Chapter 10: Configuring IP Routing Protocols on Cisco Routers

This chapter discusses the implementation of static and dynamic IP routing on Cisco routers in an enterprise network environment. To start, we evaluate route distribution methods and explain the general configuration elements that apply to configuring dynamic routing protocols on Cisco routers. Then, we review basic steps for configuring and monitoring IP static routing, RIP, IGRP, EIGRP, AURP, and OSPF. For easy reference, each section starts with a command summary list and utilizes (whenever possible) the same core network topology (testnet, our example network) illustrated in Figure 10.1. This chapter concludes with the implementation of BGPv4 and an overview of IP route redistribution.

The best way to learn something is to do it. Therefore, as you did in Chapter 7, "Introduction to Cisco Routers," you should recreate the configuration examples using a single router or pair of routers. The section examples revolve around the asbr-a1/2 and asbr-b1/2. Loopback interfaces can be used to simulate additional network interfaces. To create a loopback interface, from global configuration mode, type the following:

```
Router#config t
Router(config)#interface loopback 1
Router(config-if)#
```

Figure 10.1
The testnet network map.

The Boston network will use networks with the classful root of 172.17.0.0, and Albany will use 192.168.0.0 for its IP address space. Classless/VLSM addressing will be used with the protocols that can support it.

Choosing the Right Protocol

The task of enabling IP route advertisement varies in complexity, depending on the operational scope and size of the network. Static routing, which is by far the easiest and most problem-free method, is tedious to manage in large networks and provides no recovery facility when link failures occur. Alternatively, dynamic routing protocols address the shortcomings of static routing, but do, however, come with their own operational concerns.

Unfortunately, when selecting an IP announcement methodology, there is no single correct answer. There is, however, a wrong answer, which is to select a method that will not be able to meet the networks operational requirements or scale appropriately to enable future growth. The way to avoid this problem is to examine the network’s topology, availability, and performance requirements through a series of questions:

- What kind of network is this (ISP backbone/multipolicy, single GW LAN/WAN, multipoint LAN/WAN, enterprise backbone/single-policy, and so on)?
- What is the network diameter (more or fewer than 15 hops, how many routers, and so on)?
• Is CIDR/VLSM addressing support required?

• Does the network use redundant or multiple paths between network segments?

• What type of equipment will be used to route traffic on the network, and is a standards-based routing protocol required?

• What are the performance requirements of the routers in the network? For example, is convergence time a factor?

After you have a list of requirements, you can review your available options and determine the solution appropriate for your networking environment. Table 10.1 provides a brief feature comparison of the popular IP routing options supported on the Cisco IOS.

Table 10.1 Comparison of Routing Protocol Features

<table>
<thead>
<tr>
<th>Protocol Feature</th>
<th>RIP v1/v2</th>
<th>IGRP</th>
<th>EIGRP</th>
<th>OSPF</th>
<th>Static</th>
<th>BGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supports classful addressing</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Interior Gateway Protocol</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Exterior Gateway Protocol</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Supports classless addressing</td>
<td>yes (V2)</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Supports load sharing</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Supports authentication</td>
<td>yes (v2)</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Easy implementation</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Routing algorithm</td>
<td>DV</td>
<td>DV</td>
<td>DUAL</td>
<td>LS</td>
<td>none</td>
<td>DV</td>
</tr>
<tr>
<td>Supports weighted metrics</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Fast convergence</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Uses broadcasts for route updates</td>
<td>yes (v1)</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Uses multicast for routing updates</td>
<td>yes (v2)</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Supports large network diameters</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
**Route Selection**

The router will use all the available sources of reachability information to construct the most accurate and efficient routing table possible, based on the available information. Each information source is assigned an administrative distance, which is used to determine a route’s integrity. Table 10.2 lists the IOS’s default administrative distance values for its supported IP routing protocols.

**Table 10.2 IOS Administrative Distances**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected interface</td>
<td>0</td>
</tr>
<tr>
<td>Static route</td>
<td>1</td>
</tr>
<tr>
<td>EIGRP summary route</td>
<td>5</td>
</tr>
<tr>
<td>BGP (external)</td>
<td>20</td>
</tr>
<tr>
<td>EIGRP (internal)</td>
<td>90</td>
</tr>
<tr>
<td>IGRP</td>
<td>100</td>
</tr>
<tr>
<td>OSPF</td>
<td>110</td>
</tr>
<tr>
<td>RIP</td>
<td>120</td>
</tr>
<tr>
<td>EIGRP (external)</td>
<td>170</td>
</tr>
<tr>
<td>BGP (internal)</td>
<td>200</td>
</tr>
<tr>
<td>Unknown</td>
<td>255</td>
</tr>
</tbody>
</table>

The lower the administrative distance value, the more trusted the route. So, you can see in situations where multiple routing protocols are being used for route advertisement, the router will prefer information provided from certain protocols over others.

One of the big advantages of employing this advertisement hierarchy is that it gives you another way to manage traffic flow in dynamic routing. By using static routes (which have a lower administrative distance than any dynamic protocol), you can overrule dynamic announcements in multipath networks to specify the route path to specific hosts. By the same token, it is possible to set a static route to use a higher administrative distance so a dynamic route is preferred and the static route is only used in the event that dynamic route announcement is lost.

**Displaying General Routing Information**

There are several IOS commands used for controlling and displaying information about IP routing.
Display commands:

- `<show ip route>`
- `<show ip route connected>`
- `<show ip route [address/hostname]>`
- `<show arp>`
- `<show ip protocol>`
- `<show ip masks>`
- `<show ip masks [network address]>`
- `<traceroute>`

Control commands:

- `<clear ip route *>`
- `<clear ip route [network] [mask]>`
- `<clear arp>`

**Displaying IP Network Information**

In Chapter 7, "Introduction to Cisco Routers," the `<show ip route>` user EXEC command was introduced. Now, let’s take a closer look at the IOS’s route table display:

```
BBR-172#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
    D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
    i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
    U - per-user static route, o - ODR
    T - traffic engineered route
Gateway of last resort is 172.16.191.220 to network 0.0.0.0
    172.16.0.0/16 is variably subnetted, 17 subnets, 6 masks
      S    172.16.80.0/24 [1/0] via 172.16.191.220
      B    192.168.91.0/24 [20/0] via 192.168.0.2, 6d14h
      O    172.16.48.0/21 [1/0] via 172.16.191.140
      O    192.168.0.1/29 is directly connected, Serial0/2
      C    172.16.12.0/24 is directly connected, Ethernet3/1
      O IA  172.16.160.96/30 [110/1563] via 172.16.191.220, 4d21h, FastEthernet2/0
      C    172.16.192.0/21 is directly connected, Fddi6/0
      B    192.168.176.0/21 [20/0] via 192.168.85.75, 6d14h
      C    172.16.181.0/24 is directly connected, Ethernet3/0
      O    172.16.186.0/24 [1/0] via 147.225.12.250
      O    172.16.185.0/24 [1/0] via 147.225.12.250
      C    172.16.191.0/24 is directly connected, FastEthernet2/0 [1/0]
```

The `<show ip route>` command output displays all the known network routes and the following information (from left to right):

- The source of the route. The legend for all available protocols is displayed at the top of the route table.

- The network and netmask. The netmask can be displayed using bitcount (shown), decimal (default), or hexadecimal. To set the netmask display format, use the `<ip netmask-format [format]>` command. The command can be used for a temporary `<terminal>` session.
<terminal ip netmask-format [format]> or used to permanently set the mask format display in configuration EXEC line mode command:

```
router# config t
Enter configuration commands, one per line. End with CNTL/Z.
router(config)# int vty 0 4
router(config-line)# ip netmask-format bit-count
Router(config-line)#^Z
Router#
```

- The route’s administrative distance and routing metric.
- The network’s next hop gateway. If the route is learned dynamically, the route’s age or last update time and the interface that the route was learned on may also be listed.
- The gateway of last resort, if one has been set.

Keep in mind that the routing table displays the best routes, not necessarily all available routes. Only the route with the best administrative distance/metric will appear in the table. In cases where routes of equal cost to the same network are available, both will be listed. Routes to directly connected networks will not appear for disabled interfaces, and neither will static routes that are dependent on the disabled interface. All dynamic routing protocols also require that at least one active interface exists and that its address corresponds to a network being announced by the protocol. Although it would seem odd to announce a route for a network that is not active, it is quite common for dial-on-demand and dial access-servers to use network address space on interfaces that are not available all the time. Static routes are usually used to provide reachability information in these situations. An alternative is to use a loopback interface, which acts as a placeholder for the network, and allows it to be announced dynamically.

In addition to the basic `<show ip route>` form of the command, the `<show ip route connected>` form of the command can be quite useful. `<show ip route connected>` displays all the network routing information about all the active router interfaces:

```
Router>sh ip route connected
C  192.168.0.0/24 is directly connected, Serial0
    192.168.1.0/26 is subnetted, 3 subnets
    C  192.168.1.64 is directly connected, Ethernet0
    C  192.168.1.128 is directly connected, Loopback2
    C  192.168.1.192 is directly connected, Loopback1
```

The `<show ip route connected>` command will display all the routing entry information for the specified network or host:

```
Router>sh ip route 172.16.191.1
Routing entry for 172.16.191.0/24
    Known via "static", distance 20, metric 0
    Routing Descriptor Blocks:
    * 192.168.0.2
        Route metric is 0, traffic share count is 1
```

This command variant is quite useful when debugging route announcement issues that arise when you use dynamic routing protocols and route redistribution.
The `<show ip masks [address]>` command is also invaluable when you’re employing classful or classless subnetting. Errors in subnetting large address spaces are common. This command lists all the mask variations used with a given IP address space:

```
router#show ip masks 172.16.0.0
Mask      Reference count
255.255.255.255 4
255.255.255.252 1
255.255.255.248 2
255.255.255.128 1
255.255.255.0  6255.255.248.0  3
```

A problem commonly associated with dynamically addressed networks is incorrect ARP table entries. For example, a node’s IP address can change or a new IP address is assigned to a node that already had an address on the same segment (an expired DHCP lease, perhaps). These are prime targets for bad ARP entries. Since the router plays such an important role in data delivery for all the connected nodes, it is essential that the router’s ARP table be correct. A good place to start looking when nodes are suddenly unreachable and no hardware failure exists is the router’s ARP tables. Like the IP routing table, the ARP table is viewable in user EXEC mode. To display the ARP table, the `<show arp>` command is used:

```
Router>show arp
Internet 192.168.191.220    -  00e0.1ef2.15a1 ARPA  Ethernet2/0
Internet 192.168.191.253     0  00c0.4906.9488 ARPA  Ethernet2/0
Internet 192.168.191.240     0  00e0.1e34.a758 ARPA  Ethernet2/0
Internet 192.168.191.141  37  00e0.b08d.ccf0 ARPA  Ethernet2/0
Internet 192.168.190.136     -  00e0.1ef2.15b1 ARPA  Ethernet3/0
Internet 192.168.190.137     -  00e0.1ef2.15b1 ARPA  Ethernet3/0
Internet 192.168.190.138     -  00e0.1ef2.15b1 ARPA  Ethernet3/0
Internet 192.168.190.139     -  00e0.1ef2.15b1 ARPA  Ethernet3/0
```

A great tool for displaying information about the dynamic IP routing protocols running on the router is `<show ip protocols>`. This command can display a verbose or terse report. Verbose is the default, and it displays all the running processes, their update status, the networks being announced, and any neighbor routers:

```
ASBR-34#sh ip protocols
Routing Protocol is "bgp 66"
Sending updates every 60 seconds, next due in 0 seconds
Outgoing update filter list for all interfaces is
Incoming update filter list for all interfaces is
IGP synchronization is enabled
Automatic route summarization is disabled
Neighbor(s):
Address     FiltIn FiltOut DistIn DistOut Weight RouteMap
172.16.85.75     20   6d16h
Distance: external 20 internal 200 local 200
```

The terse report is retrieved by using the `<show ip protocols summary>` command. It only lists the protocol’s process and, if applicable, the process ID:

```
Router#sh ip protocols summary
Index Process Name
 0  connected
 1  static
 2  ospf 45
 3  rip
Router#
```

Managing IP Routing Information

Problems do arise, and when they do, it is best to start slowly by verifying that what you believe to be happening is in fact happening. The `<show>` commands work well in this regard. In most situations where IP routing is the suspect, the problem is lack of a route, or a bad route or ARP entry.

To flush the entire routing table, the privileged EXEC command `<clear ip route *>` is used. To remove an individual IP route entry, use

```
<clear ip route [address] [mask]>
```

When the route table is flushed, the router will recreate the table starting with its directly attached interfaces. If a dynamic protocol is in use, the flush will trigger an update immediately. Static route entries that are loaded from the configuration file at boot will also be restored shortly after the table has been flushed. Be aware that while the router rebuilds its route table, connectivity will be lost. The connectivity outage will vary in length from a few seconds to several minutes, depending on the size of network and the routing table.

The `<clear arp>` privileged EXEC command clears the router’s ARP table. If a situation requires the entire IP route table to be cleared, it is not a bad idea to clear the ARP table as well, preferably before you flush the IP route table. This way, you will ensure that the router has accurate ARP entries for its adjacent neighbors before the route table is cleared. The ARP table will reconstruct itself quickly, since it is only dependent on connectivity with nodes that are directly accessible to its connected interfaces.

Because dynamic protocols refresh route entries automatically, only the most drastic situations will require the entire route table to be flushed. In most cases, only a single failed route is the problem, such as a dynamic route in a hold-down state (waiting to expire) or a static route that points to a nonexistent gateway. To remove an invalid dynamic route from the routing table, use `<clear ip route [network] [mask]>`. It is also possible to clear a single ARP table entry with `<clear arp [mac address]>`. Removal of an invalid static route must be done from global configuration EXEC mode. After you’re in configuration mode, place a `<no>` in front of the same command string you would use to create the route:

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip route 192.160.0.4 255.255.255.252 172.16.0.6 20
Router(config)#^Z
```

After the static route is removed from the configuration, the IOS will purge it from the IP routing
Note - When changing any IP routing behavior, it is best to perform changes from the router’s console. VTY line connectivity can be interrupted while some changes are taking effect. Consequently, if the change fails, all VTY connectivity might be lost, leaving the router inaccessible.

Managing Static Routing

Display commands:

<show ip route><show interface [type/slot/port]>

Global configuration commands:

[ip route [network] [mask] [gateway] [administrative distance]>
[ip classless>
[ip subnet-zero>
[ip forward-protocol udp [port number]>
[no ip source-route>

Interface configuration subcommands:

[ip helper address [ip address]>
[bandwidth>
[mtu>
[no ip redirects>
[no ip unreachable>

Control commands:

<copy tftp [route-table-name]>

Although static routing is not an ideal total solution for IP route announcement in large networks, it is a very effective solution for small (single gateway) networks and provides needed gateway and route redirection services. Static routes are also essential to announcing networks where the access links are unstable or temporary (as with dial-up connections).

Static routes are set in the Cisco IOS using the <ip route> configuration EXEC command. As noted in the previous section, static routes are set in the router’s configuration file. The IOS’s handling of static routes is different than, say, a UNIX or NT workstation that has static entries in its startup configuration file. With the IOS, static route entries are managed as an IP routing process, so they are reloaded after routing table flush. Static routes entered on an end-station would typically need to be re-entered or reloaded (by rebooting the system) after the routing table has been flushed.

Configuring Default Routing

If the IP network is closed (network reachability is limited to explicitly announced networks defined
in the routing table), a default route is not needed. A typical example is a closed or private network where there is no Internet access or where access is provided through a proxy server or firewall. In closed network architectures, typically, traffic destined for unannounced networks is discarded and the user is notified with an ICMP message. To disable ICMP notifications and ICMP redirection attempts, the `<no ip unreachables>` and `<no ip redirects>` commands can be set on the router’s connected interfaces. These options also provide an additional level of network security by limiting the capability of IP traffic to be redirected across a path that may insecure.

In most situations, however, a default route is needed. The most common one is where a single point Internet connection exists or where it is undesirable to exchange routing information but reachability information is required.

In Chapter 7, we first used the `<ip route>` command to set the default route on the Concord and Ridgefield routers. In this case, we are setting the default route for asbr-a2 to forward all traffic that has no explicit route out asbr-a1’s dedicated Internet link through Fast Ethernet 0/0 (refer to Figure 10.1). If the link fails, it should forward all of the traffic out interface s2/0 to asbr-b2. Notice on the route entry for 192.168.0.6, we’ve added a number for administrative distance of 20:

```
asbr-a1#config t
Enter configuration commands, one per line. End with CNTL/Z.
asbr-a1(config)#ip route 0.0.0.0 0.0.0.0 192.168.0.6 20
asbr-a1(config)#ip route 0.0.0.0 0.0.0.0 192.168.191.20
asbr-a1(config-if)#^Z
```

When interface Fast Ethernet 0/0 is up, the router designates 192.168.191.19 as its gateway of last resort:

```
asbr-a2#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route, o - ODR
Gateway of last resort is 192.168.191.20 to network 0.0.0.0

    192.168.181.0/25 is subnetted, 1 subnets
    C 192.168.12.0/24 is directly connected, Ethernet1/0
    C 192.168.181.128 is directly connected, Ethernet1/3
    C 192.168.10.0/24 is directly connected, Ethernet1/2
    C 192.168.191.0/24 is directly connected, FastEthernet0/0
    192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
    C 192.168.0.4/30 is directly connected, Serial2/0
    C 192.168.192.0/21 is directly connected, Ethernet1/1
```

If interface Fast Ethernet 2/0 fails, 192.168.0.6 (accessible through interface serial2/0) is designated as the gateway of last resort:

```
asbr-a2#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
```
By setting the administrative distance on the static route pointing to asbr-b2, the route only appears in the routing table when a route with a lesser administrative distance does not exist (in other words, when the router’s interface is down). In this case, the default static route (which uses the default administrative distance of 0) directs all remote traffic out through the local Internet connection via asbr-a1.

The problem with this solution is the intrinsic nature of static routes. The route fail-over only occurs if the interface on the router fails. If the next hop interface is unavailable but the local router interface is still on line, the router will continue to forward traffic to the unavailable interface. This solution does offer some redundancy over having nothing at all, because statistically, the Fast Ethernet hub will probably fail (disabling the interface) before the next hop router’s hardware interface will.

### Using Static Routing in the Enterprise

Static routing, although not the best method, is often used for route advertisement in enterprise networks because it is stable and removes the possibility of IP traffic being misdirected by some runaway, misconfigured, or broken dynamic routing process. Static routing is nevertheless difficult to scale when you manage networks with large routing tables.

One way to scale a static routing configuration is to manage the static tables like access-lists, using `copy` and TFTP. Because the most tedious task in maintaining large static routing tables is adding and removing static routes, copying those changes to the `<running-config>` is more efficient than making single line changes in configuration mode. This approach also has the major advantage of having a copy of the routing table separate from the router configuration; it’s easier to load the table to a different router in case of a hardware failure.

For asbr-a2 to reach all the testnet networks, 15 static routes are required. Here is the static route config file for asbr-a2. Like the default gateway scenario, a secondary route is provided with a higher administrative distance, to be used in case of a link failure:

```
! -To reach networks connected to asbr-b1 via asbr-a1 (primary) and asbr-b2 (backup)
ip route 172.16.3.0 255.255.255.0 192.168.191.20
ip route 172.16.3.0 255.255.255.0 192.168.0.6 40
ip route 172.16.2.192 255.255.255.192 192.168.191.20
ip route 172.16.2.192 255.255.255.192 192.168.0.6 40
ip route 172.16.2.64 255.255.255.192 192.168.191.20
ip route 172.16.2.64 255.255.255.192 192.168.0.6 40
ip route 172.16.2.192 255.255.255.192 192.168.191.20
ip route 172.16.2.192 255.255.255.192 192.168.0.6 40
ip route 172.16.2.128 255.255.255.192 192.168.191.20
ip route 172.16.2.128 255.255.255.192 192.168.0.6 40
```
Equal hop networks off asbr-b1 and asbr-b2

```
! Equal hop networks off asbr-b1 and asbr-b2
! ip route 172.16.91.0 255.255.255.0 192.168.191.20
ip route 172.16.91.0 255.255.255.0 192.168.0.6
ip route 192.168.0.128 255.255.255.128 192.168.191.20
ip route 192.168.0.128 255.255.255.128 192.168.0.6
ip route 192.168.0.9 255.255.255.252 192.168.191.20
ip route 192.168.0.9 255.255.255.252 192.168.0.6
ip route 192.168.0.13 255.255.255.252 192.168.191.20
ip route 192.168.0.13 255.255.255.252 192.168.0.6
! -To reach networks connected to asbr-b2 via asbr-b2 (primary) and asbr-a1 (backup)
! ip route 172.16.1.0 255.255.255.192 192.168.0.6
ip route 172.16.1.0 255.255.255.192 192.168.191.20 40
ip route 172.16.1.64 255.255.255.192 192.168.0.6
ip route 172.16.1.64 255.255.255.192 192.168.191.20 40
ip route 172.16.1.128 255.255.255.192 192.168.0.6
ip route 172.16.1.128 255.255.255.192 192.168.191.20 40
ip route 172.16.1.192 255.255.255.192 192.168.0.6
ip route 172.16.1.192 255.255.255.192 192.168.191.20 40
ip route 172.16.2.0 255.255.255.0 192.168.0.6
ip route 172.16.2.0 255.255.255.0 192.168.191.20 40
! To reach networks of asbr-a1 via asbr-a1 (primary) and asbr-b2 (backup)
ip route 192.168.160.96 255.255.255.252 192.168.191.20
ip route 192.168.160.96 255.255.255.252 192.168.0.6 40
! Default routes
ip route 0.0.0.0 0.0.0.0 192.168.0.6 40
ip route 0.0.0.0 0.0.0.0 192.168.191.20end
```

To load the static table, use the `<copy tftp running-config>` privileged EXEC command:

```
  asbr-a2#copy tftp run
  Host or network configuration file [host]? 192.168.191.202
  Name of configuration file [asbr-a1-confg]? asbr-a2.static
  Configure using asbr-a2.static from 192.168.191.202? [confirm]
  Loading asbr-a2.static from 192.168.191.202 (via Ethernet0): !
  [OK - 1984/32723 bytes] asbr-a2#
```

After the routing table is loaded, the IP route table looks like this:

```
  asbr-a1#sh ip route
  Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
          D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
          N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
          E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
          i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
          U - per-user static route, o - ODR

  Gateway of last resort is 192.168.191.20 to network 0.0.0.0

  192.168.181.0/25 is subnetted, 1 subnets
     C  192.168.181.128 is directly connected, Ethernet1/3
  192.168.10.0/24 is directly connected, Ethernet1/2

  192.168.160.0/30 is subnetted, 1 subnets

  192.160.0/16 is variably subnetted, 10 subnets, 2 masks
     S  172.16.0.0/16 is variably subnetted, 10 subnets, 2 masks
        S  172.16.1.128/26 [1/0] via 192.168.0.6
        S  172.16.2.128/26 [1/0] via 192.168.191.20
        S  172.16.1.192/26 [1/0] via 192.168.0.6
        S  172.16.2.192/26 [1/0] via 192.168.191.20
```
In the event that the Fast Ethernet 0/0 on asbr-2a fails, the routing would adjust itself to only use 192.168.0.6 as the next hop gateway:

```
asbr-al#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
       U - per-user static route, o - ODR
Gateway of last resort is 192.168.0.6 to network 0.0.0.0

192.168.181.0/25 is subnetted, 1 subnets
 C   192.168.181.128 is directly connected, Ethernet1/3
 C  192.168.10.0/24 is directly connected, Ethernet1/2
192.168.160.0/30 is subnetted, 1 subnets
 S   192.168.160.96 [40/0] via 192.168.0.6
 S  172.16.0.0/16 is variably subnetted, 1 subnets, 2 masks
 [ ] via 192.168.0.6
 S  172.16.2.128/26 [40/0] via 192.168.0.6
 S  172.16.2.192/26 [40/0] via 192.168.0.6
 S  172.16.1.0/26 [1/0] via 192.168.0.6
 S  172.16.2.0/24 [1/0] via 192.168.0.6
 S  172.16.3.0/24 [40/0] via 192.168.0.6
 S  172.16.91.0/24 [1/0] via 192.168.0.6
 S  172.16.1.64/26 [1/0] via 192.168.0.6
 S  172.16.2.64/26 [40/0] via 192.168.0.6
192.168.0.0/24 is variably subnetted, 3 subnets, 3 masks
 S  192.168.0.0/24 [1/0] via 192.168.0.4
 C  192.168.0.4/30 is directly connected, Serial2/0
 [1/0] via 192.168.0.6
 S*  0.0.0.0/0 [40/0] via 192.168.0.6
 C  192.168.192.0/21 is directly connected, Ethernet1/1
```

If the asbr-2a interface came back up, 192.168.191.20 would become the default route again.

As you can see, it is possible to manage IP route announcements using only static routing. However, this example also shows the work required to manage static tables sanely.

**Configuring Classless Routing**
Cisco IOS provides support for both classful and classless addressing. Depending on the IOS, version support for both is enabled by default. To verify if classless support is enabled, use the privileged EXEC command `show running-config` and check the configuration file for the configuration command `ip classless`. If it appears, it is enabled. If not, and classless support is required, then in global configuration EXEC mode, enter the command `ip classless`. When enabled, variable subnetting is permitted of classless address spaces.

One easy way to use VLSM is to break up the address space in blocks of four, because the theory behind classless addressing is any address space that is a multiple of two (for example, \(2 \times 2 = 4 = /30\) mask, \(2 \times 4 = 8 = /29\) mask, and so on) minus the network and broadcast address. Although this is correct in theory, in order to maintain computability with classful subnetting, by default, IOS does not permit the usage of network addresses that end on natural zero boundaries. This is done because these addresses can be mistaken for network broadcast addresses because the netmask is not sent along in the packet’s address information. To enable the use of "zero boundary" network addresses, which is particularly useful when working with CIDR’s created out of Class C addresses, IOS provides the global configuration EXEC command `ip subnet-zero`. Here is an example of what happens when a zero boundary address is used without enabling `ip subnet-zero`:

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int e0
Router(config-if)#ip address 192.168.0.1 255.255.255.252
Bad mask /30 for address 192.168.0.1
Router(config-if)#
With the <ip subnet zero> option enabled, the address is permitted:
Router#config t
Enter configuration commands, one per line. End with CNTL/Z
Router(config)#ip subnet-zero
Router(config)#int e0
Router(config-if)#ip address 192.168.0.1 255.255.255.252
Router(config-if)#^Z
```

**Configuring IP Control Services**

IOS supports a number of options to adjust IP’s behavior. Some of these commands enhance the security of IP, and others enhance IP’s performance:

- `<ip forward-protocol udp [port number]>`—This command allows IP UDP broadcasts to be forwarded between IP network segments that are connected to each other across the router. IP (Layer 3) broadcasts are not forwarded between networks by default. This option allows UDP broadcast service requests like DHCP and BOOTP to be relayed to servers on remote networks. This command must be used with `<ip helper address [address]>`, which is set on the router interface that is performing the forwarding. The address specifies where the request should be forwarded. To configure DHCP/BOOTP forwarding:

  ```
  Router(config)#ip forward-protocol udp 67
  Router(config)#int e0
  Router(config-if)#ip helper-address 192.168.1.24
  ```

- `<no ip source-route>`—IP source routing is an option built into the IP protocol suite for testing (source routing is indicated in the IP datagram's header). It provides the capability to dictate the route path that a datagram will travel. Source routing, like ICMP redirects, can be
exploited to create potential security problems. Unless required, source routing should be disabled.

- `<ip mtu [bytes]>`—Each router interface has a default MTU size, which is used for determining if IP fragmentation is needed. The IOS supports MTU path discovery, which allows the devices along the route path to adjust for differences in maximum MTU sizes. Ideally, an MTU of 1500 is preferred, because it provides for the maximum PDU size allowable for Ethernet, which will eliminate the need for resizing. Certain Layer 2 protocols support MTU sizes larger than 1500, which means that when packets are exchanged between the two different protocols, fragmentation occurs. Whenever possible it is best to reduce interfaces with larger MTU sizes down to 1500. The MTU value is part of the IGRP and EIGRP route metric set, but has no direct role in determining a route’s vector (metric cost).

- `<bandwidth [kilobytes]>`—The `<bandwidth>` is a configurable parameter on each of the router’s interfaces used for setting the interface’s data transfer rate in kilobytes per second. This value is also used by IGRP, EIGRP, and OSPF to calculate route metrics. An interface's MTU and bandwidth values are displayed as part of the `<show interface [type/slot/port]>` command output:

```
Router#show interfaces e0
Ethernet0 is administratively down, line protocol is down
Hardware is Lance, address is 0010.7b37.b27c (bia 0010.7b37.b27c)
Internet address is 192.168.0.1/30
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1
```

### Configuring Dynamic IGP and EGP IP Routing Protocols

All dynamic IP routing protocols are configured as IOS sub-processes, much the same way a router interface is configured. The `<router [protocol] [process id]>` global configuration command enables the process:

```
asbr-a1#config t
asbr-a1(configure)#router ospf 89
asbr-a1(config-router)#network 172.16.0.0 0.0.255.255 area 0.0.0.0
asbr-a1(config-router)#
```

After a process is enabled, it does not become active until a `<network [IP network address]>` statement that corresponds to a configured router interface is added to the subprocess.

After the process is configured, you need to add each directly connected network that you wish the protocol to announce. If the network is not added to the process, it will not be announced by the protocol. Occasionally, you might want to receive, but not send, network announcements on an interface (and you still want the network announced by the routing process). This is common when a routing protocol is used to manage a "private" network that is attached to an Internet gateway, which is also using a routing protocol. In situations like this, the interface can be run in passive mode using the routing configuration subprocess command `<passive-interface [interface]>`:

```
asbr-a1#config t
asbr-a1(configure)#router ospf 89
asbr-a1(config-router)# passive-interface s1
```
After the interface is in passive mode, it will only receive routing announcements. When a router interface has been configured to operate in passive mode, it will only accept routing announcements. This option is useful when you only want the router to obtain reachability information, but not announce. Initially, this might seem odd, this is a common practice in situations where data services are being provided by an outside network (for example, Internet Gateway or News Service feed). In these situations, the provider will use a static route to reach your gateway, but otherwise, has no need for your routing information. However, by having the gateway listening to the provider’s route announcements, the router will then incorporate the reachability information into its routing update that it transmits to the other routers. This way, when the provider makes changes to its network, your network is dynamically updated instead of you having to add or remove static routes.

**Note** - The IGP process ID is often the same number as the network’s autonomous system (AS) number. This is a practice and not a requirement; if your network does not have an assigned AS number, any number will do. When using RIP, no process ID is required because only one RIP process is supported. This is in contrast to IGRP, EGRP, and OSPF, where multiple processes can be supported.

Additionally, at least one interface must be configured with an IP address in order for any routing process to be created.

Here is an example of using the `<passive-interface>` command to manage an internal and external route announcement. The Albany campus network uses `asbr-a1` to announce of the local network to its local ISP (refer to Figure 10.1). It then uses OSPF to manage announcements internally with the Boston network.

Here are the RIP and OSPF configurations used on `asbr-a1`:

```plaintext
router rip
  version 2
  network 12.0.0.0
  network 192.168.160.0
  network 192.168.191.0
  network 192.168.161.0
  passive-interface e1
  passive-interface s1
  neighbor 12.14.116.66
  no auto-summary
router ospf 89
  network 12.14.116.64 0.0.0.3 area 0.0.0.0
  network 192.168.0.0 0.0.0.3 area 0.0.0.0
  network 192.168.191.0 0.0.0.255 area 0.0.0.0
  network 192.168.160.96 0.0.0.3 area 0.0.0.0
  network 192.168.161.128 0.0.0.128 area 0.0.0.0 passive-interface s1
```

By enabling the interfaces as passive, only the active interfaces announce all the networks that are part of the process. The RIP process announces the local network to the ISP using interface `s1`. The `e1` interface is also active, so it can send/receive RIP updates with `asbr-a2`. None of the internal WAN networks are included in the RIP process. The OSPF process suppresses announcements on interface `s1` and includes `asbr-a1`’s WAN link, which exchanges network announcements with the
Using RIPv1 and RIPv2

Global configuration commands:

```plaintext
<router rip>
<key chain [key chain name]>
<key [id number]>
<key-string>
<default key-string>
<access-list [1-99]>
```

Global router subprocess commands:

```plaintext
<neighbor>
<timers basic [update] [invalid] [holddown] [flush]>
<distance>
<auto-summary>
<version [1] [2]>
<offset-list [acl] [in/out] [metric] [interface type/slot/port]>
```

Global interface subprocess commands:

```plaintext
<ip rip send version [1/2/1 2]>
<ip rip receive version [1/2/1 2]>
<ip rip authentication mode [md5] [text]>
```

Control commands:

```plaintext
<debug rip>
<debug rip events>
```

RIP is an IGP, distance vector-based routing protocol defined in RFC 1058. RIP uses a single routing metric—hop count—to determine the best route path. RIP, you may recall, has two versions: 1 and 2. The main differences are that version 1 uses UDP broadcasts and has no support for authentication and VLSM. RIP version 2 was added to Cisco IOS version 11.1, and it uses multicast for sending routing updates, as well as supports MD5 and text key authentication and VLSM. The IOS, by default, sends version 1 updates but will receive both version 1 and 2 updates. RIP’s strength as a protocol is that it is supported in some form by virtually every network hardware vendor and operating system platform.

In the previous example, RIP made the network announcements between the local Albany network and its ISP. The Albany network uses both classful and classless addressing, and since netmasks need to be sent in the update messages, RIP version 2 was required. You can force this in the configuration by using the `<version [1] [2]>` subcommand. Using the version command disables RIPv1 support altogether. Alternatively, if you need to support some legacy hardware that only speaks RIPv1, you can specify on each interface what message types to support:

```plaintext
asbr-al(config)# interface e0
asbr-al(config-if)#ip rip send version 1 2
asbr-al(config-if)#ip rip receive version 2
```
This configuration would only accept RIPv2 messages but send both v1 and v2 updates. Let’s examine the RIP configurations used on \texttt{asbr-a1} and \texttt{asbr-a2}:

\begin{verbatim}
asbr-a1#config t
Enter configuration commands, one per line. End with CNTL/Z.
asbr-a1(config)#router rip
asbr-a1(config-router)#version 2
asbr-a1(config-router)#network 12.0.0.0
asbr-a1(config-router)#network 192.168.160.0
asbr-a1(config-router)#network 192.168.161.0
asbr-a1(config-router)#network 192.168.191.0
asbr-a1(config-router)#

asbr-a2#config t
Enter configuration commands, one per line. End with CNTL/Z.
asbr-a2(config)#router rip
asbr-a2(config-router)#version 2
asbr-a2(config-router)#network 192.168.181.0
asbr-a2(config-router)#network 192.168.191.0
asbr-a2(config-router)#network 192.168.192.0
asbr-a2(config-router)#network 192.168.10.0
asbr-a2(config-router)#network 192.168.12.0
\end{verbatim}

After all the network advertisement statements have been added, RIP is up and running. Here is the IP route table from \texttt{asbr-a1}:

\begin{verbatim}
asbr-a1#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
        U - per-user static route, o - ODR
Gateway of last resort is not set
R  192.168.12.0/24 [120/1] via 192.168.191.19, 00:00:26, Ethernet0
12.0.0.0/30 is subnetted, 1 subnets
 C    12.14.116.4 is directly connected, Serial1

asbr-a1#\end{verbatim}

IOS allows you to adjust how the RIP process behaves and interprets route advertisements.

---

**Note** - RIP version 1 does not send subnet masks in its update messages. So, networks are announced using the address space’s classful root. Routers then apply their locally configured subnet mask to interpret the address space. RIP version 2 sends both the network address and mask in its updates. However, RIPv2 summarizes at network boundaries, announcing only the classful root. RIPv2 uses the subnet masks on the corresponding interfaces to construct the subnet mask elements of its update messages. If you use classless addressing with a large classful space, this can present problems if
discontinuous subnets are used. The RIP subprocess command `<no auto-summary>` disables network summarization.

### Adjusting Distance and Metrics

To adjust the administrative distance applied to all route advertisements received by the RIP process, use the router subcommand `<distance [10-255]>`.

If you need to change a route’s metric, a standard access list is used in combination with the RIP subcommand `<offset-list [acl] [in/out] [metric] [interface type/slot/port]>`. The Albany network has three access points to the 192.168.192.64 /26 network: asbr-a2, asbr-a3 and asbr-a4. In order to make all three routes have the same cost, the advertisement for the 192.168.192.64 network must be adjusted on asbr-a2. First, a standard IP access-list is created:

```
asbr-a2(config)#access-list 1 permit 192.168.192.64 0.0.0.0
```

Then, in the RIP subprocess, the metric adjustment is applied. This adjustment forces the metric for IP address 192.168.192.64 from 1 to 3 only on announcements sent to asbr-a1 from asbr-a2:

```
asbr-a2(config-router)#offset-list 1 out 2 ethernet 0
```

To verify that the adjusted route metric is being announced, enable RIP debugging with the `<debug ip rip>` privileged EXEC command and examine the RIP update message.

```
RIP: sending v2 update to 224.0.0.9 via Ethernet0 (192.168.191.19)
  192.168.10.0/24  ->  0.0.0.0, metric 1, tag 0
  192.168.12.0/24  ->  0.0.0.0, metric 1, tag 0
  192.168.20.0/24  ->  0.0.0.0, metric 16, tag 0
  192.168.192.64/26 ->  0.0.0.0, metric 3, tag 0
  192.168.181.0/24  ->  0.0.0.0, metric 1, tag 0
```

### Adjusting RIP Process Timers

RIP’s default is to send a message update every 30 seconds (the router’s entire routing table) whether there is a change or not. The IOS RIP updates can have up to a 4.5 second variation in frequency; this is to prevent all of the routing tables from sending updates at the same time.

In addition to the `[update]` timer, there is the `[invalid]` timer, which sets how long a route can remain in the routing table without being updated. If the `[invalid]` timer expires, the route is set to a metric of 16 and the `[flush]` is started. Upon expiration, the route is purged from the table.

IOS also supports a `[hold down]` timer, which is triggered when an update has a different metric than one previously recorded. These timers can be adjusted with the RIP routing subprocess command `<timers basic [update] [invalid] [holddown] [flush]>` to reduce message update times. In our present example, because the RIP process is only used for advertising routes to the ISP and changes in the topology are rare, setting the timers for updates that are more infrequent is preferable over the default timers. All values are set in seconds:

```
asbr-a1(config-router)#timers basic 120 240 180 300
```
Note - This option is also available with IGRP.

In situations where you have networks that consist of nonbroadcast media, or where it is desirable only to exchange routing updates between two hosts connected over broadcast-supported media, IOS provides the router subprocess command `<neighbor [ip address]>`. The ISP provider for the Albany network uses Frame Relay over HDLC, so a neighbor must be specified:

```
    asbr-a1(config-router)#neighbor 12.14.116.66
```

Configuring Authentication

When you employ any routing protocol in a publicly accessible environment, it is wise to use authentication to verify with whom you are exchanging route advertisements. RIP version 2 supports clear text and MD5 authentication. Both routers must exchange the same password regardless of the authentication method. To use authentication, each router needs to create a `<key chain>` and key and then enable authentication on the appropriate interfaces. To set up a key chain, use the following:

```
    asbr-a1(config)#key chain test
    asbr-a1(config-keychain)#key 1
    asbr-a1(config-keychain)#key-string atestkey
    asbr-a1(config-keychain)#accept-lifetime 12:00:00 31 dec 1998 infinite
    asbr-a1(config-keychain)#send-lifetime 12:00:00 31 dec 1998 infinite
```

The example above creates a key-chain called "test" and a single key with the password "atestkey". A key chain containing a key with "atestkey" as a password must also be created on all routers that `asbr-a1` wishes to exchange route announcements. After the keys are in place, the interface is configured:

```
    asbr-a1(config)# interface s1
    asbr-a1(config-if)#ip rip authentication key-chain test
    asbr-a1(config-if)#ip rip authentication mode md5
```

Note - The `<neighbor>` command is a general routing protocol configuration command that is usable with all IGP routing protocols.

Note - This key generation process is also used to generate MD5 keys for use with OSPF MD5 authentication.

Examining the RIP Process

To verify RIP operation and troubleshoot possible configuration problems, the privileged EXEC commands `<show ip protocols>`, `<debug ip rip>`, and `<debug ip rip events>` can be used.
<show ip protocols> provides status information on the process state. <debug ip rip events> displays process event messages about RIP message updates. <debug ip rip> displays the contents of RIP update message being sent. To disable the debug commands, use the <no> command in front of the debug command or <no debug all>.

**Using IGRP/EIGRP**

Global configuration commands:

- `<router igrp [process id]>`
- `<router eigrp [process id]>`
- `<appletalk routing eigrp [process id]>`
- `<ipx router eigrp [process id]>`

Global router subprocess commands:

- `<neighbor>`
- `<timers basic [update] [invalid] [holddown] [flush]>`
- `<distance>`
- `<no auto-summary>`
- `<network [IP address|ip network number]>`
- `<offset-list [acl] [in/out] [metric] [interface type/slot/port]>`
- `<distribute-sap-list [IPX access-list number] [in|out]>`
- `<variance [multiplier]>`
- `<traffic-share [balanced|min]>`

Global interface subprocess commands:

- `<ip eigrp-bandwidth-percent [percent]>`
- `<appletalk eigrp-bandwidth-percent [percent]>`
- `<ipx bandwidth-percent eigrp [process id] [percent]>`
- `<ip summary-address eigrp [ip address] [netmask]>`

Display commands:

- `<show ip protocols>`
- `<show ip eigrp topology>`
- `<show ip eigrp traffic>`
- `<show ip eigrp neighbors [options]>`
- `<show ip eigrp traffic>`

Interior Gateway Routing Protocol (IGRP) and Enhanced Interior Gateway Routing Protocol (EIGRP) are Cisco Systems’ proprietary dynamic routing protocols. IGRP, like RIP, is a distance vector protocol that broadcasts its router table out of all its interfaces at regular (adjustable) intervals. Unlike RIP, IGRP supports unequal path traffic sharing, uses its own transport protocol (similar to UDP) to send its update messages, and supports a network diameter of 224 hops compared to RIP’s 15.

Two elements make IGRP a significant improvement over RIP. The first is its support of five routing metrics compared to RIP’s single hop-count metric:

- Bandwidth (K1)
- Delay (K2)
The second is its significantly faster convergence time, which uses "flash" updates to immediately propagate changes to the network topology whenever they occur.

EIGRP, as I'm sure you have guessed, is an enhancement to Cisco's IGRP protocol and represents a change in the routing algorithm. IGRP, as a true distance vector protocol, sends updates as tuples or distance pairs containing the network (vector) and path cost (distance). EIGRP uses the DUAL algorithm (Diffusing Update Algorithm), developed at SRI International. DUAL is considered a hybrid protocol because it employs the distance vector and route metric basis established for IGRP, but also uses link state information collection and announcement features. EIGRP (like OSPF) establishes relationships with other EIGRP neighbors, through the use of a "hello" protocol. After a neighbor relationship has been established, routing information is exchanged via unicast or multicast transport—using EIGRP RTP (Reliable Transport Protocol)—depending on the type of messages being exchanged. DUAL, unlike distance vector and link state protocols, uses a system of diffusing calculations shared across multiple routers to compute routing calculations. This is a different approach than what is used with distance vector and link state protocols, where each router performs its own route calculations.

**Note** - IGRP and EIGRP, straight out of the box, determine a route's metric using the additive sum of all the segment delays plus the bandwidth value of the slowest interface in the path. In a network comprised of same-bandwidth links, hop-count is used to determine desirability. In mixed media networks (with links of differing bandwidth), the lowest metric path is chosen. IGRP and EIGRP calculate an interface's bandwidth using the interface's <bandwidth> setting in KBPS divided by 10 to the 7th power or 10,000,000. All five metrics can be enabled using <metric weight [tos] [k1] [k2] [k3] [k4] [k5]>. This is not, however, advisable. The defaults are tos=0 k1=1 k2=0 k3=1 k4=0 k5=0. If k5 is set to 1, reliability and load will be used in addition to bandwidth and delay for calculating the route metric.

EIGRP's other enhancements include the following:

- Multiprotocol routing support (IP, IPX, and AppleTalk)
- CIDR support for classless summarization
- Fast convergence and partial updates, so only changes in topology are announced and only routers affected by a network changes are forced to re-compute their routing tables
- VLSM support
From a configuration perspective, IGRP and EIGRP are quite similar, so our examples will use EIGRP. An indication will be made if an option is only available with EIGRP. One difference should be noted outright. After convergence is achieved, EIGRP will only send out routing updates when topology changes occur. IGRP, on the other hand, uses timed, regular updates (regardless of changes in topology) and flash updates (when changes occur between scheduled updates). To adjust IGRP timers, use the router configuration subcommand `<timers basic [update] [invalid] [holddown] [flush]>`. IGRP has a 90-second default update interval, 270-second invalid timer, and 630-second flush timer. Like RIP, if you adjust the times on one router, you have to adjust them on all the routers.

Basic EIGRP (IGRP) Configuration

To get started, we will configure EIGRP on asbr-a2. The EIGRP process is started with the global configuration EXEC command `<router eigrp [process id]>`, and IGRP is started with `<router igrp [process id]>`. The process ID must be a number between 1–65,535. Multiple EIGRP and IGRP processes can operate on the same router. This permits you to segregate routing policies inside the internetwork and then use redistribution to announce between processes (a concept often used in large corporate networks where network management responsibility is decentralized). The process ID is used to identify the separate routing processes, so the process ID must be the same for each router that participates in the same EIGRP/IGRP process. Because EIGRP/IGRP is dependent on the process ID (PID) being uniform across all the routers, the PID is often referred to as an AS number. This does not mean you need an AS number to EIGRP/IGRP; any number will do, it just needs to be consistent across all the routers that are meant to exchange routing information:

```
asbr-a2(config)#router eigrp 99
```

After the process is enabled, IP network announcements are entered using the `<network [ip address|ip network number]>` router configuration subcommand. Announcements are entered using the network number or IP address of the interface:

```
asbr-a2(config-router)#network 192.168.191.20
asbr-a2(config-router)#network 192.168.12.0
asbr-a2(config-router)#network 192.168.192.0
asbr-a2(config-router)#network 192.168.10.0
asbr-a2(config-router)#network 192.168.181.0
asbr-a2(config-router)#network 192.168.0.0
asbr-a2(config-router)#no auto-summary
```

No matter how the announcement entry is made, it will appear in the router's configuration as classful network address summaries:

```
router eigrp 99
network 192.168.191.0
network 192.168.192.0
network 192.168.10.0
network 192.168.181.0
network 192.168.0.0
no auto-summary
```

How the announcements are displayed in the configuration has no bearing on how they will be announced; that is determined by which protocol is used. If IGRP is in use, the networks will be announced classfully, depending on the subnet mask of the router for mask interpretation. Remember
that when you subnet class A and B address spaces, you must use a consistent subnet mask for the entire address space. If EIGRP is in use, the network will be announced along with their subnet masks. EIGRP by default uses classful auto-summarization; this should be disabled if VLSM and a single classful space is being used for addressing. Disabling classful auto-summarization can be accomplished with the \texttt{<no auto-summary>} router configuration subcommand. The configuration outlined thus far represents the minimum required configuration to enable IP EIGRP/IGRP routing.

\textbf{Note -} IGRP/EIGRP will only send routing information updates on interfaces with addresses that correspond to those announced in its configuration. In situations where a large classful address space is used, this may not be desirable, especially if multiple IGRP/EIGRP processes are announcing different segments of the network that have the same classful root address (for example, 172.16.0.0 is a class B root address). In these situations, the \texttt{<passive-interface>} command can be used to suppress routing message updates from being sent out of an interface as part of the particular process.

Before we look at the EIGRP specific configuration options, a quick word about default routing: IGRP and EIGRP both use the \texttt{<ip default network [ip network number]>} global configuration EXEC variable to set and distribute default network information. This setting can be set on one gateway router and redistributed to the rest of the members of the IGRP/EIGRP routing process, or it can be set on each router in the process. The idea behind the default-network is to send external traffic not to a specific gateway, but rather to a default network where the gateway can be determined (in the case of multiple network exit points) by the locally attached router. It should be noted that a default gateway setting using the \texttt{<ip route 0.0.0.0 0.0.0.0 [ip remote gateway]>} may be required on the actual outbound gateway that is attached to the default network.

**EIGRP Specific Configuration Options**

EIGRP’s support of IP summary-address announcements has an impact on CIDR supernets. Although it is recommended that you disable classful summarizations or avoid classful summarization altogether, if you use CIDR supernetting with a class C address space (not uncommon with sites that use small regional ISPs for Internet access), you will need to use summary announcements in order to have EIGRP announce the superset correctly. IP summarizations are configured using the interface configuration subcommand \texttt{<ip summary-address eigrp>}. Here is an example: The ISP for the Albany network (refer to Figure 10.1) has provided four class C addresses to support a network expansion that will be attached to \texttt{asbr-2a}. The addresses were assigned as a single /22:

\begin{verbatim}
192.168.4.0 /22 = 192.168.4.0, 192.168.5.0, 192.168.6.0, 192.168.7.0
\end{verbatim}

Moreover, we want to maintain this address provisioning. Under normal operation, EIGRP would announce each class C network as a separate route entry, even though the address block is being used as a contiguous CIDR superblock. By using an aggregate summary address, the 192.168.4.0 /22 network is announced correctly as a supernet:

\begin{verbatim}
 asbr-a2(config)#int e2/0
 asbr-a2(config-if)#ip address 192.168.4.1 255.255.252.0
 asbr-a2(config-if)#ip summary-address eigrp 99 192.168.4.1 255.255.252.0
\end{verbatim}
Using the `<show ip protocol>` command we can see the summarization listed (notice that auto-
summarization has been disabled):

Routing Protocol is "eigrp 99"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Default networks flagged in outgoing updates
Default networks accepted from incoming updates
EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
EIGRP maximum hopcount 100
EIGRP maximum metric variance 1
Redistributing: eigrp 99
Automatic network summarization is not in effect
Address Summarization:
  192.168.4.0/22 for Ethernet0/0
Routing for Networks:
  192.168.191.0
  192.168.12.0
  192.168.192.0
  192.168.10.0
  192.168.181.0
  192.168.0.0
  192.168.4.0
Routing Information Sources:
  Gateway     Distance   Last Update
  Distance: internal 90 external 170

One of EIGRP’s biggest enhancements is the support of IPX and AppleTalk routing. Enabling EIGRP
AppleTalk routing is done using the `<appletalk routing eigrp [process id]>` command. Let’s
look at a configuration example:

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# appletalk routing eigrp 99
```

To enable EIGRP IPX routing, the global configuration command `<ipx router eigrp [process
id]>` is used. When enabled, IPX networks are added using an IPX version of the EIGRP `<network
[ipx network address]>` or `<network all>` command:

```
Router(config)#ipx router eigrp 99
Router(config-ipx-router)# networks all
```

IPX EIGRP functionality is similar to IP EIGRP; it supports redistribution of IPX, RIP and NLSP,
route filtering, and IPX-specific adjustments such as variable SAP announcements (interface
subcommand `<ipx sap-incremental-eigrp>`) and SAP filtering with the IPX EIGRP subcommand
 `<distribute-sap-list [IPX access-list number] [in|out]>`. Each of the protocol-specific
EIGRP processes run distinctly, so different PID should be used when configuring the separate
instances.

**Enabling Load Sharing and Balancing with EIGRP/IGRP**

EIGRP is particularly sensitive to bandwidth consumption when it sends routing update information.
The amount of bandwidth that can be used to send updates is adjustable (for all EIGRP protocol
implementations). The bandwidth allocation for EIGRP is set as an interface configuration
subcommand:

- `<ip eigrp-bandwidth-percent [percent]>`
- `<appletalk eigrp-bandwidth-percent [percent]>`
- `<ipx bandwidth-percent eigrp [process id] [percent]>`

IGRP and EIGRP provide the capability to distribute network traffic load over links of unequal costs for load balancing. IGRP supports asymmetric load balancing over four unequal-cost paths. EIGRP supports up to six. The router configuration subcommand `<variance [multiplier]>` is used to enable this feature, and its default setting is 1. The variance value is used to recompute the metrics of routes of lesser value than that of the primary route. For example, if the variance was set to 3, any route with a metric 3 times greater (the lower the number, the better the metric) will be used to forward traffic.

There is also a way to balance the traffic by using the router configuration subcommand `<traffic-share [balanced|min]>`. This configuration helps the router determine if the traffic should be balanced (default) or should favor the lowest path cost first.

**Adjusting Administrative Distances**

EIGRP and IGRP handle metrics the same way, but compute administrative distances differently. IGRP uses a single administrative distance of 100. This value is adjustable using the `<distance [1-255]>` router configuration subcommand. EIGRP uses three administrative distances:

- **Internal distance** is used for routes announced from the EIGRP routing process, and its default setting is 90. The `<distance [1-255]>` subcommand will adjust this value to set which route will be preferred if multiple routing protocols are providing the same announcement.

- **Summary distance** is used for routes generated by summary addresses statements, and it has a default of 5.

- **External distance** is used when you redistribute routes from OSPF, and its default is 170. The values are adjustable using the `<distance eigrp [internal distance] [summary distance] [external distance]>` EIGRP configuration subcommand.

**Monitoring IGRP and EIGRP**

The IOS provides various informational commands to help you monitor your EIGRP processes, but not as many for IGRP. To display basic summary information, use the `<show ip protocols>` command:

```
Routing Protocol is "eigrp 99"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Default networks flagged in outgoing updates
Default networks accepted from incoming updates
EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
EIGRP maximum hopcount 100
EIGRP maximum metric variance 1
Redistributing: eigrp 99, igrp 99
Automatic network summarization is not in effect
Address Summarization:
```
192.168.4.0/22 for Ethernet0/0

Routing for Networks:
192.168.191.0
192.168.12.0
192.168.192.0  192.168.10.0

There are four `<show ip eigrp>` subcommands:

- `<show ip eigrp neighbors [options]>` provides operational information on each of the neighbors the router knows about:
  
  asbr-a2#show ip eigrp neighbors ?
  <1-65535> AS Number
  Ethernet  IEEE 802.3
  Loopback  Loopback interface
  Null     Null interface
  Serial   Serial
  detail   Show detailed peer information  <cr>

- The neighbor’s IP address and accessible interface are listed, along with the period since the router has heard from the neighbor (uptime).

- `<show ip eigrp interfaces>` and `<show ip eigrp traffic>` provide EIGRP traffic statistics:
  
  asbr-a2#show ip eigrp interfaces ?
  <1-65535> AS Number
  Ethernet  IEEE 802.3
  Loopback  Loopback interface
  Null     Null interface
  Serial   Serial
  detail   Show detailed peer information  <cr>

  asbr-a2#show ip eigrp traffic ?
  <1-65535> AS Number
  <cr>

- `<show ip eigrp topology>` provides summary information on the EIGRP topology information source database:
  
  asbr-a2#show ip eigrp topology ?
  <1-65535> AS Number
  A.B.C.D     Network to display information about
  active      Show only active entries
  all-links   Show all links in topology table
  pending     Show only entries pending transmission
  summary     Show a summary of the topology table
  zero-successors Show only zero successor entries  <cr>

- IOS also supports various debugging options for IGRP and EIGRP:
  
  asbr-a2#debug ip igrp
  events    IGRP protocol events
  transactions IGRP protocol transactions

  asbr-a2#debug ip eigrp
  <1-65535> AS number
Using OSPF

Display commands:

<show ip protocols>
<show ip ospf [local process id]>
<show ip ospf neighbor>
<show ip ospf database>
<show ip ospf virtual-links>

Interface configuration commands:

<ip ospf cost>
<ip ospf demand-circuit>
<ip ospf network [broadcast or non-broadcast]>
<ip ospf network point-to-multipoint>
<ip ospf authentication-key [password]>
<ip ospf priority [0-255]>

---

**Note** - As with all debugging commands, use these commands with caution. To disable a debugging command, use <no> preceding the command or use <no debug all>.

---

OSPF configuration subprocess commands:

<router ospf [local process id]>
<network <IP address/network number> <reverse network mask> <area id>
<area [area id] range [ip network address] [standard network mask]>
<distance [10-255]>
<distance ospf [external/inter-area/inter-area] [10-255]>
<area [area id] stub>
<area [area id] nssa>
<area [transit area id] virtual-link [router id]>
<redistribute [process] subnets>
<no ospf auto-cost>
<no ospf auto-cost-determination>
<passive-interface><neighbor [ip address] [priority (0-255)]>

Additional commands:

<ip classless><ip subnet-zero>

Implementing OSPF requires some up-front planning. Unlike other IGPs, the network topology plays a role in how OSPF functions. OSPF divides a large internetwork or AS into a collection of centrally connected hierarchical segments called *areas*. Each area should consist of no more than 50 routers, and must have one router that provides access to the backbone area. The *backbone area* (0.0.0.0) is used for exchanging Link State Announcements (LSAs) and acts as a transit network for all inter-area datagram traffic. All IP traffic exchanged between areas traverses the backbone network.
OSPF Router Type Definitions

You should recall that there are four classifications of OSPF routers:

- **Backbone Routers**—These routers maintain complete routing information for all the networks (all the areas or domains) that are connected to the backbone.

- **Area Border Routers (ABRs)**—These routers connect one or more areas to the backbone area, and only maintain information about the backbone and the areas they are attached to. Any router that has interfaces attached to the backbone and at least one area is an ABR.

- **Internal Routers (IRs)**—These routers are only involved in intra-area routing. IRs only contain information about the area they operate in. All extra-area traffic is forwarded to the area border router because only summaries of other areas within the network are maintained locally.

- **Autonomous System Boundary Routers (ASBRs)**—These routers are used to exchange routing information between OSPF and EGP. Any OSPF router that redistributes routing information is considered an ASBR.

Network topology plays a role in OSPF performance because each router maintains a database on the state of the network. OSPF was designed to be implemented in large-scale internetworks with 30 or more routers.

This does not mean OSPF will not function properly in small network environments; on the contrary, it will work well. Actually, in situations where a standards-based routing protocol is required, OSPF is a significantly better choice than RIP, especially when it comes to speed and network utilization.

Ideally, the backbone routers should bear the brunt of the processing by constructing a complete topology map from the LSAs sent by the ABRs that have established adjacencies with the IR. Therefore, the more network topology and addressing lends itself to subdivision and summarization, the more efficient and scalable OSPF becomes. OSPF's tightly structured model, where each router class performs a function, gives OSPF the capability to efficiently manage the route processing load efficiently.

**Configuring OSPF**

Configuring OSPF is similar to setting up IGRP and EIGRP. To get the OSPF process started, it requires a local process ID in the range of 1 to 65,535 (just like IGRP and EIGRP). In addition, like IGRP and EIGRP, it is possible to run multiple OSPF instances. This should only be done in special circumstances and generally avoided. Each OSPF process builds its own topological database and has its own routing algorithm process, so running multiple OSPF instances adds unneeded load to the router.

Where OSPF differs from a configuration standpoint is in the way the network announcements are
entered. OSPF is a true classless protocol. When configuring RIPv2, EIGRP, and so on, networks are entered (and listed in the configuration) using their natural classful boundaries. OSPF provides the capability to enter the network announcements in their classless form, along with a network mask to interpret the network address. One thing about the netmask: It is entered in reverse form, just like access lists. Also, like access lists, the same calculation formula is used to calculate the reverse mask:

\[
\begin{align*}
255.255.255.255 & - 255.255.252.0 \\
= & 0.0.3.255
\end{align*}
\]

The capability to enter the network and mask gives you a lot of flexibility in terms of how networks are announced. With other IGP protocols, network summarization can present problems, especially in large internetworks where classless addressing is used extensively. To address this problem, summarization is disabled with the router configuration subcommand \texttt{no auto-summary}. OSPF does not summarize unless you configure it to.

In large LANs where subnetting is used and the possibility of discontinuous subnets exists, auto-summarization can be a pain. In a large internetwork, however, summarization is an efficiency gain, since summarization reduces the number of routes required in the routing table. On large internetworks, whenever possible, you want to deploy your network address space using classless boundaries that can be treated as CIDR supernets. This makes it easy to use CIDR addressing to summarize network announcements. Establishing an effective network addressing hierarchy is essential for OSPF to function efficiently.

\textbf{Note} - A situation where multiple OSPF processes might be desirable is when the router is acting as gateway between two separate internetworks. A common example would be if OSPF was being used as an EGP by a network service provider and as an IGP by a client. The client gateway router would run both an internal and external OSPF process. The router would construct its routing table using announcements from both processes.

With this in mind, the test network has been readdressed to take advantage of address summarization. Figure 10.2, in addition to illustrating the network’s readdressing, also shows the OSPF area partitioning. OSPF area IDs can be any number between 1 and 4,294,967,295 or a variation of the IP network address in dotted quad form. The Boston LAN networks use area ID 172.16.0.0 and the Albany LAN networks use area ID 192.168.0.0.

To get our feet wet, let’s configure OSPF on asbr-b1 and asbr-a1. The network’s topology, as well as the IP addressing hierarchy, has an effect on how well OSPF functions. Because all inter-area traffic must flow across the backbone network, it makes sense to have all the WAN and backbone LAN networks reside in area 0.0.0.0 and to treat each of the respective LANs as their own area. When you are designing the network topology and/or partitioning the network into areas, it is a good idea to list which contains each of the routers and lists which router interface is in which area. This helps you visualize the traffic flow.

\textbf{Figure 10.2}
The testnet using CIDR summarization.
<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>e0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>bri0</td>
<td>192.168.0.96</td>
</tr>
<tr>
<td>lo1</td>
<td>192.168.0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>e0/0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>e0/1</td>
<td>192.168.0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>e0/2</td>
<td>192.168.0.0</td>
</tr>
<tr>
<td>e0/3</td>
<td>192.168.0.0</td>
</tr>
<tr>
<td>fe1/0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>s2/0</td>
<td>0.0.0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>e0/0</td>
<td>192.168.0.0</td>
</tr>
<tr>
<td>e0/1</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>fe0/1</td>
<td>192.168.9.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1/0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>e1/1</td>
<td>192.168.9.0</td>
</tr>
<tr>
<td>fe0/0</td>
<td>192.168.0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Int</td>
<td>192.168.0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Int</td>
<td>192.168.0.0</td>
</tr>
</tbody>
</table>

Table 10.3 Router Interface Summary for the OSPF Networks in Figure 10.2
For OSPF, at least one interface must be configured on the router in order to enable the process. The OSPF process is enabled using the configuration EXEC command `<router ospf [local process id]>`. The PID is local only to the router, and OSPF uses the largest IP address configured on the router as its PID. If this interface is removed or shut down, the OSPF process will select a new ID and resend all its routing information. There is no way to specify which interface to use as the PID. The IOS will, however, choose a loopback interface over any other regardless of the IP size. One method you can use to control the OSPF process IDs is to assign addresses from the unregistered space 192.168.254.0 /24 to loopback interfaces on each of the routers in the process. Each router would get a different incremental address, and this way you could directly associate an OSPF ID with each router in your network:

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>s0/1</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>fe1/0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>e4/0</td>
<td>172.16.0.0</td>
</tr>
<tr>
<td>e4/1</td>
<td>172.16.0.0</td>
</tr>
<tr>
<td>e4/2</td>
<td>172.16.0.0</td>
</tr>
<tr>
<td>e4/3</td>
<td>172.16.0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>s0/0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>fe2/0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>e1/0</td>
<td>172.16.0.0</td>
</tr>
<tr>
<td>e1/1</td>
<td>172.16.0.0</td>
</tr>
<tr>
<td>e1/2</td>
<td>172.16.0.0</td>
</tr>
<tr>
<td>e1/3</td>
<td>172.16.0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>e0/0</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>e0/1</td>
<td>172.16.0.0</td>
</tr>
<tr>
<td>async1</td>
<td>172.16.0.0</td>
</tr>
<tr>
<td>async2</td>
<td>172.16.0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Int</td>
<td>172.16.0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interface</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Int</td>
<td>172.16.0.0</td>
</tr>
</tbody>
</table>
asbr-a1#config t
asbr-b2(config)# int loopback
1asbr-a2(config-if)#ip address ip address 192.168.254.1 255.255.255.0

Entering OSPF Route Announcements

OSPF network entries are made with the OSPF configuration subcommand <network <IP address/network number> <network mask> <area id>. As mentioned before, with OSPF there is more than one way to configure network announcement entries. You will find that the best method is to use the network address and its corresponding mask. This way only the network space you want announced gets announced. It also forces you look at how you are using CIDR and VLSM beforehand, to ensure that you have your addressing straight. Here is the configuration for asbr-b2 using this approach:

asbr-b2#config t
Enter configuration commands, one per line. End with CNTL/Z.
asbr-b2(config)#router ospf 202
asbr-b2(config-router)#network 192.168.0.0 0.0.0.3 area 0.0.0.0
asbr-b2(config-router)#network 172.16.1.0 0.0.0.255 area 172.16.0.0
asbr-b2(config-router)#network 172.16.2.192 0.0.0.63 area 172.16.0.0
asbr-b2(config-router)#network 172.16.2.128 0.0.0.63 area 172.16.0.0
asbr-b2(config-router)#network 172.16.2.64 0.0.0.63 area 172.16.0.0
asbr-b2(config-router)#network 172.16.3.0 0.0.0.255 area 0.0.0.0

An alternative to this is to use the network addresses of the connected interfaces. When using this approach, the netmask/wild card bits are set to 0.0.0.0. Router asbr-a1 is configured using this approach:

asbr-a1(config)#router ospf 57
asbr-a1(config-router)#network 192.168.3.19 0.0.0.0 area 0.0.0.0
asbr-a1(config-router)#network 192.168.0.1 0.0.0.0 area 0.0.0.0
asbr-a1(config-router)#network 192.168.0.97 0.0.0.0 area 192.168.0.0

Now that both routers are configured, let’s check the routing table on asbr-a1 to make sure we are in fact getting announcements from asbr-b2:

asbr-a1#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
      U - per-user static route, o - ODR
Gateway of last resort is 12.14.116.6 to network 0.0.0.0

192.168.160.0/30 is subnetted, 1 subnets
C    192.168.160.96 is directly connected, Ethernet0/0
172.16.0.0/32 is subnetted, 4 subnets
 0 IA  172.16.2.129 [110/65] via 192.168.0.2, 00:00:51, Serial10/0
 0 IA  172.16.2.193 [110/65] via 192.168.0.2, 00:00:51, Serial10/0
 0 IA  172.16.1.1 [110/65] via 192.168.0.2, 00:00:56, Serial10/0
 0 IA  172.16.2.65 [110/65] via 192.168.0.2, 00:00:41, Serial10/0
C    192.168.191.0/24 is directly connected, Ethernet0/1
 172.16.0.0/30 is subnetted, 1 subnets
C    192.168.0.0 is directly connected, Serial10/0
12.0.0.0/30 is subnetted, 1 subnets
According to the routing table, everything is fine; the 172.16.2.x /26 and 172.16.1.0 /24 network spaces are being announced as inter-area routes just as they are supposed to be.

As with RIP, EIGRP, and so on, it is possible to use just a natural address mask. This approach saves typing, but should be used with caution. If the address space is being used in different OSPF areas, routes will not be announced correctly. Let’s reconfigure the asbr-b1 router to see what can happen and how to avoid these problems.

Because the Boston network uses only address space from 117.12.0.0 /16 and 192.168.0.0 /24, we can save ourselves a lot of typing by just entering networks classfully:

```
asbr-b2(config)#router ospf 202
asbr-b2(config-router)#network 172.16.0.0 0.0.255.255 area 0.0.0.0
asbr-b2(config-router)#network 192.168.0.0 0.0.0.255 area 0.0.0.0
asbr-b2(config-router)#network 172.16.0.0 0.0.255.255 area 172.16.0.0
```

Again, let’s look at the asbr-a1 routing table:

```
asbr-a1>sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR
Gateway of last resort is 12.14.116.6 to network 0.0.0.0

192.168.160.0/30 is subnetted, 1 subnets
 C  192.168.160.96 is directly connected, Ethernet0/0
172.16.0.0/32 is subnetted, 4 subnets
 O  172.16.2.129 [110/65] via 192.168.0.2, 00:05:05, Serial0/0
 O  172.16.2.193 [110/65] via 192.168.0.2, 00:05:05, Serial0/0
 O  172.16.1.1 [110/65] via 192.168.0.2, 00:05:05, Serial0/0
 O  172.16.2.65 [110/65] via 192.168.0.2, 00:05:05, Serial0/0
 C  192.168.191.0/24 is directly connected, Ethernet0/1
192.168.0.0/30 is subnetted, 1 subnets
 C  192.168.0.0 is directly connected, Serial0/0
12.0.0.0/30 is subnetted, 1 subnets
 C  12.14.116.4 is directly connected, Serial1/0
 S*  0.0.0.0/0 [1/0] via 12.14.116.6
```

Do you see anything wrong? (Refer back to the previous route table.) Now, all the networks are being announced as area 0.0.0.0. Let’s look at the running-config on asbr-b1:

```
! router ospf 202
   network 172.16.0.0 0.0.255.255 area 0.0.0.0
   network 192.168.0.0 0.0.0.255 area 0.0.0.0
!
ip classless
!
line con 0 line 33 48
```
The second address and area entry was never accepted because it was considered part of the first announcement entry. If you change the entry to

```
asbr-b2(config)#router ospf 202
asbr-b2(config-router)#network 192.0.0.0 0.255.255.255 area 0.0.0.0
asbr-b2(config-router)#network 172.16.0.0 0.255.255.255 area 172.16.0.0
```

Then, all the 172.16.0.0 /16 networks appear as inter-area routes. This is not correct either because we want 172.16.3.0 /24 to be part of the backbone so we can have asbr-b3 function as an ABR. So, let’s add 172.16.3.0 /24 to the OSPF configuration:

```
asbr-b2(config-router)#network 172.16.3.0 0.0.0.255 area 0.0.0.0
% OSPF: "network 172.16.3.0 0.0.0.255 area 0.0.0.0" is ignored. It is a subset of a previous entry.
```

Again, OSPF sees the subnet as part of the broad 172.16.0.0 /16 network. So, route announcement entry is ignored. To avoid this problem, always start with the specific networks first and then use the broader network space entries:

```
asbr-b2(config-router)#network 172.16.3.0 0.0.0.255 area 0.0.0.0
asbr-b2(config-router)#network 192.168.0.0 0.0.0.255 area 0.0.0.0
asbr-b2(config-router)#network 172.16.0.0 0.0.255.255 area 172.16.0.0
```

To make sure that the correct interfaces are in the OSPF areas, you can use the user EXEC command <show ip ospf>:

```
sbr-b2#sh ip ospf
Routing Process "ospf 202" with ID 172.16.2.193
Supports only single TOS(TOS0) routes
It is an area border router
Summary Link update interval is 00:30:00 and the update due in 00:28:24
SPF schedule delay 5 secs, Hold time between two SPF's 10 secs
Number of DCbitless external LSA 0
Number of DoNotAge external LSA 0
Number of areas in this router is 2. 2 normal 0 stub 0 nssa
Area BACKBONE(0.0.0.0)
    Number of interfaces in this area is 2
    Area has no authentication
    SPF algorithm executed 3 times
    Area ranges are
    Link State Update Interval is 00:30:00 and due in 00:28:17
    Link State Age Interval is 00:20:00 and due in 00:18:10
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
Area 172.16.0.0
    Number of interfaces in this area is 4
    Area has no authentication
    SPF algorithm executed 2 times
    Area ranges are
    Link State Update Interval is 00:30:00 and due in 00:28:22
    Link State Age Interval is 00:20:00 and due in 00:18:22
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
asbr-b2#
```
This output tells us there are two interfaces in the backbone area and four interfaces in area 172.16.0.0, which is just how we configured it to be. As you can see, there is more than one way to configure announcements, and each has its own merits. The first two approaches are quite effective for most configurations. The third approach should be avoided unless you feel comfortable with address subnetting and IP masking.

**OSPF Address Summarization**

In the examples above, all the networks are announced individually. With OSPF, network summarization is an option permitted on ABRs. To configure route summarization on an ABR, the `<area>` OSPF subcommand is used. The `<area>` command is used for configuring several different options; this command is shown below. OSPF route summarization is limited to networks within the same OSPF area. To illustrate this, we are going to reconfigure asbr-la to summarize 117.16.2.x /26 network spaces as 117.16.2.0 /23. Currently, these routes appear on asbr-a1 as follows:

```
172.16.0.0/32 is subnetted, 4 subnets
O IA  172.16.2.129 [110/65] via 192.168.0.2, 00:24:23, Serial0/0
O IA  172.16.1.1/32 [110/65] via 192.168.0.2, 00:37:25, Serial0/0
O IA  172.16.2.193 [110/65] via 192.168.0.2, 00:24:23, Serial0/0
O IA  172.16.2.0/24 [110/65] via 192.168.0.2, 00:24:23, Serial0/0
```

After all the involved networks are configured to be announced, the `<area>` command is used to establish the summarization network range:

```
asbr-b2(config-router)#area 172.16.0.0 range 172.16.2.0 255.255.255.0
```

Now, all the 172.16.2.x /26 networks are summarized as one /24 network:

```
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
O IA  172.16.1.1/32 [110/65] via 192.168.0.2, 00:37:25, Serial0/0
O IA  172.16.2.0/24 [110/65] via 192.168.0.2, 00:03:09, Serial0/0
```

Summarization can be done with any classless or classful address range, provided it falls on the correct boundaries. Now, this may not seem like much of a savings on a small network, but on a large internetwork where you are subnetting several Class A or B sized address spaces, every route counts. As you can see from the examples above, OSPF does require some forethought in terms of addressing segmentation and area planning. Failure to do so will often result in problems down the road, which can range from the annoying to the disastrous.

**Adjusting OSPF Distance and Metrics**

OSPF has a default administrative distance of 110 and uses a single route metric: route cost. This information, along with the route’s age, route source, and accessible interface, is displayed as part of an OSPF-generated route entry in the route table. All regular OSPF routes are indicated with an "O"; an additional legend marker is to used indicate the type of OSPF route:

```
asbr-a1>sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
```
Gateway of last resort is 12.14.116.6 to network 0.0.0.0

192.168.160.0/30 is subnetted, 1 subnets
C 192.168.160.96 is directly connected, Ethernet0/0
172.16.0.0/32 is subnetted, 4 subnets
 O 172.16.2.129 [110/65] via 192.168.0.2, 00:05:05, Serial0/0
 O 172.16.2.193 [110/65] via 192.168.0.2, 00:05:05, Serial0/0
 O 172.16.1.1 [110/65] via 192.168.0.2, 00:05:05, Serial0/0
 O 172.16.2.65 [110/65] via 192.168.0.2, 00:05:05, Serial0/0
C 192.168.191.0/24 is directly connected, Ethernet0/1
192.168.0.0/30 is subnetted, 1 subnets

The default administrative distance for all OSPF routes (which is 110) can be adjusted using the router subcommand <distance [10-255]>. It is also possible to adjust the distances of external, inter-area, and intra-area routes using the OSPF configuration subcommand <distance ospf [external/inter-area/intra-area] [10-255]>.

**OSPF Route Types**

An "O" type route indicates that the route is directly connected to the backbone area. "IA" routers indicate that the route is an inter-area. Backbone and inter-area route costs reflect the cost of the entire route path. "E1" indicates that route is an external, type 1 route. Type 1 metrics show the total cost of route path between the source and destination. All directly connected external networks use type 1 metrics since the actual cost to reach the network is the cost of the internal path to the ASB router. The "E2" marker is used for type 2 routes. Redistributed routes are announced as type 2 metrics. The route cost associated with a type 2 metric only reflects the cost to the source of the external route.

You should recall from Chapter 8, "TCP/IP Dynamic Routing Protocols," that OSPF uses a structured scale for establishing interface costs based on bandwidth, (see Table 10.4).

**Table 10.4 OSPF Interface Bit Value and Cost**

<table>
<thead>
<tr>
<th>Network Bandwidth (in bits)</th>
<th>Bandwidth (in KBit/s)</th>
<th>Path Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>100,000,000</td>
<td>100,000 (Fast Ethernet)</td>
<td>1</td>
</tr>
<tr>
<td>16,000,000</td>
<td>16,000 (16Mb Token Ring)</td>
<td>6</td>
</tr>
<tr>
<td>10,000,000</td>
<td>10,000 (Ethernet)</td>
<td>10</td>
</tr>
<tr>
<td>4,000,000</td>
<td>4,000 (4Mb Token Ring)</td>
<td>25</td>
</tr>
<tr>
<td>1,544,000</td>
<td>1,544 (serial T1)</td>
<td>64</td>
</tr>
<tr>
<td>64,000</td>
<td>640 (single B-ISDN)</td>
<td>1562</td>
</tr>
<tr>
<td>56,000</td>
<td>560 (single T-</td>
<td>1786</td>
</tr>
</tbody>
</table>
The importance of setting the interface’s bandwidth correctly cannot be stressed enough. Setting this parameter correctly ensures that the route costs are properly calculated. The route cost is the sum of all interface costs in the route path, divided by 100,000,000. The sum total is then rounded to the highest number, yielding the route cost.

If your network uses interfaces that support bandwidth speeds above Fast Ethernet (in other words, Gigabit Ethernet, POS, Smart Trunk Interfaces), you might want to set OSPF interface costs manually:

- Disable OSPF auto-cost using `<no ospf auto-cost>` for IOS version 12.x and later, or `<no ospf auto-cost-determination>` with IOS 11.x.
- Set each interface’s cost with the interface configuration EXEC command `<ip ospf cost>`.

**Stub Networks**

You will also notice on the `<show ip route>` route type code legend OSPF NSSA Type 1 (N1) and Type 2 (N2) routes. NSSA stands for not so stubby area. An OSPF stub area is an area with one or more outbound access paths that only receive internal routing information and cannot redistribute external routing information. Traffic destined for networks external to the local internetwork or AS use a default route which is statically set or generated by the ASBR and distributed as part of the OSPF announcement (see the section "Route Control and Redistribution," for more details). Commonly, a stub area is a small office/home office (SOHO) or remote office router connected to the internetwork over a DDR connection.

To minimize the amount of routing information sent across the link, the area is configured as a stub. To configure an area as a stub on the ABR, the command `<area [area-id] stub no summary>` is used. The DDR BRI interface on `asbr-a1` is configured as an OSPF stub area:

```
router ospf 67
    network 192.168.0.97 0.0.0.0 area 192.168.0.96
    area 192.168.0.96 stub no-summary
```

An NSSA network is very similar. An NSSA area only has routing information about the internal internetwork provided by the OSPF process. The difference is that an NSSA area can redistribute external routes into the general OSPF routing policy, so they can be forwarded to the rest of the internetwork. If the remote router attached to `asbr-a1` was a remote office that was connected to the larger internetwork, we could reconfigure area 192.168.0.96 on `asbr-a1` as an NSSA area and have
the routing information redistributed into the OSPF process. The OSPF routing subcommand `<area
[area_id] nssa>` is used to designate an area as an NSSA:

```
asbr-a1(config-router)#area 192.168.0.96 nssa
```

The configuration on `ir-a2` looks like this:

```
rip 89
network 192.168.64.0
network 192.168.65.0
network 192.168.66.0
network 192.168.67.0

router ospf 34
network 192.168.0.98 0.0.0.0 area 192.168.0.96
area 192.168.0.96 nssaredistribute rip subnets
```

All the routes redistributed from the NSSA network will appear as NSSA type 2 routes (N2). When configuring an area as an NSSA, all the routers in the area must be configured to recognize the area as an NSSA.

**Adjusting OSPF’s Behavior**

It is not always possible to construct a network topology that allows for proper router/area segmentation. It is possible to connect an ABR across a non-area-zero transit area. A virtual link is established between a true ABR and the "virtual ABR" via the OSPF subprocess configuration command `<area [transit area id] virtual-link [router id]>`. Each side of the link uses the router id of the remote end. Let’s examine a virtual link configuration between `ir-a1` and `asbr-a2` (the `<show ip ospf>` command will display the router’s OSPF ID):

```
hostname ir-a1
!
router ospf 78
network 192.168.12.252 0.0.0.0 area 192.168.0.0
network 192.168.2.0 0.0.0.255 area 0.0.0.0
area 192.168.0.0 virtual-link 192.168.254.5

hostname asbr-a2
!
router ospf 78
network 192.168.0.5 0.0.0.0 area 0.0.0.0
network 192.168.1.19 0.0.0.255 area 0.0.0.0
network 192.168.4.1 0.0.0.0 area 192.168.0.0
network 192.168.6.1 0.0.0.0 area 192.168.0.0
network 192.168.5.0 0.0.0.266 area 192.168.0.0
network 192.168.7.1 0.0.0.63 area 192.168.0.0 area 192.168.0.0 virtual-link 1
```

OSPF, like RIP, EIGRP, and IGRP, supports passive mode on any interfaces. To configure an interface to only receive OSPF messages, use the general router configuration subprocess command `<passive-interface>`.

**OSPF Authentication**

OSPF supports both text password and MD5 key authentication to verify route message exchanges between routers. Authentication needs to be enabled as part of the OSPF process using the OSPF
configuration subcommand `<area authentication>` for clear text or `<area authentication >[message-digest]>` to use MD5 key authentication (see the section, "Configuring Authentication" for MD5 key generation instructions).

All the testnet area 0.0.0.0 routers use clear text authentication; the password is "quickkey". Let’s take a look at asbr-a1 and asbr-a2’s configurations:

```plaintext
hostname asbr-a1
!
interface Ethernet0
  ip address 192.168.1.19 255.255.255.0
  no ip directed-broadcast
  ip ospf authentication-key quickkey
!
interface Serial1
  ip address 192.168.0.1 255.255.255.0
  no ip directed-broadcast
  ip ospf authentication-key quickkey
!
router ospf 67
  network 192.168.0.97 0.0.0.0 area 192.168.0.96
  network 192.168.1.0 0.0.0.255 area 0.0.0.0
  network 192.168.0.0 0.0.0.3 area 0.0.0.0
  network 192.168.0.128 0.0.0.128 area 192.168.0.0
  area 0.0.0.0 authentication
  area 192.168.0.96 nssa
hostname asbr-a2
!
interface Ethernet0
  ip address 192.168.1.20 255.255.255.0
  ip ospf authentication-key quickkey
!
interface Serial0
  ip address 192.168.0.5 255.255.255.252
  ip ospf authentication-key quickkey
!
router ospf 78
  network 192.168.0.5 0.0.0.0 area 0.0.0.0
  network 192.168.4.1 0.0.0.0 area 192.168.0.0
  network 192.168.7.1 0.0.0.0 area 192.168.0.0
  network 192.168.6.1 0.0.0.0 area 192.168.0.0
  network 192.168.5.1 0.0.0.0 area 192.168.0.0
  network 192.168.1.20 0.0.0.0 area 0.0.0.0
  area 192.168.0.0 virtual-link 92.168.254.4 authentication-key quickkey
  area 0.0.0.0 authentication
```

After authentication has been enabled for an OSPF area, all message exchanges halt until all of the authentication keys have been set on the involved interfaces. Do not forget that all the keys have to be the same in an area.

**Designated and Backup Designated Router Election**

It is possible to configure which routers will be the designated and backup designated routers within an area by using the `<ip ospf priority [0-255]> interface command, the highest priority value winning the election. By default, the IOS sets all OSPF interfaces with a priority of 1. When configuring your router, set higher priority values on the router interface you want to act as DR and BDR. It is also possible to exempt an interface from the DR/BDR election by setting its priority to
zero. This, like most of the configuration options available with OSPF, requires planning. Simply changing a router's interface will not make it the areas DR or BDR if they already exist. Only a re-election will make the change. In situations where a DR and BDR already exist and you want to force an election, simply shut down the DR/BDR interface momentarily; this will force an election which will be won by your re-prioritized interface.

---

**Note** - The use of virtual interfaces is highly discouraged and generally reflects poor planning on the part of the network designer. It was included only to provide transitional capability, not features.

---

**Using OSPF with Nonbroadcast and DDR Networks**

In Chapter 8, "TCP/IP Dynamic Routing Protocols," we reviewed how OSPF classifies connections into three categories:

- Broadcast
- Point-to-point
- Nonbroadcast multiaccess networks

When you use a transport medium that does not support broadcasts (Frame Relay being the most common type), the routers involved need to have a means to exchange OSPF messages. To get OSPF messages to exchange properly, you need to configure the interface to conform to the type of network connection (point-to-point, multipoint, or point-to-multipoint) you have in place. If you have a point-to-point or multipoint Frame Relay connection, you can configure the interface as a broadcast or nonbroadcast network using the interface configuration subcommand `<ip ospf network [broadcast or non-broadcast]>`. If you configure a Frame Relay interface (or any other media type, such as Ethernet, Token Ring, and so on) as a nonbroadcast network type, you need to specify a neighbor with which to exchange OSPF messages. The neighbor is specified with the OSPF configuration subcommand `<neighbor [ip address] [priority (0-255)]>:

```bash
interface Serial0
ip address 192.168.0.1 255.255.255.252
encapsulation frame-relay
frame-relay lmi-type ansi
ip ospf network non-broadcast
!
ospf 87
network 192.168.0.1 0.0.0.3 area 0.0.0.0
neighbor neighbor 192.168.0.2 priority 2
```

It is possible to fool point-to-point and multipoint frame-relay interfaces into acting like broadcast media by configuring them as `<ip ospf network broadcast>` interface types. Provided that each link is directly connected, this approach will work fine. With this scenario, only routers with non-zero priority values need to have `<neighbor>` routers specified.

In situations where Frame Relay is being used in a point-to-multipoint scenario, the `<ip ospf network [point-to-multipoint]>` network type is specified. Under these circumstances, neighbor
routers are not used. A point-to-multipoint scenario uses a single IP address space, creating effectively a virtual subnet, where the multipoint Frame Relay router acts as a traffic bridge between disconnected point-to-point routers (see Figure 10.3).

**Figure 10.3**
A point-to-multipoint Frame Relay mesh.

When using DDR links with OSPF, it is quite common to filter out OSPF messages from seeming like interesting outbound traffic to bring up the link. It is, however, desirable to have the routing tables across the routing domain as accurate as possible. With the interface command `<ip ospf demand-circuit>`, it is possible to have the local end router suppress periodic OSPF messages that contain no topology changes. If an OSPF message does contain topology change information and the DDR link is brought up, the messages are passed on to the remote router. When the link is up for normal data use, OSPF messages are exchanged normally. This feature is enabled on the network access router, not the remote router.

**OSPF Monitoring Commands**

Throughout the section, we have used various commands to display different types of information relating to the functioning of the OSPF process. Here is a command and result summary of the more useful `<show ip ospf>` commands:

- `<show ip ospf [local process id]>` provides operational information on the OSPF process—in other words, what kind of OSPF process is being deployed, area information, link state update and age information, and SPF algorithm statistics. This command can also be executed just as `<show ip ospf>`. In this form, it will list all the local OSPF processes running on the router. Because you should only be running one OSPF process, you can use just this command form.

- `<show ip ospf interface [type/slot/port]>` provides OSPF-related information about a specific interface, such as its cost, network type, adjacencies, and designated router information:

```
asbr-a1>ip ospf int fa1/0
FastEthernet1/0 is up, line protocol is up
Internet Address 192.168.19.20/24, Area 0.0.0.0
Process ID 101, Router ID 147.225.181.19, Network Type BROADCAST, Cost: 1
Transmit Delay is 1 sec, State DROTHER, Priority 1
Designated Router (ID) 192.168.181.2, Interface address 192.168.181.2
Backup Designated router (ID) 192.168.181.3, Interface address 192.168.18
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:03
Neighbor Count is 6, Adjacent neighbor count is 2
Adjacent with neighbor 192.168.181.2 (Designated Router)
Adjacent with neighbor 192.168.181.3 (Backup Designated Router)
Supress hello for 0 neighbor(s)
asbr-b2>sh ip ospf neighbor
```

- `<show ip ospf neighbor>` provides information about the router's relationship to the other routers. Who are the designated and backup designated routers and what are those the routers' adjacency states and interface priorities?

```
asbr-b2>sh ip ospf neighbor
```
Neighbor ID   Pri  State    Dead Time  Address    Interface
12.14.116.5    1  FULL/ -  00:00:34  192.168.0.1  Serial0/0

<show ip ospf database> displays the routers entire link state (topologic

OSPF Router with ID (192.168.254.1) (Process ID 57)

Router Link States (Area 0.0.0.0)

Link ID     ADV Router   Age   Seq#    Checksum Link count
12.14.116.5   12.14.116.5  1109  0x80000011 0x2CAB  3
172.16.2.193  172.16.2.193  144   0x8000000D 0xC737  3

Summary Net Link States (Area 0.0.0.0)

Link ID     ADV Router   Age   Seq#    Checksum
172.16.1.1   172.16.2.193  756   0x80000007 0xC135
172.16.2.65  172.16.2.193  756   0x80000007 0x3481
172.16.2.129 172.16.2.193  757   0x80000007 0xB1C3
172.16.2.193 172.16.2.193  757   0x80000007 0x2F06
192.168.160.96 12.14.116.5   1043  0x80000006 0x4BE7

Router Link States (Area 192.168.0.0)

Link ID     ADV Router   Age   Seq#    Checksum Link count
12.14.116.5   12.14.116.5  1042   0x80000006 0xBDD6  1

Summary Net Link States (Area 192.168.0.0)

Link ID     ADV Router   Age   Seq#    Checksum
172.16.1.1   12.14.116.5   1043   0x80000006 0x475D
172.16.2.65  12.14.116.5   1290   0x80000006 0xB9A9
172.16.2.129 12.14.116.5   1290   0x80000006 0x37EB
172.16.2.193 12.14.116.5   1290   0x80000006 0xB42E
172.16.91.0   12.14.116.5   1290   0x80000006 0xC978
192.168.0.0   12.14.116.5   1290   0x80000006 0x14E9
192.168.191.0 12.14.116.5   1291   0x80000006 0xCAA6

- <show ip ospf virtual-links> will display information about the state of virtual links running on the router.

Note - IOS also provides an extensive <debug> toolkit for viewing the various database and functional elements that make up the OSPF process:

```
asbr-a2#debug ip ospf ?
adj       OSPF adjacency events
events    OSPF events
flood     OSPF flooding
lsa-generation   OSPF lsa generation
packet    OSPF packets
retransmission   OSPF retransmission events
spf       OSPF spf
snapshot   OSPF snapshot
```

Note - As with all debug commands, these should be used with caution and disabled when no longer required because they can impact router performance. If you are
experiencing difficulties with OSPF, start out with the `<debug ip ospf events>` command.

### Using AURP

Display commands:

- `<show applletalk aurp events>`
- `<show applletalk aurp topology>`
- `<show applletalk routes>`
- `<show applletalk zone>`
  Configuration commands:
  - `<interface tunnel [number]>`
  - `<applletalk protocol AURP>`
  - `<tunnel source [ip address]>`
  - `<tunnel destination [ip address]>`
  - `<tunnel mode [aurp]>`

In Chapter 3, "The Networker's Guide to AppleTalk, IPX, and NetBIOS," we discussed the function of AppleTalk's AURP (AppleTalk Update Based Routing Protocol), which is an enhancement to the RTMP (Routing Table Maintenance Protocol) that extends AppleTalk networks across large IP internetworks. AURP works by essentially encapsulating AppleTalk routing and data packets into TCP/IP UDP packets.

IOS implements AURP using `<tunnel>` interfaces. To establish the tunnel, a router on each of the LANs creates a tunnel interface. When the tunnel is created, it is bound to a "real" router interface that serves as the tunnel’s source address. This "source" address is used by the remote end to send data to the tunnel interface. Along with a tunnel "source", a tunnel "destination" address is configured. The "destination" address is the "real" interface where all the tunnel’s outbound traffic is addressed. When configuring an AURP tunnel, the source interface is the "IP only" interface on the router. The destination interface is the "IP only" interface of the remote router. Figure 10.4 illustrates the different "interfaces" involved with configuring the AURP tunnel.

AppleTalk must be enabled on the router(s) the tunnel interfaces are configured on because the AppleTalk network needs to be directly accessible through one of the tunnel router’s interfaces.

To monitor AURP, the `<show applletalk aurp events>` and `<show applletalk aurp topology>` commands are available in the user EXEC. The `<show applletalk route>` and `<show applletalk zone>` user EXEC commands are also useful when setting up AURP.

**Figure 10.4**
AURP tunnel interfaces.

Using our sample network, illustrated in Figure 10.5, let’s configure a AURP tunnel between routers asbr-a1 and asrb-b1:

```
hostname asbr-a1
!
interface Tunnel1
  no ip address
  no ip directed-broadcast
```
appletalk protocol aurp
  tunnel source 192.168.0.1
  tunnel destination 192.168.0.2
  tunnel mode aurp
interface serial1
  ip address 192.168.0.1
  no ip directed-broadcast
  no ip route-cache
  no ip mroute-cache

interface Ethernet0
  ip address 192.168.191.19 255.255.255.0
  no ip redirects
  no ip directed-broadcast
  ip route-cache same-interface
  no ip route-cache
  no ip mroute-cache
  appletalk cable-range 65015-65015 65015.117
  appletalk zone Albany Backbone
  appletalk glean-packets

hostname asbr-b1

interface Tunnel1
  no ip address
  no ip directed-broadcast
  appletalk protocol aurp
  tunnel source 192.168.0.2
  tunnel destination 192.168.0.1
  tunnel mode aurp

interface serial1
  ip address 192.168.0.2
  no ip directed-broadcast
  no ip route-cache
  no ip mroute-cache

interface Fast Ethernet1/0
  ip address 172.16.3.19 255.255.255.0
  no ip redirects
  no ip directed-broadcast
  ip route-cache same-interface
  no ip route-cache
  no ip mroute-cache
  appletalk cable-range 60001-60001 60001.125
  appletalk zone Boston Backbone appletalk glean-packets

**Figure 10.5**
AppleTalk network segments on asbr-al and asbr-bl.

**Using BGP**

Display commands:

  <show ip bgp>
  <show ip bgp summary>
  <show ip bgp neighbor><show ip bgp peer-group>

Configuration commands:

  <router bgp [as number]>
For many users, the usefulness of BGP is questionable. BGP, although considered a routing protocol, could be more aptly described as a network reachability protocol. Its focus is not on announcing how to reach a destination, but rather on how to get somewhere that can reach the destination. With this goal in mind, let’s review the three types of routing BGP is used for:

- Inter-autonomous system routing—A collection (two or more) routers from different autonomous systems exchange BGP information with the goal of maintaining a complete reachability table for the entire intranetwork.
- Intra-autonomous system routing—Used by peer-routers connected to the same autonomous system to exchange external AS information.
- Pass-through autonomous system routing—Used to exchange BGP reachability information between a collection of routers belonging to the same or different ASs across an uninvolved third-party AS. In this case, the ASs exchanging reachability information are dependent on the third-party AS’s IGP protocol to transport the BGP traffic.

As stated earlier, almost all applications of BGP occur within the context of the public Internet. Most private enterprise networks do not have these types of requirements, and those that do can safely accomplish their goals with OSPF or EIGRP. However, with businesses’ growing dependency on the Internet and the hyper-growth of many corporate intranets, the usefulness of BGP is finally reaching the private network. It is particularly applicable to environments where access to multiple private/public networks links is required. Because BGP was designed to manage the Internet, it is powerful, and as you can imagine, can be quite complex to configure. With this in mind, we will look at some basic inter-autonomous and intra-autonomous BGP configurations.

**Inter-Autonomous BGP**
For most BGP applications, the goal is to provide network-reachability information for your local network between two or more ISP gateways. In this situation, your goal is not to configure a large BGP routing policy. All you want is to have Internet access redundancy. An easy way to achieve this goal is to use *ships in the night (SIN) routing*. SIN routing exists where a router is running multiple dynamic protocols, without redistributing them into one another. The effect is that only the SIN router processes a complete internal and external network routing table. The other internal network routers use the SIN router as the default gateway, so they blindly forward all of their external traffic to the SIN router, which then uses its full routing table to decide the best next hop. When configuring BGP for the SIN routing context, you want to set up the BGP process on your internal network’s gateway router instead of configuring BGP on each of your Internet access gateways.

*Figure 10.6* illustrates a SIN routing router topology configuration example.

**Figure 10.6**
A multipoint access Internet gateway model.

The example shows both ISP routers being directly accessible from the internal gateway router. The SIN approach can also be followed using a single internal/Internet access router where the internal network and the multiple Internet links are all homed into the same router. This approach does have some drawbacks. The first is the fact that there is no hardware redundancy; if the internal gateway fails, you are completely down. To overcome this and still maintain a single gateway for end-stations and servers, use HSRP (or VIRP) with two gateway routers. The second drawback has to do with security. Internet gateways are managed by you and in part by your ISP. It is usually not a good idea to have your ISP accessing your internal network gateway router.

After the connectivity has been established between the internal gateway router and the Internet gateway routers, you want to establish a local BGP process on your gateway router. This is started using the global configuration EXEC command `<router bgp [as number]>`. If you are planning to use BGP to provide reachability information over the Internet, you will need a registered AS number from an Internet Addressing Authority (IAA), such as the American Registry for Internet Numbers (ARIN). For unannounced BGP (for internal use only), any number between 32768 and 64511 will do. For this example, we are going to announce networks 12.14.116.0 /22 and 192.168.180 /22.

Before we start configuring our BGP process, we first need to establish routes to our remote-AS BGP neighbors. In most cases, you will peer with the router on the other end of your Internet link. In this case, the routers are (ATT) 12.127.40.65 and (PSI) 192.144.36.1. We can accomplish peering with static routes:

```
ASBR-Master#config t
ASBR-Master(config)#ip route 12.127.40.64 255.255.255.252 12.14.116.3
ASBR-Master(config)#ip route 192.174.36.0 255.255.255.252 12.14.116.2
```

Now, let’s enable our BGP process. The process ID is the AS number, which, in this case, is 789. The process ID is used for communication with other BGP processes. If you are planning to run BGP on more than one internal router, be sure that the same AS is used:

```
ASBR-Master(config)#router bgp 789
ASBR-Master(config-router)#
```
After the process starts, we add the networks we want announced. Like OSPF, BGP uses netmasks as part of the network number announcements. This means that you can announce parts of networks as VLSM subnets, as CIDR address ranges, or as classful networks. All that matters is that the addressing has correct bit boundaries and that you want them announced. For example, we do not want the 12.14.117.0 and 118.0 spaces announced and we are using 12.14.116.0 in various VLSM address segments. So, we will only announce the networks we know should be accessed from the Internet:

```plaintext
ASBR-Master(config-router)#network 12.14.116.0 mask 255.255.255.252
ASBR-Master(config-router)#network 12.14.116.128 mask 255.255.255.128
ASBR-Master(config-router)#network 12.14.119.0 mask 255.255.255.0
ASBR-Master(config-router)#network 192.168.180.0 mask 255.255.252.0
```

Now, let’s configure our remote neighbors, using the BGP configuration subprocess command `<neighbor [ip-address] remote-as [as number]>`:

```plaintext
ASBR-Master(config-router)#neighbor 12.127.40.65 remote-as 46
ASBR-Master(config-router)#neighbor 192.174.36.1 remote-as 67
ASBR-Master(config-router)#neighbor 192.174.36.1 version 4
ASBR-Master(config-router)#neighbor 12.127.40.65 version 4
ASBR-Master(config-router)#neighbor 192.174.36.1 soft-reconfiguration inbound
ASBR-Master(config-router)#neighbor 12.127.40.65 soft-reconfiguration inbound
```

The `<neighbor [ip address] version 4>` command forces which version of BGP will be used, instead of using the default negotiation process which starts at version 4 and moves downward until a version match is found. The `<neighbor [ip address] soft-reconfiguration inbound>` command indicates that changes to the inbound BGP configuration should be enabled automatically. Generally, whenever a change is made to the BGP configuration, the connection between neighbors needs to be reset. This is accomplished with `<clear ip bgp *>` (to reset all sessions) or `<clear ip bgp [neighbor ip address]>` to reset a specific session. With our configuration, our outbound connections need to be reset whenever we make a change.

After BGP routing is established with our Internet access providers, we can also establish BGP adjacencies with other ASs, if we desire. To establish a peer session with BGP neighbors that are not accessible over a direct connection, the `<neighbor [ip address] ebgp-multihop>` BGP configuration subprocess command must added to the BGP configuration in addition to the `<remote-as>` declaration. In this example, our BGP router is the default gateway for the internetwork. Once the externally destined traffic arrives, it will decide which ISP gateway to forward the traffic to based on its local routing table. Alternatively, we can redistribute the BGP derived routing information into the local IGP process.

When using BGP to provide reachability information about networks within the AS (in perhaps a large global private internetwork), two approaches can be taken:

- The first is to create BGP peer groups, which will exchange internal BGP routing information and provide gateway services for the local internetworks.
- The second is to redistribute the BGP process into the IGP process and have the IGP routers determine the destination path.
The IBGP approach will be discussed in the next section. BGP redistribution will be covered in the section "Route Control and Redistribution." In most cases, you want to avoid BGP-to-IGP and IGP-to-BGP redistribution.

Intra-Autonomous BGP Configuration

In the inter-autonomous BGP example, we used a single "distribution" router to act as a gateway between the BGP and IGP processes. In large private enterprise network environments, it is not uncommon for each location to have an Internet gateway and one or more internal access gateways to some or all of the other internal network sites. Our testnet example uses just such a network model. When configuring Internal BGP (IBGP), all the IBGP routers need to have peering sessions with all the other IBGP routers.

Let’s examine the IBGP/EBGP configurations on asbr-a1 and asbr-a2. For this example, the test network uses AS number 30001. Both Albany and Boston use an IGP protocol to manage local route announcement. BGP is used to advertise routes over the private WAN links and with the ISPs. asbr-a1/a2 and asbr-b1/b2 use identical BGP configurations. Here are the BGP configurations for asbr-a1 and asbr-b1:

```
hostname asbr-a1
!
autonomous-system 30001
!
router bgp 30001
network 192.168.0.0 mask 255.255.255.252
network 192.168.0.4 mask 255.255.255.252
network 192.168.0.128 mask 255.255.255.128
network 192.168.1.0 mask 255.255.255.0
network 192.168.2.0 mask 255.255.255.0
network 192.168.3.0 mask 255.255.255.0
network 192.168.4.0 mask 255.255.255.0
network 192.168.5.0 mask 255.255.255.0
network 192.168.6.0 mask 255.255.255.0
network 192.168.7.0 mask 255.255.255.0
network 192.168.8.0 mask 255.255.255.0
network 192.168.9.0 mask 255.255.255.0
network 12.14.116.4 mask 255.255.255.252
neighbor 192.168.0.6 remote-as 30001
neighbor 192.168.0.6 filter-list 30001 4 weight 45000
neighbor 12.14.116.5 remote-as 8446
neighbor 12.14.116.5 weight 40000
neighbor 192.127.30.1 remote-as 899
neighbor 192.127.30.1 ebgp-multihop
neighbor 192.127.30.1 weight 30000
neighbor 192.168.1.20 remote-as 30001
neighbor 192.168.1.20 filter-list 30001 4 weight 42000
no auto-summary
!
ip as-path access-list 4 permit 30001
!
hostname asbr-b1
!
autonomous-system 30001
!
router bgp 30001
network 192.127.30.0 mask 255.255.255.252
network 172.16.0.0 mask 255.255.240.0
neighbor 192.127.30.1 weight 40000
```
Routers asbr-a1/a2 and asbr-b1/b2 are exchanging external BGP and internal BGP routing information about their local networks. As you can see, both IBGP and EBGP peer sessions (also referred to as speakers) are configured using the `<neighbor [ip address] remote-as [as number]>` command. What demonstrates the existence of IBGP peer session is that the remote AS number is identical to that of the local AS number. Because there are four IBGP routers in AS-30001 that have EBGP peer sessions and provide similar network reachability information, administrative weights are used to set preferences for which routes the speakers’ broadcasts will be sent to.

---

**Note** - To prevent routing loops, external network information learned from IBGP peers is not readvertised to other IBGP peers.

---

Weights are used to ensure that the external routes provided through the locally adjacent peer are preferred over the remote peer. This same approach is used with the IBGP routers using a special kind of access list for filtering `as-path` information. `as-path acl` is created in general configuration EXEC mode using `<ip as-path access-list [acl number] [permit|deny] [string]>`. The filter is then applied to a neighbor using the `<neighbor [ip address] filter-list [acl number] weight [0 - 65535]>`.

Because all the AS-30001 routers are using similar configurations, a BGP peer-group could be created to reduce some of the configuration overhead. BGP peer-groups provide a way of configuring a core set of BGP parameters that are used by all of the peer-group members. Peer-groups are created in the BGP configuration subprocess:

```
asbr-b2(config-router)#neighbor as30001 peer-group
```

The command shown above sets up the peer-group and establishes it as an IBGP peer-group. The following command establishes all the members of the IBGP peer-group:

```
asbr-b2(config-router)#neighbor 192.168.0.5 as30001
asbr-b2(config-router)#neighbor 192.168.0.6 as30001
asbr-b2(config-router)#neighbor 192.168.0.1 as30001
asbr-b2(config-router)#neighbor 192.168.0.2 as30001
```

The following command enables MD5 authentication (and the password) to be used when establishing BGP peer sessions:

```
asbr-b2(config-router)#neighbor as30001 password as-30001
```

It is also possible to enable filtering with `<neighbor [peer-group name] filter-list [acl]`
Additional options include router distribution, route-maps, version, advertisement interval—any configuration option that can be applied group-wide. It is also possible to override a peer-group option by configuring an option locally.

After they are configured, neighbor routers just create the same peer-group within their local BGP configuration using `<neighbor [peer-group name] peer-group>`. Any member of the peer-group can make contributions to the group-wide configuration.

**Note** - An alternative to using weights is to use the administrative distances to determine route preference. With BGP, three distances are used: external, internal, and local. To change the distances from their default settings, use the BGP configuration subcommand `<distance [external] [internal] [local]>`. The range, in case you forgot, is 1 to 255.

---

**BGP Reflectors**

An alternative to having a group of routers configured for external peer relationships is to have one router form an external peer relationship and then act as a BGP reflector. IBGP routers do not re-advertise external routes learned from IBGP peers to other IBGP peers. This is done so routing loops are not created. This was the reason each of the IBGP peers needed its own EBGP session to construct a full routing table. When a reflector is used, it acts as a routing information hub which all the IBGP peers connect to, removing the need to have each of the IBGP peers have peer sessions with all the IBGP routers within the AS. In situations where more than one reflector is needed (or desired), the involved IBGP routers will require a cluster id for route identification. In other scenarios where only one reflector is used, the cluster is identified by the reflector router's ID. Figure 10.7 illustrates an EBGP/IBGP reflector configuration using our test network.

Here is the IBGP reflector configuration from `asbr-a1`:

```
hostname asbr-a1
!
autonomous-system 30001
!
router bgp 30001
bgp cluster-id 30001
network 192.168.0.0 mask 255.255.255.252
network 192.168.0.4 mask 255.255.255.252
network 192.168.0.128 mask 255.255.255.128
network 192.168.1.0 mask 255.255.255.0
network 192.168.2.0 mask 255.255.255.0
network 192.168.3.0 mask 255.255.255.0
network 192.168.4.0 mask 255.255.255.0
network 192.168.5.0 mask 255.255.255.0
network 192.168.6.0 mask 255.255.255.0
network 192.168.7.0 mask 255.255.255.0
network 192.168.8.0 mask 255.255.255.0
network 192.168.9.0 mask 255.255.255.0
network 12.14.116.4 mask 255.255.255.252
neighbor 12.14.116.5 remote-as 8446
neighbor 192.127.30.1 remote-as 899
neighbor 192.127.30.1 ebgp-multihop
neighbor 192.168.0.2 remote-as 30001
```
neighbor 192.168.0.5 remote-as 30001
neighbor 192.168.0.5 remote-as 30001
neighbor 192.168.0.5 route-reflector-client
neighbor 192.168.1.20 route-reflector-client

**Figure 10.7**
The EBGP and IBGP peer relationships in our test network.

Router `asbr-b1` uses an identical configuration. Notice that `asbr-a1` and `asbr-a2` have a full peer relationship and are not configured as reflector clients. As for the client configuration, IBGP sessions are only established with the reflector(s).

**Monitoring the BGP Process**

As with the IGP protocols, BGP has a set of IOS commands that reveal various operational details about the process. Here is a command list and brief command result summary:

- `<show ip bgp>`—This command displays BGP summary route information, the route's weight, path source, router's ID, cluster attributes, and so on.

- `<show ip bgp summary>`—This is the most often used of all the BGP informational commands. This command yields summary statistics about the running BGP process, and, most importantly, provides the connection status of all the neighbor peer sessions: the activity state, bytes transferred, up/down status, version, and so on.

- `<show ip bgp peer-group>`—This command provides summary information on the peer-group configuration, particularly what options have been inherited, and/or added or overwritten.

- `<show ip bgp neighbors>`—Another very useful tool, this command provides detailed information on the status and configuration of the router's configured peers.

Debugging commands are also available. `<debug ip bgp>` enables all of them, although this is not recommended. `<debug ip bgp events>` echoes BGP operational events—peer session establishment and communication—and other housekeeping. `<debug ip bgp updates>` displays the route update information and exchanges between peers. All the `<show>` and `<debug>` commands are displayed using the root of the command and a question mark:

```
show ip bgp ?
A.B.C.D Network in the BGP routing table to display
cidr-only Display only routes with non-natural netmasks
community Display routes matching the communities
community-list Display routes matching the community-list
dampened-paths Display paths suppressed due to dampening
filter-list Display routes conforming to the filter-list
flap-statistics Display flap statistics of routes
inconsistent-as Display only routes with inconsistent origin ASs
```

```
display ?
```

Route Control and Redistribution

IOS, for better or worse, supports the redistribution of both static and dynamically derived routing information through dynamic routing protocols. You should take notice of the introductory qualification of "for better or worse." The "better" is the capability to use route redistribution to have different dynamic routing protocols announce each other's routing information. Redistribution is helpful when multiple dynamic routing protocols are used to manage different parts of a large internetwork. This condition is often found in large corporate networks where different sites (and sometimes departments) are managed individually, but need to interact with the larger corporate whole. Here, two-way route redistribution is a tremendous win.

Internet or private global network connections are another context where distinct dynamic routing protocols are in use to enforce the separate routing policies of the two networks. With single-point connections, static routes can be used to send packets destined to external networks to the Internet gateway. In situations where multiple Internet and private network links exist as an alternative to using BGP, it is possible to use filtered redistribution to limit the inbound and outbound network announcements between the different routing policies using distribution lists and route-maps.

Static route redistribution is often used in dial-access and DDR networks where dynamic protocols are not always effective. Instead of adding static routes on every router on the network, a single router can redistribute a collection of static routes. Static route redistribution can also be used to limit network announcements inside internal networks. By using multiple dynamic routing processes to manage different sections of the network, and then statically redistributing on a selective basis, you will only announce the networks that need to be announced between the separate routing processes.

The "worse" is the potential for routing loops and the difficulty surrounding the translation of different routing protocol metrics and distances. Here are some guidelines to help you avoid the "worse" side of using redistribution:

- Ask yourself, "What is the advantage to using redistribution (in a given situation) over using a static route?" In most cases, a static route or default route can be used to provide a network announcement. If you have only a few networks to redistribute, use static routes. Ideally, you only want to be running a single routing protocol. In situations where legacy equipment might reduce your choices in terms of which protocols you can use, it might be more advantageous to use a single protocol or static routing (to address legacy hardware issues) than to add the additional processing and network traffic associated with multiple protocol redistribution.

- If possible, avoid redistributing between classless and classful protocols if you are using VLSM. Classful protocols do not include network address masks in their messages. Instead, they apply the mask of the interface that receives the route announcements. Consequently, only VLSM networks that have compatible masks will be advertised, and the others will not be redistributed.
When using redistribution with multigateway networks (or HSRP) it is essential that the gateway routers use metrics and distances that favor one router over the other. Earlier in this chapter, we discussed how a route’s administrative distance determines its desirability in comparison to the same route from another source. The lower the administrative distance, the more desirable the route. When redistributing routes from one protocol to another, the redistributed routes are assigned a default administrative distance. If the same network information is redistributed by more than one router in the routing process, a routing loop will result. To avoid this while still maintaining redundancy, adjust the administrative distance for the redistribution on the routers so one is preferable to the other. This will result in having the "preferable" routes used, and in the event that the preferred router fails, the "less-preferable" routes will be available.

**Basic Redistribution**

Basic route redistribution is enabled with the routing protocol configuration subcommand
<redistribute [source] [process id] [metrics]>.

The redistribution metrics can be set specifically for the redistributed routes in the <redistribution> command stanza or by using a <default-metric [metrics]> router subprocess declaration. In either case, the metric settings reflect those used specifically by the protocol, such as hop-count for RIP, cost for OSPF, and so on. Figure 10.8 illustrates a simple redistribution example. ASBR-20 is redistributing RIP, OSPF 20, and EIGRP 22 so ir-1a, ir-1b, and ir-3c can reach each other’s networks.

As you can see in Figure 10.8, the redistribution configuration is similar for all the protocols. The major difference is that each protocol applies its own metrics to the redistributed routes. OSPF uses cost, which you can estimate adding the sum costs of all the interfaces. In our example, the interfaces directly connecting the RIP and EIGRP networks are Ethernet, which have an OSPF cost of 10. In fact, all the involved interfaces are Ethernet, so the cost was determined by adding the interface cost for each gateway a datagram would have to pass through to reach its destination. A similar approach was followed in determining the RIP metric: the average hop-count was used as the metric. The EIGRP (same for IGRP) metric you should recall is as follows:

- Bandwidth in KB/s
- Delay (in tenths of milliseconds)
- Reliability 1-255
- Load 1-255
- MTU

When configuring EIGRP metrics for redistribution, use the redistribution source interface’s default metrics. These can be retrieved (you should know this, by the way) using the <show interface [type/slot/port]> command. Unless metrics are specified, the redistributing protocol will use a default value, which in most cases identifies the routes as directly connected. This makes each redistributed route appear to have the same path metric, which basically defeats the advantage of using a dynamic routing protocol to determine the best route path. With simple redistributions, using the default will generally work fine. In situations where multiple gateways and redistribution are
combined, however, the use of metrics is strenuously recommended.

**Figure 10.8**
An example network running three routing protocols that require redistribution.

OSPF requires a little more specificity than the other protocols when redistributing routes. The `<metric-type [1 or 2]>` is an option that enables the redistributed routes to be identified as OSPF external type 1 or type 2 routes. The `<subnets>` command is specific only to OSPF; it specifies that subnet information will be redistributed along with the network information. If `<subnets>` is not used, the netmask information is not sent, only the classful network address is sent. If VLSM is used with large classful A or B space, and if `<subnets>` is not specified, redistribution will not work. As a rule, the `<subnets>` command should be part of any redistribution (both static and dynamic) declaration. When an OSPF router redistributes any routing information, it becomes an ASBR. As an ASBR, the router can also redistribute default route information. To enable OSPF to generate and announce a default route statement, use the OSPF configuration subcommand `<default-information originate always>`. This command should only be configured on the router that functions as the network’s default gateway.

As an alternative to SIN routing, here is an OSPF/BGP redistribution using default route distribution:

```conf
router ospf 45
  redistribute static subnets
  redistribute bgp 23 metric 1600 metric-type 2 subnets
  network 192.168.1.0 0.0.0.255 area 0.0.0.0
  network 192.168.2.96 0.0.0.3 area 192.168.0.0
  network 192.168.5.0 0.0.0.255 area 192.168.0.0
  default-information originate always
!
router bgp 23
  network 172.16.0.0
  network 192.168.0.0 mask 255.255.240.0
  neighbor 12.14.116.2 remote-as 45550
```

The `<ip default-network [ip address]>` used by IGRP and EIGRP generates a default network statement.

**Controlling Redistribution**

In some cases, it is not always desirable to redistribute all the routing announcements from one protocol into another. To restrict the flow of dynamic route announcements between routers belonging to the same routing policy or between protocol redistributions, IOS provides route filtering. Route filtering is particularly advantageous for administrators who need to keep strict control over the route announcements exchanged between routers.

**Filtering with Distribution Lists**

The route filter’s responsibility is to suppress unwanted routing information from being redistributed, entered in or advertised out of the routing table. Although the result is the same as redistribution, this process behaves differently if a link state or distance-vector protocol is being used. When you use a route filter with a distance vector protocol, the filter affects not only the router the filter refers to, but also all the routers that receive advertisements from the filtered router. This is the result of the
distance vector using their routing tables as the basis for the network announcements. When filtering is employed with a link state protocol, however, the filter will only affect the local routing table. Link state protocols use link state announcements to construct their routing tables, and these are unaffected by route filters.

Routing filters are configured in two parts, starting with the filter list, which is a standard access list. Access list entries are entered using only the network address:

```
asbr-a1(config)#access-list 2 permit 129.0.0.0
asbr-a1(config)#access-list 2 permit 192.168.0.0
asbr-a1(config)#access-list 2 permit 192.168.10.0
```

After the list is created, it is applied to the routing protocol configuration with the routing configuration subcommand `<distribute-list [acl number] [in|out] [protocol|interface]>`. Inbound filter lists are applied to interfaces. They permit or deny the flow of route announcements through a router interface. Inbound filters are used for suppressing route propagation with distance vector protocols or the addition of route announcements on a local router when used with a link state protocol. They are often used to correct a side effect of route redistribution known as route feedback. Route feedback occurs when routing information is advertised in the wrong direction across a redistributing router. Inbound filters are also useful from a security perspective, if you are only routing to or from a specific set of networks. Applying an inbound route filter will suppress any unauthorized networks from being injected into the routing table.

Here are inbound filter examples using OSPF and RIP (see Figure 10.9). ABR-a57 is filtering announcements from ir-23 and ir-24. Only the 192.168.x.0 networks need to be accessible throughout the network. The 10.0.x.0 networks are used for local testing and only need to be available to the networks directly connected to ir-33 and ir-34. The first router configuration example shows OSPF using an inbound distribution list to suppress the addition of the 10.0.x.0 routes on the local routing table. The next example shows the same thing with RIP. The RIP usage of the inbound `<distribute-list>` command also prevents the 10.0.x.0 routes from being announced to ABR-a58 and ABR-a56:

```
hostname ABR-a57
!
access-list 1 permit 192.168.21.0
access-list 1 permit 192.168.22.0
access-list 1 permit 192.168.23.0
access-list 1 permit 192.168.24.0
access-list 1 permit 192.168.25.0
access-list 1 permit 192.168.26.0
access-list 1 permit 192.168.27.0
!
ospf 57
network 172.16.192.0 0.0.0.255 area 57
network 172.16.60.28 0.0.0.3 area 0.0.0.0
network 172.16.60.32 0.0.0.3 area 0.0.0.0distribute-list 1 in FastEthernet0/0
```

**Figure 10.9**
A campus backbone network example for using the `<distribute-list>` route distribution/suppression command.

```
hostname ABR-a57
!
access-list 1 permit 192.168.21.0
```

file://J:\NewRiders\chapters\ze392.html 7/12/01
Outbound filters on distance-vector protocols can be used for protocol redistributions and for filtering the network announcements sent out of a router’s interface. Take for instance our example illustrated in Figure 10.7. In the RIP example above, the goal was to limit the network accessibility to the 10.0.x.0 networks. Because ABR-a57 is also an Internet gateway, we might want the 10.0.x.0 users to have Internet access (assuming our usage of the 10.0.0.0 network address was permissible), but still want to limit their Internal access. So, instead of using an inbound filter on the ABR-a57 Fe0/0 interface, which prevents the 10.0.x.0 routes from being added to the ABR-a57 routing table, outbound filters can be applied to interface e5/0 and e5/1 to suppress the 10.0.x.0 network from being announced to the rest of the private network:

hostname ABR-a57

access-list 1 deny 10.0.0.0
access-list 1 permit any

rip
version 2
network 172.16.0.0

Link state protocols, however, can only use outbound filters for limiting the network announcements being redistributed into another routing protocol. Again, let’s use Figure 10.7. In this scenario, OSPF is being used to manage all the internal routing (including the test networks) for the network. This information is then redistributed into BGP, which is being used to announce reachability for the network segments. However, because the test networks are using "unregistered" IP address space that cannot be announced publicly, they need to be filtered from the redistribution. To prevent their announcement, they are being suppressed using an outbound distribution list:

hostname abr-a57.

router ospf 45
network 172.16.192.0 0.0.0.255 area 57
network 172.16.60.32 0.0.0.3 area 0.0.0.0
network 172.16.60.28 0.0.0.3 area 0.0.0.0

default-information originate always
distribute-list 2 out bgp 23

bgp 890
redistribute ospf 45
neighbor 12.14.116.2 remote-as 2889
neighbor 12.14.116.2 password gimmyroutes
neighbor 12.14.116.2 soft-reconfiguration inbound
neighbor 12.14.116.2 version 4
neighbor 12.14.116.2 distribute-list 4 out
neighbor 12.14.116.2 distribute-list 3 in
The OSPF distribution list redistributes all the networks except the 10.0.x.0 test networks. The distribution lists on the BGP process do not restrict announcements in either direction. When using redistribution to announce networks, however, particularly to the Internet, it is a good idea to implement at the very minimum a basic filter so when the need arises to restrict something from the redistribution it is simply an access-list change.

Filtering with Route-Maps

In Chapter 9, we used route-maps to configure policy routing. Route-maps can also be used to control the flow of route redistributions and apply metrics to redistributed routes for IGP and EGP protocols. Here is a simple IGP example that does both.

Asbr-2b is redistributing EIGRP 45, the local routing process for the Boston network into OSPF 87, which is the routing process for Albany network. The problem is that asbr-b1 is redistributing OSPF 87 into EIGRP 45, so asbr-b2 needs to filter any local Albany routes that are being announced from the redistribution running on asbr-b1. The filtering is being done by a route-map named "grinder."

Let’s take a look at how the route-map "grinder" is created. First, a standard IP ACL containing all of the Albany local routes needs to be created:

```
  !
  access-list 2 deny 10.0.1.0
  access-list 2 deny 10.0.2.0
  access-list 2 deny 10.0.3.0
  access-list 2 deny 10.0.4.0
  access-list permit any
  !
  access-list 4 permit any
  !access-list 3 permit any
```

Note - Route-maps are used extensively for controlling BGP routing announcements. This application, however, is way beyond the scope of this book. For more information on using route-maps and BGP, check the additional resources section at the end of the chapter.

Now, we configure the route-map. The first line permits any route announcement that matches any address in access-list 2 to be redistributed with an OSPF type 2 metric:

```
  asbr-b2(config)#route-map grinder permit 10
  asbr-b2(config-route-map)#match ip address 2
```
The second line configures the route-map "grinder" to deny any route announcement that matches any address in the access list 1:

```
asbr-b2(config)#route-map grinder 10asbr-b2(config-route-map)#match ip address
```

After the map is completed, it just needs to be applied to the OSPF process:

```
asbr-a2(config-router)#$redistribute eigrp 99 route-map grinder
```

Here are the basic match and set statements you can use to control route redistributions:

```
<match ip address [ACL]>
<match metric [metric]>
<match ip route-source [ACL]>
<set metric [metric values used by the redistributing protocol]>
<set metric type [type-1|type-2]>
```

Keep in mind that route-maps, like access-lists, use deny as their default action, so you need to declare permit statements when configuring the map statements.

**Summary**

The focus of this chapter has been the configuration of static and dynamic IP routing on Cisco routers using the Cisco IOS. At this point, you should have a familiarity with the following concepts:

- Managing static routing
- Implementing RIP, OSPF, and EIGRP
- Adjusting routing preferences with administrative distances
- Implementing AURP
- Implementing BGP
- Route redistribution

In Chapter 11, "Network Troubleshooting, Performance Tuning, and Management Fundamentals," we look at network management, troubleshooting, and performance fundamentals. The chapter’s goal will be to introduce basic network management and diagnostics concepts. Covered topics include SNMP, RMON, hardware and software network management tools, network management models, network performance baselining, and common network problem identification.

**Related RFCs**
<table>
<thead>
<tr>
<th>RFC</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1245</td>
<td>OSPF Protocol Analysis</td>
</tr>
<tr>
<td>1246</td>
<td>Experience with the OSPF Protocol</td>
</tr>
<tr>
<td>1397</td>
<td>Default Route Advertisement in BGP2 and BGP3 Versions of the Border Gateway Protocol</td>
</tr>
<tr>
<td>1467</td>
<td>Status of CIDR Deployment in the Internet</td>
</tr>
<tr>
<td>1518</td>
<td>An Architecture for IP Address Allocation with CIDR</td>
</tr>
<tr>
<td>1519</td>
<td>Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy</td>
</tr>
<tr>
<td>1582</td>
<td>Extensions to RIP to Support Demand Circuits</td>
</tr>
<tr>
<td>1586</td>
<td>Guidelines for Running OSPF over Frame Relay Networks</td>
</tr>
<tr>
<td>1587</td>
<td>The OSPF NSSA Option</td>
</tr>
<tr>
<td>1721</td>
<td>RIP Version 2 Protocol Analysis</td>
</tr>
<tr>
<td>1774</td>
<td>BGP-4 Protocol Analysis</td>
</tr>
<tr>
<td>1817</td>
<td>CIDR and Classful Routing</td>
</tr>
</tbody>
</table>

**Additional Resources**


© Copyright Pearson Education. All rights reserved.