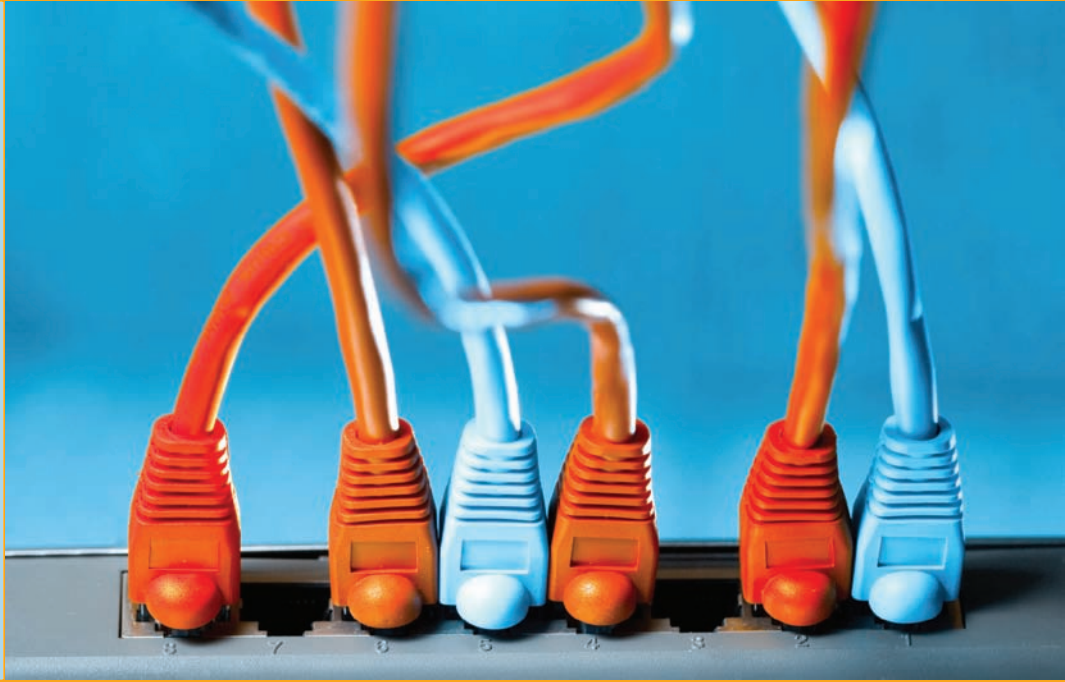


chapter  
8

# The Wonderful World of Routing

*"Youngsters read it, grown men understand it, and old people applaud it."*

—MIGUEL DE CERVANTES



## In this chapter, you will learn how to

- Explain how routers work
- Describe dynamic routing technologies
- Install and configure a router successfully

The true beauty, the amazing power of TCP/IP lies in one word: routing. Routing enables us to interconnect individual LANs into WANs. Routers, those magic boxes that act as the interconnection points, have all the built-in smarts to inspect incoming packets and forward them toward their eventual LAN destination. Routers are for the most part automatic. They require very little in terms of maintenance once their initial configuration is complete because of their capability to talk to each other to determine the best way to send IP packets. The goal of this chapter is to take you into the world of routers and show you exactly how they do this.

The chapter discusses how routers work, including an in-depth look at different types of Network Address Translation (NAT), and then dives into an examination of various dynamic routing protocols. You'll learn about distance vector protocols, Routing Information Protocol (RIP), and Border Gateway Protocol (BGP), among others. The chapter finishes with the nitty-gritty of installing and configuring a router successfully. Not only will you understand how routers work, you should be able to set up a basic home router and diagnose common router issues by the end of this chapter.

# Historical/Conceptual

## ■ How Routers Work

A **router** is any piece of hardware or software that forwards packets based on their destination IP address. Routers work, therefore, at Layer 3, the Network layer.

Classically, routers are dedicated boxes that contain at least two connections, although many routers contain many more connections. In a larger business setting, for example, you might see a Cisco 2600 Series device, one of the most popular routers ever made. The 2611 router shown in Figure 8.1 has two connections (the other connections are used for maintenance and configuration). The two “working” connections are circled. One port leads to one network; the other to another network. The router reads the IP addresses of the packets to determine where to send the packets. (More on how that works in a moment.)

Most techs today get their first exposure to routers with the ubiquitous home routers that enable your PC to connect to a DSL receiver or cable modem (Figure 8.2). The typical home router is more than it appears at first glance, usually combining into that one box a router, a switch, and other features as well, such as a firewall to help protect your network from unwanted intrusion.

Figure 8.3 shows the electronic diagram for a two-port Cisco router, whereas Figure 8.4 shows the diagram for a Linksys home router. Note that both boxes connect two networks. The big difference is that the LAN side of the Linksys home router connects immediately to the built-in switch. That’s convenient! You don’t have to buy a separate switch to connect multiple computers to the cable modem or DSL receiver. Many new techs look at that router, though, and say “it has five ports,” when in reality it can only connect two networks. The extra physical ports belong to the built-in switch.

All routers, big and small, plain or bundled with a switch, examine packets and then send the packets to the proper destination. Let’s take a look at that process in more detail now.



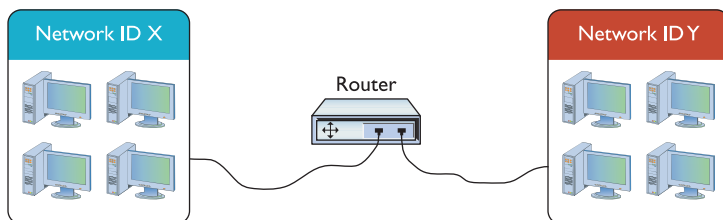
• Figure 8.1 Cisco 2611 router



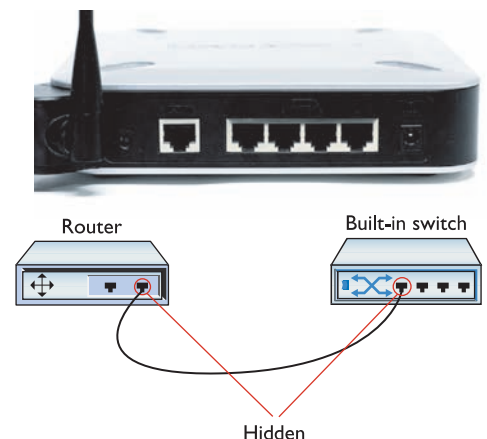
• Figure 8.2 Business end of a typical home router



See Chapter 17, “Protecting Your Network,” for an in-depth look at firewalls and other security options.



• Figure 8.3 Cisco router diagram



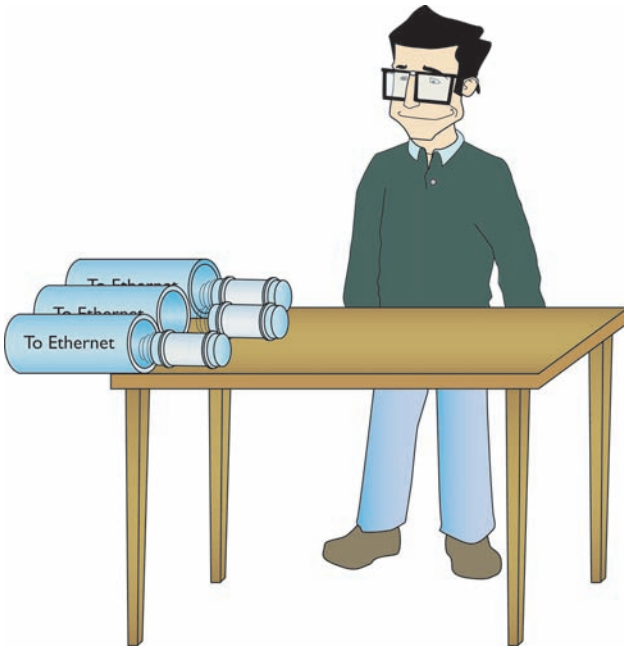
• Figure 8.4 Linksys home router diagram

## Test Specific

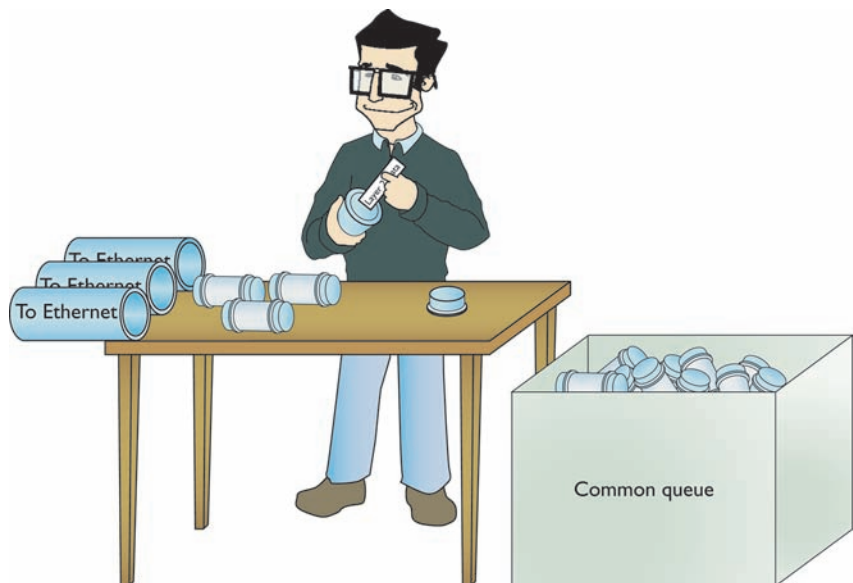
### Routing Tables

Routing begins as packets come into the router for handling (Figure 8.5). The router immediately strips off any of the Layer 2 information and drops the resulting IP packet into a queue (Figure 8.6). The important point to make here is that the router doesn't care where the packet came *from*. Everything is dropped into the same queue based on the time it arrived.

The router inspects each packet's destination IP address and then sends the IP packet out the correct port. Each router comes with a **routing table** that tells the router exactly where to send the packets. Figure 8.7 shows the routing table for a typical home router. This router has only two ports internally: one that connects to whichever service provider you use to bring the Internet into your home (cable/DSL/fiber or whatever)—labeled as Interface WAN in the table—and another one that connects to a built-in four-port switch—labeled LAN in the table. Figure 8.8 is a diagram for the router. The routing table is the key to the process of forwarding packets to their proper destination.



• **Figure 8.5** Incoming packets



• **Figure 8.6** All incoming packets stripped of Layer 2 data and dropped into a common queue



## Cross Check

### What's Up with Layer 2?

You first read about routers roughing up packets and relieving them of all their Layer 2 change way back in Chapter 2, “Building a Network with the OSI Model,” so check your memory now. What defines the Layer 2 information? How is it assigned? How does it interact with Layer 1?

Each row in the routing table defines a single route. Each column identifies specific criteria. Reading Figure 8.7 from left to right shows the following:

- **Destination LAN IP** A defined network ID. Every network ID directly connected to one of the router's ports is always listed here.
- **Subnet Mask** To define a network ID, you need a subnet mask (described in Chapter 7).

Your router uses the combination of the destination LAN IP and subnet mask to see if a packet matches that route. For example, if you had a packet with the destination 10.12.14.26 coming into the router, the router would check the network ID and subnet mask. It would quickly determine that the packet matches the first route shown in Figure 8.7. The other two columns in the routing table then tell the router what to do with the packet.

- **Gateway** The IP address for the **next hop** router; in other words, where the packet should go. If the outgoing packet is for a network ID that's not directly connected to the router, the Gateway column tells the router the IP address of a router to which to send this packet. That router then handles the packet and your router is done (you count on well-configured routers to make sure your packet will get to where it needs to go!). If the network ID is directly connected, then you don't need a gateway. So this is set to 0.0.0.0 or to the IP address of the directly connected port.
- **Interface** Tells the router which of its ports to use. On this router it uses the terms “LAN” and “WAN.” Other routing tables will use the port's IP address or some other type of abbreviation.

The router compares the destination IP address on a packet to every listing in the routing table and then sends the packet out. There is no top-down or bottom-up to this comparison process; every line is read and then the router decides what to do. The most important trick to reading a routing table is to remember that a zero (0) means “anything.” For example, in Figure 8.7 the first route's destination LAN IP is 10.12.14.0. You can compare that to the subnet mask (255.255.255.0) to confirm that this is a /24 network.

Routing Table Entry List

Destination LAN IP	Subnet Mask	Gateway	Interface
10.12.14.0	255.255.255.0	0.0.0.0	LAN
76.30.4.0	255.255.254.0	0.0.0.0	WAN
0.0.0.0	0.0.0.0	76.30.4.1	WAN

Figure 8.7 Routing table from a home router

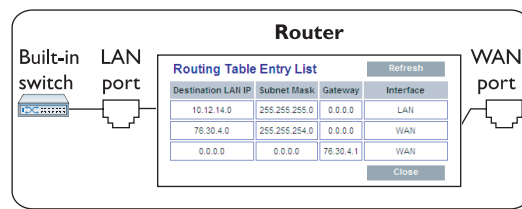
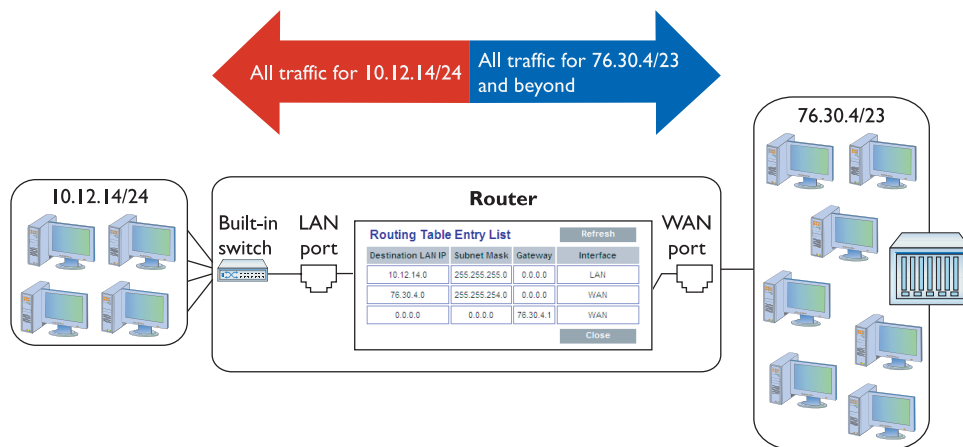


Figure 8.8 Electronic diagram of the router



• **Figure 8.9** The network based on the routing table

This tells you that any value (between 1 and 254) is acceptable for the last value in the 10.12.14/24 network ID.

Routing tables tell you a lot about how the network connects. From just this single routing table, for example, the diagram in Figure 8.9 can be drawn.

So how do I know the 76.30.4.1 port connects to another network? The third line of the routing table shows the default route for this router, and every router has one. (There's one exception to this. See the Tech Tip



### Tech Tip

#### Top o' the Internet

*There are two places where you'll find routers that do not have default routes: private (as in not on the Internet) networks where every router knows every other router, and the monstrous "Tier One" backbone, where you'll find routers that make the main connections of the Internet. Every other router has a default route.*

"Top o' the Internet.") This line says (Any destination address) (with any subnet mask) (forward it to 76.30.4.1) (using my WAN port).

Destination LAN IP	Subnet Mask	Gateway	Interface
0.0.0.0	0.0.0.0	76.30.4.1	WAN

The default route is very important because this tells the router exactly what to do with every incoming packet *unless* another line in the routing table gives another route. Excellent! Interpret the other two lines of the routing table in Figure 8.7 in the same fashion: (Any packet for the 10.12.14.0) (/24 network ID) (don't use a gateway) (just ARP on the LAN interface to get the MAC address and send it directly to the recipient).

Destination LAN IP	Subnet Mask	Gateway	Interface
10.12.14.0	255.255.255.0	0.0.0.0	LAN

(Any packet for the 76.30.4.0) (/23 network ID) (don't use a gateway) (just ARP on the WAN interface to get the MAC address and send it directly to the recipient).

Destination LAN IP	Subnet Mask	Gateway	Interface
76.30.4.0	255.255.254.0	0.0.0.0	WAN

I'll let you in on a little secret. Routers aren't the only devices that use routing tables. In fact, every node (computer, printer, TCP/IP-capable soda dispenser, whatever) on the network also has a routing table.

At first this may seem silly—doesn't every computer only have a single Ethernet connection and therefore all data traffic has to go out that port? First of all, many computers have more than one NIC. (These are called multihomed computers. See the Tech Tip "Multihoming" for more details.) But even if your computer has only a single NIC, how does it know what to do with an IP address like 127.0.0.1? Secondly, every packet sent out of your computer uses the routing table to figure out where the packet should go,



whether directly to a computer or to your gateway. Here's an example of a routing table in Windows. This machine connects to the home router described earlier, so you'll recognize the IP addresses it uses.

```
C:\>route print
=====
Interface List
0x1 ..... MS TCP Loopback interface
0x2 ...00 11 d8 30 16 c0 ..... NVIDIA nForce Networking Controller
=====

Active Routes:
Network Destination        Netmask          Gateway         Interface      Metric
0.0.0.0                    0.0.0.0          10.12.14.1      10.12.14.201   1
10.12.14.0                 255.255.255.0    10.12.14.201    10.12.14.201   1
10.12.14.201              255.255.255.255  127.0.0.1       127.0.0.1      1

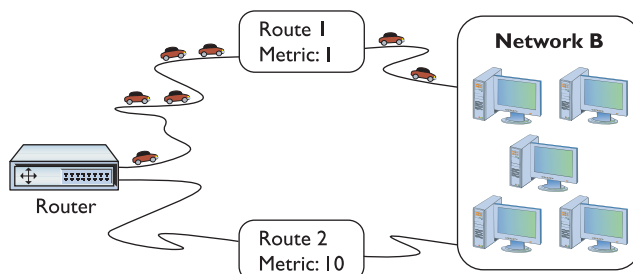
127.0.0.0                 255.0.0.0        127.0.0.1       127.0.0.1      1
169.254.0.0               255.255.0.0      10.12.14.201    10.12.14.201   20
224.0.0.0                 240.0.0.0        10.12.14.201    10.12.14.201   1
255.255.255.255          255.255.255.255  10.12.14.201    10.12.14.201   1
Default Gateway:          10.12.14.1
=====

Persistent Routes:
None
C:\>
```

Unlike the routing table for the typical home router you saw in Figure 8.7, this one seems a bit more complicated, if for no other reason than it has a lot more routes. My PC has only a single NIC, though, so it's not quite as complicated as it might seem at first glance. Take a look at the details. First note that my computer has an IP address of 10.12.14.201, /24 subnet, and 10.12.14.1 as the default gateway.

You should note two differences in the columns from what you saw in the previous routing table. First, the interface has an actual IP address—10.12.14.201, plus the loopback of 127.0.0.1—instead of the word “LAN.” Second—and this is part of the magic of routing—is something called the metric.

A **metric** is just a relative value that defines the “cost” of using this route. The power of TCP/IP is that a packet can take more than one route to get to the same place. Figure 8.10 shows a networked router with two routes to the same place. The router has a route to network X with a metric of 1 using router X, and a second route to network X using router Y with a metric of 10.



• **Figure 8.10** Two routes to the same network



## Tech Tip

### Multihoming

*Multihoming is using more than one NIC in a system, either as a backup or to speed up a connection. Systems that can't afford to go down (like Web servers) often have two NICs that share the same IP address. If one NIC goes down, the other kicks in automatically.*



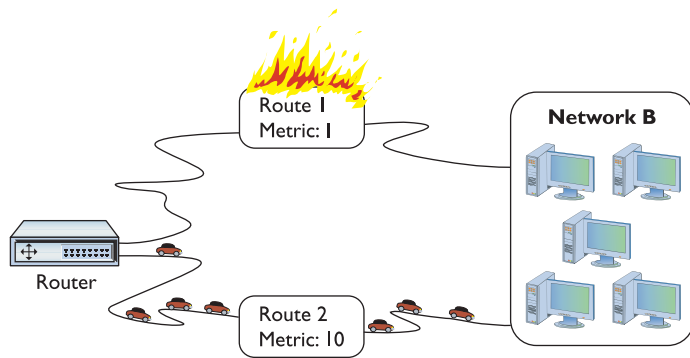
## Tech Tip

### Viewing Routing Tables in Linux and OS X

*Every modern operating system gives you tools to view a computer's routing table. Most techs use the command line or terminal window interface—often called simply terminal—because it's fast. To see your routing table in Linux or in Macintosh OS X, for example, just type `netstat -r` at a terminal. (The command will work in Windows as well.) In Windows, try **route print** as an alternative.*



When a router has more than one route to the same network, it's up to the person in charge of that router to assign a different metric for each route.



• **Figure 8.11** When a route no longer works, the router automatically switches.

Lowest routes always win. In this case the router will always use the route with the metric of 1, unless that route suddenly stopped working. In that case, the router would automatically switch to the route with the 10 metric (Figure 8.11). This is the cornerstone of how the Internet works! The entire Internet is nothing more than a whole bunch of big, powerful routers connected to lots of other big, powerful routers. Connections go up and down all the time and routers (with multiple routes) constantly talk to each other, detecting when a connection goes down and automatically switching to alternate routes.

I'll go through this routing table one line at a time. Remember, every address is compared to every line in the routing table before it goes out, so it's no big deal if the default route is at the beginning or the end.

This line defines the default route. (Any destination address) (with any subnet mask) (forward it to my default gateway) (using my NIC) (Cost of 1 to use this route).

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	10.12.14.1	10.12.14.201	1

The next line defines the local connection. (Any packet for the 10.12.14.0) (/24 network ID) (don't use a gateway) (just ARP on the LAN interface to get the MAC address and send it directly to the recipient) (Cost of 1 to use this route).

Network Destination	Netmask	Gateway	Interface	Metric
10.12.14.0	255.255.255.0	10.12.14.201	10.12.14.201	1

So, if a gateway of 10.12.14.201 here means "don't use a gateway," why put a number in here at all? Local connections don't use a default gateway, though every routing table has a gateway column. The Microsoft folks had to put *something* there, thus they put the IP address of the NIC. That's why the gateway address is the same as the interface address. Personally, I've always found this confusing. Wouldn't calling the gateway 0.0.0.0, as you saw in the previous routing table, make more sense? Better yet, wouldn't it be even better if we just said, "This is a local call so no gateway is needed"? Well, this is Windows XP. In Windows Vista the gateway value for local connections just says "on-link"—a much more accurate description! Part of the joy of learning routing tables is getting used to how different operating systems deal with issues like these.

Okay, on to the third line. This one's easy. Anything addressed to this machine should go right back to it through the loopback (127.0.0.1).

Network Destination	Netmask	Gateway	Interface	Metric
10.12.14.201	255.255.255.255	127.0.0.1	127.0.0.1	1

This next line is another loopback, but look carefully. Earlier you learned that only 127.0.0.1 is the loopback, but according to this route, any 127/8 address is the loopback.

Network	Destination	Netmask	Gateway	Interface	Metric
	127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1

The next route says that any addresses in the 169.254/16 network ID are part of the LAN (remember, whenever the gateway and interface are the same it's a local connection). If your computer uses Dynamic Host Configuration Protocol (DHCP) and can't get an IP address, this route would enable you to communicate with other computers on the network who hopefully are also having the same DHCP problem. Note the high metric.

Network	Destination	Netmask	Gateway	Interface	Metric
	169.254.0.0	255.255.0.0	10.12.14.201	10.12.14.201	20

This is the multicast address range. Odds are good you'll never need it, but most operating systems put it in automatically.

Network	Destination	Netmask	Gateway	Interface	Metric
	224.0.0.0	240.0.0.0	10.12.14.201	10.12.14.201	1

This line defines the default IP broadcast. If you send out an IP broadcast (255.255.255.255), your NIC knows to send it out to the local network.

Network	Destination	Netmask	Gateway	Interface	Metric
	255.255.255.255	255.255.255.255	10.12.14.201	10.12.14.201	1



## Try This!

### Getting Looped

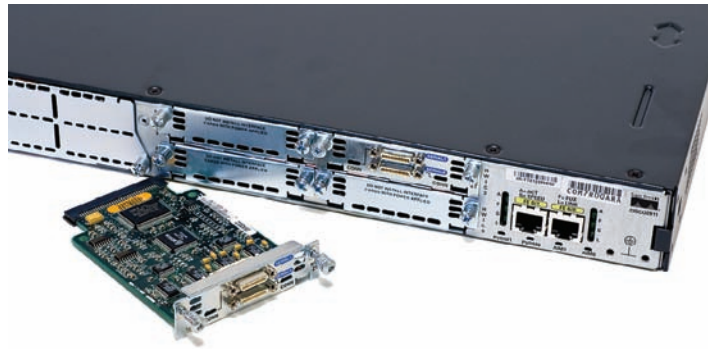
Try pinging any 127/8 address to see if they loop back like 127.0.0.1. What happens?

## Freedom from Layer 2

Routers enable you to connect different types of network technologies. You now know that routers strip off all of the Layer 2 data from the incoming packets, but thus far you've only seen routers that connect to different Ethernet networks—and that's just fine with routers. But routers can connect almost anything that stores IP packets. Not to take away from some very exciting upcoming chapters, but Ethernet is not the only networking technology out there. Once you want to start making long-distance connections, Ethernet disappears and technologies with names like Data Over Cable Service Interface Specification (DOCSIS) (cable modems), Frame Relay, and Asynchronous Transfer Mode (ATM) take over. These technologies are not Ethernet. Their frames don't use MAC addresses, although just like Ethernet frames they do store IP packets.

Most serious (that is, not home) routers enable you to add ports. You buy the router and then you snap in different types of ports depending on your needs. Note the Cisco router in Figure 8.12. Like most Cisco routers, it comes with removable modules. If you're connecting Ethernet to ATM, you buy an Ethernet module and an ATM module. If you're connecting Ethernet to a DOCSIS (cable) network, you buy an Ethernet module and a DOCSIS module.





• **Figure 8.12** Modular Cisco router

## Network Address Translation

The ease of connecting computers together using TCP/IP and routers creates a rather glaring security risk. If every computer on a network must have a unique IP address and TCP/IP applications enable you to do something on a remote computer, what's to stop a malicious programmer from writing a program that does things on your computer that you don't want done? All he'd need is the IP address for your computer and he could target you from anywhere on the network. Now expand this concept to the Internet. A computer sitting in Peoria can be attacked by a program run from Bangkok as long as both computers connect directly to the Internet. And this happens all the time.

Security is one problem; two other problems are the finite number of IP addresses available and their cost. IP addresses, once thought limitless, are quickly running out. Most of the available IP numbers have already been allocated, making public IP addresses more and more rare. Anything that's rare costs more money. Legitimate, public IP addresses are therefore more expensive to come by. Wouldn't it be great to lease only one public IP address instead of tens or even hundreds for every computer on your network?

Routers running some form of **Network Address Translation (NAT)** hide the IP addresses of computers on the LAN, but still enable those computers to communicate with the broader Internet. NAT addresses the problems of IP addressing on the Internet. NAT has become extremely common and is heavily in use, so it's important to learn how it works. Note that many routers offer NAT as a feature *in addition to* the core capability of routing. NAT is not routing, but a separate technology. With that said, you are ready to dive into how NAT works to protect computers connected by router technology and conserve IP addresses as well.

### The Setup

Here's the situation. You have a LAN with eight computers that need access to the Internet. With classic TCP/IP and routing, several things have to happen. First, you would need to get a block of legitimate, unique, expensive IP addresses from an Internet service provider (ISP). You could call up an ISP and

purchase a network ID, say 1.2.3.136/29. Second, you would assign an IP address to each computer and to the LAN connection on the router. Third, you'd assign the IP address for the ISP's router to the WAN connection on the local router, such as 1.2.4.1. After everything was configured, the network would look like Figure 8.13. All of the clients on the network have the same default gateway (1.2.3.137). This router, called a **gateway router** (or simply a *gateway*), acts as the default gateway for a number of client computers.

This style of network mirrors how computers in LANs throughout the world connected to the Internet for the first 20 years of the Internet, but the three major problems of security, running out of IP addresses, and the expense of leasing more than one address worsened as more and more computers connected.

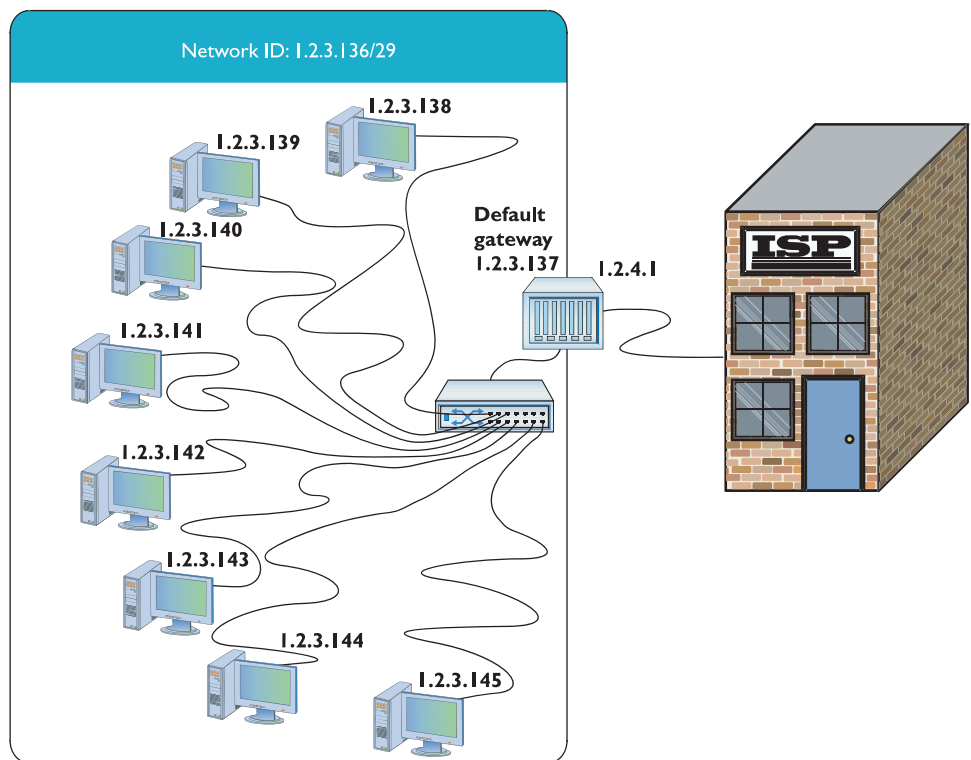
NAT solves all these issues. NAT is a simple concept: you replace the source IP address of a computer with the source IP address from the router on outgoing packets. More complex NAT methods use TCP/IP port numbers to increase the number of computers using a single routable IP address.

## Translating IP Addresses

With basic NAT, you tell your NAT-capable router to replace the source IP address of a computer with the source IP address from the router on outgoing packets. The outside world never sees the IP addresses used by the internal network, which enables you to use any network ID you wish for the internal network. In most cases, you use a private IP address range, such as 192.168.1.0/24.

The traditional IP address-translating NAT comes in a variety of flavors, with names like Source NAT, Destination NAT, Static NAT, and Dynamic NAT. Not surprisingly, these names get shortened to acronyms that add to the confusion: SNAT, DNAT, SNAT, and DNAT. Ugh! Here's the scoop.

With **Source NAT** and **Destination NAT**, the source or destination IP addresses, respectively, get translated by the NAT-capable router. Many NAT-capable routers can do both.



• Figure 8.13 Network setup



NAT replaces the source IP address of a computer with the source IP address from the router on outgoing packets. NAT is performed by NAT-capable routers.



Despite the many uses in the industry of the acronym SNAT, the CompTIA Network+ exam uses SNAT for Static NAT exclusively.



### Tech Tip

#### Two More SNATs

*As if Static NAT and Source NAT didn't stir up enough problems with the acronym soup, Microsoft and Cisco use SNAT to describe two other technologies proprietary to their companies. For Microsoft, SNAT refers to Secure Network Address Translation (also called SecureNAT), a driver extension that enables multiple computers to use a single routable IP address with a Windows server, among other things. Most networking folks refer to the features of SecureNAT more generically as overloaded NAT.*

*Cisco uses the term SNAT for Stateful NAT or Stateful Failover Network Address Translation. Cisco's SNAT simply enables multiple routers to do NAT redundantly, so that if one router goes down, the other(s) picks up the traffic.*

**Static NAT (SNAT)** maps a single routable (that is, not private) IP address to a single machine, enabling you to access that machine from outside the network. The NAT keeps track of the IP address or addresses and applies them permanently on a one-to-one basis with computers on the network.

With **Dynamic NAT**, in contrast, many computers can share a pool of routable IP addresses that number fewer than the computers. The NAT might have 10 routable IP addresses, for example, to serve 40 computers on the LAN. LAN traffic uses the internal, private IP addresses. When a computer requests information beyond the network, the NAT doles out a routable IP address from its pool for that communication. Dynamic NAT is also called *Pooled NAT*. This works well enough—unless you're the unlucky 11th person to try to access the Internet from behind the company NAT—but has the obvious limitation of still needing many true, expensive, routable IP addresses.

### Adding Ports to the Mix

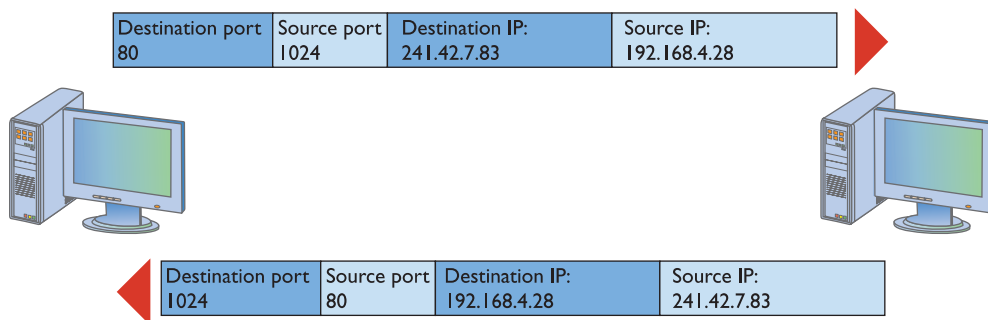
Translating IP addresses in one or more of the ways just described makes NAT useful, but still doesn't quite solve the inherent problems with TCP/IP addressing. TCP/IP communication involves more than just IP addresses, though; using TCP/IP port numbers in conjunction with IP addresses solves the dual problems of security and limited IP addresses handily. Let's look at port numbers first, and then turn to the implementations of using ports with NAT with overloaded NAT and port forwarding.

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

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

Each packet has two ports assigned, a destination port and an ephemeral port. The **destination port** is a fixed, predetermined number that defines the function or session type. Common TCP/IP session types use destination port numbers in the range 0–1023. The **ephemeral port** is an arbitrary number generated by the sending computer; the receiving computer uses the ephemeral port as a destination address so that the sending computer knows which application to use for the returning packet. Ephemeral ports usually fall in the 1024–5000 range, but this varies slightly among the different operating systems.

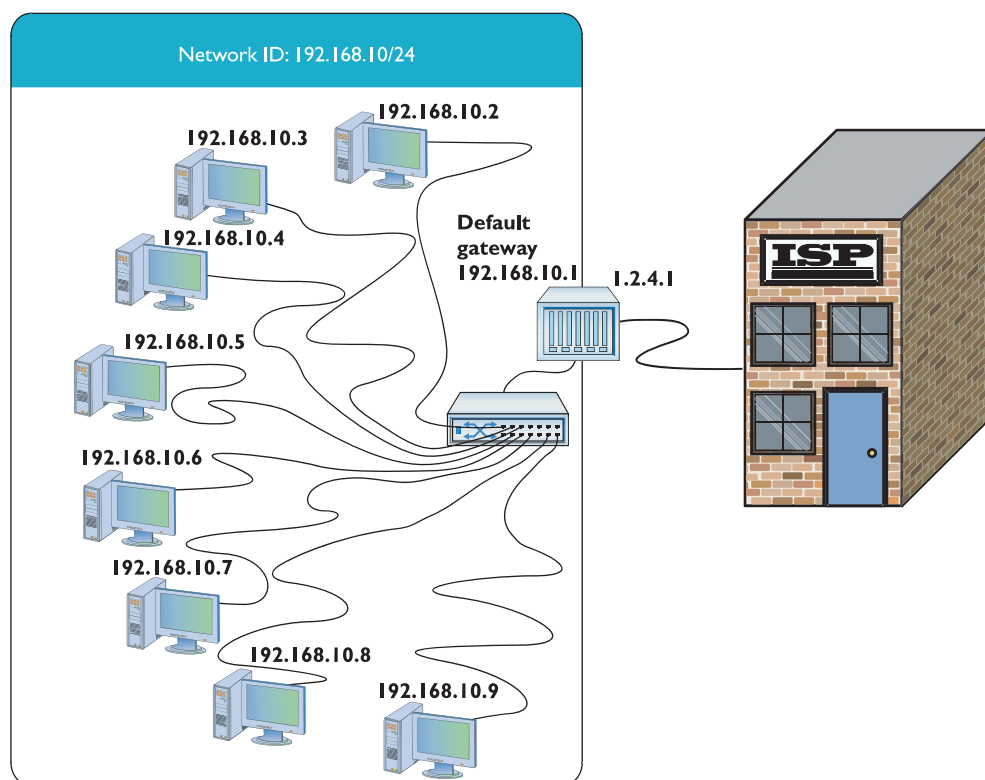
Figure 8.14 shows two packets from a conversation between a Web client and a Web server. The top shows a TCP packet with the client requesting a Web page from the Web server. Note the destination port of 80 and the ephemeral port of 1024. The bottom packet shows the Web server starting to send back the Web page using port 1024 as the destination port and port 80 as the source port. Note that this is not called an ephemeral port for the return trip, because the server does not generate it. The server simply uses the ephemeral port given to it by the client system.



• **Figure 8.14** Ports at work

You'll learn quite a bit more about ports in the next chapter, but you now know enough of the concept to go back to NAT and appreciate the importance of port numbers in this process.

**NAT, Overloaded** In the most popular type of NAT, called **overloaded NAT**, a single public IP address is shared by a number of computers that, in most cases, share a private network ID. To set up a small, eight-port LAN connected to a router, like you did earlier, you'd use private IP addresses rather than public ones for each of the computers on the network. But you'd only need one public address, the one assigned to the router's Internet connection. Figure 8.15 changes Figure 8.13 slightly, this time using the network



• **Figure 8.15** Redone network IDs; nodes in the LAN use private IP addresses internally

ID range of 192.168.10/24 for the LAN. By noting the source ephemeral port of each computer making connections, the number of possible connections goes up tremendously.

Let's zero in on what happens inside the gateway router when a computer on the LAN needs information from beyond the LAN. This router has overloaded NAT capability enabled. One of the computers inside the network, 192.168.10.202, needs to send a packet to a faraway computer, 12.43.65.223. The 12.43.65.223 address is clearly not a part of the 192.168.10.0

network, so this packet will be going out the gateway and into the Internet.

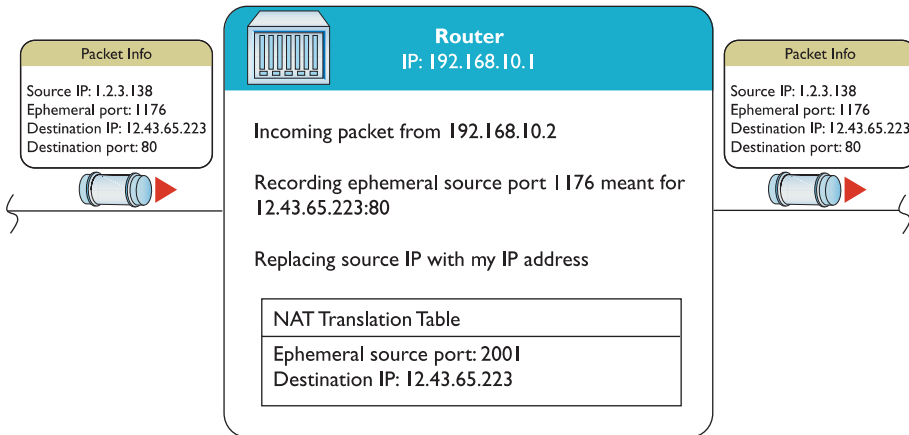
As the outgoing IP packet enters the router, the router replaces the sending computer's source IP address with its own public IP address. It then adds the destination IP address and the source ephemeral port to a special database called the **NAT translation table** (Figure 8.16).

When the receiving system sends the packet back, it reverses the IP addresses and ports. The overloaded NAT router compares the incoming destination port and source IP address to the entry in the NAT translation table

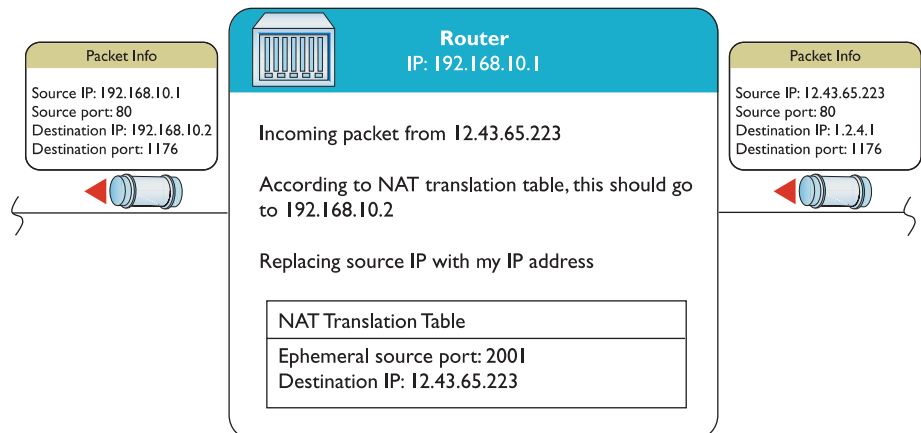
to determine which IP address to put back on the packet (Figure 8.17). It then sends the packet to the correct computer on the network.

Overloaded NAT takes care of all of the problems facing a network exposed to the Internet. You don't have to use legitimate Internet IP addresses on the LAN and the IP addresses of the computers behind the routers are invisible and protected from the outside world.

Overloaded NAT is so common that the term NAT almost always means overloaded NAT.



• Figure 8.16 NATing a packet



• Figure 8.17 Updating the packet



Since the router is revising the packets and recording the IP address and port information already, why not enable it to handle ports more aggressively? Enter port forwarding, stage left.

**Port Forwarding** **Port forwarding** hides a port number from the wilds of the Internet, enabling public servers to work behind a NAT router. Port forwarding gives servers the protection of NAT while still allowing access to that server. Suppose you have a Web server behind a NAT router. You know from earlier in the book that Web servers look for incoming port 80 addresses. A port-forwarding router recognizes all incoming requests for a particular port and then forwards those requests to an internal IP address. To support an internal Web server, the router is configured to forward all port 8080 packets to the internal Web server at port 80, as shown in Figure 8.18.

**Port Address Translation** Different manufacturers use the term **Port Address Translation (PAT)** to refer to both overloaded NAT and port forwarding, though not at the same time. The Cisco router in Figure 8.19, for example, calls overloaded NAT Port Address Translation (PAT).



The CompTIA Network+ exam follows the Cisco definition of Port Address Translation, making the term synonymous with overloaded NAT.

**Single Port Forwarding - Mozilla Firefox**

http://10.12.14.1/SingleForwarding.htm

**LINKSYS**  
A Division of Cisco Systems, Inc.

Firmware Version: V1.1.03

**Wireless N Gigabit Security Router with VPN WRT5440N**

**Firewall** | Setup | Wireless | Firewall | VPN | QoS | Administration | IPS | L2 Switch | Status

Basic Settings | IP Based ACL | Internet Access Policy | **Single Port Forwarding** | Port Range Forwarding | More... >>

**Single Port Forwarding**

Application	External Port	Internal Port	Protocol	IP Address	Enabled
HTTP	8080	80	TCP	10.12.14.150	<input checked="" type="checkbox"/>
FTP	21	21	TCP	10.12.14.0	<input type="checkbox"/>
FTP-Data	20	20	TCP	10.12.14.0	<input type="checkbox"/>
Telnet	23	23	TCP	10.12.14.0	<input type="checkbox"/>
SMTP	25	25	TCP	10.12.14.0	<input type="checkbox"/>
TFTP	69	69	UDP	10.12.14.0	<input type="checkbox"/>
finger	79	79	TCP	10.12.14.0	<input type="checkbox"/>
NTP	123	123	UDP	10.12.14.0	<input type="checkbox"/>
POP3	110	110	TCP	10.12.14.0	<input type="checkbox"/>
NNTP	119	119	TCP	10.12.14.0	<input type="checkbox"/>
SNMP	161	161	UDP	10.12.14.0	<input type="checkbox"/>
CVS	2401	2401	TCP	10.12.14.0	<input type="checkbox"/>
SMS	2701	2701	TCP	10.12.14.0	<input type="checkbox"/>
SMS-rmctd	2702	2702	TCP	10.12.14.0	<input type="checkbox"/>
			TCP	10.12.14.0	<input type="checkbox"/>

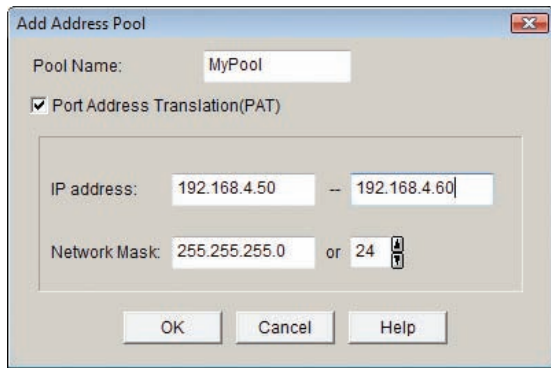
**Use the Single Port Forwarding screen when you want to open specific services (that use single port). This allows users on the Internet to access this server by using the WAN port address and the matched external port number. When users send these types of request to your WAN port IP address via the Internet, the NAT Router will forward those requests to the appropriate servers on your LAN.**

[More...](#)

**Save Settings** **Cancel Changes**

Done

• **Figure 8.18** Setting up port forwarding on a home router



• **Figure 8.19** Configuring Port Address Translation on a Cisco router



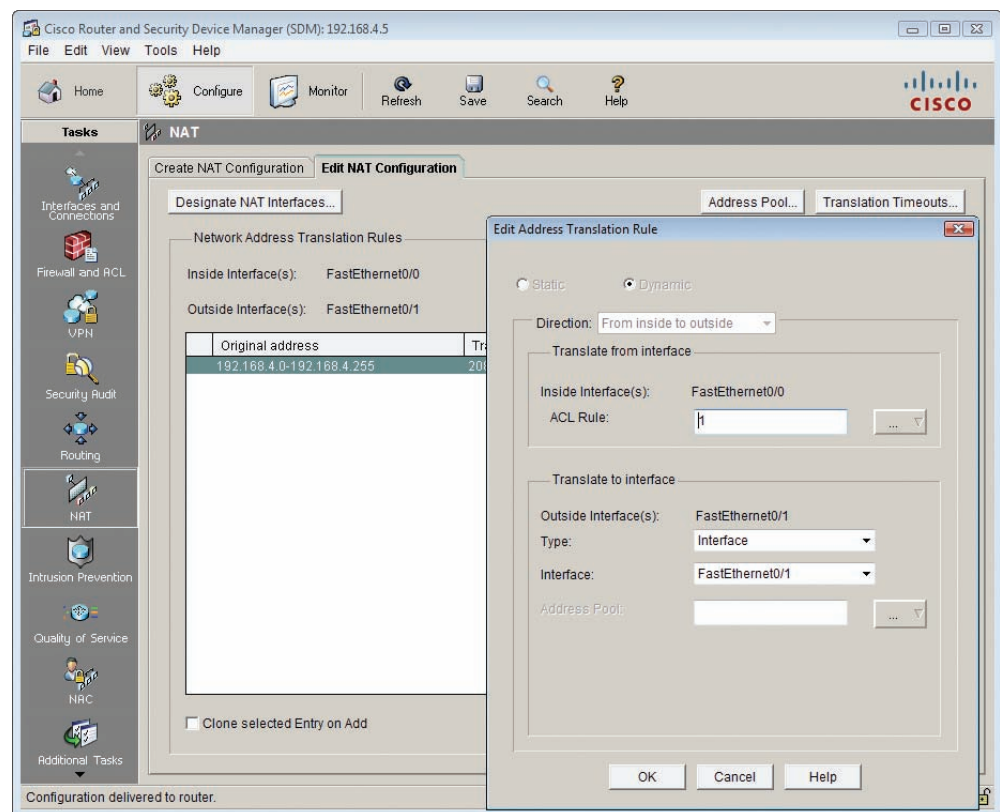
• **Figure 8.20** NAT setup on home router

## Configuring NAT

Configuring NAT on home routers is a no-brainer as these boxes invariably have NAT turned on automatically. Figure 8.20 shows the screen on my home router for NAT. Note the radio buttons that say Gateway and Router.

By default the router is set to Gateway, which is Linksys-speak for “NAT is turned on.” If I wanted to turn off NAT, I would set the radio button to Router.

Commercial-grade routers use NAT more explicitly, enabling you to do Static NAT, Pooled NAT, port forwarding, and more. Figure 8.21 shows a router configuration screen on a Cisco router.



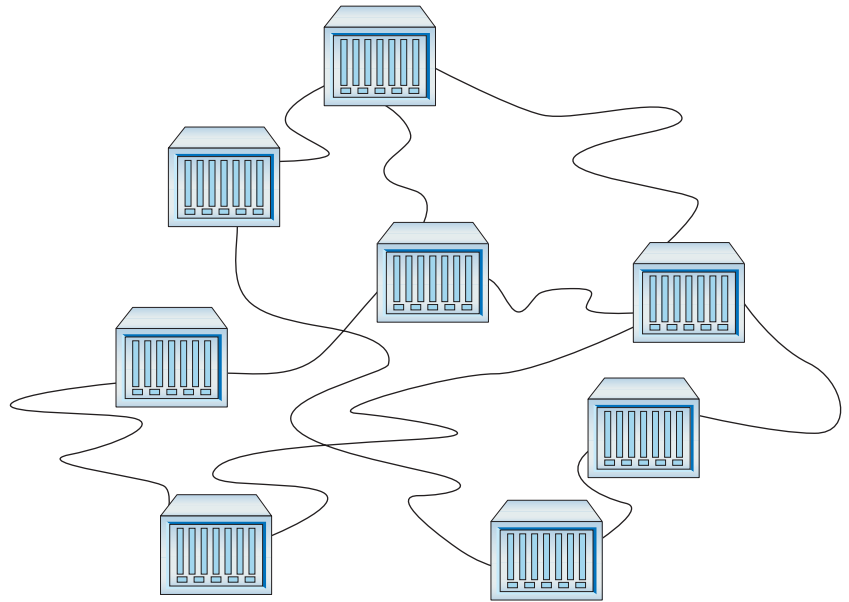
• **Figure 8.21** Configuring NAT on a commercial-grade router

## ■ Dynamic Routing

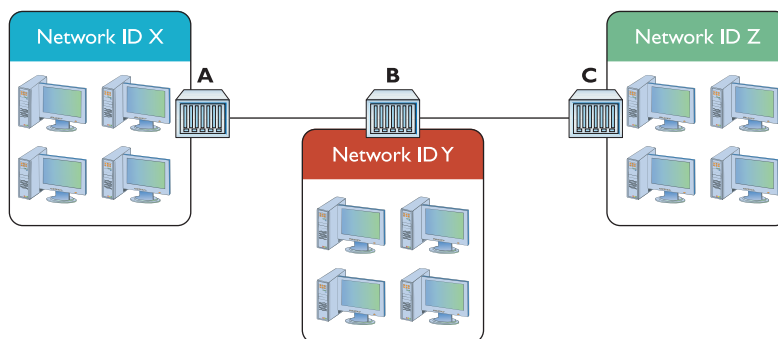
Based on what you've read up to this point, it would seem that routes in your routing tables come from two sources: either they are manually entered or they are detected at setup by the router. In either case, a route seems to be a static beast, just sitting there and never changing. And based on what you've seen so far, that is absolutely true. Routers have **static routes**. But most routers also have the capability to update their routes *dynamically*, assuming they're provided with the extra smarts in the form of **dynamic routing** protocols.

If you've been reading carefully you might be tempted at this point to say: "Why do I need this dynamic routing stuff? Don't routers use metrics so I can add two or more routes to another network ID in case I lose one of my routes?" Yes, but metrics really only help when you have direct connections to other network IDs. What if your routers look like Figure 8.22?

Do you really want to try to set up all these routes statically? What happens when something changes? Can you imagine the administrative nightmare? Why not just give routers the brainpower to talk to each other so that they know what's happening not only to the other directly connected routers but also to routers two or three or more routers away? Each time a packet goes through a router is defined as a **hop**. Let's talk about hops for a moment. Figure 8.23 shows a series of routers. If you're on a computer in Network ID X and you PING a computer in Network ID Y, you go one hop. If you PING a computer in Network ID Z, you go two hops.



• Figure 8.22 Lots of routers



• Figure 8.23 Hopping through a WAN

Routing protocols have been around for a long time and, like any technology, there have been a number of different choices and variants over those years. CompTIA Network+ competencies break these many types of routing protocols into three distinct groups: distance vector, link state, and hybrid. CompTIA obsesses over these different types of routing protocols, so this chapter does too!

## Distance Vector

Distance vector routing protocols were the first to appear in the TCP/IP routing world. The cornerstone of all distance vector routing protocols is some form of total cost. The simplest total cost adds up the hops (the hop count) between a router and a network, so if you had a router one hop away from a network, the cost for that route would be 1; if two hops away, the cost would be 2.

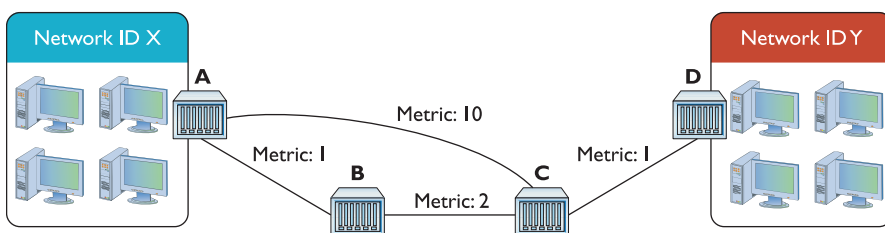
All network connections are not equal. A router might have two one-hop routes to a network—one using a fast connection and the other using a slow connection. Administrators set the metric of the routes in the routing table to reflect the speed. So the slow single-hop route, for example, might be given the metric of 10 rather than the default of 1 to reflect the fact that it's slow. So the total cost for this one-hop route is 10, even though it's only one hop. Don't assume a one-hop route always has a cost of 1.

**Distance vector** routing protocols calculate the total cost to get to a particular network ID and compare that cost to the total cost of all the other routes to get to that same network ID. The router then chooses the route with the lowest cost.

For this to work, routers using a distance vector routing protocol transfer their entire routing table to other routers in the WAN. Each distance vector routing protocol has a maximum number of hops that a router will send its routing table to keep traffic down.

Assume that you have four routers connected as shown in Figure 8.24. All of the routers have static routes set up between each other with the

metrics as shown. You add two new networks, one that connects to Router A and the other to Router D. For simplicity, call them Network ID X and Network ID Y. A computer on one network wants to send packets to a computer on the other network, but the routers in between Routers A and D don't yet know the two new network IDs. That's when distance vector routing protocols work their magic.



• **Figure 8.24** Getting a packet from Network ID X to Network ID Y? No clue!

Because all of the routers use a distance vector routing protocol, the problem gets solved quickly. At a certain defined time interval (usually 30 seconds or less) the routers begin sending each other their routing tables (the routers each send their entire routing table, but for simplicity just concentrate on the two network IDs in question). On the first iteration, Router A sends its route to Network ID X to Routers B and C. Router D sends its route to Network ID Y to Router C (Figure 8.25).

This is great—Routers B and C now know how to get to Network ID X and Router C can get to Network ID Y; but there’s still no complete path between Network ID X and Network ID Y. That’s going to take another interval. After another set amount of time, the routers again send their now updated routing tables to each other, as shown in Figure 8.26.

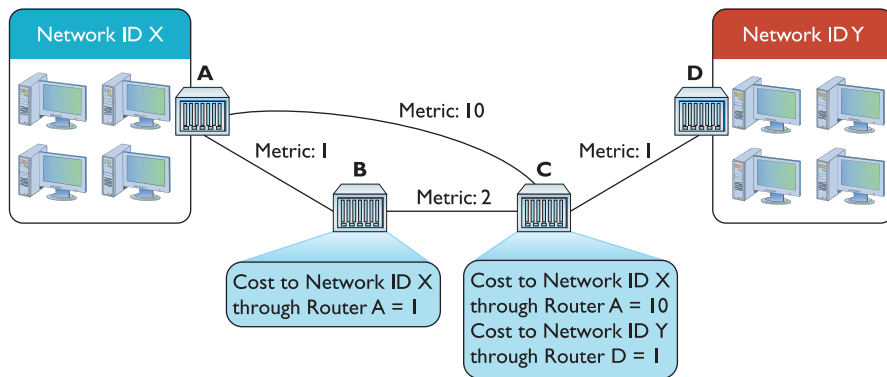
Router A knows a path now to Network ID Y, and Router D knows a path to Network ID X. As a side effect, Router B and Router C have two routes to Network ID X. Router B can get to Network ID X through Router A and through Router C. Similarly, Router C can get to Network ID X through Router A and through Router B. What to do? In cases where the router discovers multiple routes to the same network ID, the distance vector routing protocol deletes all but the route with the lowest total cost (Figure 8.27).

On the next iteration, Routers A and D get updated information about the lower-total-cost hops to connect to Network IDs X and Y (Figure 8.28).

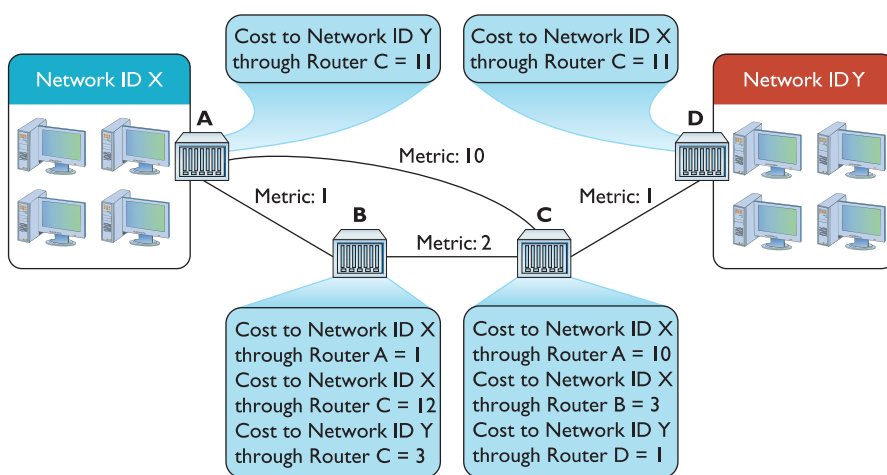
Just as Routers B and C only kept the routes with the lowest cost, Routers A and D keep only the lowest-cost routes to the networks (Figure 8.29).

Now Routers A and D have a lower-cost route to Network IDs X and Y. They’ve removed the higher-cost routes and begin sending data.

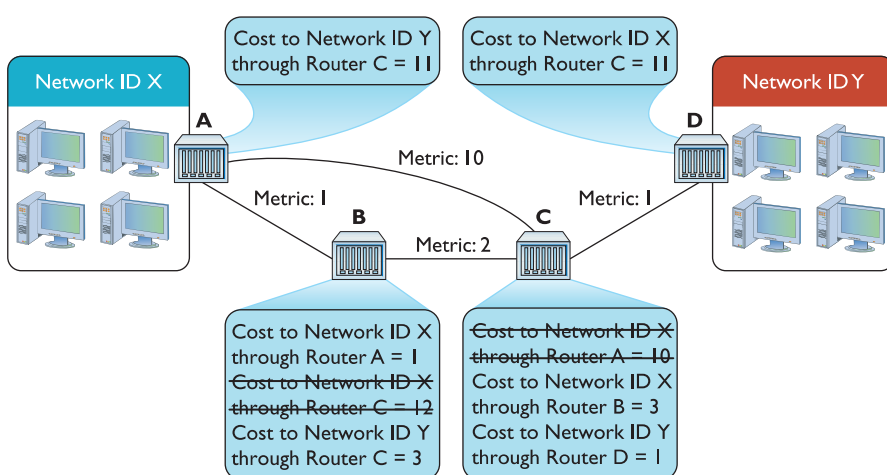
At this point if routers were human they’d realize that each router has all the information about the network and stop sending each other routing tables. Routers using distance vector routing protocols, however, aren’t that smart. The routers continue to



• **Figure 8.25** Routes updated

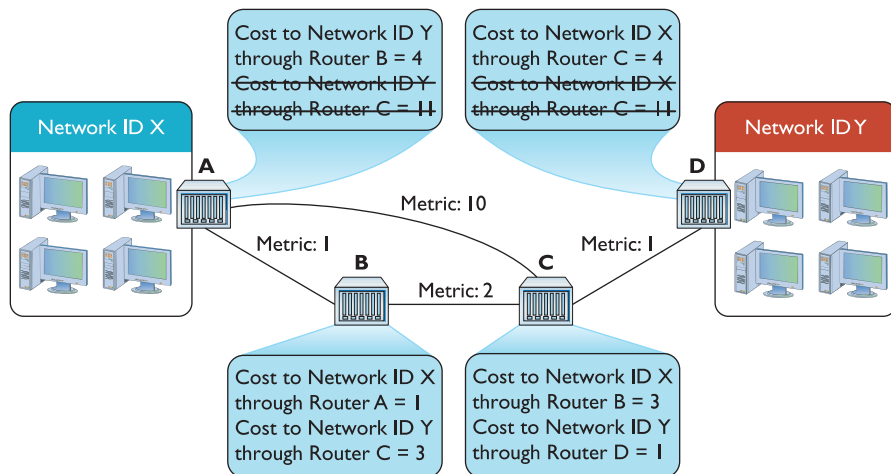


• **Figure 8.26** Updated routing tables

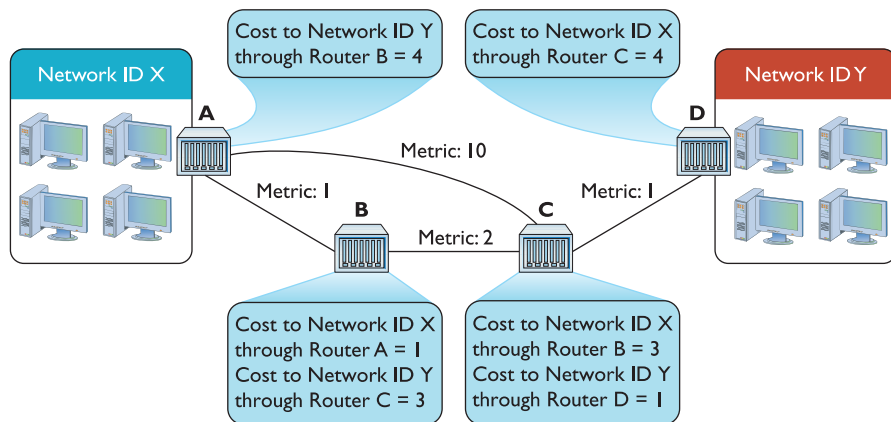


• **Figure 8.27** Deleting higher-cost routes





• **Figure 8.28** Argh! Multiple routes!



• **Figure 8.29** Last iteration

send their complete routing tables to each other, but because there's no new information, the routing tables stay the same.

At this point the routers are in **convergence** (also called *steady state*), meaning the updating of the routing tables for all the routers has completed. Assuming nothing changes in terms of connections, the routing tables will not change. In this example, it takes three iterations to reach convergence.

So what happens if the route between Routers B and C breaks? The routers have deleted the higher-cost routes, only keeping the lower-cost route that goes between Routers B and C. Does this mean Router A can no longer connect to Network ID Y and Router D can no longer connect to Network ID X? Yikes! Yes it does. At least for a while.

Routers that use distance vector routing protocols continue to send to each other their entire routing table at regular intervals. After a few iterations, Routers A and D will once again know how to reach each other, though through the once-rejected slower connection.

Distance vector routing protocols work fine in a scenario such as the previous one that has only four routers. Even if you lose a router, a

few minutes later the network returns to convergence. But imagine if you had tens of thousands of routers (the Internet). Convergence could take a very long time indeed. As a result, a pure distance vector routing protocol works fine for a network with a few (<10) routers, but isn't good for large networks.

Routers can use one of three distance vector routing protocols: RIPv1, RIPv2, or BGP.

## RIPv1

The granddaddy of all distance vector routing protocols is the **Routing Information Protocol (RIP)**. The first version of RIP—called **RIPv1**—dates from the 1980s, although its predecessors go back all the way to the beginnings of the Internet in the 1960s. RIP has a maximum hop count of 15 so your router will not talk to another router more than 15 routers away.

This ended up being a problem because a routing table request could literally loop all the way around back to the initial router.

RIPv1 sent out an update every 30 seconds. This also turned into a big problem because every router on the network would send its routing table at the same time, causing huge network overloads.

As if these issues weren't bad enough, RIPv1 didn't know how to use Classless Inter-Domain Routing (CIDR) subnets and could only route classful subnets. Plus RIPv1 routers had no authorization, leaving them open to hackers sending false routing table information. RIP needed an update.

## RIPv2

**RIPv2** is the current version of RIP, adopted in 1994. It works the same way as RIPv1, but fixes many of the problems. CIDR support has been added, updates are set at random intervals, and authentication is built into the protocol. (The maximum hop count of 15 continues to apply to RIPv2.)

Most routers still support RIPv2, but RIP's many problems, especially the time to convergence for large WANs, makes it obsolete for all but small, private WANs that consist of a few routers. The growth of the Internet demanded a far more robust dynamic routing protocol. That doesn't mean RIP rests in peace! RIP is both easy to use and easy for manufacturers to implement in their routers, so most routers, even home routers, have the ability to use RIP (Figure 8.30).



• **Figure 8.30** Setting RIP in a home router

## BGP

The explosive growth of the Internet in the 1980s required a fundamental reorganization in the structure of the Internet itself. The entities that govern how the Internet works do so in a highly decentralized fashion. Even the organized groups, such as the Internet Society (ISOC) and its better known committees—Internet Assigned Numbers Authority (IANA) and Internet Engineering Task Force (IETF)—are made up of many different individuals, companies, and government organizations from across the globe. The reorganization process took time and many meetings.

What came out of the reorganization eventually was a multitiered structure. At the top of the structure sits many Autonomous Systems. An **Autonomous System (AS)** is one or more networks that are governed by a single protocol within that AS. Figure 8.31 illustrates the central structure of the Internet.



• **Figure 8.31** The Internet

```
Router2811(config)#router bgp ?
<1-65535> Autonomous system number

Router2811(config)#router bgp 1902|
```

• **Figure 8.32** Configuring a Cisco router to use an ASN

Autonomous Systems do not use IP addresses, but rather use a special globally unique Autonomous System Number (ASN) assigned by the IANA. Originally a 16-bit number, the current ASNs are 32 bits, displayed as two 16-bit numbers separated by a dot. So, 1.33457 would be a typical ASN. Just as you would assign an IP address to a router, you would configure the router to use or be the ASN assigned by the IANA. See Figure 8.32.

Autonomous Systems communicate with each other using a protocol, called generically an Exterior Gateway Protocol (EGP). The network or networks within an AS communicate with protocols as well, called generically Interior Gateway Protocols (IGPs).

Many protocols are used *within* Autonomous Systems, such

as RIP, but the Internet has settled on one protocol for communication between each AS: the **Border Gateway Protocol (BGP)**. BGP is the glue of the Internet, connecting all of the Autonomous Systems. The current version of BGP is BGP-4.

The CompTIA Network+ Exam Objectives list BGP as a distance vector routing protocol, but it's really somewhat different. BGP doesn't have the same type of routing table as you've seen so far. Instead, BGP routers are manually configured (these types of connections aren't the type that go down very often!) and advertise information passed to them from different Autonomous Systems' **edge routers**—that's what the AS-to-AS routers are called. BGP forwards these advertisements that include the ASN and other very non-IP items.

BGP also knows how to handle a number of situations unique to the Internet. If a router advertises a new route but that route isn't reliable, most BGP routers ignore it. BGP also supports policies that ISPs can use to limit who and how other routers may access them.

BGP is an amazing and powerful dynamic routing protocol, but unless you're working deep in the router room of an AS, odds are good you'll never see it in action. Those who need to connect a few routers together usually turn to a family of dynamic routing protocols that work very differently from distance vector routing protocols.

## Link State

The limitations of RIP motivated the demand for a faster protocol that took up less bandwidth on a WAN. The basic idea was to come up with a dynamic routing protocol that was more efficient than routers that simply sent out their entire routing table at regular intervals. Why not instead simply announce and forward individual route changes as they appeared? That is the basic idea of a **link state** dynamic routing protocol. There are only two link state dynamic routing protocols: OSPF and IS-IS.



### Try This!

#### Discovering the Autonomous System Numbers

You can see the AS for most Web sites by using this handy little Firefox add-on: [www.asnumber.networx.ch](http://www.asnumber.networx.ch). It doesn't work for every Web site but it's still interesting.



There's no reason you can't use BGP within an AS to connect networks. So you can and do run into situations where BGP is both the interior and exterior protocol for an AS. To distinguish between the two uses of the protocol, network folks refer to the BGP on the interior as the *internal BGP* (iBGP); the exterior connection then becomes the *external BGP* (eBGP).



Please remember that in the earlier general distance vector routing example, I chose not to show that every update was an entire routing table! I only showed the changes, but trust me, the entire routing table is transmitted roughly every 30 seconds (with some randomization).

## OSPF

**Open Shortest Path First (OSPF)** is the most commonly used IGP in the entire Internet. Most large Internet users (as opposed to ISPs) use OSPF on their internal networks. Even an AS, while still using BGP on its edge routers, will use OSPF internally because OSPF was designed from the ground up to work within a single AS. OSPF converges dramatically faster and is much more efficient than RIP. Odds are good that if you are using dynamic routing protocols, you're using OSPF.

Before you see OSPF in action, I need to warn you that OSPF is a complex protocol for routers. You won't find OSPF on cheap home routers because making it work takes a lot of computational firepower. But OSPF's popularity and CompTIA's coverage make this an important area for you to understand. The description here, while more than enough to get you through the CompTIA Network+ exam, is still only a light touch on the fascinating world of OSPF.

Let's head back to the four-router setup used to explain RIP, but this time replace RIP with OSPF. Since OSPF is designed to work with the Internet, let's give Router B an upstream connection to the organization's ISP. When you first start up OSPF-capable routers, they send out *link state advertisements (LSAs)*, called *Hello packets*, looking for other OSPF routers (Figure 8.33).

A new router sends out a lot of LSAs when it first starts up. This is called *flooding* and it normally only happens at the first boot.

One of the big differences between OSPF and RIP is the hop cost. While single hops in RIP have a cost of 1 unless manually changed, in OSPF the cost is based on the speed of the link. The formula is

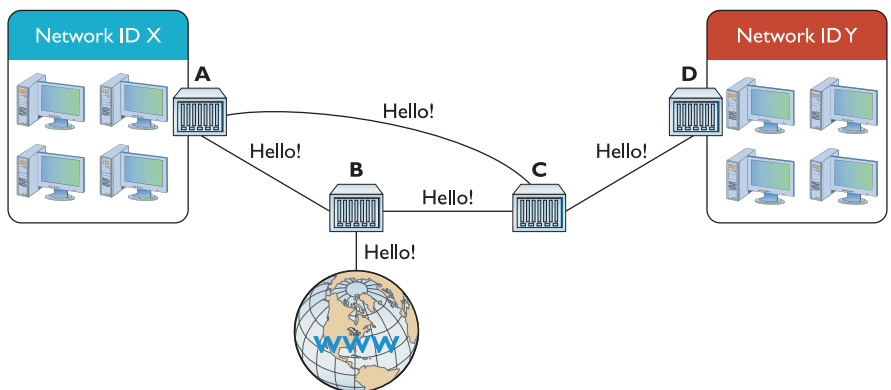
$100,000,000/\text{bandwidth in bps}$

So a 10BaseT link's OSPF cost is  $100,000,000/10,000,000 = 10$ . The faster the bandwidth, the lower the cost. You can override this manually if you wish.

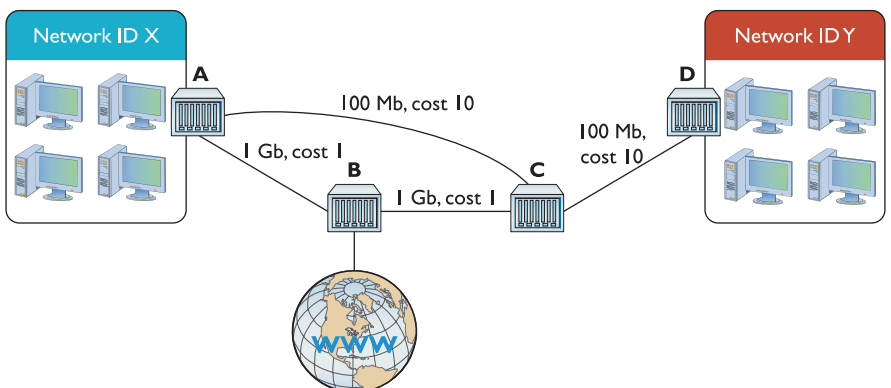
To appreciate the power of OSPF, Figure 8.34 makes Figure 8.33 a bit more complex, adding speeds to each connection. When OSPF routers send LSA Hellos, they exchange this information and update their link state databases.



Even though OSPF takes serious router firepower, it's usually embarrassingly easy to implement. In many cases you just tell your router to use it and it works perfectly.



• Figure 8.33 Hello!



• Figure 8.34 Link states



Even though OSPF Area IDs look like IP addresses, they have nothing to do with IP!

These LSA Hellos are forwarded on to every OSPF router in the network. Every router knows the link state for every other router. This happens in a few seconds.

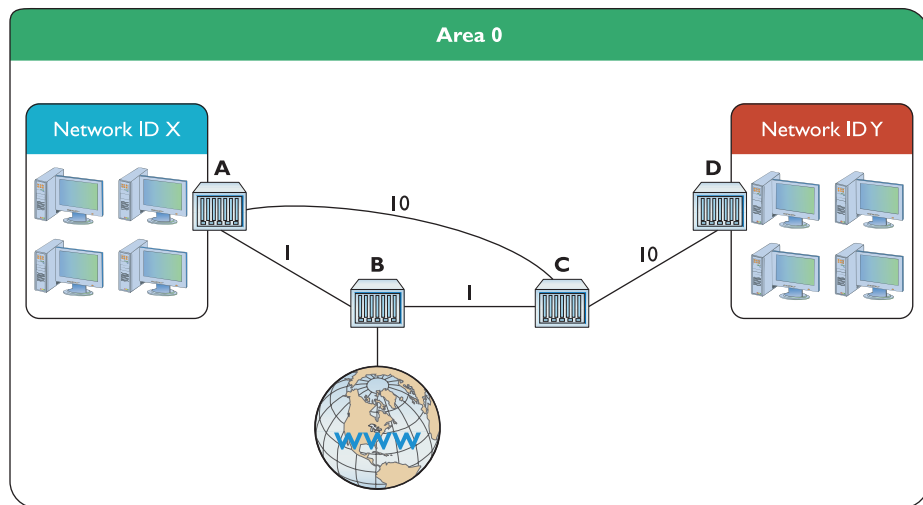
You don't want the routers to flood anywhere beyond your own routers, so every router is assigned an **Area ID**. Area IDs (unfortunately in my opinion) look exactly like IP addresses. Every OSPF router is designed to accept an Area ID that you enter into the routers. In this case all of the routers are given the Area ID of 0.0.0.0. This is commonly called Area 0.

Area 0 is rather important in the world of OSPF. If your network gets more complex, you can make multiple areas. Area 0 is the most important area, however, and thus is called the backbone. In this example, all of the routers are part of Area 0 (Figure 8.35).

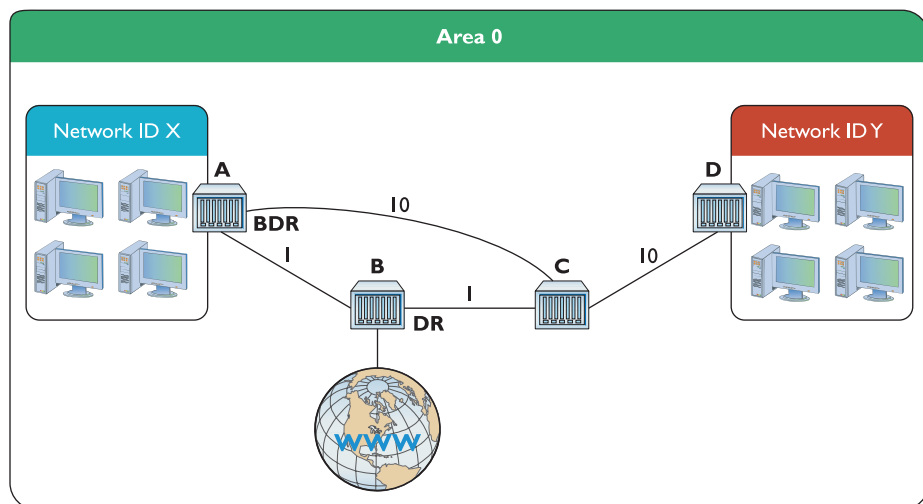
Areas are very important for OSPF. To minimize router traffic, every area has one "El Supremo" router that relays information to all of the other routers in the area. This router is called the **designated router (DR)**. A second router is called the **backup designated router (BDR)** in case the DR isn't available. As the routers first begin to communicate, a DR and BDR election automatically begins. The router with the lowest total priority wins. In this case Router B becomes the DR and Router A becomes the BDR. This actually takes place during the initial Hello packet exchange (Figure 8.36). In most cases you simply let the routers decide, but you can manually set a router as the DR and BDR if you desire (which is rare).

Once the elections take place, it's finally time to get some routes distributed across the area. This part is actually almost boring. Routers A and

B send a separate LSA telling all routers in the area that they are connected to Network IDs X and Y, respectively. These are *not* the entire routing tables, but rather only a single route that is almost instantly dispersed across the routers in the OSPF area (Figure 8.37).

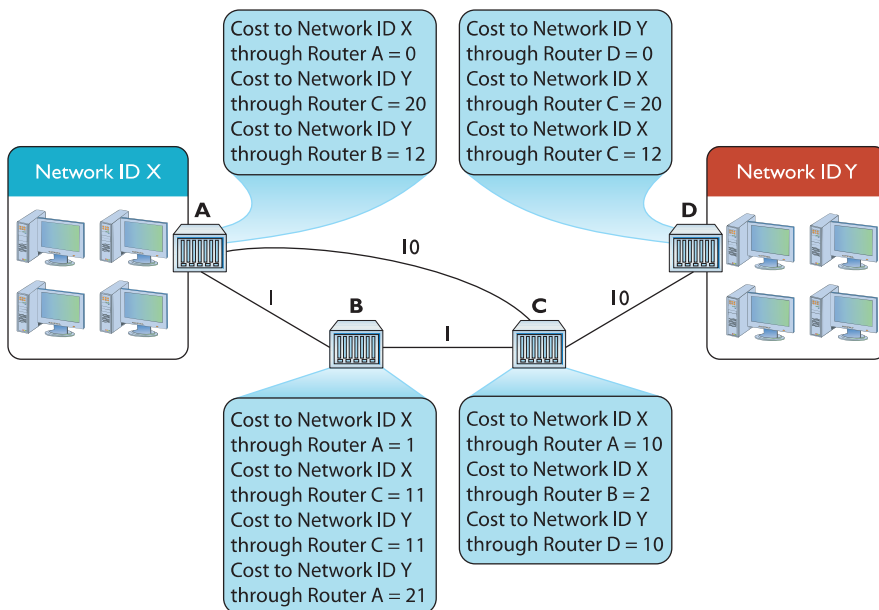


• Figure 8.35 Area defined



• Figure 8.36 DR and BDR



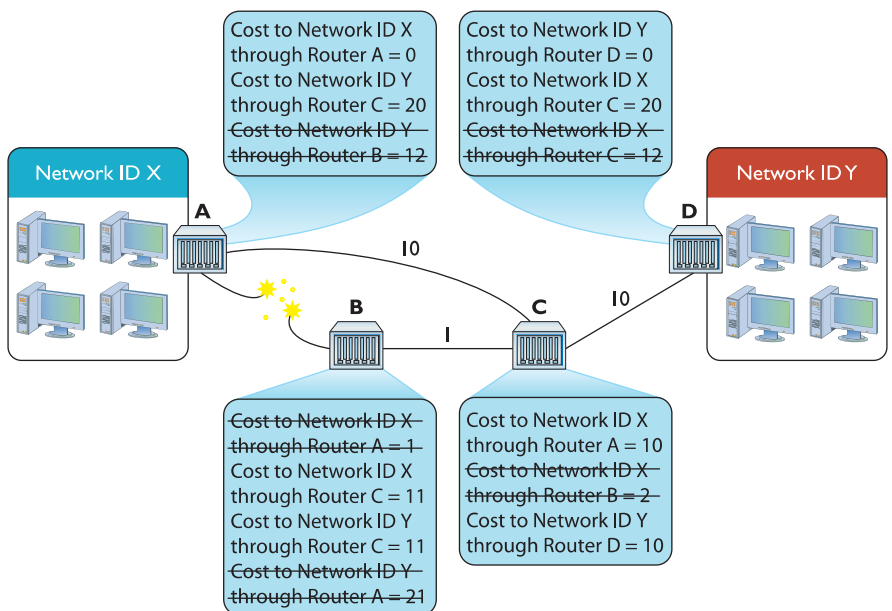


• **Figure 8.37** All routers updated

As you can see, OSPF areas almost instantly gain convergence compared to RIP. Once convergence is reached, all of the routers in the area send each other Hello LSAs every 30 minutes or so unless they detect a break in the link state. Also notice that OSPF routers keep alternate routes to the same network ID.

So what happens when something changes? For example, what if the connection between Routers A and B were to disconnect? In that case both Routers A and B would almost instantly detect the break (as traffic between the two would suddenly stop). Each router would first attempt to reconnect. If this was unsuccessful (over a few seconds), the routers would then send out an LSA announcing the connection between the two was broken (Figure 8.38). Again, this is a single route, not the entire routing table. Each router updates its routing table to remove the route that no longer works.

OSPF isn't popular by accident. It's easy to use, scales to large networks quite well, and is supported by all but the most basic



• **Figure 8.38** Announcing a disconnect



OSPF is a widely used dynamic link-state IGP routing protocol that operates in an AS. OSPF corrects link failures and creates convergence almost immediately, making it the routing protocol of choice in most large enterprise networks. OSPF Version 2 is used for IPv4 networks, and OSPF Version 3 includes updates to support IPv6.

routers. By the way, did I forget to mention that OSPF also supports authentication and that the shortest-path-first method by definition prevents loops? Why would anyone use anything else? Well, OSPF had one problem that wasn't repaired until fairly recently: support for something called IPv6 (see Chapter 13 for details on IPv6). Not to preempt Chapter 13, but IPv6 is a new addressing system for IP that dumps the old 32-bit address, replacing it with a 128-bit address. IPv6 is quickly gaining popularity and will one day replace 32-bit IP addressing. Just for the record, I've been predicting the end of 32-bit IP addressing for so long I'm now afraid to predict anymore when it's going to happen—but it will eventually.

## IS-IS

If you want to use a link state dynamic routing protocol and you don't want to use OSPF, your only other option is **Intermediate System to Intermediate System (IS-IS)**. IS-IS is extremely similar to OSPF. It uses the concept of areas and send-only updates to routing tables. IS-IS was developed at roughly the same time as OSPF and had the one major advantage of working with IPv6 from the start. IS-IS has some adoption with ISPs, but for the most part plays a distant second to the popularity of OSPF. Make sure you know that IS-IS is a link state dynamic routing protocol and if you ever see two routers using it call me as I've never seen IS-IS in action.

## EIGRP—the Lone Hybrid

There is exactly one protocol that doesn't really fit into either the distance vector or link state camp: Cisco's proprietary **Enhanced Interior Gateway Routing Protocol (EIGRP)**. Back in the days when RIP was dominant, there was a huge outcry for an improvement to RIP, but OSPF wasn't yet out. Cisco, being the dominant router company in the world (a crown it still wears to this day), came out with the Interior Gateway Routing Protocol (IGRP), quickly replaced with EIGRP.

EIGRP has aspects of both distance vector and link state protocols, placing it uniquely into its own "hybrid" category. EIGRP is (arguably) fading away in the face of nonproprietary IGP protocols, especially OSPF.

**Table 8.1** Dynamic Routing Protocols

Protocol	Type	IGP or BGP?	Notes
RIPv1	Distance vector	IGP	Old; only used classful subnets
RIPv2	Distance vector	IGP	Supports CIDR
BGP-4	Distance vector	BGP	Used on the Internet, connects Autonomous Systems
OSPF	Link state	IGP	Fast, popular, uses Area IDs (Area 0/backbone)
IS-IS	Link state	IGP	Alternative to OSPF
EIGRP	Hybrid	IGP	Cisco proprietary

## Dynamic Routing Makes the Internet

Without dynamic routing, the complex, self-healing Internet we enjoy today couldn't exist. So many routes come up and go down so often that manually updating static routes simply couldn't keep up. Review Table 8.1 to help you familiarize with the differences between the different types of dynamic routing protocols.

## ■ Working with Routers

Understanding the different ways routers work is one thing. Actually walking up to a router and making it work is a different animal altogether. This section examines practical router installation. Physical installation isn't very complicated. With a home router, you give it power and then plug in connections. With a business-class router, you insert it into a rack, give it power, and plug in connections.

The complex part of installation comes with the specialized equipment and steps to connect to the router and configure it for your network needs. This section, therefore, focuses on the many methods and procedures used to access and configure a router.

The single biggest item to keep in mind here is that while there are a large number of different methods to connect, hundreds of interfaces, and probably millions of different configurations for different routers, the functions are still the same. It doesn't matter if you're using an inexpensive home router or a hyper-powerful Internet backbone router, you always are working to do one main job: connect different network IDs.

Also keep in mind that routers, especially gateway routers, often have a large number of other features that have nothing to do with routing. Gateway routers, since they act as a separator between the computers and "The Big Scary Rest of the Network," are a convenient place for all kinds of handy features like DHCP, protecting the network from intrusion (better known as firewalls), and NAT.

### Connecting to Routers

When a new router comes out of the box, it's not good for very much. You need to somehow plug into that shiny new router and start telling it what you want to do. There are a number of different methods, but one of the oldest (yet still very common) ways is using a special serial connection. This type of connection is almost completely unique to Cisco-brand routers, but Cisco's massive market share makes understanding this type of connection a requirement for anyone who wants to know how to configure routers. Figure 8.39 shows the classic Cisco console cable, more commonly called a *rollover* or **Yost cable**.

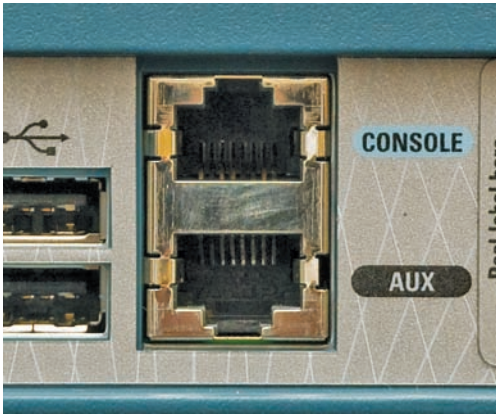
At this point I need to add an important point: switches as well as routers often have some form of configuration interface. Granted, there's nothing for you to configure on a basic switch, but in later chapters you'll discover a number of network features that more advanced



The term Yost cable comes from its creator's name, Dave Yost. For more information visit <http://yost.com/computers/RJ45-serial>.



• **Figure 8.39** Cisco console cable



• **Figure 8.40** Console port



### Tech Tip

#### Terminals and Consoles

A lot of initial router configuration harkens back to the methods used in the early days of networking, when massive mainframe computers were the computing platform available. Researchers used dumb terminals—machines that were little more than a keyboard, monitor, and network connection—to connect to the mainframe and interact. You connect to and configure many modern routers using software that enables your PC to pretend to be a dumb terminal. These programs are called terminal emulators; the screen you type into is called a console.



IOS used to stand for *Internetwork Operating System*, but it's just IOS now with a little trademark symbol.

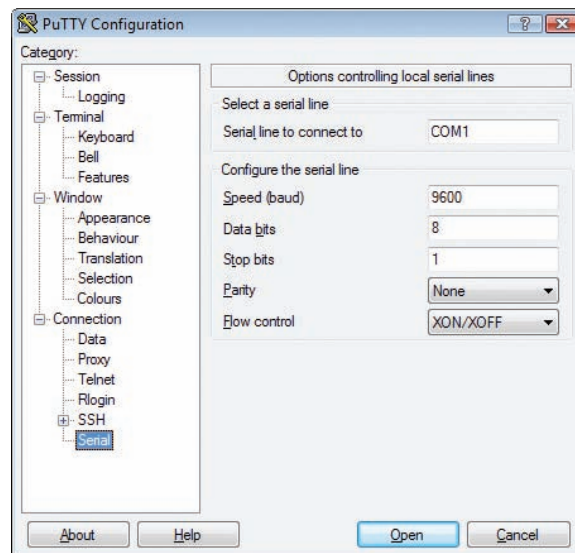
switches need to be configured to use. Both routers and these advanced switches are called **managed devices**. While in this section I use the term *router*, it's important for you to appreciate that all routers and many better switches are all managed devices. The techniques shown here work for both!

When you first unwrap a new Cisco router, you plug the rollover cable into the console port on the router (Figure 8.40) and a serial port on a PC. If you don't have a serial port, then buy a USB-to-serial adapter.

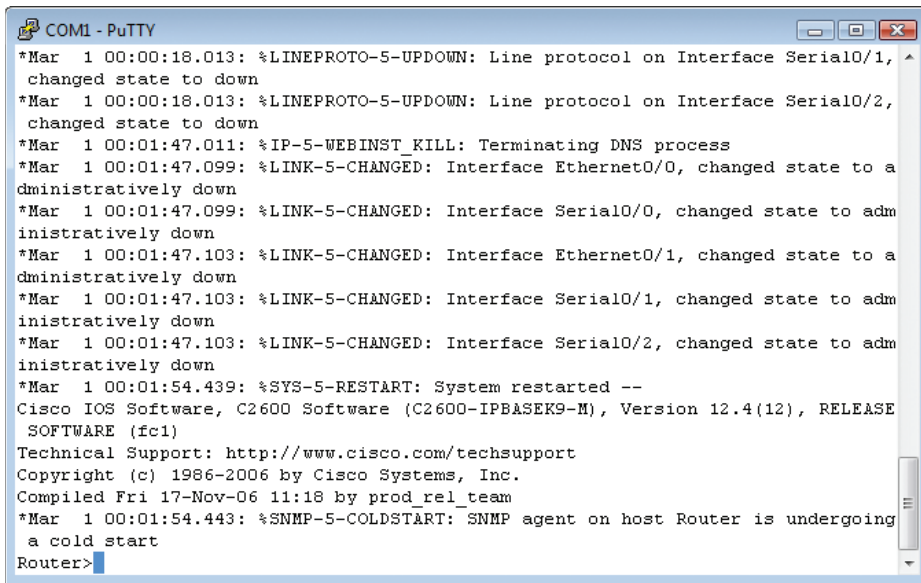
Once you've made this connection, you need to use a terminal emulation program to talk to the router. The two most popular programs are PuTTY ([www.chiark.greenend.org.uk/~sgtatham/putty](http://www.chiark.greenend.org.uk/~sgtatham/putty)) and HyperTerminal ([www.hilgraeve.com/hpte/download.html](http://www.hilgraeve.com/hpte/download.html)). Using these programs requires you to know a little about serial ports but these basic settings should get you connected: 9600 baud, 8 data bits, 1 stop bit, and no parity. Every terminal emulator has some way for you to configure these settings. Figure 8.41 shows these settings using PuTTY.

Now it's time to connect. Most Cisco products run **Cisco IOS**, Cisco's proprietary operating system. If you want to configure Cisco routers you must learn IOS. Learning IOS in detail is a massive job and outside the scope of this book. No worries, Cisco provides a series of certifications to support those who wish to become "Cisco People." While CompTIA Network+ won't challenge you in terms of IOS, it's important to get a taste of how this amazing operating system works.

Once you've connected to the router and started a terminal emulator, you should see the initial router prompt, as shown in Figure 8.42. (If you plugged in and then started the router, you can actually watch the router boot up first.)



• **Figure 8.41** Configuring PuTTY



```
COM1 - PuTTY
*Mar 1 00:00:18.013: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/1, ^
changed state to down
*Mar 1 00:00:18.013: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/2, ^
changed state to down
*Mar 1 00:01:47.011: %IP-5-WEBINST_KILL: Terminating DNS process
*Mar 1 00:01:47.099: %LINK-5-CHANGED: Interface Ethernet0/0, changed state to a
dministratively down
*Mar 1 00:01:47.099: %LINK-5-CHANGED: Interface Serial0/0, changed state to adm
inistratively down
*Mar 1 00:01:47.103: %LINK-5-CHANGED: Interface Ethernet0/1, changed state to a
dministratively down
*Mar 1 00:01:47.103: %LINK-5-CHANGED: Interface Serial0/1, changed state to adm
inistratively down
*Mar 1 00:01:47.103: %LINK-5-CHANGED: Interface Serial0/2, changed state to adm
inistratively down
*Mar 1 00:01:54.439: %SYS-5-RESTART: System restarted --
Cisco IOS Software, C2600 Software (C2600-IPBASEK9-M), Version 12.4(12), RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Fri 17-Nov-06 11:18 by prod_rel_team
*Mar 1 00:01:54.443: %SNMP-5-COLDSTART: SNMP agent on host Router is undergoing
a cold start
Router>
```

• **Figure 8.42** Initial router prompt

This is the IOS user mode prompt—you can't do too much here. To get to the fun you need to enter exec mode. Type **enable**, press ENTER, and the prompt changes to

```
Router#
```

From here IOS gets very complex. For example, the commands to set the IP address for one of the router's ports look like this:

```
Router#configure terminal
Router(config)#interface Ethernet 0/0
Router(config-if)#ip address 192.168.4.10 255.255.255.0
Router(config-if)#^Z
Router#write memory
```

Cisco has long appreciated that initial setup is a bit of a challenge, so a brand-new router will show you the following prompt:

```
Would you like to enter the initial configuration dialog?
[yes/no]?
```

Just follow the prompts and the most basic setup is handled for you.

You will run into Cisco equipment as a network tech, and you will need to know how to use the console from time to time. For the most part, though, you'll access a router—especially one that's already configured—through Web access or network management software.

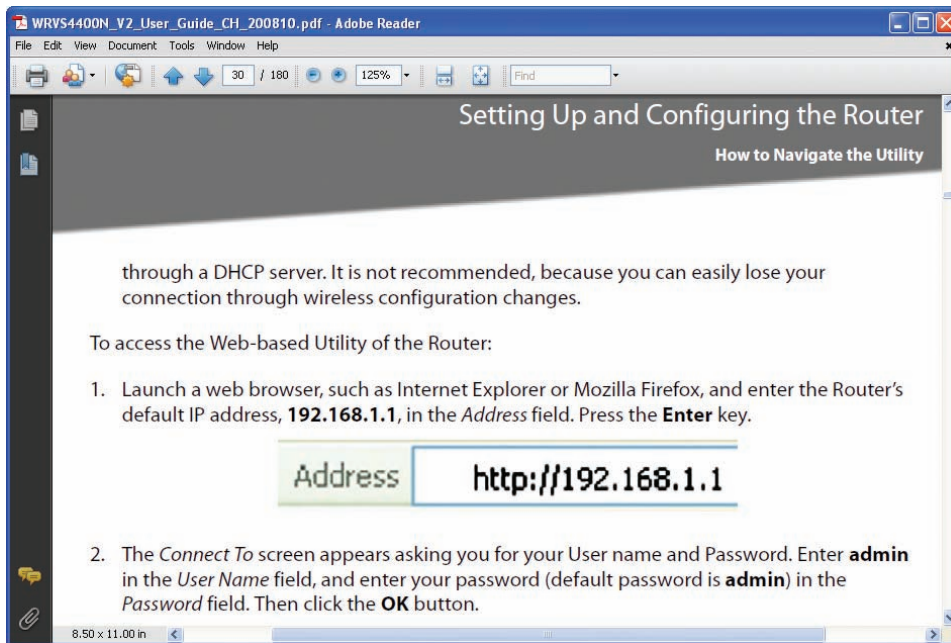
## Web Access

Most routers come with a built-in Web server. These Web interfaces enable you to do everything you need on your router and are much easier to use than Cisco's command-line IOS. For a Web interface to work, however, the router must have a built-in IP address from the factory or you have to



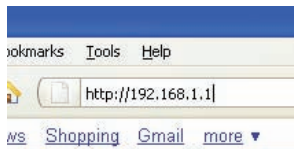
A new Cisco router won't have a password, but all good Cisco people know to add one. Once a router has a password you'll get a prompt to enter that password after you type in **enable**.





• **Figure 8.43** Default IP address

Many routers are also DHCP servers, making the initial connection much easier. Check the documentation to see if you can just plug in without setting an IP address on your PC.



• **Figure 8.44** Entering the IP address

**Tech Tip**

**Default Names and Passwords**

Every brand of router tends to use the same default user names and password. Just about every Linksys router, for example, uses a blank user name and the password "admin." An admin who fails to change the default password is asking to get hacked!

is set to 192.168.1.1/24 from the factory, set your computer's IP address to 192.168.1.2/24. Then connect to the router (some routers tell you exactly where to connect, so read the documentation first) and check the link lights to verify you're properly connected. Open up your Web browser and type in the IP address, as shown in Figure 8.44.

Assuming you've done everything correctly, you'll almost always need to enter a default user name and password, as shown in Figure 8.45.

The default user name and password come with the router's documentation. If you don't have that information, there are plenty of Web sites that list this data. Do a Web search on "**default user name password**" to find one.

Once you've accessed the Web interface, you're on your own to poke around to find the settings you need. There's no standard interface, even between different versions of the same router make and model. When you encounter a new interface, take some time and inspect every tab and menu to learn about the router's capabilities. You'll almost always find some really cool features!



• **Figure 8.45** User name and password

enable the Web interface after you've given the router an IP address. Bottom line? If you want to use a Web interface, you have to know the router's IP address. If a router has a default IP address, you will find it in the documentation, as shown in Figure 8.43.

Never plug a new router into an existing network! Most router people use a laptop and a crossover cable to connect to the new router. To get to the Web interface, first set a static address for your computer that will place your PC on the same network ID as the router. If, for example, the router

## Network Management Software

The idea of a “Web-server-in-a-router” works well for single routers, but as a network grows into lots of routers, administrators need more advanced tools that describe, visualize, and configure their entire network. These tools, known as **Network Management Software (NMS)**, know how to talk to your routers, switches, and even your computers to give you an overall view of your network. In most cases NMS manifests as a Web site where administrators may inspect the status of the network and make adjustments as needed.

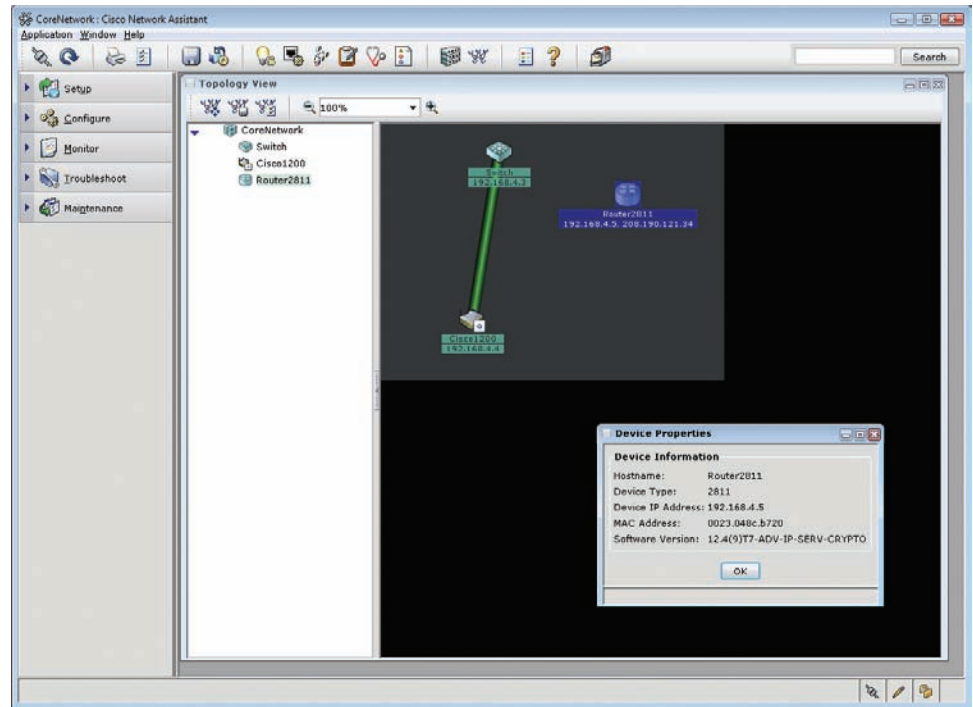
I divide NMS into two camps: proprietary tools made by the folks who make managed devices (OEM), and third-party tools. OEM tools are usually very powerful and easy to use, but only work on that OEM’s devices. Figure 8.46 shows an example of Cisco Network Assistant, one of Cisco’s NMS applications. Others include the Security Device Manager and CiscoWorks, their enterprise-level tool.

There are a number of third-party NMS tools, even some pretty good freeware NMS options. These tools are invariably harder to configure and must constantly be updated to try to work with as many devices as possible. They usually lack the amount of detail you see with OEM NMS and lack interactive graphical user interfaces. For example, CiscoWorks enables you to change the IP address of a port, whereas third-party tools will only let you see the current IP settings for that port. Figure 8.47 shows OpenNMS, a popular open source NMS.

There is no single NMS tool that works perfectly. Network administrators are constantly playing with this or that NMS tool in an attempt to give themselves some kind of overall picture of their networks.

## Other Connection Methods

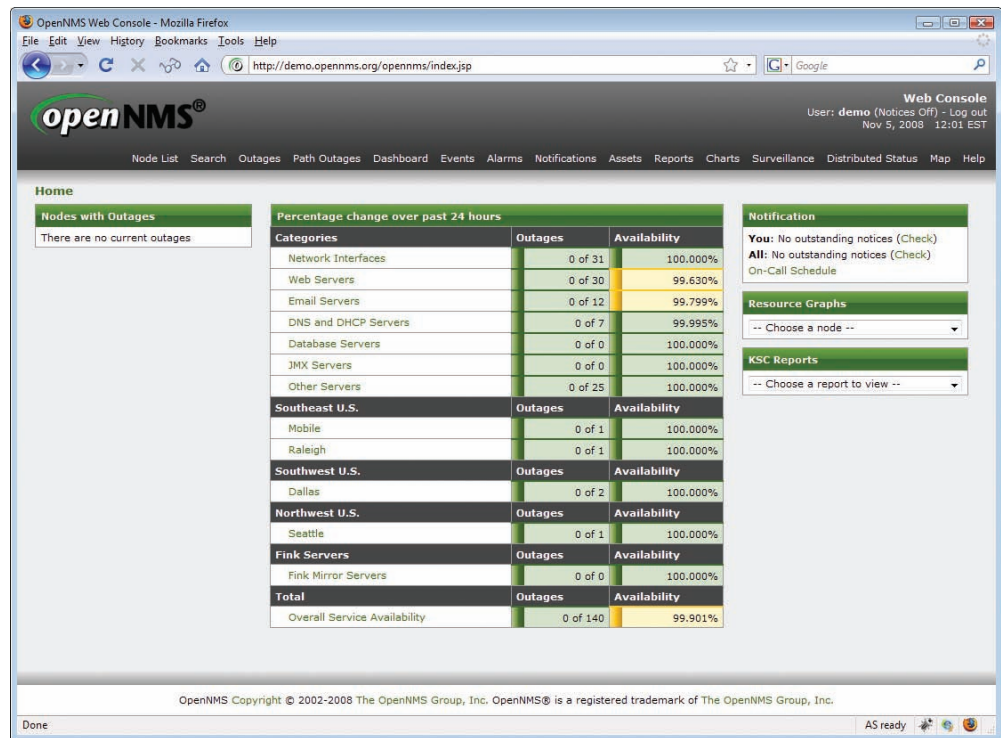
Be aware that most routers have even more ways to connect. Many home routers come with USB ports and configuration software. More powerful routers may enable you to connect using the ancient Telnet protocol or its newer and safer equivalent Secure Shell (SSH). These are terminal emulation protocols that look exactly like the terminal emulators seen earlier in this chapter but use the network instead of a serial cable to connect (see Chapter 9, “TCP/IP Applications,” for details on these protocols).



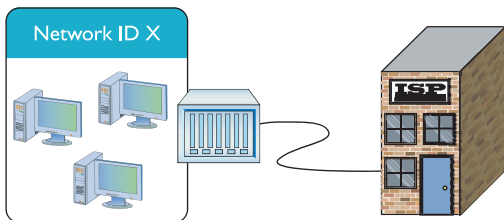
• **Figure 8.46** Cisco Network Assistant



The PuTTY utility works with the old-style terminal emulation and with Telnet and SSH.



• **Figure 8.47** OpenNMS



• **Figure 8.48** The setup

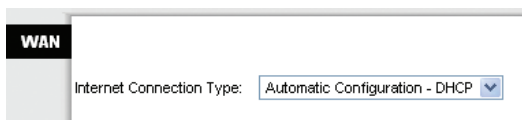
## Basic Router Configuration

A router by definition must have at least two connections. When you set up a router, you must configure every port on the router properly to talk to its connected network IDs and you must make sure the routing table sends packets to where you want them to go. As a demonstration, Figure 8.48 uses an incredibly common setup: a single gateway router used in a home or small office that's connected to an ISP.

There are a number of other settings here that I'm simply ignoring for the moment. In later chapters most of these will be revisited.

### Step 1: Set Up the WAN Side

To start, you need to know the network IDs for each side of your router. The WAN side invariably connects to an ISP, so you need to know what the ISP wants you to do. If you bought a static IP address, it's time to type it in now. However—brace for a crazy fact—most home Internet connections use DHCP! That's right, DHCP isn't just for your PC. You can set up your router's WAN connection to use it too. DHCP is by far the most common connection to use for home routers. Access your router and locate the WAN connection setup. Figure 8.49 shows the setup for my home router—set to DHCP.



• **Figure 8.49** WAN router setup

But what if I called my ISP and bought a single static IP address? This is rarely done anymore, but virtually every ISP will gladly do this for you (although you will pay three to four times as much for the connection). If you use a static IP, your ISP will tell

you what to enter, usually in the form of an e-mail message like the following:

Dear Mr. Meyers,  
Thank you for requesting a static IP address from  
totalsem.com!  
Here's your new static IP information:  
IP address: 1.151.35.55  
Default Gateway: 1.151.32.132  
Subnet Mask: 255.255.128.0  
  
Installation instructions can be found at:  
<http://totalsem.com/setup/>  
  
Support is available at:  
<http://helpdesk.totalsem.com> or by calling (281)922-4166.

In such a case, I would need to change the router setting to Static IP (Figure 8.50). Note how changing the drop-down menu to Static IP enables me to enter the information needed.

Once you've set up the WAN side, it's time to head over to set up the LAN side of the router.

## Step 2: Set Up the LAN

Unlike the WAN side, you usually have total control on the LAN side of the router. You need to choose a network ID, almost always some arbitrarily chosen private range unless you do not want to use NAT. This is why so many home networks have network IDs of 192.168.1/24, 192.168.0/24, and so forth. Once you decide on your LAN-side network ID, you need to assign the correct IP information to the LAN-side NIC. Figure 8.51 shows the configuration for a LAN NIC on my home router.

## Step 3: Establish Routes

Most routers are pretty smart and use the information you provided for the two interfaces to build a routing table automatically. If you need to add more routes, every router provides some method to add routes. The following shows the command line entered on a Cisco router to add a router to one of its NICs. The term "eth0/0" is how Cisco describes Ethernet NICs in its device software (although the full "Ethernet" label often graces their hardware, as you saw way back in Figure 8.1).

```
ip route 192.168.100.0 255.255.255.0 eth0/0 192.168.1.10
```

## Step 4 (Optional): Configure a Dynamic Protocol

The rules to using any dynamic routing protocol are fairly straightforward. First, dynamic routing protocols are tied to individual NICs, not the entire router. Second, when you connect two routers together, make sure that

The screenshot shows the 'WAN' configuration page with a 'Static IP Settings' sidebar. The main area has a dropdown menu for 'Internet Connection Type' set to 'Static IP'. Below this are input fields for 'Internet IP Address' (1.151.35.55), 'Subnet Mask' (255.255.128.0), 'Default Gateway' (1.151.32.132), 'Primary DNS', and 'Secondary DNS'.

• Figure 8.50 Entering a static IP

The screenshot shows the 'LAN' configuration page with an 'IPv4' sidebar. The main area has input fields for 'Local IP Address' (192.168.5.1), 'Subnet Mask' (255.255.255.0), and 'Wireless Local IP Address' (192.168.5.2) with a range of 1..254.

• Figure 8.51 Setting up an IP address for LAN side

those two NICs are configured to use the same dynamic routing protocol. Third, unless you're in charge of two or more routers, you're probably not going to use any dynamic routing protocol.

The amazing part of a dynamic routing protocol is how easy it is to set up. In most cases you just figure out how to turn it on and that's about it. It just starts working.

### Document and Back Up

Once you have your routes configured, take some time to document what you've done. A good router works for years without interaction, so by that time in the future when they do go down, odds are good you've forgotten why you added the routes. Last, take some time to back up the configuration. If a router goes down there's a chance it will forget everything and require you to set it up all over again. But every router has some method to back up the configuration so that you can restore it later.

## Router Problems

The CompTIA Network+ exam will challenge you on some basic router problems. All of these questions should be straightforward for you as long as you do the following:

- Consider other issues first, because routers don't fail very often.
- Keep in mind what your router is supposed to do.
- Know how to use a few basic tools that can help you check the router.

Any router problem starts with someone not connecting to someone else. Even in a small network, there are a number of NICs, computers, switches, and routers between you and to whatever it is you're not connecting. Compared to most of these, a router is a pretty robust device and shouldn't be considered as the problem until you've checked out just about everything else first.

In their most basic forms, routers route traffic. Yet you've seen in this chapter that routers can do more than just plain routing—for example, NAT. As this book progresses you'll find that the typical router often handles a large number of duties beyond just routing. Know what your router is doing and appreciate that you may find yourself checking a router for problems that don't really have anything to do with routing at all.

When it comes to tools, the networking world comes with so many utilities and magic devices that it staggers the imagination. Some, like good old PING and route, you've already seen, but let's add two more tools: TRACEROUTE and MTR.

**TRACEROUTE**, as its name implies, records the route between any two hosts on a network. TRACEROUTE is something like PING in that it sends a single packet to another host, but as it progresses it returns information about every router between them.

Every operating system comes with TRACEROUTE, but the actual command varies among them. In Windows the command is TRACERT and



looks like this (I'm running a TRACEROUTE to the router connected to my router—a short trip):

```
C:\>tracert 96.165.24.1
```

Tracing route to 96.165.24.1 over a maximum of 30 hops:

```
 1      1 ms      1 ms      1 ms  10.12.14.1
 2     10 ms     10 ms      8 ms  96.165.24.1
```

Trace complete.

The UNIX/Linux command is TRACEROUTE and looks like this:

```
michaelm@ubuntu:~$ traceroute 96.165.24.1
traceroute to 96.165.24.1 (96.165.24.1), 30 hops max, 40 byte
packets
 1   10.12.14.1 (10.12.14.1)  0.763 ms 0.432 ms  0.233 ms
 2   96.165.24.1 (96.165.24.1) 12.233 ms 11.255 ms 14.112 ms
michaelm@ubuntu:~$
```

TRACEROUTE is a handy tool not so much for what it tells you when everything's working well, but for what it tells you when things are not working. Take a look at the following:

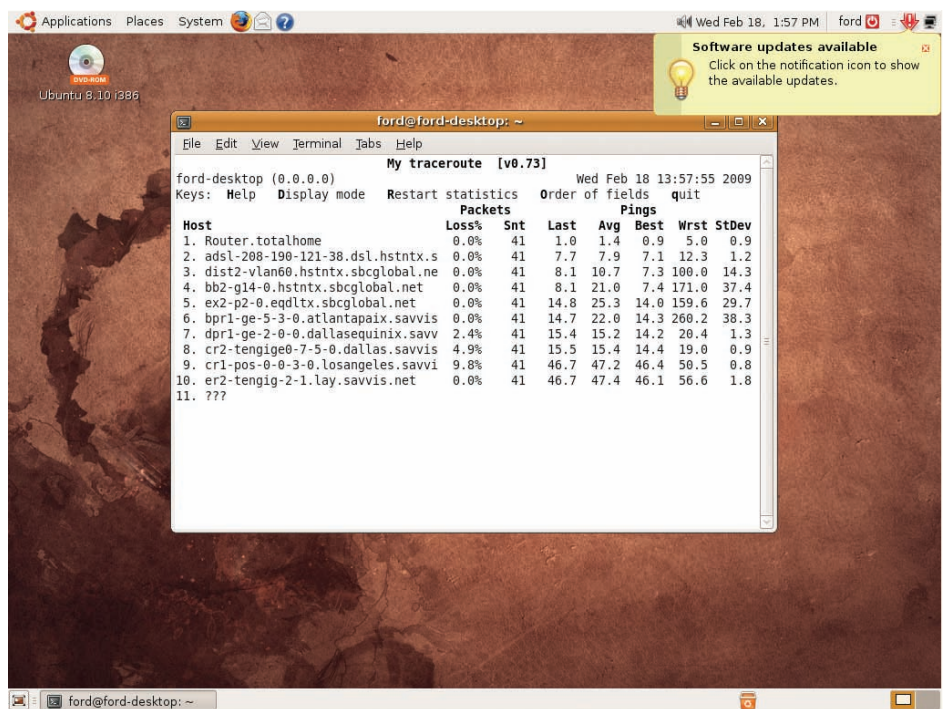
```
C:\>tracert 96.165.24.1
```

Tracing route to 96.165.24.1 over a maximum of 30 hops

```
 1      1 ms      1 ms      1 ms  10.12.14.1
 2      *          *          *      Request timed out
 3   96.165.24.1  reports: Destination host unreachable.
```

If this TRACEROUTE worked in the past but now it no longer works, you know that there's something wrong between your router and the next router upstream. You don't know what's wrong exactly. The connection may be down, the router may not be working; but at least TRACEROUTE gives you an idea where to look for the problem and where not to look.

**My TRACEROUTE (MTR)** is very similar to TRACEROUTE, but it's dynamic, continually updating the route that you've selected (Figure 8.52). MTR is a Linux tool; you won't find it in Windows.



• **Figure 8.52** MTR in action

# Chapter 8 Review

## ■ Chapter Summary

After reading this chapter and completing the exercises, you should understand the following about routing.

### Explain how routers work

- A router is any piece of hardware that forwards network packets based on their destination IP addresses.
- A routing table is the chart of information kept on a router to aid in directing the flow of packets through computer networks.
- Some routers have only two ports—one to connect to the Internet and another to connect to a LAN switch. However, some routers have an integrated switch and thus have more than two ports.
- Routers learn new routes as they go, interacting with each other by exchanging the routing table information. The routing tables are checked and can be updated dynamically as data flows across a network, with routers chatting with each other for the latest network and IP address information periodically.
- Routers can connect dissimilar networks, such as Ethernet, Frame Relay, ATM, and DOCSIS networks.
- The acronym NAT stands for Network Address Translation. NAT is a method that allows an internal network (LAN) to use only one outgoing connection to access the Internet.
- NAT saves a table of information, so it knows which system is communicating with which external site. NAT solutions can be software based, or included as part of a hardware device such as a router.
- There are many varieties of NAT, including Source NAT, Destination NAT, Static NAT, Dynamic NAT, Pooled NAT, SecureNAT, Stateful NAT, and overloaded NAT.

- Static NAT maps a single IP address to a single machine, enabling you to access that machine from outside the network.
- Dynamic NAT can share a pool of routable IP addresses with multiple computers.
- All TCP/IP packets include two port numbers—a destination port and an ephemeral port—which aid in determining what software protocol to use.
- The destination port is a fixed, predetermined number that defines the function or session type while the ephemeral port is an arbitrary number generated by the sending computer.
- Port forwarding hides port numbers from the public side of a network. The router simply forwards packets from one port number to another as the packet passes from the public to the private side of the router.

### Describe dynamic routing technologies

- Routing table entries are entered manually on static routers and do not change. Dynamic routers, in contrast, automatically update their routing table. This is accomplished by using special routing protocols.
- There are three distinct groups of routing protocols: distance vector, link state, and hybrid.
- Distance vector routing protocols calculate the total cost to get to a particular network ID over differing routes. Routing tables are shared with other routers, and the route with the lowest cost is automatically chosen.
- Distance vector routing protocols are not recommended for networks with more than 10 routers because of the time it takes for the routers to reach convergence.
- Distance vector routing protocols include RIPv1, RIPv2, and BGP.

- RIPv1 has a maximum hop count of 15, with routing table updates sent every 30 seconds. Because RIPv1 was limited to routing classful subnets only, lacked authorization, and experienced network overloads as every router sent its routing table at the same time, the RIPv2 update was developed.
- RIPv2 supports CIDR subnets in addition to classful subnets, sends routing tables at random intervals to lessen network overhead, and provides authentication to prevent hackers from sending false routing table information. RIPv2's lengthy time to convergence for large networks led to the development of Border Gateway Protocol (BGP).
- An Autonomous System (AS) consists of one or more networks that are governed by a single protocol. Autonomous Systems do not use IP addresses, but instead use a special globally unique Autonomous System Number assigned by the IANA.
- The protocol used by Autonomous Systems to communicate with each other is generically called an Exterior Gateway Protocol. Networks within an Autonomous System use an Interior Gateway Protocol. Edge routers connect an AS network to another AS network.
- Interior Gateway Protocols include RIP or other protocols. At this time, the Border Gateway Protocol is the only Exterior Gateway Protocol used on the Internet. It connects all of the Autonomous Systems.
- Link state protocols include OSPF and IS-IS. Link state protocols overcome the slow and bandwidth-heavy usage of RIP.
- OSPF stands for the Open Shortest Path First routing protocol. It is the most commonly used Interior Gateway Protocol in the Internet. It is more efficient than RIP, converges dramatically faster than RIP, and supports IPv6 as of OSPF Version 3.
- OSPF broadcasts link state advertisements (Hello packets) when an OSPF-enabled router first boots up. Routers are assigned an Area ID to prevent LSAs from flooding routers on other networks. An Area ID looks like an IP address but has nothing to do with IP.
- The most important area is called Area 0, or the backbone, and has an Area ID of 0.0.0.0.
- The designated router (DR) relays information to all other routers in the area while the backup designated router (BDR) takes over if the designated router is unavailable.
- Intermediate System to Intermediate System (IS-IS) is another link state dynamic routing protocol similar to OSPF. It has supported IPv6 from the start, but is far behind OSPF in popularity and usage.
- Enhanced Interior Gateway Routing Protocol (EIGRP) is a hybrid protocol, proprietary to Cisco, that has aspects of both distance vector and link state protocols.

### **Install and configure a router successfully**

- A Yost cable (rollover cable) is a special serial cable used to connect directly to a Cisco router for configuration purposes.
- Once a direct connection has been made to a router, use a terminal emulation program such as PuTTY or HyperTerminal to communicate.
- Most Cisco products run Cisco's proprietary operating system, Cisco IOS. While not covered on the CompTIA Network+ certification exam, understanding IOS is a must for anyone who wants to become Cisco Certified.
- Most routers include a built-in Web server for configuration. You must know the router's IP address to make this type of connection.
- Many techs use a laptop and a crossover cable to connect to a Web server-enabled router for the initial configuration. This also requires setting a static IP address on the connected laptop, unless the router includes a DHCP server.
- Network Management Software (NMS) is used to describe, visualize, and configure an entire network. NMS is made both by the companies that make managed devices and by third-party companies.
- In general, NMS made by the companies that make managed devices is easy to use, but only works on specific hardware. Much third-party NMS is available as freeware, but is typically harder to use

and must be constantly updated to work with as many devices as possible.

- Some routers may be connected to via USB, Telnet, or SSH.
- When you set up a router, you must configure every port on the router properly to talk to its connected network IDs and to make sure the routing table sends packets to where you want them to go.
- Setting up a router can be broken down into five steps: set up the WAN side, set up the LAN, establish routes, optionally configure a dynamic routing protocol, and finally document and back up your settings.
- The TRACEROUTE utility records the route between any two hosts on a network and can be used to troubleshoot routing problems.

## ■ Key Terms

---

**Area ID** (192)

**Autonomous System (AS)** (189)

**backup designated router (BDR)** (192)

**Border Gateway Protocol (BGP-4)** (190)

**Cisco IOS** (196)

**convergence** (188)

**designated router (DR)** (192)

**Destination NAT** (179)

**destination port** (180)

**distance vector** (186)

**Dynamic NAT** (180)

**dynamic routing** (185)

**edge router** (190)

**Enhanced Interior Gateway Routing Protocol (EIGRP)** (194)

**ephemeral port** (180)

**gateway router** (179)

**hop** (185)

**Intermediate System to Intermediate System (IS-IS)** (194)

**link state** (190)

**managed device** (196)

**metric** (175)

**My TRACEROUTE (MTR)** (203)

**NAT translation table** (182)

**Network Address Translation (NAT)** (178)

**Network Management Software (NMS)** (199)

**next hop** (173)

**Open Shortest Path First (OSPF)** (191)

**overloaded NAT** (181)

**port** (180)

**Port Address Translation (PAT)** (183)

**port forwarding** (183)

**RIPv1** (188)

**RIPv2** (189)

**router** (171)

**Routing Information Protocol (RIP)** (188)

**routing table** (172)

**Source NAT** (179)

**Static NAT (SNAT)** (180)

**static route** (185)

**TRACEROUTE** (202)

**Yost cable** (195)

## ■ Key Term Quiz

---

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

1. A device called a(n) \_\_\_\_\_ is also called a Layer 3 switch.
2. The external routing protocol used on the Internet is \_\_\_\_\_.
3. The variety of \_\_\_\_\_ methods would include RIP, OSPF, BGP, and IGRP.
4. A(n) \_\_\_\_\_ is normally entered manually into a router.
5. A(n) \_\_\_\_\_ connects one Autonomous System to another Autonomous System.

6. \_\_\_\_\_ is a routing protocol that updates routing tables about every 30 seconds, resulting in overloaded network traffic.
7. When all routers can communicate with each other efficiently, they are said to have reached \_\_\_\_\_.
8. Multiple networks that do not use IP addresses and are governed by a single protocol are known as \_\_\_\_\_.
9. You can use the \_\_\_\_\_ utility to troubleshoot routing problems.
10. A(n) \_\_\_\_\_ is an arbitrary number generated by the sending computer that enables the receiving computer to know what application to use for the returning packet.

## ■ Multiple-Choice Quiz

---

1. How many IP addresses should a router have?
  - A. One
  - B. One or more
  - C. Two
  - D. Two or more
2. Choose the Cisco Systems proprietary routing protocols from the following items. (Select two.)
  - A. BGP-4
  - B. EIGRP
  - C. IGRP
  - D. OSPF
3. If specialty accounting software being used at your company requires that packet headers remain unchanged, which item cannot be used on your network?
  - A. RIP
  - B. NAT
  - C. OSPF
  - D. TRACEROUTE
4. How does a router use a routing table to determine over which path to send a packet?
  - A. The first line in the routing table is used if the path is available; otherwise the router tries the next line down, and so on.
  - B. The last line in the routing table is used if the path is available; otherwise the router tries the next line up, and so on.
  - C. After examining all rows in the routing table, the router sends the packet along the path with the highest metric.
  - D. After examining all rows in the routing table, the router sends the packet along the path with the lowest metric.
5. Which version of NAT maps a single routable IP address to a single network node?
  - A. Static NAT
  - B. Dynamic NAT
  - C. Pooled NAT
  - D. SecureNAT
6. Which is not a valid port number?
  - A. 0
  - B. 255
  - C. 1024
  - D. 65,536
7. How is the distance between routers measured?
  - A. In meters
  - B. In hops
  - C. In routes
  - D. In segments
8. Distance vector routing protocols include which of the following? (Select two.)
  - A. RIP
  - B. OSPF



- C. BGP
  - D. ASN
9. Which of the following are benefits of RIPv2 over RIPv1? (Select two.)
    - A. Longer convergence times
    - B. Support for authentication
    - C. Support for CIDR subnets
    - D. Support for metrics
  10. What is one way Autonomous Systems differ from typical Ethernet networks?
    - A. They require a minimum of 10 nodes.
    - B. They cannot exceed a maximum of 255 nodes.
    - C. They are not able to interact with the Internet.
    - D. They do not use IP addresses.
  11. Why are link state protocols more efficient than RIP?
    - A. Entire routing tables are updated on a stricter schedule.
    - B. They forward only changes to individual routes instead of forwarding entire routing tables.
    - C. Packets can be sent along multiple routes at the same time.
    - D. Link state can send larger packets.
  12. What happens when you first connect and turn on an OSPF router?
    - A. It floods the network with Hello packets as it looks for other OSPF routers.
    - B. It floods the network by requesting routing tables from every computer on the network.
    - C. It is unavailable for several hours as it builds its default routing table.
    - D. It runs a self-test to determine if it should run in hybrid mode (RIP and OSPF) or native mode (OSPF only).
  13. Which of the following is a valid Area ID for an Area 0 backbone?
    - A. 0
    - B. 0.0.0.0
    - C. 1.0
    - D. 255
  14. How can you connect directly to a router for configuration purposes? (Select three.)
    - A. Parallel cable
    - B. USB cable
    - C. Crossover cable
    - D. Rollover cable
  15. Once you have made a physical direct connection to a router, what utility/program can you use to issue commands and instructions to it? (Select three.)
    - A. PuTTY
    - B. HyperTerminal
    - C. IOS
    - D. Internet Explorer

## ■ Essay Quiz

1. You have been introduced to a lot more “alphabet soup” in this chapter. Quickly jot down what each of the following stands for: BGP-4, NAT, RIP, OSPF, NMS, PAT, EIGRP, IS-IS, AS, ASN, EGP, IGP, DR, and BDR.
2. Explain why a router is sometimes called a Layer 3 switch.
3. Write a short essay about OSPF and its uses, as well as its benefits over using RIPv2.
4. Briefly explain the difference between a destination port and an ephemeral port.

## Lab Projects

- **Lab Project 8.1**

A classmate of yours is all excited about some upcoming classes available at your school that will cover Cisco routing. He keeps talking about EIGRP and its importance in the workplace, as well as how much cash can be earned if you know EIGRP. Use the Internet to research EIGRP—its history, its uses, what devices run using EIGRP, and what salaries

can be earned by Cisco Certified professionals—possibly your next certification after passing the CompTIA Network+ exam. Then share this information with your instructor and your classmate to compare your findings. What does EIGRP do for corporate networks? What salaries are realistically possible? What were your sources?

- **Lab Project 8.2**

Start a command prompt at your computer and enter **netstat -nr** to view its routing table. Create a screenshot of the output and paste it into a word processing document. Under the pasted screenshot,

briefly explain what each column is for. Compare your routing table to your classmates' routing tables and explain to each other what the differences are and why there are differences.