CHAPTER 23

Local Area Networking

In this chapter, you will learn how to

- Explain networking technologies
- Explain network operating systems
- Install and configure wired networks
- Troubleshoot networks

Networks dominate the modern computing environment. A vast percentage of businesses have PCs connected in a small local area network (LAN), and big businesses simply can't survive without connecting their many offices into a single wide area network (WAN). Even the operating systems of today demand networks. Windows XP, Vista, and 7, for example, come out of the box *assuming* you'll attach them to a network of some sort just to make them work past 30 days (product activation), and they get all indignant if you don't.

Because networks are so common today, every good tech needs to know the basics of networking technology, operating systems, implementation, and troubleshooting. Accordingly, this chapter teaches you how to build and troubleshoot a basic network.

Historical Conceptual

Networking Technologies

When the first network designers sat down at a café to figure out how to get two or more PCs to share data and peripherals, they had to write a lot of details on little white napkins to answer even the most basic questions. The first big question was: *How*? It's easy to say, "Well, just run a wire between them!" Although most networks do manifest themselves via some type of cable, this barely touches the thousands of questions that come into play here. Here are a few of the *big* questions:

- How will each computer be identified? If two or more computers want to talk at the same time, how do you ensure that all conversations are understood?
- What kind of wire? What gauge? How many wires in the cable? Which wires do which things? How long can the cable be? What type of connectors?

- If more than one PC accesses the same file, how can they be prevented from destroying each other's changes to that file?
- How can access to data and peripherals be controlled?

Clearly, making a modern PC network entails a lot more than just stringing up some cable! Most commonly, you have a *client* machine, a PC that requests information or services. It needs a *network interface card* (*NIC*) that defines or labels the client on the network. A NIC also helps break files into smaller data units, called *packets*, to send across the network, and it helps reassemble the packets it receives into whole files. Second, you need some medium for delivering the packets between two or more PCs—most often this is a wire that can carry electrical pulses; sometimes it's radio waves or other wireless methods. Third, your PC's operating system has to be able to communicate with its own networking hardware and with other machines on the network. Finally, modern PC networks often employ a *server* machine that provides information or services. Figure 23-1 shows a typical network layout.

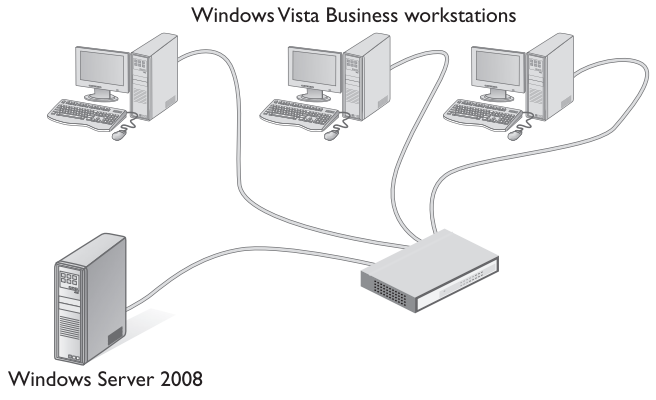


Figure 23-1
A typical network

This section of the chapter looks at the inventive ways network engineers found to handle the first two of the four issues. After a brief look at core technology, the chapter dives into four specific types of networks. You'll dig into the software side of things later in the chapter.

Topology

If a bunch of computers connect together to make a network, some logic or order must influence the way they connect. Perhaps each computer connects to a single main line that snakes around the office. Each computer might have its own cable, with all of the cables coming together to a central point. Or maybe all of the cables from all of the computers connect to a main loop that moves data along a track, picking up and dropping off data like a circular subway line.

A network's *topology* describes the way that computers connect to each other in that network. The most common network topologies are called *bus*, *ring*, *star*, and *mesh*.

Figure 23-2 shows the four types: a *bus topology*, where all computers connect to the network via a main line called a *bus cable*; a *ring topology*, where all computers on the network attach to a central ring of cable; a *star topology*, where the computers on the network connect to a central wiring point (usually called a *hub*); and a *mesh topology*, where each computer has a dedicated line to every other computer—the mesh topology is mostly used in wireless networks. There are also *hybrid topologies*, such as star bus or star ring, that combine aspects of the other topologies to capitalize on their strengths and minimize their weaknesses. You'll look at the most important hybrid topology, star bus, in a moment, but for now, make sure you know the four main topologies!

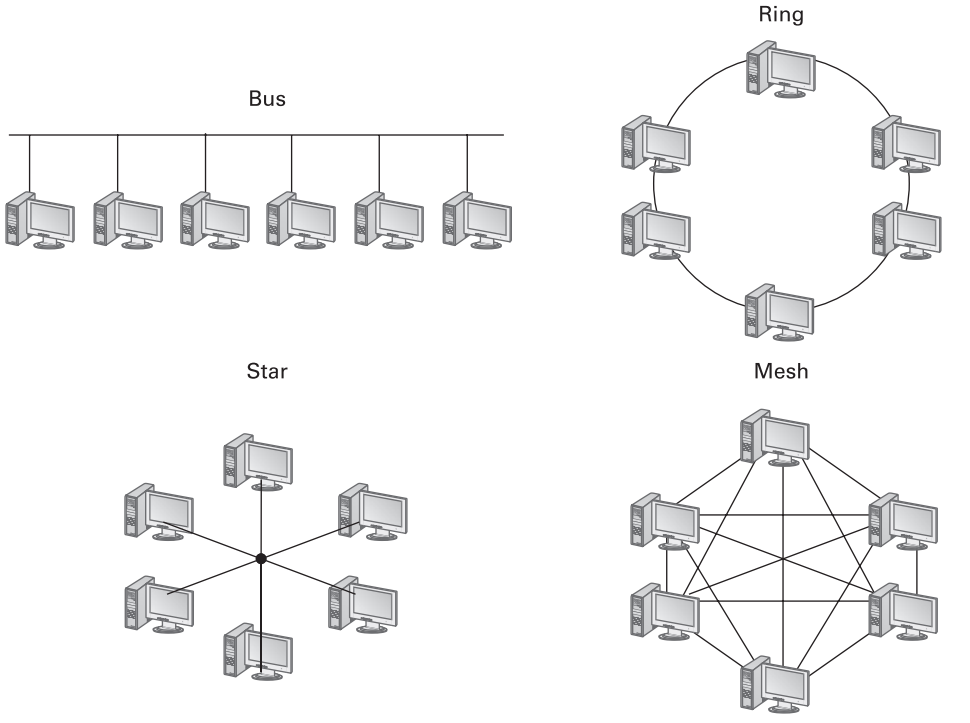


Figure 23-2 Clockwise from top left: bus, ring, mesh, and star topologies

If you're looking at Figure 23-2 and thinking that a mesh topology looks amazingly resilient and robust, it is—at least on paper. Because every computer physically connects to every other computer on the network, even if half of the PCs crash, the network functions as well as ever (for the survivors). In a practical sense, however, implementing a true mesh topology network would be an expensive mess. For example, even for a tiny network with only 10 PCs, you would need *45* separate and distinct pieces of cable to connect every PC to every other PC. What a mesh mess! Because of this, mesh topologies have never been practical in a cabled network.

Although a topology describes the method by which systems in a network connect, the topology alone doesn't describe all of the features necessary to make a cabling system work. The term *bus topology*, for example, describes a network that consists of some number of machines connected to the network via the same piece of cable. Notice that this definition leaves a lot of questions unanswered. What is the cable made of? How long can it be? How do the machines decide which machine should send data at a specific moment? A network based on a bus topology can answer these questions in a number of different ways.

Most techs make a clear distinction between the *logical topology* of a network—how the network is laid out on paper, with nice straight lines and boxes—and the physical topology. The *physical topology* describes the typically messy computer network, with cables running diagonally through the ceiling space or snaking their way through walls. If someone describes the topology of a particular network, make sure you understand whether they're talking about the logical or physical topology.

Over the years, manufacturers and standards bodies created several specific network technologies based on different topologies. A *network technology* is a practical application of a topology and other critical technologies to provide a method to get data from one computer to another on a network.

Essentials

Packets/Frames and NICs

Data is moved from one PC to another in discrete chunks called *packets* or *frames*. The terms *packet* and *frame* are interchangeable. Every NIC in the world has a built-in identifier, a binary address unique to that single network card, called a *media access control* (*MAC*) *address*. You read that right—every network card in the world has its own unique MAC address! The MAC address is 48 bits long, providing more than 281 *trillion* MAC addresses, so there are plenty of MAC addresses to go around. MAC addresses may be binary, but we represent them by using 12 hexadecimal characters. These MAC addresses are burned into every NIC, and some NIC makers print the MAC address on the card. Figure 23-3 shows the System Information utility description of a NIC, with the MAC address highlighted.

NOTE Even though MAC addresses are embedded into the NIC, some NICs allow you to change the MAC address on the NIC. This is rarely done.

Hey! I thought we were talking about packets? Well, we are, but you need to understand MAC addresses to understand packets. The many varieties of packets share certain common features (Figure 23-4). First, packets contain the MAC address of the network card to which the data is being sent. Second, they have the MAC address of the network card that sent the data. Third is the data itself (at this point, we have no idea what the data is—certain software handles that question), which can vary in size depending on the type of frame. Finally, some type of data check—such as a *cyclic redundancy check*

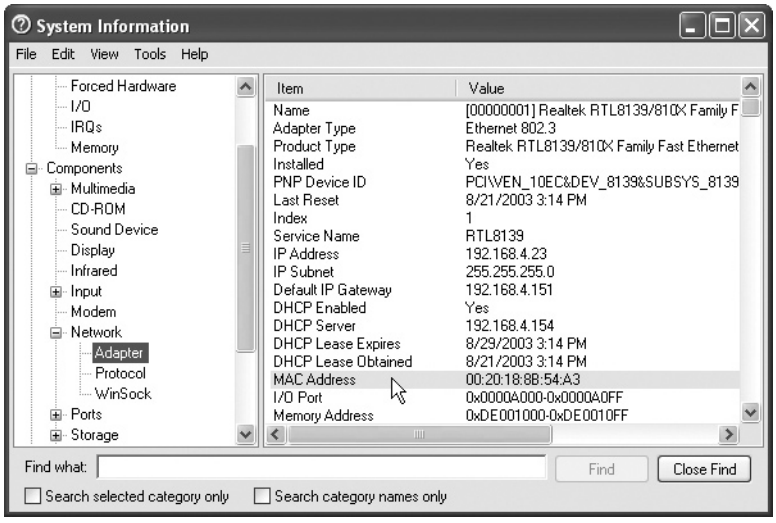


Figure 23-3
MAC address

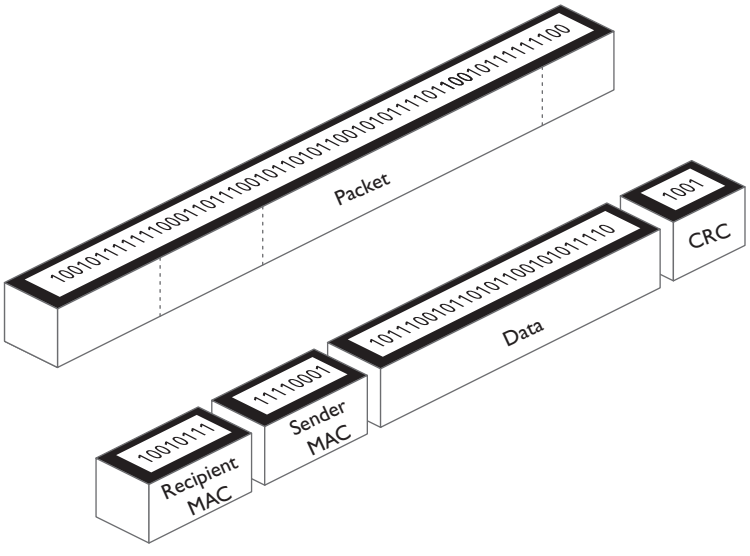


Figure 23-4
Generic packet/ frame

(*CRC*)—is performed and information is stored in the packet to enable the receiving network card to verify if the data was received in good order.

This discussion of packets raises the question, how big is the packet? Or more specifically, how much data do you put into each packet? How do you ensure that the receiving PC understands the *way* the data was broken down by the sending machine and can thus put the pieces back together? The problem in answering these questions is that they encompass so many items. When the first networks were created, *everything* from the frames to the connectors to the type of cable had to be invented from scratch.

To make a successful network, you need the sending and receiving PCs to use the same hardware protocol. A *hardware protocol* defines many aspects of a network, from the topology, to the packet type, to the cabling and connectors used. A hardware protocol defines everything necessary to get data from one computer to another. Over the years, many hardware protocols have been implemented, with such names as Token Ring, FDDI, and ARCnet, but one hardware protocol dominates the modern PC computing landscape: Ethernet.

Introducing Ethernet

A consortium of companies centered on Digital Equipment, Intel, and Xerox invented the first network in the mid 1970s. More than just creating a network, they wrote a series of standards that defined everything necessary to get data from one computer to another. This series of standards was called *Ethernet*, and it is the dominant standard for today's networks. Ethernet comes in two main flavors defined by cabling type: unshielded twisted pair and fiber optic. Because all flavors of Ethernet use the same packet type, you can have any combination of hardware devices and cabling systems on an Ethernet network and all of the PCs will be able to communicate just fine.

Most modern Ethernet networks employ one of three technologies (and sometimes all three), *10BaseT*, *100BaseT*, or *1000BaseT*. As the numbers in the names suggest, 10BaseT networks run at 10 Mbps, 100BaseT networks run at 100 Mbps, and 1000BaseT networks—called Gigabit Ethernet—run at 1000 Mbps, or 1 Gbps. All three technologies—sometimes referred to collectively as *10/100/1000BaseT* or just plain Ethernet—use a *star bus* topology and connect via a type of cable called *unshielded twisted pair* (*UTP*).

NOTE You'll sometimes hear or read 10/100/1000BaseT referred to as 10xBaseT.

Star Bus

Imagine taking a bus network and shrinking the bus down so it will fit inside a box. Then, instead of attaching each PC directly to the wire, you attach them via cables to special ports on the box (Figure 23-5). The box with the bus takes care of all of the tedious details required by a bus network. The bus topology would look a lot like a star topology, wouldn't it?

The central box with the bus is called a hub or switch. The *hub* provides a common point for connection for network devices. Hubs can have a wide variety of ports. Most consumer-level hubs have four or eight, but business-level hubs can have 32 or more ports. A hub is the old-style device, rarely used in today's networks. A *switch* is a far superior and far more common version of a hub. Figure 23-6 shows a typical consumer-level switch.

Different networks come in different speeds. One common speed is 100 Megabits per second (Mbps). We call this speed a *bandwidth*. If you put 32 PCs on a 32-port 100 Mbps hub, you have 32 PCs sharing the 100 Mbps bandwidth. A switch addresses

Figure 23-5
Star bus

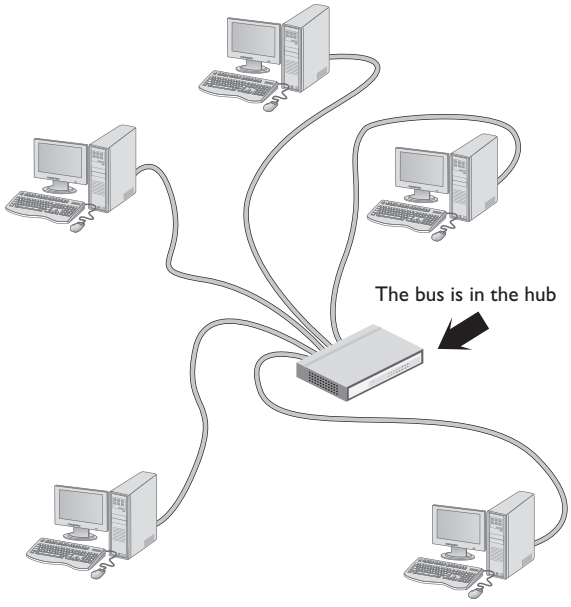


Figure 23-6
A switch

that problem by making each port its own separate network. Each PC gets to use the full bandwidth available. The bottom line? Swap out your old hubs for newer switches and you'll dramatically improve your network performance.

Cheap and centralized, a star bus network does not go down if a cable breaks. True, the network would go down if the hub itself failed, but that is rare. Even if a hub fails, replacing a hub in a closet is much easier than tracing a bus running through walls and ceilings and trying to find a break!

EXAM TIP Although Token Ring is very rarely seen today, CompTIA wants you to know a little of its terminology. Just as in Ethernet, the creators of Token Ring decided to move to a star topology and put the ring (as opposed to Ethernet's bus) in a box that looks like a hub/switch. They call the box the *media access* (or sometimes *attachment*) *unit* (*MAU*). Some techs call it a Multistation Access Unit (MSAU), but CompTIA uses MAU, so remember that for the exam!

Unshielded Twisted Pair

Unshielded twisted pair (UTP) cabling is the specified cabling for 10/100/1000BaseT and is the predominant cabling system used today. Many types of twisted pair cabling are available, and the type used depends on the needs of the network. Twisted pair cabling consists of AWG 22–26 gauge wire twisted together into color-coded pairs. Each wire is individually insulated and encased as a group in a common jacket.

CAT Levels UTP cables come in categories that define the maximum speed at which data can be transferred (also called *bandwidth*). The major categories (CATs) are as follows:

CAT 1	Standard phone line	**CAT 2**	Data speeds up to 4 Mbps (ISDN and T1 lines)
CAT 3	Data speeds up to 16 Mbps	**CAT 4**	Data speeds up to 20 Mbps
CAT 5	Data speeds up to 100 Mbps	**CAT 5e**	Data speeds up to 1 Gbps
CAT 6	Data speeds up to 10 Gbps		

The CAT level should be clearly marked on the cable, as Figure 23-7 shows.

Figure 23-7 Cable markings for CAT level

The *Telecommunication Industry Association/Electronics Industries Alliance* (*TIA/EIA*) establishes the UTP categories, which fall under the TIA/EIA 568 specification. Currently, most installers use CAT 5e or CAT 6 cable. Although many networks run at 10 Mbps, the industry standard has shifted to networks designed to run at 100 Mbps and faster. Because only CAT 5 or better handles these speeds, just about everyone is installing the higher rated cabling, even if they are running at speeds that CAT 3 or CAT 4 would do.

Consequently, it is becoming more difficult to get anything but CAT 5, CAT 5e, or CAT 6 cables.

Shielded Twisted Pair

Shielded twisted pair (*STP*), as its name implies, consists of twisted pairs of wires surrounded by shielding to protect them from EMI, or electromagnetic interference. STP is pretty rare, primarily because there's so little need for STP's shielding; it only really matters in locations with excessive electronic noise, such as a shop floor area with lots of lights, electric motors, or other machinery that could cause problems for other cables.

Implementing 10/100/1000BaseT

The 10BaseT, 100BaseT, and 1000BaseT cabling standards require two pairs of wires: a pair for sending and a pair for receiving. 10BaseT runs on CAT 3, CAT 4, or CAT 5 cable. 100BaseT requires at least CAT 5 to run. 1000BaseT is a special case because it needs all four pairs of wires in a CAT 5e or CAT 6 cable. These cables use a connector called an *RJ-45* connector. The *RJ* (*registered jack*) designation was invented by Ma Bell (the phone company, for you youngsters) years ago and is still used today. Currently only two types of RJ connectors are used for networking: RJ-11 and RJ-45 (Figure 23-8). *RJ-11* is the connector that hooks your telephone to the telephone jack. It supports up to two pairs of wires, though most phone lines use only one pair. The other pair is used to support a second phone line. RJ-11 connectors are primarily used for dial-up networking (see Chapter 25, "The Internet") and are not used in any common LAN installation, although a few weird (and out of business) "network in a box" companies used them. RJ-45 is the standard for UTP connectors. RJ-45 has connections for up to four pairs and is visibly much wider than RJ-11. Figure 23-9 shows the position of the #1 and #8 pins on an RJ-45 jack.

Figure 23-8
RJ-11 and RJ-45

The TIA/EIA has two standards for connecting the RJ-45 connector to the UTP cable: the TIA/EIA 568A and the TIA/EIA 568B. Both are acceptable. You do not have to follow any standard as long as you use the same pairings on each end of the cable; however, you will make your life simpler if you choose a standard. Make sure that all of your cabling uses the same standard and you will save a great deal of work in the end. Most importantly, *keep records*!

Like all wires, the wires in UTP are numbered. However, a number does not appear on each wire. Instead, each wire has a standardized color. Table 23-1 shows the official TIA/EIA Standard Color Chart for UTP.

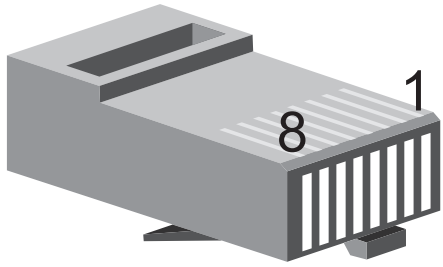


Figure 23-9 RJ-45 pin numbers

Pin	568A	568B	Pin	568A	568B
1	White/Green	White/Orange	5	White/Blue	White/Blue
2	Green	Orange	6	Orange	Green
3	White/Orange	White/Green	7	White/Brown	White/Brown
4	Blue	Blue	8	Brown	Brown

Table 23-1 UTP Cabling Color Chart

Plenum versus PVC Cabling Most workplace installations of network cable go up above the ceiling and then drop down through the walls to present a nice port in the wall. The space in the ceiling, under the floors, and in the walls through which cable runs is called the *plenum* space. The potential problem with this cabling running through the plenum space is that the protective sheathing for networking cables, called the *jacket*, is made from plastic, and if you get any plastic hot enough, it creates smoke and noxious fumes. Standard network cables usually use PVC (polyvinyl chloride) for the jacket, but PVC produces noxious fumes when burned. Fumes from cables burning in the plenum space can quickly spread throughout the building, so you want to use a more fire-retardant cable in the plenum space. Plenum-grade cable is simply network cabling with a fire-retardant jacket and is required for cables that go in the plenum space. Plenum-grade cable costs about three to five times more than PVC, but you should use it whenever you install cable in a plenum space.

Figure 23-10 NIC built into motherboard

Multispeed Cards All Ethernet networks share the same language, so you can easily have mixed or combined networks. All it takes is a network card capable of running at multiple speeds or even over multiple cables. Most NICs built into motherboards today, for example, are Gigabit auto-sensing cards (Figure 23-10). If you plug into a 100BaseT network, they automatically run at 100 Mbps. If you plug into a 1000 Mbps network, they quickly ramp up and run at 1000 Mbps.

Crossover Cables You can actually hook two network cards together without a hub by using a special UTP cable called a *crossover cable*. A crossover cable is a standard UTP cable but with one RJ-45 connector using the 568A standard and the other using the 568B. This reverses the signal between sending and receiving wires and thus does the job of a hub or switch. Crossover cables work great as a quick way to network two PCs. You can purchase a crossover cable at any computer store.

Duplex and Half-Duplex All modern NICs can run in *full-duplex* mode, meaning they can send and receive data at the same time. The vast majority of NICs and switches use a feature called *auto-sensing* to accommodate very old devices that might attach to the network and need to run in half-duplex mode. Half-duplex means that the device can send and receive, but not at the same time. An obvious example of a half-duplex device is the walkie-talkies you played with as a kid that required you to press and hold the orange button to transmit—at which time you couldn't hear anything. Half-duplex devices are exceedingly rare in modern computers, but you need to understand this option. Some NICs just can't handle full-duplex communication when you plug them directly to another NIC by using a crossover cable—that is, no switch. Dropping both NICs down from full-duplex or auto-sensing can sometimes enable these odd NICs to communicate.

Link Lights All NICs made today have some type of light-emitting diode (LED) *status indicator* that gives information about the state of the NIC's link to whatever's on the other end of the connection. Even though you know the lights are actually LEDs, get used to calling them *link lights*, as that's the term all network techs use. NICs can have between one and four different link lights, and the LEDs can be any color. These lights give you clues about what's happening with the link and are one of the first items to check whenever you think a system is disconnected from the network (Figure 23-11).

Figure 23-11 Mmmm, pretty lights!

Hubs and switches also have link lights, enabling you to check the connectivity at both ends of the cable. If a PC can't access a network, always first check the link lights. Multispeed devices usually have a link light that tells you the speed of the connection. In

Figure 23-12, the light for port 2 on the top photo is orange, for example, signifying that the other end of the cable is plugged into either a 10BaseT or 100BaseT NIC. The same port connected to a Gigabit NIC—that's the lower picture—displays a green LED.

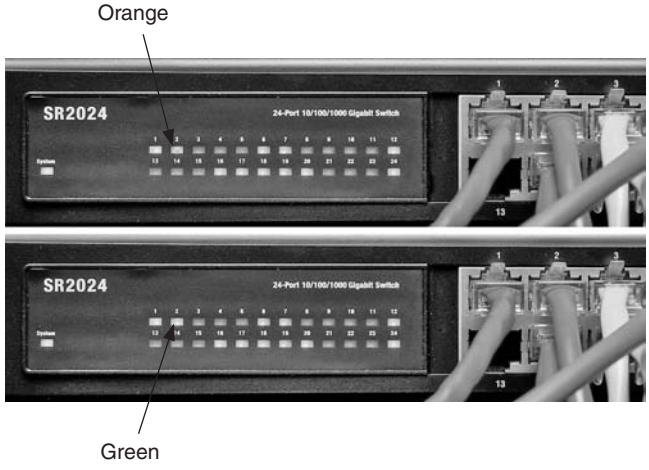


Figure 23-12 Multispeed lights

A properly functioning link light is steady on when the NIC is connected to another device. No flickering, no on and off, just on. A link light that is off or flickering shows a connection problem.

Another light is the *activity light*. This little guy turns on when the card detects network traffic, so it makes an intermittent flickering when operating properly. The activity light is a lifesaver for detecting problems, because in the real world, the connection light sometimes lies to you. If the connection light says the connection is good, the next step is to try to copy a file or do something else to create network traffic. If the activity light does not flicker, you have a problem.

No standard governs how NIC manufacturers use their lights; as a result, they come in an amazing array of colors and layouts. When you encounter a NIC with a number of LEDs, take a moment to try to figure out what each one means. Although different NICs have different ways of arranging and using their LEDs, the functions are always the same: link, activity, and speed.

EXAM TIP Though no real standard exists for NIC LEDs, CompTIA will test you on some more-or-less de facto LED meanings. You should know that a solid green light means connectivity, a flashing green light means intermittent connectivity, no green light means no connectivity, and a flashing amber light means there are collisions on the network (which is sometimes okay). Also, know that the first things you should check when having connectivity issues are your NIC's LEDs.

Fiber Optic Ethernet

Fiber optic cable is a very attractive way to transmit Ethernet network packets. First, because it uses light instead of electricity, fiber optic cable is immune to electrical problems such as lightning, short circuits, and static. Second, fiber optic signals travel much

farther, up to 2000 meters (compared with 100 meters on UTP) with some standards. Most fiber Ethernet networks use *62.5/125 multimode* fiber optic cable. All fiber Ethernet networks that use these cables require two cables. Figure 23-13 shows three of the more common connectors used in fiber optic networks. Square *SC* connectors are shown in the middle and on the right, and the round *ST* connector is on the left.

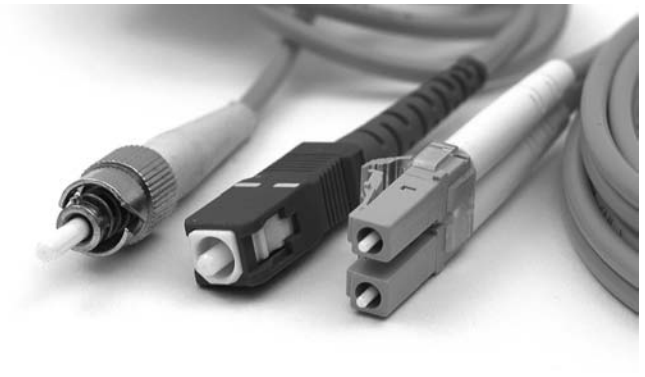


Figure 23-13 Typical fiber optic cables with connectors

Like many other fiber optic connectors, the SC and ST connectors are half-duplex, meaning data flows only one way—hence the need for two cables in a fiber installation. Other half-duplex connectors you might run into are FC/PC, SMA, D4, MU, and LC. They look similar to SC and ST connectors but offer variations in size and connection. Newer and higher-end fiber installations use full-duplex connectors, such as the MT-RJ connectors.

EXAM TIP ST, SC, LC and MT-RJ fiber connectors will likely be questioned on both exams.

Light can be sent down a fiber optic cable as regular light or as laser light. Each type of light requires totally different fiber optic cables. Most network technologies that use fiber optics use light-emitting diodes (LEDs) to send light signals. These use *multimode* fiber optic cabling. Multimode fiber transmits multiple light signals at the same time, each using a different reflection angle within the core of the cable. The multiple reflection angles tend to disperse over long distances, so multimode fiber optic cables are used for relatively short distances.

Network technologies that use laser light use *single-mode* fiber optic cabling. Using laser light and single-mode fiber optic cables allows for phenomenally high transfer rates over long distances. Except for long-distance links, single-mode is currently quite rare; if you see fiber optic cabling, you can be relatively sure it is multimode.

The two most common fiber optic standards are called 1000BaseSX and 10GBaseSR. The major difference is the speed of the network (there are also some important differences in the way systems interconnect, and so on). Fiber optic cabling is delicate, expensive, and difficult to use, so it is usually reserved for use in data centers and is rarely used to connect desktop PCs.

Coax/BNC

Early versions of Ethernet ran on *coaxial cable* instead of UTP. Coax consists of a center cable (core) surrounded by insulation. This in turn is covered with a *shield* of braided cable. The inner core actually carries the signal. The shield effectively eliminates outside interference. The entire cable is then surrounded by a protective insulating cover. This type of coax looks like a skinny version of the RG-59 or RG-6 coax used by your cable television, but it is quite different. The RG rating is clearly marked on the cable. If it isn't, the cable should say something like "Thinnet" or "802.3" to let you know you had the right cable. To connect the cable to individual machines, a twist-on *BNC connector* is used.

Parallel/Serial

It would be unfair not to give at least a token nod to using the parallel or serial ports on a pair of PCs to make a direct cable connection. All versions of Windows have complete support for allowing two, and no more than two, systems to network together, using either parallel or serial cables. You need crossover versions of IEEE 1284 cables for parallel and RS-232 cables for serial. These should be considered only as a last resort option, given the incredibly slow speeds of parallel and especially serial cable transmission compared to that of Ethernet. You should never use direct cable connections unless no other viable alternative exists.

FireWire

You can connect two computers by using FireWire cables. Apple designed FireWire to be network aware, so the two machines will simply recognize each other and, assuming they're configured to share files and folders, you're up and running. See the section "Sharing and Security" later in this chapter for more details.

USB

You can also connect two computers by using USB, but it's not quite as elegant as FireWire. The most common way is to plug a USB NIC into each PC and then run a UTP crossover cable between the Ethernet ports. You also can buy a special USB crossover cable to connect the two machines. Finally, at least one company makes a product that enables you to connect with a normal USB cable, called USB Duet.

Practical Application

Network Operating Systems

At this point in the discussion of networking, you've covered two of the four main requirements for making a network work. Through Ethernet, you have a NIC for the PC that handles splitting data into packets and putting the packets back together at the destination PC. You've got a cabling standard to connect the NIC to a hub or switch,

thus making that data transfer possible. Now it's time to dive into the third and fourth requirements for a network. You need an operating system that can communicate with the hardware and with other networked PCs, and you need some sort of server machine to give out data or services. The third and fourth requirements are handled by a network operating system.

EXAM TIP Both CompTIA A+ exams assume you have a working knowledge of network operating systems.

In a classic sense, a *network operating system* (*NOS*) is a portion of your operating system that communicates with the PC hardware and makes the connections among multiple machines on a network. The NOS enables one or more PCs to act as server machines and share data and services over a network—to share *resources*, in other words. You then need to run software on client computers so those computers can access the shared resources on the server machine.

Before you can share resources across a network, you must answer a number of questions. How do you make a resource available to share? Can everyone share his or her hard drives with everyone else? Should you place limits on sharing? If everyone needs access to a particular file, where will it be stored? What about security? Can anyone access the file? What if someone erases it accidentally? How are backups to be handled? Different versions of Windows answer these questions differently. Let's look at network organization and then turn to protocols, client software, and server software.

Network Organization

All NOSs can be broken into three basic organizational groups: client/server, peer-to-peer, and domain-based. Let's take a look at traditional network organization.

Client/Server

In a *client/server network*, one machine is dedicated as a resource to be shared over the network. This machine will have a dedicated NOS, optimized for sharing files. This special OS includes powerful caching software that enables high-speed file access. It will have extremely high levels of protection and an organization that permits extensive control of the data. This machine is called a *dedicated server*. All of the other machines that use the data are called *clients* (because it's what they usually are) or *workstations*.

The client/server system dedicates one machine to act as a server, whose purpose is to serve up resources to the other machines on the network. These servers do not run Windows XP or Vista. They use highly sophisticated and expensive NOSs that are optimized for the sharing and administration of network resources. Dedicated server operating systems include Windows Server 2008, big UNIX systems such as IBM AIX and HP-UX, and some versions of Linux.

NOTE The terms *client* and *server* are, to say the least, freely used in the Windows world. Keep in mind that a *client* generally refers to any process (or in this context, computer system) that can request a resource or service, and a server is any process (or system) that can fulfill the request.

Peer-to-Peer

Some networks do not require dedicated servers—every computer can perform both server and client functions. A *peer-to-peer network* enables any or all of the machines on the network to act as a server. Peer-to-peer networks are much cheaper than client/server networks because the software costs less and does not require that you purchase a high-end machine to act as the dedicated server. The most popular peer-to-peer NOSs today are the various versions of Windows and Macintosh OS X.

The biggest limiting factor to peer-to-peer networking is that it's simply not designed for a large number of computers. Windows has a built-in limit (10) to the number of users who can concurrently access a shared file or folder. Microsoft recommends that peer-to-peer workgroups not exceed 15 PCs. Beyond that, creating a domain-based network makes more sense (see the following section).

Security is the other big weakness of peer-to-peer networks. Each system on a peer-to-peer network maintains its own security.

With the Windows Professional/Business versions, you can tighten security by setting NTFS permissions locally, but you are still required to place a local account on every system for any user who's going to access resources. So even though you get better security in a Windows Professional/Business peer-to-peer network, system administration entails a lot of running around to individual systems to create and delete local users every time someone joins or leaves the workgroup. In a word: bleh.

Peer-to-peer workgroups are little more than a pretty way to organize systems to make navigating through Windows networks a little easier (Figure 23-14). In reality, workgroups have no security value. Still, if your networking needs are limited—such as a small home network—peer-to-peer networking is an easy and cheap solution.

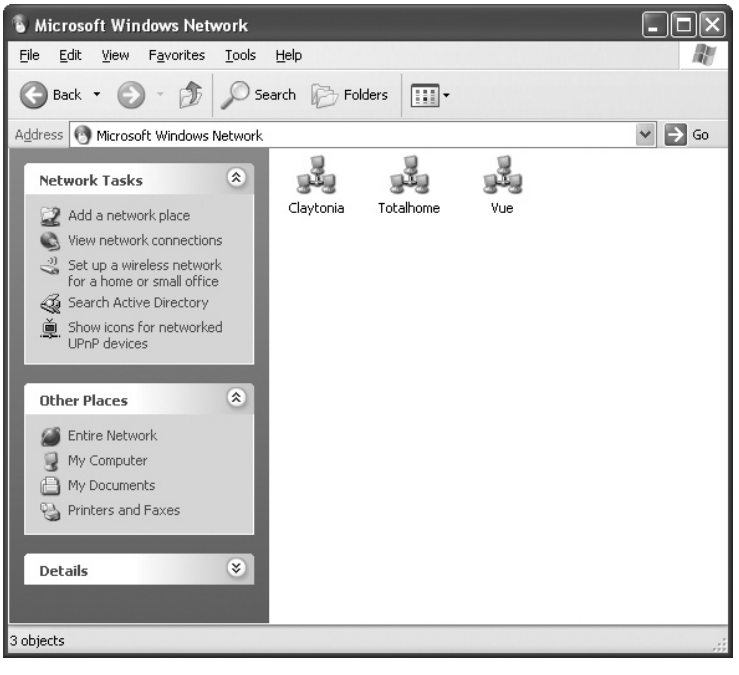


Figure 23-14 Multiple workgroups in a network

Domain-Based

One of the similarities between the client/server network model and peer-to-peer networks is that each PC in the network maintains its own list of user accounts. If you want to access a server, you must log on. When only one server exists, the logon process takes only a second and works very well. The trouble comes when your network contains multiple servers. In that case, every time you access a different server, you must repeat the logon process (Figure 23-15). In larger networks containing many servers, this becomes a time-consuming nightmare not only for the user, but also for the network administrator.

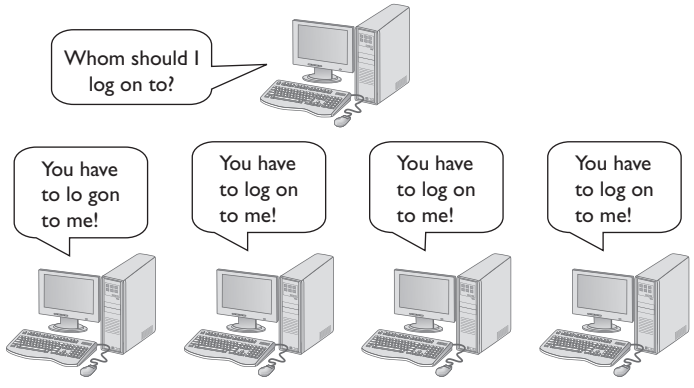


Figure 23-15 Multiple logins in a peer-to-peer network

A *domain-based network* provides an excellent solution for the problem of multiple logins. In a domain-based environment, one or more dedicated servers called *domain controllers* hold the security database for all systems. This database holds a list of all users and passwords in the domain. When you log on to your computer or to any computer, the logon request goes to an available domain controller to verify the account and password (Figure 23-16).

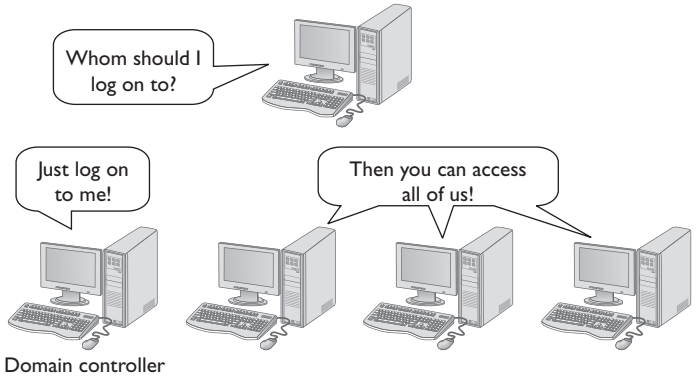


Figure 23-16 A domain controller eliminates the need for multiple logins.

Modern domain-based networks use what is called a *directory service* to store user and computer account information. Large Microsoft-based networks use the *Active Directory* (*AD*) directory service. Think of a directory service as a big, centralized index, similar to a telephone book, that each PC accesses to locate resources in the domain.

Server versions of Microsoft Windows look and act similar to the workstation versions, but they come with extra networking capabilities, services, and tools so they can take on the role of domain controller, file server, *remote access services* (*RAS*) server, application server, Web server, and so on. A quick glance at the options you have in Administrative Tools shows how much more full-featured the server versions are compared to the workstation versions of Windows. Figure 23-17 shows the Administrative Tools options on a typical Windows Vista workstation. These should be familiar to you. Figure 23-18 shows the many extra tools you need to work with Windows 2008 Server.

Figure 23-17 Administrative Tools in Windows Vista Business

Every Windows system contains a special account called the *administrator account*. This one account has complete and absolute power over the entire system. When you install Windows, you must create a password for the administrator account. Anyone who knows the administrator password can install/delete any program, read/change/delete any file, run any program, and change any system setting. As you might imagine, you should protect the administrator password carefully. Without it, you cannot create additional accounts (including additional accounts with administrative privileges) or

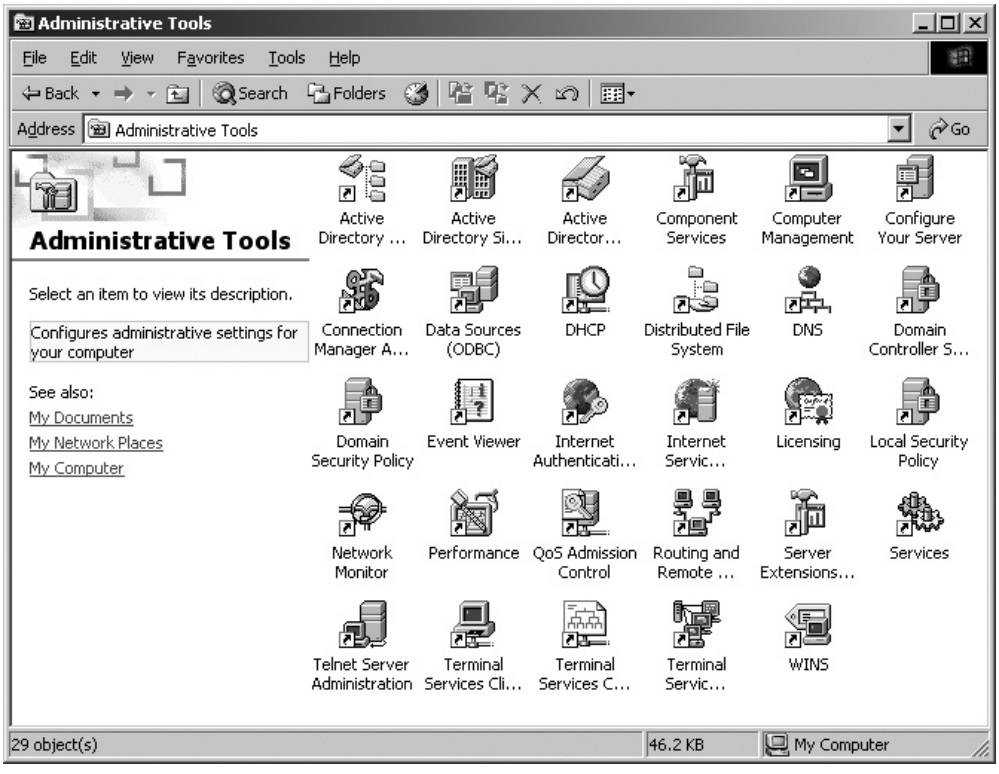


Figure 23-18 Administrative Tools in Windows Server 2003

change system settings. If you lose the administrator password (and no other account with administrative privileges exists), you have to reinstall Windows completely to create a new administrator account—so don't lose it!

In Windows XP, open the Properties window for My Computer, and select the Computer Name tab, as shown in Figure 23-19. This shows your current selection. Windows Vista and 7 show the computer name right on the System Properties dialog box and give you a link to the 2000/XP-style dialog box (Figure 23-20). Clicking the Network ID button opens the Network Identification Wizard, but most techs just use the Change button, which brings up the Computer Name/Domain Changes dialog box (Figure 23-21). Clicking the Change button does the same thing as clicking the Network ID button except that the wizard does a lot of explaining that you don't need if you know what you want to do. Make sure you have a valid domain account or you won't be able to log into a domain.

At this point, you've prepared the OS to network in general, but now you need to talk to the specific hardware. For that, you need to load protocols.

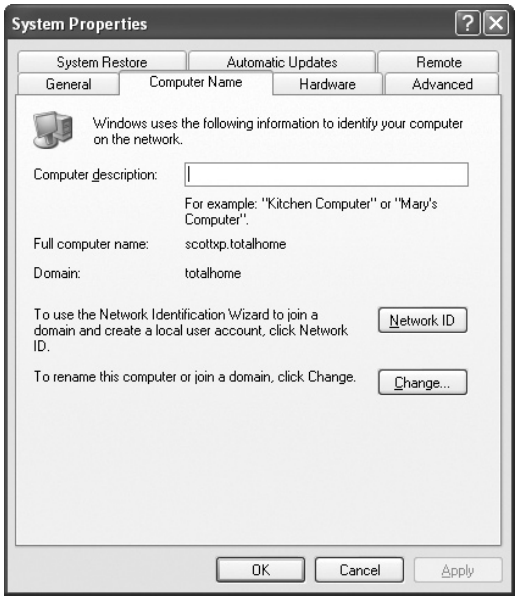


Figure 23-19 Computer Name tab in Windows XP

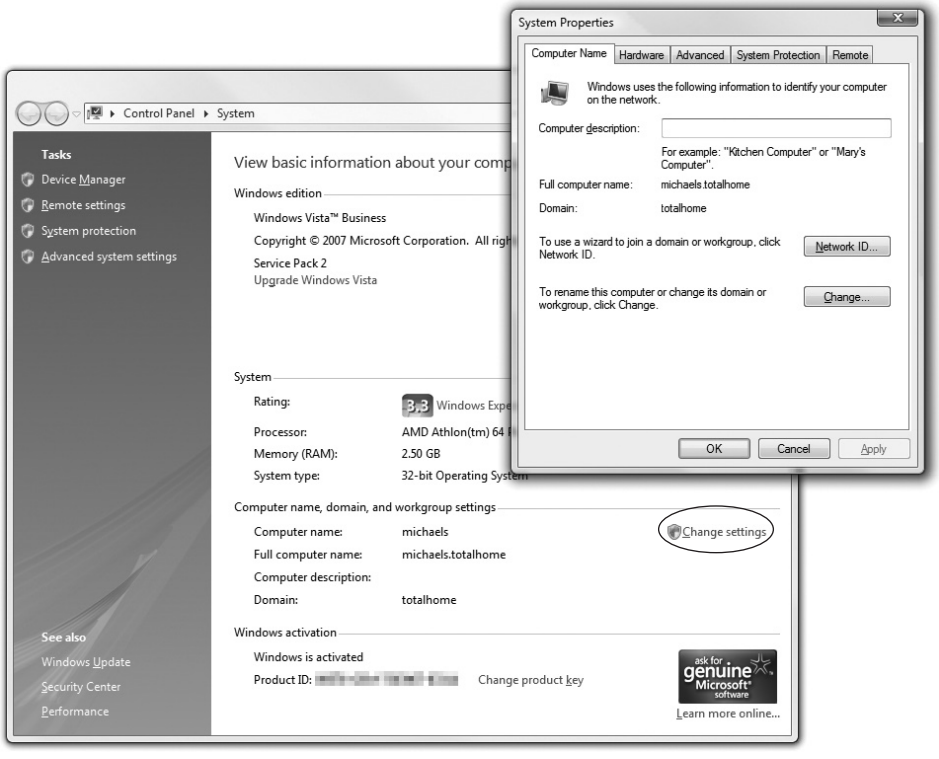


Figure 23-20 Computer Name location in Vista

Figure 23-21 Using the Change button

Protocols

Simply moving data from one machine to another is hardly sufficient to make a complete network; many other functions need to be handled. For example, if a file is being copied from one machine to another, something must keep track of all of the packets so the file can be properly reassembled. If many machines are talking to the same machine at once, that machine must somehow keep track of which packets it sends to or receives from each of the other PCs.

Another issue arises if one of the machines in the network has its network card replaced. Up to this point, the only way to distinguish one machine from another was by the MAC address on the network card. To solve this, each machine must have a name, an identifier for the network, which is "above" the MAC address. Each machine, or at least one of them, needs to keep a list of all of the MAC addresses on the network and the names of the machines, so that packets and names can be correlated. That way, if a PC's network card is replaced, the network, after some special queries, can update the list to associate the name of the PC with its new network card's MAC address.

Network protocol software takes the incoming data received by the network card, keeps it organized, sends it to the application that needs it, and then takes outgoing data from the application and hands it to the NIC to be sent out over the network. All networks use some protocol. Although many protocols exist, one dominates the world of PCs—TCP/IP.

NetBEUI/NetBIOS

Before we talk about TCP/IP, we need to discuss a little history. During the 1980s, IBM developed *NetBIOS Extended User Interface* (*NetBEUI*), the default protocol for Windows for Workgroups, LANtastic, and Windows 95. NetBEUI offers small size, easy configuration, and a relatively high speed, but it can't be used for routing. Its inability to handle routing limits NetBEUI to networks smaller than about 200 nodes.

NOTE A *node* is any device that has a network connection—usually this means a PC, but other devices can be nodes. For example, many printers now connect directly to a network and can therefore be deemed nodes. I use the term node extensively in the rest of the chapter in place of *PC* or *networked computer*. This is especially true when I talk about wireless technologies, because that's the term the manufacturers use.

You can connect multiple smaller networks into a bigger network, turning a group of LANs into one big WAN, but this raises a couple of issues with network traffic. A computer needs to be able to address a packet so that it goes to a computer within its own LAN or to a computer in another LAN in the WAN. If every computer saw every packet, the network traffic would quickly spin out of control! Plus, the machines that connect the LANs—called *routers*—need to be able to sort those packets and send them along to the proper LAN. This process, called *routing*, requires routers and a routing-capable protocol to function correctly.

NetBEUI was great for a LAN, but it lacked the extra addressing capabilities needed for a WAN. A new protocol was needed, one that could handle routing.

TCP/IP

Transmission Control Protocol/Internet Protocol (*TCP/IP*) was originally developed for the Internet's progenitor, the *Advanced Research Projects Agency Network* (*ARPANET*) of the U.S. Department of Defense. In 1983, TCP/IP became the built-in protocol for the popular BSD (Berkeley Software Distribution) UNIX, and other flavors of UNIX quickly adopted it as well. TCP/IP is the best protocol for larger networks with more than 200 nodes. The biggest network of all, the Internet, uses TCP/IP as its protocol. Windows also uses TCP/IP as its default protocol.

NOTE Novell developed the *Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX)* protocol exclusively for its NetWare products. The IPX/SPX protocol is speedy, works well with routers, and takes up relatively little RAM when loaded. Although once popular, it has all but disappeared in favor of TCP/IP. Microsoft implements a version of IPX/SPX called *NWLink*.

Client Software

To access data or resources across a network, Windows needs to have client software installed for every kind of server you want to access. When you install a network card and drivers, Windows installs at least one set of client software, called Client for Microsoft Networks (Figure 23-22). This client enables your machine to do the obvious: connect to a Microsoft network! Internet-based services work the same way. You need a Web client (such as Mozilla Firefox) to access a Web server. Windows PCs don't just access shared data magically but require that client software be installed.

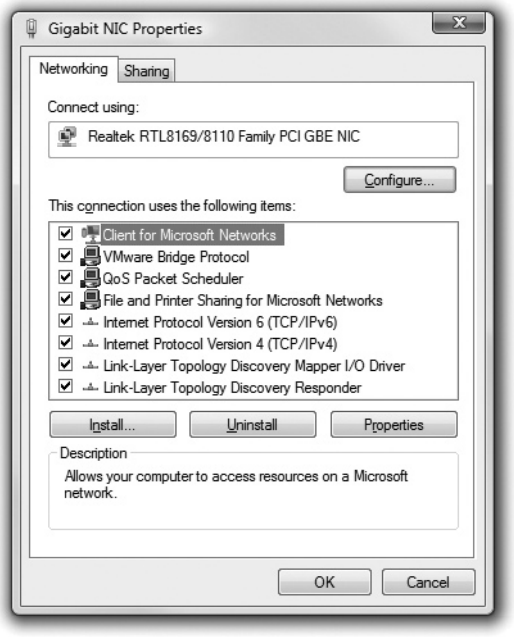


Figure 23-22 LAN Properties window showing Client for Microsoft Networks installed (along with other network software)

NOTE If you right-click on the Network button in Windows Vista/7, you're taken to the Network and Sharing Center. In the Network and Sharing Center, you can view the status of your network connection and easily enable or disable various network settings, such as file sharing, network discovery, and printer sharing. You can also see what type of network you're on: Public, Private, or Domain. Windows Vista lets you select which type of network you're on, either Public or Private, the first time you join a particular network and modifies your network settings based on the type of network you select. Public networks are assumed not to be secure; as such, Windows automatically turns off all of the network sharing options so that bad people can't access your computer. Private networks are assumed safe, so all of the file sharing options are turned on. If your computer is on a domain, your network administrator will control your network options.

Server Software

You can turn any Windows PC into a server simply by enabling the sharing of files, folders, and printers. Windows has file and printer sharing installed but not activated by default (though a simpler form of file sharing, creatively named Simple File Sharing, is enabled by default in Windows XP Home to make sharing media over a home network easier). Activating file and printer sharing requires nothing more than a click on a checkbox, as you can see in Figure 23-23.

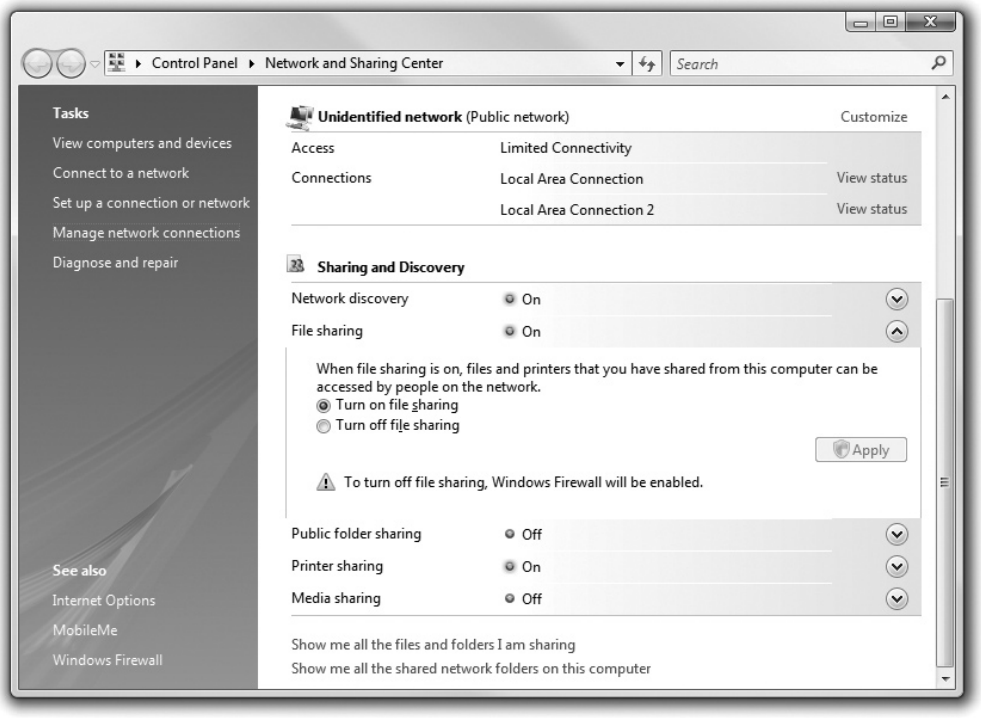


Figure 23-23 Enabling file and printer sharing in Windows Vista

NOTE Every version of Windows since Windows XP SP 2 has included a built-in firewall that blocks out harmful Internet traffic. Windows Firewall functions slightly differently in each version of Windows, but you should be aware of one quirk in Windows XP: namely that the Firewall will block file and printer sharing by default. So if you find that you can't access shared folders or printers, you can check to make sure Windows Firewall isn't blocking them. You can do this by going into Control Panel and opening the Windows Firewall applet. Once that's open, click the Exceptions tab and make sure that the checkbox next to File and Printer Sharing is checked. If it isn't, that's your problem!

Installing and Configuring a Wired Network

Almost halfway through the chapter and we're finally getting to the good stuff: installing and configuring a network! To have network connectivity, you need to have three things in place:

- **NIC** The physical hardware that connects the computer system to the network media.

- **Protocol** The language that the computer systems use to communicate.
- **Network client** The interface that allows the computer system to speak to the protocol.

If you want to share resources on your PC with other network users, you also need to enable Microsoft's File and Printer Sharing. Plus, of course, you need to connect the PC to the network hub or switch via some sort of cable (preferably CAT 6 with Gigabit Ethernet cranking through the wires, but that's just me!). When you install a NIC, by default Windows installs the TCP/IP protocol, the Client for Microsoft Networks, and File and Printer Sharing for Microsoft Networks upon setup.

Installing a NIC

The NIC is your computer system's link to the network, and installing one is the first step required to connect to a network. NICs are manufactured to operate on specific media and network types, such as 1000BaseT Ethernet. Follow the manufacturer's instructions for installation. If your NIC is of recent vintage, it will be detected, installed, and configured automatically by Windows. You might need a driver disc or a driver download from the manufacturer's Web site if you install that funky PC Card or gamer NIC.

The Add Hardware Wizard automates installation of non–plug-and-play devices or plug-and-play devices that were not detected correctly. Start the wizard by clicking Start | Settings | Control Panel (2000 or classic start menu) or Start | Control Panel (XP – 7) and then double-clicking the icon for the Add Hardware applet. (Note that Windows 2000 calls this the Add/Remove Hardware applet.) Click the Next button to select the hardware task you wish to perform, and follow the prompts to complete the wizard.

NOTE If you have the option, you should save yourself potential headaches and troubleshooting woes by acquiring new, name-brand NICs for your Windows installation.

Configuring a Network Client

To establish network connectivity, you need a network client installed and configured properly. You need a client for every type of server NOS to which you plan to connect on the network. Let's look at Microsoft's client.

Installed as part of the OS installation, the Client for Microsoft Networks rarely needs configuration, and, in fact, few configuration options are available. To start it in Windows Vista/7, click Start; then right-click Network and select Properties. Then click *Manage network connections* on the left. In Windows XP, click Start, and then right-click My Network Places and select Properties. In Windows 2000, click Start | Settings | Network and Dial-up Connections.

In all versions of Windows, your next step is to double-click the Local Area Connection icon, click the Properties button, highlight Client for Microsoft Networks, and click the Properties button. Note that there's not much to do here. Unless told to do something by a network administrator, just leave this alone.

Configuring TCP/IP

This final section on protocols covers TCP/IP, the primary protocol of most modern networks, including the Internet. For a PC to access the Internet, it must have TCP/IP loaded and configured properly. TCP/IP has become so predominant that most network folks use it even on networks that do not connect to the Internet. Although TCP/IP is powerful, it is also a bit of a challenge to set up. So whether you are installing a modem for a dial-up connection to the Internet or setting up 500 computers on their own private *intranet*, you must understand some TCP/IP basics. You'll go through the following basic sections of the protocol and then you'll look at specific steps to install and configure TCP/IP.

Network Addressing

Any network address must provide two pieces of information: it must uniquely identify the machine and it must locate that machine within the larger network. In a TCP/IP network, the IP address identifies the PC and the network on which it resides.

IP Addresses In a TCP/IP network, the systems don't have names but rather use IP addresses. The *IP address* is the unique identification number for your system on the network. Part of the address identifies the network, and part identifies the local computer (host) address on the network. IP addresses consist of four sets of eight binary numbers (octets), each set separated by a period. This is called *dotted-decimal notation*. So, instead of a computer being called SERVER1, it gets an address like so:

```
202.34.16.11
```

Written in binary form, the address would look like this:

```
11001010.00100010.00010000.00001011
```

To make the addresses more comprehensible to users, the TCP/IP folks decided to write the decimal equivalents:

```
00000000 = 0
00000001 = 1
00000010 = 2
...
11111111 = 255
```

IP addresses are divided into class licenses that correspond with the potential size of the network: Class A, Class B, and Class C. Class A licenses were intended for huge companies and organizations, such as major multinational corporations, universities, and governmental agencies. Class B licenses were assigned to medium-size companies, and Class C licenses were designated for smaller LANs. Class A networks use the first octet to identify the network address and the remaining three octets to identify the host. Class B networks use the first two octets to identify the network address and the remaining two octets to identify the host. Class C networks use the first three octets to identify the network address and the last octet to identify the host. Table 23-2 lists range (class) assignments.

IP Address Shortage Solution

The IP addresses I'm showing you here are technically IP version 4, or IPv4 addresses, but this type of addressing has a bit of a problem—namely, that we're running out of possible IP addresses, and there won't be any left in a few years. No big deal. Now, before you go running out into the streets shouting about the impending demise of the Internet or start hoarding canned food in your basement, let me tell you about the solution.

IP version 6, the newest version of the Internet protocol, which will save us all from an Internetless world, uses a 128-bit address instead of IPv4's 32-bit address. What this means is that there are more possible addresses than with IPv4. A lot more. My favorite illustration is to think of all of the molecules that make up the Earth, and divide them by 7. That's how many possible IPv6 addresses there are.

The drawback is that IPv6 addresses are not quite as svelte and easy to remember as in IPv4. For example, an IPv6 address looks like this: 2001:0db8:85a3:0000:0000:8a2e:0370:7334. Not quite as easy to work with as 192.168.1.1, eh?

IPv6 also handles routing and various other things differently than IPv4, but the main things to know are that the IP addresses look remarkably different and there are enough of them to last for a while. There's no solid plan yet for when everyone is going to switch to IPv6, but it'll be a big change when it happens.

Network Class	Address Range	No. of Network Addresses Available	No. of Host Nodes (Computers) Supported
A	1–126	129	16,777,214
B	128–191	16,384	65,534
C	192–223	2,097,152	254

Table 23-2 Class A, B, and C Addresses

You'll note that the IP address ranges listed above skip from 126.x.x.x to 128.x.x.x. That's because the 127 address range (i.e., 127.0.0.1–127.255.255.255) is reserved for network testing (loopback) operations. (We usually just use the address 127.0.0.1 for loopback purposes and call it the *localhost* address, but any address that starts off with *127* will work just as well.) That's not the only reserved range, either! Each network class has a specific IP address range reserved for *private* networks—traffic from these networks doesn't get routed to the Internet at large. Class A's private range goes from 10.0.0.1 to 10.255.255.255. Class B has two private address ranges: 172.16.0.1 up to 172.16.255.255 for manually configured addresses and 169.254.0.1 to 169.254.255.254 (link-local addresses) to accommodate the *Automatic Private IP Addressing* (*APIPA*) function discussed later. Class C's private addresses range from 192.168.0.0 to 192.168.255.255.

NOTE Pinging the loopback is the best way to test whether a NIC is working properly. To test a NIC's loopback, the other end of the cable must be in a working switch or you must use a loopback device such as a loopback adapter/plug.

NOTE If APIPA is enabled and the DHCP configured client can't reach a DHCP server, the client will automatically be configured with an APIPA link-local IP address in the range between 169.254.0.1 to 169.254.255.254 and get a Class B subnet mask of 255.255.0.0 until the DHCP server can be reached.

Subnet Mask The *subnet mask* is a value that distinguishes which part of the IP address is the network address and which part of the address is the host address. The subnet mask blocks out (or masks) the network portions (octets) of an IP address. Certain subnet masks are applied by default. The default subnet mask for Class A addresses is 255.0.0.0; for Class B, it's 255.255.0.0; and for Class C, 255.255.255.0. For example, in the Class B IP address 131.190.4.121 with a subnet mask of 255.255.0.0, the first two octets (131.190) make up the network address, and the last two (4.121) make up the host address.

EXAM TIP The CompTIA A+ certification exams do not require you to break down IP addresses and subnet masks into their binary equivalents or to deal with non-standard subnet masks such as 255.255.240.0, but you should know what IP addresses and subnet masks are and how to configure your PC to connect to a TCP/IP network.

A New Kind of Port

The term "port" has several meanings in the computer world. Commonly, port defines the connector socket on an Ethernet NIC, where you insert an RJ-45 jack. That's how I've used the term for the most part in this book. It's now time to see another use of the word ports.

In TCP/IP, *ports* are 16-bit numbers between 0 and 65,535, assigned to a particular TCP/IP session. All TCP/IP packets (except for some really low-level maintenance packets) contain port numbers that the two communicating computers use to determine not only the kind of session—and thus what software protocol—to use to handle the data in the packet, but also how to get the packet or response back to the sending computer.

Each packet has two ports assigned, a destination port and an ephemeral port. The *destination port* is a fixed, predetermined number that defines the function or session type. Common TCP/IP session types use destination port numbers in the range 0–1023. The *ephemeral port* is an arbitrary number generated by the sending computer; the receiving computer uses the ephemeral port as a destination address so that the sending computer knows which application to use for the returning packet. Ephemeral ports

usually fall in the 1024–5000 range, but this varies slightly among the different operating systems.

Ports enable one computer to serve many different services, such as a Web server and e-mail server, at the same time. We will discuss the most common ports and the associated services in the next chapter.

TCP/IP Services

TCP/IP is a different type of protocol. Although it supports File and Printer Sharing, it adds a number of special sharing functions unique only to it, lumped together under the umbrella term *TCP/IP services*. The most famous TCP/IP service is called *Hypertext Transfer Protocol* (*HTTP*), the language of the World Wide Web. If you want to surf the Web, you must have TCP/IP. But TCP/IP supplies many other services beyond just HTTP. By using a service called Telnet, for example, you can access a remote system as though you were actually in front of that machine.

Another example is a handy utility called PING. *PING* enables one machine to check whether it can communicate with another machine. Figure 23-24 shows an example of PING running on a Windows Vista system. Isn't it interesting that many TCP/IP services run from a command prompt? Good thing you know how to access one! I'll show you other services in a moment.

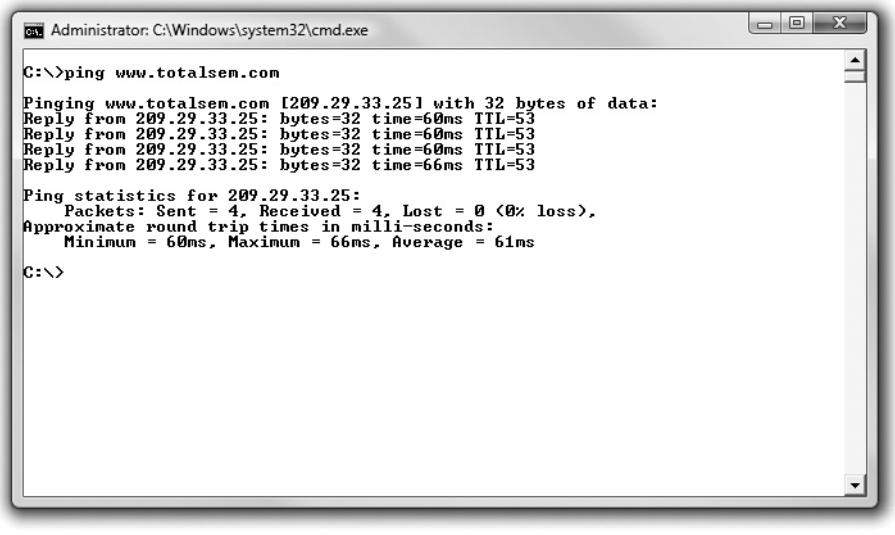


Figure 23-24 PING in action

The goal of TCP/IP is to link any two hosts (remember, a host is just a computer in TCP/IP lingo), whether the two computers are on the same LAN or on some other network within the WAN. The LANs within the WAN are linked together with a variety of connections, ranging from basic dial-ups to dedicated high-speed (and expensive) data

lines (Figure 23-25). To move traffic between networks, you use routers (Figure 23-26). Each host sends traffic to the router only when that data is destined for a remote network, cutting down on traffic across the more expensive WAN links. The host makes these decisions based on the destination IP address of each packet.

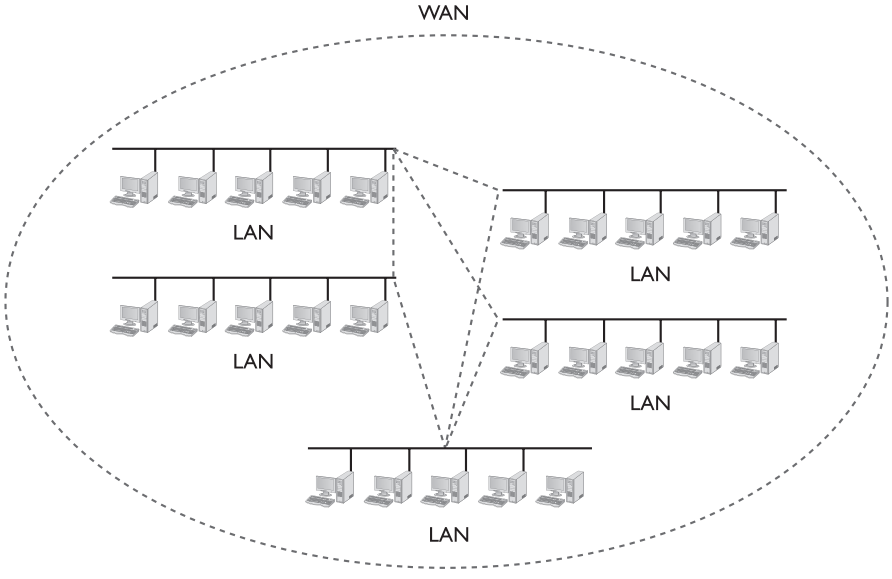


Figure 23-25 WAN concept

Figure 23-26
Typical router

TCP/IP Settings

TCP/IP has a number of unique settings that you must set up correctly to ensure proper network functioning. Unfortunately, these settings can be quite confusing, and there are quite a few of them. Not all settings are used for every type of TCP/IP network, and it's not always obvious where you go to set them.

Windows makes this fairly easy by letting you configure both dial-up and network connections by using the Network Connections dialog box (Figure 23-27). To get there, right-click on My Network Places (Windows 2000/XP) or Network (Windows Vista/7) and select Properties. In Vista/7, you have to click the *Manage network connections* button, but in 2000 and XP, you simply select the connection you wish to configure and then set its TCP/IP properties.

Figure 23-27 Network Connections dialog box showing dial-up and LAN connections

The CompTIA A+ certification exams assume that someone else, such as a tech support person or some network guru, will tell you the correct TCP/IP settings for the network. Your only job is to understand roughly what they do and to know where to enter them so the system works. Following are some of the most common TCP/IP settings.

EXAM TIP The CompTIA A+ certification exams have a rather strange view of what you should know about networking. Take a lot of time practicing how to get to certain network configuration screens. Be ready for questions that ask, "Which of the following steps will enable you to change a particular value?"

Default Gateway A computer that wants to send data to another machine outside its LAN is not expected to know exactly how to reach every other computer on the Internet. Instead, all IP hosts know the address of at least one router to which they pass all of the data packets they need to send outside the LAN. This router is called the *default gateway*, which is just another way of saying "the local router" (Figure 23-28).

Domain Name Service (DNS) Knowing that users could not remember lots of IP addresses, early Internet pioneers came up with a way to correlate those numbers with more human-friendly computer designations. Special computers, called *domain name service* (*DNS*) servers, keep databases of IP addresses and their corresponding names. For example, a machine called TOTALSEMINAR1 will be listed in a DNS directory with a corresponding

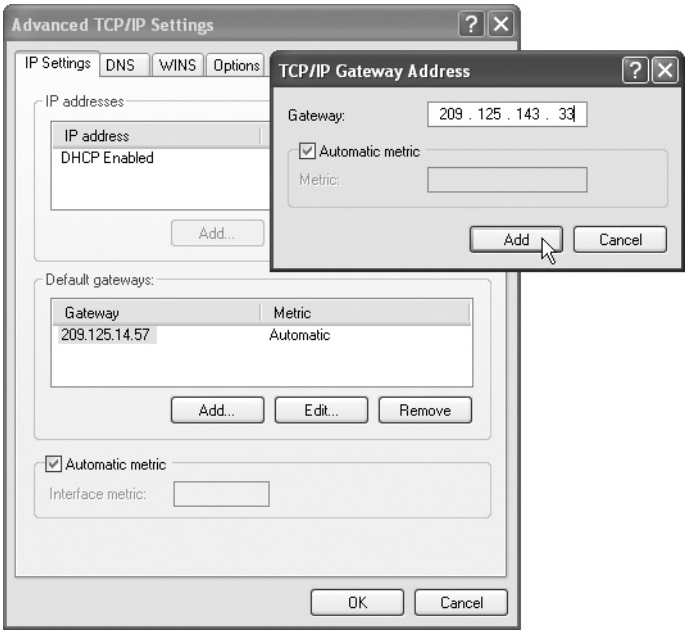


Figure 23-28 Setting a default gateway

IP address, such as 209.34.45.163. So instead of accessing the \\209.34.45.163\FREDC share to copy a file, you can ask to see \\TOTALSEMINAR1\FREDC. Your system will then query the DNS server to get TOTALSEMINAR1's IP address and use that to find the right machine. Unless you want to type in IP addresses all of the time, a TCP/IP networks will need at least one DNS server (Figure 23-29).

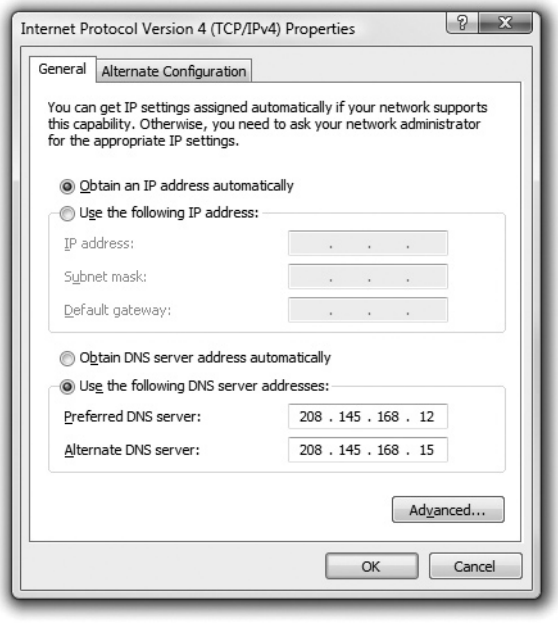


Figure 23-29 Adding two DNS servers in Windows Vista

The Internet has regulated domain names. If you want a domain name that others can access on the Internet, you must register your domain name and pay a small yearly fee. In most cases, your ISP can handle this for you. Originally, DNS names all ended with one of the following seven domain name qualifiers, called *top level domains* (*TLDs*):

.com	General business	**.org**	Nonprofit organizations
.edu	Educational organizations	**.gov**	Government organizations
.mil	Military organizations	**.net**	Internet organizations
.int	International		

As more and more countries joined the Internet, an entire new level of domains was added to the original seven to indicate a DNS name in a particular country, such as .uk for the United Kingdom. It's common to see DNS names such as www.bbc.co.uk or www.louvre.fr. The *Internet Corporation for Assigned Names and Numbers* (*ICANN*) announced the creation of several more new domains, including .name, .biz, .info, and others. Given the explosive growth of the Internet, these are unlikely to be the last ones! For the latest developments, check ICANN's Web site at www.icann.org.

WINS Before Microsoft came fully on board with Internet standards such as TCP/IP, the company implemented its own type of name server: *Windows Internet Name Service* (*WINS*). WINS enables NetBIOS network names such as SERVER1 to be correlated to IP addresses, just as DNS does, except these names are *Windows* network names such as SERVER1, not fully qualified domain Internet names (FQDNs) such as server1.example .com. NetBIOS names must be unique and contain 15 or fewer characters, but other than that there isn't much to it. Assuming that a WINS server exists on your network, all you have to do to set up WINS on your PC is type in the IP address for the WINS server (Figure 23-30). Windows 2000-7 based networks don't use WINS; they use an improved "dynamic" DNS (DDNS) that supports both Internet names and Windows names. On older networks that still need to support the occasional legacy Windows NT 4.0 server, you may need to configure WINS, but on most TCP/IP networks you can leave the WINS setting blank.

DHCP The last feature that most TCP/IP networks support is *dynamic host configuration protocol* (*DHCP*). To understand DHCP, you must first remember that every machine must be assigned an IP address, a subnet mask, a default gateway, and at least one DNS server (and maybe a WINS server). These settings can be added manually by using the TCP/IP Properties window. When you set the IP address manually, the IP address will not change and is called a *static IP address* (Figure 23-31).

DHCP enables you to create a pool of IP addresses that are given temporarily to machines. DHCP is especially handy for networks of a lot of laptops that join and leave the network on a regular basis. Why give a machine that is on the network for only a few hours a day a static IP address? For that reason, DHCP is quite popular. If you add a NIC to a Windows system, the default TCP/IP settings are set to use DHCP. When you accept those automatic settings, you're really telling the machine to use DHCP (Figure 23-32).

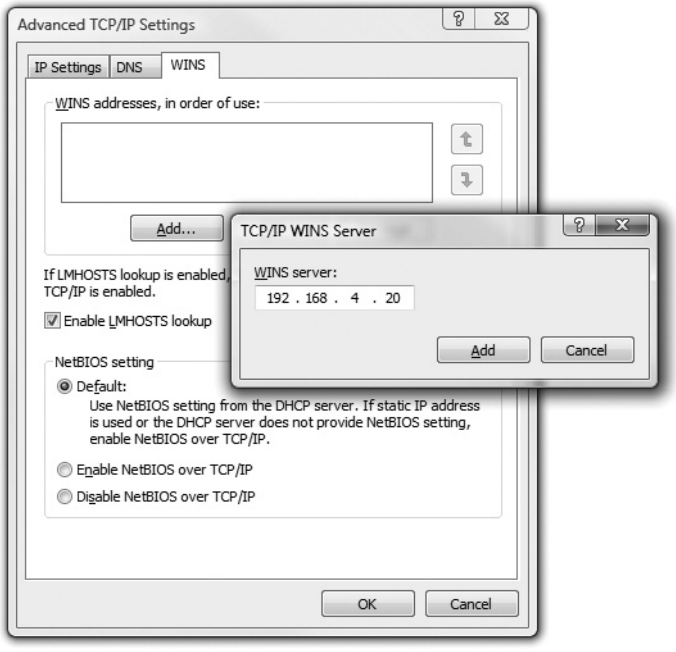


Figure 23-30 Setting up WINS to use DHCP

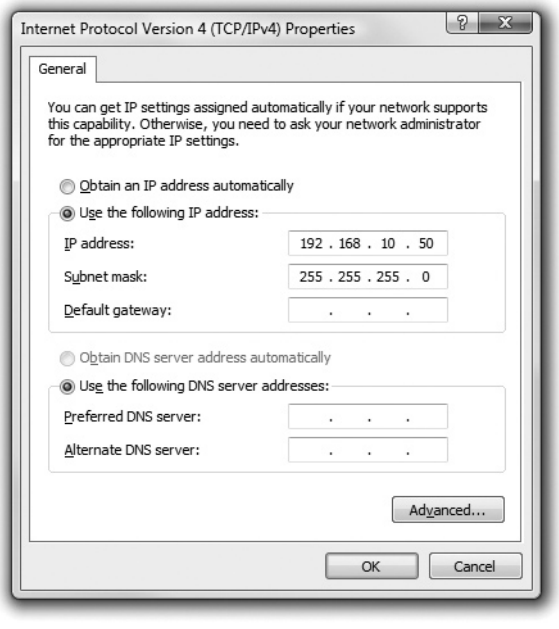


Figure 23-31 Setting a static IP address

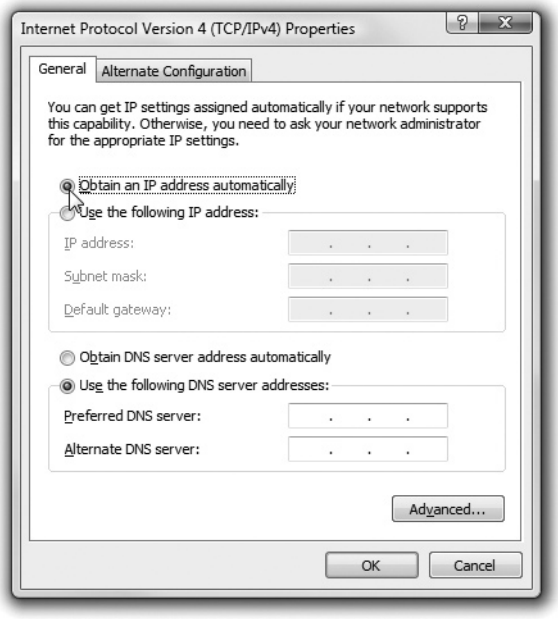


Figure 23-32 Automatically obtain an IP address

TCP/IP Tools

All versions of Windows come with handy tools to test TCP/IP. Those you're most likely to use in the field are PING, IPCONFIG, NSLOOKUP, and TRACERT. All of these programs are command prompt utilities. Open a command prompt to run them; if you just place these commands in the Run command, you'll see the command prompt window open for a moment and then quickly close!

PING You've already seen *PING*, a really great way to see if you can talk to another system. Here's how it works. Get to a command prompt and type **ping** followed by an IP address or by a DNS name, such as **ping www.chivalry.com**. Press the ENTER key on your keyboard and away it goes! Figure 23-33 shows the common syntax for PING.

PING has a few options beyond the basics that CompTIA wants you to know about. The first option is *–t*. By using the *–t* switch, PING continuously sends PING packets until you stop it with the break command (CTRL-C). The second option is the *–l* switch that enables you to specify how big a PING packet to send. This helps in diagnosing specific problems with the routers between your computer and the computer you PING.

IPCONFIG Windows offers the command-line tool *IPCONFIG* for a quick glance at your network settings. Click Start | Run and type **CMD** to get a command prompt. From the prompt, type **IPCONFIG /ALL** to see all of your TCP/IP settings (Figure 23-34).

When you have a static IP address, IPCONFIG does little beyond reporting your current IP settings, including your IP address, subnet mask, default gateway, DNS servers, and WINS servers. When using DHCP, however, IPCONFIG is also the primary tool for releasing and renewing your IP address. Just type **ipconfig /renew** to get a new IP address or **ipconfig /release** to give up the IP address you currently have.

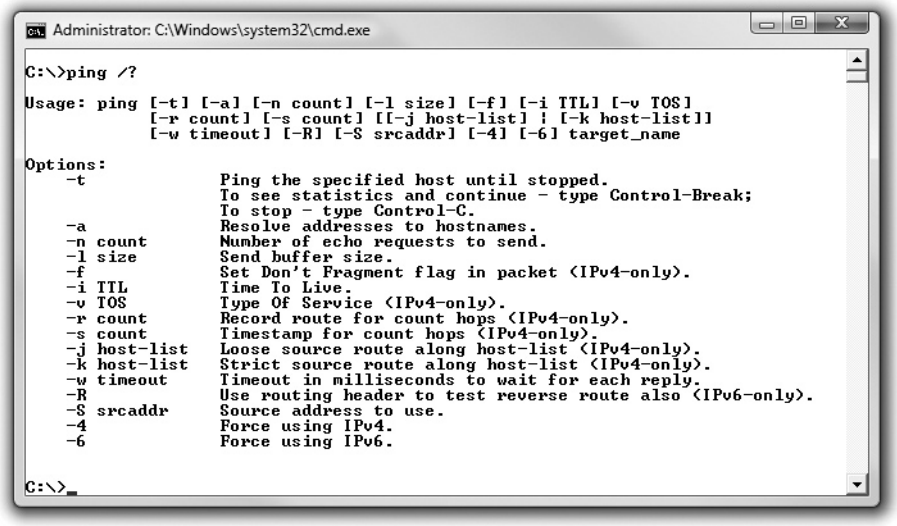


Figure 23-33 PING syntax

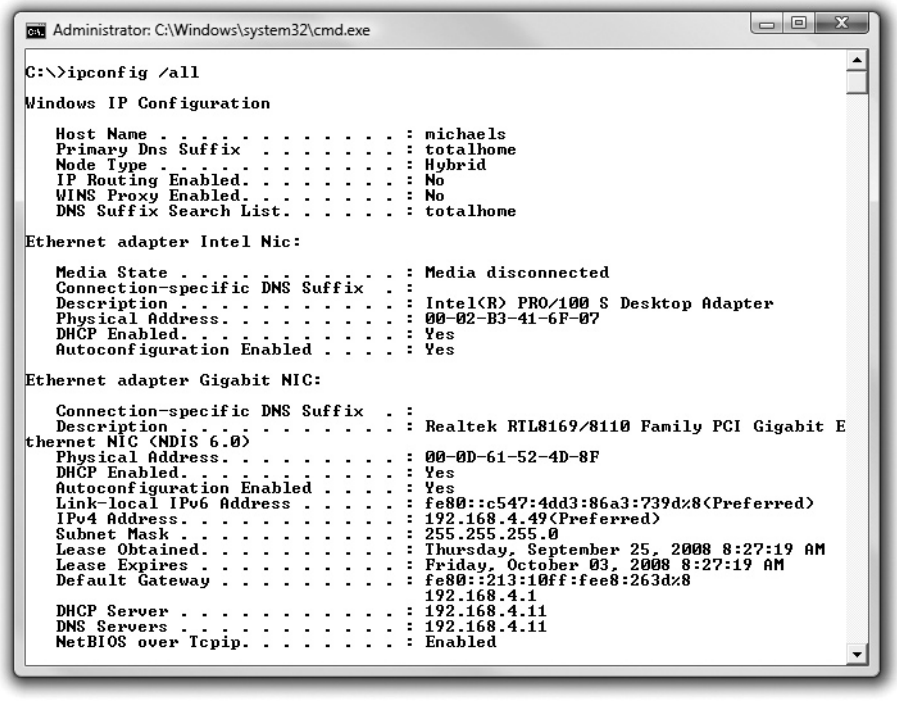


Figure 23-34 IPCONFIG /ALL on Windows Vista

NSLOOKUP *NSLOOKUP* is a powerful command-line program that enables you to determine exactly what information the DNS server is giving you about a specific host name. Every version of Windows makes NSLOOKUP available when you install TCP/IP. To run the program, type **NSLOOKUP** from the command line and press the ENTER key (Figure 23-35). Note that this gives you a little information but the prompt has changed? That's because you're running the application. Type exit and press the ENTER key to return to the command prompt.

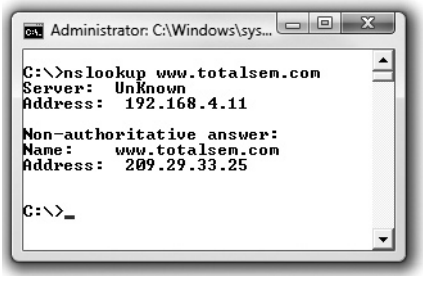


Figure 23-35 NSLOOKUP in action

NOTE You can do some cool stuff with NSLOOKUP, and consequently some techs absolutely love the tool. It's way outside the scope of CompTIA A+ certification, but if you want to play with it, type **HELP** at the NSLOOKUP prompt and press ENTER to see a list of common commands and syntax.

TRACERT The *TRACERT* utility shows the route that a packet takes to get to its destination. From a command line, type **TRACERT** followed by a space and an IP address. The output describes the route from your machine to the destination machine, including all devices the packet passes through and how long each hop takes (Figure 23-36).

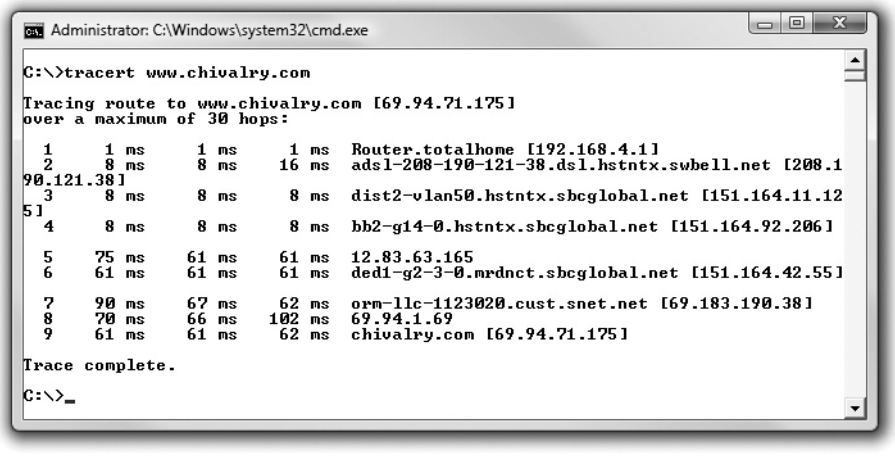


Figure 23-36 TRACERT in action

TRACERT can come in handy when you have to troubleshoot bottlenecks. When users complain of difficulty reaching a particular destination by using TCP/IP, you can run this utility to determine whether the problem exists on a machine or connection over which you have control, or if it is a problem on another machine or router. Similarly, if a destination is completely unreachable, TRACERT can again determine whether the problem is on a machine or router over which you have control.

Configuring TCP/IP

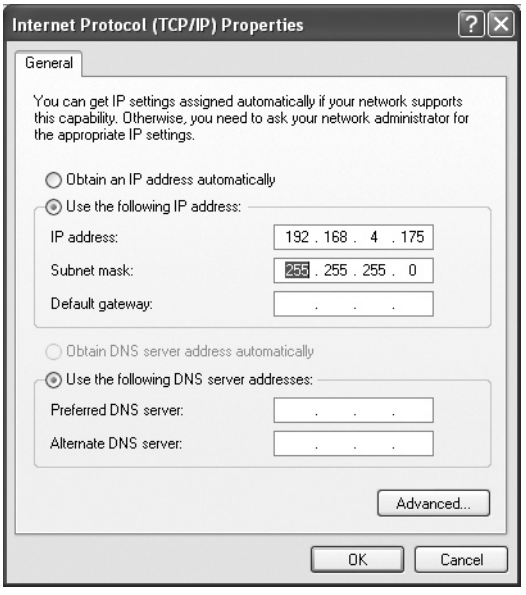
By default, TCP/IP is configured to receive an IP address automatically from a DHCP server on the network (and automatically assign a corresponding subnet mask). As far as the CompTIA A+ certification exams are concerned, Network+ techs and administrators give you the IP address, subnet mask, and default gateway information and you plug them into the PC. That's about it, so here's how to do it manually:

1. In Windows XP, open the Control Panel and double-click the Network Connections applet. Double-click the Local Area Connection icon. In Windows 2000, click Start | Settings | Network and Dial-up Connections, and double-click the Local Area Connection icon. In Windows Vista/7, right-click on Network and then click *Manage network connections*. After that, double-click the Local Area Network icon.
2. Click the Properties button, highlight Internet Protocol (TCP/IP), and click the Properties button. In Windows Vista/7, you should highlight Internet Protocol Version 4 (TCP/IPv4) because Vista and 7 both have IPv4 and IPv6 installed by default.
3. In the dialog box, click the radio button next to *Use the following IP address*.
4. Enter the IP address in the appropriate fields.
5. Press the TAB key to skip down to the Subnet mask field. Note that the subnet mask is entered automatically, although you can type over this if you want to enter a different subnet mask (see Figure 23-37).
6. Optionally, enter the IP address for a default gateway (a router or another computer system that will forward transmissions beyond your network).
7. Optionally, enter the IP addresses of a primary and a secondary DNS server.
8. Click the OK button to close the dialog box.
9. Click the Close button to exit the Local Area Connection Status dialog box.
10. Windows will alert you that you must restart the system for the changes to take effect.

Automatic Private IP Addressing

Windows supports a feature called Automatic Private IP Addressing (APIPA) that automatically assigns an IP address to the system when the client cannot obtain an IP address automatically. The Internet Assigned Numbers Authority, the nonprofit corporation responsible for assigning IP addresses and managing root servers, has set aside the range of addresses from 169.254.0.1 to 169.254.255.254 for this purpose.

Figure 23-37
Setting up IP



If the computer system cannot contact a DHCP server, the computer randomly chooses an address in the form of 169.254.*x.y* (where *x.y* is the computer's identifier) and a 16-bit subnet mask (255.255.0.0) and broadcasts it on the network segment (subnet). If no other computer responds to the address, the system assigns this address to itself. When using APIPA, the system can communicate only with other computers on the same subnet that also use the 169.254.*x.y* range with a 16-bit mask. APIPA is enabled by default if your system is configured to obtain an IP address automatically.

NOTE A computer system on a network with an active DHCP server that has an IP address in this range usually indicates a problem connecting to the DHCP server.

Sharing and Security

Windows systems can share all kinds of resources: files, folders, entire drives, printers, faxes, Internet connections, and much more. Conveniently for you, the CompTIA A+ certification exams limit their interests to folders, printers, and Internet connections. You'll see how to share folders and printers now; Internet connection sharing is discussed in Chapter 25, "The Internet."

Sharing Drives and Folders

All versions of Windows share drives and folders in basically the same manner. Simply right-click any drive or folder and choose Properties. Select the Sharing tab (Figure 23-38). Select *Share this folder*, add something in the Comment or User Limit fields if you wish (they're not required), and click Permissions (Figure 23-39).

Figure 23-38 Windows XP Sharing tab on NTFS volume

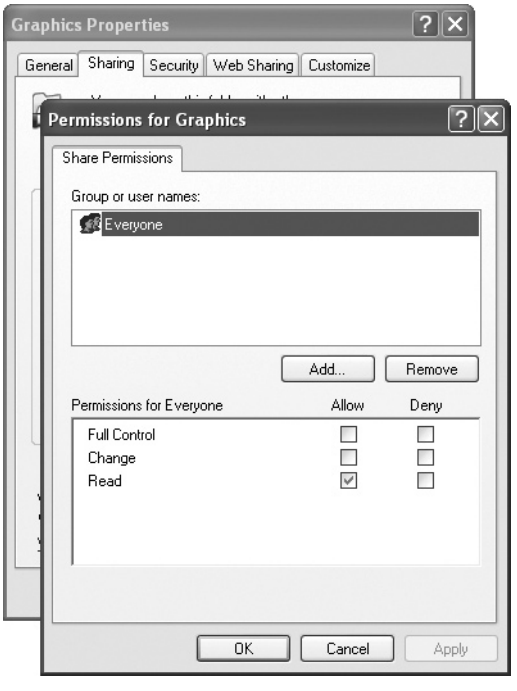


Figure 23-39 Network permissions

Hey! Doesn't NTFS have all those wild permissions such as Read, Execute, Take Ownership, and all that? Yes, it does, but NTFS permissions and network permissions are totally separate beasties. Microsoft wanted Windows to support many different file systems (NTFS, FAT16, FAT32), old and new. Network permissions are Microsoft's way of enabling you to administer file sharing on any type of partition supported by Windows, no matter how ancient. Sure, your options will be pretty limited if you are working with an older file system, but you *can* do it.

The beauty of Windows is that it provides another tool—NTFS permissions—that can do much more. NTFS is where the power lies, but power always comes with a price: You have to configure two separate sets of permissions. If you are sharing a folder on an NTFS drive, as you normally are these days, you must set *both* the network permissions and the NTFS permissions to let others access your shared resources. Some good news: This is actually no big deal! Just set the network permissions to give everyone full control, and then use the NTFS permissions to exercise more precise control over *who* accesses the shared resources and *how* they access them. Open the Security tab to set the NTFS permissions.

NOTE Windows offers two types of sharing: share-level and NTFS permissions.

Accessing Shared Drives/Directories

Once you have set up a drive or directory to be shared, the final step is to access that shared drive or directory from another machine. Windows 2000 and XP use My Network Places and Windows Vista and Windows 7 use Network, although you'll need to do a little clicking to get to the shared resources (Figure 23-40).

You can also map network resources to a local resource name. For example, the FREDC share can be mapped to be a local hard drive such as E: or F:. From within any Explorer window (such as My Documents or Documents), choose Tools | Map Network Drive to open the Map Network Drive dialog box (Figure 23-41). In Windows Vista/7, you'll need to press the ALT key once to see the menu bar. Click the Browse button to check out the neighborhood and find a shared drive (Figure 23-42).

In Windows 2000, you can also use the handy Add Network Place icon in My Network Places to add network locations you frequently access without using up drive letters. Windows XP removed the icon but added the menu option in its context bar on the left; Windows Vista and Windows 7 have removed it altogether. Here's how it looks on a Windows 2000 system (Figure 23-43).

Mapping shared network drives is a common practice, as it makes a remote network share look like just another drive on the local system. The only downside to drive mapping stems from the fact that users tend to forget they are on a network. A classic example is the user who always accesses a particular folder or file on the network and then suddenly gets a "file not found" error when the workstation is disconnected from the network. Instead of recognizing this as a network error, the user often imagines the problem is a missing or corrupted file.

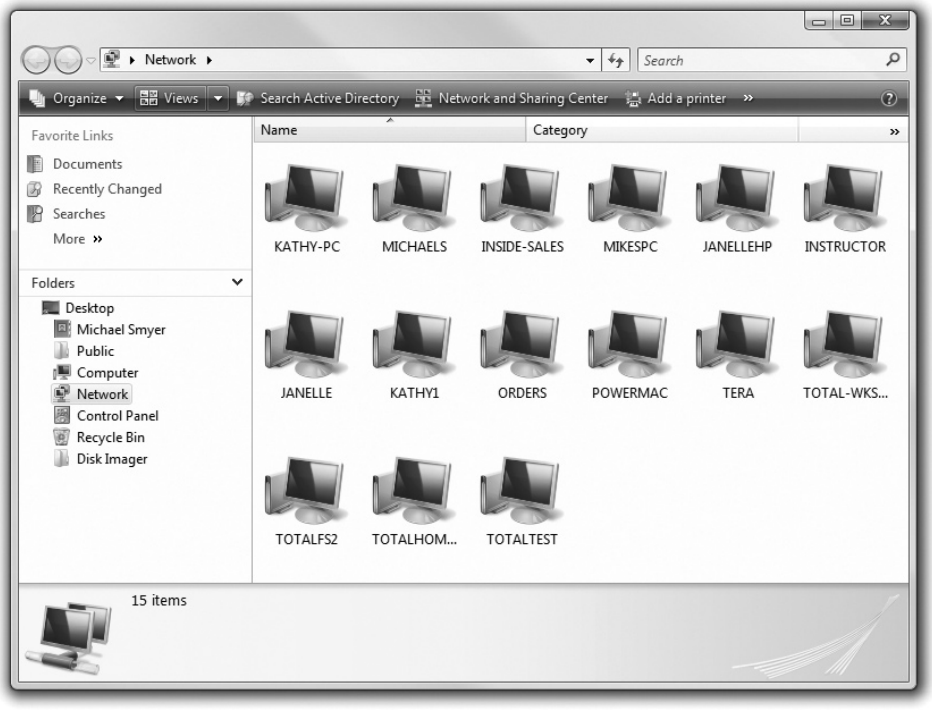


Figure 23-40 Shared resources in Network

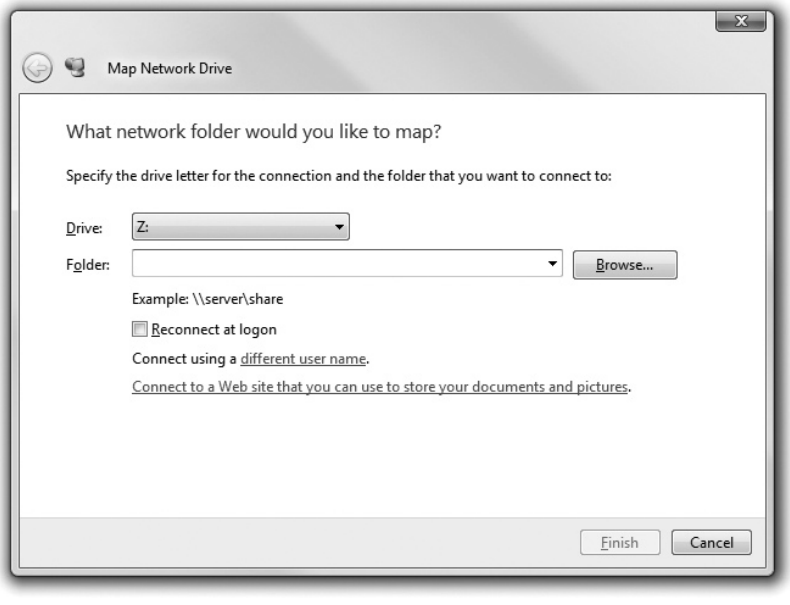


Figure 23-41 Map Network Drive dialog box in Vista

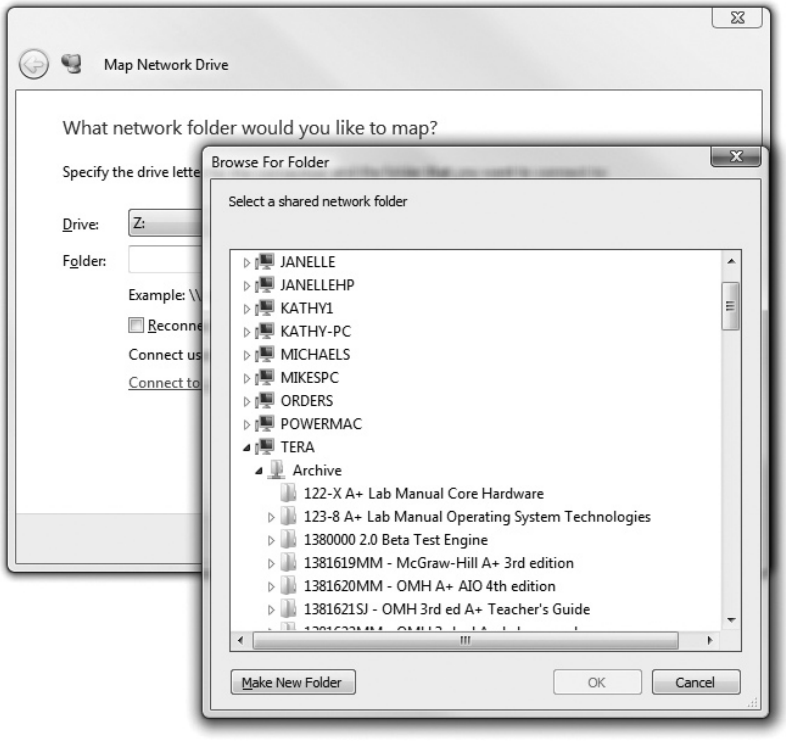


Figure 23-42 Browsing for shared folders

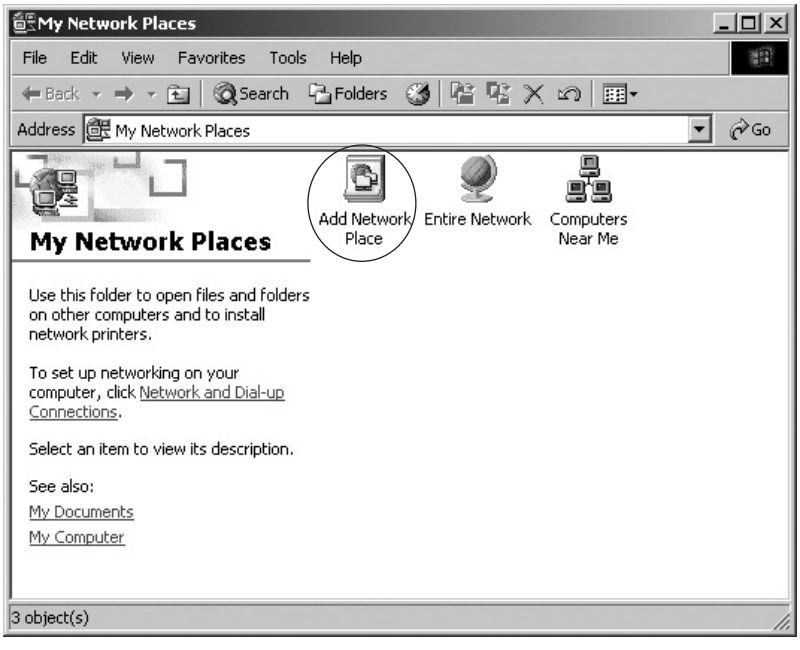


Figure 23-43 Add Network Place icon in Windows 2000

TIP All shared resources should show up in My Network Places (or Network in Vista/7). If a shared resource fails to show up, make sure you check the basics first: Is File and Printer Sharing activated? Is the device shared? Don't let silly errors fool you!

UNC

All computers that share must have a network name, and all of the resources they share must also have network names. Any resource on a network can be described by combining the names of the resource being shared and the system sharing. If a machine called SERVER1 is sharing its C: drive as FREDC, for example, the complete name would look like this:

```
\\SERVER1\FREDC
```

This is called the *universal naming convention* (*UNC*). The UNC is distinguished by its use of double backslashes in front of the sharing system's name and a single backslash in front of the shared resource's name. A UNC name can also point directly to a specific file or folder:

```
\\SERVER1\FREDC\INSTALL-FILES\SETUP.EXE
```

In this example, INSTALL-FILES is a subdirectory in the shared folder FREDC (which may or may not be called FREDC on the server), and SETUP.EXE is a specific file.

NET Command

Windows enables you to view a network quickly from the command line through the *NET command*. This works great when you plug into a network for the first time and, naturally, don't know the names of the other computers on that network. To see the many options that NET offers, type **net** at a command prompt and press ENTER. The VIEW and USE options offer excellent network tools.

You can think of NET VIEW as the command-line version of My Network Places. When run, NET VIEW returns a list of Windows computers on the network. Once you know the names of the computers, you type **NET VIEW** followed by the computer name. NET VIEW will show any shares on that machine and whether they are mapped drives.

```
C:\>NET VIEW SERVER1
Shared resources at SERVER1
Share name  Type  Used as  Comment
-------------------------------------------------------------
FREDC       Disk
Research    Disk  W:
The command completed successfully.
```

NET USE is a command-line method for mapping network shares. For example, if you wanted to map the Research share shown in the previous example to the X drive, you simply type:

```
C:\>NET USE X: \\SERVER1\Research
```

This will map drive X to the Research share on the SERVER1 computer.

Sharing Printers

Sharing printers in Windows is just as easy as sharing drives and directories. Assuming that the system has printer sharing services loaded, just go to the Printers folder in the Control Panel or Start Menu and right-click the printer you wish to share. Select Sharing; then click *Shared as* (Windows 2000) or *Share the printer* (Windows XP/Vista/7) and give it a name (see Figure 23-44).

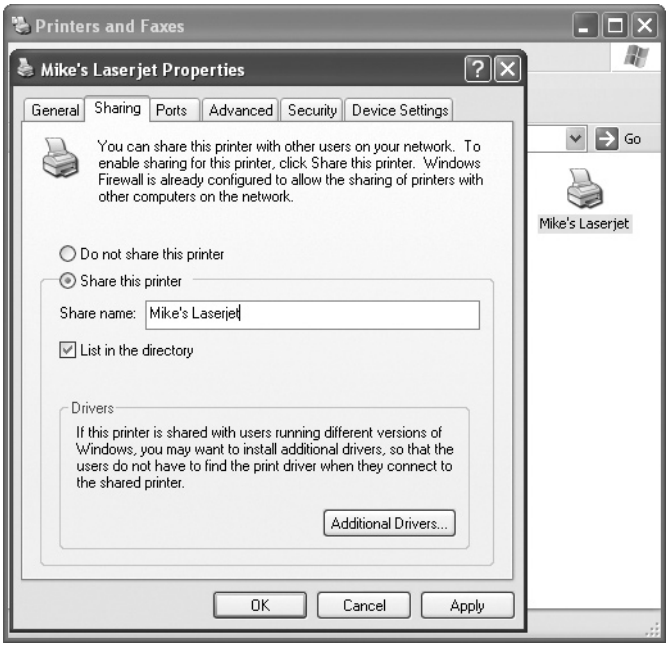


Figure 23-44 Giving a name to a shared printer on Windows XP

To access a shared printer in any version of Windows, simply click the Add Printer icon in the Printers folder. When asked if the printer is Local or Network, select Network; browse the network for the printer you wish to access, and Windows takes care of the rest! In almost all cases, Windows will copy the printer driver from the sharing machine. In the rare case where it doesn't, it will prompt you for drivers.

One of the most pleasant aspects of configuring a system for networking under all versions of Microsoft Windows is the amazing amount of the process that is automated. For example, if Windows detects a NIC in a system, it automatically installs the NIC driver, a network protocol (TCP/IP), and Client for Microsoft Networks (the NetBIOS part of the Microsoft networking software). So if you want to share a resource, everything you need is automatically installed. Note that although File and Printer Sharing is also automatically installed, you still must activate it by clicking the appropriate checkbox in the Local Area Connection Properties dialog box, as explained earlier in the chapter.

Essentials

Troubleshooting Networks

Once you go beyond a single PC and enter the realm of networked computers, your troubleshooting skills need to take a giant leap up in quality. Think of the complexity added with networks. Suddenly you have multiple PCs with multiple users who could, at the drop of a hat, do all kinds of inadvertent damage to a fully functional PC. Networked PCs have a layer of networked hardware and resource sharing that adds a completely new dimension to a user's cry for help, "I can't print!"

EXAM TIP The "Troubleshooting Networks" section covers a range of questions you're likely to see on the "Operational Procedure" exam domain. See also Chapter 27, "The Complete PC Technician," for the more on the topic.

Where can the problem lie in a *non-networked situation* if a person cannot print? Here are the obvious ones:

- Printer is not connected to the PC.
- Printer is out of ink.
- PC doesn't have the proper driver loaded.
- PC points by default to a printer other than the one that the user thinks should print.

That's about it. Maybe the parallel port configuration is wrong in CMOS or the USB drivers aren't correct, but still.… Now do the same thing with a *networked situation* where a user can't print. Here are the obvious *extra* issues, because all of the local machine issues apply as well:

- Print server is down.
- Printer is locked by another user.
- The client PC doesn't have network connectivity.
- The NIC driver is bad or incorrect.
- The client PC doesn't have the proper printer drivers installed for the networked printer.
- The cable between the client PC's NIC and the nearest switch is bad.
- The port to which the cable connects is bad.
- The switch failed.
- Somebody in an office down the hall spilled coffee on the printer, inside the mechanism, and then didn't fess up to the accident.

That's a lot of variables, and they just scratch the surface of possibilities. You live in a networked world—it's time to elevate your troubleshooting skills and methodologies to the next level. This section offers a series of steps you can use when performing any type of PC or network troubleshooting. You'll look at ways to apply your tech skills and general communication skills to get to the bottom of a problem and get that problem fixed.

Verify the Symptom

The one thing that all PC problems have in common is a symptom. If something odd wasn't happening (or not happening) to users as they tried to do whatever they need to do on their computers, you wouldn't have a problem at all, would you? Unfortunately, the vast majority of users out there aren't CompTIA A+ certified technicians. As a tech, you need to overcome a rather nasty communication gap before you can begin to consider a fix. Let's bridge that gap right now.

EXAM TIP Look for lots of questions on communication with users on the Essentials exam.

It usually starts with a phone call:

> **You:** "Tech Support, this is Mike. How can I help you?"
>
> **User:** "Uh, hi, Mike. This is Tom over in Accounting. I can't get into the network. Can you help me?"

Tom just started over in the Accounting department this week and has been a pain in the rear end so far. Ah, the things you might want to say to this person: "No. I only help non-pain-in-the-rear accountants." Or how about this? "Let me check my appointment schedule.… Ah, yes. I can check on your problem in two weeks. Monday at 4:00 P.M. okay for you?"

But, of course, you had the audacity to choose the beloved profession of IT tech support, so you don't get to ask the questions you want to ask. Rather, you need to take a position of leadership and get to the bottom of the problem, and that means understanding the symptom. Take a deep breath, smile, and get to work. You have two issues to deal with at this point. First, if you're working with a user, you must try to get the user to describe the symptom. Second, whether you're working on a system alone or you're talking to a user on the telephone, you must verify that the symptom is legitimate.

Getting a user to describe a symptom is often a challenge. Users are not techs and as a result their perception of the PC is very different than yours. But on the same token, most users know a bit about PCs and you want to take advantage of a user's skill and experience whenever you can. A personal example of verifying the symptom: Once I got a call from a user telling me that his "screen was blank." I told him to restart his system. To which he responded, "Shouldn't I shut down the PC first?" I said: "I thought you just told me the screen was blank!" He replied: "That's right. There's nothing on the screen but my desktop."

When Did It Happen?

Once you know the symptom, you need to try to inspect the problem yourself. This doesn't mean you need to go to the system; many real problems are easily fixed by the user, under your supervision. But you must understand when the problem occurs so that you can zero in on where to look for the solution. Does it happen at boot? It might be a CMOS/BIOS issue. Is it taking place as the OS loads? Then you need to start checking initialization files. Does it take place when the system runs untouched for a certain amount of time? Then maybe the power management could come into play.

What Has Changed?

Systems that run properly tend to continue to run properly. Systems that have undergone a hardware or software change have a much higher chance of not running properly than a system that has not been changed. If something has gone wrong, talk to the user to determine whether anything in particular has occurred since the system last worked properly. Has new software been installed? Did the user add some new RAM? Change the Windows Domain? Run a Windows Update? Drop the monitor on the floor? Not only do you need to consider those types of changes, but you must also make sure that any unrelated changes don't send you down the wrong path. The fact that someone installed a new floppy drive yesterday probably doesn't have anything to do with the printer that isn't working today.

Last, consider side effects of changes that don't seem to have anything to do with the problem. For example, I once had a customer whose system kept freezing up in Windows. I knew he had just added a second hard drive, but the system booted up just fine and ran normally—except it would freeze up after a few minutes. The hard drive wasn't the problem. The problem was that he unplugged the CPU fan in the process of installing it. When I discover a change has been made, I like to visualize the process of the change to consider how that change may have directly or indirectly contributed to a problem. In other words, if you run into a situation where a person added a NIC to a functioning PC that now won't boot, you need to think about what part of the installation process could be fouled up to cause a PC to stop working.

Check the Environment

I use the term *environment* in two totally different fashions in this book. The first way is the most classic definition: the heat, humidity, dirt, and other outside factors that can affect the operation of the system. The other definition is more technical and addresses the computing environment of the system and other surrounding systems: What type of system do they run? What OS? What is their network connection? What are the primary applications they use? What antivirus program do they run? Do other people use the system?

Answering these questions gives you an overview of what is affecting this system both internally and externally. A quick rundown of these issues can reveal possible problems that might not be otherwise recognized. For example, I once got a call from a user complaining she had no network connection. I first checked the NIC to ensure it

had link lights (always the first thing to check to ensure a good physical connection!) only to discover that she had no link lights—someone had decided to turn on a space heater, which destroyed the cable!

Reproducing the Problem

My official rule on problems with a PC is this: "If a problem happens only once, it is not a problem." PCs are notorious for occasionally locking up, popping errors, and displaying all types of little quirks that a quick reboot fixes, and they don't happen again. Why do these things happen? I don't know, although I'm sure if someone wanted me to guess I could come up with a clever explanation. But the majority of PCs simply don't have redundancy built in, and it's okay for them to occasionally hiccup.

A problem becomes interesting to me if it happens more than once. If it happens twice, the chances are much higher that it will happen a third time. I want to see it happen that third time—under my supervision. I will direct the user to try to reproduce the problem while I am watching to see what triggers the failure. This is a huge clue to helping you localize the real problem. Intermittent failures are the single most frustrating events that take place in a technician's life. But do remember that many seemingly intermittent problems really aren't intermittent—you have simply failed to reproduce the events exactly enough to see the consistency of the problem. Always take the time to match every step that leads to a problem to try to re-create the same error.

Isolating the Symptom

With so many bits and pieces to a PC, you must take the time to try to isolate the symptom to ensure your fix is going to the software or hardware that really needs it. In hardware, that usually means removing suspect parts until only one possible part remains. In software, that usually means removing background programs, booting into Safe mode, or trying to create a situation where only the suspected program is running.

Isolation takes on a whole new meaning with networks. One of the greatest tools in networking is isolation—does this problem happen on other systems, on other workgroups, on other PCs running DHCP? Whenever a problem takes place in networking, isolation is the key to determining the problem.

Separating Hardware from Software

Many problems that occur on a PC are difficult to isolate given that it is difficult to determine whether the problem lies in the software or the hardware. If you find yourself in this situation, you can take a few steps to help you zero in on which side of the PC to suspect.

Known Good Hardware

The absolute best way to know whether a problem is hardware or software related is to replace the suspected piece of hardware with a known good part. If you can't tell whether a Windows page fault comes from bad RAM or a software incompatibility, quickly replacing the RAM with known good RAM should help you determine whether the RAM or the software is to blame.

Cable and Loopback Test

A bad NIC can also generate a "can't see the network" problem. Use whatever utility was provided with your OS to verify that the NIC works. If you have a NIC with diagnostic software, run it—this software will check the NIC's circuitry. The NIC's female connector is a common failure point, so NICs that come with diagnostic software often include a special test called a loopback test. A loopback test sends data out of the NIC and checks to see if it comes back. Some NICs perform only an internal loopback that tests the circuitry that sends and receives, but not the actual connecting pins. A true external loopback requires a *loopback plug* inserted into the NIC's port (Figure 23-45). If a NIC is bad, replace it—preferably with an identical NIC so you don't have to reinstall drivers.

Figure 23-45
Loopback plug

The network cable is a common source of network troubles. You can use a cable tester if you suspect a cable problem. With the right equipment, diagnosing a bad cabling run is easy. Anyone with a network should own a midrange cable tester such as the Fluke Microscanner. With a little practice, you can easily determine not only whether a cable is disconnected, but also where the disconnection takes place. Sometimes patience is required, especially if the cable runs aren't labeled, but you will find the problem.

Uninstall/Reinstall

If you can do so easily, try uninstalling the suspected software and reinstalling. Many hardware/software problems magically disappear with a simple uninstall/reinstall.

Patching/Upgrading

Many hardware or software problems take place due to incompatibilities between the two suspect sides. Try upgrading drivers. Download patches or upgrades to software, especially if the hardware and the software are more than two years apart in age.

Virus Check

Last (maybe I should have put this first), always check for viruses. Today's viruses manifest so many different symptoms that failure to check for them is a study in time wasting. I recently got a new hard drive that started to make a nasty clicking noise—a sure sign of a failing hard drive. However, I ran an extensive virus check and guess what—it was a virus! Who would have thought? I checked with the hard drive maker's Web site, and my fears were confirmed. It just goes to show you—even the best of techs can be caught by the simplest problems.

Research

Once you have your mind wrapped around the problem, it's time to fix it. Unless the problem is either simple (network cable unplugged) or something you've seen before and know exactly how to fix, you'll almost certainly need to research it. The Internet makes this easy. I use one of my favorite tricks is when I get some bizarre error text: I type the error message into my search engine—that would be Google, of course—and most times find a quick fix!

Make the Fix and Test

Once you have a good idea as to the problem and how to fix it, it's time to do the fix. Always make backups—or at least warn the user of the risk to the system. If possible, try to remember how the system was configured before the fix so you can go back to square one if the fix fails to work. After you perform the fix, do whatever you need to do to make sure the system is again working properly. Make sure the user sees that the system is working properly and can sign off on your work.

OSI Seven-Layer Model

A lot of people think about networks and troubleshoot networking issues by using the OSI seven-layer model. Using this model (or my four-layer model, described in the next section of this chapter) helps you isolate problems and then implement solutions. Here are the seven layers of the OSI model:

- **Layer 1** Physical
- **Layer 2** Data Link
- **Layer 3** Network
- **Layer 4** Transport
- **Layer 5** Session
- **Layer 6** Presentation
- **Layer 7** Application

The *Physical layer* defines the physical form taken by data when it travels across a cable. Devices that work at the physical layer include NICs and hubs. Figure 23-46

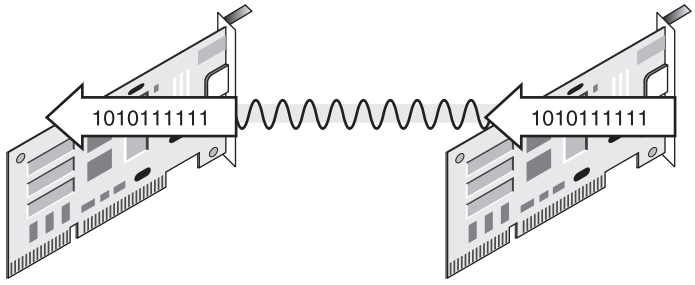


Figure 23-46 The Physical layer turns binary code into a physical signal and back into ones and zeros.

shows a sending NIC turning a string of ones and zeros into an electrical signal, and a receiving NIC turning it back into the same ones and zeros.

NOTE Basic switches reside at Layer 2 (Data Link) of the OSI model. They provide filtering based on MAC. More advanced switches that can perform InterVLAN and protocol support operate at Layer 3 (Layer 3 switch). Routers are often called Layer 3 switches.

The *Data Link layer* defines the rules for accessing and using the Physical layer. MAC addresses and Ethernet's CSMA/CD operate at the Data Link layer.

The *Network layer* defines the rules for adding information to the data packet that controls how routers move it from its source on one network to its destination on a different network. The IP protocol that handles IP addressing works on Layer 3.

The *Transport layer*, Layer 4, breaks up data it receives from the upper layers (that is, Layers 5–7) into smaller pieces for transport within the data packets created at the lower layers. In TCP/IP networks, the protocols that typically handle this transition between upper and lower layers are TCP and UDP.

The *Session layer* manages the connections between machines on the network. Protocols such as NetBIOS and sockets enable a computer to connect to a server, for example, and send and receive e-mail or download a file. Each different task you can perform on a server would require a different kind of session.

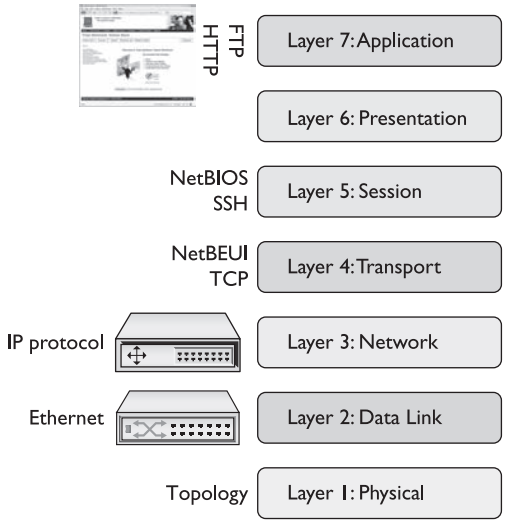
The *Presentation layer* presents data from the sending system in a form that a receiving system can understand. Most Layer 6 functions are handled by the same software that handles Layer 7 functions.

The *Application layer* is where you (or a user) get to interact with the computers. These are programs that make networking happen, such as Web browsers and e-mail applications. Chapter 25, "The Internet," covers these applications in a lot more detail.

The key to using the OSI seven-layer model is to ask the traditional troubleshooting question: What can the problem be? If Jill can't browse a Web site, for example, could this be a Layer 7 issue? Sure: If her browser software was messed up, this could stop her from browsing. It could also be a lower-level problem, though, and you need to run through the questions. Can she do anything over the network? If her NIC doesn't show flashing link lights, that could point all the way down to the Physical layer and a bad NIC, cable, or hub.

If she has good connectivity to the overall network but can't ping the Web server, that could point to a different problem altogether. Figure 23-47 shows the OSI seven-layer model graphically.

Figure 23-47
OSI



The only drawback to the OSI seven-layer model, in my view, is that it's too complex. I like to conceptualize network issues into fewer layers—four to be precise. Let's take a look.

Mike's Four-Layer Model

Network problems, by the very nature of the complexity of a network, usually make for more complex problems. Given that, I have created a four-step process that I modestly call "Mike's Four-Layer Model." These four things go through my mind every time I have a problem. I think about four distinct categories to help me isolate the symptoms and make the right fix.

Hardware

Hardware is probably the most self-explanatory of the four categories. This covers the many ways data can be moved from one PC to another. Does the system have a good connection? How's the cabling? This also covers network cards: Are they installed properly and tested? Plus, the Hardware category hits on all of those interesting boxes, such as hubs, switches, and repeaters, among which all of the wires in the network run. If you can see it, it's under this category.

Protocols

This category covers the protocols, such as TCP/IP or NetBEUI. Is the protocol installed? Is it configured properly? Does any particular system's configuration prevent it from working with another system?

Network

The network category has two parts: servers and clients. Network operating systems must differentiate systems that act as server from those that do not. If a system is a server, some process must take place to tell it to share resources. Additionally, if a system is intended to share, it must be given a name. This category also includes defining and verifying users and groups; does your system need them? Do the right accounts exist, and are they working properly?

Shared Resources

Once all of the systems, users, and groups are working properly, you need to identify the resources they will share. If a drive or folder is to be shared, the OS must provide a way to identify that drive or folder as available for sharing. The rules for naming shared resources are called *naming conventions*. A great example would be a system that offers its D:\FRED directory for sharing. This D:\FRED directory needs a network name, such as FRED_FILES. This network name is displayed to all of the devices on the network.

Sharing a resource is only half the battle. Individual systems need to be able to access the shared resources. The network needs a process whereby a PC can look out on the network and see what is available. Having found those available resources, the PC then needs to make them look and act as though they were local resources. A network also needs to control access to resources. A laser printer, for example, might be available for sharing, but only for the accounting department, excluding other departments.

Chapter Review Questions

1. To provide a computer with a physical and electronic connection to a network, what must be installed?
   - **A.** A hub
   - **B.** A router
   - **C.** A NIC
   - **D.** A bridge
2. Which of the following is needed to configure a PnP NIC in a Windows XP system?
   - **A.** CMOS
   - **B.** Configuration software
   - **C.** Device driver
   - **D.** DMA
3. How far apart can two PCs that share the same 100BaseT switch be placed?
   - **A.** 100 meters
   - **B.** 200 meters

C. 330 meters

D. 1000 meters

4. What is the minimum specification of cable types for 100BaseT networks?

A. CAT 2

B. CAT 3

C. CAT 4

D. CAT 5

5. Joe needs to network two computers in his office using an Ethernet peer-to-peer connection. What kind of cable does he need?

A. CAT-5

B. Crossover

C. UTP

D. STP

6. What are the two TIA/EIA standards for connecting an RJ-45 connector to UTP cable?

A. 10BaseT/100BaseT

B. CAT5/CAT5e

C. RG-58/RG-59

D. 568A/568B

7. Steven's Windows XP system can't connect to the Internet, and he comes to you, his PC tech, for help. You figure out that it's a DHCP problem. What program should you run to get him a new DHCP lease?

A. IPCONFIG

B. WINIPCFG

C. CONFIG

D. DHCP /RENEW

8. What command would you use to view the path taken by an Ethernet packet?

A. PING

B. IPCONFIG

C. TRACERT

D. NSLOOKUP

9. What type of network configuration has one machine configured to host data and services on the network for a number of other machines?

A. Client/Server

B. Peer-to-peer

C. Ethernet

D. Token Ring

10. Helga, the panicky intern, comes to your desk one day shouting that her Internet connection isn't working. What is the first step you should take to help solve her problem?

A. Install a new NIC in her computer.

B. Reset her computer's IP address.

C. Ask her to reboot her computer.

D. Verify the symptom of her networking problem.

Answers

1. **C.** A system must have a NIC to participate in any type of network.
2. **C.** PnP only requires the proper driver.
3. **B.** Each system can be 100 meters from the switch, so any two systems can be up to 200 meters apart.
4. **D.** 100BaseT requires CAT 5 rated cabling.
5. **B.** Joe needs a crossover cable to network two computers in his office using an Ethernet peer-to-peer connection.
6. **D.** The TIA/EIA has two standards for connecting the RJ-45 connector to the UTP cable: TIA/EIA 568A and TIA/EIA 568B.
7. **A.** You should run IPCONFIG to get a new DHCP lease for Steven's Windows XP system. WINIPCFG was the program used by Windows 9*x* for this task. /RENEW is a valid switch for both programs, but not for CONFIG.
8. **C.** The TRACERT command traces the path a data packet takes to get to its destination.
9. **A.** A server hosts data and services on a network, and a client connects to a server.
10. **D.** You should first verify the symptom of her network problem. You can't very well fix a problem if you don't know what it is.