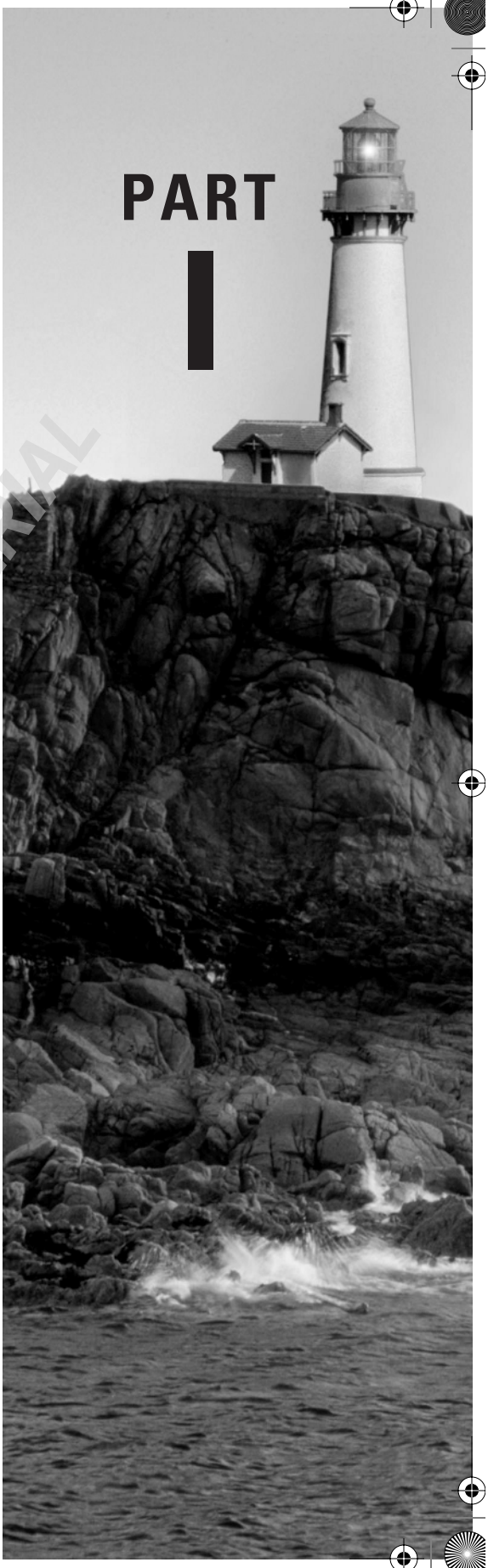
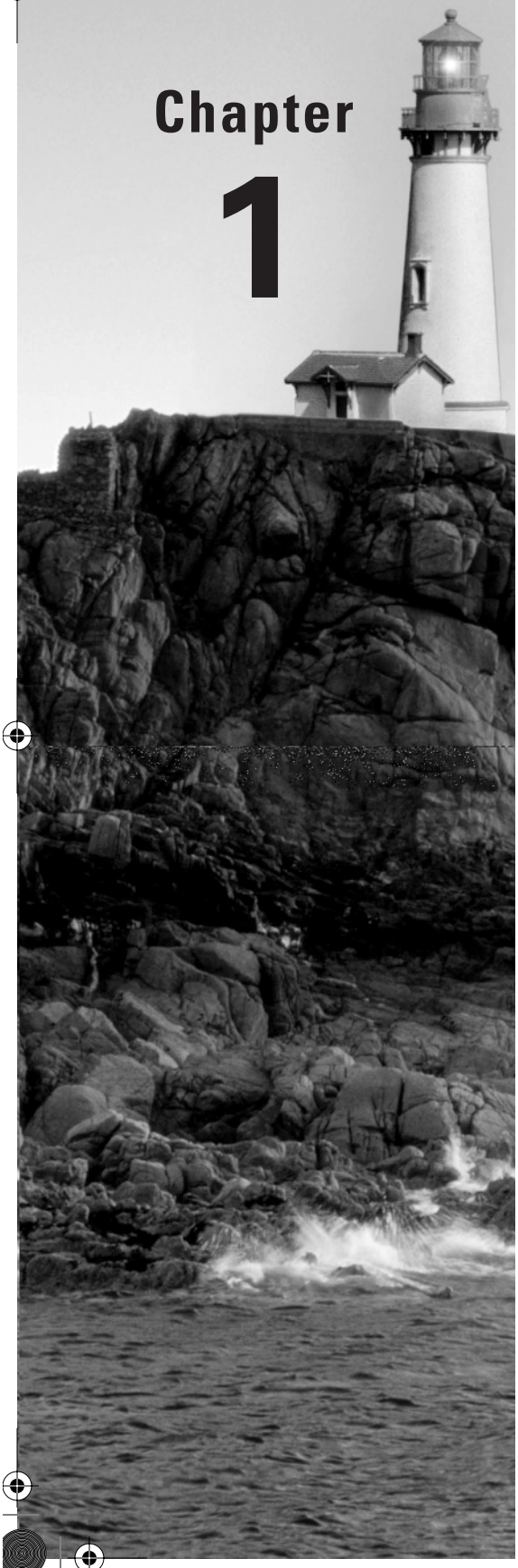
PART I

Building Scalable Cisco Internetworks (BSCI)



Chapter 1



Routing Principles

THE CCNP EXAM TOPICS COVERED IN THIS CHAPTER INCLUDE THE FOLLOWING:

- ✓ **Understand how routers route data.**
- ✓ **Know the difference between classful and classless routing.**
- ✓ **Know how link-state routing protocols operate.**
- ✓ **Know the difference between distance-vector and link-state routing protocols.**

In this chapter, you will learn the fundamentals of what is required to move a packet, or route a packet, across an internetwork. This chapter gives you an overview of the fundamentals of routing and the factors that affect routing. It also takes a look at how distance-vector routing protocols stack up to link-state routing protocols.

This is an important chapter that will provide you with a solid understanding of the covered topics before attempting the more advanced topics covered later in this book. As in sports, if you don't know the fundamentals of how to play the game, you will never be able to attain the level of excellence you could have if you had learned the fundamentals. With that in mind, let's get started!

Components of Routing Data

You may be thinking at this point, "What is routing and how does it work?" The "What is routing?" part is easy to answer. Routing is the process of forwarding a packet from one place on an internetwork to another. As for the second portion of the question, "How does it work?" that will take a little more explanation.

The first thing you will need to understand is logical addressing. Logical addressing is used to provide identification for each host on a network as well as for the network itself. Logical addressing is very similar to the way addressing works for your own home. The state, city, and zip code portion of an address is similar to the network portion of a logical address. It tells the postal service, or in this case, the router, in what general area to find your home, or the network. Your street address, or in this case the host address, tells the postal service, or router, exactly where you are. Upon receiving a packet from a host, the router will need to make a routing decision. After the decision has been made, the router will switch the packet to the appropriate interface on the router to forward it out. You heard me right; the router actually switches packets as well as routes them.

Let's take a look at the three obstacles a router must clear in order to make an accurate routing decision:

- Does the router that is sending and receiving the traffic know the protocol being used? The protocols that are most widely used are IP and IPX. Other protocols, such as AppleTalk and DECnet, may also be used.
- The router then checks to see if the destination network address exists in its routing table. The router will look for a route that matches the destination network address with the longest matching network mask. If the router does not find a route to the destination network, the router will discard the packet and send an ICMP destination network unreachable message to the source of the packet.

- A matching route must have been found or the packet will not reach this third step. From the routing table, the router determines which interface to use to forward the packet. If the routing table entry points to an IP address, the router will perform a recursive lookup on that next-hop address until the router finds an interface to use. The router switches the packet to the outbound interface's buffer. The router then determines the layer 2 address—MAC, DLCI, and so on—that maps to the layer 3 address. The packet is then encapsulated in a layer 2 frame appropriate for the type of encapsulation used by the outbound interface. The outbound interface then places the packet on the medium and forwards it to the next hop.

The packet continues this process until it reaches its destination.

Routing Tables

At this point you may be wondering, "What is a routing table?" The first thing you need to understand is what a route is. The easiest way to explain a route is to think of using an online map. You are required to enter your current location, or source location, and your destination. After you enter this information, the online map will do its nice little calculation and print the best route to take you from your source location to the destination. A route in the world of internetworking is essentially the same, with each router keeping track of the next hop in the route between itself and the next downstream router toward the destination. Once a router has learned a route, it places it in a repository for future use, assuming it has not already learned a route that it considers to be better. This repository is known as a *routing table*.

In order to view the IP routing table on your router, you need to use the command `show ip route`. Let's take a look at an actual routing table:

```
2501A#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B – BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area N1 - OSPF NSSA
external type 1, N2 - OSPF NSSA external type 2 E1 - OSPF external type 1, E2 -
OSPF external type 2, E – EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2,
* - candidate default U - per-user static route, o - ODR, P - periodic
downloaded static route T - traffic engineered route

Gateway of last resort is not set

     172.16.0.0/16 is subnetted, 1 subnets
C       172.16.50.0 is directly connected, FastEthernet0/0
C       192.168.24.0 is directly connected, FastEthernet0/0
     10.0.0.0/8 is subnetted, 1 subnets
C       10.10.10.0 is directly connected, Serial0/0
R       175.21.0.0/16 [120/1] via 10.10.10.1, 00:00:18, Serial0/0
2501A#
```

Now you may be wondering what all of this means. So, let's break it down.

The `Codes` section at the very top tells you how the route was learned. As you may have noticed, there are many different ways a route can be learned. It's not important for you to memorize all of the possible codes. What is important is for you to know how to use the `Codes` section to find out how a route was learned.

Next, note the line `Gateway of last resort is not set`. The gateway of last resort, also known as a default route, is where your router will send IP packets if there isn't a match in the routing table. In this example, the gateway of last resort has not been set. This means if the router receives a packet destined for an unknown network, it will drop the packet and send an ICMP destination network unreachable message to the originator of the packet.

The next items in the routing table are the routes the router knows about. Let's go ahead and break down a route into its components. We will use the following example:

```
R       175.21.0.0/16 [120/1] via 10.10.10.1, 00:00:18, Serial0
```

R The means by which the route entry was learned on this router. In this case, the `R` stands for RIP. From this, you can deduce that the entry you are looking at was learned by the RIP routing protocol.

175.21.0.0/16 The network address and prefix length (number of bits set to 1 in the subnet mask) of the destination network.

[120 The administrative distance of the route. (We will explain administrative distance a little later in this chapter.)

/1] The metric of the route specific to the routing protocol used to determine the route. RIP uses hops as its metric. A *hop* is how many routers away—excluding this router—the destination network is. In this example, there is one router between this router and the destination.

via 10.10.10.1 The next-hop address for the route. This is the address that the packet will need to be sent to in order for the packet to reach its destination.

00:00:18 The length of time since the route has been updated in the routing table. In this example, the route was updated 18 seconds ago.

Serial0 The interface the route was learned through. This is also the interface the packet will be switched to in order for the packet to be forwarded toward its destination. If you see another IP address here, at least one additional lookup will have to occur within the same routing table, which is defined as a recursive lookup, until a route is finally encountered that does list an exit interface in this position.

Populating the Routing Table

Now that you know what's in a routing table, you may be wondering how those routes get there. Before a route can populate a routing table, the router has to learn about the route. There are two ways a router can learn about a route:

- Static definition by an administrator
- Dynamic learning through routing protocols

Statically Defined Routes

A statically defined route is one in which a route is manually entered into the router. A static route can be entered into the router with the following command in global configuration mode:

```
ip route prefix mask {address|interface} [distance]
```

The parts of this command are as follows:

- *prefix* is the IP route prefix for the destination.
- *mask* is the prefix mask for the destination.
- *address* represents the IP address of the next hop that can be used to reach the destination.
- *interface* is the network interface to use.
- *distance* is an optional parameter that represents the administrative distance.

As you can see, with the static route you can choose to either set the next-hop address or use a connected interface on the router. You can also set the administrative distance of the static route. (We will explain administrative distance a little later in this chapter.) When a static route has the administrative distance set to a value other than the default value, it is generally done to create what is known as a floating static route. Here is an example of a configured static route:

```
2501A(config)#ip route 192.168.20.0 255.255.255.0 172.16.50.1
```

If you want to configure a default route, all you need to do for the destination prefix is set it to 0.0.0.0 and set the mask to 0.0.0.0. ANDing with a mask of all 0s turns any intended destination address into all 0s. Comparing this to the configured destination prefix of all 0s always gets a match. The mask length, however, is the shortest possible, with no 1s set, so any other match will always be chosen. When no other matches exist, this default route will be used, hence its name.

You then need to decide what to set your next hop to. This default route will send any packets that do not have a match in the routing table to the next hop defined.

The advantages to using static routes in an internetwork are that the administrator has total control of what is in the router's routing table and there is no network overhead for a routing protocol. Using static routes for a small network is fine. It's not going to be hard to implement, and you have total control in the network.

The downfall of using only static routes is they do not scale well. What do we mean by that? Let's look at an example of how many routes you would need to enter for the number of routers in these different internetworks, where the routers are daisy-chained with one link between each pair of neighbors and the two end routers have stub Ethernets, resulting in each router being connected to two network segments:

- A network with two routers would require two static routes.
- A network with three routers would require six static routes.
- A network with 100 routers would require 9,900 static routes.

The generic equation is the same one used to determine the number of full-mesh links in WAN networking: $n(n-1)$ or n^2-n, where n represents the total number of routers in the internetwork. As you can see, as an internetwork grows, the number of static routes the administrator needs to control becomes unmanageable. Keep in mind that any static route you add, edit, or delete will need to be propagated across all devices. What is the alternative? The alternative is to use a routing protocol to dynamically learn routes.

Dynamically Learned Routes

What is dynamic routing? *Dynamic routing* is a process in which a routing protocol will find the best path in a network and maintain that route. Think about the online map scenario I used earlier. There are multiple ways to get from where you are to your destination. The online map takes all those routes into consideration and uses a predefined set of rules to discover the best route to the destination.

A routing protocol works the same way. It will discover all the possible routes to one destination, implement its predefined rules, and come up with the best route to the destination. One thing a routing protocol will take into consideration that an online map will not is what happens when a portion of the route to the destination has been closed. The routing protocol will automatically find an alternate route to the destination.

Routing protocols are easier to use than static routes. This comes at a cost, though. We're not talking about a monetary cost either. A routing protocol consumes more CPU cycles and network bandwidth than a static route. For a large network, the cost is worth it.

There are two types of dynamic routing protocols in use today: Interior Gateway Protocols (IGPs) and External Gateway Protocols (EGPs). IGPs are used to exchange routing information within the same *routing domain*. A routing domain is the collection of routers and end systems that operate under a common set of administrative rules. Barring hierarchical design with areas or route filtering, two routers can be said to be in the same routing domain if each router's noncommon, directly connected networks can be expected to appear in the other router's routing table, all learned via the same dynamic routing protocol. Areas and filters make the routing domain boundary a bit more difficult to define without closer investigation.

An *autonomous system (AS)* is a collection of routing domains under the same administrative control. An EGP is used to exchange routing information between different ASs. An example of an EGP is the Border Gateway Protocol (BGP). BGP will be covered in detail in Chapter 8, "Border Gateway Protocol," and Chapter 9, "Advanced Border Gateway Protocol."

IGPs can be broken into two classes: distance-vector and link-state. IGPs can also be broken into two categories: classful routing protocols and classless routing protocols. We will first take a look at the different classes of routing protocols.

An important term to understand is *convergence*. Convergence is the process in which all routers update their routing tables and create a consistent view of the network. It will be covered in detail later in this chapter.

Distance-Vector Routing

Distance-vector routing is broken down into two parts: distance and vector. Distance is the measure of how far it is to reach the destination, or the metric to reach the destination. Vector, or direction, is the direction the packet must travel to reach that destination. This is determined by the next hop of the path.

Distance-vector routing protocols are known to *route by rumor*. What this means is that a router will learn routes from its neighbors. Those neighbors learned the routes from their neighbors. It reminds me of my old high school days when one person would tell another person something and by the end of the day the entire school knew.

So, what routing protocols are distance-vector routing protocols? The only ones we are concerned about in this book are Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), and Enhanced Interior Gateway Routing Protocol (EIGRP). Because IGRP and EIGRP are covered in great detail in Chapter 4, "IGRP and EIGRP," we will not spend much time on them here.

EIGRP is what is known as an advanced distance-vector routing protocol, or hybrid. For the BSCI course, Cisco considers EIGRP in the distance-vector routing protocol class.

Table 1.1 compares the different distance-vector routing protocols covered in this study guide.

TABLE 1.1 Distance-Vector Comparisons

Characteristic	RIPv1	RIPv2	IGRP	EIGRP
Count to infinity	✓	✓	✓	
Split horizon with poison reverse	✓	✓	✓	✓
Holddown timer	✓	✓	✓	
Triggered updates with route poisoning	✓	✓	✓	✓
Load balancing with equal paths	✓	✓	✓	✓
Load balancing with unequal paths			✓	✓
VLSM support		✓		✓

TABLE 1.1 Distance-Vector Comparisons *(continued)*

Characteristic	RIPv1	RIPv2	IGRP	EIGRP
Automatic summarization	✓	✓	✓	✓
Manual summarization		✓		✓
Metric	Hops	Hops	Composite	Composite
Hop count limit	15	15	255 (100 by default)	255 (100 by default)
Support for size of network	Small	Small	Medium	Large
Method of advertisement	Broadcast	Multicast	Broadcast	Multicast

The algorithm Cisco supports for RIP and IGRP is known as Bellman-Ford. For EIGRP, Cisco supports the Diffusing Update Algorithm (DUAL).

EIGRP and IGRP are Cisco proprietary routing protocols.

Most distance-vector routing protocols have common characteristics:

Periodic updates The length of time before a router will send out an update. For RIP, this time is 30 seconds. For IGRP, the time is 90 seconds. This means that once the periodic update timer expires, a broadcast or multicast (in the case of RIPv2) of the entire routing table is sent out. Uncharacteristic of distance-vector routing protocols, EIGRP does not send periodic updates, but ironically, OSPF can be said to do so every 30 minutes, in the form of link-state advertisement (LSA) refreshes.

Neighbors Other routers on the same logical, or data-link, connection. In a distance-vector routing protocol, a router will send its routing table to its connected neighbors. Those neighbors will send their updated routing tables to their connected neighbors. This continues until all the routers participating in the selected routing domain have updated routing tables.

Broadcast or multicast updates When a router becomes active, it will send out a routing advertisement or Hello packet to the broadcast or designated multicast address, stating that it is alive. In return, neighboring routers in the same routing domain will respond to this broadcast or multicast.

Full routing table updates Most distance-vector routing protocols will send their entire routing table to their neighbors. This occurs when the periodic update timer expires.

Routing by rumor A router will send its routing table to all of its directly connected neighbors. In return, all of the neighboring routers send their routing tables to all of their directly connected neighbors. This continues until all routers in the same distance-vector routing domain converge upon the same information.

Triggered updates and route poisoning One way to speed up convergence on a network is with the use of *triggered updates* and *route poisoning*. Instead of the router's having to wait until the periodic update timer expires to send out an update, a triggered update sends out an update as soon as a significant event occurs. An example would be if a router notices that one of its connected networks went down. The router will then send out an update stating that the downed network was unreachable, thus speeding up convergence and cutting down on the risk of network loops due to convergence issues.

Route poisoning is the immediate removal of a route from the local router's routing table, once it is determined that the route is no longer valid and subsequently advertises this fact to neighbors. Because this determination can be almost immediate in RIP, through direct connection to the failed link or through receipt of triggered updates, there is little opportunity in RIP networks these days for routes to enter a holddown state and slowly age out. Even RIP, as an example of a distance-vector routing protocol, converges in less than 30 seconds in modern networks due to route poisoning and triggered updates. IGRP still takes the long way home, as discussed in the IGRP convergence section coming up in this chapter.

Holddown timer The holddown timer is used when information about a route changes for the worse (greater metric or unreachable). When the new information is received or a route is removed, the router will place that route in a holddown state. This means that the router will advertise but will not accept worse advertisements about this route from any neighbor, other than the one from which the route was originally learned, for the time period specified by the holddown timer. After the time period expires, the router will start considering all advertisements about the route.

The benefit of using holddown timers is that, if used properly, they will cut down on the amount of wrong information being advertised about routes. The disadvantage is that convergence times may increase.

Invalid and flush timers These timers solve the problem of what happens when a router goes down. Because the router isn't sending out updates, the other routers in the network don't know that a router has gone down and that the routes are unreachable. So, the routers continue to send packets to the routes connected to the missing router. This means the packets never make it to their destination. The way an invalid timer solves this issue is by associating a period of time with a route. If the route is not updated in the routing table in this set period of time, the route is marked as unreachable, and the router will send this new information in a triggered update and in its periodic updates. Depending on the routing protocol, the default invalid timer is set at three or six times the periodic update timer and is reset for a particular route upon receipt of an update for that route.

The invalid timer should not be confused with the flush timer. Although both the invalid and flush timers are somewhat tied to when an update is received, the flush timer is set for a longer period of time, after which the route is stripped from the local routing table and no longer advertised.

Cisco suggests that you leave all timers at their default settings, but that if you must change them, for IGRP, make sure that the flush timer is equal to or greater than the sum of the invalid and holddown timers, as it is by default. Otherwise, without the route kept in holddown in the routing table, the routing table is unprotected from the routing protocol's acceptance of worse routes that are actually invalid, but may still be circulating on the network and could be accepted sooner than the holddown timer would have permitted. It's likely that the invalid routes would have been purged from the routing domain before the routing protocol resumes, if the holddown timer is permitted to run its course. While this same argument makes sense for RIP, the default timer settings do not follow this rule of thumb. For RIP, when no router actually goes down, but a route does go away and triggered updates are able to perform their duty, these timers are basically academic in nature. For IGRP, however, they work exactly as described. The upcoming discussions on RIP and IGRP convergence will clarify this point.

Split horizon with poison reverse *Split horizon* with *poison reverse* helps prevent what is known as a routing loop. A routing loop occurs when a router learns a route from a neighbor and the router turns around and sends that route back to the neighbor that the router learned it from, causing an infinite loop.

Split horizon Consider an example: Router A learns about route 10.10.10.0 from Router B, which is two hops away from 10.10.10.0. Router B tells Router A that Router A is three hops away from network 10.10.10.0. Router A, after populating its routing table with the route, sends an advertisement back to Router B stating that Router A has a route to 10.10.10.0. In this advertisement, Router A tells Router B that Router B is four hops away from network 10.10.10.0. Of course, Router B already knows of a path to network 10.10.10.0 that puts it only two hops away. So, Router B wisely ignores the less desirable route advertised by Router A.

The problem arises if network 10.10.10.0 goes down. Router B would learn that the route is down. In turn, Router B would mark the network as unreachable and pass the information to Router A at Router B's next update interval. Theoretically, before this can happen, Router A could send an update to Router B, stating, as it has all along, that Router A can reach network 10.10.10.0; remember, Router B has not informed Router A that network 10.10.10.0 is unreachable. So, Router B receives the update from Router A about being able to reach network 10.10.10.0. Router B at this point, no longer aware of a better path to network 10.10.10.0, will update its routing table with the new information. When Router B receives a packet destined for network 10.10.10.0, it will look in its routing table and see that the next hop is Router A. So, it forwards the packet to Router A. When Router A receives this packet, it looks in its routing table and notices that Router B is the next hop; remember that Router B initially sent the route to Router A in an update. This causes an infinite loop.

The only thing that stops this process from running indefinitely is the fact that distance-vector routing protocols define infinity in finite terms. Once one router increments the metric to the established infinite value, the neighbor receiving this infinite advertisement realizes that the route is unreachable and advertises the same back to its confused neighbor. This can take a bit of time,

though. Split horizon prevents this by establishing a rule that a route cannot be advertised out the same interface on which it was learned.

Poison reverse Poison reverse is related to split horizon, as it uses the same "keep track of who advertised it to you" philosophy. With IGRP, poison reverse comes into play when a neighbor router tells the local router about a downed network that the neighbor has been advertising as active, and for which the neighbor represents the only next hop to that network. In future updates sent back to the neighbor, the local router will bend the split horizon rule by advertising the route back to the neighbor from which it was learned, but by using an infinite metric to imply its inaccessibility through the local router. For IGRP, infinity is the value 4,294,967,295, which represents a 32-bit field of all 1s in binary. This is just to make sure there is no misunderstanding about the fact that the local router most certainly cannot help with an alternate route to the network. It is RIP, not IGRP, that employs what is known as local route poisoning by immediately removing the route from the local routing table. So IGRP must deal with the protracted presence of the route until it can be flushed, and poison reverse is its coping mechanism.

The other interesting issue you may notice is that IGRP keeps suspected bad routes in the routing table until after the holddown timer expires, and labels them as such. So it's your position, as the administrator or technician, to cope with the fact that a route appears to be down in the routing table but still passes traffic to the listed next-hop address. If the network comes back up, the entry will not change until after the holddown timer expires, but each router in line to the destination will operate the same way, passing the traffic until it makes it to the final destination, barring any other unforeseen circumstances. So, verification and faith and a little trick to be mentioned soon (look for the `clear` command) will have to tide you over.

Make sure you realize there is a difference between route poisoning and poison reverse. Route poisoning is explained earlier in this section.

Counting to infinity In networks where split horizon, triggered updates, and holddowns are not implemented, the phenomenon outlined in the previous split-horizon discussion, known as counting to infinity, occurs. When the destination network goes down, the updates about the destination being unreachable can arrive between scheduled update times. If an upstream (away from the downed route) neighbor's update timer expires before ours does, the local router will receive an update about the route that leads it to believe that the network is once again reachable. Any combination of split horizon, triggered updates, and holddown timers would mitigate the effects of this situation. Without any of these mechanisms, the bad route will be volleyed back and forth between the neighbors—with an incrementing metric—until the predefined maximum routing domain diameter (16 for RIP and 100, by default, for IGRP) is reached for the number of hops by each router.

Without enforcing maximum hop counts, this situation could literally go on forever. When a route reaches the maximum hop count (infinity), the route is marked as unreachable and removed from the router's routing table. Even IGRP and EIGRP report the number of hops to a destination network in their routing updates and enforce a configured maximum diameter. They just don't use hop count in the calculation of their metrics.

Now that you have an understanding of how distance-vector routing protocols function, let's take a look at them. We will cover only RIP in this chapter because IGRP and EIGRP are covered in detail in Chapter 3, "Network Address Translation."

ROUTING INFORMATION PROTOCOL (RIP)

This section is going to hit only the key areas of RIP, because that's all that is really pertinent to the BSCI exam. There are currently two versions of RIP in existence: RIP version 1 and RIP version 2. Let's take a brief look at the major differences between them.

RIP version 1 (RIPv1) is considered a classful routing protocol, whereas RIP version 2 (RIPv2) is a classless routing protocol. The key difference between a classful and classless routing protocol is that a classful routing protocol does not send a subnet mask in the update and a classless routing protocol does. Classful versus classless routing is covered in more detail later in this chapter. Other attributes RIPv2 has that RIPv1 doesn't are as follows:

- Authentication of routing updates through the use of cleartext or MD5 (optional)
- Multicast route updates
- Next-hop addresses carried with each advertised route entry

In order to activate RIPv2 you must first enter the `router rip` command in global configuration mode. After RIP has been activated on the router you must enter the command `version 2` in router configuration mode.

Real World Scenario

RIP Migration

John is the network engineer for company XYZ. Currently, XYZ has only 14 routers and is running RIPv1. Recently XYZ purchased company ABC. Company ABC had 10 routers that were also running RIP. John has been tasked with merging the two companies' networks. John remembers back when he was studying for the BSCI that RIP has a maximum consecutive device count of 15. Well, he now has 24 routers and will exceed this limit for a number of paths. Noticing the dilemma, he decides to implement EIGRP to replace the RIP network. In order to make sure the company doesn't lose connectivity, John decides he will implement EIGRP and leave RIP on the devices until EIGRP is completely implemented. By choosing to do it this way, John will be able to migrate the two networks together without losing connectivity.

What you need to concentrate on at this point is the commonality among the two versions of RIP, such as updates and timers:

- They are considered distance-vector routing protocols.
- They use the Bellman-Ford algorithm.

- The metric used to determine the best route is hop count. A route can extend through 15 routers—or hops—and then will be marked as unreachable.
- The route update timer for periodic updates is set to 30 seconds.
- The route invalid timer is set to 180 seconds. This is the time the router will wait for an update before a route will be marked as unreachable and the holddown timer will be started, which is also 180 seconds.
- The route flush timer is set to 240 seconds. This is the time between the route's last received update and the route being removed from the routing table. In the time period between the invalid timer and the flush timer, neighboring routers will be notified about the route's being unreachable, unless the holddown timer expires before the flush timer does and updates come in for the route. In that case, business resumes as usual, possibly through a path less desirable than the original, but one that's valid and the best known one.

Now that you have a good understanding of how distance-vector routing works, let's take a look at link-state routing and its functionality.

If you need to view real-time information about the operation of RIPv1 or RIPv2, you can use the `debug ip rip` command.

The `ip default-network` command can be used with RIPv1 or RIPv2 to advertise a default network to your neighboring devices.

Link-State Routing

Remember how with a distance-vector routing protocol, the router knew only the direction in which to send the packet and the distance to get there? *Link-state routing* is different in that each router knows the exact topology of the network. This in turn limits the number of bad routing decisions that can be made. Link-state routing can accomplish this because every router in the routing domain or area has a similar view of the network, placing itself at the root of a hierarchical tree. Each router in the network will report on the state of each directly connected link. Each router then plays a part in propagating this learned information until all routers in the network have it. Each router that receives this information will take a snapshot of it.

It's important to realize that the other routers do not make any change to the updates received. This in turn ensures that all routers in the process have the same relative view of the network, allowing each router to make its own routing decisions based upon the same information.

Another key difference of link-state routing is that each router does not send its entire routing table. The only information that is sent are the changes that have occurred or a message stating that nothing has changed after a given period of time has passed. This is known as a link-state advertisement (LSA). An LSA is generated for each link on a router. Each LSA includes an identifier for the link, the state of the link, and a metric for the link. With the use of LSAs, link-state protocols cut down on the amount of bandwidth utilized. The disadvantage of a link-state routing protocol is that it is more complex to configure than a distance-vector routing protocol.

The link-state routing protocols that are covered in this book are as follows:

- Open Shortest Path First (OSPF)
- Integrated Intermediate System to Intermediate System (Integrated IS-IS)

Keep in mind these are not the only link-state routing protocols. These are the ones that are covered by the BSCI exam, though.

Because we will cover link-state routing in more detail in Chapter 4, Chapter 5, "OSPF Operation in a Single Area," and Chapter 6, "Interconnecting OSPF Areas," we will give you only a brief introduction to the operation of link-state routing here.

The basic functionality of link-state routing is broken down into the following steps:

1. The first thing each router does, as it becomes active, is form an adjacency with its directly connected neighbors.
2. After forming adjacencies, the router then sends out link-state advertisements (LSAs) to each of its neighbors. After receiving and copying the information from the LSA, the router forwards—or floods—the LSA to each of its neighbors.
3. All of the routers then store the LSAs in their own database. This means that all routers have the same view of the network topology.
4. Each router then uses the Dijkstra algorithm to compute its best route to a destination.

As stated previously, this is a brief introduction to link-state routing. Link-state routing will be covered in greater detail later in this book. Table 1.2 compares the link-state routing protocols covered in this study guide. Remember that EIGRP is considered a hybrid protocol, meaning that it contains traits of both distance-vector and link-state routing protocols. Also remember that if you are forced to consider EIGRP to be one or the other only, consider it a distance-vector routing protocol.

TABLE 1.2 Link-State Comparisons

Characteristic	OSPF	IS-IS	EIGRP
Hierarchical topology supported through areas	✓	✓	
Retains knowledge of all possible routes	✓	✓	✓
Manual route summarization	✓	✓	✓
Automatic route summarization			✓
Event-triggered announcements	✓	✓	✓
Load balancing with unequal-cost paths			✓
Load balancing with equal-cost paths	✓	✓	✓
VLSM support	✓	✓	✓

TABLE 1.2 Link-State Comparisons *(continued)*

Characteristic	OSPF	IS-IS	EIGRP
Metric	Cost	Cost	Composite
Hop count limit	Unlimited	1024	100 by default
Support for size of network	Very large	Very large	Large

Now that we've discussed the different classes of routing protocols, let's focus on the two different categories of routing protocols.

Classful Routing

What is classful routing? Classful routing is used to route packets based upon the default major network boundary, derived from the class of the IP address. In order to fully understand this concept, let's review the defaults for the different classes of IP addresses.

Class A networks reserve the first octet for the network portion of the IP address, and the remaining three octets are available for host addressing. The value of the first octet of a Class A network will always be between 1 and 127, inclusive. There are a total of 126 Class A networks; 127 is reserved for diagnostic testing and thus cannot be used. There are various other reserved Class A networks, such as the 10 network, but the majority are usable and already allocated. There are 16,777,214 unsubnetted hosts available per Class A network.

Class B networks reserve the first and second octets for the network portion of the IP address, and the remaining two octets are available for host addressing. The value of the first octet of a Class B network will always be between 128 and 191, inclusive. There are a total of 16,384 Class B networks with 65,534 hosts per network.

Class C networks reserve the first, second, and third octets for the network portion of the IP address and the remaining octet for host addressing. The value of the first octet of a Class C network will always be between 192 and 223, inclusive. There are a total of 2,097,152 available Class C networks with 254 hosts per network.

Class D IP addresses have no network/host structure, as they are used solely for multicasting and, like broadcast addresses, multicast addresses can be only destination addresses, never source addresses. As a result, there is no need or way to split Class D addresses up into smaller subnets, because no device will ever be configured with a Class D address as its interface address. Furthermore, there is no subnet mask associated with Class D addresses. The value of the first octet of a Class D address will always be between 224 and 239, inclusive. There are theoretically 268,435,456 Class D addresses, which are not split into networks and hosts.

Class E networks are regarded as experimental, and, like Class D addresses, they have no network/host structure nor will they ever be assigned to a device's interface. The value of the first octet of a Class E address is always between 240 and 255, inclusive.

It's not necessary to convert an IP address in binary form to decimal in order to determine its class. When faced with such a task, the quickest way to determine the class of an IP address in binary form is to label the first four bits A, B, C, and D, after the classes of addresses. Wherever

the first 0 falls, that is the class of address you are dealing with. If all four bits are 1s, the class of address is E. For example, a first octet of 10101010 would represent a Class B address, due to the first 0 being in the B position. 10101010 converts to decimal 170, so you can see that the trick worked in this case. Trust me, it always does.

Classful routing therefore bases all of its routing decisions upon the default major network boundary derived from each of the first three classes of IP address. The major drawback to the use of classful addressing and routing is that a tremendous number of IP addresses can be wasted. We will explain this in more detail in Chapter 3.

The routing protocols covered in this book that are considered classful routing protocols are as follows:

- RIPv1
- IGRP

With all of this in mind, let's take a look at what is known as classless routing.

Classless Routing

Classless routing, also known as classless interdomain routing (CIDR), is not dependent on the default boundaries of the different classes of IP addresses. Classless routing actually allows each route's subnet mask to be sent in the routing update with the route. Classless routing also opens the door for variable-length subnet masking (VLSM), which extends IP addressing beyond the limitations of using fixed-length subnet masks (FLSMs) by allowing you to specify the most efficient subnet mask to use for the size of network in question. This allows you to conserve IP addresses, extending the use of IP address space. This topic is covered in more detail in Chapter 3.

To simplify the difference between classless and classful routing, let's use an example of a college campus. The college campus is built with buildings of identical size. It doesn't matter how many offices in each building will be occupied; all buildings are the same size and every office has to have a number. This is analogous to a classful network design, where every host has a host ID and participates in the same size network, regardless of how many hosts will ever really be on that network. The addresses that are not used on a network cannot be used on a different network that is running short of addresses. All networks will have to grow the same amount to be able to cover the largest need. All buildings have to remain identical in size, even if those that are not full must grow to keep up with the growth of the fullest building. All of the wasted office space is just that. Each group in a building is confined to that building and cannot grow into the less-populated buildings.

Introducing classless routing would be like allowing each building to be only as large as the group within the building, wasting no office space and having only enough empty offices to cover projected growth. In other words, classless routing leads to the ability to use not only subnet masks that are not the default masks, but also to use VLSM—subnet masks of different sizes—so that address waste is minimized.

The routing protocols we cover in this book that are considered classless routing protocols are the following:

- RIPv2
- EIGRP
- OSPF

- IS-IS
- BGP

So far, we have described how routes are learned, the different classes of routing protocols, and the different categories of routing protocols. Now that you have a firm grasp on these concepts, it's time to move on to how these routes you've learned actually get placed in a routing table.

The Final Decision on What Routes Populate the Routing Table

There are a couple of different factors that make the final decision on what routes will be placed in the routing table, namely, administrative distance and metric, assuming that no route filtering is in place that would specifically prevent a route from being placed in the routing table. The first factor to be taken into consideration when making the decision of what route to place in the routing table is administrative distance.

What is administrative distance? *Administrative distance (AD)* is the trustworthiness of the routing protocol that produced the route. Administrative distance can be any value between 0 and 255. The lower the number, the more trustworthy the source of the route.

Table 1.3 shows the default administrative distance that a Cisco router will use to decide the best route to a destination.

TABLE 1.3 Default Administrative Distance

Source of Route	Default Administrative Distance
Connected Interface	0
Static Route	1
EIGRP Summary	5
External BGP	20
EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
External EIGRP	170

TABLE 1.3 Default Administrative Distance *(continued)*

Source of Route	Default Administrative Distance
Internal BGP	200
Unknown	255

If more than one route exists to a given destination, the route with lowest administrative distance will be placed in the routing table. You may be wondering what happens if multiple routes to a given destination have the same administrative distance. This is when the second factor—metric—comes into play.

If you establish a static route by supplying the exit interface instead of the next-hop address, it will have a metric of 0, just like a directly connected network would, making it preferable to next hop-based static routes. This is useful with the `ip unnumbered` command or whenever you want a static route based not on the availability of the remote next-hop address but instead on the availability of the local interface.

A *metric* is the value of a route specific to a routing protocol. If multiple routes have the same administrative distance, then the metric is used as the tiebreaker. Here's a simple way to think about it: The router first looks to see which route can be trusted the most. If the router has multiple routes that are equally trustworthy, the router will then look to see which route has the lowest metric, which is the one it finds to be the most desirable. That is the route that will populate the routing table. Depending on the routing protocol and its configuration, multiple routes with the same AD and metric could be placed into the routing table simultaneously.

Let's summarize everything you've learned so far about routing tables and how they are populated. At this point, you know that routes are learned either dynamically or statically. Those routes are then placed in the routing table based on which one is the most trusted. If multiple routes exist that are equally trusted, the one that is the most desirable is placed in the routing table.

Let's revisit the life of a packet. When a packet is sent to a destination, if the destination is not on the same network as the source, the packet will be sent to a local router for the immediate network. The router then looks in its routing table to see if it has a route to the destination network. If the router does not have a route and a default gateway doesn't exist, the packet is discarded and an ICMP error message is sent to the packet's source. In fact, any router along the path to the destination network could run into this same problem and discard the packet, notifying the original source device of the execution. So, if a route exists, how does the packet reach the destination? We're going to explore getting a packet to its destination in the next section.

Reaching the Destination

After a router receives a packet, the router removes the data-link framing, or the layer 2 header and trailer, if one exists, in order to find the layer 3 destination address. Once the destination

address is read, the router looks in its routing table for a route to the destination address. Assuming a match for the destination is in the routing table, the router reads the next-hop address or exit interface to reach the destination from the entry. If the router reads a next-hop address, it will perform a recursive lookup on the address. This means that the router looks at the network portion of the next-hop address and then looks in its own routing table for an entry matching this destination address.

The router continues this process until it arrives upon an entry that designates a connected exit interface instead of a next-hop address. Once this is accomplished, the router switches the packet to the outbound interface's buffer. The router discovers the type of connection between the outbound interface and the next-hop address. After the connection type has been discovered, the packet is encapsulated in the appropriate layer 2 encapsulation for the connection. The packet will now be placed on the medium and forwarded to the next hop. This continues until the packet reaches its destination.

The entire process of how a packet gets forwarded toward the destination can be broken down into five steps:

1. As the frame's header arrives at the router's inbound interface, the MAC process checks the hardware destination address against the burned-in MAC address of the interface, the broadcast address, and any multicast addresses that the interface may be listening for. If the MAC process finds that the hardware destination address is applicable, a cyclic redundancy check (CRC) is performed on the frame to make sure it's not corrupt. If the frame passes CRC, the packet is pulled from the frame. The frame is discarded, and the packet is stored in the router's main memory.
2. The router searches the routing table for the longest match to the destination address found in the packet's header. If the router finds no match, and a default gateway does not exist, the router will discard the packet and send an ICMP destination unreachable message to the originating device. If the router does find a match, it will discover the next-hop address or the connected interface for this route. If the route points to a connected interface, a recursive lookup doesn't need to be performed and the next step can be skipped.
3. Once the next-hop address is known, the router performs a recursive lookup. This is performed to locate the directly connected interface on the router to forward the packet out; it may take multiple iterations before an entry with an exit interface is found. If any of the recursive lookups points to an IP address that the routing table has no entry for and the default gateway is not set, the router will discard the packet and notify the packet's source via ICMP.
4. The packet is now switched to the outbound interface's buffer. Assuming that the outbound interface uses layer 2 addressing, the router attempts to learn the MAC address or layer 2 identifier of the next-hop interface in order to map the layer 3 address to a layer 2 address. The router looks in the appropriate local table such as an ARP cache. In the case of ARP, if the layer 2 mapping is not found, the router will broadcast an ARP request through the outbound interface to the locally connected segment to request the MAC address of the interface associated with the local segment of the next-hop device, which may be another router or the final destination. Under normal circumstances, the next-hop device sends an ARP reply with its MAC address. All other devices hearing the broadcast

will realize that the ARP request is not for them based on layer 3 address information in the ARP header; they will not reply to the request, but instead quietly discard the ARP request packet. No layer 2 information is necessary for many point-to-point media. If a frame is placed on the wire, only the intended recipient will receive it, because it is the only other device on the wire.

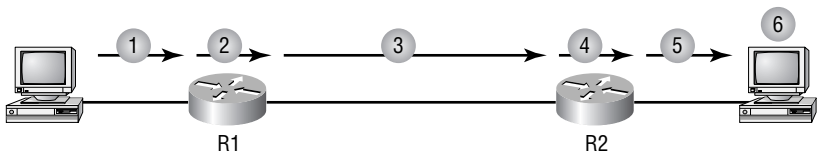
5. At this point, the type of connection between the directly connected interface and the next-hop interface is known. The router encapsulates the packet in the appropriate data-link frame for the type of connection. The outbound interface places the frame with the layer 2 address of the next-hop device on the wire. This process continues at each router that the packet encounters until it reaches its destination.

Figure 1.1 gives you a visual example of the life of a packet:

1. The packet is encapsulated with the layer 2 frame structure of the local network and sent from the originator to its default gateway.
2. The frame is received by R1; the data-link frame is removed; the destination address is found in the routing table; the next hop is discovered; the outbound interface is discovered; and the packet is switched to the outbound interface buffer.
3. The outbound interface receives the packet and resolves the layer 3 address to the layer 2 address of the next-hop router, and the packet is framed in the data-link framing of the outbound interface. The frame, with the layer 2 address of the next-hop router, is then placed on the medium.
4. The frame is received and dismantled by R2. The destination Network layer address is found in the routing table, which points to a directly connected interface. The packet is switched to the outbound interface buffer.
5. The outbound interface maps the layer 3 address to a layer 2 address that is needed to reach the destination. The packet is framed according to the interface's layer 2 technology and is then placed on the medium.
6. The packet arrives at its destination.

So far, you have followed the life of a packet. You should have a grasp on what a routing table is, how that table is populated, and how a packet reaches its destination. You now need to focus on verifying what routes are in a routing table and what tools are used to test and troubleshoot actual connectivity to the destinations they represent.

FIGURE 1.1 The life of a packet



Convergence

Convergence time is the time it takes for all routers to agree on the network topology after a change in the network. The routers have synchronized their routing tables.

There are at least two different detection methods used by all routing protocols. The first method is used by the Physical and Data Link layer protocols. When the network interface on the router does not receive three consecutive keepalives, the link will be considered down.

The second detection method is that when the routing protocol at the Network and Transport layers fails to receive three consecutive Hello messages, the link will be considered down.

After the link is considered down is when the routing protocols differ. Routing protocols have timers that are used to stop network loops from occurring on a network when a link failure has been detected. Holddown timers are used to give the network stability while new route calculations are being performed. They also allow all the routers a chance to learn about the failed route to avoid routing loops and counting to infinity problems. Because a routing domain cannot converge during this holddown period, this can cause a delay in the routing process of the domain. Because of this slow convergence penalty, link-state routing protocols do not use holddown timers.

The following section describes the convergence process for RIP, IGRP, EIGRP, and link-state protocols when a link failure occurs in a network.

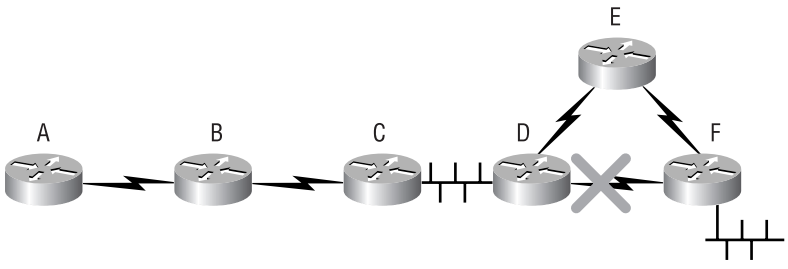
RIP Convergence

Convergence time is one of the problems associated with distance-vector routing protocols. This section details the convergence process of the RIP protocol. We'll use Figure 1.2 to help describe the RIP convergence process.

The following list describes the RIP convergence events when a problem occurs. In Figure 1.2, the WAN between Routers D and F goes down. This link was along the path from Routers A through D, when delivering packets to the Ethernet segment off of Router F. Now, these four routers, in particular Router D, must learn the path through Router E, but each of the four routers will notice an additional hop to this network. Here's what happens:

1. Router D poisons this directly connected route in its own routing table, removes it, and sends a triggered update to Routers E and C. Any routes with Router D's interface in that downed link or Router F's address on that link as a next hop will also be poisoned. This will almost definitely include the Ethernet segment off of Router F.

FIGURE 1.2 Convergence



2. Router C was using this path to access Router F's Ethernet, but Router E goes directly through Router F. So the only effect on Router E is the poisoning and removal of the WAN link from its routing table. Router C, however, poisons both routes, removing both of them from its routing table. Router C sends a triggered update to Router B, which in turn sends a triggered update to Router A, each router poisoning and removing the route locally. Router F also sends a triggered update to Router E.
3. The triggered updates that Router E received from Router D and Router F prompt it to send out its own triggered updates in each direction. These updates tell both of its neighbors that all of the routes they used to have access to are still accessible through Router E. Router D enters the new route for Router F's Ethernet—through Router E—into its routing table.
4. Router D accepts this new route to the destination network, as it did not have it in holddown because it was just purged; a route must be in the routing table in order to be in a holddown state. Because route poisoning is in use, the same situation exists on the other routers, as well. Very quickly, they will be ready to accept the new metric.
5. Router D advertises the new metric to Router C, and even if Router C had not poisoned and removed the route from its routing table, but instead had this route in holddown, it would accept the new metric, because it is from the source of the original advertisement—Router D.
6. The same effect trickles down to Routers C, B, and A in that triggered updates cause route poisoning and subsequent triggered updates. So none of them have the failed route in their routing table, nor do they have any of the routes that were accessible through the failed link, including Router F's Ethernet network, which allows their routing table entries to be updated almost immediately by the less desirable metrics. In other words, even if the advertiser of the original route was not advertising a worse metric, which itself is grounds for believing the worse metric, the fact that poisoning removed the routes from their routing tables makes these new updates look like routes they never heard of, causing them to be learned without incident.

Without triggered updates and route poisoning, the time required for Router A to converge would be the detection time, plus the holddown time, two update times, and another update time. The complete convergence to Router A could be over 240 seconds. With these mechanisms working for you, however, convergence will occur in an internetwork of this size in about 15 seconds. It will benefit you, though, to understand the logic behind the scenario that results in the convergence time of roughly 240 seconds.

If the network comes back up, both Routers D and F would immediately broadcast RIP requests for the neighbor's entire routing table, toward each other across the link between them in order to try to speed up convergence without taking a chance on waiting for the other device to start advertising across the link on its own. Instead of the normal broadcast or multicast, the response is a unicast back to the requesting interface. The industrious reader could model this scenario in a lab environment, if available. You could intentionally administratively shut down an interface, while watching the results of the `debug ip rip` command, and then bring the interface back up, allowing all that has been presented here to be observed.

IGRP Convergence

Despite the leaps and bounds Cisco has taken in improving RIP convergence, IGRP still converges according to the standard theory, by default, taking quite a bit of time if left to its own devices, but resulting in an environment that is more resistant to some forms of network instability. IGRP resets its invalid and holddown timers each time an update is received, including triggered updates, but IGRP does not immediately reset its flush timer when it receives a triggered update that a route is down. It waits until the next scheduled update time to start the flush timer. What this means is that the flush time could be as much as 90 seconds longer than configured, when measured from the triggered update that advertises the route as unreachable. A well-placed `clear ip route *` command will speed things along, though.

The following process can be modeled in the lab by issuing the `shutdown` command on the interface of Router D that faces Router F. Using Figure 1.2 as an example, let's take a look at IGRP convergence, keeping in mind that the following enumerated list is not necessarily in chronological order, nor could an exact order be guaranteed:

1. Router D detects the failure on the link to Router F. Router D poisons this directly connected route in its own routing table by removing it, as well as the route for the Ethernet segment off of Router F, because this link was in the path to get there. Router D sends a triggered update to Routers C and E.
2. Router F detects the failure on the same link and poisons the route locally, as well as any routes with Router D's nearest interface address or Router F's interface on that link as a next hop. Router F sends out a triggered update to Router E, detailing these lost routes.
3. Router C sends a triggered update to Router B, and Router B sends one to Router A. Routers A, B, C, and E all start invalid and holddown timers for each of the inaccessible routes, unless there was one or more equal-cost paths (or unequal, with use of the `variance` command) to one or more of them, in which case the downed route will be removed and all traffic will use the remaining route or routes. At the next scheduled update time for each route, the routers will start the routes' flush timers, unless the original source notifies the routers that the links are back up, in which case the route is reinstated as an operational routing table entry.

One tricky point of contention here is that the routing table will likely say "`is possibly down`" next to the destination network, even when the route has been reestablished. Attempts at verifying connectivity through `ping` or `traceroute` should meet with success, regardless. Some nominal amount of time after the holddown timer expires, you'll be able to observe the route entry returning to its normal state. Issuing the `clear ip route *` command will speed the process along.

4. Router D broadcasts a request to Router C. Both Router D and Router F broadcast a request to Router E, basically to all remaining active interfaces, asking for the entire routing table of each router in hopes of jump-starting new methods of access to the lost networks. Router C sends back a poison-reverse response to Router D for those routes it originally learned through Router D that are affected by the outage. Router E does the same for Router F.

5. It's where Router D and its downed WAN link are concerned that Router E could create a slight mess initially. The good news is that Router D will learn the alternate—and currently the only—route to the Ethernet segment off of Router F. Router D's request may well arrive at Router E before Router F's triggered update, as three keepalives must be missed before Router F will consider the link down and you manually shut the link down on Router D's side. In such a case, Router E will unicast its reply to Router D that the downed link is available through Router E. This is because Router E had an equal-cost alternative path to the downed network that it learned through Router F. So, to advertise Router E's alternative path to Router D as accessible is not a violation of the poison-reverse rule. That's only for affected routes that Router E learned through Router D, of which there are none. Router E is content to advertise only a single route for each unique destination, and the route through Router F will do nicely—or will it? Because Router D removed its own directly connected route entry, there is nothing stopping it from using this new advertisement that once looked suboptimal when Router D itself had direct access to this network. The triggered update from Router F initiates a triggered update from Router E, and a subsequent resetting of all the appropriate timers, which serves to set Router D straight that the network is truly down.
6. At this point, it appears that the route is possibly down, but the path through Router E apparently has become engraved in the routing table, while attempts to ping an interface on the downed network will no doubt fail. This is only IGRP's optimism showing through, erring on the side of too much information. Remember, even after the link has been re-established, this confusing entry will remain, and the holddown timer will have to expire before the entry is cleaned up and joined by its equal-cost partner. Again, `clear ip route *` works like a charm here to get you back to where you feel you should be. Additionally, watching the festivities through `debug ip igrp transactions` will clarify this process and reassure you that Router D has been informed that the route truly is down, in any direction. Still, Router D is confused enough to return alternating destination unreachable messages, instead of the usual timing out, during ping, as long as the network remains down.
7. Router D then sends a triggered update out all active interfaces participating in the IGRP routing process for the appropriate autonomous system, which includes this new entry.
8. Routers A, B, and C receive this update in turn. They would ignore the new route since it is in holddown, but because each one receives the update from the source of the original route, they each implement the new route, although the entry in the table will not appear to change until after the holddown timer expires. Because they point to the previous next hop still, connectivity will tend to be maintained. While in holddown, each router will continue to use poison reverse for the affected routes, back toward their respective advertising router.
9. Once all holddown timers expire, respective routing tables are then updated with an accurate routing table entry.

Without triggered updates and dumb luck or the smart design of the IGRP protocol that gets these routers to continue to use suspect routes to successfully pass traffic, the time it could take for Router A to converge could be the detection time, plus the holddown time, plus two update times, plus another update time, which is over 490 seconds.

EIGRP Convergence

Let's take a look at the convergence time of Enhanced IGRP (EIGRP). We will again use Figure 1.2 to help describe the convergence process:

1. Router D detects the link failure between Routers D and F and immediately checks its topology table for a feasible successor. We will assume Router D does not find an alternate route in the topology table and puts the route into active convergence state. In reality, taking bandwidth and delay into account, all that must be true for the path through Router E to be a feasible successor and for the process to stop right here is for the metric from Router E to the Ethernet segment off of Router F (the reported distance, RD, from Router E for this route) to be less than Router D's metric of the route that just went down (the feasible distance, FD). If this is the case, this path will be in the topology table, and convergence will already be over. I give you the beauty of EIGRP. Beware the ugliness, of which being a proprietary protocol is a good example.
2. Router D sends a QUERY message advertising the routes that it lost with infinite metrics (4,294,967,295, same as for IGRP) out all active interfaces looking for a route to the failed link and affected networks. Routers C and E acknowledge the QUERY.
3. Router C sends back a REPLY message advertising the routes requested with infinite metrics. Router D acknowledges the REPLY.
4. Router E sends back a REPLY message with routes to the networks that Router D lost, including to the downed network, thinking that it still has the alternate, equal-cost route to offer, not yet having been informed by Router F of its demise. Router D acknowledges the REPLY.
5. Router D places the new routes in the topology table, which then updates the routing table, due to their unopposed selection as successors.
6. Because both neighbors sent back REPLY messages, Router D sends both of them UPDATE messages, thinking that its local routing table has settled down with the changes. Because Router C has been using poison reverse, Router D updates it with the two new routes it learned. But because it learned these from Router E and has nothing new from Router C, the UPDATE to Router E is blank. Return UPDATE messages are often considered acknowledgments for earlier UPDATE messages, with no separate acknowledgment messages necessary.
7. Router C responds with an UPDATE, which Router D acknowledges. Router E sends an UPDATE with the link between Router D and Router F, which it still thinks is accessible through Router F. The reason Router E includes it in an UPDATE message is because it once thought there were two equal-cost paths to get there. Any such changes are eventually sent in an UPDATE message, because these are the messages that suggest the dust has settled and these are the results.
8. However, shortly thereafter, Router E learns from Router F that the network between Router D and Router F is truly down, and immediately sends out a QUERY to Router D looking for another path to the downed network, which also serves to notify Router D that the network is inaccessible through Router E now, as well. Router D acknowledges the QUERY.

9. Router D sends a REPLY, which is acknowledged by Router E, advertising that it too has an infinite metric to that network. Router D and Router E now consider themselves synchronized. Router D also updates its own EIGRP tables with the fact that the network is lost, but it still knows that the Ethernet segment off of Router F is accessible through Router E.
10. In response to this latest news from Router E, Router D sends a QUERY to Router C, just to make sure that it hasn't learned of this network in the meantime. Router C acknowledges this QUERY and sends back a REPLY message that confirms to Router D that no path exists to the downed network. After acknowledging this REPLY, Router D and Router C consider themselves synchronized. The EIGRP routing domain has converged.

Router A convergence time is the total time of detection, plus the query and reply time, plus the update propagation time—about two seconds total. However, the time can be slightly longer.

In case it was not apparent from the foregoing discussion, EIGRP employs various message types, including UPDATE, QUERY, and REPLY. It makes sense that UPDATE and REPLY messages can carry new routing information that could alter the receiving router's EIGRP tables. More subtly, QUERY messages also carry route information that the receiving router treats as new, with respect to the sending router. The QUERY/REPLY pair is invaluable to EIGRP to make sure that one or more specific routes are synchronized between the pair of routers exchanging these messages, especially when there was an earlier discrepancy. For example, in step 8 in the preceding description, Router E used a QUERY message, not an UPDATE message, to inform Router D that it agreed that the link between Router D and Router F was down.

Link-State Convergence

Using Figure 1.2 as a reference, let's now take a look at the convergence cycle used in link-state routing protocols within a single area:

1. Router D detects the link failure between Routers D and F. The route entry for that link and any dependent links are removed from Router D. A link-state advertisement (LSA) for OSPF, or a link-state PDU (LSP) for IS-IS, is sent out all eligible OSPF or IS-IS interfaces on Router D.
2. Routers C and E receive the LSA or LSP and forward it out to all eligible interfaces, which are normally all active interfaces except for the interface where the LSA was received, unless Router C is the OSPF-designated router of the Ethernet network it shares with Router D and Router D is not the backup designated router. If that's the case, then Router C will flood the LSA back out that interface also, as part of its duties as a designated router.
3. All routers wait five seconds, by default, and then run the shortest path first (SPF) algorithm. After the algorithm is run, Router D adds the route through Router E, and Routers C, B, and A update the metric in their routing table to that route.

4. After what could be another 30 seconds, Router F sends out an LSA or LSP to all eligible OSPF or IS-IS interfaces after timing out the link to Router D. All routers wait five seconds after receipt of the advertisement and then run the SPF algorithm, and all routers now know that the route to the Ethernet segment off of Router F is through Router E.

Router A convergence time is the time of detection, plus the LSA forwarding time, plus five seconds. This is about six seconds. However, if Router F's time to converge is considered, then the time can be about 36 seconds.

RFC 2328 is suggested reading for those interested in learning just about all they ever wanted to know about how OSPF operates at the nuts-and-bolts level.

Verifying and Testing a Route

Verifying and testing routes are very important topics to understand. If you understand all of the details that go into making a map, but you do not understand how to read that map and drive a car, you will be lost.

We start this section with an explanation of how to verify what routes are in a routing table and conclude the section with a way to test the connectivity to the routes.

Verifying Routes

Verifying routes is actually a simple item to understand. No matter what routing protocol or, for that matter, routing protocols the router has in use, the process is the same.

You will first log into the router, which will be in user EXEC mode. You will know you're in this mode because the router name will be followed with a > symbol:

```
2501A>
2501A>show ip route
```

After you enter the command, the routing table will be displayed:

```
2501A>sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B – BGP D -
EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area N1 - OSPF NSSA
external type 1, N2 - OSPF NSSA external type 2 E1 - OSPF external type 1, E2 -
OSPF external type 2, E – EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2,
* - candidate default U - per-user static route, o - ODR, P - periodic
downloaded static route T - traffic engineered route


Gateway of last resort is not set
```

```
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.50.0 is directly connected, FastEthernet0/0
C       192.168.24.0 is directly connected, FastEthernet0/0
R       175.21.0.0/16 [120/1] via 10.10.10.1, 00:00:18, Serial0
2501A#
```

You will now be able to verify all connected, statically defined, and dynamically learned routes.

As you can see, it's easy to verify the routes the router knows. After discovering the routes on the router, you can start testing the connectivity to a route.

Remember, if you ever want to clear all the routes in your routing table, use the command `clear ip route *`. This will cause your router to purge its routing table and relearn all active routes. As you've seen, this is very useful in case you want to get the most up-to-date routing information.

Testing and Troubleshooting Routes

What you need to understand at this point is that the tools you will use to test connectivity will also be used to troubleshoot connectivity issues.

There are two tools that can be used for these tasks:

- Ping
- Traceroute

One of the tools you should use in the testing and troubleshooting phase is Ping. The `ping` command is used to test IP connectivity to a destination. Ping uses ICMP to accomplish this task. With debugging turned on for ICMP packets, let's take a look at how Ping accomplishes this:

```
3640#debug ip icmp
ICMP packet debugging is on
3640#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
3640#
2d01h: ICMP: echo reply rcvd, src 10.10.10.1, dst 10.10.10.2
2d01h: ICMP: echo reply rcvd, src 10.10.10.1, dst 10.10.10.2
2d01h: ICMP: echo reply rcvd, src 10.10.10.1, dst 10.10.10.2
2d01h: ICMP: echo reply rcvd, src 10.10.10.1, dst 10.10.10.2
2d01h: ICMP: echo reply rcvd, src 10.10.10.1, dst 10.10.10.2
```

So, what happened? Router 3640 sent an ICMP echo to 10.10.10.1 on router 2501. Router 2501 received the ICMP echo from router 3640 and sent an ICMP echo reply telling router 3640 the packet has reached its destination of 10.10.10.1 on router 2501, signifying a successful ping. If the destination network were unreachable, when router 2501 received the ICMP echo from router 3640, it would have dropped the packet and returned an ICMP destination unreachable message.

Now that you understand the concept of Ping and how it works, you need to learn how to implement it. Using Ping is relatively simple. All you need to do is enter the command **ping** followed by the address or host name of the device you want to ping (omitting the address/host name parameter from the command will begin the extended ping dialog, allowing you to alter the default settings and have more control over the process):

```
3640#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```

Let's examine the information you receive back:

```
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
```

Sending 5 This means you are sending five packets.

100-byte The size of each packet.

ICMP Echos The type of packet sent.

10.10.10.1 The destination address.

timeout is 2 seconds The packet will be deemed dropped if an echo reply is not received within two seconds.

The ! symbol represents a successful ping. A ping that has timed out would be represented by a period, such as:

```
.....
```

Let's examine the last line of the sequence:

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```

Success rate The percentage of successful packets sent and received.

(5/5) This means five packets were sent and five packets were received back.

round-trip min/avg/max These values represent the shortest, average, and longest times it took to receive an ICMP echo reply.

Ping may be an easy concept to grasp, but it is the one tool you will use the most as a network engineer.

The other tool that will be used for testing and troubleshooting a route is Traceroute. The `traceroute` command gives you a router-by-router account of the path a packet takes to get to a destination. It does not, however, supply information on the return path of user or ICMP packets, which could be different, depending on reverse-route selection among the routers in the internetwork. Traceroute is best used when you need to discover the location where the packet is being dropped or a routing loop occurs.

Traceroute takes advantage of the time to live (TTL) field of an IP packet. The value of the TTL field represents how many layer 3 devices (hops) a packet can enter before it is dropped. Traceroute exploits this by setting the TTL to a value of 1 in the IP header of a UDP port 33434 packet that it sends toward the destination.

IANA reserves this TCP/UDP port number for traceroute use. There are also about 800 additional port numbers following this value that are unassigned and possibly available for traceroute use. The key is to use a port number that will not be active on the destination device.

The packet will reach the first router in the path, which will, as one of its first layer 3 tasks, decrease the TTL by 1 to 0 and drop the packet with no further processing. The executing router sends an ICMP time exceeded message to the traceroute originator. The originator then increases the TTL by 1 to a value of 2 and sends the packet toward the destination. The packet reaches the first router in the path and the TTL is decreased by 1 to a value of 1. That router then forwards the packet toward the second router in the path to the destination. The second router then decreases the TTL by 1 to a value of 0.

At this point, the packet is dropped and an ICMP time exceeded message is sent to the originator. This process continues until the destination is reached or the maximum TTL (30, by default) has been used. The originator displays the identity of the executioner upon receipt of each time exceeded message, creating a sequenced list of devices between the traceroute originator and target. The originator knows the trace is over when it receives an ICMP destination port unreachable message from the traceroute target, indicating that the packet made it all the way to the intended recipient, and there are no more intermediate devices to discover. Cisco devices offer an extended `traceroute` command, while in privileged EXEC mode, that can be used to adjust defaults, like maximum TTL and source IP address.

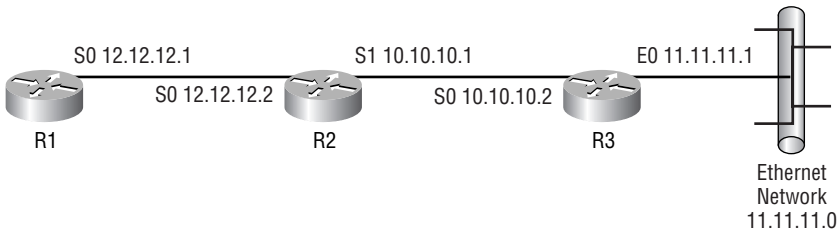
All you need to do in order to use the basic `traceroute` command is to enter **`traceroute`** followed by the destination address. Just entering **`traceroute`** will begin the dialog for the extended `traceroute` command. Here's an example of a successful traceroute from router R1 in Figure 1.3:

```
R1#traceroute 11.11.11.1

Type escape sequence to abort.
Tracing the route to 11.11.11.1

  1 12.12.12.2 12 msec 12 msec 12 msec
  2 10.10.10.2 24 msec 24 msec *
R1#
```

FIGURE 1.3 Traceroute



Here's what's going on in Figure 1.3:

1. R1 sends a packet toward the destination with the TTL set to a value of 1.
2. R2 receives the packet, decreases the TTL by 1 to a value of 0, and sends an ICMP time exceeded message back to R1.
3. R1 sends another packet toward the destination with the TTL set to a value of 2.
4. R2 receives the packet, decreases the TTL by 1 to a value of 1, and forwards to packet to R3.
5. R3 receives the packet, realizes that the destination of the packet is itself, looks in the protocol field of the IP header, and finds that UDP is the next protocol to get the datagram. Once R3's UDP process sees that the destination port of 33434 is not an active application, R3 triggers an ICMP destination port unreachable message back to R1, which ends the trace.

From this, you learn there are two hops to the destination. The numbers preceding the lines correspond to the value of the TTL for that series of three packets (by default). You can change the minimum and maximum TTL values in the extended Traceroute utility, which will cause a value other than 1 in the first column of the first line to be reported. Tracing to a non-existent address on a network known to the originator can cause multiple lines of asterisks to display until the maximum TTL value is reached, at which point the trace ends (just in case you want to see that happen in a production environment without establishing a career-limiting routing loop).

If the packet had made it through the first hop and not the second, you would have then been able to locate where the break in the connection occurred. If there had been a routing loop, the traceroute would have shown the path bouncing between two or more routers multiple times before the TTL expired. Asterisks (*) represent packets that went out but were unanswered before timing out, which is a period of three seconds, by default.

As you can tell, Ping and Traceroute are two valuable tools for testing and troubleshooting routes. Trust me, you will come to love these two commands in your production environment.

This is a very important chapter to understand. You need to have a strong understanding of the information covered to be able to grasp the concepts that will be covered from here on out. If you don't feel you have fully grasped the concepts covered, please review this chapter until you feel comfortable with it.

Summary

Routing allows information from one network to be shared with a different network. In order for this to be accomplished, a router must understand how to reach the destination. This is accomplished through static or dynamic routing. Static routing requires you to manually configure the router paths to remote destinations. Static routing works well in a small network but doesn't scale well.

When the network you are on is a larger one, it's a better idea to use dynamic routing. Dynamic routing allows routers to dynamically discover the available paths to a destination. Using the metrics associated with the various advertised routes, the router is able to determine the best path to a destination.

Dynamic routing comes in two forms: distance-vector routing and link-state routing. Distance-vector routing protocols share their entire routing table with their directly connected neighbors; this means that a distance-vector routing protocol sees the network only from the perspective of its neighbor. These routing tables are broadcast or multicast out at fixed intervals. Distance-vector routing protocols work well for small networks but do not scale well to larger networks.

Link-state routing protocols work extremely well for large networks. They are less bandwidth intensive than distance-vector routing protocols and also provide a view of the entire network. Link-state routing protocols accomplish this by keeping a link-state database. The link-state database is a list of all possible routes to all destinations. Link-state routing protocols are less bandwidth intensive than distance-vector routing protocols because they send out routing updates only when a change has occurred to the network, unlike distance-vector routing protocols that send out their entire routing table at fixed intervals.

EIGRP is a Cisco-proprietary advanced distance-vector or distance-vector/link-state hybrid routing protocol. It routes by rumor and sees the network only from the perspective of its neighbors, just as distance-vector routing protocols do. But in addition, EIGRP sends Hello packets for neighbor discovery and as connectivity keepalives, builds tables other than the routing table, and does not send out periodic updates containing the entire routing table, just as link-state routing protocols do.

Exam Essentials

Understand how routers route data. Routers receive frames that should be, under normal circumstances with no special configuration, addressed to the router. The frame is discarded after CRC comparison passes. The router examines the layer 3 packet header information to determine the layer 3 address of the destination device. Armed with this information, the router performs a lookup in its own protocol-based (such as IP) routing table for the protocol that formatted the packet. If it finds a suitable entry in its table, it switches the packet to the outbound buffer of the exit interface for encapsulation in a frame suitable for the media for which that interface is configured. The match may be in the form of a default route, which will be used, as long as no longer prefix matches exist. If the router fails to find an entry that

matches the destination address and no default route is known, then the router drops the packet and sends an ICMP destination unreachable message back to the source of the packet, as determined by another examination of the layer 3 header.

Describe classful and classless routing protocols. RIPv1 and IGRP are classful routing protocols; RIPv2, EIGRP, OSPF, IS-IS, and BGP are classless routing protocols. Classful routing protocols do not send a subnet mask in routing updates; classless routing protocols do. This means that a classful routing protocol assumes classful IP boundaries when it does have enough information about the true subnet mask, resulting in the inability to have more than one subnet mask per classful IP network (no VLSM) or to separate subnets even of the same mask length by a different classful network (discontiguous subnets). A classless protocol does not have these restrictions, as long as automatic summarization is turned off.

Understand a routing table and how it is populated. A router's routing table contains the routes that the router will use to send packets to their destinations. When populating the routing table, the router first looks at the administrative distance of the route. A router will select the routes with the lowest administrative distance. If multiple routes exist to a destination with the same administrative distance, the route with the lowest metric will then be selected to populate the routing table.

Know the difference between distance-vector and link-state routing protocols. RIPv1, RIPv2, and IGRP are all distance-vector routing protocols. These protocols send their entire routing table to neighbors at fixed intervals. OSPF and IS-IS are link-state routing protocols. These routing protocols will send out an update only when a change has occurred to the network. EIGRP is known as a hybrid, or advanced distance-vector, routing protocol. It has characteristics of both a distance-vector and a link-state routing protocol.

Understand what convergence time is. Convergence time is the amount of time that is required for all the routers within a routing domain to have the same relative understanding of a network from their individual viewpoints. Because distance-vector routing protocols send their entire routing table at fixed intervals, they require more time to converge than link-state routing protocols.