

# TCP/IP and the Internet

The Network+ Certification exam expects you to know how to

- 1.6 Identify the purposes, features, and functions of the following network components: routers, gateways, firewalls
- 2.10 Define the purpose, function, and use of the following protocols used in the TCP/IP (Transmission Control Protocol/Internet Protocol) suite: FTP, TFTP, SMTP, HTTP, HTTPS, POP3/IMAP4, Telnet
- 2.13 Identify the purpose of network services and protocols (for example: NAT – Network Address Translation)
- 3.6 Identify the purpose, benefits, and characteristics of using a proxy service

To achieve these goals, you must be able to

- Explain how routers work using routing tables
- Define static and dynamic routers, and name different dynamic routing standards
- Explain network address translation (NAT) and proxy serving
- Define FTP, TFTP, SMTP, HTTP, HTTPS, POP3/IMAP4, and Telnet

The Internet uses TCP/IP to enable computers—and people—from all over the world to communicate. You’ve seen TCP/IP in operation on lesser networks in previous chapters and have by now a good understanding of how the protocol suite works. This chapter enables you to apply that knowledge to the grandest network of all, the Internet. The chapter starts with an in-depth look into the machines that form the backbone of the Internet—routers—and then explores the many tools that make the Internet run, such as network address translation (NAT), proxy servers, HTTP, FTP, e-mail, Telnet, and SSH (Secure Shell).

## Test Specific

### Real World Routers

Routers, routers, routers! The word “router” invariably seems to send chills down the spines of folks just starting out in the networking world as they contemplate these magic boxes that create all the connections that make up the Internet. Although I’ve referenced routers in numerous spots—you’ve even had a peek at a routing table—you need to know more. We know a *router* directs incoming network protocol packets from one LAN



**Figure 15-1** Routers

to another based on OSI Network layer information stored in the incoming packets—in the case of TCP/IP that means IP addresses. Routers determine where packets must go via their routing tables. To route these packets, a router by definition must have at least two interfaces, although some routers have three or more, depending on the needs of the network. If you think about it for a moment, a router acts a lot like a switch, except it works on the OSI Network layer (Layer 3), while a typical switch works on the OSI Data Link layer (Layer 2). That's why you hear a lot of network folks call a regular switch a *layer 2 switch* and a router a *layer 3 switch*. Be comfortable using both terms, as techs tend to use them interchangeably, even in the same sentence (see Figure 15-1).

Routers come in a dizzying variety of shapes, sizes, and functions. You find little routers used in homes and small businesses, like the handy-dandy Linksys router I use at my house (see Figure 15-2), and mid-sized routers used to connect a couple of buildings. At the top end are the massive backbone routers that literally make the big connections on the Internet (see Figure 15-3).



**Figure 15-2** Little router



**Figure 15-3** Big routers

You don't necessarily need special hardware to have a router. Pretty much every modern operating system enables you to turn a PC into a router by adding an extra NIC, modem, or some other device to connect to another LAN. Figure 15-4 shows a screen shot from my old router, a beat-up Pentium 166 system with two NICs, no hard drive or CD-ROM drive, running a handy little Linux-based router program—called Coyote Linux ([www.coyotelinux.com](http://www.coyotelinux.com))—completely from the floppy drive.

```
Coyote Linux Gateway -- configuration menu

1 ) Network settings                4) Change system password
2 ) System settings
3 ) Package settings

c) Show running configuration       b) Back-up configuration
                                   h) Help
q) quit
-----
Selection:
```

**Figure 15-4** Coyote Linux configuration screen

The vast majority of routers seen in the small- to medium-sized networks act as nothing more than a tool to link your LAN to the Internet via your local ISP. In almost all cases, these routers, whether a special box or just a PC in the network, will have two interfaces: a NIC that connects to your LAN and some other connection that links to a regular phone line, a more advanced telephone connection like ISDN, ADSL, or T1, or maybe a cable modem. Whatever the connection type, these two-interface-only routers are extremely common.

People often describe the Internet as a network of computers, but there's a strong argument to call the Internet a network of routers. Because the Internet is a world-wide network, the most important connections—often called *backbones*—are the long distance connections between cities. When a piece of fiber-optic cable runs from, say, Houston to Chicago, the ends of that connection go to powerful routers like the ones displayed in Figure 15-3, not to computers! Add a few thousand more connections like the one running between Houston and Chicago, and then you see the Internet—a massive network of routers, spanning the entire globe.



**NOTE** In terms of basic function, there's not much difference between a powerful, high-end router and a small SOHO router other than the amount of traffic they can handle.

We can divide all routers into one of two functions. A regular router is typically a device that only connects to other routers. A *gateway* or *gateway router* connects individual LANs to a larger network—usually the Internet. Gateway routers also have extra functions built into them to protect the LAN or to support individual computers that a regular router doesn't. We'll see some of these gateway functions in this chapter.

People often become confused when they hear the terms gateway and default gateway, and in truth, the terms are often synonymous. The term *default gateway* can refer to the hardware router (as in, "that box is my default gateway to the Internet"), but you'll also find it specifically referred to as the *IP address* of the router interface that connects to your LAN, called the *local side* or the *local interface* of your router (see Figure 15-5). The distinction is not of critical importance, but just be aware of different uses of the term when a fellow tech starts talking.

The Network+ exam doesn't expect you to know how to configure a router, but you should be familiar with some of the ways those who do configure routers do it. To use an analogy, if this were an exam about automobiles, you wouldn't actually have to drive a car, but you would need to be able to explain how drivers steer and brake. Get it? A key basic feature of the router is the *routing table*. All routers have a built-in routing table that tells them how to send packets. Where does this routing table information come from? Let's look at how routing tables get created.

## Static Routes

In the bleak old days of the Internet, routing tables were composed entirely of static entries. In other words, somebody who really understood routers (and subnetting) had to type this information into the routing table. The router person would link into the

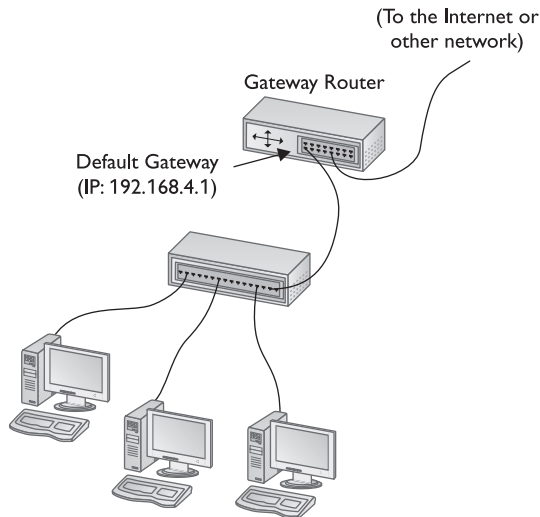


Figure 15-5 A default gateway

router using a serial cable or something called Telnet (see the “Telnet” section later in this chapter) and type in a command to add or remove static routes from the routing table. Would you like to see a routing table? If you’re sitting at a Windows or UNIX/Linux system, get to a prompt and type in one of these two commands: NETSTAT—NR or ROUTE PRINT. Both of these commands result in basically the same output as shown in Figure 15-6.

```
Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-1999.

C:\>route print
=====
Interface List
0x1 ..... MS TCP Loopback interface
0x1000003 ...00 80 ad 7b 48 39 ..... PCI Bus Master Adapter
=====
Active Routes:
Network Destination    Netmask          Gateway          Interface        Metric
0.0.0.0                0.0.0.0          192.168.4.152    192.168.4.15     1
127.0.0.0              255.0.0.0        127.0.0.1        127.0.0.1        1
192.168.4.0            255.255.255.0    192.168.4.15     192.168.4.15     1
192.168.4.15          255.255.255.255  127.0.0.1        127.0.0.1        1
192.168.4.255         255.255.255.255  192.168.4.15     192.168.4.15     1
224.0.0.0              224.0.0.0        192.168.4.15     192.168.4.15     1
255.255.255.255       255.255.255.255  192.168.4.15     192.168.4.15     1
Default Gateway:      192.168.4.152
=====
Persistent Routes:
None
C:\>_
```

Figure 15-6 ROUTE PRINT output

Every IP client has a routing table. Does that mean that every IP client is a router? Well, sort of. IP clients are routers in the sense that they need to know how to address their outgoing packets. An IP client refers to its routing table when it sends packets. Now, you may be thinking that it seems kind of silly for a client to have a routing table when it only has one interface—I mean, where else is it going to send these packets? Two exceptions should make the need for a routing table clear. First, some packets—such as loopback—don’t go beyond the PC. The host needs to know *not* to send loopback packets out, but rather to loop them back. Second, a single host may have multiple NICs, with each NIC assigned a different job. The best way to see this is to take a few moments to understand how to read a routing table, using Figure 15-6 as our guide.



**NOTE** The routing table examples used here are from a Windows system. All routing tables—from the one in your Windows PC, to the one in a Linux system, to the routing table in a hardware router—are virtually identical.

Routing tables typically consist of a number of routes, each listed as a single line in the routing table. Each route consists of five columns of information needed to determine where a packet is to go on the network: the Network Destination, Netmask, Gateway, Interface, and Metric.

- **Network Destination** This is the IP address of the outgoing packet and refers to either a single address or a network ID.
- **Netmask** The netmask is similar to a subnet mask and is compared to the Network destination to determine where the packet is sent. Zeroes in the netmask mean any value is acceptable. Ones mean the value must be exact. Default netmask usually use the classfull 0 and 255 values but they can also use classless values. Remember your subnetting rules if you see values other than 0 or 255 in the netmask!
- **Gateway** This determines the gateway for a packet. On an IP client, this is commonly either the true default gateway for the network, the loopback, or the IP address of the client’s NIC.
- **Interface** This determines through which NIC to send out the packet. On an IP client, this is either the loopback or the NIC’s IP address.
- **Metric** This determines the number of hops to the destination. A *hop* is the number of local networks the packet must move through. In most cases, this is just 1: the client’s local network.

Great, let’s now look at each line in Figure 15-6 and see how the routing table works for a client system. In a routing table, the least restrictive route is listed first, followed by more and more restrictive routes with the most restrictive routes at the bottom of the list. Let me explain.

Here’s the first line in the routing table (with column headers added to each route for clarity):

Network	Destination	Netmask	Gateway	Interface	Metric
	0.0.0.0	0.0.0.0	192.168.4.152	192.168.4.27	1

The network destination of all zeroes means no restriction on the IP address. The netmask of all zeroes means no restriction on the netmask. Your machine interprets this entry to mean that every packet goes out on interface 192.168.4.27 (the client's NIC) via gateway 192.168.4.152, which is in this local network (Metric of 1). In other words, send everything out to the gateway. This is great except for the fact that you'll have other options further down in the routing table. The best way to think of how the routing table works is to think that every packet goes through every route of the routing table until it finds the "best fit" for the route it needs. Even though the first route defines a default route for any packet, other routes further down the list might be more detailed. (As we go further into the list this will make more sense.) As you read the next line, think of the word "except" as you read it. Here's the next line:

Network	Destination	Netmask	Gateway	Interface	Metric
	127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1

So, this provides the first "except" clause: "Except if the packet is addressed for the 127.0.0.0/8 network." In that case, send it out to the 127.0.0.1 interface (the loopback) using the loopback as the gateway. This packet won't use a gateway; the routing table says this by using the client's loopback address rather than the true gateway. Any packet that starts with 127 will loop back. Next line!

Network	Destination	Netmask	Gateway	Interface	Metric
	192.168.4.0	255.255.255.0	192.168.4.27	192.168.4.27	1

Here's the next except clause: "Except if the IP address is in the network ID of 192.168.4/24." In that case, send the packet out on the client's NIC without using the gateway. This is a local address. When you (or when DHCP) enter the IP address and the subnet mask into your system, you are just updating the routing table! Neat, eh? Next line!

Network	Destination	Netmask	Gateway	Interface	Metric
	192.168.4.27	255.255.255.255	127.0.0.1	127.0.0.1	1

Here's the next except clause: "Except if the destination address is exactly for 192.168.4.27." In that case just loopback! Next line!

Network	Destination	Netmask	Gateway	Interface	Metric
	192.168.4.255	255.255.255.255	192.168.4.27	192.168.4.27	1

Here's the next except clause: "Except if the destination address is specifically for 192.168.4.255 (the broadcast address for the local network)." In that case, send it out the NIC without using a gateway. Next line!

Network	Destination	Netmask	Gateway	Interface	Metric
	224.0.0.0	224.0.0.0	192.168.4.27	192.168.4.27	1

Here's the next except clause: "Except if the packet is a multicast packet." In that case send it out to the local network through the NIC without a gateway. Programs that generate multicast packets are uncommon but they do have their niches. Symantec's Ghost drive imaging software is one example of a program that uses multicast addresses. Next line!

Network	Destination	Netmask	Gateway	Interface	Metric
255.255.255.255	255.255.255.255		192.168.4.27	192.168.4.27	1

And the final except clause: "Except if the destination address is exactly for 255.255.255.255 (another broadcast address for the network)." In that case send it out the NIC without using a gateway.

Keep in mind that there's nothing to stop a client from having more than one NIC. I often put a second NIC in my system when I want to test some network thingy without trashing my real network. In that case, my routing table is going to look a lot more complex, as you can see in Figure 15-7. Just look at the Interface column. You'll see I now have two IP addresses: 192.168.4.27 and 203.14.12.1, one for each NIC. Without the routing table, my system wouldn't know which NIC to use to send packets.

By the same token, a regular IP client is not a router in that both NICs are completely separated on the routing table. If you look at the routing table, you won't see any rows that say to send anything with a network destination of network ID 192.168.4.27 to Interface 203.14.12.1. If this were a router you would see rows that instructed the system how to route data from one interface to the other and trust me, there are none here. So, even though IP clients do have routing tables, they don't route by default in the classic sense of moving packets from one network to another. They do route in the sense that it is the routing table that decides whether to send a packet to the gateway or just to keep it on the local network (although nothing stops you from turning the client into a router).

```
C:\>route print
=====
Interface List
0x1 ..... MS TCP Loopback interface
0x1000003 ...00 a0 c9 98 97 7f ..... Intel(R) PRO/100+ PCI Adapter
0x3000005 ...00 50 56 c0 00 01 ..... VMware Virtual Ethernet Adapter
=====
Active Routes:
Network Destination        Netmask          Gateway          Interface        Metric
0.0.0.0                    0.0.0.0          192.168.4.152    192.168.4.27     1
127.0.0.0                  255.0.0.0        127.0.0.1       127.0.0.1       1
192.168.4.0                255.255.255.0    192.168.4.27    192.168.4.27     1
192.168.4.27              255.255.255.255  127.0.0.1       127.0.0.1       1
192.168.4.255             255.255.255.255  192.168.4.27    192.168.4.27     1
203.14.12.0               255.255.255.0    203.14.12.1     203.14.12.1     1
203.14.12.1               255.255.255.255  127.0.0.1       127.0.0.1       1
203.14.12.255            255.255.255.255  203.14.12.1     203.14.12.1     1
224.0.0.0                 224.0.0.0        192.168.4.27    192.168.4.27     1
224.0.0.0                 224.0.0.0        203.14.12.1     203.14.12.1     1
255.255.255.255          255.255.255.255  192.168.4.27    192.168.4.27     1
Default Gateway:         192.168.4.152
=====
Persistent Routes:
None
C:\>
```

**Figure 15-7** Routing table for Mike's PC with two NICs



The question now becomes this: Where did this routing table come from? Was it entered statically? Thank goodness no! Your system generates this table at boot based on your IP information. Static IP addresses are rarely used in client systems unless something unique is taking place. Remember the subnetting scenario discussed in Chapter 11, “TCP/IP,” where you had to configure the routing table to split one subnet into two? That would be one situation that would require you to do this—but leave that to the router gurus!

## SNMP

One big question I often receive from someone new to routers is “How do I access the router to make changes? For example, how can I add a static route to a router?” The oldest way is using a terminal. Heavy duty routers come with special configuration ports, usually serial ports. You connect a PC to that port, and then use a terminal program to connect to the router. Cisco’s IOS routing software is a classic example. These interfaces are text-based and challenging to use but powerful. Lower-end routers will use a web-based interface like the one shown in Figure 15-8.

A *static route* tells a router a specific path to use to reach another network. In a small network, you could tell the router which IP to use for Internet traffic, for example, and which to use for internal traffic.

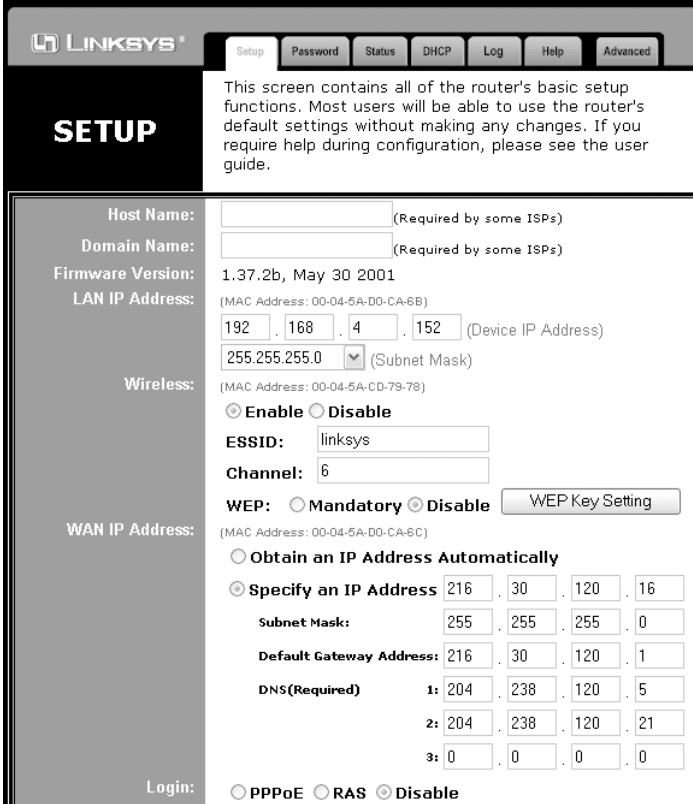
You can use a powerful protocol called *Simple Network Management Protocol (SNMP)* to track the status of routers. SNMP gives smart devices—routers, switches, and individual PCs—the capability to report their status and to allow administrators to make changes. SNMP-capable devices are common on all but the lowest echelon of networking hardware.

Most small- to medium-sized routers today do not use static routing—that would be entirely too cumbersome for hard working router administrators! Rather, they use some sort of dynamic routing. Let’s go there now.

## Dynamic Routing

Early on in the life of the Internet it became painfully clear that routers using only static IPs were incapable of handling the demands of anything but a network where nothing changed. If a new router was introduced to the network, it was useless until humans could get in and update not only the new router but also all of the new router’s neighbors. While this might have worked when the number of routers on the Internet was small, it simply did not work as the number began to grow past a few dozen.

Furthermore, neither TCP/IP nor the Internet was ever designed to run on only static routers. When DARPA first created the entire concept of TCP/IP and the Internet, they were tasked by the U.S. military to create a network that could survive having any single part disappear under a mushroom cloud. In reality, the Internet invented routers more than routers invented the Internet. The Internet’s designers visualized a mesh of routers, each having at least three connections, to provide a large level of redundancy in the case of multiple connection failures. They never realized just how large this mesh of routers would someday become!



**LINKSYS®**

Setup Password Status DHCP Log Help Advanced

## SETUP

This screen contains all of the router's basic setup functions. Most users will be able to use the router's default settings without making any changes. If you require help during configuration, please see the user guide.

Host Name:  (Required by some ISPs)

Domain Name:  (Required by some ISPs)

Firmware Version: 1.37.2b, May 30 2001

LAN IP Address: (MAC Address: 00-04-5A-D0-CA-6B)  
 192  168  4  152 (Device IP Address)  
 255.255.255.0 (Subnet Mask)

Wireless: (MAC Address: 00-04-5A-CD-79-78)  
☒ Enable ☐ Disable  
 ESSID:  linksys  
 Channel:  6  
 WEP: ☐ Mandatory ☒ Disable

WAN IP Address: (MAC Address: 00-04-5A-D0-CA-6C)  
☐ Obtain an IP Address Automatically  
☒ Specify an IP Address  216  30  120  16  
 Subnet Mask:  255  255  255  0  
 Default Gateway Address:  216  30  120  1  
 DNS(Required) 1:  204  238  120  5  
 2:  204  238  120  21  
 3:  0  0  0  0

Login: ☐ PPPoE ☐ RAS ☒ Disable

**Figure 15-8** Linksys router web interface

All these routers needed to be able to communicate with each other in such a way that they could detect changes to the network and redirect routes to new interfaces without human intervention. Certainly, a router would initially have a few routes listed on its routing table, but once the router started operating, it would need to update the table. The answer: *dynamic routing*.

Like every other aspect of the Internet and TCP/IP, dynamic routing methods have grown in number and complexity over the years. The variety of these methods—with fun acronyms and initials such as RIP, OSPF, BGP, and IGRP—has reached a point where we categorize them into two types: interior routing methods and exterior routing methods. Interior routing methods are used primarily in routed private networks and smaller Internet ISPs. The main Internet backbone and large ISPs use exterior gateway routing methods. Let's check them out.

The oldest of all routing methods is called *Routing Information Protocol (RIP)*. Developed in the late 1970s and early 1980s, RIP stood alone as the only routing method for many years. RIP uses a *distance vector algorithm*—basically just a nice way to say that neighboring routers share their routing tables. RIP is now only used as an interior routing

protocol, having long been kicked off the more critical Internet routers. RIP has a number of shortcomings. In particular, RIP routers do not respond rapidly to changes and tend to flood the network with information as they update. Regardless of these shortcomings, you can count on any router knowing how to do RIP. Most interior routers still use RIP, but it is slowly being replaced by OSPF.



**TIP** Among its many shortcomings, one item in particular prevents RIP from use on the Internet: it can handle a maximum of only 16 hops.

The *Open Shortest Path First (OSPF)* methodology is a much newer and far better way to update routers dynamically. OSPF routers use a *link state* algorithm: routers constantly monitor their neighbors with tiny messages—called *hellos*—and share more detailed information—called *link state advertisements*. If a connection is lost or created, the routers share only the changed information with their neighboring routers. A RIP router sends routing information at a set interval, even if there is no change.

Knowing that an exterior gateway protocol is used on the Internet backbone, and that there's only one Internet, you shouldn't be too surprised to learn that the Internet uses only one exterior gateway routing protocol: *Border Gateway Protocol (BGP)*. BGP has also been around for quite some time, but it has gone through a number of iterations. The current one is BGP-4. BGP works using a distance vector methodology like RIP, but once the routers have initially exchanged routing tables, they only pass changes in their tables, rather than entire tables, dramatically reducing router traffic.



**NOTE** I couldn't discuss routing methods without at least mentioning Interior Gateway Routing Protocol (IGRP) and its successor, Enhanced Interior Gateway Routing Protocol (EIGRP). Cisco developed these protocols to work in enterprise-wide routing environments. If Cisco had its way, EIGRP would replace BGP-4 as the primary Internet protocol!

In addition to routing protocols, most routers offer extra features such as the capability to read an incoming IP packet and send it along on another port to another destination based on network ID and subnet mask. Although these are not "official" router functions, these processes—called natural address translation (NAT) and proxy serving—have become closely associated with routers, in particular gateway routers that connect local networks to the Internet. Let's look at both of these router-like features in detail and see how they are used to connect to the Internet.

## Connecting to the Internet

If you're going to connect your local TCP/IP network to the Internet or to any other TCP/IP network, you're going to need a gateway router. Now, a router in the sense of a device that connects two different IP network is a marvel of technology, but it also has some limitations. First of all, every computer on the local network must have a legitimate Internet IP address. This is not always the best idea. Second, there just aren't that many

IP addresses available and getting enough IP addresses for every computer in your network might be rather expensive. Finally, “real” IP addresses mean your computers are exposed to hacking from outside your network.

Real IP addresses are not the only problem with routers. Another issue comes in the form of TCP/UDP ports. Routers, or at least the classic routers we’ve discussed so far, ignore port numbers. But hackers can take advantage of ports to do mean and nasty things to your computers.

The answer to both of these issues is to hide IP addresses and/or ports. The best place to go about this hiding business is at the interface between your local network and the larger network, that is, at the router. Most every router now has extra tools to do exactly this type of hiding. Two technologies that handle the majority of these hiding chores are called NAT and proxy server.

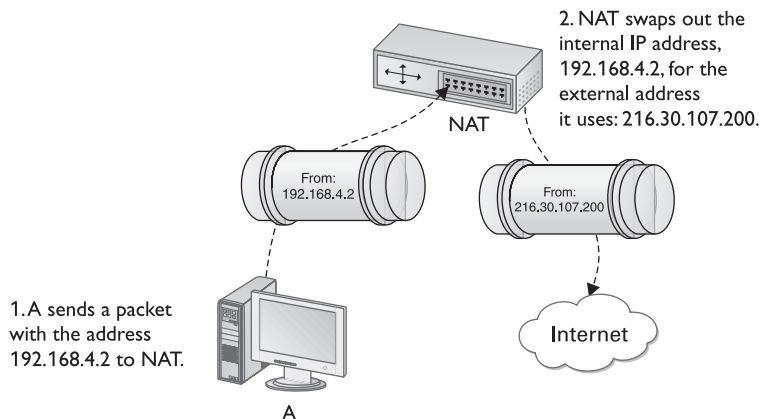
## NAT

The Internet has a real problem with IP addresses, or rather a lack of IP addresses. Not only is it difficult to get public IP addresses for every system in your network, but most ISPs charge you for them. Additionally, any system using a public IP address is susceptible to hacking, requiring the use of protection devices called firewalls (see the firewall discussion in Chapter 17, “Protecting Your Network”). *Network address translation (NAT)* was created in an effort to reduce the demand for public IP addresses, and to provide more security to systems.

Network address translation is a process whereby a NAT program running on a system or a router translates a system’s IP address into a different IP address before it’s sent out to a larger network. A network using NAT will provide its systems with private IP addresses—192.168.1.x addresses are the most popular, but other private IP addresses work equally well. The system running the NAT software will have two interfaces, one connected to the internal network and the other connected to the larger network. The NAT program takes packets from the client systems bound for the larger network and translates their internal private IP addresses to its own public IP address, enabling many systems to share that single IP address (see Figure 15-9).

When a system sends a packet destined for another system on the Internet, it includes the destination IP address, the destination port, and its own IP address. It also includes an arbitrarily generated origination port number that will usually be used as the incoming port number for the return data of the session. The NAT uses the origination port number of the outgoing IP packet, recording that port number along with the internal IP address of the sending system into its own internal table. The NAT then replaces the sending system’s IP address with its own IP address and sends the packet out to the Internet. When the packet returns from the larger network, the NAT refers to its internal table of IP addresses and port numbers to determine which system should receive the incoming packet (see Figure 15-10).

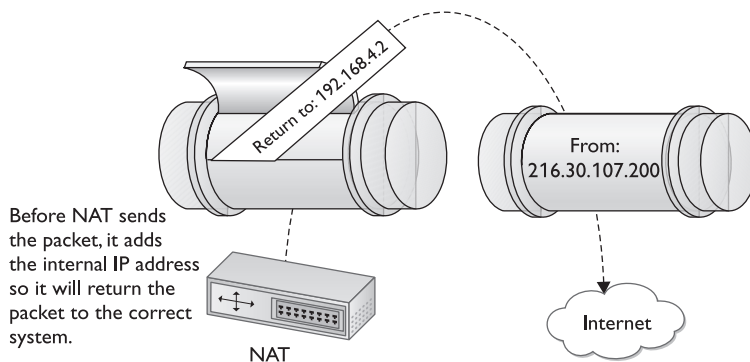
NAT is not the perfect solution for everything. It works well for networks where the clients access the Internet but are not themselves accessed by systems on the Internet. You don’t place web servers as NAT clients, for example, because systems outside the



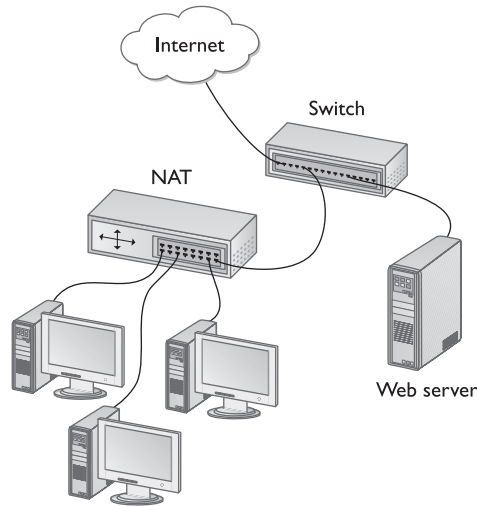
**Figure 15-9** NAT swapping private and public IP addresses with wild abandon

network typically cannot access systems on the NAT. If you want a browser to be able to access your web server, you place the web server *outside* the NAT-controller area. Figure 15-11 shows a typical placement for a web server in a NAT network.

As more and more gateway routers have appeared in small networks, some folks wanted to place web servers and other servers inside the protected network, yet also make those servers accessible for folks across the Internet. As a result, many routers now come with a special feature called *port forwarding*. A router with port forwarding enables you to direct incoming traffic based on port number (like port 80 for HTTP) to a specific computer in your network. This computer will have a private address, but the router will forward packets with that specific port number to that one system.



**Figure 15-10** NAT adding more information



**Figure 15-11** Typical placement of a web server in a NAT network

There are an amazing number of ways to implement NAT. You can use the NAT functions built into the OS (nearly every operating system comes with NAT capabilities) or you can buy a third-party NAT program to make any system with two interfaces a NAT server. You can even buy a router with built-in NAT. Many operating systems come with NAT programs, but in many cases you simply do not see them—they just work! Many of the popular gateway routers come with DHCP and NAT built into the same box. Even my little Linksys router (refer to Figure 15-8) has NAT built in. I just give it an IP address and a subnet mask for its internal interface and it automatically translates any IP address from my network. There simply aren't any NAT settings in many cases!



**NOTE** Not all TCP/IP applications work well with NAT.

NAT also provides a strong defense against hacking since outsiders simply cannot see any of the systems behind the NAT system. To other systems on the Internet, your entire private network looks like just one system—the NAT system. Any system running a NAT gets labeled with the term *firewall*, since it acts as a protector of the private network. You'll learn that a firewall means much more than simply NAT when you get to Chapter 17, "Network Security."

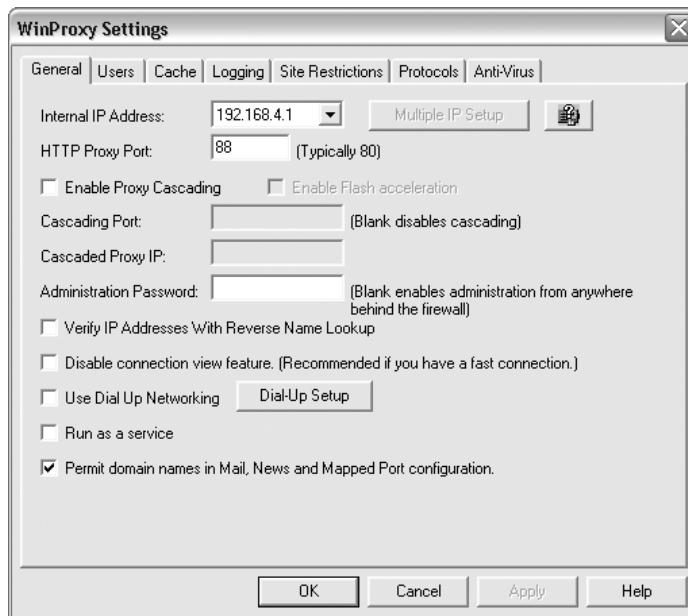
NAT has become extremely popular for networking. In fact, it has become so popular that the long-anticipated day when the world runs out of IP addresses has thus far failed to materialize due to the dominance of networks using NAT (combined with dumping class

licenses for CSLIDs—refer to Chapter 11, “TCP/IP,” to refresh yourself on CSLID). A NAT’s capability to enable multiple systems to share a single IP address, combined with strong protection against hacking, have made NATs as common as servers in most networks.

## Proxy Server

A *proxy server* also hides your internal computers from outside networks but uses a totally different method. While a NAT translates incoming and outgoing IP addresses by switching out its own IP address for each system’s IP address, a proxy server receives and sends encapsulated packets from specific applications. Proxy servers commonly translate the TCP port number to another port number. Because of this encapsulation and port translation, applications like web browsers and e-mail clients must be *proxy aware*—that is, they must be able to (a) know the IP address of the proxy server so they can encapsulate packets and (b) change their standard ports to whatever the proxy server uses. For example, HTTP uses TCP port 80 by default, but we can change the proxy server to accept only certain TCP port numbers, like port 88 for HTTP requests from clients. When the proxy server receives those requests, it will change the client system’s IP address to its own, change the port back to 80, and then send the request out to the Internet (see Figure 15-12).

Proxy servers can dramatically improve performance for groups of users. This is because a proxy server can *cache* requests from users, greatly reducing network traffic. Most businesses access a few web sites frequently. Proxy servers can hold on to the information resulting from user requests for a prespecified amount of time, eliminating the need to reaccess that information from the remote site that contains the HTML document.



**Figure 15-12** A proxy server at work

Proxy servers can also be used to filter requests. This can further secure a network by limiting the types of web sites and Internet resources its users can access. For instance, you can use a proxy server to block certain web sites (like BestBuy.com and Amazon.com) if you find that your employees spend work time shopping for music online.

## **So What's the Big Difference Between NAT and a Proxy, Anyway?**

As you know by now, both proxies and NAT mask IP addresses, enabling a network to have a set of internal private IP addresses that use one public IP address to communicate with the Internet. The difference between using proxies and NAT is where they operate in the network structure. Proxies work at the application level, which means the relevant applications must know how to interact with a proxy, whereas NAT works at the router level, providing transparent Internet access to users.

Think of a proxy server as an old-time telephone operator in a hotel. Just as the hotel operator receives incoming calls for hotel guests and forwards them to the proper room, a proxy server takes incoming requests for Internet services (such as FTP) and forwards them to the actual applications that perform those services. Conversely, just as a guest needing an outside line would go through the hotel operator, a network user needing an Internet service goes through the proxy to the Internet. Because proxies provide replacement connections and act as gateways, they are sometimes known as *application-level gateways*.

Now that we know *how* information travels outside of your network, let's take a look at the information your users have been trying so hard to find.

## **TCP/IP Applications**

Once you have a PC connected to the Internet, but safely tucked behind a protective router, you're ready to get some serious work done. The Internet, or more specifically, TCP/IP, offers a phenomenal variety of applications and protocols, from the World Wide Web to e-mail, FTP, and Telnet. The last section of this chapter covers these topics in detail. Let's do it!

### **The Web**

Where would we be without the World Wide Web? The Web functions as the graphical face for the Internet. Most of you have used it, 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 of the chapter looks at the Web and the tools that make it function, specifically the protocols that enable communication over the Internet.

You can find an HTML document on the Internet by entering a URL in your web browser. A *URL*, short for *Uniform Resource Locator*, is a global address that all documents and other resources on the Web must have. When you type the URL of a web page, such as `http://www.mhtechd.com`, you are telling the browser which TCP/IP application to use and typing the address by which your browser locates the HTML document on a remote computer.



## HTTP

*HTTP* is short for *HyperText Transfer Protocol*. It is the underlying protocol used by the World Wide Web, and it runs by default on TCP/IP port 80. Notice the HTTP at the beginning of the URL in Figure 15-13. The HTTP at the beginning of the URL defines how messages are formatted and transmitted, and what actions web servers and browsers should take in response to various commands. When you enter a URL in your browser, it sends an HTTP command to the web server directing it to find and return the requested web page.

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, Active Server Pages, and cookies.

## Publishing Web Pages

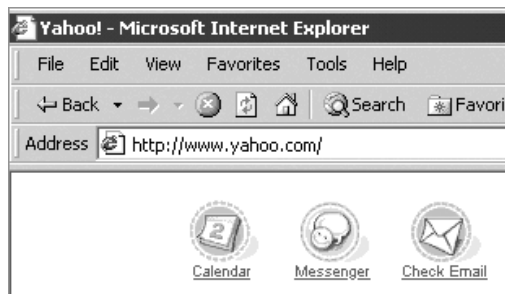
Once you've designed and created a web document, you can share it with the rest of the world. Sharing a page on the World Wide Web is quite an easy matter. Once the web document is finished, you need to find a server that will host the site. Most ISPs provide web servers of their own, or you can find relatively inexpensive web hosting elsewhere. 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.

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 vanity or fan web page, but if you are doing 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 selected your web host, you need to select your domain name. Domain names have to be registered through InterNIC; this enables your web site name to be resolved to the IP address of the server that has your web site. Fortunately, registering your domain name is a breeze. Most web hosts will offer to register your domain name for you (for a nominal fee). The cost of registering your domain name is usually about \$10 a year.

**Figure 15-13**

A run-of-the-mill URL



The trickiest aspect of registering your domain name is finding a domain name that has not already been taken. The last time I checked, the Web had about 11 million registered domain names, and that number has undoubtedly climbed steadily since. Many web sites that offer registration for domain names (such as [www.register.com](http://www.register.com)) offer a search that will check to see whether a name has already been registered.

Once you have your domain name registered and your web hosting covered, it's time to upload your web page to the web server. What's a web server? I'm glad you asked!

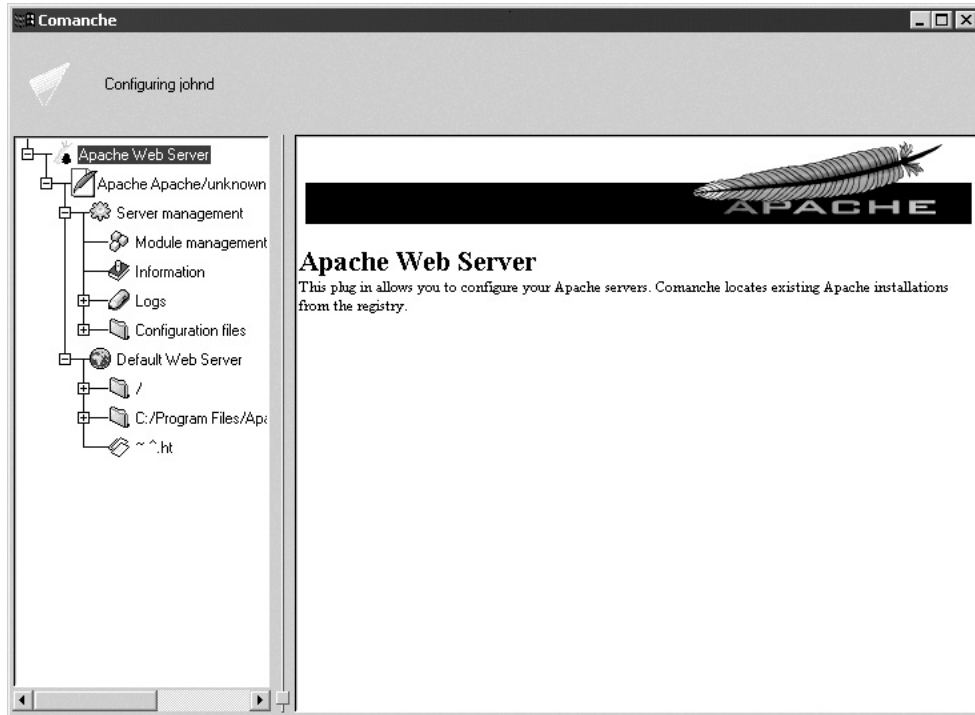
## Web Servers and Web Clients

A web server is a computer that delivers (or *serves up*) web pages. Every web server has at least one static IP address and at least one domain name. For example, if you enter the URL [www.mhtechd.com/index.html](http://www.mhtechd.com/index.html), your browser sends a request to the server named [www](http://www.mhtechd.com) whose domain name is [mhtechd.com](http://www.mhtechd.com). The server then fetches the page named [index.html](http://www.mhtechd.com/index.html) and sends it to your browser. A web server *must* have a static IP address, because once you register your domain name, browsers must be able to resolve that domain name to a steady, unchanging IP address.

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. Windows 95/98/Me operating systems make poor web servers, for example, due to their poor support for multiple connections. More than ten connections to a Windows 98 or Me system can cause the system to lock up and Microsoft strongly recommends against using 95/98/Me as an operating system for your web server. Web serving programs vary greatly in their capabilities. If you use a program called Personal Web Server (PWS), you'll serve a maximum of one web site out to a limited number of users. The Windows NT Workstation OS has similar limitations. The Windows 2000 Professional OS can run a light version of Microsoft's Internet Information Server (IIS), but as with Personal Web Server, it can only host one web site and has a ten-connection limit. You can, however, use more powerful, third-party, web server programs such as Apache and get around this limit.

Windows 2000 Server, Windows Server 2003, and UNIX/Linux-based operating systems can serve as full-blown web servers. This means they can host multiple web sites with multiple domain names, as well as multiple FTP and newsgroup servers. The two most popular web server software applications are Apache and Microsoft's Internet Information Server (IIS). As of this writing, Apache serves about 60 percent of the web sites on the Internet. Apache is incredibly popular because it's full-featured and powerful, runs on multiple operating systems (including Windows), and best of all, it's *free*! Better yet, you can add on different GUIs such as WebAdmin or Comanche that make administering Apache a breeze. Figure 15-14 illustrates the wonderful simplicity that is Comanche.

Microsoft's IIS is both easy to use (although complex to configure and secure properly) and very powerful. IIS not only serves web pages, it also can create FTP servers and newsgroup servers, and offers a large number of administrative options. You can even administer your IIS server remotely using an administrative web page. The IIS console runs from the Microsoft Management Console, and it's simple to use, as you can see in Figure 15-15. Alas, IIS is only available on Windows-based systems.



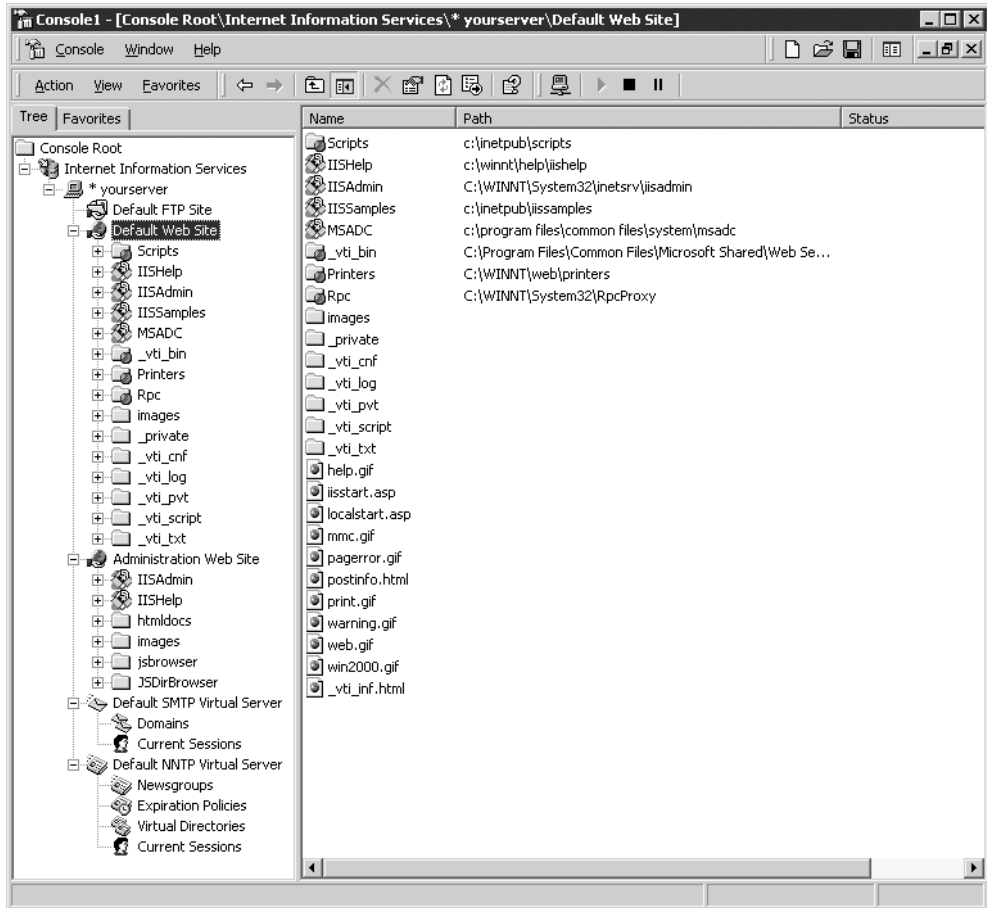
**Figure 15-14** The Comanche GUI

There are many other web server solutions to choose from besides Apache and IIS, however, including Netscape Enterprise, iPlanet Web Server, and Enterprise for NetWare.

Web clients are the programs we use to surf the Internet. A user uses a client program (an Internet *browser*) to read web pages and interact with the Internet. Most browsers can handle multiple functions, from reading HTML documents to offering FTP services and even serving as an e-mail or newsgroup reader. The two biggest Internet browsers out there are Microsoft's *Internet Explorer* and Netscape's *Netscape Navigator*. Both are full-featured browsers that offer nearly identical services. Another fine Internet browser is Mozilla's *Firefox*. Firefox offers many options for the more experienced Internet surfer, and accesses information from the Internet quickly. The best thing about all of these browsers is that they're free!

## Secure Sockets Layer and HTTPS

Because the Web has blossomed into a major economic player, the concern over security has become a near panic. 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



**Figure 15-15** The IIS console

easy online buying into a nightmare of being robbed blind and ending up living in a refrigerator box.

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 purchase *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. Both Netscape Navigator and Internet Explorer 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 URL. By convention, URLs that use an SSL connection start with *https* instead of *http*. *HTTPS* stands for *HyperText Transport Protocol with SSL*.

## Configuring a Web Browser

Configuring your web browser to run on the Internet is a simple process. In the case of Windows, an Internet Connection Wizard walks you through all the steps you need to get Internet Explorer to surf the Web with ease. Netscape Navigator and Firefox are much the same. You need to make sure your TCP/IP settings are correct.

Two things you can control on your browser are the number of cookies the browser uses and the caching of web pages. Each browser has its own particular way of accessing the user configuration areas. In Netscape Navigator, the preferences are set in Edit | Preferences. In Internet Explorer, go to Tools | Internet Options. (See Figure 15-16.)

## Troubleshooting

If you're having problems connecting to the Internet, one of the first things to try is *pinging* a domain using the PING command at a command prompt. This command will tell you if there are connectivity problems.



**Figure 15-16** Internet Options in Internet Explorer

When using your browser to see if there are connectivity issues, either call up a web site you have not accessed for a long time, or reload the page once or twice, because browsers are usually set to cache frequently accessed web pages, which can therefore appear even when your Internet access isn't working.

Whether you are using a dial-up modem or a cable modem (to a lesser degree), your connection is likely to slow down during peak usage times on the Internet when masses of people are trying to access the same information at once. Lunchtime, supertime, and evenings are popular times for surfing and checking e-mail. If you must join the crowd, you may have to be patient.

When buying on the Internet from a secured web page using SSL, be careful to follow the instructions exactly about not over-clicking buttons. Occasionally, information travels a little more slowly over an SSL connection than over a regular unsecured connection, due to the time it takes to encrypt and decrypt the information. If you keep clicking the order button over and over again, you may be sending in order after order to the server. However, if you don't mind paying for the same item 50 times, go right ahead!

Now that we've taken a look at the World Wide Web, it's time to learn about the second most popular feature of the Internet: e-mail.

## E-mail

*E-mail*, short for *electronic 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.

When you create an e-mail message, you must specify the recipient's e-mail address, consisting of the user's name and a domain name: for instance, `MyName@Mhtechd.com`. When you send an e-mail message, it travels from router to router until it finds the domain in question. Then your message is directed to the specific user to whom it's addressed. If you want to, you can also send the same message to several users at the same time. This is called *broadcasting*.

When a message is sent to your e-mail address, it is normally stored in an electronic mailbox on your ISP's server until you come and get it. Some ISPs limit the amount of time they keep messages around, so always check this aspect of your user agreement! Most e-mail client programs can be configured to signal you in some way when a new message has arrived. Once you read an e-mail message, you can archive it, forward it, print it, or delete it. Many e-mail programs are configured to automatically delete messages from the ISP's 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 IMAP to receive e-mail.

## SMTP, POP3, and IMAP, Oh My!

The previous discussion might lead you to think e-mail is directly connected with the World Wide Web, but in fact, the two are quite separate and different. HTML pages use the HTTP protocol, whereas e-mail is sent and received using a number of different protocols. 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/IP port 25, and is used by clients to send messages. You need to specify the POP or IMAP server as well as the SMTP server when you configure your e-mail application, of course. Otherwise, you could send but not receive e-mail!

**POP3** *POP3* is the protocol that receives the e-mail from the server. It stands for *Post Office Protocol version 3*, and uses TCP/IP port 110. Most e-mail clients use this protocol, although some use IMAP.

**IMAP** IMAP is an alternative to POP3. *IMAP* stands for *Internet Message Access Protocol*, and like POP3, it retrieves e-mail from an e-mail server. IMAP uses TCP/IP port 143. The latest version, *IMAP4*, 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.

Other Internet protocols include Extended Simple Mail Transfer Protocol (ESMTP), Authenticated Post Office Protocol (APOP), Multipurpose Internet Mail Extensions (MIME), and Directory Access Protocol (DAP). Many mail servers are also adding S/MIME, SSL, or RSA support for message encryption; and Lightweight Directory Access Protocol (LDAP) support to access operating system directory information about mail users.

## Alternatives to SMTP, POP3, and IMAP

While SMTP and POP3 or IMAP 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 Microsoft's MSN Hotmail and Yahoo! Mail.

The key benefits of web-based 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).



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 at an extra charge).

The best example of proprietary e-mail is the popular America Online (AOL). When you subscribe to the AOL service, you are in a sense accessing the Internet through a gated community—you see the Internet, but using the America Online interface. Figure 15-17 shows a typical AOL client installed on a Windows system.



**NOTE** If you have an AOL account, you can also access your e-mail through a web-based interface, just like with Hotmail or Yahoo! Mail, from any computer connected to the Internet.

## E-mail Servers and E-mail Clients

To give you a clearer idea of how the whole enchilada works, I'm now going to describe an e-mail server and an e-mail client.

**E-mail Server** Many people have heard of web servers and know what they do, but for some reason e-mail servers remain a mystery. This is odd, because e-mail servers are nearly as prevalent on the Internet as web servers. E-mail is used daily by millions of people, both within private networks and on the Internet. This means that e-mail servers are a vital part of any large network.

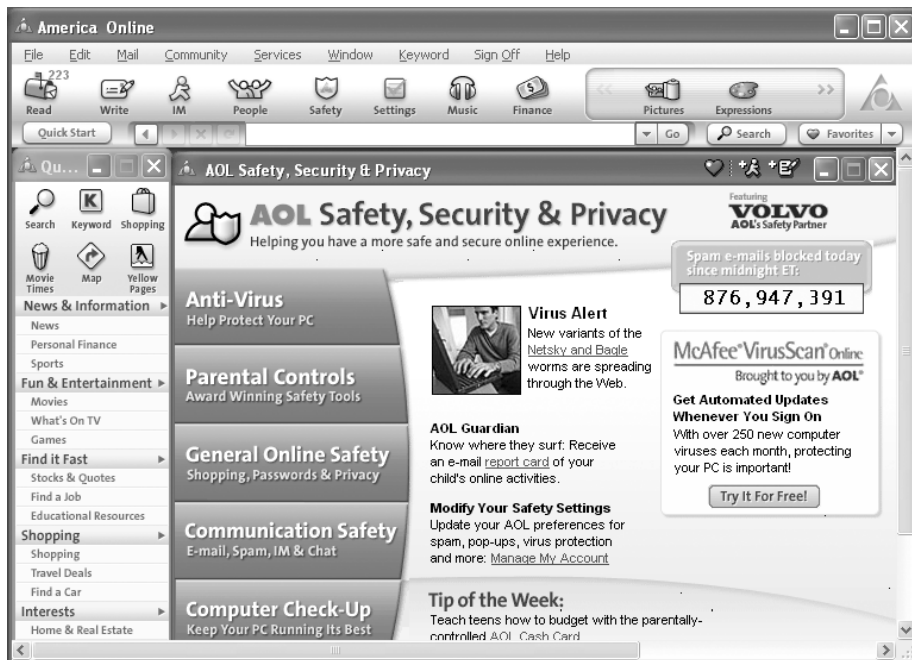


Figure 15-17 Typical AOL interface



*E-mail servers* accept incoming mail and sort out the mail for recipients into 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 interface, 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 e-mail server and downloads the messages from the e-mail server to the client computer.

**Configuring an E-mail Client** Configuring a client is an easy matter. You need the POP3 or IMAP address and the SMTP address for the e-mail server. The SMTP address for MHTechEd, for example, is mail.mhteched.com. 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. The user name will usually be part of the e-mail address. For example, the user name for the e-mail address ghengizsam@mhteched.com will probably be ghengizsam.

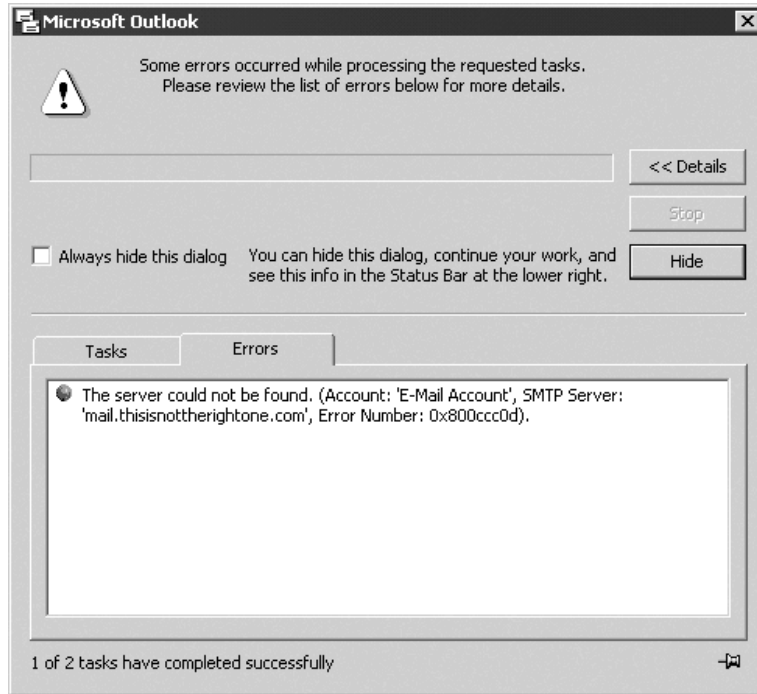
## Troubleshooting E-mail

If you encounter a problem with an SMTP or POP3 name, it's likely you have a problem with the DNS not recognizing the name of the mail server you have entered. Figure 15-18 shows you the error screen in Microsoft Outlook resulting from a bad SMTP connection. In this case, you can check to see if there is a problem with the mail server. Because an SMTP or POP3 name is a name resolved in DNS, how can you check to see if that server is online? You can ping it. Ping the mail server to see if you can find it online. If it is not responding to the ping, you have your answer right there.

Another common problem is a bad password. A mail server is like any other server: you need permission to access its resources. If you can't get access, your password may not be correct. Most e-mail client programs will prompt a dialog box to appear if your e-mail password is set incorrectly, as seen in Figure 15-19.

## FTP

*File Transfer Protocol (FTP)* is the 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/IP ports 21 and 20 by default, although you can often change the port number for security reasons.



**Figure 15-18** Bad SMTP!

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 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.



**Figure 15-19** What was my password again?

## FTP Servers and FTP Clients

Like many Internet applications, FTP uses a client and server arrangement. The FTP server does all the real work of storing the files, keeping everything secure, 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.

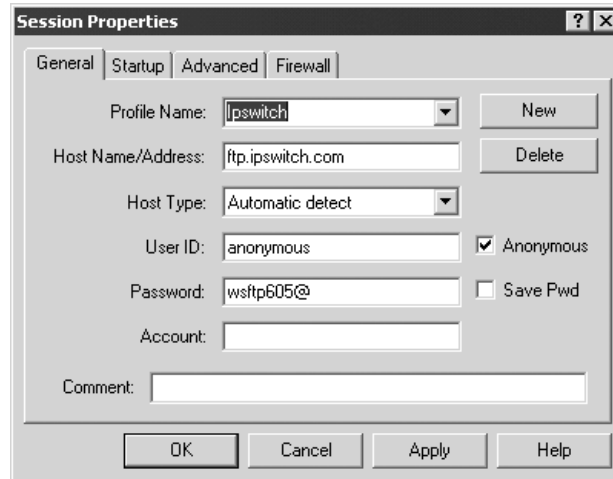
**FTP Servers** Most web servers come with their own internal FTP server software, enabling you to set up an FTP server with a minimum of fuss. These bundled versions of FTP server are robust, but do not provide all the options one might want. Luckily for you, many specialized FTP server software applications provide a full array of options for the administrator.

One aspect of FTP servers you should be aware of concerns FTP passwords. Although it may sound like a good idea to set up a secure, user-only FTP server, in the end this is *less* secure than an anonymous FTP server. How can this be? The problem is that FTP passwords are unencrypted (that is, they are sent over the Internet as plain text). Suppose you are an administrator setting up an FTP server for a company. You decide to extend user rights and login permissions to your FTP server so your company employees may access and download some important programs. When a remote user enters a user name and password to log onto the FTP server—the *same* data they use to log on and off the network—they send this data in *clear text* to the FTP server. Anyone who happens to be eavesdropping on the *network* can intercept this user name and password and use it to log onto the network as if they were an authorized user. This, as you can see, is very, *very* bad for security! In the end, it's safer just to set up a general FTP server with anonymous access, and then change the ports it uses. That way only people you authorize to know about the FTP server can find out which port their FTP client software must use to access the data. This is still not foolproof—someone taking the trouble to tap into your dial-up line can retrieve the FTP port information, and then log in anonymously—but it certainly deters casual mischief.

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.

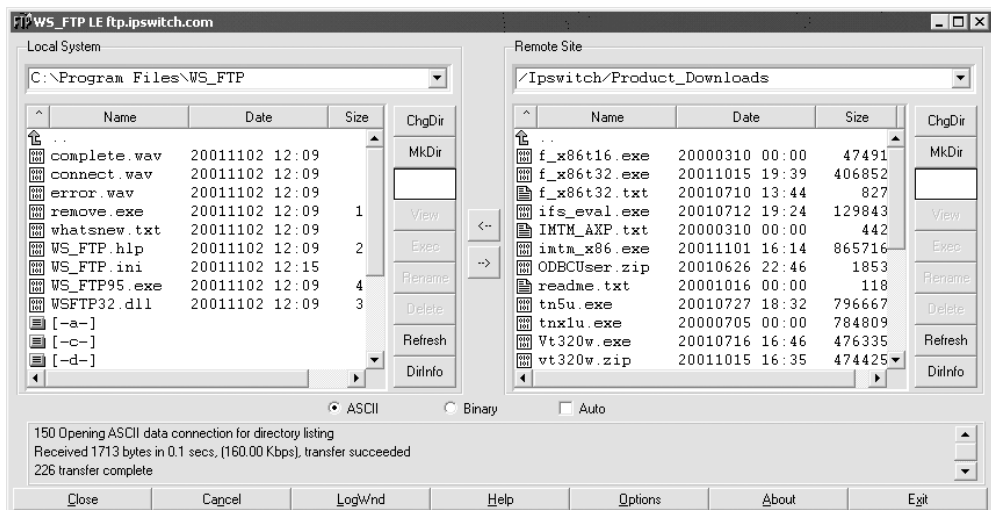
**FTP Clients** FTP clients, as noted before, can access an FTP server through a web site, a command line, or a special FTP application. Usually special FTP applications offer the most choices for accessing and using an FTP site.

**Configuring an FTP Client** Using an FTP client to upload content is a simple process. To transfer files via FTP, you must have an FTP client installed on your PC. Most FTP sites require a user name of *some* sort to log in, even if the FTP server allows anonymous logins. In the case of anonymous FTP, it's common for the user name to be anonymous and the password to be your e-mail address. You must also know the host name of the FTP server. This name is an IP address, which is resolved to a host name using DNS. The MHTechEd FTP server, for example, is `ftp.mhteched.com`.



**Figure 15-20** FTP login using WS\_FTP LE

When you first start up an FTP client program, a dialog box will appear in which you can enter this information (as shown in Figure 15-20). After you log in, you will have access to the files on the FTP server's hard drive. One pane will display the contents of your hard drive, and the other will show you the FTP site's hard drive (see Figure 15-21).



**Figure 15-21** Downloading fun with WS\_FTP LE!

Your FTP client should let you select which file transfer mode you want to use, either ASCII or binary. *ASCII* mode is used to transfer text files, while *binary* mode is used to transfer binary files, like programs and graphics. Most FTP clients have an Automatic transfer mode option, which automatically detects which transfer mode is correct for each file.

## Troubleshooting FTP

If you can't connect to an FTP site, first make sure you have an *active* dial-up or direct Internet connection. FTP programs are not automatic dialers. If your connection closes after a certain number of minutes of inactivity, you have run afoul of the FTP server. This is a feature for system administrators: you can set an FTP server to boot out users who have been inactive for some number of minutes or hours. This is particularly important on anonymous FTP servers that have a limit regarding how many users can be on simultaneously. Most FTP sites will close a connection after a few minutes of inactivity. If the files you transfer are corrupted, the most likely problem is the transfer mode you (often unwittingly, by not changing a previous selection) chose. If you try to transfer a binary file in ASCII mode, you can damage the file. Check to make sure you have selected the proper mode, and do so manually if you think Automatic mode is not working correctly.

## Telnet

*Telnet* is a terminal emulation program for TCP/IP networks that runs on TCP/IP port 23. Telnet enables you to connect to a server and run commands on that server as if you were sitting right in front of it. 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 special user name and password to run Telnet.

Telnet is mostly used nowadays to control web servers remotely. This is rather important because web servers often need extra care and attention. Suppose you're the administrator for your company's web server. You are sitting at home when you get a call. There is a problem with your company's web page: a hacker has broken into your web server and replaced the web page with a picture of someone . . . ah . . . *fabricly challenged*. You can use Telnet to connect remotely to the web server and remove the offending page, and then run processes and administer the web server without ever having left your comfy chair. A wonderful capability for the overworked system admin!

## Telnet Servers and Clients

A Telnet server enables users to log onto a host computer and perform tasks as if they're working on the remote computer itself. Users can access the host through the Telnet server from anywhere in the world using a Telnet client. When you create a Telnet server, you can also create a web page that handles server management. Most web server software will do this for you. For instance, IIS enables you to manage an IIS web server via a secured web page.

A Telnet client is the computer from which you log onto the remote server. To use Telnet from the client, you must have the proper permissions. If you do not have a web site that will handle the remote connection for you, you can select from a number of terminal emulators that enable you to operate from a GUI.

## Configuring a Telnet Client

When you configure a Telnet client, you must provide the host name, your user logon name, and the password. As I mentioned previously, you must have permission to access the server to use Telnet.

**Host Name** 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.mhtechd.com`.

**Login Name** 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 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.

**Password** 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.

## 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 a serious flaw—it has no security. Telnet passwords as well as data are transmitted in clear text and are thus easily hacked. To that end, a new (well, newer) TCP application has now replaced Telnet on most real-world servers: *Secure Shell*, better known by its initials, *SSH*. 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, are encrypted. SSH also uses TCP port 22 instead of Telnet's port 23. Figure 15-22 shows the popular Windows SSH tool, Cygwin, running on a Windows system.

SSH has come into vogue for many other traditionally unsecured applications such as FTP. Secure FTP (SFTP) is FTP running over an encrypted SSH connection. SFTP typically uses SSH's port 22, although many SFTP servers may change this to a non-standard port such as 199. SSH even works with some rather offbeat protocols such as Secure Copy Protocol (SCP), a secure replacement for the ancient command line RCP (remote copy program) that allows for file transfer between two systems.

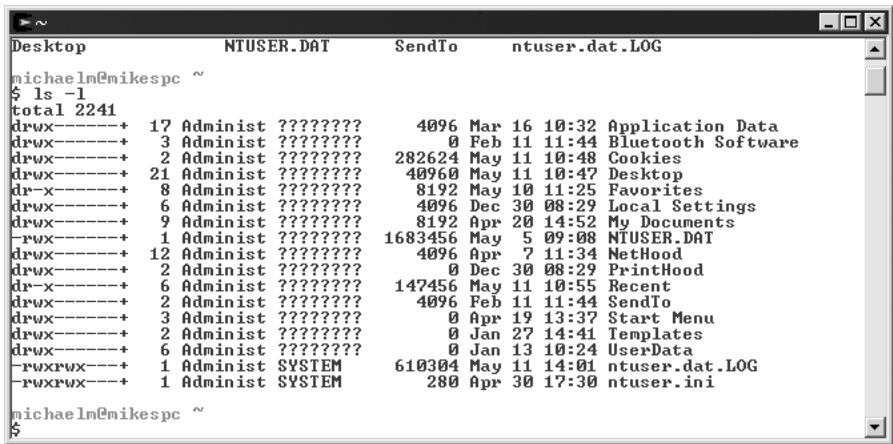


Figure 15-22 The Cygwin program

# Chapter Review

## Questions

1. What device directs incoming network protocol packets from one LAN to another based on OSI Network layer information stored in the incoming packets?  
A. Hub  
B. Switch  
C. Bridge  
D. Router
2. To route packets, a router must have at least \_\_\_\_\_ interfaces.  
A. One  
B. Two  
C. Three  
D. Four
3. What device connects two LANs that use different hardware?  
A. Gateway  
B. Switch

- C. Bridge
  - D. Router
4. The IP address of the router interface that connects to your LAN is called a(n):
- A. Subnet mask
  - B. IP address
  - C. DNS
  - D. Default gateway
5. What device translates a system's IP address into another IP address before sending it out to the larger network?
- A. A firewall
  - B. A NAT
  - C. A router
  - D. A proxy server
6. What device, acting at the Application level, translates a port number to a different port number to add more security to the system?
- A. A firewall
  - B. A NAT
  - C. A router
  - D. A proxy server
7. The protocol developed by Netscape for transmitting private documents over the Internet is known as
- A. SSS
  - B. SSA
  - C. SSL
  - D. NSSL
8. Which of the following are key benefits of web-based mail? (Select all that apply.)
- A. You can use a third-party application, like Microsoft Outlook, to download your e-mail.
  - B. You can access your e-mail from anywhere in the world from a computer with a browser and an Internet connection.
  - C. They are completely spam-free.
  - D. They're great for making throw-away accounts.
9. An SSL URL connection starts with
- A. HTTP
  - B. WWW



- C. FTP
  - D. HTTPS
10. Joe likes to surf the Web instead of doing his work. He calls you and tells you he can't connect to the Internet. Which of the following would be one of the first utilities you would use to diagnose his problem?
- A. NBSTAT
  - B. TRACEROUTE
  - C. ROUTE PRINT
  - D. PING

## Answers

1. **D.** Routers direct incoming network protocol packets from one LAN to another based on OSI Network layer information stored in the incoming packets.
2. **B.** To route these packets, a router by definition must have at least two interfaces, although some routers have three or more depending on the needs of the network.
3. **A.** A gateway, in contrast to a regular router, connects two LANs that use different hardware. You wouldn't refer to a router that connects two Ethernet LANs, for example, as a gateway. A router that connects an Ethernet LAN to a DSL router or to a Token Ring LAN is an example of a gateway.
4. **D.** The default gateway is the IP address of the router interface that connects to your LAN. That interface is called the local side or the local interface on your router.
5. **B.** A NAT translates a system's IP address into another IP address before sending it out to the larger network. NATs work at the Network layer.
6. **D.** Proxy servers translate port numbers to a different port number to add more security to the system. Proxy servers work at the Application layer.
7. **C.** Secure Sockets Layer (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.
8. **B, D.** You can access a web-based e-mail account from any browser on any machine connected to the Internet. These accounts are great for making throw-away e-mail addresses.
9. **D.** URLs that use an SSL connection start with HTTPS instead of HTTP.
10. **D.** One of the first things to try is the PING utility; type **PING** at a command prompt. Pinging is a great way to discover connectivity problems.

