What You Will Learn

After reading this chapter, you should be able to answer the following questions:

- ✔ What is TCP/IP?

- ✔ How are TCP/IP and the Internet related?

- ✔ What are five critical communications functions that TCP/IP can perform?

- ✔ How do you know when you are using TCP/IP?

- ✔ What can't TCP/IP do? (In other words, what communications functions must it rely on other mechanisms to do?)

Answers to these basic questions demystify TCP/IP and create the foundation of knowledge that you build upon throughout this book. After finishing this book, you will have a solid appreciation for TCP/IP, how it works, and why it's important.

TCP/IP: The Official Protocol of the Internet!

TCP/IP isn't just random characters left in the bottom of the bowl when you're finished with your alphabet soup; it just looks that way! ***TCP/IP*** is a mouthful of an acronym that stands for ***Transmission Control Protocol/Internet Protocol***. One look at the full name and it's immediately obvious why you use the acronym instead. Neither its full name nor its acronym do TCP/IP any justice. Both are uninformative and even a bit misleading.

It's impossible to understate TCP/IP's value and significance. Without it, you literally wouldn't be able to access the Internet or anything connected to it. Think about it: no swapping MP3 or DivX files, no catching the latest news or stock quotes, no e-mail, no instant messages or online chat rooms…okay, so maybe you could live without chat rooms! The point remains: Everything you do via the Internet would be impossible without TCP/IP. A bunch of other private networks would be equally unusable, too!

TCP/IP is a set of data communications mechanisms, embodied in software, that let you use the Internet and countless other private networks. Each mechanism—also known as a ***protocol***—is designed to perform a specific function. These protocols are divided into two categories based on their function:

- One focuses on processing and handling data from applications and is known as the Transmission Control Protocol (TCP) suite.

- The other is more network oriented and designed to accommodate the transmission and receipt of application data across a network. This second suite is called the Internet Protocol (IP) suite.

This extensive set of protocols forms the TCP/IP suite of protocols. TCP/IP enables different types of computers and other devices to use networks to contact each other and share information in a wide variety of ways including e-mail. Networks and networking protocols existed before TCP/IP came along; they just didn't allow a network as powerful and universal as the Internet to be built! That was the beauty of TCP/IP.

What Is TCP/IP?

TCP/IP is an open-standard communications protocol suite that is the standard for communicating on the Internet. That's the official line, but what exactly *is* it? That's a tougher question to answer! Part of the reason TCP/IP is a murky and mysterious topic—even to those people knowledgeable about data networks—is that it's so vast and complex. It contains all the fundamental mechanisms needed to support any and all types of networked communications.

A good way to make sense of this is to look at your computer's operating system. Today's operating systems can easily be several gigabytes in size. They include thousands of small files designed to do just one specific task. Separately, they are almost worthless. Together, they form a powerful and comprehensive system that enables your computer to support a seemingly infinite variety of applications. TCP/IP is just like that: lots of little functions designed to do one specific task. Together, they enable your computer to support any type of networked communications activity you'd care to do.

Virtually every network-capable application in existence today relies on TCP/IP to function properly. That's no small accomplishment when you stop to think that today just about every software application is network capable. E-mail and web browsing are applications that obviously require a network to function. The World Wide Web has permeated computing so completely that it is difficult to say where the browser ends and the operating system begins. Consequently, even applications such as word-processing software now have features that require network connectivity—and TCP/IP—to work properly. To further complicate matters,

TCP/IP also contains a number of useful applications that enable you to do things like transfer files or log on to remote computers. These are native to TCP/IP and are what's left of the earliest TCP/IP software utilities.

Today, TCP/IP is maintained in an open forum. The forum consists of representatives from various companies that have a vested interest in the Internet and its technology base. Each representative participates in developing proposed changes, analyzing the impact of each proposal, and voting on whether to actually incorporate each change into the TCP/IP suite.

Changes to the protocol suite requires consensus among all the participants in that forum. This consensus-based approach enables computers made by different companies to communicate and share data despite their differences. The result is a set of communications mechanisms that does not belong to any one company. Instead, it is a public, or *open*, technology that can be implemented by anybody. Chapter 3, "The Quest for Freedom of Choice," more closely looks at this open forum and its consensus-based approach to building standards.

The combination of TCP/IP's sheer size and the unconventional way in which it is maintained make for a rather steep learning curve for anyone trying to figure it out. To best understand this arcane subject, look at where it came from and trace its evolution.

Finding TCP/IP's Roots

Given the enormous success of the Internet and TCP/IP, it is ironic that TCP/IP was intended for use in a small environment. Even more irony lies in the fact that TCP/IP wasn't even designed! It emerged slowly over time, first appearing as shareware created by computer engineers, scientists, and researchers at different organizations as a way to communicate and collaborate on defense projects. Despite such humble origins, TCP/IP is now actually used more broadly than the Internet.

The Birth of TCP/IP

The roots of TCP/IP are difficult to pin down due to the nature of its piecemeal development, but one recognized milestone (sometimes called its "birth") occurred on January 1, 1983. On this date, approximately 400 computers connected to a network called ARPAnet all started communicating with each other with a collection of mechanisms known informally as TCP/IP. ***ARPAnet*** was a network sponsored by the U.S. Department of Defense (DoD) in support of its *Advanced Research Projects Agency (ARPA)* to facilitate the research and development of military and defense technologies. This network interconnected computers that were owned and operated by different companies, government agencies, research bodies, and other institutions, which all contributed to the ARPA's research and development efforts.

These 400 computers also consisted of a wide variety of different brands and models in a time when each hardware manufacturer went out of its way to make sure that its products were incompatible with anybody else's products. This approach, known as ***closed*** or ***proprietary architecture***, was based on the belief that it forced brand loyalty. In a closed architecture, if you bought Brand X for your mainframe computer (this was before the client/server revolution), you had to also buy Brand X's printers, workstations, disk and tape drives, and even wires! After you purchased those products you had to purchase ongoing maintenance or product support from that manufacturer. Products made by different vendors quite simply weren't designed to be interoperable.

TCP/IP became a unifying element that enabled collaboration and interoperability across incompatible systems. The benefits of this approach proved so compelling as to literally revolutionize networked computing and transform the DoD's little internetwork into the Internet!

The Early Days of Networking

ARPAnet existed (in fact, it dated back to the 1960s) prior to January 1, 1983, but lacked a single official mechanism for communications. Thus, the proprietary architectures of that era's computer systems posed significant barriers to networked

communications. The result was that some computers could communicate with others, but that ability was far from ubiquitous. Some of the systems administrators who operated these disparate computers talked to each other and developed special-purpose programs. One person might have, for example, developed a small program that transferred data files. Another might have developed a program that enabled you to log into a remote computer. Somebody else might have developed a utility that helped track how many machines your data was passing through before reaching its destination. These files might have been developed informally, but they were enormously useful. Consequently, the programs were shared throughout ARPAnet.

This informal and sometimes collaborative process of developing utility software slowly built up the set of shared mechanisms that ARPAnet used. It actually took a couple of decades of piecemeal, consensus-based development for TCP/IP to reach a critical mass in terms of features and functionality. After ARPAnet declared TCP/IP its official communications protocol, there was no turning back!

Enabling the Internet's Success

TCP/IP soldiered on through the 1980s, laboring in the shadows of other, more popular and successful commercial network protocols. ARPAnet continued growing and eventually became known as the *Internet*. TCP/IP remained dedicated to supporting this network, but saw widespread acceptance. This occurred when the U.S. federal government converted Internet from a private military-oriented research facility into a public facility to which anyone could connect from around the world.

Commercialization of the Internet brought forth a frenzy of activity, both technical and economic. The first knee-jerk reaction was akin to a 20th century gold rush. Companies rushed to establish connectivity and a website presence on the Internet. Often, this rush was done with undue haste and without thought of what said companies would do with that connectivity or website.

When the Internet fever subsided, rational thought once again was possible. Individual people learned how to use the Internet, and commercial enterprises learned

what those individual consumers wanted. One stark fact became obvious: TCP/IP had just experienced a dramatic increase in its user base. Not only was it now the de facto standard communications protocol for all connected residential users, but enterprises and organizations also had to support it. That meant many businesses were using one protocol suite for communicating across their internal network and another (TCP/IP) to connect to the Internet.

It didn't take long for businesses to realize that TCP/IP could be used throughout their internal networks and could free them from the burden of supporting two different protocols. In this manner, TCP/IP literally became more popular than the Internet.

One of TCP/IP's most basic yet critical functions is its preparation of application data for transmission across a network. TCP/IP accepts data of virtually any size from applications (typically in the form of a file) and chops it up into smaller, more manageable chunks. These chunks are known as *segments* and the process of splitting files is known as *segmentation*. Segments of data are then wrapped in a data structure known as a *packet*. A packet contains all the information a network needs to deliver it to its destination and then to acknowledge delivery.

Packetization is the communication by two or more computers by sending and receiving individual packets of data. It was first demonstrated on September 2, 1969, when two computers at the University of California-Los Angeles were linked together with a 15-foot cable. This cable functioned as a primitive internetwork, and the test validated the theory that computers could send and receive packetized data. Until then, data was sent one bit at a time without the benefit of having first been segmented for transmission. This simple test paved the way for radical changes in technology and society that no one could have predicted.

The Abilities and Limitations of TCP/IP

TCP/IP is a communications protocol that fits in the middle of a larger set of mechanisms. TCP/IP provides the linkage between communicating software applications and a physical network that carries messages sent by computers to other computers or devices. In this regard, TCP/IP complements and extends the capability of a physical network, but can't work without that network. Although TCP/IP does blur across some boundaries, it actually fits between the application software, operating system, and network or communications device such as a cable modem or local-area network (LAN). To visualize this, see Figure 1-1.

Figure 1-1   TCP/IP Fits Between the Network and Your Software

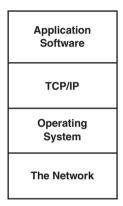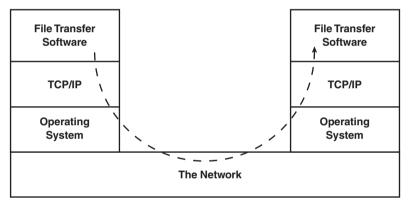| |
|---|
| **Application Software** |
| **TCP/IP** |
| **Operating System** |
| **The Network** |

Figure 1-1 shows a logical model. Logical models can be confusing due to their nature, but the best way to understand is to think of them as a necessary sequence of events. In this particular model, if you want to communicate across the Internet, you start with your application. Say you are sending a file directly to a friend across the Internet. The first thing you do is pick an application that lets you do that.

Fortunately, TCP/IP contains just such an application: It's called the *File Transfer Protocol (FTP)*. You supply FTP with the necessary information, which is basically your friend's computer address and the file you want to send. FTP needs to engage the operating system to access your computer's physical resources including memory, CPU, and the peripheral device that connects your computer to the physical network.

The preceding example shows how TCP/IP can blur the traditional distinctions between types of software. Even though TCP/IP is just a communications protocol, it also contains numerous utilities or applications, such as FTP. It's part communications protocol and part application software.

On the receiving end, your friend's computer performs the same process, only in reverse. The file comes through the network, where your operating system uses whatever physical resources it must and ensures that TCP/IP gets the file via its own copy of FTP. Figure 1-2 illustrates this process.

Figure 1-2   A Logical Perspective of Using TCP/IP to Transfer a File to a Friend

In its role as intermediary between software and the physical network, TCP/IP provides the rules and mechanisms that enable two or more different machines—even if they are very different machines—to communicate and share information with each other.

TCP/IP's mechanisms are designed to enable communications between any two or more machines connected to a network regardless of the device type, who made them, or their physical proximity to each other. Bridging the gap between dissimilar devices is a phenomenally powerful concept: You have created a mechanism that brings everybody together regardless of where they are or what technologies they use.

As shown in the preceding FTP example, the boundaries between operating system, application software, and communications protocols are blurred quite a bit by TCP/IP. When installed, TCP/IP functions as an extension of the operating system. So seamless is this integration that many people aren't even aware that they are using TCP/IP to communicate!

Even more confusion is added when you remember that TCP/IP contains quite a few simple applications that enable a user to directly do simple tasks like push files through a network or log on to remote computers. Part IV, "User Services—Making the Most Use of Your Network," looks more closely at those utilities. Users who are familiar with such utilities might think of TCP/IP as an application. They'd have justification for thinking that way! However, these utilities are just one small piece of the comprehensive set of protocols that you know and love and call TCP/IP.

Economically, too, TCP/IP has been a boon. The concept of an open mechanism that is not owned or controlled by any one commercial organization has opened doors few people could have imagined possible. For example, TCP/IP has become widely accepted around the world simply because it becomes the means by which different manufacturers' products can interoperate with each other. More importantly, TCP/IP broke new ground by pioneering a new method for developing new technologies that anyone could manufacture. Now, entities trying to develop a new communications-based product or technology can simply start with TCP/IP's openly published specifications and focus on adding value based on that preexisting set of capabilities. More importantly, developing products around TCP/IP brings a tremendous base of users who could easily integrate any new products based on that protocol suite.

TCP/IP's Top Five Critical Functions

By now you're probably anxious to get into what TCP/IP can do for you. Some of the applications that rely on TCP/IP (such as web surfing and e-mail) have been discussed, but you have seen how one set of mechanisms can so successfully support a rich and diverse array of applications.

The answer lies in determining the basic underlying capabilities any application requires to communicate. At the risk of sounding like a late-night talk show host, TCP/IP's top five critical functions follow:

1. **Globally Unique Addresses**—An addressing system that permits you to uniquely identify any given machine connected to the Internet. This system affords the ability to create literally billions of unique addresses.

2. **Packet**—Merely having an address for a networked machine isn't enough to permit communication. You also need a mechanism that allows you to send data to, and receive data from, the machine with that address. That mechanism is called a *packet*. Packets contain all the address information that a TCP/IP network needs to enable two or more devices to talk across a network.

3. **Segmentation and Reassembly**—The ability to chop up application data into manageable pieces called *segments* is crucial to communicating across a network. These segments are then placed inside a packet for transmission across a network. The machine receiving this stream of packets can extract the segments of data and restore the data to its original form.

4. **Resequencing**—TCP/IP also enables computers that receive a series of related packets to put them back into the correct order. This ability is important because you can't guarantee that packets will be received in the order they were sent. Networks are made of intelligent and autonomous devices known as *routers*, which decide where to send each packet. Their decisions can vary based on congestion and failures discovered within the network. The result is that packets are commonly received out of sequence!

5. **Data Integrity**—Lastly, you can't always assume that what you put on a network will get to its destination intact. For that matter, you can't assume that it will get there at all! Lots of things can go wrong when data is traveling through a network. The result can be garbled data or even packets that get lost or dropped. TCP/IP is smart enough to detect data that was damaged or lost as it traveled through the network. Data that was damaged is discarded and TCP/IP's mechanisms ensure that—when appropriate—the damaged or lost block is retransmitted until it is successfully received.

TCP/IP can, of course, do many other things. Thus, this list is not comprehensive. Instead, these five things form the basis for communicating. Other features, including those that are part of TCP/IP and those commercially developed around TCP/IP, build upon this foundation.

What It Can't Do

TCP/IP is not a proverbial silver bullet. It offers tremendous capabilities, but it does have its limitations. Those limitations have more to do with the intended scope of the protocol suite than weaknesses. Please consider all the "can't dos" in this section as being by design rather than indicative of a flawed implementation.

First of all, TCP/IP is not an *operating system (OS)*. An OS is the software that runs on a computer and creates the environment in which other applications can run. The various Windows platforms are examples of operating systems. When installed, TCP/IP becomes an important extension of your computer's operating system, but it does not replace it.

There was a time when some software vendors billed their operating system software as a *Network Operating System (NOS)*, but that term is misleading. It indicates a computer OS that supports a robust ability to communicate. One critical component of that ability was the inclusion of communications protocols, such as TCP/IP. However, bundling a set of communications protocols with an operating system doesn't make those protocols the operating system! As networking ability has become a more integral part of both operating systems and application software, this distinction has fallen by the wayside; consumers began expecting their operating systems to offer a rich networking capability.

TCP/IP also isn't a network! Installing and configuring TCP/IP on your workstation or home computer does not, in and of itself, mean you can use that computer to communicate with other computers or devices. TCP/IP, by itself, is incapable of transmitting or carrying data. Those functions remain in the purview of a physical

network such as an Ethernet or a wide-area network (WAN). TCP/IP prepares data received from applications for transmission and provides all the logical mechanisms needed to send and receive data between two devices connected via a physical network.

Chapter Summary

TCP/IP is a large set—sometimes called a *suite*—of communications protocols that fit in the middle of a larger set of mechanisms. These protocols provide the linkage between applications that can communicate and a physical network that carries the communicated data. Thus, TCP/IP complements and extends the capability of a physical network, but can't work without it.

However, the best thing about TCP/IP isn't what it can do, but what it has enabled! TCP/IP has literally and truly enabled people and organizations around the world to get past the artificial boundaries of technology choices and vendor preferences. The protocol enables the global quest for freedom of choice and rewards those who have implemented TCP/IP with an unprecedented bounty of online information and services.

Enough of the history and the hype: You're ready for some more concrete information that solidifies your grasp of what TCP/IP is and how you use it every time you communicate via a network. The next chapter looks more closely at what TCP/IP means to you.

Chapter Review Questions

The following questions reinforce the key concepts in this chapter.

1. What is TCP/IP?

2. How are TCP/IP and the Internet related?

3. What five critical communications functions can TCP/IP perform?

4. How do you know when you are using TCP/IP?

5. What can't TCP/IP do? In other words, what communications functions must it rely on other mechanisms to do?