Finds any random domain controller, and then finds the primary domain controller. The test then attempts to find a Windows 2000–based domain controller. Checks the local GUID against the domain controller's GUID and verifies their integrity. If the `/fix` switch is used, the test attempts to fix problems with the local GUID.
DcList	DC List Test	Displays a list of all the domain controllers in the domain.
Trust	Trust Relationship Test	Tests the state of the trust relationships if the computer is connected to a domain.
Kerberos	Kerberos Test	Tests Kerberos if the computer is connected to a domain and the user is not logged on locally.
LDAP	Lightweight Directory Access Protocol (LDAP) Test	Tests LDAP on all active domain controllers if Active Directory is present.
Route	Route test	Performs a test similar to the `route` command explained earlier in the chapter.
NetStat	NetStat test	Performs a test similar to the `netstat` tool explained in Chapter 3.

TABLE 1.6 Netdiag Tests *(continued)*

Test	Name	Explanation
Bindings	Bindings test	Lists details about network bindings.
WAN	WAN test	Tests the current remote access connection.
Modem	Modem test	Tests all of the computer's modem devices and displays configuration information.
NetWare	NetWare test	Tests the status of NetWare.
IPX	IPX test	Tests and displays the current IPX configuration.
IPSec	IP Security test	Tests IPSec and displays a list of the currently active IPSec policies.

Summary

In this chapter, you learned:

- Which network protocols are included with Windows 2000 and what they do
- How TCP/IP works, including IP addresses and subnet masks
- How to install network protocols, including NetBEUI and TCP/IP
- How to troubleshoot network problems

Exam Essentials

Know what protocols are and how they work. Protocols are an agreed-upon way in which two objects (people, computers, home appliances, or whatever) can exchange information. It is the protocols at a particular level in the OSI model that provide that level's functionality.

Know which major network protocols Windows 2000 supports. Microsoft networking products come with four network transports, which are intended for networks of different sizes with different requirements. In general, NetBEUI is intended for small, single-server networks. NWLink is intended for medium-sized networks (in a single facility, perhaps) or for networks that require access to Novell NetWare file servers. AppleTalk's primary use is interoperating with Macintosh computers. TCP/IP is a complex transport sufficient for globe-spanning networks such as the Internet.

Understand the difference between Class A, Class B, and Class C networks. In a Class A network, the first byte is the network address, and the three remaining bytes are used for the node addresses. In a Class B network, the first two bytes are assigned to the network address, and the remaining two bytes are used for node addresses. The first three bytes of a Class C network are dedicated to the network portion of the address, with only one byte remaining for the node address.

Understand what subnetting is and when to use it. If an organization is large and has many computers, or if its computers are geographically dispersed, it's sensible to divide its large network into smaller ones connected by routers. These smaller nets are called subnets. Subnetting is the process of carving a single IP network into smaller, logical subnetworks.

Understand subnet masks. For the subnet address scheme to work, every machine on the network must know which part of the host address will be used as the subnet address. The network administrator creates a 32-bit subnet mask consisting of ones and zeros. The ones in the subnet mask represent the positions that refer to the network or subnet addresses. The zeros represent the positions that refer to the host part of the address.

Understand how routing works. Each packet on a network has a source address and a destination address, which means that any device that receives the packet can inspect its headers to determine where it came from and where it's going. If such a device also has some information about the network's design and implementation—like how long it takes packets to travel over a particular link—it can intelligently change the routing to minimize the total cost, which refers to the time it takes a packet to travel from the source to the destination.

Know how to install network protocols. You install network protocols through the Local Area Network Connection Properties dialog box, which lists all of the known protocols on your Windows 2000 machine. Protocols marked with a check indicate that they're bound to the adapter whose properties you're inspecting.

Know how to configure TCP/IP settings. TCP/IP requires only two pieces of information to function: the IP address you want to use for the system and the subnet mask that corresponds to the network subnet the client is on. If you're configuring a Windows 2000 Professional machine, you're probably using DHCP with it. In that case, the default TCP/IP settings will work fine, since they configure the TCP/IP stack to get configuration parameters from any available DHCP server.

Know the steps for troubleshooting network protocols. First, figure out what the problem is. Arm yourself with as many details as possible about how the problem is manifesting itself (including exact error messages), when it started, and whether or not it's consistent. Knowing which users or computers are affected by a problem is very important, since that gives you insight into possible causes (including user mistakes) and helps you select a course of action. When you first notice a network problem, be sure to verify that all of your network cables are correctly connected; that your hub, router, or switch has power; and so on.

Know how to use the *ipconfig* tool. `Ipconfig` is used to view and, with switches such as `renew` and `release`, modify the configuration of TCP/IP interfaces on your local machine. Typing **`ipconfig`** *without the switches* into a Windows 2000 command prompt window produces a neat summary of your current IP configuration, including the local DNS name, the IP addresses, and the subnet masks configured for all adapters on the computer.

Know how to use the *arp* tool. The `arp` command allows you to view and modify the arp cache. The arp cache stores logical-to-physical-address mappings.

Know how to use the *route* command. The `route` command allows you to view and change the local static routing table. The two most commonly used `route` switches are `route add`, which creates a new static route, and `route print`, which displays the current routing table.

Know how to use the *ping* tool. When you ping a remote computer using the `ping` utility in its default mode, your computer sends out four ICMP ping packets and measures the time required before each packet's corresponding response arrives. When it finishes, `ping` gives you a helpful summary showing the number of packets sent and received; the minimum, maximum, and average round-trip times; and a percentage indicating how many ping packets got no response.

Know how to use the *tracert* tool. The `tracert` tool takes advantage of the TTL in each IP packet to map out the path that the packets are taking as they flow to a remote system.

Know how to use the *pathping* tool. The `pathping` tool provides the functionality of both `ping` and `tracert` and adds some of its own features. The tool displays the packet loss at each hop in a route.

Know how to use the *netdiag* command. `Netdiag` is a tool that shows you just about everything about the state of the local computer's network configuration. The most commonly used switch is the `/fix` switch, which fixes simple DNS client configuration errors.

Key Terms

Before you take the exam, be certain you are familiar with the following terms:

default gateway

default subnet mask

dynamic routing

hierarchical address

Internet Control Message Protocol (ICMP)

Internet Protocol (IP)

`ipconfig`

NBT

NetBEUI

network address

node address

NWLink

packet payload

packets

`ping`

static routing

subnet address

subnet mask

subnets

Transmission Control Protocol (TCP)

Review Questions

1. You are working at a manufacturing company that occupies an entire city block. Management informs you that they have acquired another business on the other side of town that previously had been a supplier to your company. The Windows 2000 network that you have been supporting now needs to be connected to the new location through a router. You also have several NetBIOS applications that need to continue functioning properly. What protocols are available for you to use to ensure that these criteria are met? (Choose all that apply.)

   A. NWLink

   B. TCP/IP

   C. XNS

   D. NetBEUI

2. The company you work for manufactures handballs and has an Intel PC–based Windows 2000 network. To cut packaging costs, the management of the company has acquired a graphics arts company. Their network is entirely Macintosh-based and is currently using AppleTalk as the protocol to communicate among workstations. You have to integrate the two networks so that they can easily share information. What protocols must you have on your network for communication among all the workstations on this network?

   A. AppleTalk

   B. TCP/IP

   C. NWLink

   D. NetBEUI

3. Your multinational company has a Windows NT and Novell NetWare network that is built on several subnetworks. To provide interoperability, you have been using NWLink on the NT network and IPX for the NetWare network. You have been told that the Windows NT network must be migrated to Windows 2000 because it's less expensive to administer. You know that the administrative cost benefits are a result of utilizing Active Directory, so you include this service in your migration plan. What are you going to have to do immediately in order to install and begin using Active Directory on this network?

   A. Change the protocol to TCP/IP.

   B. Make sure that you install a copy of Active Directory on the NetWare servers as well as on the Windows 2000 Servers.

   C. As you upgrade the Windows NT Servers, make sure that you choose to upgrade some of them as domain controllers so that you can install Active Directory on them.

   D. Install NetBEUI in order to provide connectivity for the NetBIOS components of Windows 2000.

4. Your company just purchased a new router. The router's address is `173.24.12.2/24` and your computer's address is `173.24.12.3/24`. You want to test the router's performance by routing all information to the `173.25.14.3/24` IP address through the new router. What command should you use before performing the test?

   A. `route delete 173.24.12.2`

   B. `route add 173.24.12.2 mask 255.255.255.0 173.25.14.3`

   C. `route print 173.25.14.3`

   D. `route add 173.25.14.3 mask 255.255.255.0 173.24.12.2`

5. Recently you have been experiencing performance problems with network traffic between your computer and several other computers on the network. Your company manages several routers between your host and the troublesome section of the network. You suspect that one of the routers along the line is dropping packets, but you want to be sure. What is the best command to use for obtaining detailed information about packet loss at each hop in a network transmission?

A. `ping`

B. `ipconfig`

C. `winipcfg`

D. `pathping`

E. `tracert`

F. `netdiag`

6. For several years you have been administering a small network that is fully contained in one building. You recently finished the migration from Windows NT to Windows 2000 and changed the protocol from NetBEUI to TCP/IP. You have just learned that a very large company that houses a network that contains several subnets connected by routers has acquired your company. Since you are not fully up to speed on all the details of TCP/IP, the acquiring company's IS department is going to send you a preconfigured router with the network address that you provided to them. After the line is installed, you receive the router and power it up. The network administrator of your new IS department checks out the router connections, and everything looks good. However, when you try to connect to resources on the other networks, the attempts fail. All the local workstations continue to function properly, but none of them can access anything across the router. When you ping the router interface, however, you get the proper response. What is most likely the problem?

A. You have provided an incorrect subnet mask.

B. You are running the wrong version of TCP/IP.

C. You have not provided a default gateway address.

D. You have not provided any DNS information.

E. You have provided an incorrect IP address.

7. You have spent a great deal of time upgrading your 500-node network from Windows NT to Windows 2000. During the migration you finally took advantage of the centralized management that DHCP brings to TCP/IP by redesigning your IP subnets and creating the scope necessary to cover all the workstations. You have activated the scope. You haven't implemented Active Directory yet, but you plan to do that after you confirm that everything works fine. During the weekend of the final rollover, at each workstation you run a script that edits the Registry to convert the IP configuration from static IP addressing to support DHCP; then you reboot all the machines. You test a few random machines and connect them to resources across your routers, and they all connect to the servers appropriately. On Monday morning you receive a flurry of phone calls from users who complain that the Internet connection is down. You check the Internet connection from your Windows 2000 Server, and the connection is fine. What is the probable cause of this problem?

A. The subnet mask is incorrect on some of the workstations.

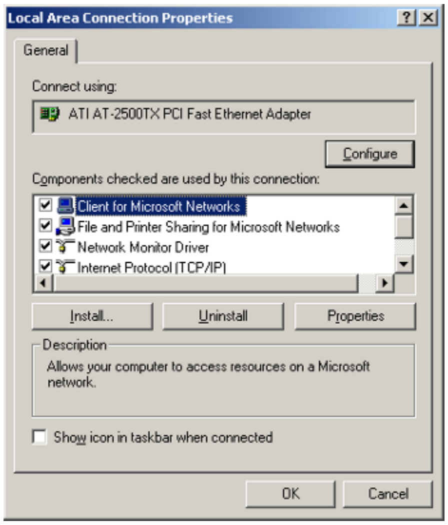
B. The default gateway is incorrect.

C. The DNS configuration on the workstations is overriding the configuration in the DHCP server.

D. The IP address scheme that you created is not valid.

E. The WINS server is not configured properly.

8. You administer a computer lab for a university. The lab consists entirely of Macintosh computers that use the AppleTalk protocol. You want to add a single Windows 2000 Server computer to the lab, and it needs to be able to talk to the other computers that are there. Where should you click in the following exhibit in order to accomplish this?



Connect using:

ATI AT-2500TX PCI Fast Ethernet Adapter

Components checked are used by this connection:

☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☑ Network Monitor Driver
☑ Internet Protocol (TCP/IP)

Description

Allows your computer to access resources on a Microsoft network.

☐ Show icon in taskbar when connected

A. Install

B. Uninstall

C. Properties

D. Configure

9. You are helping the lead systems engineer design a network for a new company location. You determine that in the future the company could conceivably grow to include as many as 200 different networks. Each network could also contain as many as 1,000 clients. Your boss insists that he doesn't want to subnet the network. Which network address class should you use?

A. Class A

B. Class B

C. Class C

D. Class D

10. You work for Carpathian Worldwide Enterprises, which has more than 50 administrative and manufacturing locations around the world. The size of these organizations varies greatly, with the number of computers per location ranging from 15 to slightly fewer than 1,000. The sales operations use more than 1,000 facilities, each of which contains two to five computers. Carpathian is also in merger talks with another large organization; if the merger materializes as planned, you will have to accommodate another 100 manufacturing and administrative locations, each with a maximum of 600 computers, as well as 2,000 additional sales facilities. You don't have any numbers for the future growth of the company, but you are told to keep growth in mind. You decide to implement a private addressing plan for the entire organization. More than half of your routers don't support variable subnet masking. What subnet masks would work for this situation? (Choose all that apply.)

A. `255.255.224.0`

B. `255.255.240.0`

C. `255.255.248.0`

D. `255.255.252.0`

E. `255.255.254.0`

11. For several years you have been administering a small network that is fully contained in one building. You recently finished the migration from Windows NT to Windows 2000 and changed the protocol from NetBEUI to TCP/IP. You have just learned that a very large company that houses a network that contains several subnets connected by routers has acquired your company. Since you are not fully up to speed on all the details of TCP/IP, the corporate administrators are going to send you a pre-configured router with the network address that you provided to them. After the line is installed, you receive the router and power it up. The network administrator of your new IS department checks out the router connections, and everything looks good. However, when you try to connect to resources on the other networks, the attempts fail. All the local workstations continue to function properly, but none of them can access anything across the router. What tool would you use to troubleshoot this problem?

    A. `netstat -a`

    B. `nslookup`

    C. `ipconfig`

    D. `winipcfg`

    E. `nbtstat`

12. You want to install two network cards on one computer. You definitely want to configure the IP address of one of the cards manually. How must you configure the other card?

    A. Both cards must be configured manually.

    B. Both cards must be configured dynamically.

    C. One card must be configured manually, and the other must be configured dynamically.

    D. Both cards can be configured either manually or dynamically.

13. The company you work for is growing dramatically via acquisitions of other companies. As the network administrator, you need to keep up with the changes because they affect the workstations and you need to support them. When you started, there were 15 locations connected via routers, and now there are 25. As new companies are acquired, they are migrated to Windows 2000 and brought into the same domain as another site. Management says that they are going to acquire at least 10 more companies in the next two years. The engineers have also told you that they are redesigning the company's Class B address into an IP addressing scheme that will support these requirements and that there will never be over 1000 network devices on any subnet. What will be the appropriate subnet mask to support this network when the changes are completed?

A. `255.255.252.0   63N   1023`

B. `255.255.248.0   31N   2047`

C. `255.255.255.0   254   255`

D. `255.255.255.128   511N   127w`

14. The company that you work for has two separate divisions: one that handles sporting event ticketing and the other that handles leasing event venues. They are completely separate from a financial operations perspective, but they are located in the same building, connected by a single router. You are the administrator for both divisions. Even though the companies are managed separately, they share some of their IS resources, such as their Internet connection and an IIS server that is physically on the ticketing side of the company. One of the workstations in the venue side of the house cannot connect to any of the resources on the other side, including the Internet. The following machines are configured as follows:

IIS server:	
Node address	`192.23.64.23/24`
Gateway	`192.23.64.1/24`
Router:	
Ticketing interface	`192.23.64.1/24`
ISP interface	`10.2.223.23/28`
Venue interface	`204.45.36.1/24`
Problem workstation:	
Node address	`204.45.36.2/24`
Gateway	`10.2.223.23/28`

What do you need to do to allow the workstation to access the Internet?

A. Change the IIS server gateway to `10.2.223.23/28`.

B. Change the ISP interface to `192.23.64.1/24`.

C. Change the workstation gateway to `204.45.36.1/24`.

D. Change the workstation gateway to `192.23.64.1/24`.

15. You are the administrator of a network that has completed its migration from Windows NT. The entire company is located in one building, and the network has been a flat subnet until recently. The company has experienced accelerated growth, and there are now more than 1,000 users on the network. The network engineers have decided to break the network into two segments separated by a multi-homed Windows 2000 Server. One NIC has an address of `172.160.0.1`, and the other NIC has an address of `172.150.0.1`. The workstations have been largely divided between the two segments and have been configured by the engineer. You immediately start getting calls from users on both sides, stating that they cannot reach the other side of the router. You run the `route print` command, and it displays the following information:

```
Destination    Netmask            Gateway        Interface
172.160.0.0    255.255.0.0        172.160.0.1    172.160.0.1
172.160.0.1    255.255.255.255    127.0.0.1      127.0.0.1
172.150.0.0    255.255.0.0        172.150.0.11   172.150.0.1
172.150.0.1    255.255.255.255    127.0.0.1      127.0.0.1
```

What `route` command do you need to execute on the router to resolve the address resolution problem?

A.
```
route delete 172.160.0.0
route –p add 172.160.0.0 mask 255.255.0.0  172.160.0.1
```

B.
```
route delete 172.160.0.1
route –p add 172.160.0.1 mask 255.255.255.255
172.160.0.1
```

C.
```
route delete 172.150.0.
route –p add 172.150.0.0 mask 255.255.0.0  172.150.0.1
```

D.
```
route delete 172.150.0.1.
route –p add 172.150.0.1 mask 255.255.255.255
172.150.0.1
```

Answers to Review Questions

1. A, B. Both NWLink and TCP/IP are routable and both can function properly with NetBIOS applications, since they are both Microsoft's versions and have the interface for proper communication. XNS is a routable protocol but is not provided with Windows 2000. With the overwhelming popularity of TCP/IP, XNS is generally no longer used in networks. NetBEUI, although it supports the NetBIOS programs, is not routable.

2. B. Although Macintosh computers can use AppleTalk to communicate with each other, these computers can also run TCP/IP, so AppleTalk won't be necessary when these two networks are merged. You could add AppleTalk to the servers in the network, and the two machine types could share files back and forth, but if you can reduce the number of protocols on any network, it's best practice to do so.

3. A. Active Directory requires TCP/IP in order to function. Even though you can have TCP/IP and IPX coexisting on the same network, it's not beneficial to have multiple protocols, as they increase the level of support necessary for the network. Active Directory does not run on NetWare, and NetBEUI is not required for NetBIOS communication. Finally, Active Directory can be installed and uninstalled on any Windows 2000 Server computer. It's a service that is added rather than a particular type of server that is installed, as with Windows NT.

4. D. The `route add` command is used to add a static route to the local routing table. The correct syntax for the route add command is `route add <destination> mask <netMask> <gateway> <metric> <interface>`.

5. D. `Pathping` is similar to `tracert` but gives more detailed information about each hop. For example, you can see exactly how many packets each router drops.

6. C. You need to provide a default gateway address so that the computer can route packets to other subnets and networks. When a packet is formed and is addressed to a different subnet, the local IP stack looks to a special address in its configuration to forward the packet to. This is called the default gateway. If the default gateway isn't configured, all local IP traffic will function properly, but the machines missing this gateway configuration won't be able to reach any other networks.

   If the subnet mask or IP addresses were incorrect, local communication would not work properly. The DNS configuration is used for name resolution and would not result in this type of failure.

7. C. A static DNS configuration can override the DHCP configuration that negotiated with the DHCP client. Since the Registry was edited from static IP to DHCP, the DNS information wasn't changed and is still entered as static information, overriding the DHCP configuration. With the incorrect DNS configuration, the workstations cannot resolve a URL into the IP addresses necessary to connect to resources on the Web. If the subnet mask and IP addresses were incorrect, they would not be able to communicate on the local network. The default gateway is accurate because you were able to make connections across the routers. WINS is not involved with web services browsing.

8. A. The new Windows computer needs to have AppleTalk installed in order to communicate with the Macintosh machines in the lab. To install any services or protocols, you need to click the Install button in the Local Area Connection Properties dialog box.

9. B. Class A networks can have a maximum of 127 networks. Class C networks can have a maximum of 254 nodes per network. Class D networks are used for multicasting only.

10. B, C, D. When you add up the locations that currently need to be given a network address, the total is 3,150, and the maximum number of hosts at any one of these locations is less than 1,000. The subnet masks need to support those requirements. Each of the subnet masks given in the second, third, and fourth answers will provide the address space to support the requirements outlined above. The subnet mask `255.255.240.0` supports more than 4,000 subnets and 4,000 hosts. The subnet mask `255.255.248.0` supports more than 8,000 subnets and more than 2,000 hosts. The subnet mask `255.255.252.0` supports more than 16,000 subnets and more than 1,000 hosts.

Although each of these subnet masks will work, at the rate that this company is growing, `255.255.252.0` is probably the best mask to prepare for the future. It's unlikely that there will ever be more than 1,000 hosts on any given network. In fact, that number would probably cause performance problems on that subnet. Therefore, it's better to have more subnets available to deploy as the company grows.

The subnet mask `255.255.224.0` supports more than 2,000 subnets—an insufficient number to cover the locations. The subnet mask `255.255.254.0` supports more than 32,000 subnets but only 500 hosts, which is not enough hosts to cover all the locations.

11. C. `Ipconfig` displays the configuration information of the IP stack of the machine you run it on. To view the details of the IP configuration, including the DNS information, you need to add the `/all` switch to the command.

`Netstat` shows active TCP connections, and `nbtstat` shows similar information for NetBIOS connections. `Winipcfg` is a similar command for TCP/IP configurations, but it runs on Windows 9.*x*. `Nslookup` is used to discover DNS name resolution problems.

12. D. Any adapter can use either DHCP or manual addressing without reference to what the other adapters are using.

13. A. The network mask applied to an address determines which portion of that address reflects the number of hosts available to that network. The balance with subnetting is always between the number of hosts and individual subnetworks that can be uniquely represented within one encompassing address. The number of hosts and networks that are made available depends upon the number of bits that can be used to represent them. This scenario requires more than 35 networks and fewer than 1000 workstations on each network. If you convert the subnet masks as described in the chapter, you will see that the mask in option A allows for more than 60 networks and more than 1000 hosts. All of the other options are deficient in either the number of networks or hosts that they represent.

14. C. The workstation gateway address is the address that IP uses to send packets that are off the network. Regardless of whether the ultimate destination is to the IIS server on the other LAN or an address somewhere on the Internet, the only way packets can reach this destination is if they first reach the gateway that can contact other routers to forward the packets. You do not make the gateway address the address of a particular machine that you want to reach.

15. C. The host ID or network ID is displayed in the Destination field. The Netmask column places the correct mask that is used to define the relationship between the network ID and the range of host IDs. The important column here is Gateway, which shows where packets will be sent if their destination is off the network. The Interface column shows the IP address that is configured for this device, which is the router. The command in the C option corrects a problem where the router is forwarding packets to the wrong address. It needs to send the packets to the address that represents the interface of the network that it is trying to reach.