chapter 9

TCP/IP Applications

"The World Wide Web is the only thing I know of whose shortened form—www—takes three times longer to say than what it's short for."

—Douglas Adams

In this chapter, you will learn how to

- **Describe common Transport layer protocols**
- **Explain the power of port numbers**
- **Define common TCP/IP applications such as HTTP, HTTPS, Telnet, e-mail (SMTP, POP3, and IMAP4), and FTP**

We network to get work done. Okay, sometimes that "work" involves a mad gaming session in which I lay some smack down on my editors, but you know what I mean. Thus far in the book, everything you've read about networking involves connecting computers together. This chapter moves further up the OSI seven-layer model to look at applications such as Web browsers, e-mail messaging, and more.

To understand the applications that use TCP/IP networks, a tech needs to know the structures *below* those applications that make them work. Have you ever opened multiple Web pages on a single computer? Have you ever run multiple Internet programs, such as a Web browser, an e-mail client, and a chat program, all at the same time? Clearly, lots of data is moving back and forth between your computer and many other computers. If you have packets coming in from two, three, or more other computers, there has to be a mechanism, a process, something that knows where to send and receive that data.

In this chapter, you'll discover the process used by TCP/IP networks to make sure the right data gets to the right applications on your computer. This process works through the use of some very important Transport layer protocols, TCP, UDP, and ICMP, as well as something lightly touched on in the previous chapter, the magic of port numbering. When used together, TCP and UDP along with port numbers enable work to get done on a network.

Historical/Conceptual

Transport Layer Protocols

I hate to tell you this, but you've been lied to. Not by me. Even though thus far I've gone along with this Big Lie, it's time to stand up and tell you the truth.

There is no such thing as TCP/IP. *TCP over IP* is really many other things, such as *HTTP, DHCP, POP, and about 500 more terms over TCP; plus UDP and ICMP over IP*. Given that this overly complex but much more correct term is too hard to use, the people who invented this network protocol stack way back when decided to call it *TCP/IP*, even though that term is way too simplistic to cover all the functionality involved.

There is a strong movement out there that prefers the term "Internet Protocol" instead of the term "TCP/IP." This movement has not yet reached the CompTIA Network+ certification.

To enable you to appreciate how TCP/IP applications work, this chapter breaks down the many unmentioned protocols and shows how they help make applications work. To start this process, let's consider how human beings communicate and you'll see some very interesting commonalities between computers and people.

How People Communicate

Imagine you walk into a school cafeteria to get some lunch. You first walk up to the guy making custom deli sandwiches (this is a great cafeteria!) and say, "Hello!" He says, "How may I help you?" You say, "I'd like a sandwich please." He says, "What kind of sandwich would you like?" and you order your sandwich. After you get your sandwich you say, "Thanks!" and he says, "You're welcome." What a nice guy! In the networking world we would call this a **connection-oriented** communication. Both you and the lunch guy first acknowledge each other. You then conduct your communication; last, you close the communication.

While you're in line you see your friend Janet sitting at your usual table. The line is moving fast so you yell out, "Janet, save me a seat!" before you rush along in the line. In this case you're not waiting for her to answer; you just yell to her and hope she hears you. We call this a **connectionless** communication. There is no acknowledgment or any closing. You just yell out your communication and hope she hears it.

In the networking world any single communication between a computer and another computer is called a **session**. When you open a Web page, you make a session. When you text chat with your buddy, you make a session. All sessions must begin and eventually end.

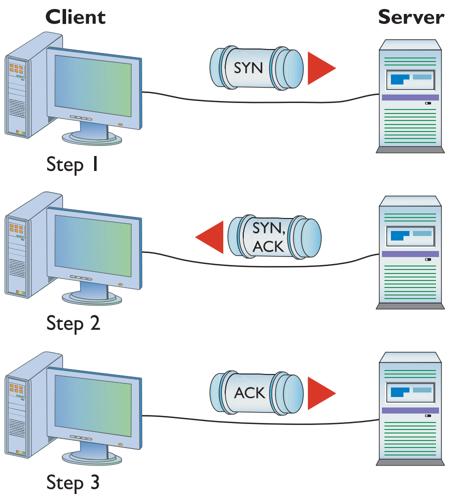
Test Specific

TCP

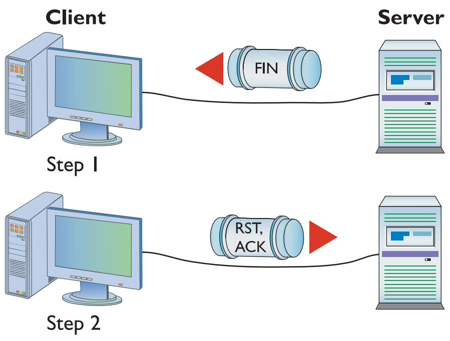
The **Transmission Control Protocol (TCP)** is how TCP/IP does connection-oriented communication. TCP is by far the most common type of session on a typical TCP/IP network. Figure 9.1 shows two computers. One computer (Server) runs a Web server and the other (Client) runs a Web browser. When you enter a computer's address in the browser running on Client, it sends a single SYN (synchronize) packet to the Web server. If Server gets that packet, it sends back a single SYN, ACK (synchronize, acknowledge) packet. Client then sends Server a single ACK packet, and immediately requests that Server begin sending the Web page.

Once Server completes sending the Web page, it sends a FIN (finished) packet. Client responds with an RST, ACK (reset, acknowledge) packet and the session is over (Figure 9.2).

Most TCP/IP applications use TCP because connection-oriented sessions are designed to check for errors. If a receiving computer detects a missing packet, it just asks for a repeat as needed.



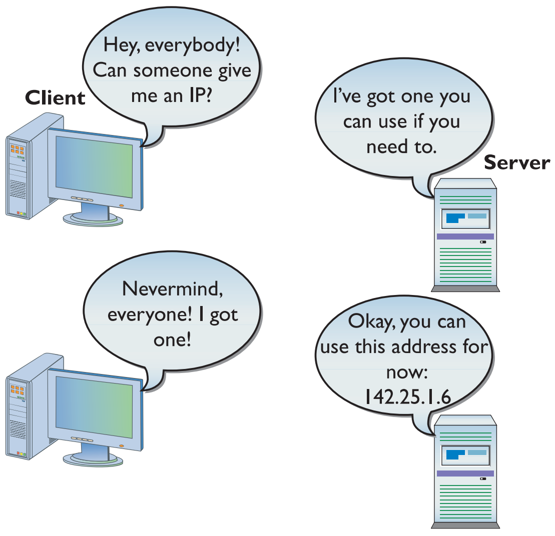
• **Figure 9.1** A connection-oriented session starting



• **Figure 9.2** A connection-oriented session ending

UDP

User Datagram Protocol (UDP) runs a distant second place to TCP in terms of the number of applications that use it, but don't let that make you think UDP is not important. UDP is perfect for the types of sessions that don't require the overhead of all that connection-oriented stuff. Probably the best example of an application that uses UDP is the Dynamic Host Configuration Protocol (DHCP). DHCP can't assume there's another computer ready on either side of the session, so each step of a DHCP session just sends the information for that step without any confirmation (Figure 9.3).


Client
Hey, everybody! Can someone give me an IP?
I've got one you can use if you need to.
Server
Nevermind, everyone! I got one!
Okay, you can use this address for now: 142.25.1.6

• **Figure 9.3** DHCP steps

You might also be tempted to think that UDP wouldn't work for any situation in which critical data transfer takes place—untrue! **Trivial File Transfer Protocol (TFTP)** is one example of a protocol with which you can transfer files from one machine to another. But TFTP, using UDP, doesn't have any data protection, so you would never use TFTP between computers across the Internet. TFTP is popular for moving files between computers on the same LAN, where the chances of losing packets is very small.

ICMP

While TCP and UDP differ dramatically—the former connection-oriented and the latter connectionless—both manage and modify packets in the classic sense with destination IP address, source IP address, destination port numbers, and source port numbers. A single session might be one packet or a series of packets.

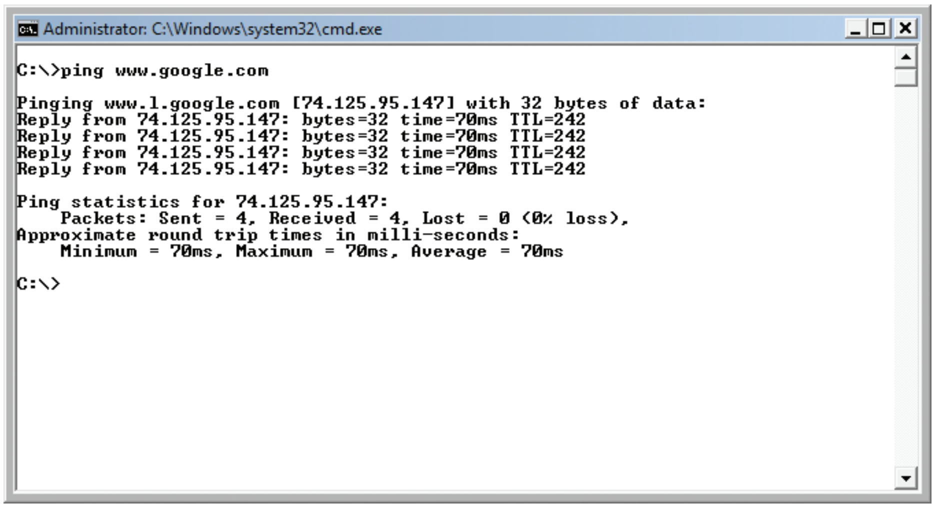
On the other hand, sometimes applications are so simple that they're always connectionless and never need more than a single packet. These are called **Internet Control Message Protocol (ICMP)** applications. Although you won't see ICMP too often in configuration files and such, you do see the results. ICMP handles mundane issues such as disconnect messages (host unreachable) that applications use to let the other side of a session know what's happening.

Good old PING is one place where you'll see ICMP in action. PING is an ICMP application that works by sending a single ICMP packet called an echo request to an IP address you specify. All computers running TCP/IP (assuming there is no firewall involved) respond to echo requests with an echo reply, as shown in Figure 9.4.

A *firewall* is a device or software that filters all the packets between two computers (or groups of computers) and acts like a club bouncer deciding who gets in and who gets blocked. Firewalls are vital for securing modern networks and will be discussed in Chapter 17.

IGMP

The **Internet Group Management Protocol (IGMP)** is another little-seen but important Transport layer protocol. Do you remember the idea of IP



```
C:\>ping www.google.com

Pinging www.l.google.com [74.125.95.147] with 32 bytes of data:
Reply from 74.125.95.147: bytes=32 time=70ms TTL=242
Reply from 74.125.95.147: bytes=32 time=70ms TTL=242
Reply from 74.125.95.147: bytes=32 time=70ms TTL=242
Reply from 74.125.95.147: bytes=32 time=70ms TTL=242

Ping statistics for 74.125.95.147:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 70ms, Maximum = 70ms, Average = 70ms

C:\>
```

• **Figure 9.4** PING in action

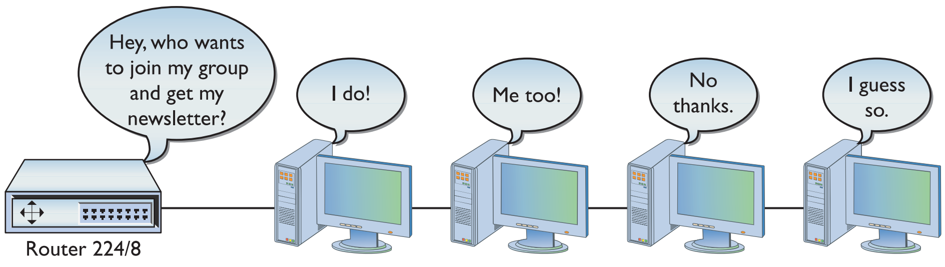
There is some argument over where ICMP and IGMP fit in the OSI seven-layer model. Both of these protocols lack port numbers, so for many people that makes ICMP and IGMP fit more into the Network layer than the Transport layer. When in doubt, place TCP and UDP in the Transport layer and ICMP and IGMP in the Network layer.

Cross Check

Multicast

You first saw multicast back in Chapter 7, "TCP/IP Basics," when you learned about classful IP addressing. Refer back to that chapter and see if you can answer these questions. What IP numbers are reserved for multicast? What Class is that? What is the difference between unicast and multicast?

multicast addresses, described in Chapter 7? The challenge to multicasting is determining who wants to receive the multicast and who does not. IGMP is the glue that routers use to communicate with hosts to determine a "group" membership. As you might remember from Chapter 7, the multicast Class D licenses are assigned the network ID of 224/8. However, multicast doesn't assign IP addresses to individual hosts in the same manner as you've seen thus far. Instead, a particular multicast (called an *IGMP group*) is assigned to a certain 224/8 address and those who wish to receive this multicast must tell their upstream router or switch (which must be configured to handle multicasts) that they wish to receive it. To do so, they join the IGMP group and then the router or switch knows to send that multicast data stream to that client (Figure 9.5).



• **Figure 9.5** IGMP in action

The Power of Port Numbers

If you want to understand the power of TCP/IP, you have to get seriously into port numbers. If you want to pass the CompTIA Network+ exam, you need to know how TCP/IP uses port numbers and you have to memorize a substantial number of common port numbers. As you saw in the previous chapter, the port numbers make NAT work. As you progress through this book, you'll see a number of places where knowledge of port numbers is critical to protect your network, to make routers work better, and to address a zillion other issues. There is no such thing as a network administrator who isn't deeply into the magic of port numbers and who cannot manipulate them for his or her network's needs.

Let's review and expand on what you learned about port numbers in the previous chapter. Thus far, you know that every TCP/IP application requires a server and a client. There are very clearly defined port numbers for every popular or *well-known* TCP/IP application. A port number is a 16-bit value between 0 and 65,535. Web servers, for example, use port number 80. Port numbers from 0 to 1023 are called **well-known port numbers** and are reserved for specific TCP/IP applications.

TCP/IP port numbers between 0 and 1023 are the well-known port numbers. You'll find them at *every* party.

When a Web client (let's say your computer running Firefox) sends an HTTP ACK to a Web server to request the Web page, your computer's IP packet looks like Figure 9.6.

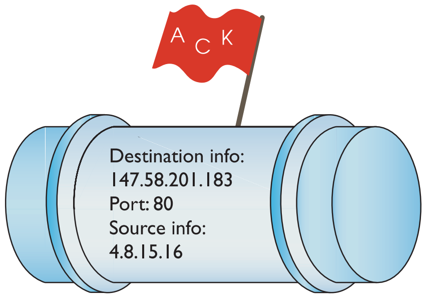
As you can see, the destination port number is 80. The computer running the Web server reads the destination port number, telling it to send the incoming packet to the Web server program (Figure 9.7).

The Web client's source port number is generated pseudo-randomly by the Web client computer. This value varies by operating system, but generally falls within the values 1024–5000—the port numbers classically assigned as **ephemeral port numbers**—and 49152–65535—the **dynamic** or **private port numbers**. In the early days of the Internet, only ports 1024–5000 were used, but modern computers can use up all of those. More port numbers were added later. The Internet Assigned Numbers Authority (IANA) today recommends using only ports 49152–65535 as ephemeral port numbers. Let's redraw Figure 9.6 to show the more complete packet (Figure 9.8).

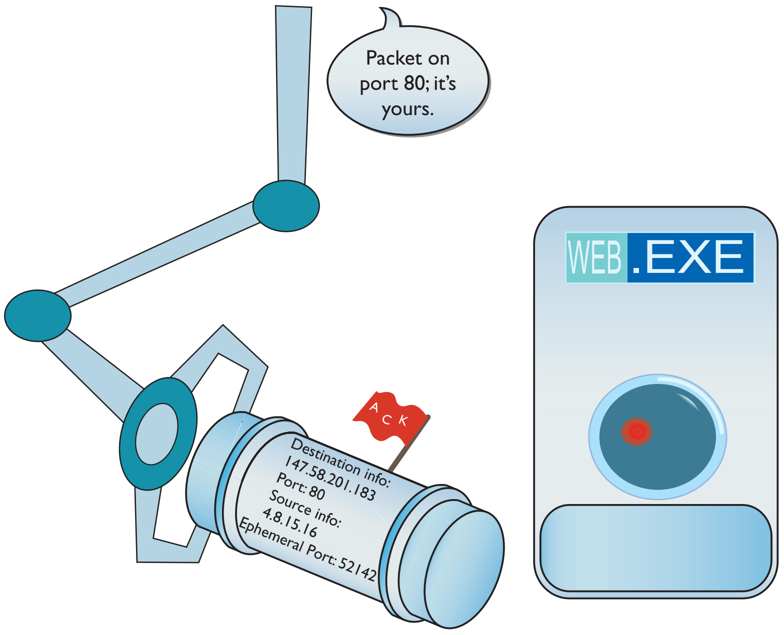
When the serving system responds to the Web client, it uses the ephemeral port number as the destination port to get the information back to the Web client running on the client computer (Figure 9.9).

Registered Ports

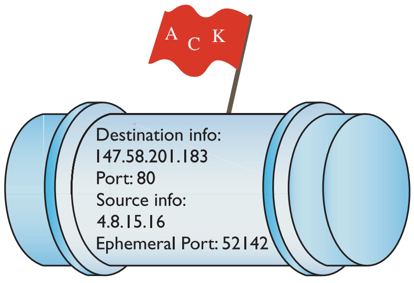
The port numbers from 1024 to 49151 are called **registered ports**. Less-common TCP/IP applications can register their ports with the IANA. Unlike well-known ports, anyone can use these port numbers for their servers or for ephemeral numbers on clients. Most operating systems steer away (or are in the



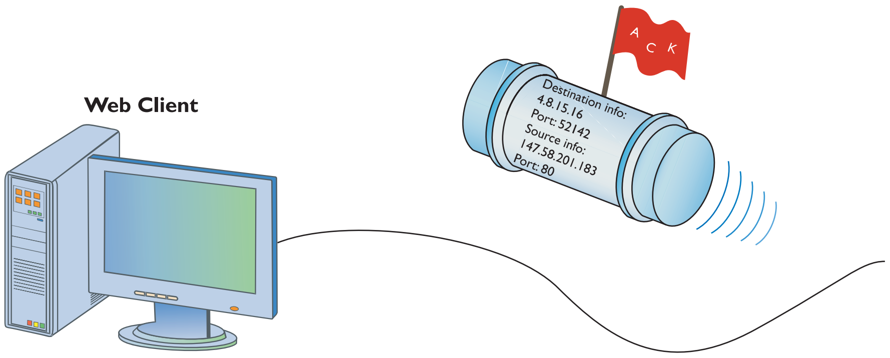
• **Figure 9.6** HTTP ACK packet



• **Figure 9.7** Dealing with the incoming packet



• **Figure 9.8** A more complete IP packet



• **Figure 9.9** Returning the packet

process of steering away) from using these port numbers for ephemeral ports, opting instead for the dynamic/private port numbers. Here's the full list of ports:

0–1023	Well-known port numbers
1024–49151	Registered ports
49152–65535	Dynamic or private ports

Each computer on each side of a session must keep track of the status of the communication. In the TCP/IP world, the session information stored in RAM is called a **socket** or **endpoint**. When discussing the data each computer stores about the connection between two computers' TCP/IP applications, the term to use is **socket pairs** or **endpoints**. A *session* or **connection** refers to the connection in general, rather than specific to TCP/IP. Many people still use the term *session*.

- Terms for the connection data stored on a single computer: socket or endpoint
- Terms for the connection data stored on two computers about the same connection: socket pairs or endpoints
- Terms for the whole interconnection: connection or session

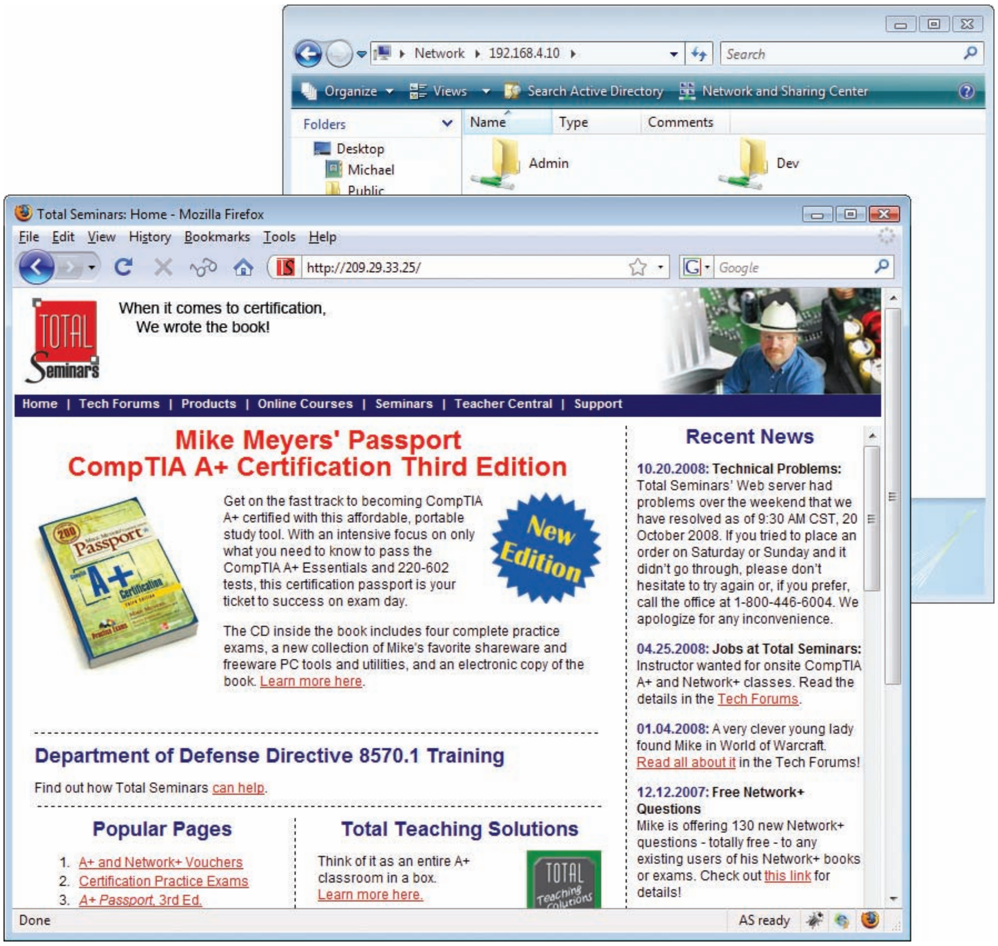
As two computers begin to communicate, they store the information about the session—the endpoints—so that they know where to send and receive data. At any given point in time your computer probably has a large number of communications going on. If you want to know whom your computer is communicating with, you need to see this list of endpoints. As you'll recall from Chapter 8, "The Wonderful World of Routing," Windows, Linux, and Macintosh OS X come with **NETSTAT**, the universal "show me the endpoint" utility. NETSTAT works at the command line, so open one up and type `netstat -n` to see something like this:

Even though almost all operating systems use NETSTAT, there are subtle differences in options and output.

```
C:\>netstat -n
Active Connections
  Proto  Local Address          Foreign Address        State
  TCP    192.168.4.27:57913     209.29.33.25:80        ESTABLISHED
  TCP    192.168.4.27:61707     192.168.4.10:445       ESTABLISHED
C:\>
```

When you run `netstat -n` on a typical computer, you'll see many more than just two connections! The preceding example is simplified for purposes of discussing the details. It shows two connections. My computer's IP address is 192.168.4.27. The top connection is an open Web page (port 80) to a server at 209.29.33.25. The second connection is an open Windows Network browser (port 445) to my file server (192.168.4.10). Looking on my Windows Desktop, you would certainly see at least these two windows open (Figure 9.10).

Don't think that a single open application always means a single connection. The following example shows what `netstat -n` looks like when I open the well-known www.microsoft.com Web site (I took out the



TOTAL Seminars

When it comes to certification,
We wrote the book!

Home | Tech Forums | Products | Online Courses | Seminars | Teacher Central | Support

Mike Meyers' Passport
CompTIA A+ Certification Third Edition

New Edition

Get on the fast track to becoming CompTIA A+ certified with this affordable, portable study tool. With an intensive focus on only what you need to know to pass the CompTIA A+ Essentials and 220-602 tests, this certification passport is your ticket to success on exam day.

The CD inside the book includes four complete practice exams, a new collection of Mike's favorite shareware and freeware PC tools and utilities, and an electronic copy of the book. Learn more here.

Department of Defense Directive 8570.1 Training

Find out how Total Seminars can help.

Popular Pages

1. A+ and Network+ Vouchers
2. Certification Practice Exams
3. A+ Passport, 3rd Ed.

Total Teaching Solutions

Think of it as an entire A+ classroom in a box.
Learn more here.

Recent News

10.20.2008: Technical Problems: Total Seminars' Web server had problems over the weekend that we have resolved as of 9:30 AM CST, 20 October 2008. If you tried to place an order on Saturday or Sunday and it didn't go through, please don't hesitate to try again or, if you prefer, call the office at 1-800-446-6004. We apologize for any inconvenience.

04.25.2008: Jobs at Total Seminars: Instructor wanted for onsite CompTIA A+ and Network+ classes. Read the details in the Tech Forums.

01.04.2008: A very clever young lady found Mike in World of Warcraft. Read all about it in the Tech Forums!

12.12.2007: Free Network+ Questions
Mike is offering 130 new Network+ questions - totally free - to any existing users of his Network+ books or exams. Check out this link for details!

• **Figure 9.10** Two open windows

connections that were not involved with the Web browser's connections to www.microsoft.com):

```
C:\>netstat -n
Active Connections
  Proto  Local Address          Foreign Address        State
  TCP    192.168.4.27:50015     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50016     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50017     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50018     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50019     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50020     80.12.192.51:80        ESTABLISHED
  TCP    192.168.4.27:50021     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50022     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50023     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50024     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50025     80.12.192.51:80        ESTABLISHED
  TCP    192.168.4.27:50027     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50028     80.12.192.40:80        ESTABLISHED
  TCP    192.168.4.27:50036     80.12.192.75:80        ESTABLISHED
```

NETSTAT enables you to see active TCP/IP connections at a glance.

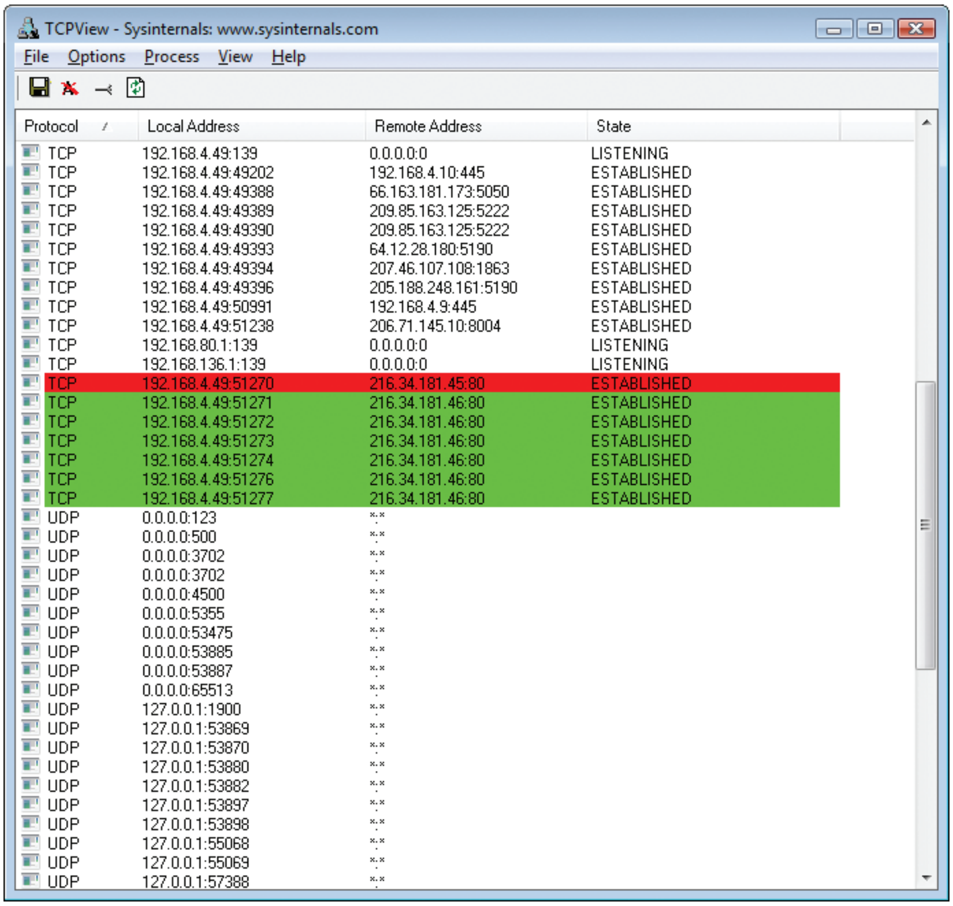
A single simple Web page needs only a single connection, but this is a very complex Web page. Different elements in the Web page, such as advertisements, each have their own connection.

NETSTAT is a powerful tool and you will see it used throughout this book. The CompTIA Network+ exam will also test your NETSTAT skills. On the other hand, connections come and go constantly on your computer and NETSTAT, being a command-line utility, can't update to reflect changes automatically. All of the cool, hip, network techs use graphical end-point tools. Take a moment right now and download the popular, powerful, and completely free TCPView, written by Mark Russinovich, the Guru of Windows utilities. Just type **TCPView** into your search engine to find it or try going here:

http://technet.microsoft.com/en-us/sysinternals/default.aspx

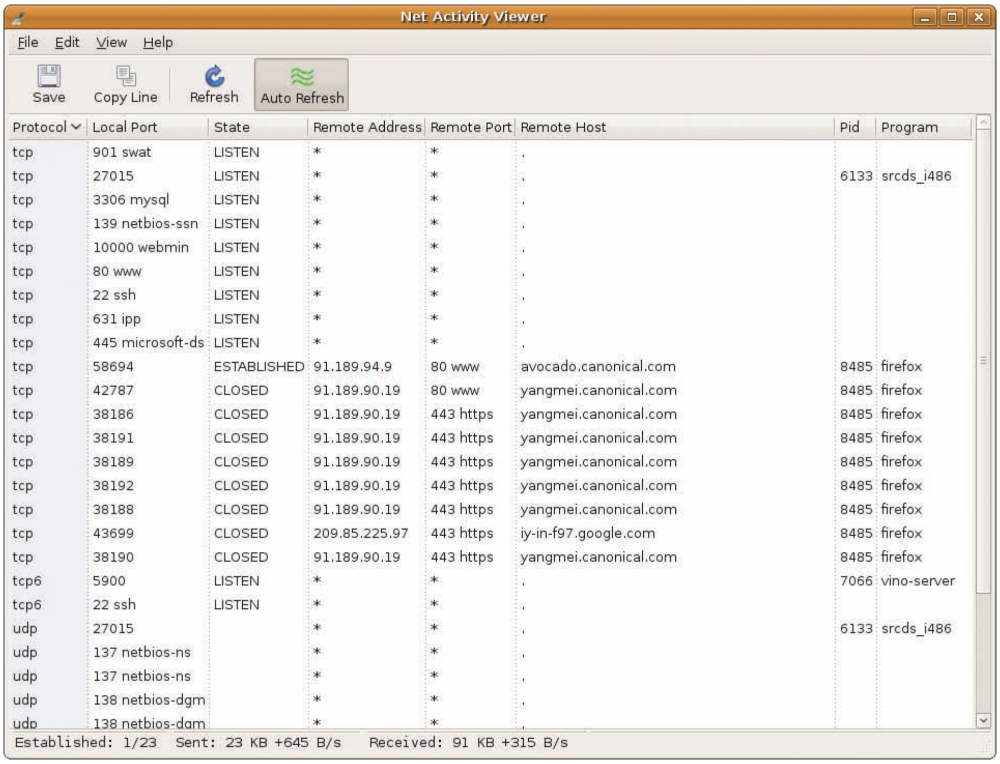
Click the **Networking Utilities** icon to get the latest copy. Figure 9.11 shows TCPView in action. Note the red and green bars: red is for closing connections and green shows new connections as they appear.

TCPView won't work on anything but Windows, but other operating systems have equivalent programs. Linux folks often use the popular Net Activity Viewer (Figure 9.12). You can grab a copy of this program at http://netactview.sourceforge.net.


TCPView - Sysinternals: www.sysinternals.com
File Options Process View Help

Protocol	Local Address	Remote Address	State
TCP	192.168.4.49:139	0.0.0.0:0	LISTENING
TCP	192.168.4.49:49202	192.168.4.10:445	ESTABLISHED
TCP	192.168.4.49:49388	66.163.181.173:5050	ESTABLISHED
TCP	192.168.4.49:49389	209.85.163.125:5222	ESTABLISHED
TCP	192.168.4.49:49390	209.85.163.125:5222	ESTABLISHED
TCP	192.168.4.49:49393	64.12.28.180:5190	ESTABLISHED
TCP	192.168.4.49:49394	207.46.107.108:1863	ESTABLISHED
TCP	192.168.4.49:49396	205.188.248.161:5190	ESTABLISHED
TCP	192.168.4.49:50991	192.168.4.9:445	ESTABLISHED
TCP	192.168.4.49:51238	206.71.145.10:8004	ESTABLISHED
TCP	192.168.80.1:139	0.0.0.0:0	LISTENING
TCP	192.168.136.1:139	0.0.0.0:0	LISTENING
TCP	192.168.4.49:51270	216.34.181.45:80	ESTABLISHED
TCP	192.168.4.49:51271	216.34.181.46:80	ESTABLISHED
TCP	192.168.4.49:51272	216.34.181.46:80	ESTABLISHED
TCP	192.168.4.49:51273	216.34.181.46:80	ESTABLISHED
TCP	192.168.4.49:51274	216.34.181.46:80	ESTABLISHED
TCP	192.168.4.49:51276	216.34.181.46:80	ESTABLISHED
TCP	192.168.4.49:51277	216.34.181.46:80	ESTABLISHED
UDP	0.0.0.0:123	*:*	
UDP	0.0.0.0:500	*:*	
UDP	0.0.0.0:3702	*:*	
UDP	0.0.0.0:3702	*:*	
UDP	0.0.0.0:4500	*:*	
UDP	0.0.0.0:5355	*:*	
UDP	0.0.0.0:53475	*:*	
UDP	0.0.0.0:53885	*:*	
UDP	0.0.0.0:53887	*:*	
UDP	0.0.0.0:65513	*:*	
UDP	127.0.0.1:1900	*:*	
UDP	127.0.0.1:53869	*:*	
UDP	127.0.0.1:53870	*:*	
UDP	127.0.0.1:53880	*:*	
UDP	127.0.0.1:53882	*:*	
UDP	127.0.0.1:53897	*:*	
UDP	127.0.0.1:53898	*:*	
UDP	127.0.0.1:55068	*:*	
UDP	127.0.0.1:55069	*:*	
UDP	127.0.0.1:57388	*:*	

● **Figure 9.11** TCPView in action



Protocol	Local Port	State	Remote Address	Remote Port	Remote Host	Pid	Program
tcp	901 swat	LISTEN	*	*	.		
tcp	27015	LISTEN	*	*	.	6133	srcds_i486
tcp	3306 mysql	LISTEN	*	*	.		
tcp	139 netbios-ssn	LISTEN	*	*	.		
tcp	10000 webmin	LISTEN	*	*	.		
tcp	80 www	LISTEN	*	*	.		
tcp	22 ssh	LISTEN	*	*	.		
tcp	631 ipp	LISTEN	*	*	.		
tcp	445 microsoft-ds	LISTEN	*	*	.		
tcp	58694	ESTABLISHED	91.189.94.9	80 www	avocado.canonical.com	8485	firefox
tcp	42787	CLOSED	91.189.90.19	80 www	yangmei.canonical.com	8485	firefox
tcp	38186	CLOSED	91.189.90.19	443 https	yangmei.canonical.com	8485	firefox
tcp	38191	CLOSED	91.189.90.19	443 https	yangmei.canonical.com	8485	firefox
tcp	38189	CLOSED	91.189.90.19	443 https	yangmei.canonical.com	8485	firefox
tcp	38192	CLOSED	91.189.90.19	443 https	yangmei.canonical.com	8485	firefox
tcp	38188	CLOSED	91.189.90.19	443 https	yangmei.canonical.com	8485	firefox
tcp	43699	CLOSED	209.85.225.97	443 https	iy-in-f97.google.com	8485	firefox
tcp	38190	CLOSED	91.189.90.19	443 https	yangmei.canonical.com	8485	firefox
tcp6	5900	LISTEN	*	*	.	7066	vino-server
tcp6	22 ssh	LISTEN	*	*	.		
udp	27015		*	*	.	6133	srcds_i486
udp	137 netbios-ns		*	*	.		
udp	137 netbios-ns		*	*	.		
udp	138 netbios-dgm		*	*	.		
udp	138 netbios-dgm		*	*	.		

```
Established: 1/23   Sent: 23 KB +645 B/s    Received: 91 KB +315 B/s
```

• **Figure 9.12** Net Activity Viewer

Connection Status

Connection states change continually and it's helpful when using tools such as NETSTAT or TCPView to understand their status at any given moment. Let's look at the status of connections so you understand what each means—this information is useful for determining what's happening on networked computers.

A socket that is prepared to respond to any IP packets destined for that socket's port number is called an **open port** or **listening port**. Every serving application has an open port. If you're running a Web server on a computer, for example, it will have an open port 80. That's easy enough to appreciate but you'll be amazed at the number of open ports on just about *any* computer. Fire up a copy of NETSTAT and type `netstat -an` to see all of your listening ports. Running `netstat -an` gives a lot of information, so let's just look at a small amount:

```
C:\>netstat -an
Active Connections
  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:7              0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:912            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:990            0.0.0.0:0              LISTENING
  TCP    127.0.0.1:27015        0.0.0.0:0              LISTENING
  TCP    127.0.0.1:52144        127.0.0.1:52145        ESTABLISHED
  TCP    127.0.0.1:52145        127.0.0.1:52144        ESTABLISHED
```

The –a switch tells NETSTAT to show all used ports. –n instructs NETSTAT to show raw port numbers and IP addresses.

```
TCP    127.0.0.1:52146        127.0.0.1:52147        ESTABLISHED
TCP    127.0.0.1:52147        127.0.0.1:52146        ESTABLISHED
TCP    192.168.4.27:139       0.0.0.0:0              LISTENING
TCP    192.168.4.27:52312     74.125.47.108:80       TIME_WAIT
TCP    192.168.4.27:57913     63.246.140.18:80       CLOSE_WAIT
TCP    192.168.4.27:61707     192.168.4.10:445       ESTABLISHED
```

First look at this line:

```
TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
```

This line shows a listening port, listening and ready for incoming packets that have a destination port number of 445. Notice the local address is 0.0.0.0. This is how Windows tells you that the open port works on all NICs on this PC. In this case my PC has only one NIC (192.168.4.27), but even if you have only one NIC, NETSTAT still shows it this way. This computer is sharing some folders on the network. Since at this moment there's no one connected, NETSTAT shows the Foreign Address as 0.0.0.0. Incoming requests use port number 445 to connect to those shared folders. If another computer on my network (192.168.4.83) was accessing the shared folders, this line would look like

```
TCP    192.168.4.27:445       192.168.4.83:1073      ESTABLISHED
```

Established ports are active, working endpoint pairs.

Over time all connections eventually close like this one:

```
TCP    192.168.4.27:57913     63.246.140.18:80       CLOSE_WAIT
```

This line shows a Web browser making a graceful closure, meaning that each side of the conversation sees the session closing normally.

Not all connections close gracefully. The following line shows a Web browser that has lost the connection to the other side and is waiting a defined amount of time:

```
TCP    192.168.4.27:52312     74.125.47.108:80       TIME_WAIT
```

This is called a timeout period. Most Web browsers time out in around two minutes.

If data's going to move back and forth between computers, there always must be some program that's doing the sending and/or receiving. Take a look at this line from `netstat -an`:

```
TCP    192.168.4.27:52312     74.125.47.108:80       ESTABLISHED
```

You see the 80 and might assume the connection is going out to a Web server. But what program on the computer is sending it? Enter the command `netstat -ano` (the –o switch tells NETSTAT to show the process ID). While you'll see many lines, the one for this connection looks like this:

```
Proto  Local Address          Foreign Address        State           PID
TCP    192.168.4.27:52312     74.125.47.108:80       ESTABLISHED     112092
```

Every running program on your computer gets a process ID (PID), a number used by the operating system to track all the running programs. Numbers aren't very helpful to you, though, because you want to know the name of the running program. In most operating systems, finding this out is fairly easy to do. In Linux you can use the ps command:

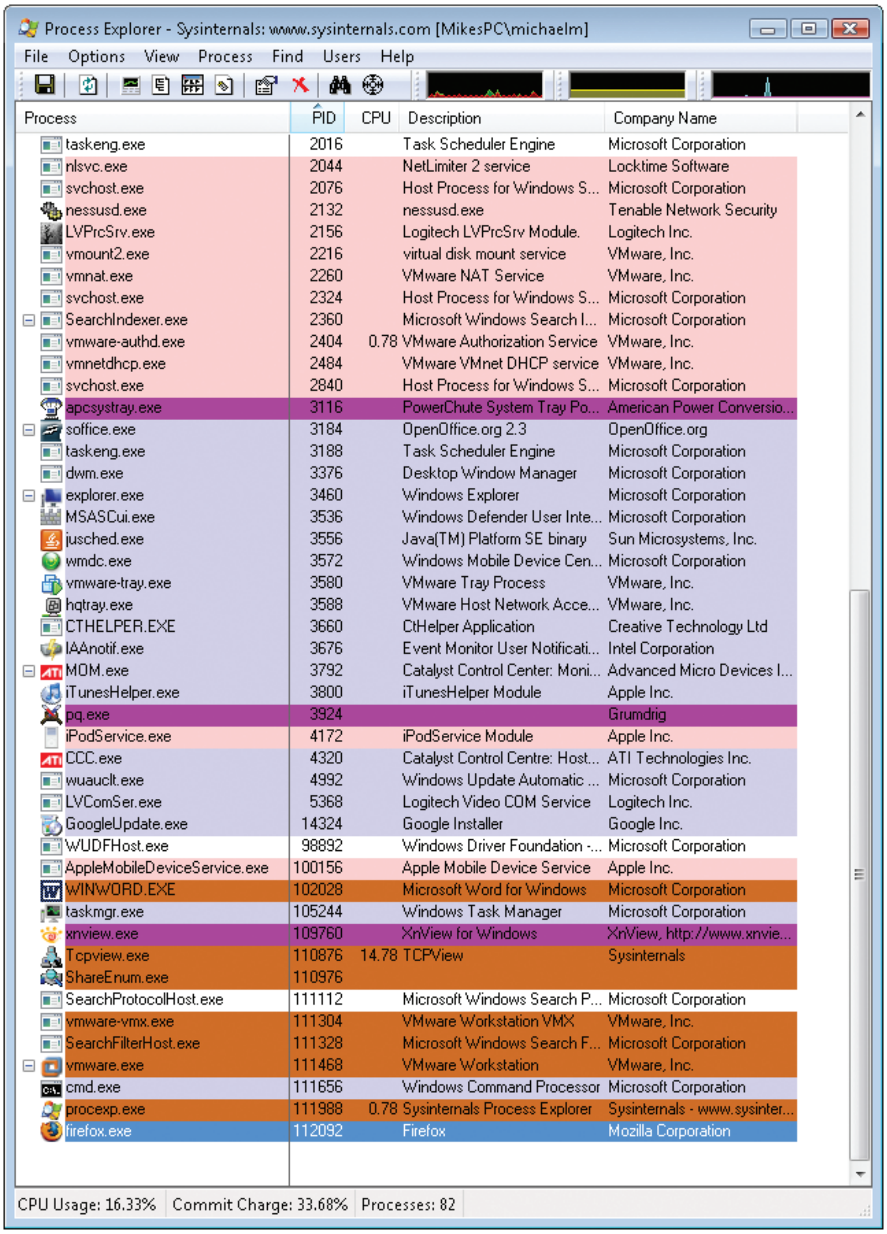
```
michaelm@ubuntu:~$ ps
PID TTY          TIME CMD
3225 pts/1    00:00:00 bash
3227 pts/1    00:00:00 ps
```

Windows doesn't come with an easy tool to determine what programs are using a certain PID, so once again we turn to Mark Russinovich. His Process Explorer is a perfect tool for this (Figure 9.13). The figure shows Process Explorer scrolled down to the bottom so that you can see the program using PID 112092—good old Firefox!

To get Process Explorer, enter **"Process Explorer"** in your search engine to find it or try going here:

http://technet.microsoft.com/en-us/sysinternals/default.aspx

Click the **Process Utilities** icon to get the latest copy.


Process Explorer - Sysinternals: www.sysinternals.com [MikesPC\michaelm]
File Options View Process Find Users Help

Process	PID	CPU	Description	Company Name
taskeng.exe	2016		Task Scheduler Engine	Microsoft Corporation
nlsvc.exe	2044		NetLimiter 2 service	Locktime Software
svchost.exe	2076		Host Process for Windows S...	Microsoft Corporation
nessusd.exe	2132		nessusd.exe	Tenable Network Security
LVPrcSrv.exe	2156		Logitech LVPrcSrv Module.	Logitech Inc.
vmount2.exe	2216		virtual disk mount service	VMware, Inc.
vmnat.exe	2260		VMware NAT Service	VMware, Inc.
svchost.exe	2324		Host Process for Windows S...	Microsoft Corporation
SearchIndexer.exe	2360		Microsoft Windows Search I...	Microsoft Corporation
vmware-authd.exe	2404	0.78	VMware Authorization Service	VMware, Inc.
vmnetdhcp.exe	2484		VMware VMnet DHCP service	VMware, Inc.
svchost.exe	2840		Host Process for Windows S...	Microsoft Corporation
apcsystray.exe	3116		PowerChute System Tray Po...	American Power Conversio...
soffice.exe	3184		OpenOffice.org 2.3	OpenOffice.org
taskeng.exe	3188		Task Scheduler Engine	Microsoft Corporation
dwm.exe	3376		Desktop Window Manager	Microsoft Corporation
explorer.exe	3460		Windows Explorer	Microsoft Corporation
MSASCui.exe	3536		Windows Defender User Inte...	Microsoft Corporation
jusched.exe	3556		Java(TM) Platform SE binary	Sun Microsystems, Inc.
wmdc.exe	3572		Windows Mobile Device Cen...	Microsoft Corporation
vmware-tray.exe	3580		VMware Tray Process	VMware, Inc.
hqtray.exe	3588		VMware Host Network Acce...	VMware, Inc.
CTHELPER.EXE	3660		CtHelper Application	Creative Technology Ltd
IAAnotif.exe	3676		Event Monitor User Notificati...	Intel Corporation
MOM.exe	3792		Catalyst Control Center: Moni...	Advanced Micro Devices I...
iTunesHelper.exe	3800		iTunesHelper Module	Apple Inc.
pq.exe	3924			Grumdrig
iPodService.exe	4172		iPodService Module	Apple Inc.
CCC.exe	4320		Catalyst Control Centre: Host...	ATI Technologies Inc.
wuauclt.exe	4992		Windows Update Automatic ...	Microsoft Corporation
LVComSer.exe	5368		Logitech Video COM Service	Logitech Inc.
GoogleUpdate.exe	14324		Google Installer	Google Inc.
WUDFHost.exe	98892		Windows Driver Foundation -...	Microsoft Corporation
AppleMobileDeviceService.exe	100156		Apple Mobile Device Service	Apple Inc.
WINWORD.EXE	102028		Microsoft Word for Windows	Microsoft Corporation
taskmgr.exe	105244		Windows Task Manager	Microsoft Corporation
xnview.exe	109760		XnView for Windows	XnView, http://www.xnvie...
Tcpview.exe	110876	14.78	TCPView	Sysinternals
ShareEnum.exe	110976			
SearchProtocolHost.exe	111112		Microsoft Windows Search P...	Microsoft Corporation
vmware-vmx.exe	111304		VMware Workstation VMX	VMware, Inc.
SearchFilterHost.exe	111328		Microsoft Windows Search F...	Microsoft Corporation
vmware.exe	111468		VMware Workstation	VMware, Inc.
cmd.exe	111656		Windows Command Processor	Microsoft Corporation
procexp.exe	111988	0.78	Sysinternals Process Explorer	Sysinternals - www.sysinter...
firefox.exe	112092		Firefox	Mozilla Corporation

CPU Usage: 16.33% Commit Charge: 33.68% Processes: 82

• **Figure 9.13** Process Explorer

You might be tempted to say "Big whoop, Mike—what else would use port 80?" Then consider the possibility that you run NETSTAT and see a line like the one just shown, but *you don't have a browser open*! You determine the PID and discover that the name of the process is "Evil_Overlord.exe." This is something running on your computer that should not be there.

Understanding how TCP/IP uses ports is a base skill for any network tech. To pass the CompTIA Network+ exam, you need to memorize a number of different well-known ports and even a few of the more popular registered ports. You must appreciate how the ports fit into the process of TCP/IP communications and know how to use NETSTAT and other tools to see what's going on inside your computer.

The biggest challenge is learning what's supposed to be running and what's not. No one on Earth can run a NETSTAT command and instantly recognize every connection and why it's running, but a good network tech should know most of them. For those connections that a tech doesn't recognize, he or she should know how to research them to determine what they are.

Rules for Determining Good vs. Bad Communications

Here is the general list of rules I follow for determining good versus bad communications (as far as networking goes, at least!):

1. Memorize a bunch of known ports for common TCP/IP applications. The next section in this chapter will get you started.
2. Learn how to use NETSTAT to see what's happening on your computer. Learn to use switches such as –a, –n, and –o to help you define what you're looking for.
3. Take the time to learn the ports that normally run on your operating system. When you see a connection using ports you don't recognize, figure out the process running the connection using a utility such as Linux's **ps** or Process Explorer for Windows.
4. Take the time to learn the processes that normally run on your operating system. Most operating systems have their own internal programs (such as Windows SVCHOST.EXE) that are normal and important processes.
5. When you see a process you don't recognize, just enter the filename of the process in a Web search. There are hundreds of Web sites dedicated to researching mystery processes that will tell you what the process does.
6. Get rid of bad processes.

Common TCP/IP Applications

Finally! You now know enough about the Transport layer, port numbering, and sockets to get into some of the gritty details of common TCP/IP applications. There's no pretty way to do this, so let's start with the big daddy of them all, the Web.

The World Wide Web

Where would we be without the World Wide Web? If you go up to a non-nerd and say "Get on the Internet," most of them will automatically open a Web browser, because to them the Web *is* the Internet. I think it's safe to assume you've used the Web, firing up your Web browser to surf to one cool site after another, learning new things, clicking links, often ending up somewhere completely unexpected . . . it's all fun! This section looks at the Web and the tools that make it function, specifically the protocols that enable communication over the Internet.

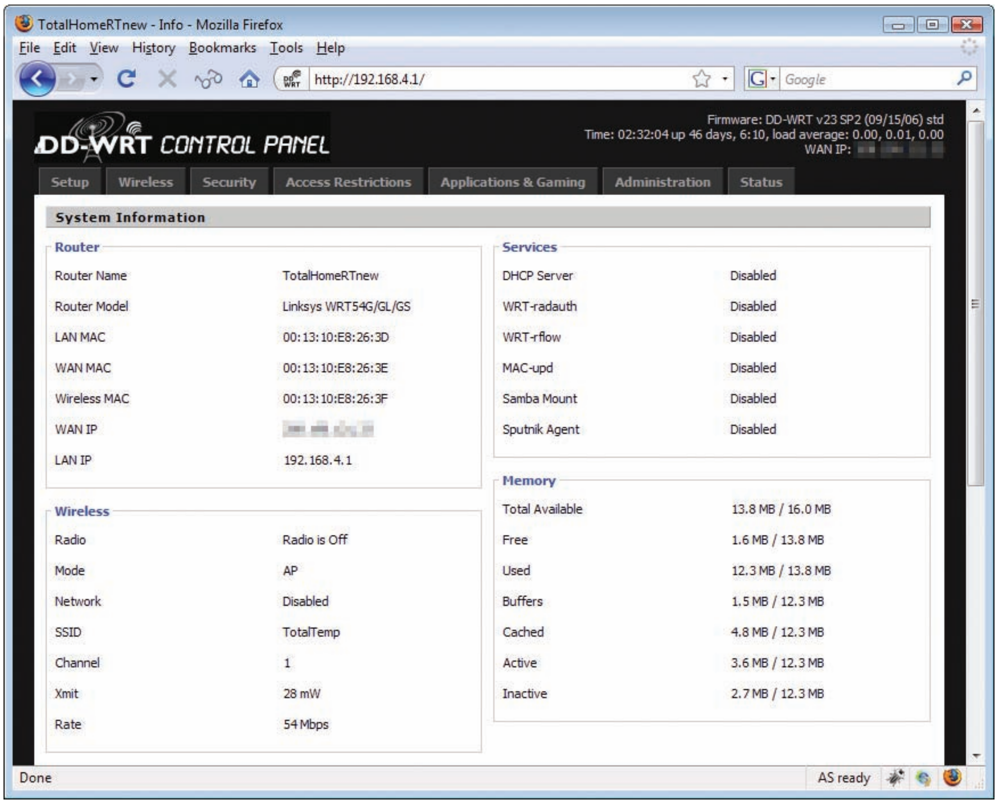
The Web is composed of servers that store specially formatted documents using a language called Hypertext Markup Language (HTML). Figure 9.14 shows the Web page built into my router.

XML is a markup language similar to HTML, but it provides more flexibility in how the data is described while being much more strict in its syntax.

HTML has been around for a long time and as a result has gone through many versions. Today many pages are being written in an updated HTML version called eXtensible HTML (XHTML), which is HTML with an XML syntax. Figure 9.15 shows a small part of the XHTML code for the Web page shown in Figure 9.14.

Web browsers are designed to request HTML pages from Web servers and then open them. To access a Web page, you enter "http://" plus the IP address of the Web server. When you type the address of a Web server, such as http://192.168.4.1, you tell the browser to go to 192.168.4.1 and ask for a Web page. All Web servers have a default Web page that they open unless you enter something more complex like http://192.168.4.1/status.

Most Web browsers are pretty forgiving. If you only type in 192.168.4.1, forgetting the "http:" part, they usually just add it for you.



Firmware: DD-WRT v23 SP2 (09/15/06) std
Time: 02:32:04 up 46 days, 6:10, load average: 0.00, 0.01, 0.00
WAN IP: [redacted]

System Information

Router

Router Name	TotalHomeRTnew
Router Model	Linksys WRT54G/GL/GS
LAN MAC	00:13:10:E8:26:3D
WAN MAC	00:13:10:E8:26:3E
Wireless MAC	00:13:10:E8:26:3F
WAN IP	[redacted]
LAN IP	192.168.4.1

Services

DHCP Server	Disabled
WRT-radauth	Disabled
WRT-rflow	Disabled
MAC-upd	Disabled
Samba Mount	Disabled
Sputnik Agent	Disabled

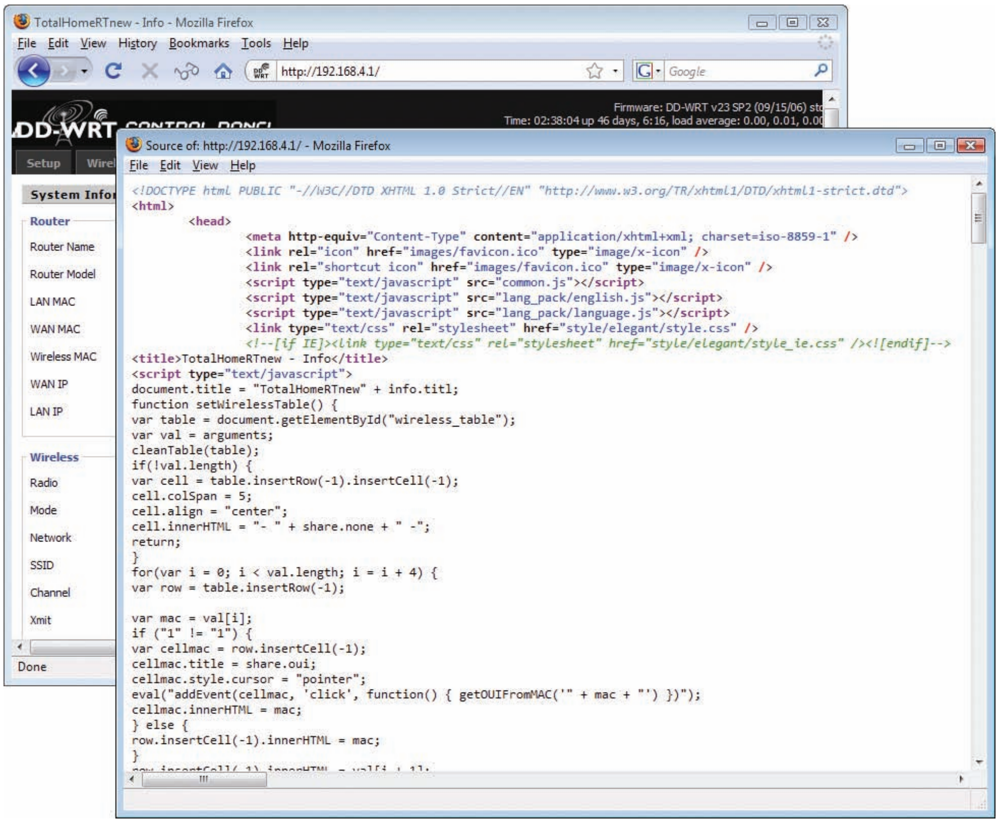
Wireless

Radio	Radio is Off
Mode	AP
Network	Disabled
SSID	TotalTemp
Channel	1
Xmit	28 mW
Rate	54 Mbps

Memory

Total Available	13.8 MB / 16.0 MB
Free	1.6 MB / 13.8 MB
Used	12.3 MB / 13.8 MB
Buffers	1.5 MB / 12.3 MB
Cached	4.8 MB / 12.3 MB
Active	3.6 MB / 12.3 MB
Inactive	2.7 MB / 12.3 MB

• **Figure 9.14** My router's Web page


TotalHomeRTnew - Info - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.4.1/
Google
Firmware: DD-WRT v23 SP2 (09/15/06) std
Time: 02:38:04 up 46 days, 6:16, load average: 0.00, 0.01, 0.00
Setup
System Info
Router
Router Name
Router Model
LAN MAC
WAN MAC
Wireless MAC
WAN IP
LAN IP
Wireless
Radio
Mode
Network
SSID
Channel
Xmit
Done
Source of: http://192.168.4.1/ - Mozilla Firefox
File Edit View Help

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
        <head>
                <meta http-equiv="Content-Type" content="application/xhtml+xml; charset=iso-8859-1" />
                <link rel="icon" href="images/favicon.ico" type="image/x-icon" />
                <link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon" />
                <script type="text/javascript" src="common.js"></script>
                <script type="text/javascript" src="lang_pack/english.js"></script>
                <script type="text/javascript" src="lang_pack/language.js"></script>
                <link type="text/css" rel="stylesheet" href="style/elegant/style.css" />
                <!--[if IE]><link type="text/css" rel="stylesheet" href="style/elegant/style_ie.css" /><![endif]-->
<title>TotalHomeRTnew - Info</title>
<script type="text/javascript">
document.title = "TotalHomeRTnew" + info.titl;
function setWirelessTable() {
var table = document.getElementById("wireless_table");
var val = arguments;
cleanTable(table);
if(!val.length) {
var cell = table.insertRow(-1).insertCell(-1);
cell.colSpan = 5;
cell.align = "center";
cell.innerHTML = "- " + share.none + " -";
return;
}
for(var i = 0; i < val.length; i = i + 4) {
var row = table.insertRow(-1);

var mac = val[i];
if ("1" != "1") {
var cellmac = row.insertCell(-1);
cellmac.title = share.oui;
cellmac.style.cursor = "pointer";
eval("addEvent(cellmac, 'click', function() { getOUIFromMAC('" + mac + "') })");
cellmac.innerHTML = mac;
} else {
row.insertCell(-1).innerHTML = mac;
}
```

• **Figure 9.15** XHTML source code

Granted, most people don't enter IP addresses into browsers, but rather enter text like www.totalsem.com or www.google.com. Memorizing text addresses is much easier than memorizing IP addresses. Web site text addresses use a naming protocol called Dynamic Naming System (DNS), which you will learn about in the next chapter. For now, just enter the IP address as shown.

HTTP

The **Hypertext Transfer Protocol (HTTP)** is the underlying protocol used by the Web, and it runs by default on TCP port 80. When you enter "http://" at the beginning of a Web server's IP address, you are identifying how messages are formatted and transmitted, requesting and responding to the transfer of HTML-formatted files. HTTP defines what actions Web servers and browsers should take in response to various commands.

HTTP has a general weakness in its handling of Web pages: it relays commands executed by users without reference to any commands previously executed. The problem with this is that Web designers continue to design more complex and truly interactive Web pages. HTTP is pretty dumb when it comes to remembering what people have done on a Web site. Luckily for Web designers everywhere, other technologies exist to help HTTP relay commands and thus support more-interactive, intelligent Web sites. These technologies include JavaScript/AJAX, server-side scripting, Adobe Flash, and cookies.

Publishing Web Pages

Once you've designed and created an HTML document, you can share it with the rest of the world. To do so, you find a Web server that will "host" the page. You most certainly can install a Web server on a computer, acquire a legitimate IP address for that computer, and host the Web page yourself. Self-hosting is a time-consuming and challenging project, though, so most people use other methods. Most Internet service providers (ISPs) provide Web servers of their own, or you can find relatively inexpensive Web hosting service companies. The price of Web hosting usually depends on the services and drive space offered. You can typically find a good Web host for around $10 a month for simple Web sites.

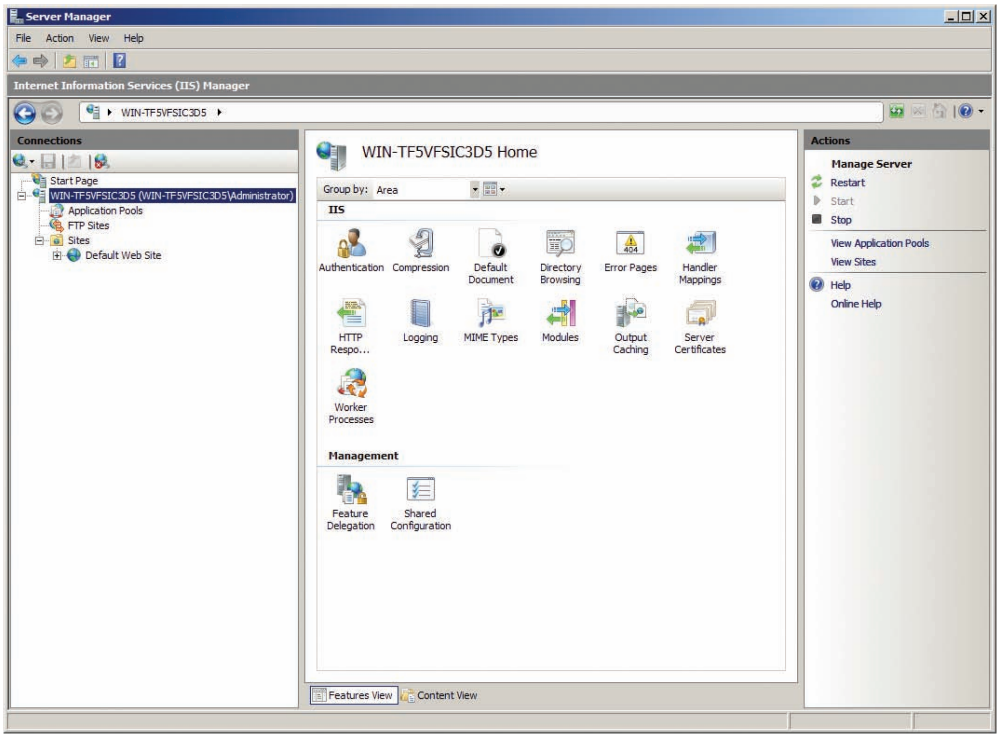
One option that has been available for a while is free Web hosting. Usually the services are not too bad, but you will run across a few limitations. Nearly all free Web hosts will insist on the right to place ads on your Web page. This is not as much of an issue if you are posting a basic blog or fan Web page, but if you do any sort of business with your Web site, this can be most annoying to your customers. The worst sort of free Web host services place pop-up ads over your Web page. Beyond annoying!

Once you have uploaded your HTML pages to your Web host, the Web server takes over. What's a Web server? I'm glad you asked!

Web Servers and Web Clients

A Web server is a computer that delivers (or *serves up*) Web pages. Web servers listen on port 80, fetching requested HTML pages and sending them to browsers. You can turn any computer into a Web server by installing server software and connecting the machine to the Internet, but you need to consider the operating system and Web server program you'll use to serve your Web site. Microsoft pushes **Internet Information Services (IIS)**, shown in Figure 9.16.

IIS enables you to set a maximum connection limit on your Web server based on available bandwidth and memory. This enables you to protect your network against an overwhelming number of requests due to a particularly popular page or a type of malicious attack called a denial of service (DoS) attack. (More on the latter in Chapter 17, "Protecting Your Network.")



• Figure 9.16 IIS in action

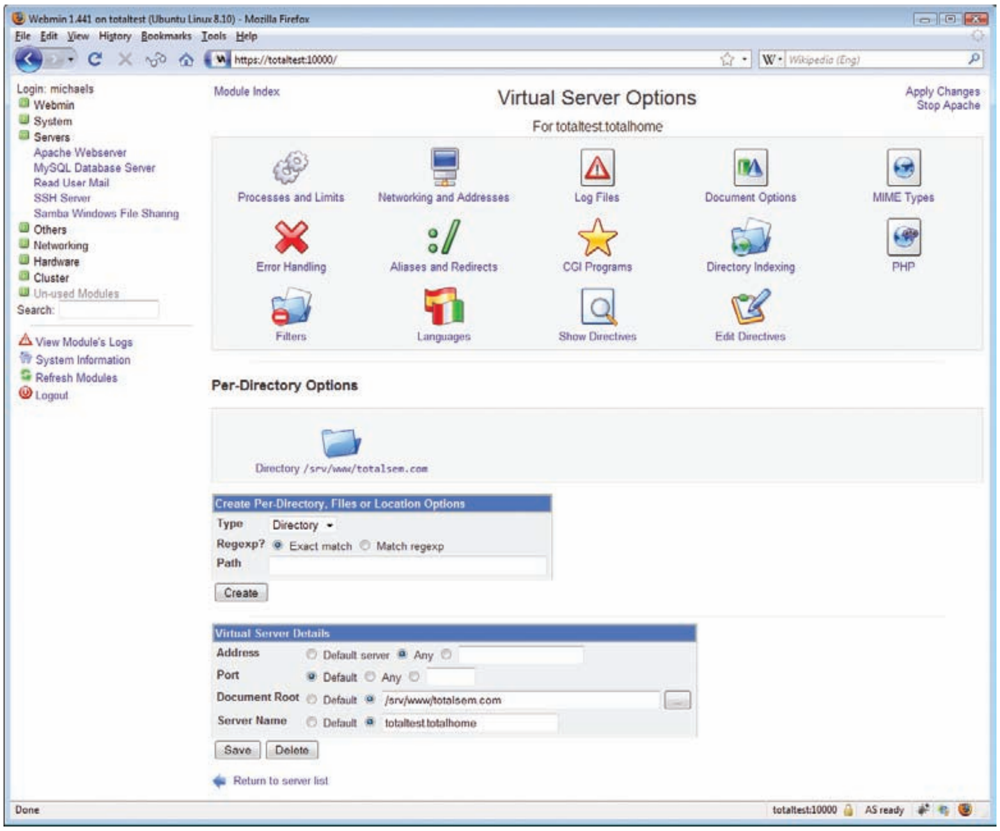
Microsoft builds an artificial 20-connection limit into Windows 2000 Professional, Windows XP, and Windows Vista so you should only run IIS on Server versions of Windows (unless you don't expect too many people to visit your Web site at one time).

UNIX/Linux-based operating systems run **Apache HTTP Server**. As of this writing, Apache serves well over 50 percent of the Web sites on the Internet. Apache is incredibly popular, runs on multiple operating systems (including Windows), and, best of all, is *free!* Apache is nothing more than an executable program and a bunch of text files, so it isn't much to look at. To ease configuration, most Web administrators use add-on graphical user interfaces (GUIs) such as Webmin that make administering Apache a breeze. Figure 9.17 illustrates the wonderful simplicity that is Webmin.

In early 2009, China released numbers for a Chinese-only Web server called QZHTTP server and, as with anything to do with China and population, the numbers for hosted sites are staggeringly large. If accurate and sustained, QZHTTP would supplant GWS as the third most popular Web server software.

IIS and Apache are by far the most common Web servers used on the Internet. In third place is Google Web Server (GWS). GWS, used only by Google's servers, has about 5 percent of the total Web server market! After those three there are literally hundreds of other Web servers, but outside of small personal Web sites you'll rarely see them.

Web clients are the programs used to surf the Web. A client program (a Web browser) reads Web pages supplied by the Web server. Most browsers handle multiple functions, from reading HTML documents to offering FTP services, and even serving as an e-mail or newsgroup reader. (You'll learn all about these functions later in the chapter.) The most popular Web browsers are Microsoft Internet Explorer, Mozilla Firefox, Apple Safari, Opera, and Google Chrome.



Create Per-Directory, Files or Location Options

Type	Directory
Regexp?	[✓ Exact match] Match regexp
Path	

Virtual Server Details

Address	Default server [✓ Any]
Port	[✓ Default] Any
Document Root	Default [✓ /srv/www/totalsem.com]
Server Name	Default [✓ totaltest.totalhome]

• **Figure 9.17** Webmin Apache module

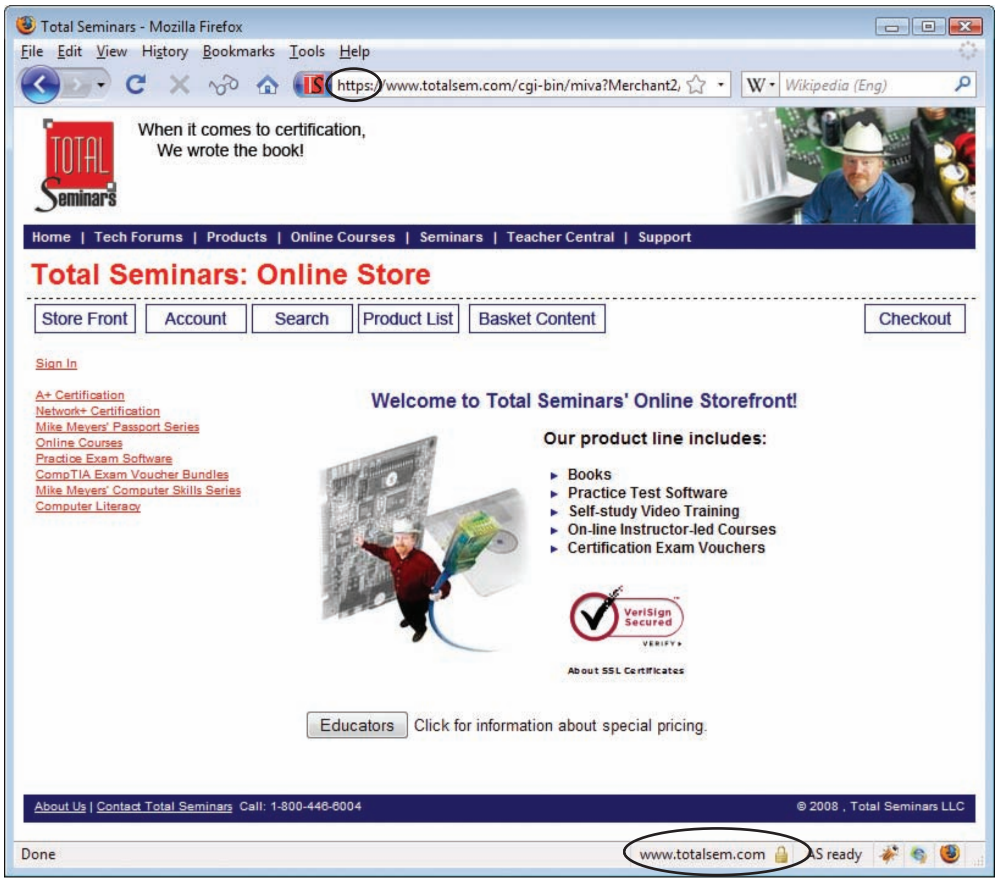
Secure Sockets Layer and HTTPS

HTTP is not a secure protocol. Any nosy person who can plug into a network can see and read the HTTP packets moving between a Web server and a Web client. Less than nice people can easily create a fake Web site to get people to think it's a legitimate Web site and steal their user names and passwords. For an Internet application to be secure it must have authentication (user names and passwords), encryption (stirring up the data so others can't read it), and nonrepudiation (verifying that you are who you really are on both sides of the conversation). While all of Chapter 11, "Securing TCP/IP," is dedicated to these concepts, we can't mention HTTP without at least touching on its secure counterpart, HTTPS. The Web has blossomed into a major economic player, requiring serious security for those who wish to do online transactions (e-commerce). In the early days of e-commerce, people feared that a simple credit card transaction on a less-than-secure Web site could transform their dreams of easy online buying into a nightmare of being robbed blind and ending up living in a refrigerator box.

Most Windows users just use Internet Explorer since it comes on Windows by default.

I can safely say that it was *never* as bad as all that. And nowadays, there are a number of safeguards on the Internet that can protect your purchases *and* your anonymity. One such safeguard is called **Secure Sockets Layer (SSL)**.

SSL is a protocol developed by Netscape for transmitting private documents over the Internet. SSL works by using a public key to encrypt sensitive data. This encrypted data is sent over an SSL connection, and then decrypted at the receiving end using a private key. All the popular Web browsers and Web servers support SSL, and many Web sites use the protocol to obtain confidential user information, such as credit card numbers. One way to tell if a site is using SSL is by looking at the Web page address. By convention, Web pages that use an SSL connection start with *https* instead of *http*. **HTTPS** stands for **Hypertext Transfer Protocol over SSL**. HTTPS uses TCP port 443. You can also look for a small key icon in the lower-right corner. Figure 9.18 shows a typical secure Web page. The https: in the address and the key icon are circled.


Total Seminars - Mozilla Firefox
File Edit View History Bookmarks Tools Help
https://www.totalsem.com/cgi-bin/miva?Merchant2,
Wikipedia (Eng)
TOTAL Seminars
When it comes to certification, We wrote the book!
Home | Tech Forums | Products | Online Courses | Seminars | Teacher Central | Support
Total Seminars: Online Store
Store Front
Account
Search
Product List
Basket Content
Checkout
Sign In
A+ Certification
Network+ Certification
Mike Meyers' Passport Series
Online Courses
Practice Exam Software
CompTIA Exam Voucher Bundles
Mike Meyers' Computer Skills Series
Computer Literacy
Welcome to Total Seminars' Online Storefront!
Our product line includes:
Books
Practice Test Software
Self-study Video Training
On-line Instructor-led Courses
Certification Exam Vouchers
VeriSign Secured VERIFY
About SSL Certificates
Educators
Click for information about special pricing.
About Us | Contact Total Seminars Call: 1-800-446-6004
© 2008 , Total Seminars LLC
Done
www.totalsem.com
S ready

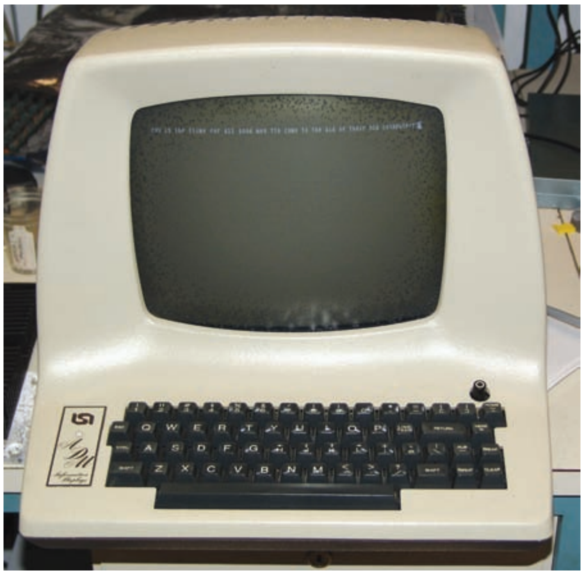
• Figure 9.18 Secure Web page

HTTP enables you to access the Web, but HTTPS gets you there securely. HTTPS uses either SSL or TLS to provide the security.

The last few years have seen SSL replaced with the more powerful *Transport Layer Security* (*TLS*). Your secure Web page still looks the same as with SSL, so only the folks setting this up really care. Just make sure you know that SSL and TLS are functionally the same with Web pages. Read Chapter 11 for more details on SSL and TLS.

Telnet

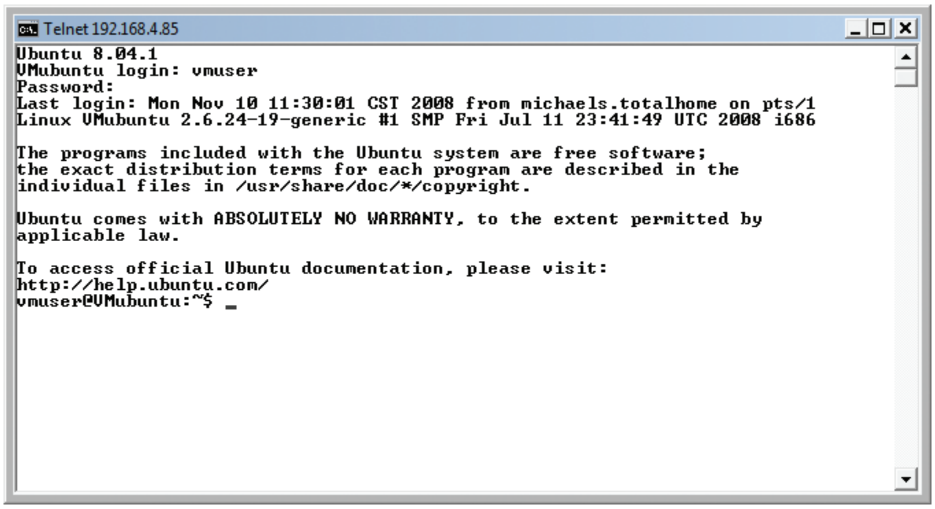
Roughly one billion years ago there was no such thing as the Internet or even networks . . . Well, maybe it was only about 40 years ago, but as far as nerds like me are concerned, a world before the Internet was filled with brontosauruses and palm fronds. The only computers were huge monsters called mainframes and to access them required a dumb terminal like the one shown in Figure 9.19.



• Figure 9.19 Dumb terminal (photo courtesy of DVQ)

Operating systems didn't have windows and pretty icons. The interface to the mainframe was a command line, but it worked just fine for the time. The cavemen who first lifted their heads up from the computer ooze known as mainframes said to themselves, "Wouldn't it be great if we could access each other's computers from the comfort of our own caves?" That was what started the entire concept of a network. Back then the idea of sharing folders or printers or Web pages wasn't even yet considered. The entire motivation for networking was so people could sit at their dumb terminals and, instead of accessing only their local mainframes, access totally different mainframes. The protocol to do this was called the Telnet Protocol or simply **Telnet**.

Even though PCs have replaced mainframes for the most part, Telnet still exists as the way to connect remotely to another computer via the command line (Figure 9.20). Telnet runs on TCP port 23, enabling you to connect to a Telnet server and run commands on that server as if you were sitting right in front of it.



```
Ubuntu 8.04.1
VMubuntu login: vmuser
Password:
Last login: Mon Nov 10 11:30:01 CST 2008 from michaels.totalhome on pts/1
Linux VMubuntu 2.6.24-19-generic #1 SMP Fri Jul 11 23:41:49 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
vmuser@VMubuntu:~$ _
```

• Figure 9.20 Telnet client

This way, you can remotely administer a server and communicate with other servers on your network. As you can imagine, this is sort of risky. If you can remotely control a computer, what is to stop others from doing the same? Thankfully, Telnet does not just allow *anyone* to log on and wreak havoc with your network. You must enter a user name and password to access a Telnet server. Unfortunately, Telnet does not have any form of encryption. If someone intercepted the conversation between a Telnet client and

Telnet server, he or she would see all of the commands you type as well as the results from the Telnet server. As a result, Telnet is rarely used on the Internet and has been replaced with **Secure Shell (SSH)**, a terminal emulation program that looks exactly like Telnet but encrypts the data.

Even though Telnet is less common than SSH, Telnet is a popular second option to connect to almost anything on a trusted TCP/IP network. Most routers have Telnet access capability (although many router admins turn it off for security). Almost every operating system has a built-in Telnet client and most operating systems—though not all Windows operating systems—come with built-in Telnet servers. Almost every type of server application has some way for you to access it with Telnet. It was once quite common, for example, to administer Apache-based Web servers through Telnet.

Telnet only enables command-line remote access, not from a GUI. If you want to access another computer's desktop remotely, you need another type of program.

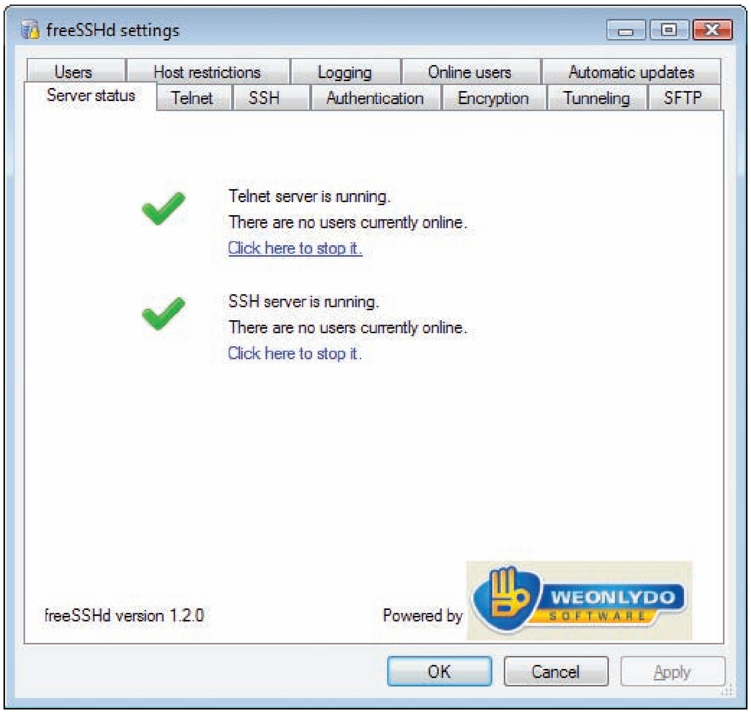
By default, Windows Server 2008 does not have Telnet Server or Telnet Client installed, though you can install both through Server Manager | Features | Add Features. Windows Vista likewise does not have Telnet Client installed. It is available for quick installation in Programs and Features.

Telnet Servers and Clients

The oldest Telnet server, found on UNIX and Linux systems, is the venerable telnetd. Like most UNIX/Linux servers, telnetd isn't much to look at, so let's move over to the Windows world. Since Windows itself doesn't have a built-in Telnet server, we lean toward third-party server programs like freeSSHd (Figure 9.21). Note the name—freeSSHd, not "freeTelnet." As Telnet fades away and SSH becomes more dominant, it's hard to find a Telnet-only server these days. All of the popular Telnet servers are also SSH servers.

A Telnet client is the computer from which you log onto the remote server. Most operating systems have a built-in Telnet client that you run from a command prompt. Figure 9.22 shows the Telnet client built into

Some versions of Windows Server came with a rather poor Telnet server that only allowed a maximum of two client connections.



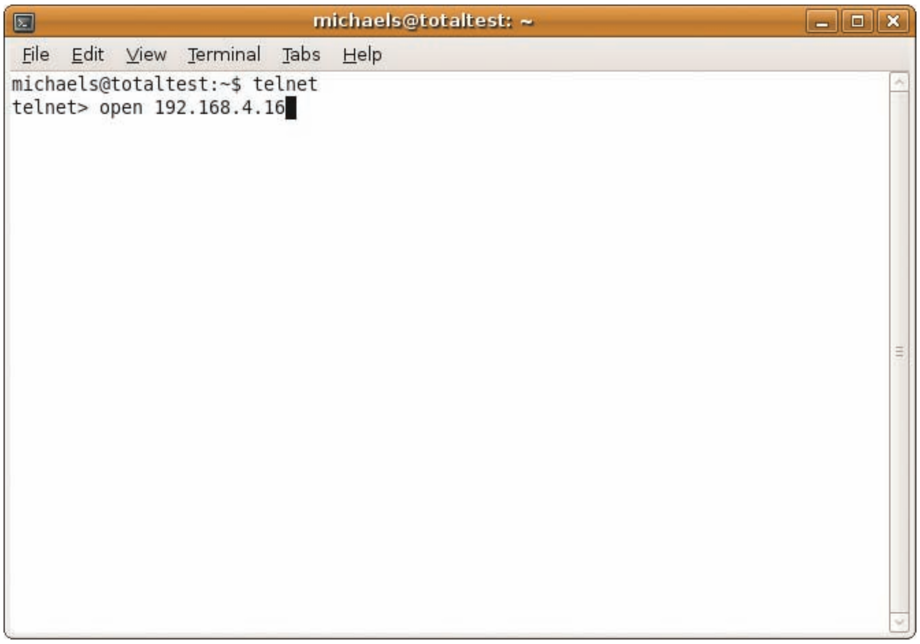
Telnet server is running.
There are no users currently online.
Click here to stop it.

SSH server is running.
There are no users currently online.
Click here to stop it.

freeSSHd version 1.2.0

Powered by WEONLYDO SOFTWARE

• **Figure 9.21** freeSSHd



```
michaels@totaltest:~$ telnet
telnet> open 192.168.4.16
```

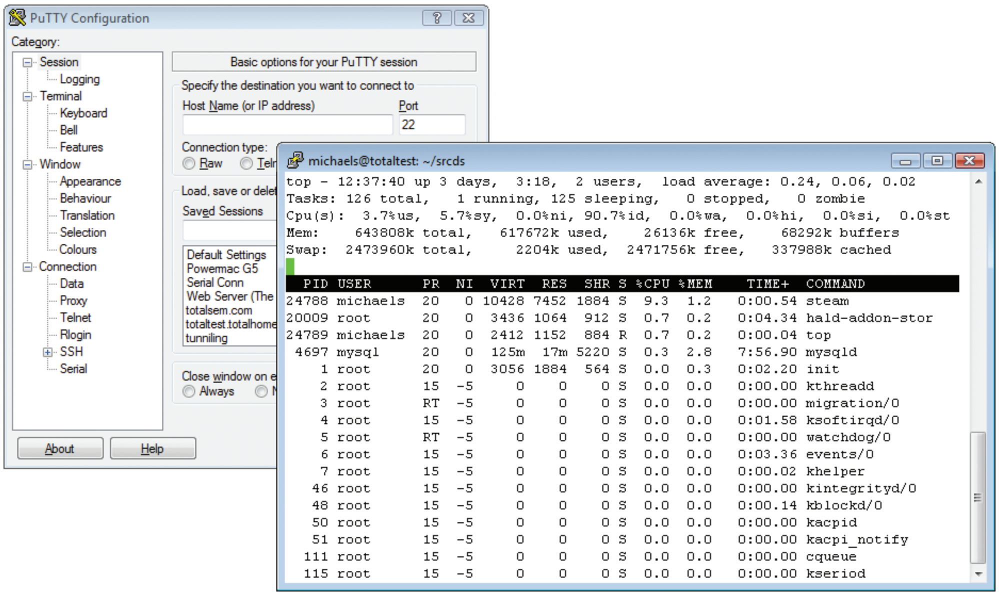
• **Figure 9.22** Ubuntu Telnet

Ubuntu Linux. Just open a terminal window and type `telnet` and the IP address of the Telnet server.

Command-prompt Telnet clients lack a number of handy features. They can't, for example, remember the IP addresses, user names, or passwords for Telnet servers, so every time you use Telnet, you have to enter all that information again. Third-party Telnet clients, such as the very popular PuTTY you saw in Chapter 8, store all this information and much more (Figure 9.23).

Configuring a Telnet Client

When you configure a Telnet client, you must provide the host name, your user login name, and the password. As I mentioned previously, you must have permission to access the server to use Telnet. A *host name* is the name or IP address of the computer to which you want to connect. For instance, you might connect to a Web server with the host name websrv.mhteched.com. The user *login name* you give Telnet should be the same login name you'd use if you logged into the server at its location. Some computers, usually university libraries with online catalogs, have open systems that enable you to log in with Telnet. These sites will



```
top - 12:37:40 up 3 days,  3:18,  2 users,  load average: 0.24, 0.06, 0.02
Tasks: 126 total,   1 running, 125 sleeping,   0 stopped,   0 zombie
Cpu(s):  3.7%us,  5.7%sy,  0.0%ni, 90.7%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    643808k total,   617672k used,    26136k free,    68292k buffers
Swap:  2473960k total,     2204k used,  2471756k free,   337988k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
24788 michaels  20   0 10428 7452 1884 S  9.3  1.2   0:00.54 steam
20009 root      20   0  3436 1064  912 S  0.7  0.2   0:04.34 hald-addon-stor
24789 michaels  20   0  2412 1152  884 R  0.7  0.2   0:00.04 top
 4697 mysql     20   0  125m  17m 5220 S  0.3  2.8   7:56.90 mysqld
    1 root      20   0  3056 1884  564 S  0.0  0.3   0:02.20 init
    2 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 migration/0
    4 root      15  -5     0    0    0 S  0.0  0.0   0:01.58 ksoftirqd/0
    5 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    6 root      15  -5     0    0    0 S  0.0  0.0   0:03.36 events/0
    7 root      15  -5     0    0    0 S  0.0  0.0   0:00.02 khelper
   46 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kintegrityd/0
   48 root      15  -5     0    0    0 S  0.0  0.0   0:00.14 kblockd/0
   50 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kacpid
   51 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kacpi_notify
  111 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 cqueue
  115 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kseriod
```

• **Figure 9.23** PuTTY

either display a banner before the login prompt that tells you what login name to use, or they'll require no login name at all. As with the login name, you use the same password for a Telnet login that you'd use to log into the server directly. It's that simple. Computers with open access will either tell you what password to use when they tell you what login name to use, or they'll require no login name/password at all.

Telnet enables you to control a remote computer from a local computer over a network.

Rlogin, RSH, and RCP

The CompTIA Network+ exam tests you on **rlogin**, **RSH**, and **RCP**. These are three old-school programs in the UNIX world. The *R* stands for *remote* and, like Telnet and SSH, these programs provide remote access and control of servers. Also like Telnet, they do not encrypt data and thus should not be used across the Internet. Here is a quick breakdown of the suite:

- **Rlogin** Remote login works very similarly to Telnet. You simply run the program with the host name of the server and you can connect and run commands just like with Telnet. It has one very nice advantage over Telnet in that it can be configured to log in automatically without the need to enter a user name and password. Rlogin works over TCP port 513.
- **RSH** Remote Shell allows you to send single commands to the remote server. Whereas rlogin is designed to be used interactively, RSH can be easily integrated into a script. RSH runs over TCP port 514 by default.
- **RCP** Remote Copy provides the capability to copy files to and from the remote server without the need to resort to FTP or NFS (Network File System, the UNIX form of folder sharing). RCP can also be used in scripts and shares TCP port 514 with RSH.

SSH and the Death of Telnet

Telnet has seen long and heavy use in the TCP world from the earliest days of the Internet, but it suffers from lack of any security. Telnet passwords as well as data are transmitted in cleartext and are thus easily hacked. To that end, SSH has now replaced Telnet for any serious terminal emulation. In terms of what it does, SSH is extremely similar to Telnet in that it creates a terminal connection to a remote host. Every aspect of SSH, however, including both login and data transmittal, is encrypted. SSH also uses TCP port 22 instead of Telnet's port 23.

SSH enables you to control a remote computer from a local computer over a network, just like Telnet. Unlike Telnet, SSH enables you to do it securely!

E-mail

Electronic mail (e-mail) has been a major part of the Internet revolution, and not just because it has streamlined the junk mail industry. E-mail provides an extremely quick way for people to communicate with one another, letting you send messages and attachments (like documents and pictures) over the Internet. It's normally offered as a free service by ISPs. Most e-mail client programs provide a rudimentary text editor for composing messages, but many can be configured to let you edit your messages using more sophisticated editors.

E-mail consists of e-mail clients and e-mail servers. When a message is sent to your e-mail address, it is normally stored in an electronic mailbox on your e-mail server until you tell the e-mail client to download the message. Most e-mail client programs can be configured to signal you in some way when a new message has arrived or to automatically download e-mails as they come to you. Once you read an e-mail message, you can archive it, forward it, print it, or delete it. Most e-mail programs are configured to automatically delete messages from the e-mail server when you download them to your local machine, but you can usually change this configuration option to suit your circumstances.

E-mail programs use a number of application-level protocols to send and receive information. Specifically, the e-mail you find on the Internet uses SMTP to send e-mail, and either POP3 or IMAP4 to receive e-mail.

SMTP, POP3, and IMAP4, Oh My!

The following is a list of the different protocols that the Internet uses to transfer and receive mail:

SMTP The **Simple Mail Transfer Protocol (SMTP)** is used to send e-mail. SMTP travels over TCP port 25, and is used by clients to send messages.

POP3 **Post Office Protocol version 3 (POP3)** is one of the two protocols that receive e-mail from SMTP servers. POP3 uses TCP port 110. Most e-mail clients use this protocol, although some use IMAP4.

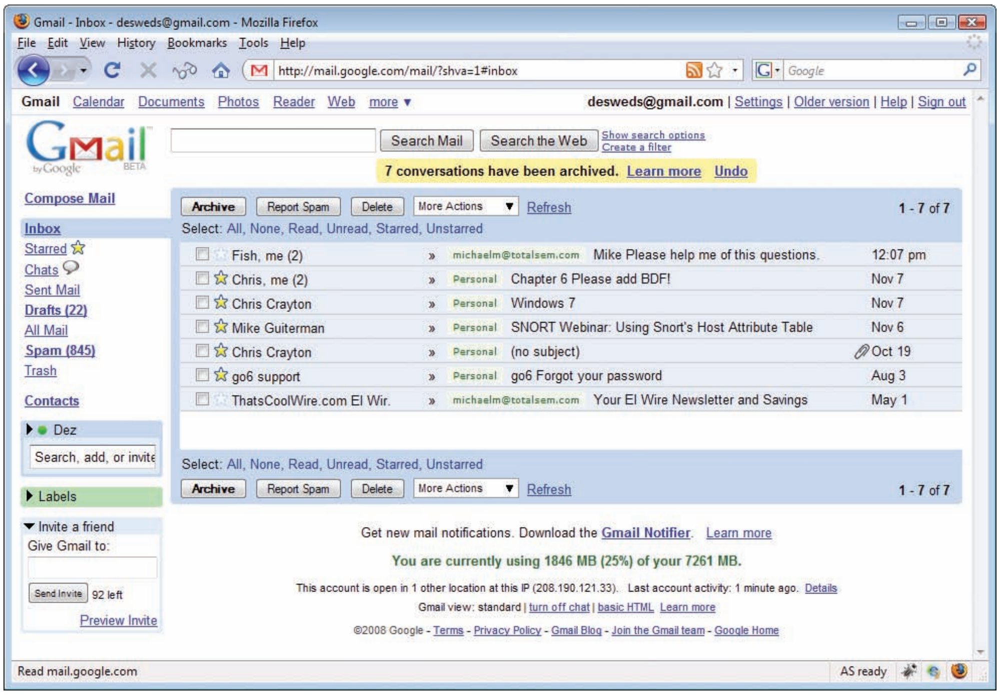
IMAP4 **Internet Message Access Protocol version 4 (IMAP4)** is an alternative to POP3. Like POP3, IMAP4 retrieves e-mail from an e-mail server. IMAP4 uses TCP port 143 and supports some features that are not supported in POP3. For example, IMAP4 enables you to search through messages on the mail server to find specific keywords, and select the messages you want to download onto your machine. IMAP4 also supports the concept of folders that you can place on the IMAP4 server to organize your e-mail. Some POP3 e-mail clients have folders, but that's not a part of POP3, just a nice feature added to the client.

Alternatives to SMTP, POP3, and IMAP4

While SMTP and POP3 or IMAP4 are by far the most common and most traditional tools for doing e-mail, two other options have wide popularity: Web-based e-mail and proprietary solutions. Web-based mail, as the name implies, requires a Web interface. From a Web browser, you simply surf to the Web-mail server, log in, and access your e-mail. The cool part is that you can do it from anywhere in the world where you find a Web browser and an Internet hookup! You get the benefit of e-mail without even needing to own a computer. Some of the more popular Web-based services are Google's Gmail (Figure 9.24), Microsoft's MSN Hotmail, and Yahoo! Mail.

The key benefits of Web-based e-mail services are as follows:

- You can access your e-mail from anywhere.
- They're free.
- They're handy for throw-away accounts (like when you're required to give an e-mail address to download something, but you know you're going to get spammed if you do).



7 conversations have been archived. Learn more Undo

Fish, me (2)	michaelm@totalsem.com	Mike Please help me of this questions.	12:07 pm
Chris, me (2)	Personal	Chapter 6 Please add BDF!	Nov 7
Chris Crayton	Personal	Windows 7	Nov 7
Mike Guiterman	Personal	SNORT Webinar: Using Snort's Host Attribute Table	Nov 6
Chris Crayton	Personal	(no subject)	Oct 19
go6 support	Personal	go6 Forgot your password	Aug 3
ThatsCoolWire.com El Wir.	michaelm@totalsem.com	Your El Wire Newsletter and Savings	May 1

Get new mail notifications. Download the Gmail Notifier. Learn more

You are currently using 1846 MB (25%) of your 7261 MB.

This account is open in 1 other location at this IP (208.190.121.33). Last account activity: 1 minute ago. Details

Gmail view: standard | turn off chat | basic HTML Learn more

©2008 Google - Terms - Privacy Policy - Gmail Blog - Join the Gmail team - Google Home

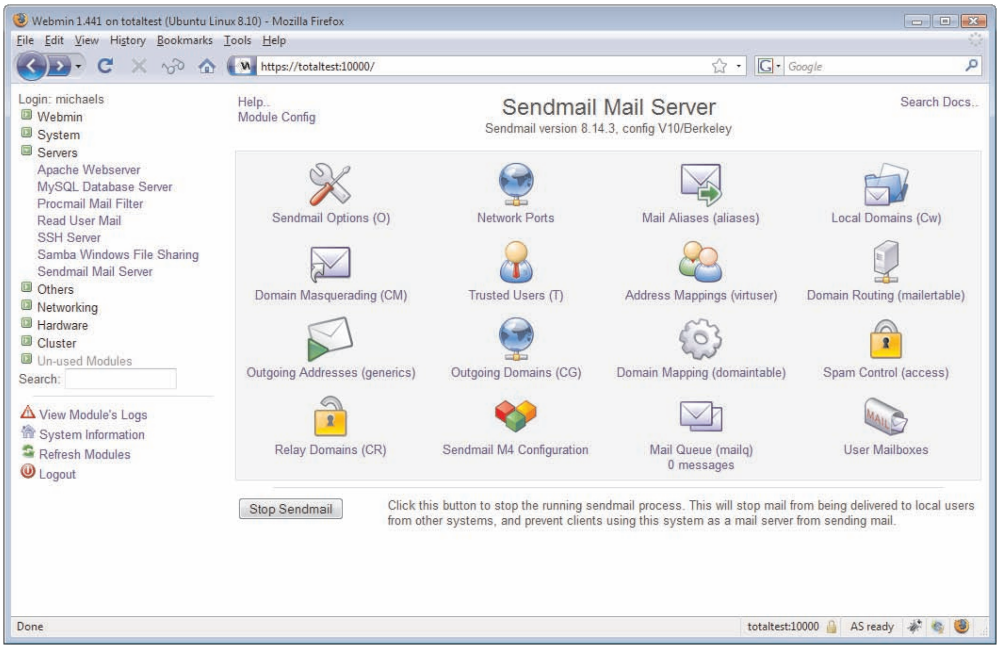
• **Figure 9.24** Gmail in action

Many traditional SMTP/POP/IMAP accounts also provide Web interfaces, but you should not confuse them with Web mail services. Web-based e-mail services are only available through the Web (although some will also give you SMTP/POP access).

E-mail Servers

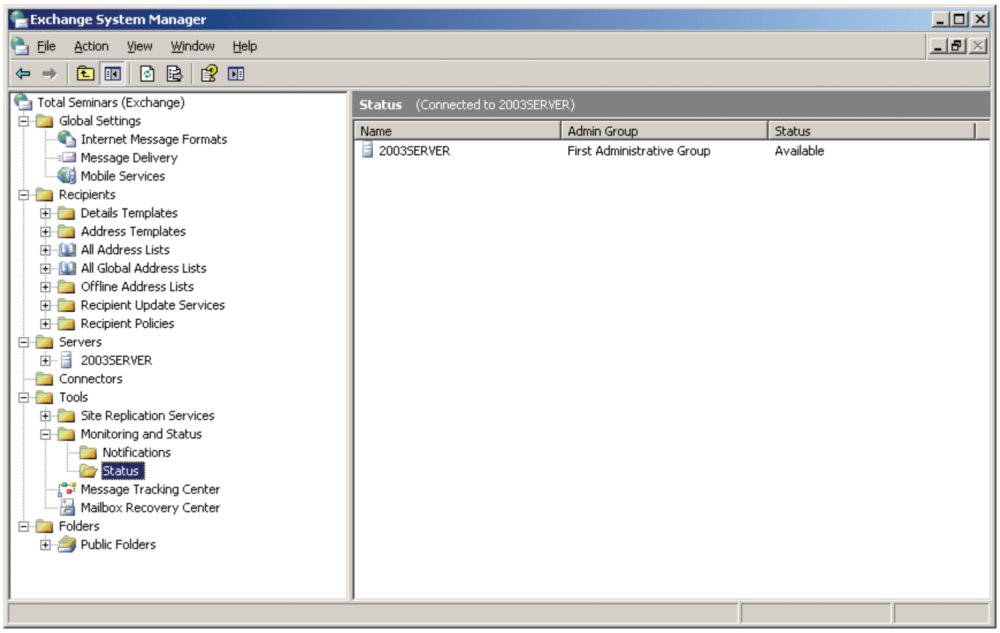
The e-mail server world is much more fragmented than the Web server world. The current leader is **sendmail** used on Linux and UNIX operating systems. Like Apache, sendmail doesn't really have an interface, but there are many different third-party interfaces to help configure sendmail, such as Webmin shown in Figure 9.25.

Sendmail controls about 20 percent of all e-mail servers, but only uses SMTP. You must run a POP3 or IMAP4



Click this button to stop the running sendmail process. This will stop mail from being delivered to local users from other systems, and prevent clients using this system as a mail server from sending mail.

• **Figure 9.25** Webmin with the sendmail module



Name	Admin Group	Status
2003SERVER	First Administrative Group	Available

• **Figure 9.26** Microsoft Exchange Server

server program to support e-mail clients. Programs like Eudora's Qpopper handle sending mail to POP3 e-mail clients. Microsoft of course has its own e-mail server, Microsoft Exchange Server, and like IIS it only runs on Windows (Figure 9.26). Exchange Server is both an SMTP and a POP3 server in one package.

E-mail servers accept incoming mail and sort out the mail for recipients into individual storage areas mailboxes. These **mailboxes** are special separate holding areas for each user's e-mail. An e-mail server works much like a post office, sorting and arranging incoming messages, and kicking back those messages that have no known recipient.

Perhaps one reason e-mail servers are so little understood is that they're difficult to manage. E-mail servers store user lists, user rights, and messages, and are constantly involved in Internet traffic and resources. Setting up and administering an e-mail server takes a lot of planning, although it's getting easier. Most e-mail server software runs in a GUI, but even the command-line-based interface of e-mail servers is becoming more intuitive.

E-mail Client An **e-mail client** is a program that runs on a computer and enables you to send, receive, and organize e-mail. The e-mail client program communicates with the SMTP e-mail server to send mail and communicates with the IMAP or POP e-mail server to download the messages from the e-mail server to the client computer. There are hundreds of e-mail programs, some of the most popular of which are Microsoft's Windows Mail (Figure 9.27), Microsoft Outlook, Mozilla Thunderbird, and Qualcomm's Eudora.

Try This

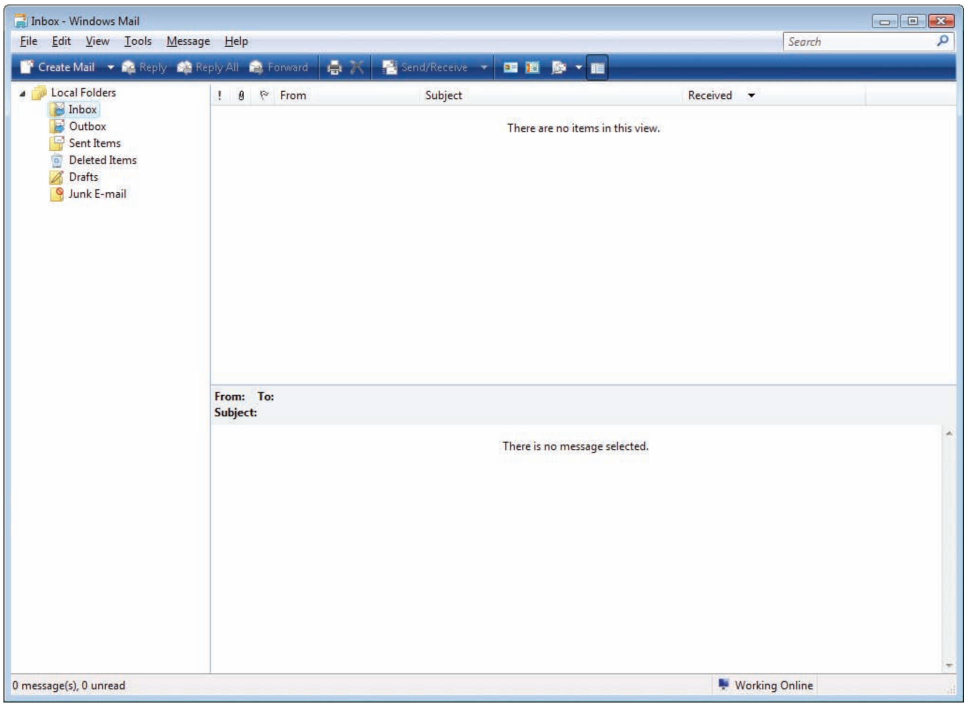
Connecting with Telnet and SMTP

Try using the TELNET command to talk to an SMTP server. Try using TELNET to send an e-mail message. Note that normally you use port 23 with Telnet, but to talk to an SMTP server you use port 25.

```
C:\>telnet mail.totalsem.com 25
HELO totalsem.com
Hello totalsem.com, I am glad to meet you
MAIL FROM:<bob@example.org>
Data Ok
RCPT TO:<alice@example.com>
Data  Ok
DATA
End data with <CR><LF>.<CR><LF>
From: "Bob Example" <bob@example.org>
To: Alice Example <alice@example.com>
Subject: Test message
Hello World.
Data OK queued as 12345
QUIT
Closing Connection
```

Configuring an E-mail Client

Configuring a client is an easy matter. Your mail administrator will give you the server's addresses and your mailbox's user name and password. You need to enter the POP3 or IMAP4 server's IP address and the SMTP server's address to the e-mail client (Figure 9.28). Every e-mail client has a different way to add the server IP addresses, so you may have to poke around but you'll find the option there somewhere! In many cases this may be the same IP address for both the incoming and outgoing servers—the folks administering the mail servers will tell you. Besides the e-mail server addresses, you must also enter the user name and password of the e-mail account the client will be managing.



There are no items in this view.

There is no message selected.

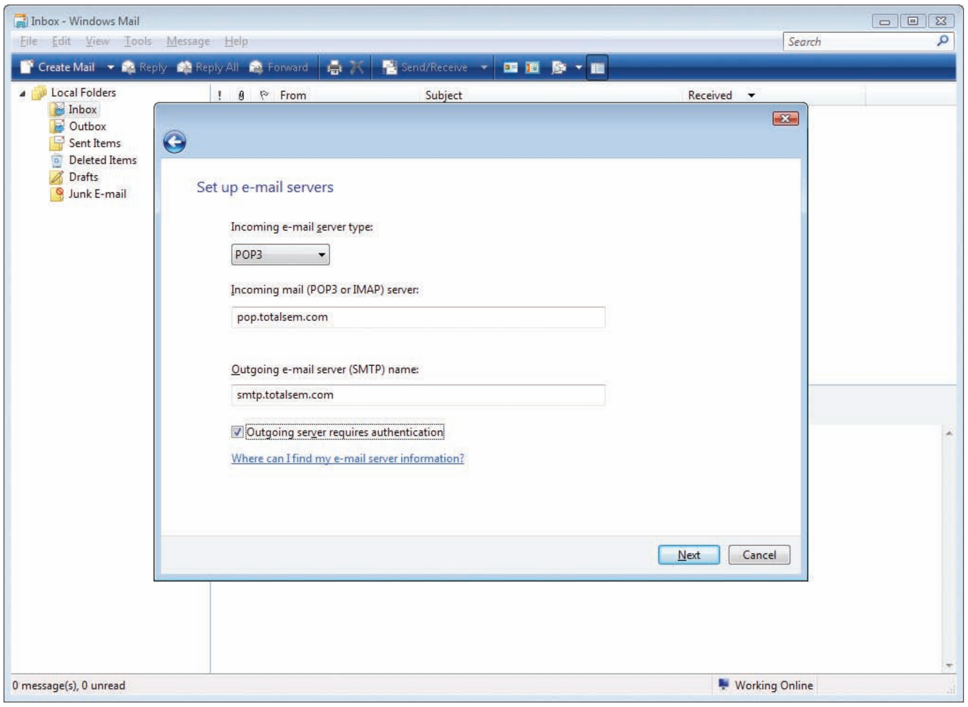
0 message(s), 0 unread

Figure 9.27 Windows Mail

FTP

File Transfer Protocol (FTP) is the original protocol used on the Internet for transferring files. Although HTTP can be used to transfer files as well, the transfer is often not as reliable or as fast as with FTP. In addition, FTP can do the transfer with security and data integrity. FTP uses TCP ports 21 and 20 by default.

FTP sites are either anonymous sites, meaning that anyone can log on, or secured sites, meaning that you must have a user name and password to be able to access the site and transfer files. A single FTP site can offer both anonymous access and protected access, but you'll see different resources depending on which way you log in.



Set up e-mail servers

Incoming e-mail server type: POP3

Incoming mail (POP3 or IMAP) server: pop.totalsem.com

Outgoing e-mail server (SMTP) name: smtp.totalsem.com

[✓] Outgoing server requires authentication

0 message(s), 0 unread

Figure 9.28 Entering server information in Windows Mail

FTP Servers and FTP Clients

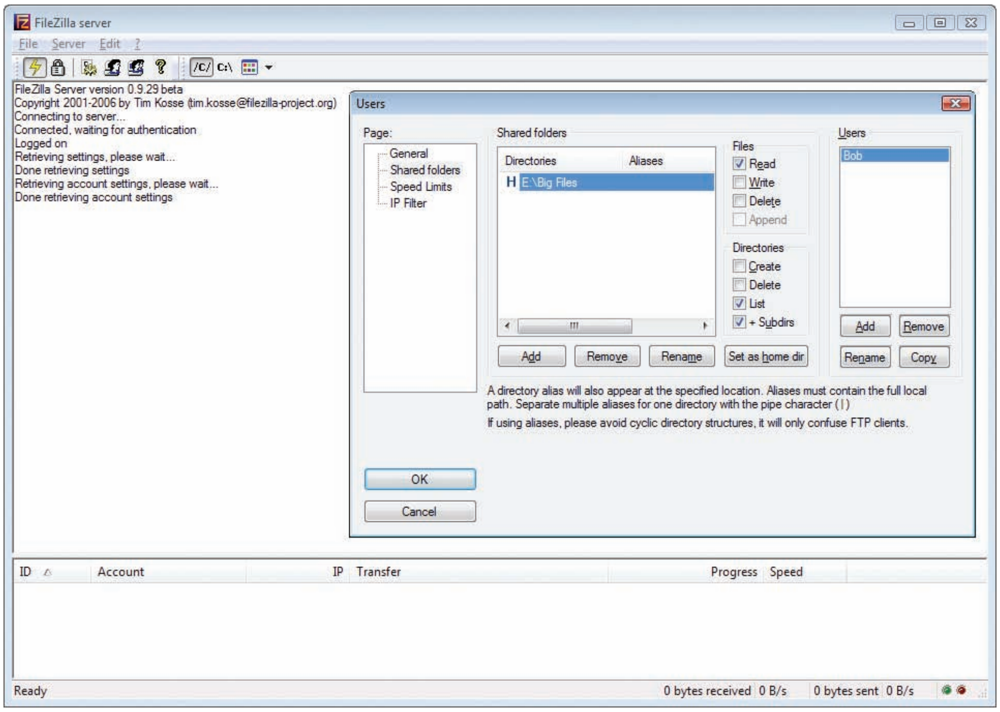
The FTP server does all the real work of storing the files, accepting incoming connections and verifying user names and passwords, and transferring the files. The client logs onto the FTP server (either from a Web site, a command line, or a special FTP application) and downloads the requested files onto the local hard drive.

Most Web servers are also FTP servers. These bundled versions of FTP servers are robust, but do not provide all the options one might want.

FTP Servers We don't set up servers for Internet applications nearly as often as we set up clients. I've set up only a few Web servers over the years whereas I've set up thousands of Web browsers. FTP servers are the one exception, as we nerds like to exchange files. If you have a file you wish to share with a lot of people (but not the entire Internet), there are few options better than whipping up a quick FTP server. Most versions of Linux/UNIX have built-in FTP servers, but many third-party applications offer better solutions. One of the best, especially for those "let me put up an FTP server so you guys can get a copy" type of situations is Mozilla's FileZilla Server (Figure 9.29).

FTP is not very secure so you don't want to use FTP for sensitive data. But you can add user names and passwords to prevent all but the most serious hackers from accessing your FTP server. I avoid using the anonymous login unless I'm not worried about who gets a hold of the files I'm offering on my FTP server.

Another thing to check when deciding on an FTP server setup is the number of clients you want to support. Most anonymous FTP sites limit the number of users who may download at any one time to around 500. This protects you from a sudden influx of users flooding your server and eating up all your Internet bandwidth.



```
FileZilla Server version 0.9.29 beta
Copyright 2001-2006 by Tim Kosse (tim.kosse@filezilla-project.org)
Connecting to server...
Connected, waiting for authentication
Logged on
Retrieving settings, please wait...
Done retrieving settings
Retrieving account settings, please wait...
Done retrieving account settings
```

Users — Page: Shared folders

Shared folders: H E:\Big Files

Files: [✓ Read] Write Delete Append

Directories: Create Delete [✓ List] [✓ + Subdirs]

Users: Bob

A directory alias will also appear at the specified location. Aliases must contain the full local path. Separate multiple aliases for one directory with the pipe character (|)

If using aliases, please avoid cyclic directory structures, it will only confuse FTP clients.

Ready 0 bytes received 0 B/s 0 bytes sent 0 B/s

• **Figure 9.29** FileZilla Server

Try This!

Doing FTP

Never done FTP? Do a Web search for **"Public FTP servers"** and try accessing them from your Web browser. Then download a dedicated FTP client and try again! There are thousands of public FTP servers out there.

Every operating system has a command-line FTP client. I avoid using them unless I have no other choice, because they lack important features like the ability to save FTP connections to use again later.

FTP Clients FTP clients, as noted before, can access an FTP server through a Web site, a command line, or a special FTP application. Usually special FTP applications offer the most choices for accessing and using an FTP site.

You have a lot of choices when it comes to FTP clients. For starters, most Web browsers handle FTP as well as HTTP, although they lack a few features. For example, Firefox only supports an anonymous login. To use your Web browser as an FTP client, type **ftp://** followed by the IP address of the FTP server (Figure 9.30).

The best way to use FTP is to use a dedicated FTP client. There are so many good ones that I find myself using a different one all the time. FileZilla comes in a client version, but these days I'm using an add-on to Firefox called FireFTP (Figure 9.31).

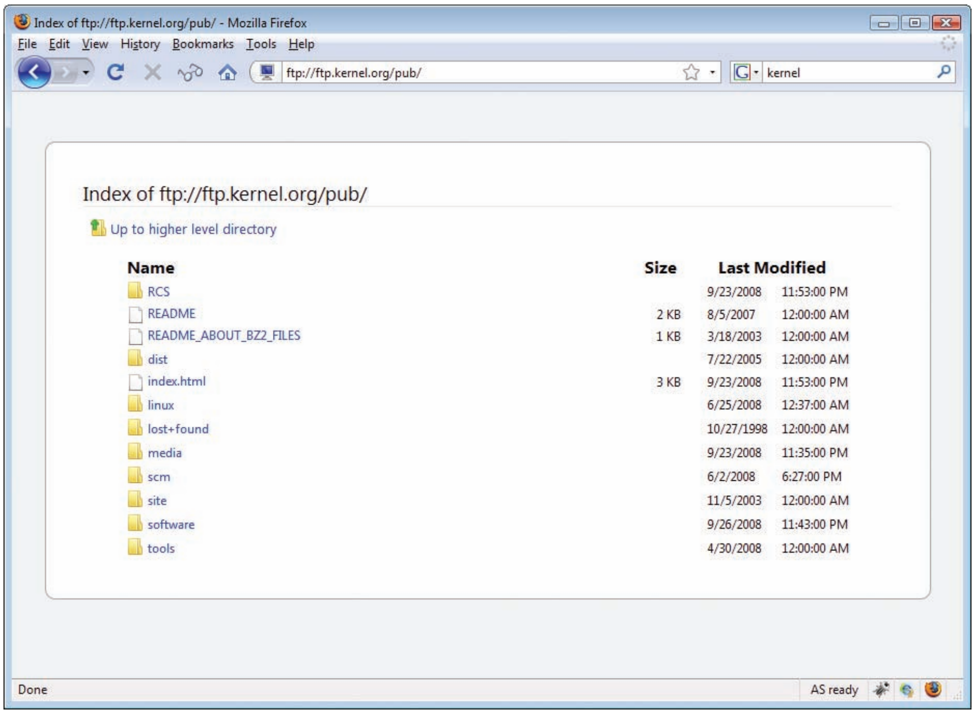
Tech Tip

Firefox Add-ons

Firefox enables programmers to create add-ons, small programs that extend the capabilities of the browser with some pretty impressive results. Are you unfamiliar with Firefox add-ons? Start Firefox. Click ***Tools/Add-ons*** *and a whole new world will open for you. A couple of my favorites are Mouse Gestures—where you can flick the mouse left or right to navigate through windows and Web sites you've visited—and Speed Dial—quick access to your favorite sites.*

Passive vs. Active FTP

FTP has two ways to transfer data, called active and passive FTP. Traditionally, FTP uses the active process—let's see how this works. Remember that FTP uses TCP ports 20 and 21? Well, when your client sends an FTP request, it goes out on port 21. However, when your FTP server responds, it

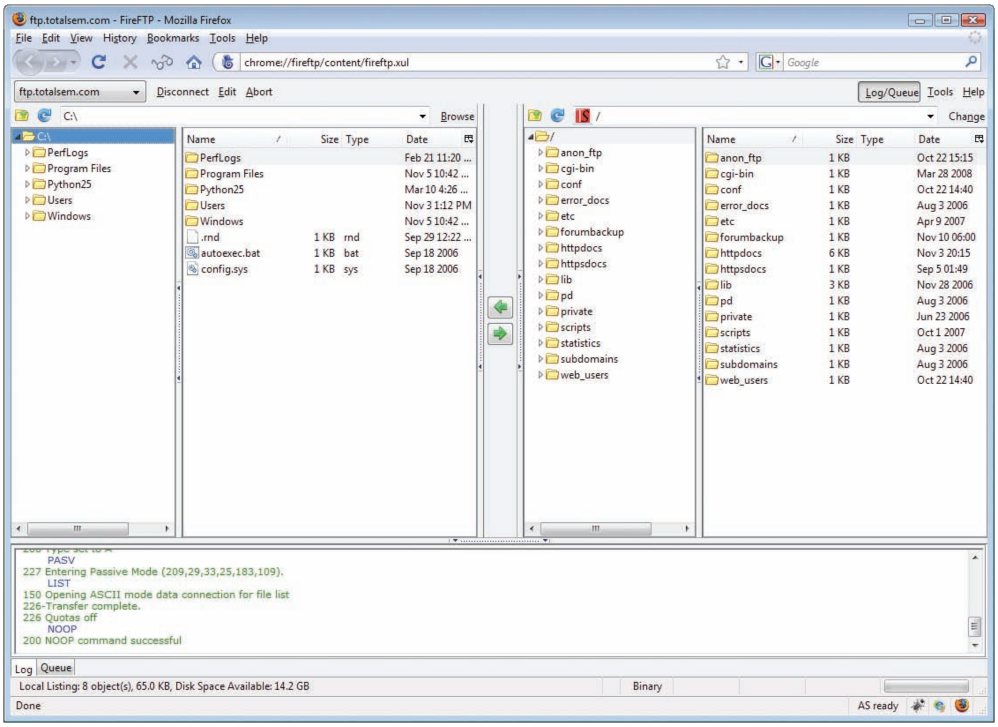

Index of ftp://ftp.kernel.org/pub/ - Mozilla Firefox
File Edit View History Bookmarks Tools Help
ftp://ftp.kernel.org/pub/
kernel
Done
AS ready

Index of ftp://ftp.kernel.org/pub/

Up to higher level directory

Name	Size	Last Modified	
RCS		9/23/2008	11:53:00 PM
README	2 KB	8/5/2007	12:00:00 AM
README_ABOUT_BZ2_FILES	1 KB	3/18/2003	12:00:00 AM
dist		7/22/2005	12:00:00 AM
index.html	3 KB	9/23/2008	11:53:00 PM
linux		6/25/2008	12:37:00 AM
lost+found		10/27/1998	12:00:00 AM
media		9/23/2008	11:35:00 PM
scm		6/2/2008	6:27:00 PM
site		11/5/2003	12:00:00 AM
software		9/26/2008	11:43:00 PM
tools		4/30/2008	12:00:00 AM

• **Figure 9.30** FTP in a Web browser



Name	Size	Type	Date
PerfLogs			Feb 21 11:20 ...
Program Files			Nov 5 10:42 ...
Python25			Mar 10 4:26 ...
Users			Nov 3 1:12 PM
Windows			Nov 5 10:42 ...
.rnd	1 KB	rnd	Sep 29 12:22 ...
autoexec.bat	1 KB	bat	Sep 18 2006
config.sys	1 KB	sys	Sep 18 2006

Name	Size	Type	Date
anon_ftp	1 KB		Oct 22 15:15
cgi-bin	1 KB		Mar 28 2008
conf	1 KB		Oct 22 14:40
error_docs	1 KB		Aug 3 2006
etc	1 KB		Apr 9 2007
forumbackup	1 KB		Nov 10 06:00
httpdocs	6 KB		Nov 3 20:15
httpsdocs	1 KB		Sep 5 01:49
lib	3 KB		Nov 28 2006
pd	1 KB		Aug 3 2006
private	1 KB		Jun 23 2006
scripts	1 KB		Oct 1 2007
statistics	1 KB		Aug 3 2006
subdomains	1 KB		Aug 3 2006
web_users	1 KB		Oct 22 14:40

```
    PASV
227 Entering Passive Mode (209,29,33,25,183,109).
    LIST
150 Opening ASCII mode data connection for file list
226-Transfer complete.
226 Quotas off
    NOOP
200 NOOP command successful
```

Local Listing: 8 object(s), 65.0 KB, Disk Space Available: 14.2 GB

• **Figure 9.31** Author's FireFTP hard at work

sends the data back using an ephemeral destination port and port 20 as a source port.

Active FTP works great unless your client uses NAT. Since your client didn't initiate the incoming port 20, your NAT router has no idea where to send this incoming packet. No problem! Good FTP clients all support passive FTP. With passive FTP, the server doesn't use port 20. Instead, it sends back the packet using the ephemeral source port used by the client as the destination port and uses another ephemeral port for the source port. This way the NAT router knows where to send the packet.

The only trick to passive FTP is that the client needs to expect this other incoming data. When you configure an FTP client for passive, you're telling it to expect these packets.

TFTP is used for transferring files and has a similar-sounding name to FTP, but beyond that it is very different. TFTP uses UDP port 69 and does not use user names and passwords, although you can usually put some restriction based on the client's IP address. TFTP is not at all secure, so never use it on any network that's less than trustworthy.

Internet Applications

Use this table as a review tool to help you remember each Internet application:

Application	TCP/UDP	Port	Notes
HTTP	TCP	80	The Web
HTTPS	TCP	443	The Web, securely
Telnet	TCP	23	Terminal emulation
SSH	TCP	22	Secure terminal emulation
SMTP	TCP	25	Sending e-mail
POP3	TCP	110	E-mail delivery
IMAP4	TCP	143	E-mail delivery
FTP	TCP	20/21	File transfer
TFTP	UDP	69	File transfer

Chapter 9 Review

Chapter Summary

After reading this chapter and completing the exercises, you should understand the following about the basics of TCP/IP.

Describe common Transport layer protocols

- TCP/IP involves many more protocols other than just TCP over IP. HTTP, DHCP, POP, UDP, and ICMP are just a few of the hundreds of other protocols that operate over IP.
- Connections between computers are called sessions. If every communication requires an acknowledgment from the receiving computer, the session is said to be connection-oriented. Otherwise, the session is connectionless.
- TCP is a connection-oriented protocol while UDP is connectionless. TCP packets must create a connection between systems to ensure that each packet reached its destination intact. UDP packets do not have a built-in checking mechanism to make sure they were received correctly.
- Applications that use ICMP, such as PING, are always connectionless and never need more than a single packet.
- IGMP allows routers to forward multicast IP packets to IGMP groups.

Explain the power of port numbers

- Well-known port numbers fall within the range 0–1023. Web servers use port 80.
- Ephemeral port numbers fall within the range 1024–5000—the classic ephemeral ports—and 49152–65535—the dynamic or private ports.
- Registered ports are those that have been registered with the Internet Assigned Numbers Authority and fall within the range 1024–49151.
- Information about a session is stored in RAM and is called a socket. The sockets stored by two computers in a session with each other are called socket pairs or endpoints.
- The NETSTAT command-line utility, with the –n switch, is used to view a list of endpoints. However, it can't automatically update to display real-time information.
- An open port, or listening port, is a socket prepared to respond to incoming IP packets. You can type `netstat -an` to see all of your listening ports.
- You can use the `netstat -ano` command to identify which application is using a specific port, allowing techs to identify malicious software.
- The NETSTAT switches –a, –n, and –o are important for any tech to know.

Define common TCP/IP applications, such as HTTP, HTTPS, Telnet, e-mail (SMTP, POP3, and IMAP4), and FTP

- HTTP stands for the Hypertext Transfer Protocol. HTTP uses port 80 to transmit the common data used in Web pages.
- To make Web pages available to the public, the Web pages must reside on a computer with Web server software installed and configured. Microsoft's Internet Information Services and Apache HTTP Server are the most common Web server software.
- A Web client is a program, such as a Web browser, that displays or reads Web pages.
- HTTPS stands for Hypertext Transfer Protocol over Secure Sockets Layer (SSL), which uses port 443. HTTPS protects sensitive data, like credit card numbers and personal information, by encrypting it.
- Telnet is a protocol that enables a user with the proper permissions to log onto a host computer, acting as a Telnet client. The user could then perform tasks on a remote computer, called a Telnet server, as if he or she were sitting at the remote computer itself.
- Telnet sends passwords and data in easily detected cleartext, so most servers use Secure Shell (SSH) now.
- The UNIX utilities rlogin, RSH, and RCP enable a user to issue commands to a server remotely. They should not be used across the Internet because none of them encrypt data.

- The term e-mail stands for electronic mail. E-mail is sent using the SMTP protocol on port 25, and is received using either POP3 (on port 110) or IMAP4 (on port 143).
- E-mail servers are needed to help forward, store, and retrieve e-mail messages for end users, who need a valid user name and password to gain access. E-mail can also contain attachments of limited size, like pictures or small programs or data files.
- Sendmail is the leading e-mail server for Linux and UNIX, but it only supports SMTP. Exchange Server is the e-mail server software from Microsoft and it supports both SMTP and POP.
- A mailbox is a storage area with an e-mail server that holds all the e-mail for a specific user.
- An e-mail client allows you to send, receive, and organize e-mail. Popular e-mail clients include Microsoft Outlook, Windows Mail, Mozilla Thunderbird, and Qualcomm's Eudora.
- FTP stands for the File Transfer Protocol, which uses ports 20 and 21, and efficiently transmits large files. Many FTP sites allow anonymous access to avoid end users sending their passwords in clear-text format.
- Active FTP uses both ports 20 and 21 and can be problematic if you are using NAT. Passive FTP uses only port 20 and works fine with NAT.
- Trivial FTP (TFTP) uses UDP port 69 and does not use user names or passwords, making it very insecure.
- A good network tech knows the port numbers for popular Internet applications and protocols such as HTTP, Telnet, SSH, SMTP, POP3, IMAP4, FTP, and TFTP.

Key Terms

Apache HTTP Server *(226)*
connection *(216)*
connectionless *(211)*
connection-oriented *(211)*
dynamic port numbers *(215)*
electronic mail (e-mail) *(231)*
e-mail client *(234)*
endpoint *(216)*
endpoints *(216)*
ephemeral port numbers *(215)*
File Transfer Protocol (FTP) *(235)*
Hypertext Transfer Protocol (HTTP) *(224)*
Hypertext Transfer Protocol over SSL (HTTPS) *(227)*
Internet Control Message Protocol (ICMP) *(213)*
Internet Group Management Protocol (IGMP) *(213)*
Internet Information Services (IIS) *(225)*
Internet Message Access Protocol version 4 (IMAP4) *(232)*
listening port *(219)*
mailboxes *(234)*
NETSTAT *(216)*
open port *(219)*
Post Office Protocol version 3 (POP3) *(232)*
private port numbers *(215)*
RCP *(231)*
registered ports *(215)*
rlogin *(231)*
RSH *(231)*
Secure Shell (SSH) *(229)*
Secure Sockets Layer (SSL) *(227)*
sendmail *(233)*
session *(211)*
Simple Mail Transfer Protocol (SMTP) *(232)*
socket *(216)*
socket pairs *(216)*
Telnet *(228)*
Transmission Control Protocol (TCP) *(212)*
Trivial File Transfer Protocol (TFTP) *(213)*
User Datagram Protocol (UDP) *(212)*
well-known port numbers *(214)*

Key Term Quiz

Use the Key Terms list to complete the sentences that follow. Not all terms will be used.

1. The TCP port numbers ranging from 0–1023 are called ______________.
2. The TCP port numbers ranging from 1024–49151 are called ______________.
3. The protocol used to transmit large files over the Web using both ports 20 and 21 is called ______________.
4. The protocol that is not as popular as POP3 for receiving e-mail is ______________.
5. Port 23 is used by ______________ to emulate terminals on TCP/IP networks.
6. When you send out an e-mail message it uses ______________.
7. The quickest way to send to a few co-workers information about an upcoming meeting would be to send a(n) ______________.
8. The ______________ utility can be used to view the endpoints of your computer's sessions.
9. Telnet has largely been replaced by __________, which provides better security through data encryption.
10. TCP is ______________ in that it requires computers to acknowledge each other; while UDP is ______________ in that it provides no guarantee packets were successfully received.

Multiple-Choice Quiz

1. What port number is the well-known port used by Web servers to distribute Web pages to Web browsers?
   - **A.** Port 20
   - **B.** Port 21
   - **C.** Port 25
   - **D.** Port 80
2. What protocol handles large file transfers between Internet users?
   - **A.** FTP
   - **B.** IMAP
   - **C.** POP3
   - **D.** SMTP
3. How can you tell that a secure Web page transaction is taking place?
   - **A.** The URL in the address bar starts with https
   - **B.** The URL in the address bar starts with http/ssl
   - **C.** The URL in the address bar starts with ssl
   - **D.** The URL in the address bar starts with tls
4. Jane has been tasked to find and implement an application that will enable her boss to log into and control a server remotely and securely. Which of the following applications would work best?
   - **A.** E-mail
   - **B.** FTP
   - **C.** Telnet
   - **D.** SSH
5. How do Web pages get created on the Internet?
   - **A.** By ICANN
   - **B.** By InterNIC
   - **C.** By publishing them
   - **D.** By the FCC
6. Which of the following Microsoft operating systems limit Web site access from other systems when using Internet Information Services software? (Select three.)
   - **A.** Windows 2000
   - **B.** Windows XP
   - **C.** Windows Vista
   - **D.** Windows 2003 server
7. Which of the following are names of Web server software? (Select two.)
   - **A.** Apache
   - **B.** Exchange
   - **C.** IIS
   - **D.** Proxy server

8. Which of the following are names of Internet browser software? (Select two.)
   - **A.** Internet Surfware
   - **B.** Internet Explorer
   - **C.** Firefox
   - **D.** WS_FTP
9. Which of the following items does the *S* in HTTPS represent?
   - **A.** Proxy server
   - **B.** Secure Sockets Layer
   - **C.** Subnet mask
   - **D.** Switch
10. When using Windows, which command will show all used ports and the IP addresses using them?
    - **A.** `telnet localhost 25`
    - **B.** `telnet -ano`
    - **C.** `netstat -an`
    - **D.** `netstat -ao`
11. What is the main difference between TCP and UDP?
    - **A.** TCP is connection-oriented while UDP is connectionless.
    - **B.** TCP supports HTTPS while UDP supports SSL.
    - **C.** TCP sessions can be encrypted while UDP sessions cannot.
    - **D.** TCP is used on Windows while UDP is used on Linux/UNIX/Mac OS X.
12. What protocol is used by applications requiring a single packet over a connectionless session?
    - **A.** TCP
    - **B.** UDP
    - **C.** ICMP
    - **D.** IGMP
13. Which of the following provide Web services? (Select three.)
    - **A.** Apache
    - **B.** IIS
    - **C.** GWS
    - **D.** Exchange
14. Which UNIX utility enables you to connect to a server automatically and run commands without entering a user name and password every time?
    - **A.** Telnet
    - **B.** rlogin
    - **C.** RSH
    - **D.** RCP
15. What should you do if you are having difficulty transferring files with your FTP client when your router supports NAT?
    - **A.** Configure your FTP client to use active FTP.
    - **B.** Configure your FTP client to use passive FTP.
    - **C.** Use SSH to transfer your files instead.
    - **D.** Use Telnet to connect to the server, then use NETSTAT to transfer the files.

Essay Quiz

1. Your company is interested in setting up secure Web pages for credit card transactions. The company currently does have a Web presence. Write two short paragraphs describing the two different port numbers that would be used on the company's improved Web site.
2. After checking various e-mail settings, a colleague of yours mentions port numbers. Write down some quick notes about which TCP ports would handle e-mail.
3. Write down a few notes explaining why some Web pages have an extra *s* after the http in their Web addresses. Be prepared to discuss your findings in class.
4. Write on a sheet of paper a paragraph that describes what a Web server does. Write a second paragraph that describes what an e-mail server does.

Lab Projects

• Lab Project 9.1

Start some Internet programs, like a Web browser, e-mail or FTP client, or an instant messenger. Open a command prompt and type `netstat -ano`. Make a list of the well-known ports in use and the process ID using the port. Then write the actual name of the application identified by the process ID. Linux users can type `ps` to learn the application name of a process ID, but Windows users have to use a third-party tool like Process Explorer.

• Lab Project 9.2

Using a word processing program or a spreadsheet program, create a chart that lists all the port numbers mentioned in this chapter, similar to the following list. Use the Internet to look up other commonly used port numbers as well. Fill in the Abbreviation column, the Full Name column, and the Brief Description column. Repeat this lab exercise several times until you have memorized it fully. This activity will help you pass the CompTIA Network+ exam!

Port #	Abbreviation	Full Name	Brief Description of What This Port Does…
20			
21			
22			
23			
25			
80			
110			
143			
443			