Chapter 1

Why Network?

If you're reading this book, then you have an interest in Microsoft networking. For some people, networking sounds like a scary topic, but it really isn't. Getting a network running doesn't need to be hard, and this chapter explains many of the reasons why you want to set up a network when you have multiple machines to use. Windows Server 2008 makes networking considerably easier than ever, in fact, so you'll find that you do less work than ever before to get a network up and running.

In this chapter, we'll give you a bit of history on Server 2008 and then take a very high-altitude look at why we're using Microsoft's networking software in the first place. This is not intended to prepare you for a test on networking essentials, nor is it a complete book on Windows past and present. What I'm trying to accomplish in this chapter is to answer these questions:

- Why should you care about all of this networking stuff, anyway?
- What do you need to create a simple network?
- Why does Microsoft's networking software approach networking the way that it does?

What's the Point of Networks and Networking?

In a way, this chapter is penance for my youthful misdeeds.

When I was in the seventh grade, I had a math teacher named Mr. Schtazle. Seventh-grade math was a kind of potpourri of mathematical topics — I recall one chapter that took pains to drill into our heads the difference between precision and accuracy — and I'd plague the poor man at the beginning of every chapter by asking him, ''How will we use this?'' — a slightly more-polite version of ''Why do we care?'' Well, nowadays I find that when I'm teaching a room full of people about Windows Server, *I've* got to be careful to answer the question ''Why do you care?'' even if it isn't asked. Because if I don't answer that, then many people in the room will leave the class with a pretty good notion of *how* to accomplish a bunch of tasks but not a really good feel for *why* they'd do the tasks in the first place. And you know what? Answering the ''Why do I care?'' question can be pretty rough some times.

So, Mr. Schtazle, if you're out there. . .my apologies.

Let's consider these two questions:

- Why network in the first place?
- If we agree that networking is a good thing, why do we do it this way?

The answer to the first question will turn out to be pretty straightforward: Networking solves a set of problems for us. The answer to the question, ''Why do we do it this way?'' is a bit longer.

First and foremost, you're doing this to try to solve some problem that networking can help you with. Your company might want, for example, a great Web site, or to be able to send and receive e-mail, or a simple file and print server for a small office, or to share data with others on the Internet, or to allow employees access to your server from remote locations. These are the goals; a network is the means or tool to reach them. In short, *the ultimate goal of any networking project is to provide some kind of service*. Everything else is just a necessary evil — but there are a *lot* of those necessary evils!

Second, networks can provide many kinds of services, and every kind of service needs different software to make it work. For example, suppose you wanted to set up a Web site on the Internet. Network services, including Web sites, need two main pieces: a *server* piece and a *client* piece. To put up that great Web site, you'll create the site itself with HTML and drop that HTML onto a Web server. One way to get a Web server is by taking one of your computers and putting a piece of software on that computer to make it function as a Web server. But that's only half the story — in order for your customers to enjoy that Web server's content, they will need a piece of client software called a *Web browser*. That's our first networking piece: *Every network service needs server software and client software*.

Third, you need to ensure that there's a way for your information to get from your server to your clients, a physical system that the service can travel over. If the clients and servers are in the same building, then you need only a local area network (LAN), and setting that up merely requires pulling wires through the building (plus a few pieces of additional hardware described in the ''Networks Need Connection Hardware and Links'' section of the chapter). If, however, you want to offer your service to the world, as in the case of a Web server, then you'll need some kind of wide area network (WAN) connection to the Internet. Most companies today rely on a virtual private network (VPN) to ferry data safely across the Internet. In other cases, you'll need a WAN connection, but not to the Internet: Many organizations with more than one location connect those locations via private communications links with names like *leased line*, *T1*, or *frame relay*. That's our next networking piece: *Networks need connection hardware (switches, hubs, routers, modems) and links (phone lines, network cables, frame relay, DSL, cable modem, ISDN, and so on), or the clients can't connect to the servers*.

Fourth, to provide a service over a network, your server and your clients must agree on how to transmit information over that network. That agreement is called a *network protocol*, and the one that you'll most probably use in the Windows 2008 world is called the Transmission Control Protocol/Internet Protocol (TCP/IP). You may have heard of it before, as it's the network protocol that the Internet uses, but you needn't be on the Internet to use it. In short, *clients and servers must agree to speak using the same network protocols*.

NOTE

Windows Server 2008 provides two different versions of IP: IP version 4 (IPv4) and IP version 6 (IPv6). IPv4 is the version of IP used by the Internet today. In most cases, it's the only version of IP you need to support today. IPv6 provides additional addresses, some extra security, and a few other features. This version of IP provides functionality you need tomorrow, but you don't need to worry about it today. Because Microsoft chose to install IPv6 by default in Windows Server 2008, you may want to disable this support in order to gain some additional system performance. Chapter 2 tells you how to create an optimal setup that uses IPv4 efficiently (the companion enterprise volume, *Mastering Windows Server 2008: Enterprise Technologies*, discusses IPv6 in detail).

Fifth, once you have the channels open and before information starts flowing in both directions, you'll almost certainly need to worry about security. When you use the tool that is networking, you want to be sure it doesn't increase your risk, and in fact you can shape the tool so it reduces hazards. Briefly: *Networks need security*. (Chapter 2 introduces you to the topic of security.)

Sixth, and finally, once you've set up that terrific network service, you need a way for people to *find* that great service. You do that with a ''naming'' system. Windows 2008 has two of them — one that appeared years ago before the first version of NT (NT was the earliest version of Windows Server that Microsoft initially offered in 1993) and a newer (than NT, anyway) method that the Internet has been using for years. The last network piece, then, is that *networks must provide a way for users to find their services*.

Let's examine these pieces in order, take a closer look at why they work the way that they do, and get some insight into how Windows 2008 in particular handles them. This chapter only begins the discussion of networks. Some topics are so important that we decided to discuss them in detail in Chapters 2 and 3. You won't actually install Windows Server 2008 until Chapter 4 — these initial chapters will help you understand and prepare for your network.

Choosing a Network Type

Windows networks fall into two categories: workgroup and domain. A *workgroup* network connects multiple computers in a peer-to-peer configuration, which means that every computer can serve as both a client and a server. Workgroups are very simple, and you normally use them for smaller groups of computers — usually less than 10, but I've seen much larger workgroups of up to 100 computers. A workgroup doesn't require a centralized server, but you can certainly use one. Workgroups typically require little time to set up and configure, but they can become a nightmare to manage when they exceed a certain size. The fact that you don't necessarily need to have a server also means that workgroups can be less expensive.

A *domain* provides fully centralized services. It always requires that you set up a server and the server must provide support for advanced management features such as Active Directory. Domains provide stronger security than workgroups do because everything is under the tight control of the server. In addition, domains provide centralized administration. Normally, you use domains for larger networks. They require a lot more time and effort to set up, configure, and administer for a small number of computers, but a domain also offers significant advantages over a workgroup. As the size of your network groups increases, administration costs go down and performance increases when compared to a workgroup.

Network Client and Server Software

The reason that we network computers in the first place is so that computers acting as clients can benefit from the services of computers acting as servers. For example, suppose you want to visit my Web site, `www.minasi.com`. Two of the ingredients that you'll need to make that possible are software applications:

- You'll need a computer running a program that knows how to request Web information and then how to receive it — in other words, a *client application*.
- I'll need a computer running a program that knows how to listen for requests for Web information and then how to deliver that information — in other words, a *server application*.

As sometimes occurs *too* often in the computer business, you've got choices about both the client and the server.

The Client Piece: A Web Browser

I've said that first you'll need a computer, of course, one that's running a Web browser program such as Firefox or Internet Explorer. But let me rephrase that in basic network client-server terms.

There is technically no such thing as ''the World Wide Web.'' Instead, there is an agreement about how to transfer text, pictures, and the like, and that agreement is called the HyperText Transfer Protocol — which is normally shortened to HTTP. The phrase World Wide Web just refers collectively to all of the HTTP servers on the Internet. When you think you're surfing a Web page, what really happens is this:

1. Your client computer asks the Web server (oops, I meant *the HTTP server*) something like, ''Do you have any documents?''
2. The Web server responds by saying, ''Here's my default document,'' a simple text file that is the so-called home page for that Web server. The Web server sends that file to your client using the HTTP protocol.
3. Once your client receives the text file, it notices that the page is full of references to *other* files. For example, if the home page that you requested has pictures on it, your Web browser (HTTP client) didn't originally know to ask for them, so the Web server (HTTP server) didn't send them. Your client notices the lack of the images and requests that the server send them, which it does — again using HTTP.

Here, ''HTTP client'' just means a program that knows how to speak a language that transfers a particular kind of data — Web data. Your computer is deaf to the Web unless it knows how to request and receive data via HTTP.

Notice what *client* means here. It doesn't refer to you, or even to your computer. Instead, it just means a program that your computer runs.

The Server Piece: A Web Server

Next, let's consider what's sitting on my side of the conversation.

I'll need a computer running a special piece of software that is designed to listen for your computer (or anyone else's, for that matter) requesting to see my Web pages via HTTP and that can respond to those requests by transferring those pages to the requesting client software. You *might* call such a piece of software an ''HTTP server'' program, although almost no one calls it by that name. You'd more *commonly* call it ''Web server'' software. There is a variety of Web server software that I might run on my Windows Server 2008 computer, but I'm most likely to run the one that comes free with Server 2008, a program called Internet Information Services (IIS) 7. Alternatively, I might find, download (probably using HTTP!), and install a popular piece of free Web server software called Apache.

Once again, notice carefully what ''server'' means here. It does not really refer to the particular computer hardware that I've got stashed in my network room connected to the Internet. Instead, *server* means ''the program running on Mark's computer that listens for HTTP requests and knows how to fulfill them.''

Now that I've gone through all of that, consider again the question that I asked at the beginning of the chapter — why are you bothering with a network? The answer is probably because you want to offer a Web site, either internally or on the public Internet, and you think that IIS is the best (highest-performance, cheapest, or some combination of the two) Web server software around — which means that you must use Server 2008, because it's the only operating system that supports IIS 7. (Or you could use an earlier version of Server and an earlier version of IIS, but why not go with the latest and greatest?)

Other Types of Servers

I'll tend to use the Web client-server example for this discussion. But I don't want to lose sight of the fact that there are quite a few client-server systems, besides Web servers, that are in common use and that you may want to use 2008 to create. Returning to the theme of this chapter, then — ''Why do I care or why do I need this stuff?'' — networks offer several valuable services, and you may want to set up a computer to act as a server and offer some of those services. Here are a few besides the Web server example:

File Servers File servers act as central places to store data files. Why put them on a server rather than just keep them on your local computer? Well, in some cases someone else created the file, and placing a file on a central server is a simple way to make the files available to others. The other good thing about storing files in a central location is that they're more easily backed up that way. Server 2008 comes with file server software built in.

Print Servers Print servers let you share printers. Not everyone wants to put a printer on their desk, and besides, if you share the printers, you can afford more expensive (and presumably better) models. Server 2008 comes with print server software built in.

Application Servers Application servers provide a method for sharing an application across the Internet. In addition, you can distribute pieces of the application so that you can use multiple servers to provide a complete solution. Windows Server 2008 provides the software required to create an application server and manage the applications it hosts from a central location. This is a new feature for Windows Server 2008.

E-mail Servers Mail servers are essential if you're going to do e-mail. Some computer (or computers) must act as the post office, collecting e-mail from the local users and sending it to other mail servers across the Internet and acting as a receiving point for other mail servers to send mail destined for your organization. You *can* outsource this function by letting your ISP act as your mail server, but running your own mail server gives you more flexibility. (However, it *does* require a persistent connection to the Internet.) 2008's new features include a basic e-mail server. Yes, it's ''basic'' because Microsoft *really* wants to sell you Exchange as your mail server. But it's not a bad server for many people's needs.

Terminal Servers A terminal server harks back to the days of mainframes (think about the huge computers you may have seen in older movies — a mainframe is a single large computer used to serve a number of people). Someone using a terminal would log into the mainframe from a remote location to access the features that the mainframe provides. Modern users rely on this feature to access the server from a remote location using less capable devices. Some companies use this service to save money. Administrators rely on this feature to manage the server. Using a terminal server application called Remote Desktop means you don't have to walk to the server to perform administration tasks.

Group Scheduling Servers The centralized nature of servers means that they're a great place to keep track of scarce resources like meeting rooms or your time. Server 2008 does not come with a scheduling server, because Microsoft wants to sell you Exchange to do that sort of thing. But there are alternatives to Exchange; there are some terrific Web-based scheduling tools that work great on 2008 — for one example, take a look at `www.mattkruse.com/scripts/calendar/` or other tools, such as Lotus Notes.

SharePoint Servers A SharePoint server lets users collaborate with other users, even when they aren't physically located in the same place. Users from England, the United States, and Japan could work on documents together as if they were all located in the same place.

As with a local connection, users can also share information, such as contacts, with each other. An administrator can also use a SharePoint server to place (deploy) applications on remote systems without physically visiting those locations.

E-Commerce Online Stores If you've got something great to sell, then the Web's one place to do it. There are thousands of online stores on the Web, and a good number of them run on 2008. While 2008 includes a Web server, it doesn't include the other software that you'd need to create a complete online store. But there are a lot of consulting and programming firms that would be happy to help you create an online store atop 2008!

Microsoft has adopted new terminology for Windows Server 2008 that makes it easier to understand the difference between a service that the server provides and a piece of software that makes the server perform better or provide improved capabilities. *Roles* are the pieces of software that define the services that a server provides. For example, if you want to make your server into a Web server, you install the Web Server (IIS) role. *Features* help your server perform certain tasks better. In some cases, you must install a feature to make a role work, but most features simply add functionality. For example, if you want to use your Web server to help remote users print documents, then you install the Internet Printing Client feature.

Networks Need Connection Hardware and Links

If I want to offer a server service and ensure that you can enjoy that service, then we'll both need to be physically attached to the same network — the same series of cables, satellite links, or whatever — or your computer's requests will never get to my computer in the first place. That probably means that we're both on that huge network-of-networks called the Internet, but we could just be working for the same company in a single wired building, or a multilocation firm connected by a private intranet.

Now, notice that if I'm going to run a Web server, I'll need to be connected to our common network (Internet or otherwise) persistently: I couldn't decide to run a Web server out of my house and just dial in to the Internet now and then. Of course, if I'm only serving some private network that we share, then an Internet connection is unnecessary, because we already have a connection to a common network.

People who worry about the physical connection part of networking concern themselves with getting cables run through walls, calling the phone company to arrange for persistently connected data links of various kinds (links with names like *digital subscriber line*, *cable modem*, *frame relay*, *leased lines*, *T1* or *T3 lines*) and then working with a family of hardware that helps get the bits going off in the right direction (devices with names like *switches*, *hubs*, and *routers*).

Does 2008 help you with this part of the job? In some parts, it can. Switches and hubs are very basic, simple devices, and 2008 has nothing to do with them — although clearly 2008 depends on their presence in order to network! Routers are, however, more complex devices. You probably know that the market leader in the router world is a firm named Cisco Systems, but you might not know that a router is really just a small, single-purpose computer. If you wanted, you could use a computer running Server 2008 to replace a Cisco router. Additionally, if you wanted to allow people outside your network to dial in to your network, you could use a Windows Server 2008 to make that possible.

Considering the Hardware

This chapter has already discussed a lot of hardware. Although the hardware part of the picture isn't hard to understand, you do need to understand it in order to create your network. Networks have some basic hardware that you must have in order to ensure that everyone can communicate.

In some cases, you install optional hardware to make the network perform certain tasks or to add to the functionality that the network provides.

When PC networks first came into existence, you could find a wide range of distinctly incompatible components. Network hardware could use all kinds of odd-sounding technologies such as Token Ring and ArcNet. (Don't worry if these technology names are unfamiliar to you, we'll discuss them in more detail as the book progresses. For now, all you need to know is that they provide a kind of physical connection between computers.) Today, most networks rely on Ethernet connections and use standard components. You might have heard that networks are hard to put together, partly because they really were in the past, but luckily standardization has made creating a network significantly easier. Here are the common pieces of hardware you find on a network:

Connector For many people, the lowly connector isn't even worth mentioning, but you suddenly discover the importance of this element when your network is no longer connected and nothing is apparently wrong. The typical Ethernet connector looks just like a larger version of the connector for your telephone, as shown in Figure 1.1. In fact, that's one of the first things you need to avoid — mistaking the two types of connector. Notice that the RJ45 connector is larger and that it has 8 pins in it, rather than the 4 or 6 pins of an RJ10, RJ11, or RJ12 connector used for a telephone. When you plug a connector into a NIC, hub, switch, or router, the receptacle normally lights up to show you have a good connection. You should look for these lights when you need to find a loose connection. It's a bad idea to plug and unplug connectors too often because the connection can become loose and cause you a lot of trouble.

FIGURE 1.1 Typical Ethernet connectors. The male connector appears on each end of the cable, while the female connector appears with the computer, hub, switch, router, or other device connection.

Network Interface Card (NIC) A network interface card (NIC) connects the computer to the network. It provides all of the hardware features required to make an electrical connection and perform low-level networking tasks. A NIC won't provide the connection by itself. Windows provides software required to make the NIC functional. Most machines today have one or two NICs supplied with them. You must have one NIC for each connection you want to create. A machine with two NICs can use one of them to connect to a local network and the other to connect to the Internet. NICs have specific characteristics — some of which are important for everyone to know and some of which are only helpful to technicians. The most important NIC characteristic is its connection speed because the connection speed determines how fast the NIC can communicate with other machines connected to the network.

Cable A cable provides a physical connection between the NIC contained within the machine you want to connect to the network and the hub, switch, or router used to distribute signals to the rest of the network. Cables come in a confusing array of sizes and types. The most important characteristic of the cable is the connection speed it supports. You must match the connection speed of the cable to the NIC. Otherwise, the NIC won't be able to connect at full speed. In some special cases, you need cables with other characteristics. For example, if you want to

run the cable through a false ceiling or through air ducts, you may need special cable designed for that purpose (often called *plenum cable*). Plenum cable resists burning and doesn't produce as many noxious chemicals if it does burn, but it costs a lot more than standard cable. Check the local electrical code to ensure you use the right kind of cable for a specific purpose.

Hub A hub is the least expensive connectivity solution for a network. You connect one end of the cable into the NIC and the other end into the hub. *Voilà*, you are now part of the network. Every computer or other device (such as a printer) that wants to be part of the network has the same connection setup. Hubs can usually have 2, 4, 8, 16, or 32 computers or other devices connected to them, with 8 being the most common. Each connection to a hub is a port. You should buy a hub with enough ports to support all the devices on your network, with a few to spare. When you run out of ports, you can purchase another hub, connect the two hubs together using a special port, and then plug additional computers into the new hub. Connecting multiple hubs together is *daisy chaining*. Because hubs are very simple devices, they are also extremely reliable. However, the reliability and cost savings comes at the price of performance and ease of maintenance. If you need a high-speed connection or you have many devices to connect, then a switch is a better option than using a hub. The most important characteristic of a hub is the connection speed it supports. The connection speed must match the speed of the NICs on a network.

Switch Switches work precisely the same as hubs from the outside. You connect one end of the cable into the NIC and the other end into the switch to create a connection to the network. However, switches include additional internal circuitry and provide performance benefits. A switch can make smart connections between two devices on the network to speed communication between them. When working with a hub, all of the computers on the network hear the message that another computer sends, but switches direct the message specifically to the computer that needs to hear it. In addition, switches normally contain diagnostic hardware to make it easier to find problems on your network. Of course, you don't get this extra circuitry free — switches cost more than hubs do. As with hubs, the switch connection speed must match the connection speed of NICs on the network.

Router A router is similar to a switch or hub, but it includes something extra — the ability to connect to the outside world. A router is similar to a computer with two NICs in it. One set of connections is for the local network, while the second set of connections is for the outside world. The vendor labels the connections so you can't make a mistake in creating the required connections. Routers also include some of the features of a server, including a firewall for security purposes. The features you obtain with the router depends on the kind of router you buy. For example, some routers include special support for standard TCP/IP features, and some even include a wireless access point (WAP) to connect with wireless devices. Make sure you get a router that includes all the features you need. For example, if you have wireless devices, then getting a router with a built-in WAP is a better buy than purchasing the WAP separately.

Network-Ready Device In days gone by, most devices such as printers, camcorders, and cameras were dependent on a connection with a computer to gain network connectivity. You plugged the device into the computer and shared the device with others on the network, and then other people would access the device through this shared connection. This approach to networking wastes resources because you now need a computer to create the required connection. Network-ready devices have a NIC built into them and provide the software required to create the network connection. You access a network-ready device the same way you do any computer on the network. The network-ready devices do cost more than standard devices, so you need to weigh the added cost of the device against the cost of using the computer to

provide a connection. In some cases, such as a small network that has a server and only a few devices, using standard devices may actually prove less expensive in the long run. Most routers support only two networks. However, it's possible to find routers that support more — all the way up to 128 networks — but you'll pay a hefty price for them.

Wireless Access Point A wireless access point (WAP) is a type of router. It creates a connection between a wired network and wireless devices. The wireless devices use radio waves to communicate with the WAP. As with most network devices, you plug the WAP into a hub, switch, or router to create the connection to the wired network. The critical consideration for a WAP is the standards to which the WAP adheres. For example, if your laptop provides an 802.11g connection, then your WAP must support the 802.11g specification or the two devices won't communicate with each other. The 802.11b and 802.11g specifications are the most common in use today, but you must check your wireless devices to determine which standard they follow. In some cases, the WAP will support multiple standards. You'll want all your wireless devices to use the same standard because some wireless connection standards don't work well with others because of radio wave interference. Another consideration is the antenna range for the WAP. This range determines the maximum distance that can separate the wireless device and the WAP. Always remember that the distance between the wireless device and the WAP affects transmission speed. You may think that you'll get 54 Mbps transmission speed, but you won't when you're at the maximum distance. In fact, most WAPs provide multiple fallback speeds so you need to know the slowest speed you can expect before losing the connection completely.

It often helps to view network diagrams put together by other people before you create your own network. For example, you can see a basic home network with a WAP at `http://www.homenethelp.com/web/diagram/wireless-bridge-xp.asp`. The diagrams at `http://msdn2.microsoft.com/en-us/library/aa934598.aspx` show you how you might attach a network-ready device. The diagram at `http://www.weresc.com/home.php` is significantly more complex than the other two, but it shows you that networks can be any size. If you want to create your own network diagram using the hardware discussed in this chapter, check out the free program at `http://www.supershareware.com/info/edraw-network-diagrammer.html`.

After you look at a number of designs put together by other people, you'll want to spend some time creating your own diagram. The diagram need not be very complicated. However, you need to provide enough information to ensure you can create a good network setup. Figure 1.2 shows a network diagram that includes many of the elements that you'll probably have on your network. This diagram doesn't represent your network any more than the diagrams I referenced earlier through Web sites — it's just another example that you can use to create your own diagram.

This diagram shows a number of important features. Every computer and network-ready device can have a NIC that is separate from that device. Yes, the NIC appears inside the unit, but it may not come with the unit — you may have to purchase this item separately and ask the vendor to install it for you. The connector (shown in Figure 1.1) will appear somewhere on the case. You connect a cable from the device to the hub or switch as shown. In some cases, you might actually connect the computers and network-ready devices directly to the router, instead of using a separate connection as shown. Notice that the router provides a connection to the Internet — the hub or switch won't provide this connection.

Do you see that lightning bolt next to the WAP? That lightning bolt represents a connection made using radio waves — a wireless connection. Any device, such as a laptop, designed to use the wireless standard supported by the WAP can connect to the network through the WAP as shown. You shouldn't add a WAP to your network unless you actually need it because a WAP can

cause security breaches that you wouldn't experience when using wired connections. As shown in Figure 1.2, the WAP provides a bridge between the wired and wireless connections on your network.

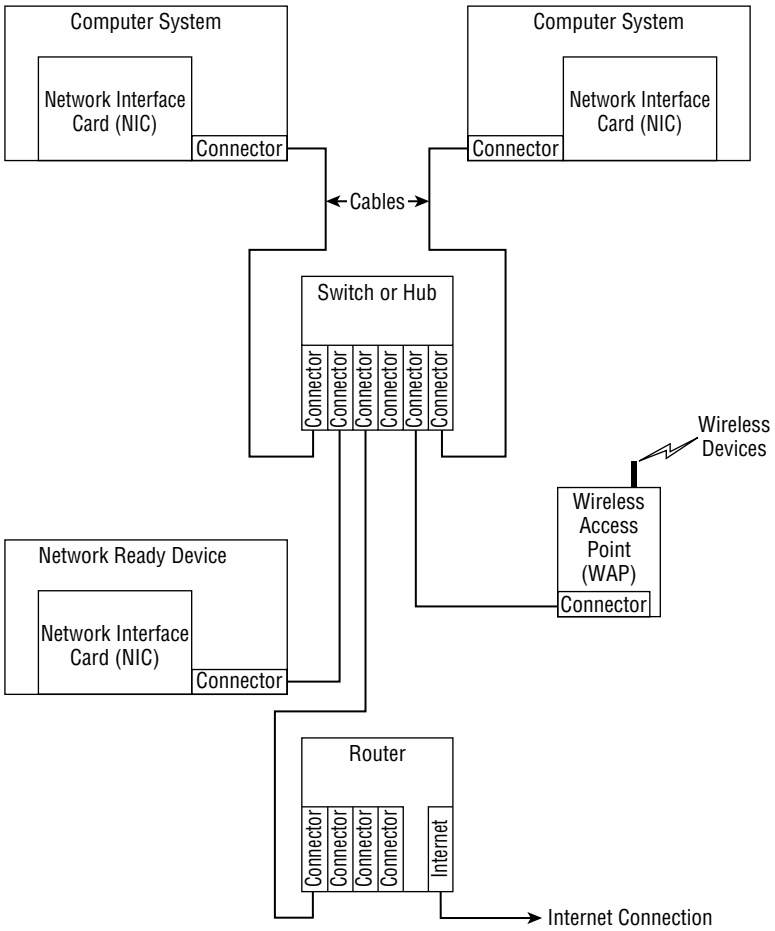

FIGURE 1.2
Creating a diagram of your network is important if you want to get good results.

Clients and Servers Must Speak the Same Protocols

But simply being connected to the same wire isn't enough — we need a common communications language. If I were to pick up a phone and dial some number in Beijing, I'd have a physical connection with whatever poor soul picked the phone on the other end — but that would be the extent of our interaction. In the same way, computer networks need to agree on things like, ''What's the biggest block of data that I can ever send you?'' and ''How shall I acknowledge that I actually *got* that block of data?'' or ''Should I bother acknowledging receipt of data at all?'' and hundreds of other questions.

The answers to all those questions are contained in the ''network language,'' or, in network techie terms, the *network transport protocol*. It probably won't surprise you that more than one

network transport protocol exists, and over the years NT and other versions of Windows Server have generally supported three of them:

- NetBEUI (Network Basic Input/Output System Extended User Interface), an old Microsoft/IBM/Sytek protocol designed to support small networks
- IPX/SPX (Internet Packet Exchange/Sequenced Packet Exchange), the protocol that Novell NetWare predominantly used for years
- TCP/IPv4 (Transmission Control Protocol/Internet Protocol), the protocol of the Internet and intranets

Windows Server 2008 changes this equation somewhat. You won't find support for IPX/SPX in Windows Server 2008, which means it won't communicate with that old NetWare server on your network. In addition, you won't find NetBEUI support in Windows Server 2008 because Microsoft has replaced this protocol with TCP/IP. However, Windows Server does add support for TCP/IPv6, which provides additional address space (which means it supports additional devices) and better security. The article at `http://technet.microsoft.com/en-us/library/bb878121.aspx` provides great information on the new features provided by TCP/IPv6.

Your only choices for transport protocols in Windows Server 2008 are TCP/IPv4 and TCP/IPv6. It's a good bet that you're using TCP/IPv4 right now. Why TCP/IPv4? Well, there have been some really great protocols over the years, but because the Internet uses TCP/IP and the Internet is so popular, TCP/IP has sort of trumped the other protocols. In fact, it's impossible to do a fair number of things that 2008 and its predecessors Windows 2000 and, to a lesser extent, Windows NT 4 are capable of doing *without* TCP/IP. Because TCP/IPv6 is so new, few Internet service providers (ISPs) require it and it's doubtful you need this protocol for your company. So, I'm going to assume for our discussion and indeed for most of this book that your network will use TCP/IPv4.

Oh, and one more thing — once you've decided that TCP/IP is your network protocol of choice, then you'll need to install several *more* servers to support TCP/IP's infrastructure. And here again, when I say ''more servers,'' I'm not suggesting that you have to buy more PCs, although you might. What I mean is that you'll have to install software on some computer or group of computers to perform three basic pieces of plumbing or infrastructure jobs:

- A Domain Naming System (DNS) server keeps track of the names of the computers in your network (an important task, believe it or not). When working with a workgroup, you can obtain DNS support automatically (without any configuration) by using the Internet Connection Sharing (ICS) feature of Windows Server 2008. You must install DNS support separately for a domain.
- A Dynamic Host Configuration Protocol (DHCP) server configures the specifics of TCP/IP on each computer in your network, both great and small. Many routers provide DHCP support, so make sure you check your router before you configure this feature on your Windows Server 2008 installation because it may be a redundant service/role that your Windows Server can do without.
- A Windows Internet Name Server (WINS) does something like what DNS does — keeps track of names — but isn't really necessary on a ''pure'' Windows 2008 network — its main job is to support older Microsoft operating systems like Windows 9*x*, Me, and NT 3.*x* and 4.

You'll learn more about the specifics of DNS, DHCP, and WINS in Chapter 11. I should point out that if you're a one-person shop, then you might not need all of that, as your ISP might be

handling it for you — but I'm assuming throughout this book that you are probably a network administrator/manager for a network of at *least* a few computers, and possibly for a tremendous number of computers.

A Brief History of Windows

Let's finish this chapter with a look at how NT has grown into Windows Server 2008 today.

Even in the early 1980s, Bill Gates knew that networking was a key to owning the computer business. So, on April 15, 1985, Microsoft released its first networking product, a tool called MS-NET, and its companion operating system, DOS 3.10. Most people knew about the new DOS and were puzzled at its apparent lack of new features. What it contained, however, were architectural changes to DOS that made it a bit friendlier to the idea of networks.

Now, Microsoft wasn't big enough at that time to create much hoopla about a new network operating system, so it let others sell it — no matter how high or low you looked, you couldn't buy a product called MS-NET. Instead, it sold mainly as an IBM product under the name of the IBM PC Network Support Program; IBM viewed it as little more than some software to go along with IBM's PC Network LAN boards and, later, its Token Ring cards. The server software was DOS-based, offered minimal security, and, to be honest, performed terribly. (Believe me, I *know*; I used to install them for people.) But the software had two main effects on the market.

First, the fact that IBM sold a LAN product legitimized the whole industry. IBM made it possible for others to make a living selling network products. And that led to the second effect: the growth of Novell. Once IBM legitimized the idea of a LAN, most companies responded by going out and getting the LAN operating system that offered the best bang for the buck. That was an easy decision: NetWare. In the early days of networking, Novell established itself as the performance leader. You could effectively serve about twice as many workstations with Novell NetWare as you could with any of the MS-NET products. So Novell prospered.

As time went on, however, Microsoft got better at building network products. 3Com, wanting to offer a product that was compatible with the IBM PC Network software, licensed MS-NET and resold it as their 3+ software. 3Com knew quite a bit about networking, however, and recognized the limitations of MS-NET. So, 3Com reworked MS-NET to improve its performance, a fact that didn't escape Microsoft's attention.

From 1985 to 1988, Microsoft worked on its second generation of networking software. The software was based on its OS/2 version 1 operating system. (Remember, Microsoft was the main driving force behind OS/2 from 1985 through early 1990. Steve Ballmer, Microsoft's number-two guy, promised publicly in 1988 that Microsoft would ''go the distance with OS/2.'' Hey, the world changes, and you've got to change with it, right?) Seeing the good work that 3Com did with MS-NET, Microsoft worked as a partner with 3Com to build the next generation of LAN software. Called Microsoft LAN Manager, this network server software was built atop the more powerful OS/2 operating system. As with the earlier MS-NET, Microsoft's intention was never to directly market LAN Manager. Instead, Microsoft envisioned IBM, 3Com, Compaq, and others selling it.

IBM did indeed sell LAN Manager (it still does in the guise of OS/2 LAN Server). 3Com sold LAN Manager for years as 3+Open but found little profit in it and got out of the software business. In late 1990, Compaq announced that it would not sell LAN Manager because it was too complex a product for dealers to explain, sell, and support. Microsoft decided then that if LAN Manager were to be sold, it would have to do the selling, so on the very same day as the Compaq withdrawal, Microsoft announced it would begin selling LAN Manager directly.

NOTE

Here's an interesting side note: Ten years after Compaq (now HP) decided that its sales force couldn't sell network software, it reversed direction and said it would sell a special version of Windows 2000 called Datacenter Server. It's special because you cannot buy it from Microsoft — you must buy it preinstalled on specially certified vendor hardware. In other words, the hardware vendors (HP is not the only one selling Datacenter) now believe that they can sell complex network operating systems. I wish them the best of luck, but stay tuned to see the outcome of this particular marketing maneuver!

LAN Manager in its first incarnation still wasn't half the product that Novell NetWare was, but it was getting there. LAN Manager 2 greatly closed the gap, and in fact, on some benchmarks LAN Manager outpaced Novell NetWare. Additionally, LAN Manager included administrative and security features that brought it even closer to Novell NetWare in the minds of many network managers. Slowly, LAN Manager gained about a 20 percent share of the network market.

When Microsoft designed LAN Manager, however, it designed it for the 286 chip (more accurately, I should say again that LAN Manager was built atop OS/2 1.*x*, and OS/2 1.*x* was built for the 286 chip). LAN Manager's 286 foundation hampered its performance and sales. In contrast, Novell designed their premier products (NetWare 3 and 4) to use the full capabilities of the 386 and later processors. Microsoft's breakup with IBM delayed the release of a 386-based product, and in a sense, Microsoft never released the 386-based product.

Instead of continuing to climb the ladder of Intel processor capabilities, Microsoft decided to build a processor-independent operating system that would sit in roughly the same market position as Unix. It could then be implemented for the 386 and later chips, and it also could run well on other processors, such as the PowerPC, Alpha, and MIPS chips. Microsoft called this new operating system NT, for ''new technology.'' Not only would NT serve as a workstation operating system, but it would also arrive in a network server version to be called LAN Manager NT. No products ever shipped with that name, but the wallpaper that NT Server displays when no one is logged in is called `LANMANNT.BMP` to this day.

In August 1993, Microsoft released LAN Manager NT with the name NT Advanced Server. In a shameless marketing move, it was labeled version 3.1 in order to match the version numbers of the Windows desktop products. This first version of NT Advanced Server performed quite well. However, it was memory-hungry, lacked Novell connectivity, and had only the most basic TCP/IP connectivity.

September 1994 brought a new version and a new name: Microsoft Windows NT Server version 3.5. Version 3.5 was mainly a ''polish'' of 3.1; it was less memory-hungry, it included Novell and TCP/IP connectivity right in the box, and it included Windows for Workgroups versions of the administrative tools so network administrators could work from a Workgroup machine rather than an NT machine. Where many vendors would spend 13 months adding silly bells and whistles, NT 3.5 showed that the Microsoft folks had spent most of their time fine-tuning the operating system, trimming its memory requirements, and speeding it up.

In October 1995 came NT version 3.51, which mainly brought support for PCMCIA cards (a real boon for us traveling instructor types), file compression, and a raft of bug fixes.

NT version 4, 1996's edition of NT, got a newer Windows 95–like face and a bunch of new features, but no really radical networking changes. Under the hood, NT 4 wasn't much different from NT 3.51.

From mid 1996 to early 2000, no new versions of NT appeared, an ''upgrade drought'' such as we'd not seen in quite some time from Microsoft. Then, in February 2000, Windows 2000 (''NT 5.0'') shipped. Windows 2000 included a whole lot of new stuff, but perhaps the most significant was a new way of storing and organizing user accounts and related information: Active Directory (AD) domains. Closely following AD in importance was the then-new notion of Group Policy, something you'll see has become quite important to anyone wanting to run a network based on XP and Server 2003.

The next version of NT shipped in pieces for the first time since 1993. First NT Workstation 5.1 or, as it's better known, XP Professional and its lesser sibling, XP Home. Microsoft intended to follow up with the server version of NT 5.1, but events conspired to compel them to wait a bit longer and produce NT Server 5.2 — that is, Windows Server 2003. Windows Server 2003 is a ''1.1'' version of Windows 2000, a welcome improvement to 2000's fit and finish.

And now we reach Windows Server 2008, which builds a wealth of functionality onto the previous offerings. Of course, it now supports TCP/IPv6, which is an addition for the future. The best news is that Windows Server 2008 provides significant new security features. For example, even the administrator doesn't have access to the root directory, `\Windows` folder, or `\Windows\System32` folder, so trying to corrupt executables within these folders is significantly more difficult. The new Windows firewall provides both incoming and outgoing firewall support, so outsiders will find it much more difficult to gain entry to your server, especially if you have other firewalls in place. The administrator also runs as an average user now and must give permission to perform certain tasks. This feature makes it much harder for an outsider to do something without the administrator's knowledge. In short, Windows Server 2008 is a welcome improvement to the one issue that people complain about most — Windows security.

For those of you who fought through Windows network configuration tasks in the past, you'll find that Windows Server 2008 greatly automates the task. Microsoft has added functionality that automatically detects your network card and begins the setup process for you as part of the installation. In some cases, you might not need to do anything with the NIC or associated connections at all except verify that your configuration is correct. You don't need to worry about these details now. The next chapter shows how to put your network together, Chapter 3 reviews Windows security, and you'll see how to install Windows Server 2008 in Chapter 4.

Well, I hope this chapter wasn't boring for those already expert in Windows while bringing the newbies up to speed. No matter what version of Windows you're running, however, you'll need to configure it. For example, Microsoft can't guess about which resources, such as hard drives, that you want to share, so the new automation can't do everything for you. And there are, as there always have been, two main ways to do it. The preferred way is through the GUI with windowed programs that offer help and a bit of error-checking, or its somewhat more complex relatives, the command-line tools. The less-preferred, but often necessary, way is to directly tweak some setting in its lair . . . a place called the Registry. The chapters that follow introduce these two configuration approaches.