



# Redistribution

---

The highest courage is to dare to be yourself in the face of adversity. Choosing right over wrong, ethics over convenience, and truth over popularity...these are the choices that measure your life. Travel the path of integrity without looking back, for there is never a wrong time to do the right thing.

—Author unknown

By now, you have discovered that minimizing routing tables and choosing the next-hop destination path are critical for a well-tuned IP network. When routing information from one routing protocol, such as OSPF, is redistributed into IGRP, you must be mindful of possible routing loops.

A *routing loop* is a path to a remote network that alternates between two routers that assume the path is reachable via each other. Hence, the time to live that is present in every IP packet expires, and the packet or user data is dropped, resulting in loss of network connectivity.

Routing that uses a single routing algorithm is usually more desirable than running multiple IP and non-IP routing protocols, especially from a configuration and troubleshooting perspective.

However, in today's changing networks and with mergers, department politics, and acquisitions, more than one IP routing protocol is likely in use.

Before “diving in,” you first need a definition of *redistribution* and *summarization* to clarify your understanding of these concepts. These are potentially confusing topics. Hopefully, they will be clearer to you after reading this chapter and Chapter 7, “Summarization.”

- **Redistribution** is when a router takes routing information it has discovered in one routing protocol and distributes it into a different routing protocol, thus allowing the redistribution of the first protocol's networks into the second routing protocol. For example, a router running RIP and OSPF has a need for those in the OSPF section of the network to know the routes in the RIP network. Redistribution is used to accomplish this feat of routing magic! Sometimes, redistribution is referred to more specifically as *route redistribution*—both terms mean the same thing.
- **Summarization** is the taking of multiple route entries and representing them by a smaller number of routes.

This smaller representation is known as *summarizing your routes*. For example, this technique is used to minimize the routing table size, which is useful when connecting to the Internet, conserving router resources, or simplifying the next hop. For example,

routing table entries represent blocks of addresses and use the subnet mask to determine the size of the block. Through summarization, the router can tell the difference between 10.20.30.0 /24 and 10.20.30.0 /22. The former is a Class C–sized block of addresses, 256 addresses. The latter is four Class C–sized blocks, or 1024 addresses.

In many books and in most chapters of this book, the case studies are found at the end of each chapter. I am going to alter that in this chapter because redistribution and summarization go well together. To build on these concepts logically, I include several configuration examples within the chapter as built-in case studies and then add to them in each section as you proceed through the chapter.

Redistribution and summarization are interesting concepts. This chapter attempts to decipher and demystify the challenges that you face when one routing algorithm is redistributed into another, when one of those protocols is OSPF (of course), or when the OSPF routing table is optimized through summarization.

## OSPF Redistribution

Given the brief overview of redistribution, take a moment and consider when you might encounter redistribution or have a need to implement it. Many people have given lists of when you would use redistribution, but in reality, you need to know just four categories:

- **Politics**—A decision-maker has forced the running of another protocol besides OSPF. Work on this person—who needs to see the benefits of OSPF; however, until the decision is changed, you must use redistribution.
- **Profit**—Perhaps, there has been a merger of two companies with different protocols, or different needs exist within different divisions of the same company and network and the network cannot afford to be without connectivity, so you are called upon to “make it happen.”
- **Certification**—You are studying for a networking certification and you need to practice, practice, practice all aspects of protocol operation and interaction.
- **Hardware support**—You might have legacy hardware that does not run OSPF or does not run it reliably. This is the number one reason to redistribute RIP into OSPF.

In defining the concept and outlining the process of redistribution, I have presented the proper information on what is happening according to today’s accepted terminology and usage. The word *redistribution* does not accurately describe what is going on, and it is rather counterintuitive. When you say, “I am redistributing RIP into OSPF,” this sounds like RIP is being told to export route information to OSPF. However, the truth is that the OSPF routing process is being told to *import* routes from the RIP routing process.

Therefore, conceptually, it makes more sense to think of it as *exporting* or *importing* routes from one routing process to another. These terms are common when using BGP and more accurately reflect what is actually going on. For example, if you configured redistribution from RIP to OSPF, the configuration might look as follows on the router:

```
router ospf 200
 network x.x.x.x
 redistribute rip metric 100
 default-metric 100
```

In reality, you can rephrase this router configuration to make a bit more sense, as follows:

```
router ospf 1
 import routes from RIP,
 throw away the RIP metrics
 replace them with OSPF metric 100
```

## Administrative Distance and Metrics

Regardless of the reason that you have encountered redistribution, there are some characteristics of how it operates within OSPF and on Cisco routers. When redistributing from one routing protocol to another, keep in mind the following items:

- Administrative distances
- Routing metrics

Previous chapters have discussed the metrics used by OSPF and how to manipulate them. Administrative distances help with route selection among different routing protocols, but they can cause problems for redistribution. These problems can be in the form of routing loops, convergence, or inefficient routing.

The key point here is that Cisco routers always make a forwarding decision based on the administrative distance of a route over any metric that it might have. For example, a router might learn about two routes to the same destination, and each route has a different metric. Which route should the router believe and use?

Administrative distance is the deciding factor in this regard because the routing protocol with the lowest administrative distance has its route believed and thus preferred. The various default administrative distances used by Cisco routers and associated with the method of sourcing are shown in Table 6-1.

**Table 6-1** *Administrative Distances by Source*

Source of the Route	Default Administrative Distance
Connected interface	0
Static route	1
Enhanced IGRP summary route	5
External BGP	20

*continues*

**Table 6-1** *Administrative Distances by Source (Continued)*

Source of the Route	Default Administrative Distance
Internal Enhanced IGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
External Enhanced IGRP	170
Internal BGP	200
Unknown	255

---

**NOTE**

As soon as a route has been redistributed, the route is then “present” in two routing protocols, and thus the routing protocol with the better (that is, lower) administrative distance is preferred for that route.

---

The next section looks at the various rules that should be considered when redistribution is in place.

## Redistribution Golden Rules

The parameters surrounding the use of redistribution have a set of rules that if used when designing, implementing, and troubleshooting can make the use of redistribution in your network a much more trouble-free process. These rules can be summarized as follows:

- 1 Whenever possible, redistribute from a less powerful routing protocol to a more powerful routing protocol**—This rule can be implemented easily by giving two examples; the first is that you would always want to redistribute RIP into OSPF, and the second is that you would always want to redistribute OSPF into BGP. You would never want to do the opposite. Redistributing BGP into OSPF, for example, is a bad design for obvious reasons.

---

**NOTE**

This section discusses the golden rule regarding redistribution into a more powerful routing protocol. The most accepted method of ranking protocols from most to least powerful is as follows: BGP, OSPF, IS-IS, EIGRP, IGRP, and RIP.

---

- 2 **Whenever possible, redistribute at only a single point**—While this is sometimes hard to accomplish when a network requires a high degree of redundancy, the presence of just one redistribution point between the same routing domain greatly reduces the burden on your networking staff and smoothes the operation of the network.
- 3 **Filter routes**—The use of filters is important when redistributing because they help ensure that you have a split-horizon mechanism in place to reduce the chance of a routing loop. A subset of this rule is to never redistribute unfiltered, unverified routes into another routing domain. So when in doubt—filter!
- 4 **Redistribute only what is needed, not what is possible**—When redistributing, do so with this rule in mind because this allows you to know what routes need to be redistributed and how to construct your filters. Ask yourself, “What information does each routing domain need to know?”
- 5 **Adjust administrative distances and routing metrics**—This rule refers to the earlier part of this section, where these concepts and their relevance to redistribution were discussed. In many cases, metrics vary so much between routing protocols that it is advisable to set them when redistributing to ensure that the routes are operating as you want them to.
- 6 **Always set a metric**—In Cisco IOS Software, you can enter the **redistribute** command without a metric into the router’s configuration. However, you should always include the metric or default metric commands (see the following); if you do not, redistribution does not work, although it appears to be correctly configured.

If you follow these rules, you can effectively design how you need to summarize routes within your OSPF network.

---

**NOTE**

Cisco IOS Software automatically redistributes between Interior Gateway Routing Protocol (IGRP) and Extended Internet Gateway Routing Protocol (EIGRP) when the same autonomous system (AS) is defined. This is the only form of automatic redistribution that Cisco IOS Software performs. All other methods must be manually configured.

---

## Redistribution Configuration

This section reviews the command that is used to configure redistribution on a Cisco router. Note that the **redistribute** command can be used in any routing protocol, not just OSPF.

Redistributing routes into OSPF from other routing protocols or from static routes causes these redistributed routes to be labeled in the routing tables as OSPF external routes. This makes sense if you recall the external route discussion in Chapter 5, “Routing Concepts and Configuration.” Because these routes are from outside the OSPF process, they are considered to be external.

To redistribute routes into OSPF, use the following command syntax in router configuration mode:

```
redistribute routing-process [process-id] [metric ospf-metric-value] [metric-type ospf-metric-type-value] [route-map map-tag-value] [subnets]
```

This syntax is an abbreviated example of the full **redistribution** command, which is much more complex, as you will see at the end of this section.

Ensure that you consider the golden rules and read this section prior to activating redistribution. Of the following command options, only *routing-process* is required. Each of the following fields, shown in italics, requires a decision and input from you when configuring redistribution:

- *routing-process*—Identifies the routing process to redistribute into OSPF. The routing process can be BGP, Connected, EGP, EIGRP, IGRP, ISIS, ISO-IGRP, Mobile, ODR, OSPF, RIP, or Static.
- *process-id*—The process ID of the routing process (if applicable). For example, if you were redistributing EIGRP into OSPF, the autonomous system number used by EIGRP would be used as its process ID when redistributing.
- *ospf-metric-value*—The metric or cost to assign to the redistributed routes when they are placed into OSPF. If this option is not used, a default metric of 1 is used for redistributed BGP routes and a default metric of 20 is used for all other protocols. The range of values is 0 to 16,777,214. You should set the metric value here that is applicable to your network. For example, a network based on 10-Mbps Ethernet would need a different metric for routes than a network based on Frame Relay.
- *ospf-metric-type-value*—Routes are redistributed into OSPF as either an OSPF external Type 1 or Type 2 route. The default is Type 2; if you want to change the type of redistributed routes, you can do that here. However, changing the metric type here affects all routes.
- *map-tag-value*—A numerical value that is attached to the redistributed routes. The route tag is not specifically used by OSPF. However, the presence of a route tag can be used as a basis for making policy decisions as defined in a route map. For example, one use is to base the decision to redistribute a route based on the value present in the *route-tag* field. The default tag value is 0. The range of values for the tag is 0 to 4,294,967,295 (32 bits).
- **subnets**—If this command is not used when redistributing, only classful networks are redistributed into OSPF. With OSPF, the keyword **subnets** is typically used when redistributing more than one network into OSPF. When redistributing routes into OSPF, only routes that are not subnetted are redistributed if the **subnets** keyword is not specified.

A route map is a method used to control the redistribution of routes between routing domains. The format of the command to configure a route map is as follows:

```
route-map map-tag [[permit | deny] | [sequence-number]]
```

The next section discusses the various methods of filtering routing information between different routing algorithms.

To configure redistribution between routing protocols, use the following command under the routing process configuration:

```
redistribute protocol [process-id] {level-1 | level-1-2 | level-2} [as-number] [metric
metric-value] [metric-type type-value] [match {internal | external 1 | external 2}]
[tag tag-value] [route-map map-tag] [weight number-value] [subnets]
```

Table 6-2 explains the **redistribution** command syntax in further detail.

**Table 6-2** Command Syntax for Redistribution

Command	Description
<i>protocol</i>	Source protocol from which routes are being redistributed. It can be one of the following keywords: <b>bgp</b> , <b>connected</b> , <b>egp</b> , <b>igrp</b> , <b>isis</b> , <b>mobile</b> , <b>ospf</b> , <b>static</b> [ <b>ip</b> ], or <b>rip</b> .  The <b>static</b> [ <b>ip</b> ] keyword is used to redistribute IP static routes. The optional <b>ip</b> keyword is used when redistributing into the Intermediate System-to-Intermediate System (IS-IS) Protocol.  The <b>connected</b> keyword refers to routes that are established automatically by virtue of having enabled IP on an interface. For routing protocols such as Open Shortest Path First (OSPF) and IS-IS, these routes are redistributed as external to the autonomous system.
<i>process-id</i>	(Optional) For the <b>bgp</b> , <b>egp</b> , or <b>igrp</b> keyword, this is an autonomous system number, which is a 16-bit decimal number.  For the <b>ospf</b> keyword, this is an appropriate OSPF process ID from which routes are to be redistributed.
<b>level-1</b>	Specifies that for IS-IS Level 1, routes are redistributed into other IP routing protocols independently.
<b>level-1-2</b>	Specifies that for IS-IS both Level 1 and Level 2, routes are redistributed into other IP routing protocols.
<b>level-2</b>	Specifies that for IS-IS Level 2, routes are redistributed into other IP routing protocols independently.
<i>as-number</i>	Autonomous system number for the redistributed route.
<b>metric</b> <i>metric-value</i>	(Optional) Metric used for the redistributed route. If a value is not specified for this option and no value is specified using the <b>default-metric</b> command, the default metric value is 0. Use a value consistent with the destination protocol.

*continues*

**Table 6-2** *Command Syntax for Redistribution (Continued)*

Command	Description
<b>metric-type</b> <i>type-value</i>	(Optional) For OSPF, the external link type associated with the default route advertised into the OSPF routing domain. It can be either a Type 1 or Type 2 external route.  If a metric type is not specified, the Cisco IOS Software adopts a Type 2 external route.  For IS-IS, it can be one of two values: <ul style="list-style-type: none"> <li>• internal: IS-IS metric that is less than 63</li> <li>• external: IS-IS metric that is between 64 and 128</li> </ul> The default is internal.
<b>match</b> { <b>internal</b>   <b>external 1</b>   <b>external 2</b> }	(Optional) The criteria by which OSPF routes are redistributed into other routing domains. It can be one of the following: <ul style="list-style-type: none"> <li>• <b>internal</b>—Routes that are internal to a specific autonomous system.</li> <li>• <b>external 1</b>—Routes that are external to the autonomous system but are imported into OSPF as Type 1 external routes.</li> <li>• <b>external 2</b>—Routes that are external to the autonomous system but are imported into OSPF as Type 2 external routes.</li> </ul>
<b>tag</b> <i>tag-value</i>	(Optional) A 32-bit decimal value attached to each external route. This is not used by OSPF itself. It can be used to communicate information between Autonomous System Boundary Routers (ASBRs). If none is specified, the remote autonomous system number is used for routes from Border Gateway Protocol (BGP) and Exterior Gateway Protocol (EGP); for other protocols, 0 is used.
route-map	(Optional) Route map that should be interrogated to filter the importation of routes from this source routing protocol to the current routing protocol. If not specified, all routes are redistributed. If this keyword is specified but no route map tags are listed, no routes are imported.
map-tag	(Optional) Identifier of a configured route map.
<b>weight</b> <i>number-value</i>	(Optional) Network weight when redistributing into BGP. The value is an integer from 0 to 65,535.
subnets	(Optional) For redistributing routes into OSPF, the scope of redistribution for the specified protocol.

## External Routes

In OSPF, external routes are classified either as Type E1 or Type E2; the difference is the calculation of the metric (cost) associated with each route. Specifically, a Type E2 route has a metric that is equal to the external cost, whereas a Type E1 route has a metric of the external cost plus the internal cost to that route. Remember the following points when performing redistribution:

- By default, all redistributed routes are automatically categorized as Type E2 by OSPF.
- You can alter the type of external routes during redistribution if you want.

You can select the type of external route that you want the new routes to be through the use of the **metric-type** keyword when using the **redistribute** command, as follows:

```
Trinity(config)#router OSPF 100
Trinity(config-router)#redistribute rip metric-type ?
  1 Set OSPF External Type 1 metrics
  2 Set OSPF External Type 2 metrics
Trinity(config-router)#
```

The next section discusses the use of default routes in OSPF and as part of redistribution.

## Default Routes

The ability to generate and propagate a default route in a network is important. Chapter 4, “Design Fundamentals,” discussed the methods of default routes in OSPF areas. This chapter covers the redistribution of an existing default route into OSPF.

You can create a default route using one of the following two methods:

- Advertise 0.0.0.0 inside the domain, but only if the ASBR itself already has a default route.
- Advertise 0.0.0.0 regardless of whether the ASBR has a default route.

The latter method can be set by adding the **always** keyword to the end of the command; this in essence forces the route to always be present in the routing table. Be careful when using the **always** keyword. If your router advertises a default route (0.0.0.0) inside your OSPF domain and there is no default route present or a path to reach the destinations, routing is broken. Complete one of the following tasks to prevent this:

- Identify the default route in the configuration; this is accomplished in one of two ways:
  - Create a static default route (for example, **ip route 0.0.0.0 0.0.0.0 next hop ip\_address**)
  - Create a default network (for example, **ip default-network ip\_address**)
- Configure OSPF to propagate the default network that was just configured. To propagate the default network, use the **default-information originate always** command.

This section provided an overview of default routes, and as such, it built on knowledge that you should have regarding default routes. The next section takes a closer look at the **default-information originate** command and its use in OSPF.

## default-information originate Command

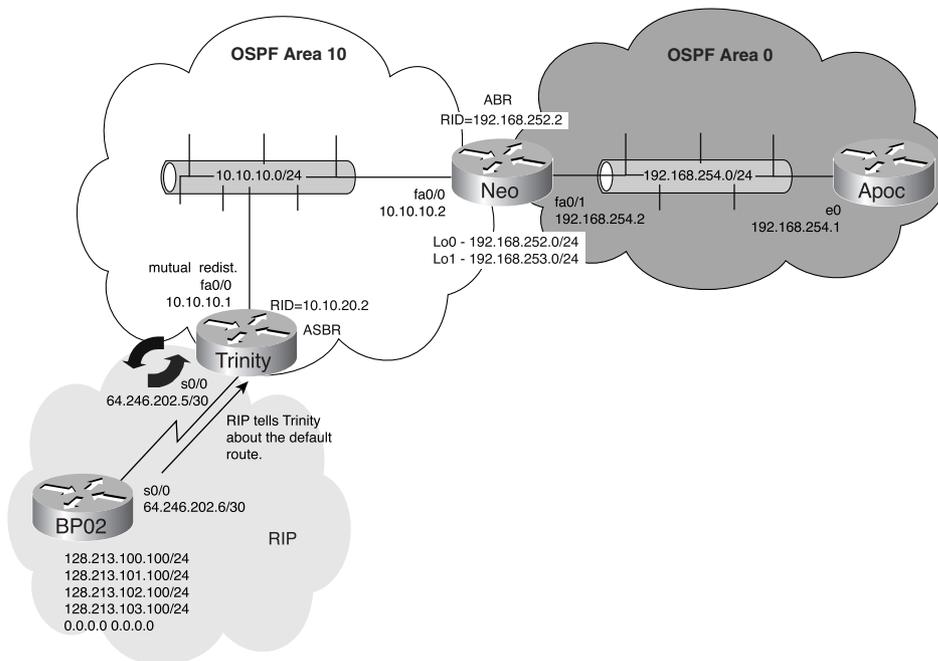
Chapter 4 discussed that, by default, normal OSPF area routers do not generate default routes. To have an OSPF router generate a default route, use the following command:

```
default-information originate [always] [metric metric-value] [metric-type type-value] [route-map map-name]
```

This command generates an external Type 2 link with link-state ID 0.0.0.0 and network mask 0.0.0.0. This makes the router an ASBR.

In a scenario where the default route for your network is via another routing process, you must be able to advertise this default route throughout the entire OSPF domain. Figure 6-1 shows that Router Trinity is receiving the default route from the RIP network.

**Figure 6-1** *Default Route*



Router Trinity thus places this default route into its routing table; however, it does not tell the rest of the network of its presence. Because this is the default route for the entire network, Router Trinity needs to advertise this route. To accomplish this objective, you must use the **default-information originate** command on the OSPF ASBR, Router Trinity, as shown in the following examples.

## Before

In this state, the default route is present in the routing table of Router Trinity. However, Router Neo does not yet have know about this default route, as shown in Example 6-1.

**Example 6-1** Router Neo's Routing Table Before the Advertisement of the Default Route on Router Trinity

```

Neo>show 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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

      64.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O E2   64.246.202.4/30 [110/200] via 10.10.10.1, 00:09:52, FastEthernet0/0
O E2   64.246.202.6/32 [110/200] via 10.10.10.1, 00:09:52, FastEthernet0/0
      128.213.0.0/22 is subnetted, 1 subnets
O E2   128.213.100.0 [110/200] via 10.10.10.1, 00:09:52, FastEthernet0/0
      10.0.0.0/24 is subnetted, 2 subnets
C       10.10.10.0 is directly connected, FastEthernet0/0
O       10.10.20.0 [110/2] via 10.10.10.1, 00:09:52, FastEthernet0/0
C       192.168.254.0/24 is directly connected, FastEthernet0/1
C       192.168.253.0/24 is directly connected, Loopback1
C       192.168.252.0/24 is directly connected, Loopback0
Neo>

```

## During

To “redistribute” the default route that Trinity knows about, use the **default-information originate** command in OSPF, as shown in Example 6-2.

**Example 6-2** Router Trinity Advertises a Default Route

```

Trinity#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Trinity(config)#router ospf 100
Trinity(config-router)#default-information originate ?
  always      Always advertise default route
  metric      OSPF default metric
  metric-type OSPF metric type for default routes
  route-map   Route-map reference
  <cr>

Trinity(config-router)#default-information originate
Trinity(config-router)#

```

**After**

After entering the command into Router Trinity's configuration, the default route is now present in the routing table of Router Neo, as shown in Example 6-3.

**Example 6-3** Router Neo's Routing Table After the Advertisement of the Default Route on Router Trinity

```

Neo>show 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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.10.10.1 to network 0.0.0.0

    64.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O E2   64.246.202.4/30 [110/200] via 10.10.10.1, 00:12:51, FastEthernet0/0
O E2   64.246.202.6/32 [110/200] via 10.10.10.1, 00:12:51, FastEthernet0/0
    128.213.0.0/22 is subnetted, 1 subnets
O E2   128.213.100.0 [110/200] via 10.10.10.1, 00:12:51, FastEthernet0/0
    10.0.0.0/24 is subnetted, 2 subnets
C       10.10.10.0 is directly connected, FastEthernet0/0
O       10.10.20.0 [110/2] via 10.10.10.1, 00:12:51, FastEthernet0/0
C       192.168.254.0/24 is directly connected, FastEthernet0/1
C       192.168.253.0/24 is directly connected, Loopback1
C       192.168.252.0/24 is directly connected, Loopback0
O*E2   0.0.0.0/0 [110/1] via 10.10.10.1, 00:01:34, FastEthernet0/0
Neo>

```

Notice in the routing table of Example 6-3 that the default route is a Type 2 external route, which as discussed is the default route type for redistributed routes. Also note that the metric has a value of 1. This is not the metric that you should be seeing.

**NOTE**

The default metric for OSPF routes according to Cisco documentation is 10; however, you can clearly see in the routing table that this is not true. Even adding the **default-metric** command to OSPF on Trinity has no effect in changing the value. I have tested this erroneous behavior on Cisco IOS Software versions ranging from 11.2(18) to 12.0(7), so be aware of it.

The question now becomes “How do you deal with this undocumented feature (that is, bug) and allow a more realistic metric to be given to a default route?” You can do this by adding the **metric** keyword to the **default-information originate** command in OSPF. Also, you can change the route from a Type 2 to a Type 1 route at the same time.

## Before

In Example 6-4, Router Trinity has the correct commands in its OSPF configuration; you should be seeing the default route with a metric of 50 (or at the very least 10!).

**Example 6-4** Router Trinity Configuration Before Altering the Default Metric

```
!
router ospf 100
 log-adjacency-changes
 summary-address 128.213.100.0 255.255.252.0
 redistribute rip metric 200 subnets tag 200
 network 10.10.10.0 0.0.0.255 area 10
 network 10.10.20.0 0.0.0.255 area 10
 default-information originate
 default-metric 50
!
```

Regardless, those metric values are not appearing as they should on Router Neo in Example 6-5, so you need to alter them and convert the default route to a Type 1 route.

**Example 6-5** Router Neo's Routing Table with the Original Default Route and Its Cost

```
Neo>show 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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.10.10.1 to network 0.0.0.0

     64.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O E2   64.246.202.4/30 [110/200] via 10.10.10.1, 00:25:17, FastEthernet0/0
O E2   64.246.202.6/32 [110/200] via 10.10.10.1, 00:25:17, FastEthernet0/0
     128.213.0.0/22 is subnetted, 1 subnets
O E2   128.213.100.0 [110/200] via 10.10.10.1, 00:25:17, FastEthernet0/0
     10.0.0.0/24 is subnetted, 2 subnets
C       10.10.10.0 is directly connected, FastEthernet0/0
O       10.10.20.0 [110/2] via 10.10.10.1, 00:25:17, FastEthernet0/0
C       192.168.254.0/24 is directly connected, FastEthernet0/1
C       192.168.253.0/24 is directly connected, Loopback1
C       192.168.252.0/24 is directly connected, Loopback0
O*E2  0.0.0.0/0 [110/1] via 10.10.10.1, 00:13:59, FastEthernet0/0
Neo>
```

## During

Example 6-6 shows the changes that you want to apply to router Trinity because it is the ASBR that is receiving and then redistributing the default route throughout our OSPF domain.

**Example 6-6** *Changing the Default Route Cost on Router Trinity*

```
Trinity#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Trinity(config)#router ospf 100
Trinity(config-router)#default-information originate metric 50 metric-type 1
Trinity(config-router)#^Z
Trinity#
```

**After**

After adding the commands to Router Trinity, you can check the routing table on Neo to see if the changes were successful in getting the results desired for the network (see Example 6-7).

**Example 6-7** *Router Neo's Routing Table with the Altered Default Route Metric*

```
Neo>show 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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.10.10.1 to network 0.0.0.0

   64.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O E2   64.246.202.4/30 [110/200] via 10.10.10.1, 00:29:15, FastEthernet0/0
O E2   64.246.202.6/32 [110/200] via 10.10.10.1, 00:29:15, FastEthernet0/0
   128.213.0.0/22 is subnetted, 1 subnets
O E2   128.213.100.0 [110/200] via 10.10.10.1, 00:29:15, FastEthernet0/0
   10.0.0.0/24 is subnetted, 2 subnets
C       10.10.10.0 is directly connected, FastEthernet0/0
O       10.10.20.0 [110/2] via 10.10.10.1, 00:29:15, FastEthernet0/0
C      192.168.254.0/24 is directly connected, FastEthernet0/1
C      192.168.253.0/24 is directly connected, Loopback1
C      192.168.252.0/24 is directly connected, Loopback0
O*E1  0.0.0.0/0 [110/51] via 10.10.10.1, 00:01:08, FastEthernet0/0
Neo>
```

The results at this point are mixed. You have the default route now being advertised as a Type E1 route; however, the metric has increased to 51, not 50 as configured. This is another issue with Cisco IOS Software, so be aware that in this case the metrics are cumulative, not set, as you would expect.

However, you are not done. There is a problem in the network where links are unstable. As a result of this, the default route is constantly being withdrawn and then added to the routing tables. This can have a negative impact on the network's operation and performance. The solution with OSPF is to add another keyword to the **default-information originate** command. If you apply the **always** keyword to the configuration for Router Trinity, Trinity always advertises the default route regardless of the status of its links. Example 6-8 demonstrates the configuration.

**Example 6-8** *Configuring Router Trinity to Always Advertise a Default Route*

```
Trinity#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Trinity(config)#router ospf 100
Trinity(config-router)#default-information originate metric 50 metric-type 1 always
Trinity(config-router)#exit
Trinity#
```

**CAUTION** Use care in applying the **always** keyword. If more than one default route exists in your network, the **always** keyword should not be used because it would cause some traffic to be discarded when another valid route that it could use was present.

There is no obvious effect on the route; however, if you were to shut down the serial link to the RIP domain, you would expect to see all RIP routes be removed from Router Neo's routing table, but the default route remains as a result of the **always** keyword. Check Example 6-9 to see if this is indeed the case.

**Example 6-9** *Router Neo's Routing Table After the **always** Keyword*

```
Neo>show 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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.10.10.1 to network 0.0.0.0

    10.0.0.0/24 is subnetted, 2 subnets
C       10.10.10.0 is directly connected, FastEthernet0/0
O       10.10.20.0 [110/2] via 10.10.10.1, 00:39:56, FastEthernet0/0
C       192.168.254.0/24 is directly connected, FastEthernet0/1
C       192.168.253.0/24 is directly connected, Loopback1
C       192.168.252.0/24 is directly connected, Loopback0
O*E1 0.0.0.0/0 [110/51] via 10.10.10.1, 00:11:49, FastEthernet0/0
```

Yes, the RIP routes are all gone, but the default route remains, so this worked as intended. In reviewing the concepts of external and default routes, this has reinforced some concepts with redistribution. Specifically, redistributed routes into OSPF are considered external routes, and a default route can also be redistributed. These characteristics might not be what you intended, so caution is warranted! Remember to go slowly with a plan and check the results at each step. The next section discusses how to assign routing metrics to a routing protocol as it is redistributed into OSPF.

## Assigning Metrics for Redistributed Protocols

The **redistribute** command enables you to assign a metric for the routes that are being redistributed into OSPF. On Cisco routers using OSPF, if you have not assigned a metric when redistributing, OSPF assigns a default value of 20 to each route. This metric might have unexpected consequences in your network. However, do not worry; Cisco has also provided a couple of “nerd knobs,” as covered in the following sections.

### Using the **redistribute** Command to Assign a Metric

You can set the metric of routes redistributed into OSPF on a per-protocol basis. Specifically, this means that when you are configuring redistribution into OSPF from EIGRP, you can have all EIGRP routes use an assigned metric. If having all routes with the same metric is acceptable to your network’s operation, review the configuration in Example 6-10.

#### Example 6-10 *Configuring a Routing Metric When Redistributing*

```
Tank#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Tank(config)#router ospf 100
Tank(config-router)#redistribute EIGRP 100 metric ?
<0-4294967295> Default metric

Tank(config-router)#redistribute EIGRP 100 metric 110
```

Therefore, the metric for all EIGRP routes in the OSPF network would have been set to 110 through this configuration.

### Using the **default-metric** Command to Assign a Metric

When you decide not to assign a metric using the **redistribute** command, you can use the **default-metric value** command in OSPF configuration mode. This command allows you to set the metric on all routes that are redistributed into OSPF without an assigned metric.

However, should you set the metric using the **redistribute** command, the **default-metric value** command does not alter what has been previously configured.

In the command output of Example 6-11, you can see that the range of metrics available to be assigned through this command is 1 to 4,294,967,295, thus giving you a metric for every conceivable need!

#### Example 6-11 *Range of the default-metric Command*

```
Trinity(config)#router ospf 100
Trinity (config-router)#default-metric ?
<1-4294967295> Default metric

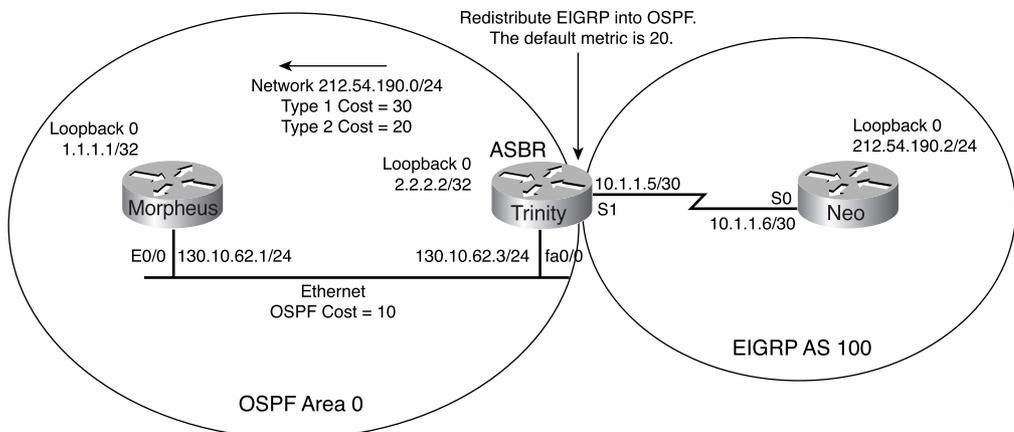
Trinity (config-router)
```

The **default-metric** command can be useful when redistributing OSPF. The next section presents some configuration examples that demonstrate how to implement and configure a router using this command.

## Configuration Example 1: Setting the Default Metric for Redistributed Routes

In Figure 6-2, Router Trinity is receiving the routes 212.54.190.0/24 and 10.1.1.4/30 from Router Neo via EIGRP. These EIGRP routes are initially redistributed into OSPF using the default metric of 20.

**Figure 6-2** *Default Metric Configuration*



Once the network begins routing OSPF, the first thing that you need to verify is that Routers Morpheus and Trinity have established an OSPF neighbor relationship. Use the **show ip ospf neighbor** command on both routers to check the state of their communication, as shown in Example 6-12.

**Example 6-12** *Verifying OSPF Neighbor Relationship*

Morpheus#show ip ospf neighbor						
Neighbor ID	Pri	State	Dead Time	Address	Interface	
2.2.2.2	1	FULL/BDR	00:00:30	130.10.62.1	Ethernet0/0	
Trinity#show ip ospf neighbor						
Neighbor ID	Pri	State	Dead Time	Address	Interface	
1.1.1.1	1	FULL/DR	00:00:38	130.10.62.3	Ethernet0	

Now that OSPF is working well, check the operation of EIGRP between Routers Trinity and Neo, as shown in Example 6-13.

**Example 6-13** *Verifying EIGRP's Operation*

Trinity#show ip eigrp neighbors								
IP-EIGRP neighbors for process 100								
H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q	Seq Cnt Num
0	10.1.1.6	Se1	10	00:13:27	399	2394	0	4
Neo#show ip eigrp neighbors								
IP-EIGRP neighbors for process 100								
H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q	Seq Cnt Num
0	10.1.1.5	Se0	11	00:13:53	28	2280	0	6

After confirming that routing is working well for the network, verify that Router Trinity is receiving the 212.54.190.0/24 network from Router Neo, as shown in Example 6-14.

**Example 6-14** *Identifying the Routing for EIGRP*

Trinity#show 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	
Gateway of last resort is not set	
	1.0.0.0/32 is subnetted, 1 subnets
O	1.1.1.1 [110/11] via 172.16.1.1, 00:13:37, FastEthernet0/0
	2.0.0.0/32 is subnetted, 1 subnets
C	2.2.2.2 is directly connected, Loopback0
	212.54.190.0/24 is subnetted, 1 subnets
D	212.54.190.0 [90/40640000] via 10.1.1.6, 00:13:38, Serial1
	130.10.62.0/24 is subnetted, 1 subnets
C	130.10.62.0 is directly connected, FastEthernet0/0
	10.0.0.0/30 is subnetted, 1 subnets
C	10.1.1.4 is directly connected, Serial1

Inspect the routing table in Example 6-15 for Router Morpheus to verify that the default cost of the redistributed route is in fact present, as you would expect.

**Example 6-15** *Verifying the Redistributed Default Route*

Morpheus#show 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	

**Example 6-15** *Verifying the Redistributed Default Route (Continued)*

```

Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
    2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/11] via 130.10.62.3, 00:27:10, Ethernet0/0
    212.54.190.0/24 is subnetted, 1 subnets
O E2   212.54.190.0 [110/20] via 130.10.62.3, 00:01:58, Ethernet0/0
    130.10.62.0/24 is subnetted, 1 subnets
C       130.10.62.0 is directly connected, Ethernet0/0
    10.0.0.0/30 is subnetted, 1 subnets
O E2   10.1.1.4 [110/20] via 130.10.62.3, 00:01:58, Ethernet0/0

```

As you can see, the redistributed EIGRP routes have a cost, or metric, of 20 by default. These routes were redistributed as OSPF external Type 2 routes, so the cost of crossing the Ethernet network is not included. For this network, the default metric for all redistributed routes should be 100 unless specified differently. Therefore, you should configure Router Trinity so that all redistributed routes, including EIGRP routes, are assigned a metric of 100, as shown in Example 6-16.

**Example 6-16** *Configuring the Default Metric for Redistribution*

```

Trinity#
router ospf 100
 redistribute eigrp 100 subnets
 network 2.2.2.2 0.0.0.0 area 0
 network 130.10.62.0 0.0.0.255 area 0
 default-metric 100

```

Next, in Example 6-17, verify that the redistributed EIGRP routes have now been assigned a metric of 100 per the configuration in Example 6-16.

**Example 6-17** *Verifying the New Default Metric*

```

Morpheus#show 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

Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
    2.0.0.0/32 is subnetted, 1 subnets
O       2.2.2.2 [110/11] via 130.10.62.3, 00:27:10, Ethernet0/0
    212.54.190.0/24 is subnetted, 1 subnets
O E2   212.54.190.0 [110/100] via 130.10.62.3, 00:01:58, Ethernet0/0
    130.10.62.0/24 is subnetted, 1 subnets

```

*continues*

**Example 6-17** *Verifying the New Default Metric (Continued)*

```
C      130.10.62.0 is directly connected, Ethernet0/0
      10.0.0.0/30 is subnetted, 1 subnets
0 E2   10.1.1.4 [110/100] via 130.10.62.3, 00:01:58, Ethernet0/0
```

As expected, the metric of the routes is now appearing with a value of 100. Example 6-18 shows the relevant portions of the router configuration files that are used for this configuration example.

**Example 6-18** *Relevant Configurations*

```
Morpheus#
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
interface Ethernet0/0
 ip address 130.10.62.0 255.255.255.0
!
router ospf 1
 network 1.1.1.1 0.0.0.0 area 0
 network 172.16.1.0 0.0.0.255 area 0

Trinity#
interface Loopback0
 ip address 2.2.2.2 255.255.255.255
!
interface FastEthernet0/0
 ip address 130.10.62.0 255.255.255.0
!
interface Serial1
 bandwidth 64
 ip address 10.1.1.5 255.255.255.252
 clockrate 64000
!
router eigrp 100
 network 10.0.0.0

router ospf 100
 redistribute eigrp 100 subnets
 network 2.2.2.2 0.0.0.0 area 0
 network 172.16.1.0 0.0.0.255 area 0
 default-metric 100

Neo#
interface Loopback0
 description Simulate the network 3.3.3.0/24
 ip address 3.3.3.3 255.255.255.0
!
interface Serial0
 bandwidth 64
 ip address 10.1.1.6 255.255.255.252
!
router eigrp 100
 network 212.54.190.0
 network 10.0.0.0
 no auto-summary
```

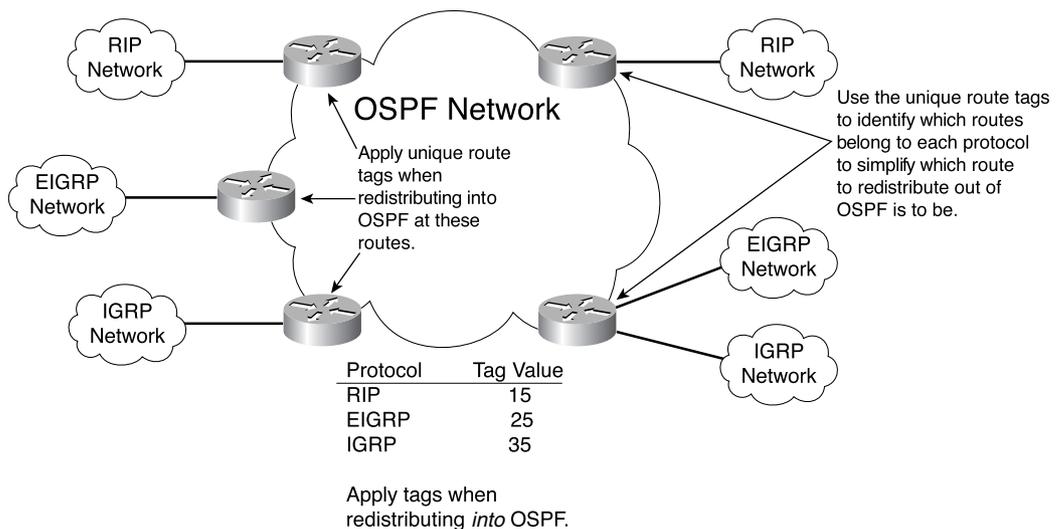
The next section discusses the concept of mutual redistribution and describes how mutual redistribution allows redistribution of routes between routing protocols at the same time. This is all accomplished through a concept known as *route tagging*.

## Route Tagging

Route tagging is one of the neatest possible ways of manipulating routes. Having learned how to manipulate routes the long way with access control lists, you are now ready for the more efficient manner of manipulation using route tagging. While many network engineers use route tagging as a way to filter routes (which is certainly a good use), there are many other uses. Although this is not a book on route filtering, I have included tagging in this redistribution chapter because it is of value in OSPF. A route tag is a number that is placed into each routing update and carried by that route until altered. A route tag allows that route to be identified, and the presence of a tag in no way alters the validity or integrity of the route. All the major IP-based routing protocols support route tagging.

Consider the simple network shown in Figure 6-3 and see how the network consists of an OSPF core surrounded by other routing protocols. This kind of situation typically occurs during business changes involving acquisitions or a major redesign of an existing network, but you can use it whenever the situation warrants.

**Figure 6-3** *When to Tag*

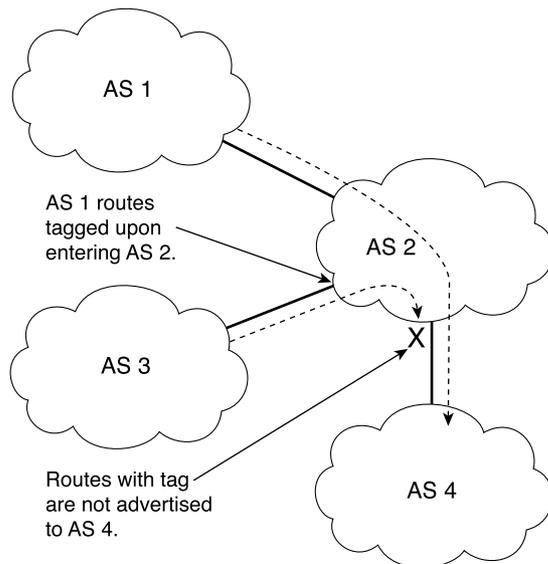


### NOTE

A route tag is similar in concept to a BGP community.

As shown in Figure 6-3, route tags are applied to keep the appropriate routes for each protocol together when redistributing into or out of OSPF. Another use for a route tag is to identify and filter routes. Figure 6-4 presents several different autonomous systems (routing domains) that are sharing routes. It is the routing policy of AS 2 to share the routes of AS 1 to AS 4. However, AS 2 also knows about the routes from AS 3, but those routes are not shared with AS 4. One of the easiest ways to configure your routers to follow this policy is through the use of route tags.

**Figure 6-4** *Tagging and Filtering Routes*



The challenge is to always tag the routes coming into AS 2 from AS 3 without having to manually change router configurations every time that AS 3 gets a new subnet or has an internal change. Therefore, when the routes from AS 3 are redistributed into AS 2, you tag them all. The routes are then shared throughout AS 2. On the router that is sending routes to AS 4, you have a route map that says “Do not advertise any routes that have this tag on them to AS 4”—very elegant and simple. This chapter looks at how to apply tags when redistributing during the configuration examples.

## Mutual Redistribution

So far, you have learned about redistribution, how to generate default routes, and how external routes represent routes that are redistributed into OSPF, but there is still a bit more to redistribution. Enter the last concept before getting into more configuration examples. This concept is known as *mutual redistribution*. As you should know by now, redistribution

is the process of importing route information from one routing protocol into another. The concept is further expanded through mutual redistribution, where routing protocol information is both imported and exported (kind of like “I tell you my routes and you tell me yours”). When beginning a mutual redistribution project, go through the following checks:

- 1 Deploy mutual redistribution between routing protocols carefully and in a controlled manner.
- 2 Use filters to ensure that unthrottled redistribution does not allow route information learned from a protocol to be injected back into the same protocol.

---

**NOTE**

Because OSPF is a link-state protocol and not a distance-vector protocol, you should be concerned about unthrottled routing information as opposed to split horizon. In other words, be careful not to inject a routing loop!

---

- 3 Deploy passive interfaces whenever possible to prevent routing loops or suboptimal routing.
- 4 Use route maps, not distribute lists, to filter OSPF routes because distribute lists have no effect in OSPF.

## Distribute List Concerns

This is an important point to note regarding the operation of OSPF. Recall that redistribution occurs on an OSPF ASBR because in OSPF an ASBR connects to other routing domains. This concept will become important shortly. There are two ways to apply a distribute list on a router:

- To affect routing updates coming inbound
- Applied outbound to affect the routing updates be sent by the router.

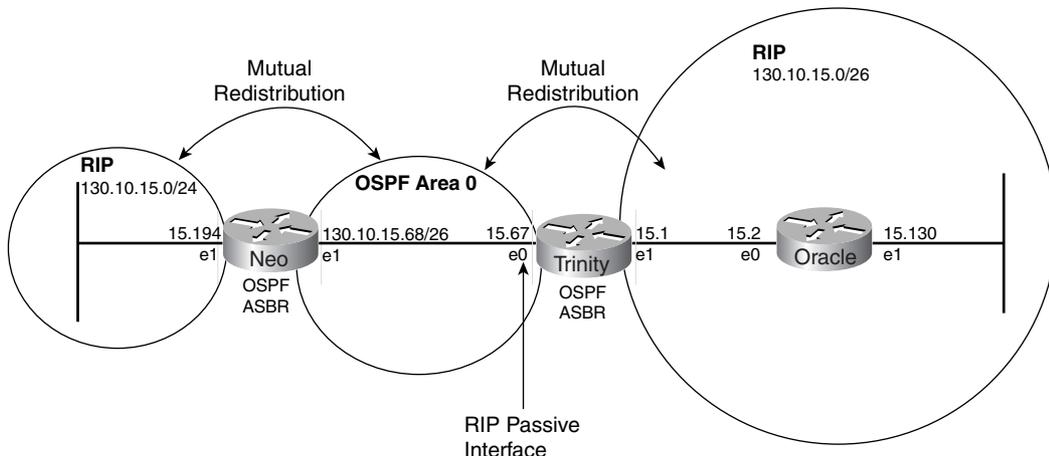
Use the following commands to achieve applying a distribute list:

- **distribute-list out**—When applying a distribute list outbound to filter OSPF routes on an ASBR into another protocol, you can expect this to work normally.
- **distribute-list in**—When applying a distribute list inbound to filter routes, the list prevents the routes from being placed into the routing table. However, the list does not prevent link-state packets with that routing information from being sent to connected routers, who then have the routes present in their routing table.

Overall, you should perform route filtering in other protocols before the routes enter the OSPF routing domain.

Refer to Figure 6-5, which shows a sample network that has Routers Neo, Trinity, and Oracle, all of which are running RIP. Notice that in this sample network, Routers Trinity and Neo are also running OSPF in the core of the network. (By the way, this network should not be considered a template for the real world; it is a bad design used to illustrate a concept.) Both Routers Trinity and Neo are doing mutual redistribution between RIP and OSPF.

**Figure 6-5** *OSPF and Distribute Lists*



Assume that you do not want the RIP routes coming from Router Oracle to be injected into the OSPF core (area 0), so you put a passive interface for RIP on interface E0 of Router Trinity. This prevents RIP from being sent out E0 into the OSPF area 0. However, you have still allowed the RIP routes coming from Router Neo to be injected into the OSPF core. Example 6-19 shows the configurations and the resulting routing table.

**Example 6-19** *Configurations Using Distribute Lists*

```
Oracle#
interface Ethernet0
 ip address 130.10.15.130 255.255.255.192
interface Serial0
 ip address 130.10.15.2 255.255.255.192
router rip
 network 130.10.15.0

Trinity#
interface Ethernet0
 ip address 130.10.15.67 255.255.255.192
interface Serial1
 ip address 130.10.15.1 255.255.255.192
router ospf 10
 redistribute rip metric 10 subnets
 network 130.10.15.0 0.0.0.255 area 0
router rip
```

**Example 6-19** *Configurations Using Distribute Lists (Continued)*

```

redistribute ospf 10 metric 2
passive-interface Ethernet0
network 130.10.15.0

```

---

```

Neo#
interface Ethernet0
ip address 130.10.15.68 255.255.255.192
router ospf 10
redistribute rip metric 10 subnets
network 130.10.15.0 0.0.0.255 area 0
router rip
redistribute ospf 10 metric 1
network 130.10.15.0

```

Notice that in the routing table of Example 6-20, Router Trinity has two routes to reach the 130.10.15.128 subnet: Ethernet 1 and Ethernet 0 (E0 is obviously the wrong path).

**Example 6-20** *Sample Routing Table with Erroneous Entry*

```

Trinity#show 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
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
Gateway of last resort is not set
130.10.15.0 255.255.255.192 is subnetted, 4 subnets
C 130.10.15.0 is directly connected, Ethernet1
C 130.10.15.64 is directly connected, Ethernet0
R 130.10.15.128 [120/1] via 130.10.15.68, 00:01:08, Ethernet0
[120/1] via 130.10.15.2, 00:00:11, Ethernet1
O 130.10.15.192 [110/20] via 130.10.15.68, 00:21:41, Ethernet0

```

This routing loop happened because Router Trinity gave that route entry to Router Neo via OSPF, and Router Neo gave it back via RIP because Router Neo did not learn about it via RIP. This example is a small scale of the routing loops that can occur because of an incorrect configuration. In large networks, this situation is even more pronounced and harder to isolate. This can also lead to route oscillation, where routes are quickly asserted and withdrawn, causing high CPU usage and overall poor router performance.

To fix the routing loop in this example, you could stop RIP routing updates from being sent on Router Neo's Ethernet 0 by making it a passive interface in RIP. This might not be suitable in case some routers in the OSPF Ethernet core are RIP-only routers. In this case, you could allow Router Trinity to send RIP on its Ethernet 0 interface; this way, Router Neo does not send it back on the wire because of split horizon. (This might not work on NBMA media if split horizon is off.) Split horizon does not allow updates to be sent back on the same interface that they were learned from (via the same protocol). Another good method is to apply distribute lists on Router Neo to deny subnets that are learned via OSPF from being put back into RIP on the Ethernet, as demonstrated in Example 6-21.

**Example 6-21** *Configuration Using Distribute Lists*

```

Neo#
interface Ethernet0
  ip address 130.10.15.68 255.255.255.192
router ospf 10
  redistribute rip metric 10 subnets
  network 130.10.15.0 0.0.0.255 area 0
router rip
  redistribute ospf 10 metric 1
  network 130.10.15.0
  distribute-list 1 out ospf 10

```

As previously discussed, the distribute list is applied outbound on the ASBR. Example 6-22 shows Trinity's resulting routing table.

**Example 6-22** *Corrected Routing Table After Application of Distribute List*

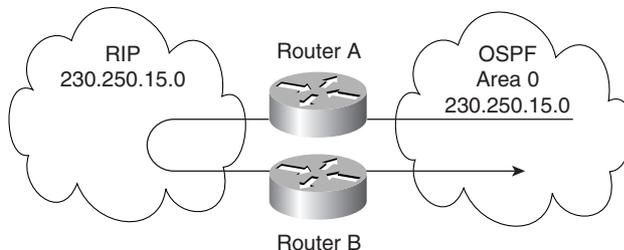
```

Trinity#show 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
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
Gateway of last resort is not set
130.10.15.0 255.255.255.192 is subnetted, 4 subnets
C 130.10.15.0 is directly connected, Ethernet1
C 130.10.15.64 is directly connected, Ethernet0
R 130.10.15.128 [120/1] via 130.10.15.2, 00:00:19, Ethernet1
O 130.10.15.192 [110/20] via 130.10.15.68, 00:21:41, Ethernet0

```

## Avoiding Redistribution Loops

Even though trying to avoid redistribution loops is a golden rule for route redistribution, these loops do occur. To summarize what is occurring, realize that Router A is distributing network 230.250.15.0 into the RIP network. Router B then sees this network advertised by RIP as a valid destination, so Router B tells the OSPF network that it can reach network 230.250.15.0. This results in a nasty routing loop, as illustrated in Figure 6-6.

**Figure 6-6** *Example of a Redistribution Loop*

Example 6-23 shows the initial configuration of Router A.

**Example 6-23** *Initial Configuration*

```
Router A (config)#
router ospf 10
redistribute rip subnets
network 230.250.0.0 0.0.255.255 area 0
```

As previously mentioned, this configuration would result in a routing loop caused by the uncontrolled redistribution. To prevent this, Example 6-24 shows modifications that must be made, this time to router B.

**Example 6-24** *Updated Router Configuration*

```
Router B (config)#
router ospf 10
redistribute rip subnets
network 230.250.0.0 0.0.255.255 area 0
distribute-list 1 out rip
access-list 1 deny 230.250.0.0
access-list 1 permit any
```

The **distribute-list** commands were designed specifically to filter routing updates based on an access list, in this case access-list 1.

But what did these commands do? Simply put, the **distribute-list 1** command is invoked under the OSPF process, and it applies access-list 1 to the outbound updates from OSPF and RIP. This configuration is applied if the redistribution source is the RIP network. In summary, this filter (access list) prevents the network 230.250.0.0 from being advertised back into the OSPF network. You should apply this solution to all routers advertising this network.

## Route Maps

Use route maps when you want detailed control over how routes are redistributed between routing processes. The destination routing protocol is the one that you specify with the **router** global configuration command. The source routing protocol is the one that you specify with the **redistribute** router configuration command. The route map is a method used to control the redistribution of routes between routing domains. The command syntax of a route map is as follows:

```
route-map map-tag [[permit | deny] | [sequence-number]]
```

When you are passing routes through a route map, a route map can have several parts. Any route that does not match at least one match clause relating to a **route-map** command is ignored; that is, the route is not advertised for outbound route maps and is not accepted for inbound route maps. To modify only some data, you must configure a second route map section with an explicit match specified or you can use a permit, depending upon your requirements.

When redistributing routes into OSPF, only routes that are not subnetted are redistributed if the **subnets** keyword is not specified.

## Configuration Example 2: RIP and OSPF

This case study is a classic for those making the transition from RIP to OSPF. It has become extremely popular within networking circles after it was published in the first edition of this book, and it has been updated.

This case study addresses the issue of integrating RIP networks with OSPF networks. Most OSPF networks also use RIP to communicate with hosts or to communicate with portions of the internetwork that do not use OSPF, such as older legacy areas or small business partners.

Cisco routers support both RIP and OSPF routing protocols and provide a way to exchange routing information between RIP and OSPF networks through redistribution. This case study provides examples of how to complete the following phases in redistributing information between RIP and OSPF networks:

- Configuring a RIP network
- Adding OSPF to the center of a RIP network
- Adding OSPF areas
- Setting up mutual redistribution

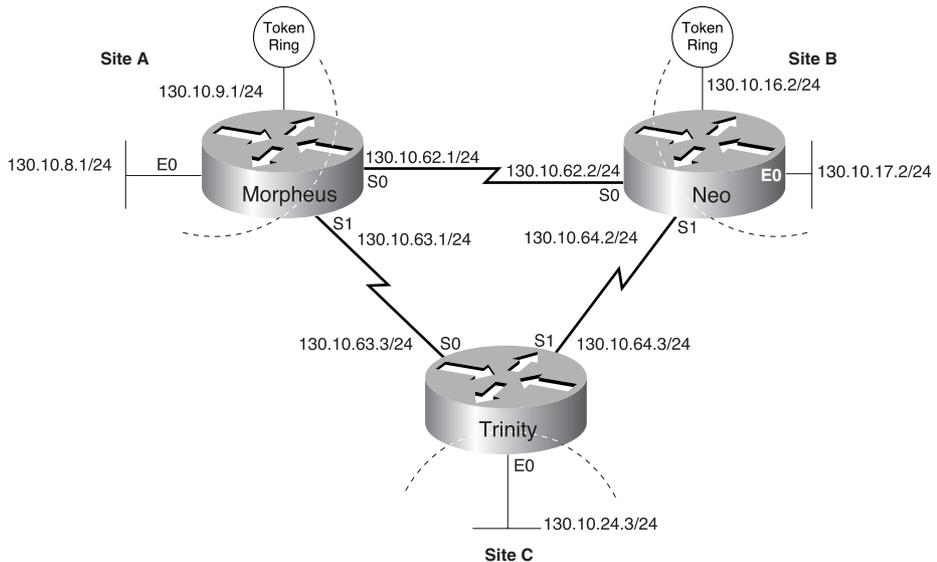
### Configuring the RIP Network

Figure 6-7 illustrates a RIP network. Three sites are connected with serial lines. The RIP network uses a Class B address and an 8-bit subnet mask. Each site has a contiguous set of network numbers assigned to it. The creators must have read the first edition of this book when designing the network because they clearly planned for future growth in the OSPF direction!

Table 6-2 lists the network address assignments for the RIP v2 network, including the network number, subnet range, and subnet masks. All interfaces indicate network 130.10.0.0; however, the specific address includes the subnet and subnet mask. For example, serial interface 0 on router C has an IP address of 130.10.63.3 with a subnet mask of 255.255.255.0.

**Table 6-3** *RIP Network Assignments*

Network Number	Subnets	Subnet Masks
130.10.0.0	Site A: 8–15	255.255.255.0
130.10.0.0	Site B: 16–23	255.255.255.0
130.10.0.0	Site C: 24–31	255.255.255.0
130.10.0.0	Core: 62–64	255.255.255.0

**Figure 6-7** Typical RIP Network

## Configuration File Examples

Example 6-25 shows the commands in the configuration file for Router Morpheus that determine the IP address for each interface and enable RIP on those interfaces.

### Example 6-25 *Morpheus Configuration*

```
interface serial 0
  ip address 130.10.62.1 255.255.255.0

interface serial 1
  ip address 130.10.63.1 255.255.255.0

interface ethernet 0
  ip address 130.10.8.1 255.255.255.0

interface tokenring 0
  ip address 130.10.9.1 255.255.255.0

router rip
  network 130.10.0.0
```

Example 6-26 shows the commands in the configuration file for Router Neo to determine the IP address for each interface and enable RIP on those interfaces.

**Example 6-26** *Neo Configuration*

```
interface serial 0
  ip address 130.10.62.2 255.255.255.0

interface serial 1
  ip address 130.10.64.2 255.255.255.0

interface ethernet 0
  ip address 130.10.17.2 255.255.255.0

interface tokenring 0
  ip address 130.10.16.2 255.255.255.0

router rip
  network 130.10.0.0
```

Example 6-27 shows the commands in the configuration file for router Trinity to determine the IP address for each interface and enable RIP on those interfaces.

**Example 6-27** *Trinity Configuration*

```
interface serial 0
  ip address 130.10.63.3 255.255.255.0

interface serial 1
  ip address 130.10.64.3 255.255.255.0

interface ethernet 0
  ip address 130.10.24.3 255.255.255.0

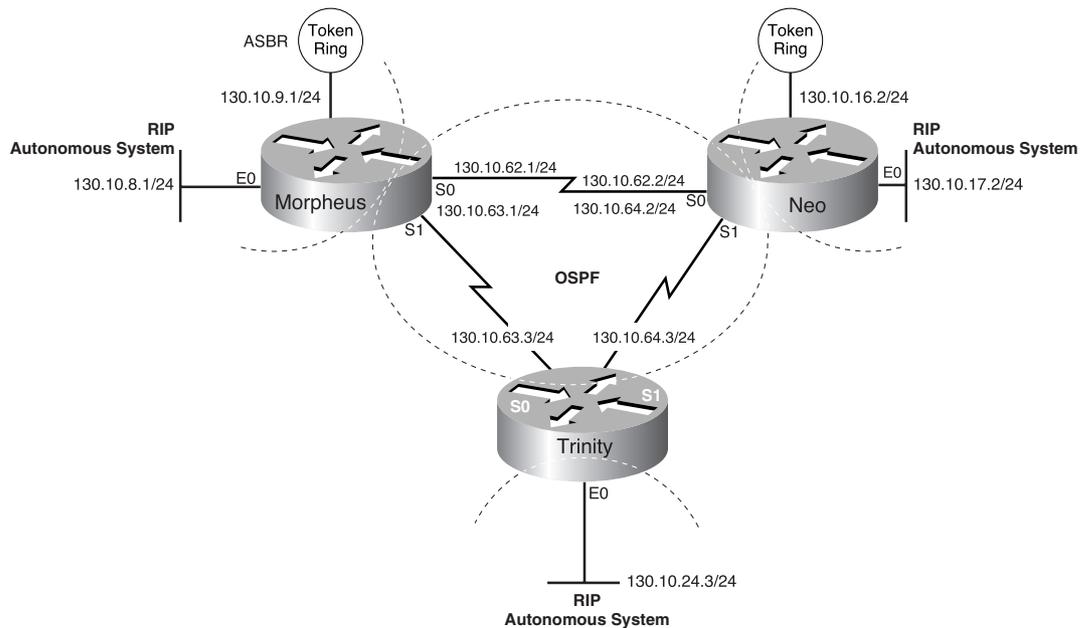
router rip
  network 130.10.0.0
```

## Adding OSPF to the Center of a RIP Network

A common first step in converting a RIP network to an OSPF network is to convert backbone routers into running both RIP and OSPF, while the remaining network edge devices run RIP. These backbone routers automatically become OSPF ASBRs (redistributing RIP into OSPF).

Each ASBR controls the flow of routing information between OSPF and RIP. In Figure 6-8, Routers Morpheus, Neo, and Trinity are configured as ASBRs when redistributing RIP into OSPF.

Figure 6-8 RIP with an OSPF Center



RIP does not need to run between the backbone routers; therefore, RIP is suppressed on all three routers with the following commands applied to each:

```
router rip
  passive-interface serial 0
  passive-interface serial 1
```

The RIP routes are redistributed into OSPF by all three routers with the following commands:

```
router ospf 109
  redistribute rip subnets
  network 130.10.0.0 255.255.0.0 area 0
```

The **subnets** keyword tells OSPF to redistribute all subnet routes. Without the **subnets** keyword, only networks that are not subnetted are redistributed by OSPF. Redistributed routes appear as external Type 2 routes in OSPF. For more detail on route types, see the *Cisco IOS Software Command Reference*.

Each RIP domain receives information about networks in other RIP domains and in the OSPF backbone area from the following commands, which redistribute OSPF routes into RIP:

```
router rip
  redistribute ospf 109 match internal external 1 external 2
  default-metric 10
```

The **redistribute** command uses the **ospf** keyword to specify that OSPF routes are redistributed into RIP. The **internal** keyword indicates the OSPF intra-area and inter-area routes: **external 1** is the external route Type 1, and **external 2** is the external route Type 2. Because the command in the example uses the default behavior, these keywords might not appear when you use the **show running-config** command. This is important because the routes being sent to the other OSPF routers contain E2 routes, the other RIP networks, so this is a crucial aspect to consider so that convergence occurs.

Because metrics for different protocols cannot be directly compared, you must specify the default metric to designate the cost of the redistributed route used in RIP updates. All routes that are redistributed use the default metric.

As illustrated in Figure 6-8, no paths are directly connecting the RIP clouds. However, in typical networks, these paths, or “back doors,” frequently exist, allowing the potential for routing loops to occur. You can use access lists to determine the routes that are advertised and accepted by each router.

For example, access list 11 in the configuration file for router Morpheus allows OSPF to redistribute information learned from RIP only for networks 130.10.8.0 through 130.10.15.0, as follows:

```
router ospf 109
 redistribute rip subnet
 distribute-list 11 out rip

access-list 11 permit 130.10.8.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.255
```

These commands prevent router Morpheus from advertising networks in other RIP domains onto the OSPF backbone, thereby preventing other boundary routers from using false information and forming a loop.

## Configuration File Examples

Compare the partial configuration options in Examples 6-25, 6-26, and 6-27 with the full configuration files in Examples 6-28, 6-29, and 6-30 for the routers in question.

### Example 6-28 Full Configuration for Router Morpheus

```
interface serial 0
 ip address 130.10.62.1 255.255.255.0
interface serial 1
 ip address 130.10.63.1 255.255.255.0
interface ethernet 0
 ip address 130.10.8.1 255.255.255.0
interface tokenring 0
 ip address 130.10.9.1 255.255.255.0
!
router rip
 default-metric 10
 network 130.10.0.0
```

**Example 6-28** *Full Configuration for Router Morpheus (Continued)*

```

passive-interface serial 0
passive-interface serial 1
redistribute ospf 109 match internal external 1 external 2
!
router ospf 109
network 130.10.62.0 0.0.0.255 area 0
network 130.10.63.0 0.0.0.255 area 0
redistribute rip subnets
distribute-list 11 out rip
!
access-list 11 permit 130.10.8.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.255

```

**Example 6-29** *Full Configuration for Router Neo*

```

interface serial 0
ip address 130.10.62.2 255.255.255.0
interface serial 1
ip address 130.10.64.2 255.255.255.0
interface ethernet 0
ip address 130.10.17.2 255.255.255.0
interface tokenring 0
ip address 130.10.16.2 255.255.255.0
!
router rip
default-metric 10
network 130.10.0.0
passive-interface serial 0
passive-interface serial 1
redistribute ospf 109 match internal external 1 external 2
!
router ospf 109
network 130.10.62.0 0.0.0.255 area 0
network 130.10.64.0 0.0.0.255 area 0
redistribute rip subnets
distribute-list 11 out rip
access-list 11 permit 130.10.16.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.255

```

**Example 6-30** *Full Configuration for Router Trinity*

```

interface serial 0
ip address 130.10.63.3 255.255.255.0
interface serial 1
ip address 130.10.64.3 255.255.255.0
interface ethernet 0
ip address 130.10.24.3 255.255.255.0
!
router rip
default-metric 10
!

```

*continues*

**Example 6-30** Full Configuration for Router Trinity (Continued)

```

network 130.10.0.0
passive-interface serial 0
passive-interface serial 1
redistribute ospf 109 match internal external 1 external 2
!
router ospf 109
network 130.10.63.0 0.0.0.255 area 0
network 130.10.64.0 0.0.0.255 area 0
redistribute rip subnets
distribute-list 11 out rip
access-list 11 permit 130.10.24.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.255

```

## Adding OSPF Areas

Figure 6-9 illustrates how each of the RIP clouds can be converted into an OSPF area. All three routers then become ABRs, which control network information distribution between OSPF areas and the OSPF backbone. Each router keeps a detailed record of the topology of its area and receives summarized information from the other ABRs on their respective areas.

Figure 6-9 also illustrates VLSM addressing. VLSM uses different size network masks in different parts of the network for the same network number. VLSM conserves address space by using a longer mask in portions of the network that have fewer hosts.

Table 6-3 lists the network address assignments for the network, including the network number, subnet range, and subnet masks. All interfaces indicate network 130.10.0.0.

**Table 6-4** OSPF Area Address Assignments

Network Number	Subnets	Subnet Masks
130.10.0.0	Area 0: 62–64	255.255.255.248
130.10.0.0	Area 1: 8–15	255.255.255.0
130.10.0.0	Area 2: 16–23	255.255.255.0
130.10.0.0	Area 3: 24–31	255.255.255.0

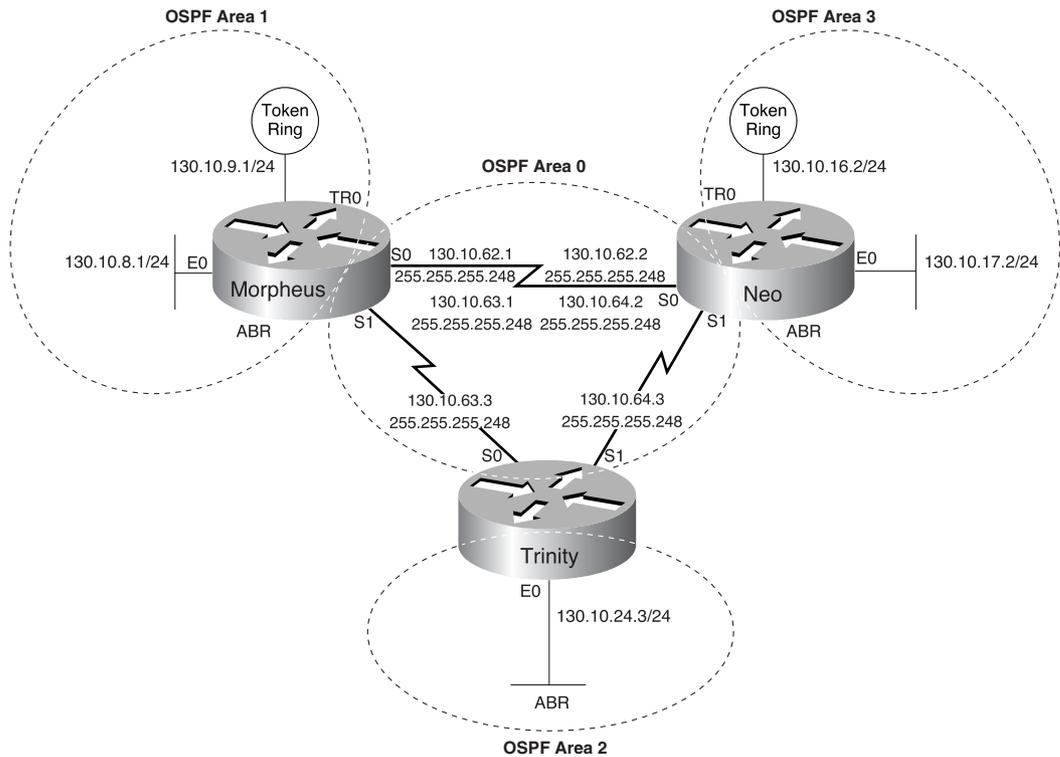
To conserve address space, a mask of 255.255.255.248 is used for all the serial lines in area 0. If an area contains a contiguous range of network numbers, an ABR uses the **range** keyword with the **area** command to summarize the routes that are injected into the backbone, as follows:

```

router ospf 109
network 130.10.8.0 0.0.7.255 area 1
area 1 range 130.10.8.0 255.255.248.0

```

**Figure 6-9** Configuring Route Summarization Between OSPF Areas



These commands allow router Morpheus to advertise one route, 130.10.8.0/255.255.248.0, which covers all subnets in area 1 into area 0. Without the **range** keyword in the **area** command, Morpheus would advertise each subnet individually, for example, one route for 130.10.8.0/255.255.255.0, one route for 130.10.9.0/255.255.255.0, and so on.

Because Morpheus no longer needs to redistribute RIP routes (all of them actually), the **router rip** command can now be removed from the configuration file.

### Configuration File Examples

The following configuration files in Examples 6-31, 6-32, and 6-33 reflect the presence of OSPF within the network and on the routers.

**Example 6-31** *Full Configuration for Morpheus*

```
interface serial 0
ip address 130.10.62.1 255.255.255.248

interface serial 1
ip address 130.10.63.1 255.255.255.248

interface ethernet 0
ip address 130.10.8.1 255.255.255.0
ip irdp

interface tokenring 0
ip address 130.10.9.1 255.255.255.0
ip irdp
router ospf 109
network 130.10.62.0 0.0.0.255 area 0
network 130.10.63.0 0.0.0.255 area 0
network 130.10.8.0 0.0.7.255 area 1
area 1 range 130.10.8.0 255.255.248.0
```

**Example 6-32** *Full Configuration for Router Neo*

```
interface serial 0
ip address 130.10.62.2 255.255.255.248

interface serial 1
ip address 130.10.64.2 255.255.255.248

interface ethernet 0
ip address 130.10.17.2 255.255.255.0
ip irdp

interface tokenring 0
ip address 130.10.16.2 255.255.255.0
ip irdp

router ospf 109
network 130.10.62.0 0.0.0.255 area 0
network 130.10.64.0 0.0.0.255 area 0
network 130.10.16.0 0.0.7.255 area 2
area 2 range 130.10.16.0 255.255.248.0
```

**Example 6-33** *Full Configuration for Router Trinity*

```
interface serial 0
ip address 130.10.63.2 255.255.255.248

interface serial 1
ip address 130.10.64.2 255.255.255.248

interface ethernet 0
ip address 130.10.24.3 255.255.255.0
```

**Example 6-33** *Full Configuration for Router Trinity (Continued)*

```

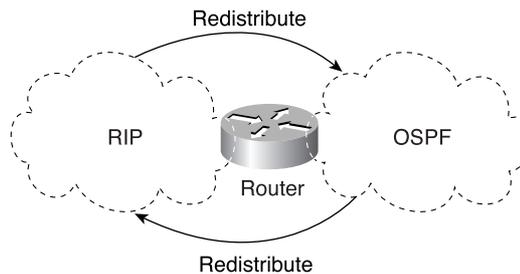
ip irdp

router ospf 109
network 130.10.63.0 0.0.0.255 area 0
network 130.10.64.0 0.0.0.255 area 0
network 130.10.24.0 0.0.0.255 area 3
area 3 range 130.10.24.0 255.255.248.0

```

**What If Mutual Redistribution Were Required?**

It is sometimes necessary to accommodate more complex network topologies, such as independent RIP and OSPF clouds, that must perform mutual redistribution. In this what-if scenario, it is important to prevent potential routing loops by filtering routes. The router in Figure 6-10 is running both OSPF and RIP.

**Figure 6-10** *Mutual Redistribution and OSPF Networks*

With the following commands, OSPF routes are redistributed into RIP. You must specify the default metric to designate the cost of the redistributed route in RIP updates. All routes that are redistributed into RIP have the following default metric:

```

! passive interface subcommand from previous example is left out for clarity!
router rip
default-metric 10
network 130.10.0.0
redistribute ospf 109

```

It is a good practice to strictly control which routes are advertised when redistribution is configured. In Example 6-34, a **distribute-list out** command causes RIP to ignore routes coming from the OSPF process that originated from the RIP domain.

**Example 6-34** *Distribute List to Ignore Routes*

```

router rip
distribute-list 10 out ospf 109
!
access-list 10 deny 130.10.8.0 0.0.7.255
access-list 10 permit 0.0.0.0 255.255.255.255
!Full Configuration for Router Morpheus

```

*continues*

**Example 6-34** *Distribute List to Ignore Routes (Continued)*

```

interface serial 0
ip add 130.10.62.1 255.255.255.0
!
interface serial 1
ip add 130.10.63.1 255.255.255.0
!
interface ethernet 0
ip add 130.10.8.1 255.255.255.0
!
interface tokenring 0
ip add 130.10.9.1 255.255.255.0
!
router rip
default-metric 10
network 130.10.0.0
passive-interface serial 0
passive-interface serial 1
redistribute ospf 109
distribute-list 10 out ospf 109
!
router ospf 109
network 130.10.62.0 0.0.0.255 area 0
network 130.10.63.0 0.0.0.255 area 0
redistribute rip subnets
distribute-list 11 out rip
!
access-list 10 deny 130.10.8.0 0.0.7.255
access-list 10 permit 0.0.0.0 255.255.255.255
access-list 11 permit 130.10.8.0 0.0.7.255
access-list 11 deny 0.0.0.0 255.255.255.255

```

Because it is common for OSPF and RIP to be used together, it is important to use the practices described here to provide functionality for both protocols on an internetwork. You can configure ASBRs that run both RIP and OSPF and redistribute RIP routes into the OSPF, and vice versa. You can also create OSPF areas using ABRs that provide route summarizations. Use VLSM to conserve address space.

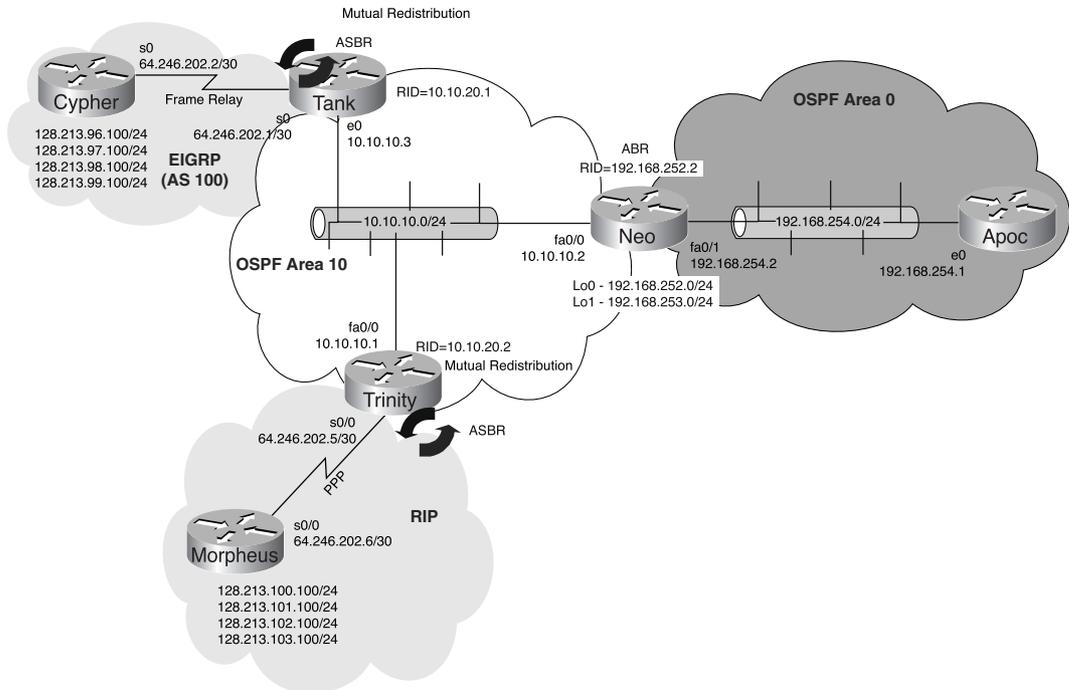
## Configuration Example 3: Redistributing Connected and Loopback Interfaces

This example looks at redistributing connected networks into OSPF, the results of this, and alternative ways of handling connected networks.

In cases where you must redistribute connected networks into OSPF instead of configuring OSPF to recognize them via the **network** command, all the connected networks show up as external routes in OSPF.

In this example, Router Neo is configured with a loopback interface, which is assigned the network 192.168.253.0/24. Figure 6-11 shows the overall network for this example. The configuration steps are described in the text that follows.

**Figure 6-11** Network Topology for Connected and Loopback Interface Redistribution



Example 6-35 shows how to assign a determined network to the loopback interface and then configure OSPF to redistribute connected networks.

**Example 6-35** Configuration to Redistribute Connected Routes

```
Neo(config)#int lo1
Neo(config-if)#ip address 192.168.253.1 255.255.255.0
Neo(config-if)#router ospf 100
Neo(config-router)#redistribute connected
```

The result of this configuration is the introduction of the network 192.168.253.0/24 into the routing table. Example 6-36 shows the results by viewing the routing table on Router Tank. Other methods can accomplish the same result (such as including changing the default network type on a loopback interface). These methods are much easier than redistributing connected networks because you might have connected interfaces as part of another routing domain.

**Example 6-36** *Redistribution Example*

```
Tank#show 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

Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 2 subnets
C       10.10.10.0 is directly connected, Ethernet0
C       10.10.20.0 is directly connected, Loopback0
    64.0.0.0/30 is subnetted, 1 subnets
C       64.246.202.0 is directly connected, Serial0
    128.213.0.0/24 is subnetted, 4 subnets
D       128.213.97.0 [90/2297856] via 64.246.202.2, 01:02:35, Serial0
D       128.213.96.0 [90/2297856] via 64.246.202.2, 01:02:35, Serial0
D       128.213.99.0 [90/2297856] via 64.246.202.2, 01:02:35, Serial0
D       128.213.98.0 [90/2297856] via 64.246.202.2, 01:02:35, Serial0
O IA 192.168.252.0/24 [110/11] via 10.10.10.2, 00:00:06, Ethernet0
O E2 192.168.253.0/24 [110/20] via 10.10.10.2, 00:00:07, Ethernet0
O IA 192.168.254.0/24 [110/11] via 10.10.10.2, 00:00:07, Ethernet0
```

The redistributed network is treated as an external network. Notice how the route is now present as an OSPF external Type 2 route, which is the default type of external route.

You can use another way to get this network into OSPF. This method is important to know if you are studying for certification or do not want to redistribute. To accomplish this, place the network assigned to the loopback interface into OSPF via the **network** command, as shown in Example 6-37.

**Example 6-37** *Advertising Without Redistributing*

```
Neo(config)#router ospf 100
Neo(config-router)#no redistribute connected
Neo(config-router)#network 192.168.253.0 0.0.0.255 area 0
Neo(config-router)#
```

This changes the OSPF route type for this network from external to inter-area (O IA in the routing table) from Router Tank's perspective because this router is area 10, as shown in Example 6-38.

**Example 6-38** *Changes to the OSPF Route Type*

```
Tank#show 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
```

**Example 6-38** *Changes to the OSPF Route Type (Continued)*

```

        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
        U - per-user static route, o - ODR

Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 2 subnets
C       10.10.10.0 is directly connected, Ethernet0
C       10.10.20.0 is directly connected, Loopback0
    64.0.0.0/30 is subnetted, 1 subnets
C       64.246.202.0 is directly connected, Serial0
    128.213.0.0/24 is subnetted, 4 subnets
D       128.213.97.0 [90/2297856] via 64.246.202.2, 01:04:02, Serial0
D       128.213.96.0 [90/2297856] via 64.246.202.2, 01:04:02, Serial0
D       128.213.99.0 [90/2297856] via 64.246.202.2, 01:04:02, Serial0
D       128.213.98.0 [90/2297856] via 64.246.202.2, 01:04:02, Serial0
O IA   192.168.252.0/24 [110/11] via 10.10.10.2, 00:00:30, Ethernet0
    192.168.253.0/32 is subnetted, 1 subnets
O IA   192.168.253.1 [110/11] via 10.10.10.2, 00:00:11, Ethernet0
O IA   192.168.254.0/24 [110/11] via 10.10.10.2, 00:00:30, Ethernet0

```

However, notice that the network is now represented as a /32 host entry in the routing table. This is not the desired result. What you want to see is a network entry (that is, /24). To correct this, use the OSPF **network type** command on the loopback interface to configure OSPF to recognize that interface as a different network type and thus alter its advertisement of the network, as demonstrated in Example 6-39.

**Example 6-39** *Setting the OSPF Network Type*

```

Neo(config-router)#int lo1
Neo(config-if)#ip ospf network ?
    broadcast          Specify OSPF broadcast multi-access network
    non-broadcast      Specify OSPF NBMA network
    point-to-multipoint Specify OSPF point-to-multipoint network
    point-to-point     Specify OSPF point-to-point network

Neo(config-if)#ip ospf network point-to-point
Neo(config-if)#

```

The result of this configuration can now be seen in the routing table of Router Tank in Example 6-40.

**Example 6-40** *How OSPF Network Type Affects Route Entries*

```

Tank#show 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

```

*continues*

**Example 6-40** *How OSPF Network Type Affects Route Entries (Continued)*

```

Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 2 subnets
C       10.10.10.0 is directly connected, Ethernet0
C       10.10.20.0 is directly connected, Loopback0
    64.0.0.0/30 is subnetted, 1 subnets
C       64.246.202.0 is directly connected, Serial0
    128.213.0.0/24 is subnetted, 4 subnets
D       128.213.97.0 [90/2297856] via 64.246.202.2, 01:05:38, Serial0
D       128.213.96.0 [90/2297856] via 64.246.202.2, 01:05:38, Serial0
D       128.213.99.0 [90/2297856] via 64.246.202.2, 01:05:38, Serial0
D       128.213.98.0 [90/2297856] via 64.246.202.2, 01:05:38, Serial0
O IA 192.168.252.0/24 [110/11] via 10.10.10.2, 00:02:06, Ethernet0
O IA 192.168.253.0/24 [110/11] via 10.10.10.2, 00:01:03, Ethernet0
O IA 192.168.254.0/24 [110/11] via 10.10.10.2, 00:02:07, Ethernet0
Tank#

```

That is perfect. The network in question now appears as an intra-area OSPF route and a /24. Example 6-41 shows the final configuration for this on Router Neo.

**Example 6-41** *OSPF Network Type Configuration Example*

```

!
interface Loopback1
 ip address 192.168.253.1 255.255.255.0
 no ip directed-broadcast
 ip ospf network point-to-point
!
router ospf 100
 network 10.10.10.0 0.0.0.255 area 10
 network 192.168.252.0 0.0.0.255 area 0
 network 192.168.253.0 0.0.0.255 area 0
 network 192.168.254.0 0.0.0.255 area 0

```

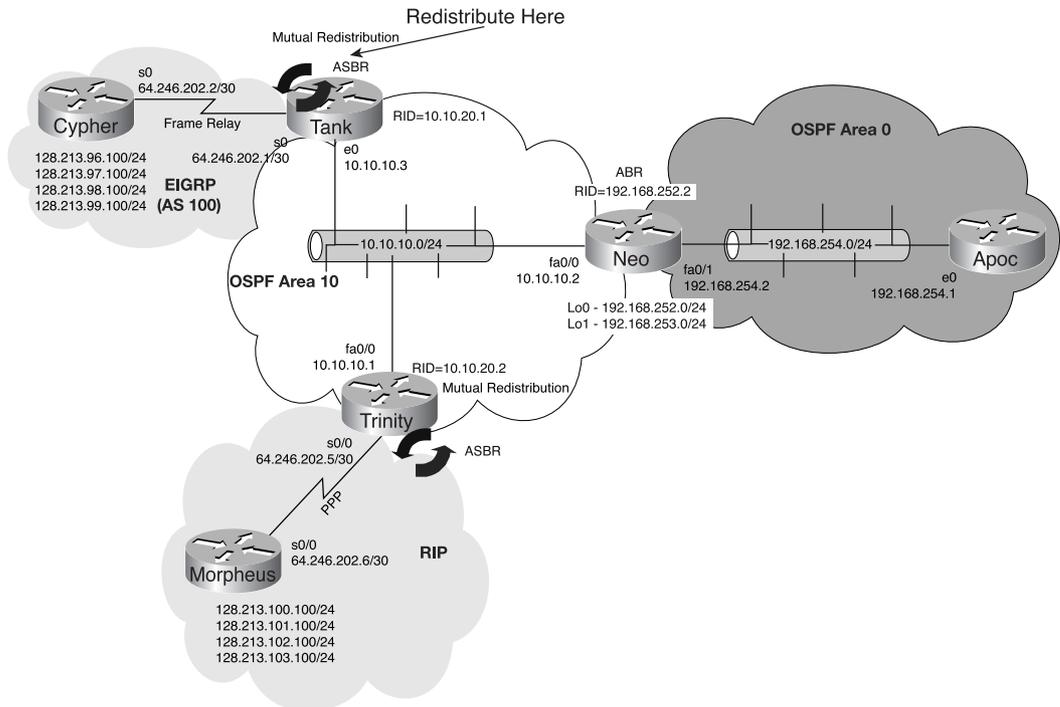
The next section discusses how to redistribute EIGRP into OSPF.

## Configuration Example 4: Redistributing OSPF and EIGRP

EIGRP is a popular routing protocol for those wishing to deploy a single-vendor network (and obviously that vendor is Cisco). Is that a smart move? That is a Layer 8—political discussion. Having options and flexibility does not mean that you only look at buying one vendor's product. I like Cisco equipment and Cisco's routing protocol EIGRP, but having both together puts too many constraints on the business of running a network. The use of EIGRP is solid, and it is useful to understand how to get routes from EIGRP into OSPF via redistribution; just remember that OSPF is number one. Figure 6-12 shows the sample network used for this configuration example.

**NOTE** For more information on EIGRP, consult the Cisco Press book *EIGRP Network Design Solutions* by Ivan Pepelnjak.

**Figure 6-12** Redistributing OSPF and EIGRP



In this network, Router Tank is connected to OSPF area 10 and EIGRP AS 100. The entire OSPF network needs to know about the following networks:

```
64.246.202.0/30
128.213.96.0/24
128.213.97.0/24
128.213.98.0/24
128.213.99.0/24
```

These networks are found on Router Tank and are part of the EIGRP routing domain. Because the objective is for the entire OSPF network to learn about these networks, the following examples monitor the routing table of Router Apoc to see when it learns of these routes.

### Before

At this time, Router Apoc has no idea that those routes are present; however, that will change when OSPF learns of them.

#### Example 6-42 Before OSPF Learns About the EIGRP Routes

```
Apoc>show 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

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O IA   10.10.10.0/24 [110/11] via 192.168.254.2, 00:23:12, Ethernet0
O IA   10.10.20.0/24 [110/12] via 192.168.254.2, 00:23:12, Ethernet0
O IA   10.10.20.1/32 [110/12] via 192.168.254.2, 00:23:12, Ethernet0
O      192.168.252.0/24 [110/11] via 192.168.254.2, 01:11:52, Ethernet0
O      192.168.253.0/24 [110/11] via 192.168.254.2, 01:11:52, Ethernet0
C      192.168.254.0/24 is directly connected, Ethernet0
Apoc>
```

### During

Router Tank, as the router with interfaces in both OSPF and EIGRP, is responsible for required to redistribution the EIGRP routes into OSPF. Example 6-43 shows the current configuration of Router Tank.

#### Example 6-43 Configuring OSPF to Redistribute EIGRP

```
!
router eigrp 100
 network 64.0.0.0
 no auto-summary
!
router ospf 100
 redistribute eigrp 100
 network 10.10.10.0 0.0.0.255 area 10
 network 10.10.20.0 0.0.0.255 area 10
!
```

As you prepare for redistribution, you need to understand the various methods of assigning metrics to the redistributed networks. In this case, you assign the metric via the **redistribute** command. Also, the **subnets** keyword is used because this is always a good design rule in today's world of many subnets! Example 6-44 shows the configuration tasks.

**Example 6-44** *Configuring the Metric and Using the subnets Keyword*

```
Tank#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Tank(config)#router ospf 100
Tank(config-router)#redistribute eigrp 100 metric 100 ?
metric      Metric for redistributed routes
metric-type OSPF/IS-IS exterior metric type for redistributed routes
route-map   Route map reference
subnets    Consider subnets for redistribution into OSPF
tag         Set tag for routes redistributed into OSPF
<cr>

Tank(config-router)#redistribute eigrp 100 metric 100 subnets
Tank(config-router)#
```

**After**

With the configuration completed on Router Tank, you would expect the routes to be sent as external routes via type LSAs through OSPF area 10 and then into area 0 to Router Apoc. Example 6-45 confirms this.

**Example 6-45** *Routing Table with EIGRP Routes Redistributed into OSPF*

```
Apoc>show 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

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O IA   10.10.10.0/24 [110/11] via 192.168.254.2, 00:01:39, Ethernet0
O IA   10.10.20.0/24 [110/12] via 192.168.254.2, 00:01:39, Ethernet0
O IA   10.10.20.1/32 [110/12] via 192.168.254.2, 00:01:39, Ethernet0
      64.0.0.0/30 is subnetted, 1 subnets
O E2   64.246.202.0 [110/100] via 192.168.254.2, 00:01:39, Ethernet0
      128.213.0.0/24 is subnetted, 4 subnets
O E2   128.213.97.0 [110/100] via 192.168.254.2, 00:01:39, Ethernet0
O E2   128.213.96.0 [110/100] via 192.168.254.2, 00:01:39, Ethernet0
O E2   128.213.99.0 [110/100] via 192.168.254.2, 00:01:39, Ethernet0
O E2   128.213.98.0 [110/100] via 192.168.254.2, 00:01:40, Ethernet0
O      192.168.252.0/24 [110/11] via 192.168.254.2, 01:40:59, Ethernet0
O      192.168.253.0/24 [110/11] via 192.168.254.2, 01:40:59, Ethernet0
C      192.168.254.0/24 is directly connected, Ethernet0
Apoc>
```

## OSPF and EIGRP Mutual Redistribution

Referring to the network in Figure 6-12, you can see in Example 6-46 that Router Cypher has no idea if any of the OSPF routes are possible via its connection to Router Tank.

### Before

**Example 6-46** *Routing Table Before Mutual Redistribution*

```
Cypher#show 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

Gateway of last resort is not set

 64.0.0.0/30 is subnetted, 1 subnets
C       64.246.202.0 is directly connected, Serial0
 128.213.0.0/24 is subnetted, 4 subnets
C       128.213.97.0 is directly connected, Loopback1
C       128.213.96.0 is directly connected, Loopback0
C       128.213.99.0 is directly connected, Loopback3
C       128.213.98.0 is directly connected, Loopback2
Cypher#
```

### During

Example 6-47 shows how to configure redistribution on Router Tank. When this is combined with currently configured redistribution, you have achieved a state of mutual redistribution.

**Example 6-47** *Configuring Mutual Redistribution*

```
Tank(config)#router eigrp 100

Tank(config-router)#redistribute ospf 100 metric ?
 <1-4294967295> Bandwidth metric in Kbits per second

Tank(config-router)#redistribute ospf 100 metric 1500 ?
 <0-4294967295> IGRP delay metric, in 10 microsecond units

Tank(config-router)#redistribute ospf 100 metric 1500 10 ?
 <0-255> IGRP reliability metric where 255 is 100% reliable

Tank(config-router)#redistribute ospf 100 metric 1500 10 255 ?
 <1-255> IGRP Effective bandwidth metric (Loading) where 255 is 100% loaded

Tank(config-router)#redistribute ospf 100 metric 1500 10 255 1 ?
 <1-4294967295> IGRP MTU of the path

Tank(config-router)#redistribute ospf 100 metric 1500 10 255 1 1500
Tank(config-router)#
```

Having configured the metrics of the OSPF routes to equal the defaults of a serial line, you are ready to check if Router Cypher now knows about the OSPF routes.

### After

The configuration was successful, and you can see in Example 6-48 that the routes are appearing as external EIGRP routes.

#### Example 6-48 *Verifying That Redistribution Occurred*

```
Cypher#show 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

Gateway of last resort is not set

  10.0.0.0/24 is subnetted, 2 subnets
D EX   10.10.10.0 [170/2221056] via 64.246.202.1, 00:00:10, Serial0
D EX   10.10.20.0 [170/2221056] via 64.246.202.1, 00:00:11, Serial0
  64.0.0.0/30 is subnetted, 1 subnets
C      64.246.202.0 is directly connected, Serial0
 128.213.0.0/24 is subnetted, 4 subnets
C      128.213.97.0 is directly connected, Loopback1
C      128.213.96.0 is directly connected, Loopback0
C      128.213.99.0 is directly connected, Loopback3
C      128.213.98.0 is directly connected, Loopback2
D EX 192.168.252.0/24 [170/2221056] via 64.246.202.1, 00:00:11, Serial0
D EX 192.168.253.0/24 [170/2221056] via 64.246.202.1, 00:00:11, Serial0
D EX 192.168.254.0/24 [170/2221056] via 64.246.202.1, 00:00:11, Serial0
Cypher#
```

So far in this example, you have not encountered a routing loop because the networks used are all different and because there are not many routes. However, in practice, such a problem is more likely to occur, and you should always use route maps to protect against routing loops. The next section discusses how to configure the route maps.

## Using Route Maps to Protect Against Routing Loops

This section shows you the configuration steps used to deploy route maps to stop routing loops from occurring.

### Before

At first, you have the same basic redistribution configuration as shown in Example 6-49. You build the route map from this base configuration.

**Example 6-49** *Beginning Basic Redistribution Configuration*

```

!
router eigrp 100
 redistribute ospf 100 metric 1500 10 255 1 1500
 network 64.0.0.0
 no auto-summary
!
router ospf 100
 redistribute eigrp 100 metric 100 subnets
 network 10.10.10.0 0.0.0.255 area 10
 network 10.10.20.0 0.0.0.255 area 10
!

```

**During**

To configure the route map to protect the networks from route feedback and thus the creation of a routing loop, take the following steps:

- Step 1** List all the networks to be permitted through the use of access lists in both directions.
- Step 2** Create the route maps.
- Step 3** Assign the route map to the redistribution action for each protocol.

For Step 1, in Example 6-50, **access-list 66** is used to identify the routes allowed from OSPF into EIGRP.

**Example 6-50** *Using Access Lists to Identify the Routes to Redistribute from OSPF into EIGRP*

```

Tank(config)#access-list 66 permit 128.213.100.0 0.0.3.255
Tank(config)#access-list 66 permit 64.246.202.4 0.0.0.3
Tank(config)#access-list 66 permit 10.10.10.0 0.0.0.255
Tank(config)#access-list 66 permit 10.10.20.0 0.0.0.255
Tank(config)#access-list 66 permit 192.168.254.0 0.0.0.255
Tank(config)#access-list 66 permit 192.168.252.0 0.0.0.255
Tank(config)#access-list 66 permit 192.168.253.0 0.0.0.255

```

In Example 6-51, **access-list 97** identifies the routes allowed from EIGRP into OSPF. Recall the implicit deny all at the end of every access list.

**Example 6-51** *Using Access Lists to Identify the Routes to Redistribute from OSPF into EIGRP*

```

Tank(config)#access-list 97 permit 128.213.96.0 0.0.3.255
Tank(config)#access-list 97 permit 64.246.202.0 0.0.0.3

```

For Step 2 (creating the route maps), give the route maps names that describe their function and then map the right access list to the route map, as demonstrated in Example 6-52.

**Example 6-52** *Assigning the Access Lists to the Route Map*

```
Tank(config)#route-map OSPF-to-EIGRP permit 10
Tank(config-route-map)#match ip address 66

Tank(config)#route-map EIGRP-to-OSPF permit 10
Tank(config-route-map)#match ip address 97
```

For Step 3 (assigning the route map to the redistribution action for each protocol), in the case of Example 6-53, the route maps were created to allow only the identified routes into the other protocol upon redistribution.

**Example 6-53** *Applying the Route Map to the redistribution Command*

```
Tank(config)#router ospf 100
Tank(config-router)#redistribute eigrp 100 route-map EIGRP-to-OSPF

Tank(config-router)#router eigrp 100
Tank(config-router)#redistribute ospf 100 route-map OSPF-to-EIGRP
```

**After**

Example 6-54 shows the final configuration of the route maps. Always check your routing table after a route map is applied to ensure that it gave you the desired results.

**Example 6-54** *Final Configuration*

```
!
router eigrp 100
 redistribute ospf 100 metric 1500 10 255 1 1500 route-map OSPF-to-EIGRP
 network 64.0.0.0
 no auto-summary
!
router ospf 100
 redistribute eigrp 100 metric 100 subnets route-map EIGRP-to-OSPF
 network 10.10.10.0 0.0.0.255 area 10
 network 10.10.20.0 0.0.0.255 area 10
!
no ip classless
access-list 66 permit 128.213.100.0 0.0.3.255
access-list 66 permit 64.246.202.4 0.0.0.3
access-list 66 permit 10.10.10.0 0.0.0.255
access-list 66 permit 10.10.20.0 0.0.0.255
access-list 66 permit 192.168.254.0 0.0.0.255
access-list 66 permit 192.168.252.0 0.0.0.255
access-list 66 permit 192.168.253.0 0.0.0.255
access-list 97 permit 128.213.96.0 0.0.3.255
access-list 97 permit 64.246.202.0 0.0.0.3
route-map OSPF-to-EIGRP permit 10
 match ip address 66
!
route-map OSPF-to-EIGRP deny 65535
!
```

*continues*

**Example 6-54** *Final Configuration (Continued)*

```
route-map EIGRP-to-OSPF permit 10
  match ip address 97
!
route-map EIGRP-to-OSPF deny 65535
!
```

Although not required, creating route maps in OSPF like this is strongly recommended, and doing so is considered best practice and appropriate whenever possible.

## Using Route Tagging to Protect Against Routing Loops

This section shows you the configuration steps used to identify routes via a route tag during mutual redistribution to prevent loops.

### Before

At first, you have the same basic redistribution configuration that was used in the previous example. From the base configuration in Example 6-55, you build the route tag configuration.

**Example 6-55** *Basic Redistribution Configuration*

```
!
router eigrp 100
  redistribute ospf 100 metric 1500 10 255 1 1500
  network 64.0.0.0
  no auto-summary
!
router ospf 100
  redistribute eigrp 100 metric 100 subnets
  network 10.10.10.0 0.0.0.255 area 10
  network 10.10.20.0 0.0.0.255 area 10
!
```

### During

During the steps to configure and assign a route tag, you are first going to create a route map that tags each route. One route map is created for EIGRP and another for OSPF. You then permit the correct routes to be redistributed and deny the others, as shown in the configuration steps that follow.

You first create the route maps for OSPF; these route maps identify the routes to be denied when they are redistributed. The route maps become part of the **redistribution** commands, so when you say “deny routes that have a route tag of 3,” the routes will have already been redistributed and thus tagged through the other route map. This provides protection of OSPF. The same concept applies in reverse as you set the route tag of 4 because it is denied as part of the EIGRP redistribution route map. Example 6-56 shows the configuration tasks necessary to configure the router to use tags.

**Example 6-56** *Assigning Route Tags to Route Maps*

```
Tank(config)#route-map DEN3-PERM4 deny 10
Tank(config-route-map)#match tag 3
Tank(config-route-map)#route-map DEN3-PERM4 permit 20
Tank(config-route-map)#set tag 4

Tank(config-route-map)#route-map DEN4-PERM3 den 10
Tank(config-route-map)#match tag 4
Tank(config-route-map)#route-map DEN4-PERM3 perm 20
Tank(config-route-map)#set tag 3
```

Now that the route maps and tags are assigned and the permit/deny statements are in sync with the primary goal, it is time to add the route maps to the **redistribution** statements in OSPF and EIGRP, as shown in Example 6-57.

**Example 6-57** *Configuring the Route Maps to Work with Redistribution*

```
Tank(config-route-map)#router ospf 100
Tank(config-router)#redistribute eigrp 100 route-map DEN3-PERM4
Tank(config-router)#router eigrp 100
Tank(config-router)#redi
Tank(config-router)#redistribute ospf 100 rout
Tank(config-router)#redistribute ospf 100 route-map DEN4-PERM3
```

Example 6-58 shows the final configurations of the routers from this section.

**After****Example 6-58** *Final Route Tag Configurations*

```
router eigrp 100
 redistribute ospf 100 metric 1500 10 255 1 1500 route-map DEN4-PERM3
 network 64.0.0.0
 no auto-summary
 !
router ospf 100
 redistribute eigrp 100 metric 100 subnets route-map DEN3-PERM4
 network 10.10.10.0 0.0.0.255 area 10
 network 10.10.20.0 0.0.0.255 area 10
 !
no ip classless
access-list 66 permit 128.213.100.0 0.0.3.255
access-list 66 permit 64.246.202.4 0.0.0.3
access-list 66 permit 10.10.10.0 0.0.0.255
access-list 66 permit 192.168.254.0 0.0.0.255
access-list 66 permit 192.168.252.0 0.0.0.255
access-list 66 permit 192.168.253.0 0.0.0.255
access-list 97 permit 128.213.96.0 0.0.3.255
access-list 97 permit 64.246.202.0 0.0.0.3
 !
route-map DEN3-PERM4 deny 10
 match tag 3
```

*continues*

**Example 6-58** *Final Route Tag Configurations (Continued)*

```

!
route-map DEN3-PERM4 permit 20
  set tag 4
!
route-map DEN4-PERM3 deny 10
  match tag 4
!
route-map DEN4-PERM3 permit 20
  set tag 3
!

```

One of the useful features of tagging routes like this is that after you have identified them, they are easy to manipulate based on the presence of that tag. The next configuration example examines how the tagging of a route is accomplished.

## Configuration Example 5: Redistributing OSPF and RIP and Tagging Routes

In your network, you have connected Router Trinity to OSPF area 10 and a RIP network as well. The entire OSPF network needs to know about the following networks:

```

64.246.202.4/30
128.213.100.0/24
128.213.101.0/24
128.213.102.0/24
128.213.103.0/24

```

These networks are found on Router Trinity and are part of the RIP routing domain. Because the objective is for the entire OSPF network to learn about them, you are going to be monitoring the routing table of Router Apoc to see when it learns of these routes.

### Before

As you would expect, in the routing table in Example 6-59, Router Apoc has no knowledge of any of the routes in the RIP network attached to Router Trinity.

**Example 6-59** *Configuration Before Redistributing RIP with Route Tags*

```

Apoc>show 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

Gateway of last resort is not set

```

**Example 6-59** *Configuration Before Redistributing RIP with Route Tags (Continued)*

```

10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O IA  10.10.10.0/24 [110/11] via 192.168.254.2, 00:01:39, Ethernet0
O IA  10.10.20.0/24 [110/12] via 192.168.254.2, 00:01:39, Ethernet0
O IA  10.10.20.1/32 [110/12] via 192.168.254.2, 00:01:39, Ethernet0
64.0.0.0/30 is subnetted, 1 subnets
O E2  64.246.202.0 [110/100] via 192.168.254.2, 00:01:39, Ethernet0
128.213.0.0/24 is subnetted, 4 subnets
O E2  128.213.97.0 [110/100] via 192.168.254.2, 00:01:39, Ethernet0
O E2  128.213.96.0 [110/100] via 192.168.254.2, 00:01:39, Ethernet0
O E2  128.213.99.0 [110/100] via 192.168.254.2, 00:01:39, Ethernet0
O E2  128.213.98.0 [110/100] via 192.168.254.2, 00:01:40, Ethernet0
O     192.168.252.0/24 [110/11] via 192.168.254.2, 01:40:59, Ethernet0
O     192.168.253.0/24 [110/11] via 192.168.254.2, 01:40:59, Ethernet0
C     192.168.254.0/24 is directly connected, Ethernet0
Apoc>

```

**During**

In Example 6-60, you alter the metric to a value of 200 to allow a more visual distinction between the routes from RIP and those from EIGRP.

**Example 6-60** *Altering the Metric for RIP Routes*

```

Trinity(config)#router ospf 100
Trinity(config-router)#redistribute rip metric 200 subnets
Trinity(config)#

```

**After**

With the redistribution of RIP into OSPF complete, review the routing table on Router Apoc in Example 6-61. You would expect to see the routes from the RIP domain (as identified in Example 6-60) with a metric of 200.

**Example 6-61** *Verifying Redistribution*

```

Apoc>show 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

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O IA  10.10.10.0/24 [110/11] via 192.168.254.2, 00:00:27, Ethernet0
O IA  10.10.20.0/24 [110/12] via 192.168.254.2, 00:00:27, Ethernet0
O IA  10.10.20.1/32 [110/12] via 192.168.254.2, 00:00:27, Ethernet0
      64.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O E2  64.246.202.4/30 [110/200] via 192.168.254.2, 00:00:27, Ethernet0

```

*continues*

**Example 6-61** *Verifying Redistribution (Continued)*

```

O E2 64.246.202.6/32 [110/200] via 192.168.254.2, 00:00:27, Ethernet0
O E2 64.246.202.0/30 [110/100] via 192.168.254.2, 00:00:27, Ethernet0
    128.213.0.0/24 is subnetted, 8 subnets
O E2 128.213.101.0 [110/200] via 192.168.254.2, 00:00:27, Ethernet0
O E2 128.213.100.0 [110/200] via 192.168.254.2, 00:00:28, Ethernet0
O E2 128.213.103.0 [110/200] via 192.168.254.2, 00:00:28, Ethernet0
O E2 128.213.102.0 [110/200] via 192.168.254.2, 00:00:28, Ethernet0
O E2 128.213.97.0 [110/100] via 192.168.254.2, 00:00:28, Ethernet0
O E2 128.213.96.0 [110/100] via 192.168.254.2, 00:00:29, Ethernet0
O E2 128.213.99.0 [110/100] via 192.168.254.2, 00:00:29, Ethernet0
O E2 128.213.98.0 [110/100] via 192.168.254.2, 00:00:30, Ethernet0
O 192.168.252.0/24 [110/11] via 192.168.254.2, 02:08:41, Ethernet0
O 192.168.253.0/24 [110/11] via 192.168.254.2, 02:08:41, Ethernet0
C 192.168.254.0/24 is directly connected, Ethernet0
Apoc>

```

The redistribution was successful. Now you can move on to activating redistribution in the other direction (OSPF to RIP) to establish a running mutual redistribution scenario.

## OSPF and RIP Mutual Redistribution

The next step is to make the necessary configuration to get the OSPF routes into RIP, as shown in Example 6-62.

**Example 6-62** *Beginning RIP Configuration with Basic Redistribution*

```

!
router rip
version 2
redistribute ospf 100 metric 5
network 64.0.0.0
!

```

Looking at Router Morpheus's routing table in Example 6-63, you can see that all the routes are available and that they have a metric of 5. You would expect this according to the redistribution configuration.

**Example 6-63** *Verifying the Operation*

```

Morpheus#show 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

Gateway of last resort is not set

```

**Example 6-63** *Verifying the Operation (Continued)*

```

        64.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       64.246.202.4/30 is directly connected, Serial0/0
C       64.246.202.5/32 is directly connected, Serial0/0
R       64.246.202.0/30 [120/5] via 64.246.202.5, 00:00:07, Serial0/0
        128.213.0.0/16 is variably subnetted, 5 subnets, 2 masks
C       128.213.101.0/24 is directly connected, Loopback1
C       128.213.100.0/24 is directly connected, Loopback0
C       128.213.103.0/24 is directly connected, Loopback3
C       128.213.102.0/24 is directly connected, Loopback2
R       128.213.0.0/16 [120/5] via 64.246.202.5, 00:00:07, Serial0/0
R       10.0.0.0/8 [120/5] via 64.246.202.5, 00:00:07, Serial0/0
R       192.168.254.0/24 [120/5] via 64.246.202.5, 00:00:07, Serial0/0
R       192.168.253.0/24 [120/5] via 64.246.202.5, 00:00:07, Serial0/0
R       192.168.252.0/24 [120/5] via 64.246.202.5, 00:00:07, Serial0/0
Morpheus#

```

**Redistributing into OSPF with Route Tagging**

There are many reasons to tag routes, some of which have already been covered. Some others reflect the desire of network engineers to manipulate routes based on tags. This is a nice way to quickly identify which routes you need. Consider that you want to know which routes are from RIP, say on Router Apoc. You could create an access list and a route map, but what if you tagged all the routes during the redistribution on Router Trinity, thus allowing a more flexible and faster identification of routes? This is possible with route tagging, and applying a route tag while redistributing is shown in Example 6-64.

**Before****Example 6-64** *Basic OSPF Redistribution Configuration*

```

!
router ospf 100
 log-adjacency-changes
 redistribute rip metric 200 subnets
 network 10.10.10.0 0.0.0.255 area 10
 network 10.10.20.0 0.0.0.255 area 10
!

```

In OSPF, you can see the route tags in the OSPF link-state database. Therefore, by checking the database before you add the tag, the database has all the tag values as 0 (default), as expected for the RIP routes. Now tag the routes, release them, and see what happens in Example 6-65.

**Example 6-65** *No Route Tags in the OSPF Link-State Database*

```

Apoc#show ip ospf database

      OSPF Router with ID (192.168.254.1) (Process ID 100)

          Router Link States (Area 0)

Link ID        ADV Router    Age          Seq#          Checksum Link count
192.168.252.2  192.168.252.2  735         0x80000011  0xD7B4   3
192.168.254.1  192.168.254.1  345         0x8000000A  0x87F2   1

          Net Link States (Area 0)

Link ID        ADV Router    Age          Seq#          Checksum
192.168.254.2  192.168.252.2  1226        0x80000006  0x8C04

          Summary Net Link States (Area 0)

Link ID        ADV Router    Age          Seq#          Checksum
10.10.10.0     192.168.252.2  479         0x8000000B  0x1398
10.10.20.0     192.168.252.2  479         0x80000009  0xB2EF
10.10.20.1     192.168.252.2  479         0x80000009  0xA8F8

          Summary ASB Link States (Area 0)

Link ID        ADV Router    Age          Seq#          Checksum
10.10.20.1     192.168.252.2  983         0x80000003  0x9C0B
10.10.20.2     192.168.252.2  1228        0x80000002  0x9413

          Type-5 AS External Link States

Link ID        ADV Router    Age          Seq#          Checksum Tag
64.246.202.0  10.10.20.1   1242        0x80000003  0x2AFB   0
64.246.202.4  10.10.20.2   1201        0x80000003  0x6ED    0
64.246.202.6  10.10.20.2   1201        0x80000003  0x4EA    0
128.213.96.0  10.10.20.1   1242        0x80000003  0x1955   0
128.213.97.0  10.10.20.1   1242        0x80000003  0xE5F    0
128.213.98.0  10.10.20.1   1242        0x80000003  0x369    0
128.213.99.0  10.10.20.1   1242        0x80000003  0xF773   0
128.213.100.0 10.10.20.2   1201        0x80000003  0xF04B   0
128.213.101.0 10.10.20.2   1201        0x80000003  0xE555   0
128.213.102.0 10.10.20.2   1201        0x80000003  0xDA5F   0
128.213.103.0 10.10.20.2   1201        0x80000003  0xCF69   0
Apoc#

```

**During**

Through the use of an additional keyword, **tag 200**, in the redistribution statement, you can easily create a route tag with a value of 200 and assign it to all routes coming into OSPF from RIP, as shown in Example 6-66.

**Example 6-66** *Configuring Redistribution via Route Tagging*

```

Trinity(config)#router ospf 100
Trinity(config-router)#redistribute rip metric 200 subnets ?
    metric      Metric for redistributed routes
    metric-type OSPF/IS-IS exterior metric type for redistributed routes
    route-map   Route map reference
    subnets    Consider subnets for redistribution into OSPF
    tag         Set tag for routes redistributed into OSPF
    <cr>

Trinity(config-router)#redistribute rip metric 200 subnets tag ?
    <0-4294967295> 32-bit tag value

Trinity(config-router)#redistribute rip metric 200 subnets tag 200
Trinity(config-router)#

```

**After**

In Example 6-67, you determine whether the external RIP routes are now tagged with a value of 200 per the configuration.

**Example 6-67** *Verifying the Presence of the Tagged Routes in the OSPF Link-State Database*

```

Apoc#show ip ospf database

      OSPF Router with ID (192.168.254.1) (Process ID 100)

<<<Some output omitted for brevity>>>

      Type-5 AS External Link States

Link ID        ADV Router    Age         Seq#          Checksum Tag
-----
64.246.202.0   10.10.20.1   1242       0x80000003   0x2AFB  0
64.246.202.4   10.10.20.2   1201       0x80000003   0x6ED   200
64.246.202.6   10.10.20.2   1201       0x80000003   0x4EA   200
128.213.96.0   10.10.20.1   1242       0x80000003   0x1955  0
128.213.97.0   10.10.20.1   1242       0x80000003   0xE5F   0
128.213.98.0   10.10.20.1   1242       0x80000003   0x369   0
128.213.99.0   10.10.20.1   1242       0x80000003   0xF773  0
128.213.100.0  10.10.20.2   1201       0x80000003   0xF04B  200
128.213.101.0  10.10.20.2   1201       0x80000003   0xE555  200
128.213.102.0  10.10.20.2   1201       0x80000003   0xDA5F  200
128.213.103.0  10.10.20.2   1201       0x80000003   0xCF69  200
Apoc#

```

Having the router display the OSPF link-state database can result in some very long output, and at times you might just want to check the presence of a tag on a single route. You can accomplish this using the **show ip route network number** command, as shown in Example 6-68.

**Example 6-68** *Viewing the Route Tag*

```

Apoc#show ip route 64.246.202.4
Routing entry for 64.246.202.4/30
  Known via "ospf 100", distance 110, metric 200
  Tag 200, type extern 2, forward metric 11
  Redistributing via ospf 100
  Last update from 192.168.254.2 on Ethernet0, 00:07:53 ago
  Routing Descriptor Blocks:
    * 192.168.254.2, from 10.10.20.2, 00:07:53 ago, via Ethernet0
      Route metric is 200, traffic share count is 1

Apoc#

```

With the presence of the routes tagged with a value of 200, you can now more easily identify these external RIP routes at a later time. The next section looks at other methods of controlling the redistribution of routes within OSPF.

## Configuration Example 6: Controlling Redistribution

To this point, the concepts of redistribution have been presented and examples were given to show how to make redistribution operate effectively. The following configuration example was placed later in the chapter for a specific reason. This example is a good review of the routing and redistribution concepts that were covered up to this point. By placing the concepts together, you can see some interesting OSPF effects. The following sections offer a review of the concepts previously presented.

### Altering Link Cost

Recall that OSPF calculates its cost (metric) to a destination based on the bandwidth of the link(s) to that destination. Therefore, to influence OSPF's routing decisions, you can either change the bandwidth on the interface, which in turn affects the cost of the link, or you can directly change the OSPF cost of the interface. You can apply the following commands on a per-interface basis:

- Router(config-if)**ip ospf cost** *1-65355*—Allows you to configure the cost of an interface in OSPF, thus overriding the process of OSPF calculating interface cost. This command is used only by OSPF. It does not affect traffic on the link, but it does affect route calculation and selection.
- Router(config-if)**bandwidth** *1-10000000*—Allows you to configure the bandwidth of an interface in kilobits per second. This command is used by routing protocols as the basis to calculate the cost for the interface. Many people forget to set this in their routers, so be careful. Changing this value does not affect traffic on the link, but it does affect route calculation and selection.
- **auto reference bandwidth**—Allows you to easily alter the link cost.

Therefore, you have a couple of ways to alter the calculation of the OSPF cost (metric) and thereby affect how OSPF calculates its route selection.

## Altering Routes

OSPF also allows the altering of routes, as discussed in Chapter 5. Specifically, OSPF also supports the direct changing of the administrative distance associated with OSPF routes, through the use of the **distance** command. OSPF can also use the **passive** interface command to prevent Hello packets and LSAs from being sent on the specified link. The list that follows shows the general syntax and descriptions for these commands that allow you to alter routes:

- Router(config-router)**distance ospf** *1-255*—Use this command to change the administrative distance of all OSPF routes from the default of 110 to a value you define. Alternatively, a more granular control is possible, as shown in the next command.
- Router(config-router)**distance ospf** {[**intra-area** *1-255*] [**inter-area** *1-255*] [**external** *1-255*]}—Use this command to change the administrative distance of specific types of OSPF routes. OSPF uses three different routes: intra-area, inter-area, and external. Routes within an area are intra-area, routes from another area are inter-area, and routes injected via redistribution are external. This command allows you to change the default administrative distance of 110 to the designated value.

---

**CAUTION** Use the **distance ospf** command with care because it can easily cause routing loops.

---

- Router(config-router)**default-metric cost** *1-4294967295*—Use this command to set the default cost of all routes redistributed into OSPF. You should always supply a default metric whenever redistributing into OSPF. Recall that a metric must always be assigned for redistribution to work.
- Router(config-router)**passive-interface** *interface\_name*—Use this command to prevent the sending of OSPF Hello packets and LSAs on the specified interface. Because Hellos are suppressed, neighbor relationships are not formed; therefore, no routing updates are sent or received.

## Filtering Routes

Two methods are used to filter routes with OSPF, distribute lists and route maps. The command syntax for configuring each is as follows:

- Router(config-router)**redistribute** *protocol-name* **route-map** *route-map-name*—A route map is a powerful tool that allows easy altering of routing information. Route maps should be used in place of distribute lists whenever possible in OSPF because route maps do not have the limitations of distribute lists, as previously discussed.
- Router(config-router)**distribute-list** [*1-99*] [**in**] [**interface**]—Use this command to call a standard access list to filter inbound routing updates. The **in** option is applied from the view of the router. In other words, to prevent a routing update from entering into the router’s routing table, use the **in** option. Recall that in OSPF, this command only filters the route, not the LSA, so the network being filtered is still placed into the LSDB. Route maps should be used instead.

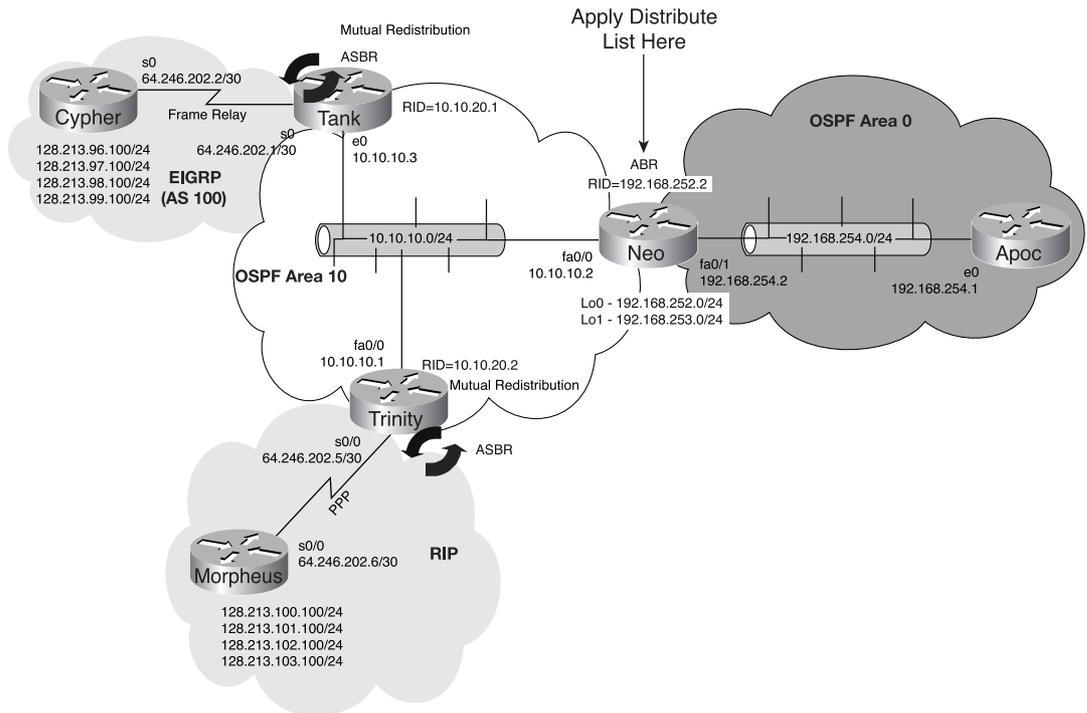
The next section demonstrates the correct manner in which to use a distribute list in OSPF.

## Distribute Lists and OSPF

This chapter has already mentioned many of the difficulties with distribute lists and how Cisco has designed them to interact incorrectly with OSPF. While the recommended practice is to use route maps instead of distribute lists whenever possible, distribute lists still have their place. This section discusses how to configure them correctly to perform route filtering. You are going to use this “gotcha” with OSPF and distribute lists to accomplish the mission at hand and in the process find out about what actually happens:

Your mission is to not allow routing on Router Neo to route network 128.213.102.0/24, but Router Apoc must have a route to it.

Configuring a distribute list has steps that are similar to those needed to configure a route map. In this example, you apply the distribute list to Router Trinity, as shown in Figure 6-13.

**Figure 6-13** Network Topology for Distribute List Example

The following configuration steps are necessary to stop network 128.213.102.0/24 from being known by Router Apoc. Example 6-69 shows a basic OSPF configuration to which you apply a distribute list.

### Before

#### Example 6-69 OSPF Configuration Before Distribution Lists

```
!
router ospf 100
 network 10.10.10.0 0.0.0.255 area 10
 network 192.168.252.0 0.0.0.255 area 0
 network 192.168.253.0 0.0.0.255 area 0
 network 192.168.254.0 0.0.0.255 area 0
!
```

As you can see in Example 6-70, the route to the network in question is present in the routing table of Router Neo, and it must be removed.

**Example 6-70** *Unwanted Route Is in Neo's Routing Table*

```

Neo>show 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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

    64.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O E2   64.246.202.4/30 [110/200] via 10.10.10.1, 00:00:57, FastEthernet0/0
O E2   64.246.202.6/32 [110/200] via 10.10.10.1, 00:00:57, FastEthernet0/0
O E2   64.246.202.0/30 [110/100] via 10.10.10.3, 00:00:57, FastEthernet0/0
    128.213.0.0/24 is subnetted, 7 subnets
O E2   128.213.101.0 [110/200] via 10.10.10.1, 00:00:57, FastEthernet0/0
O E2   128.213.100.0 [110/200] via 10.10.10.1, 00:00:57, FastEthernet0/0
O E2   128.213.103.0 [110/200] via 10.10.10.1, 00:00:57, FastEthernet0/0
O E2   128.213.102.0 [110/200] via 10.10.10.1, 00:00:57, FastEthernet0/0
O E2   128.213.97.0 [110/100] via 10.10.10.3, 00:00:57, FastEthernet0/0
O E2   128.213.96.0 [110/100] via 10.10.10.3, 00:00:57, FastEthernet0/0
O E2   128.213.99.0 [110/100] via 10.10.10.3, 00:00:57, FastEthernet0/0
O E2   128.213.98.0 [110/100] via 10.10.10.3, 00:00:57, FastEthernet0/0
    10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.10.10.0/24 is directly connected, FastEthernet0/0
O       10.10.20.0/24 [110/2] via 10.10.10.1, 00:00:59, FastEthernet0/0
O       10.10.20.1/32 [110/2] via 10.10.10.3, 00:00:59, FastEthernet0/0
C       192.168.254.0/24 is directly connected, FastEthernet0/1
C       192.168.253.0/24 is directly connected, Loopback1
C       192.168.252.0/24 is directly connected, Loopback0
Neo>

```

**During**

The first steps in configuring a distribute list are to identify the network that you want to deny (or permit) and to configure an access list accordingly. The next steps are to assign the access list to OSPF and to specify the direction and interface name. Example 6-71 shows the resulting configuration.

**Example 6-71** *Configuration of the Distribute List*

```

!
router ospf 100
 network 10.10.10.0 0.0.0.255 area 10
 network 192.168.252.0 0.0.0.255 area 0
 network 192.168.253.0 0.0.0.255 area 0
 network 192.168.254.0 0.0.0.255 area 0
 distribute-list 50 in FastEthernet0/0
!
ip classless

```

**Example 6-71** *Configuration of the Distribute List (Continued)*

```

no ip http server
!
access-list 50 deny 128.213.102.0 0.0.0.255
access-list 50 permit any
!

```

Make sure that you do not forget the **permit any** keywords, or all routes will disappear. This is a problem if you are configuring the router via Telnet for this example. Also, for practice, consider reversing the solution given and permit only the networks you want while denying all other routes.

**After**

To verify that you have accomplished your goal, check the routing table on Router Neo and ensure that network 128.213.102.0/24 is missing. As the output in Example 6-72 confirms, it is in fact missing.

**Example 6-72** *Verifying That the Route Has Been Removed*

```

Neo#show 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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

    64.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O E2   64.246.202.4/30 [110/200] via 10.10.10.1, 00:02:38, FastEthernet0/0
O E2   64.246.202.6/32 [110/200] via 10.10.10.1, 00:02:38, FastEthernet0/0
O E2   64.246.202.0/30 [110/100] via 10.10.10.3, 00:02:38, FastEthernet0/0
    128.213.0.0/24 is subnetted, 7 subnets
O E2   128.213.101.0 [110/200] via 10.10.10.1, 00:02:38, FastEthernet0/0
O E2   128.213.100.0 [110/200] via 10.10.10.1, 00:02:38, FastEthernet0/0
O E2   128.213.103.0 [110/200] via 10.10.10.1, 00:02:39, FastEthernet0/0
O E2   128.213.97.0 [110/100] via 10.10.10.3, 00:02:39, FastEthernet0/0
O E2   128.213.96.0 [110/100] via 10.10.10.3, 00:02:39, FastEthernet0/0
O E2   128.213.99.0 [110/100] via 10.10.10.3, 00:02:39, FastEthernet0/0
O E2   128.213.98.0 [110/100] via 10.10.10.3, 00:02:39, FastEthernet0/0
    10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C       10.10.10.0/24 is directly connected, FastEthernet0/0
O       10.10.20.0/24 [110/2] via 10.10.10.1, 00:02:39, FastEthernet0/0
O       10.10.20.1/32 [110/2] via 10.10.10.3, 00:02:39, FastEthernet0/0
C       192.168.254.0/24 is directly connected, FastEthernet0/1
C       192.168.253.0/24 is directly connected, Loopback1
C       192.168.252.0/24 is directly connected, Loopback0
Neo#

```

Now verify the routes that the router still knows about and that the network is still present in its OSPF link-state database, as shown in Example 6-73. Some irrelevant bits are omitted to save space.

**Example 6-73** *Verifying the Route Presence in the OSPF LSDB*

```

Neo#show ip ospf database

                OSPF Router with ID (192.168.253.1) (Process ID 100)
! some output omitted for brevity

                Type-5 AS External Link States

Link ID        ADV Router    Age           Seq#           Checksum Tag
64.246.202.0   10.10.20.1   1329         0x8000000C    0x1805  0
64.246.202.4   10.10.20.2   1515         0x8000000B    0xF5F5  200
64.246.202.6   10.10.20.2   1515         0x8000000B    0xF3F2  200
128.213.96.0   10.10.20.1   1329         0x8000000C    0x75E   0
128.213.97.0   10.10.20.1   1329         0x8000000C    0xFB68  0
128.213.98.0   10.10.20.1   1329         0x8000000C    0xF072  0
128.213.99.0   10.10.20.1   1329         0x8000000C    0xE57C  0
128.213.100.0  10.10.20.2   1515         0x8000000B    0xE053  200
128.213.101.0  10.10.20.2   1515         0x8000000B    0xD55D  200
128.213.102.0  10.10.20.2   1516         0x8000000B    0xCA67  200
128.213.103.0  10.10.20.2   1516         0x8000000B    0xBF71  200
Neo#

```

The final check, shown in Example 6-74, is to determine if the route is still in Router Apoc's routing table and to make sure that you are going to clear the routing table prior to executing the command.

**Example 6-74** *Clearing the Routing Table*

```

Apoc#clear ip route *
Apoc#show 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

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O IA   10.10.10.0/24 [110/11] via 192.168.254.2, 00:00:03, Ethernet0
O IA   10.10.20.0/24 [110/12] via 192.168.254.2, 00:00:03, Ethernet0
O IA   10.10.20.1/32 [110/12] via 192.168.254.2, 00:00:03, Ethernet0
    64.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O E2   64.246.202.4/30 [110/200] via 192.168.254.2, 00:00:03, Ethernet0
O E2   64.246.202.6/32 [110/200] via 192.168.254.2, 00:00:03, Ethernet0
O E2   64.246.202.0/30 [110/100] via 192.168.254.2, 00:00:03, Ethernet0
    128.213.0.0/24 is subnetted, 8 subnets

```

**Example 6-74** *Clearing the Routing Table (Continued)*

```

O E2 128.213.101.0 [110/200] via 192.168.254.2, 00:00:03, Ethernet0
O E2 128.213.100.0 [110/200] via 192.168.254.2, 00:00:03, Ethernet0
O E2 128.213.103.0 [110/200] via 192.168.254.2, 00:00:03, Ethernet0
O E2 128.213.102.0 [110/200] via 192.168.254.2, 00:00:03, Ethernet0
O E2 128.213.97.0 [110/100] via 192.168.254.2, 00:00:03, Ethernet0
O E2 128.213.96.0 [110/100] via 192.168.254.2, 00:00:05, Ethernet0
O E2 128.213.99.0 [110/100] via 192.168.254.2, 00:00:05, Ethernet0
O E2 128.213.98.0 [110/100] via 192.168.254.2, 00:00:05, Ethernet0
O 192.168.252.0/24 [110/11] via 192.168.254.2, 00:00:05, Ethernet0
O 192.168.253.0/24 [110/11] via 192.168.254.2, 00:00:05, Ethernet0
C 192.168.254.0/24 is directly connected, Ethernet0
Apoc#

```

This was a strange scenario, but it demonstrates the use and issues that are associated with distribute lists. This section covered redistribution with many examples and with an emphasis on the redistribution of default routes and described how external routes are used. In addition several redistribution examples were presented to include samples of mutual redistribution. The next section discusses summarization—a technique that is used to properly represent a block of IP addresses within a routing table in the fewest possible entries.

## Chapter Summary

This chapter covered route redistribution and how it operates within OSPF. You have just begun to scratch the surface of the potential for redistributing routes; you will find that from the topics presented, you are ready to solve redistribution scenarios or design them. A variety of golden rules were covered as they relate to redistribution. These rules form the basis of understanding how to design redistribution scenarios in OSPF. When you have trouble with redistribution, these golden rules are also an excellent troubleshooting tool.

A variety of different configuration examples and caveats were covered as well. It is important to understand that redistribution in OSPF can be highly dependent on what type of route you are trying to redistribute, as discussed. The discussion concluded with route tags and distribute lists. In Chapter 7, you begin to explore the complex topic of summarization.