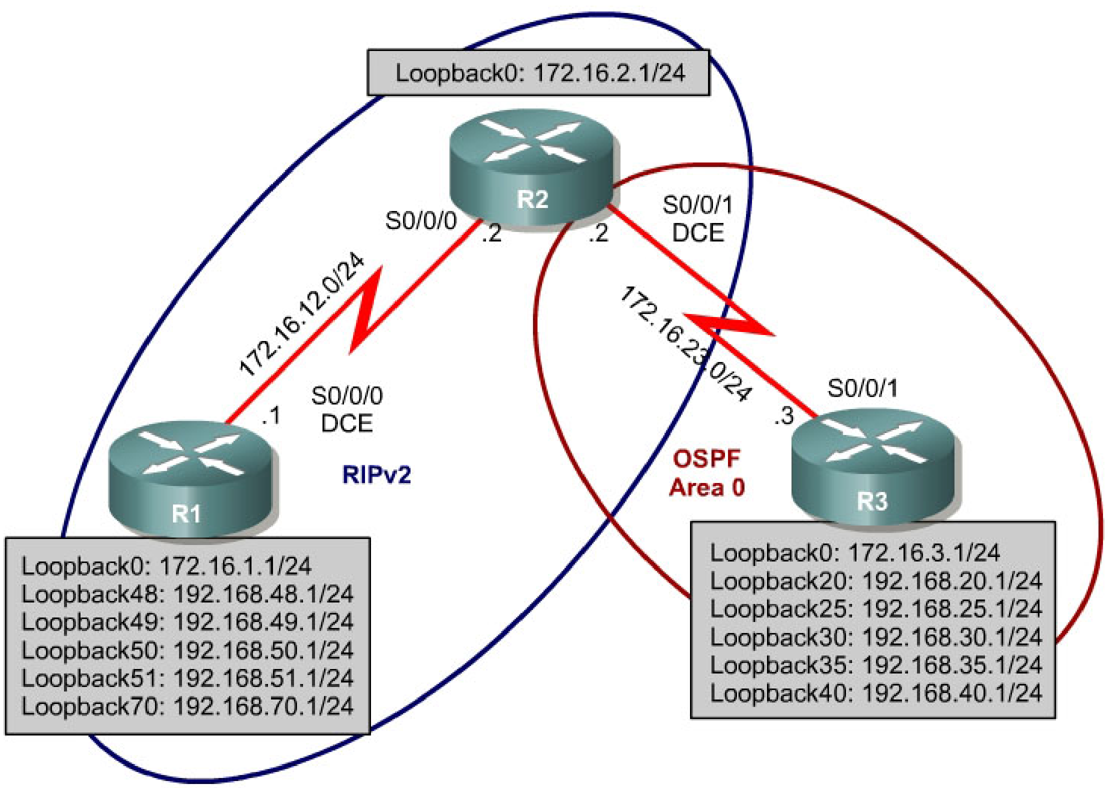
Lab 5-1 Redistribution Between RIP and OSPF

Learning Objectives

- Review configuration and verification of RIP and OSPF
- Configure passive interfaces in both RIP and OSPF
- Filter routing updates using distribute lists
- Redistribute static routes into RIP
- Redistribute RIP routes into OSPF
- Redistribute OSPF routes into RIP
- Originate a default route into OSPF
- Set a default seed metric
- Modify OSPF external network types
- Configure summary addresses

Topology Diagram



Scenario

Two online booksellers, Example.com and Example.net, have merged and now need a short-term solution to interdomain routing. Since these companies provide client services to Internet users, it is essential to have minimal downtime during the transition.

Example.com is a small firm running RIP, while Example.net has a somewhat larger network running OSPF. The diagram identifies R2 as the router that will bridge the two networks. Since it is imperative that the two booksellers continuously deliver Internet services, you should bridge these two routing domains without interfering with each router's path through its own routing domain to the Internet.

The CIO determines that it is preferable to keep the two protocol domains pictured in the diagram during the transition period, because the network engineers on each side need to understand the other's network before deploying a long-term solution. Redistribution will not be your long-term solution, but will suffice as a short-term solution.

Configure the topology above in a lab to verify the short-term solution. In this scenario, R1 and R2 are running RIPv2, but the 172.16.23.0/24 network between R2 and R3 is running OSPF. You need to configure R2 to enable these two routing protocols to interact to allow full connectivity between all networks.

Step 1: Assign Addresses

Configure all loopback interfaces on the three routers in the diagram. Configure the serial interfaces with the IP addresses, bring them up, and set a DCE clock rate where appropriate.

```
R1(config)# interface Loopback0
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# interface Loopback48
R1(config-if)# ip address 192.168.48.1 255.255.255.0
R1(config-if)# interface Loopback49
R1(config-if)# ip address 192.168.49.1 255.255.255.0
R1(config-if)# interface Loopback50
R1(config-if)# ip address 192.168.50.1 255.255.255.0
R1(config-if)# interface Loopback51
R1(config-if)# ip address 192.168.51.1 255.255.255.0
R1(config-if)# interface Loopback70
R1(config-if)# ip address 192.168.70.1 255.255.255.0
R1(config-if)# interface Serial0/0/0
R1(config-if)# ip address 172.16.12.1 255.255.255.0
R1(config-if)# no fair-queue
R1(config-if)# clock rate 64000
R1(config-if)# no shutdown

R2(config)# interface Loopback0
R2(config-if)# ip address 172.16.2.1 255.255.255.0
R2(config-if)# interface Serial0/0/0
R2(config-if)# ip address 172.16.12.2 255.255.255.0
R2(config-if)# no fair-queue
R2(config-if)# no shutdown
R2(config-if)# interface Serial0/0/1
R2(config-if)# ip address 172.16.23.2 255.255.255.0
```



```
R2(config-if)# clock rate 2000000
R2(config-if)# no shutdown

R3(config)# interface Loopback0
R3(config-if)# ip address 172.16.3.1 255.255.255.0
R3(config-if)# interface Loopback20
R3(config-if)# ip address 192.168.20.1 255.255.255.0
R3(config-if)# interface Loopback25
R3(config-if)# ip address 192.168.25.1 255.255.255.0
R3(config-if)# interface Loopback30
R3(config-if)# ip address 192.168.30.1 255.255.255.0
R3(config-if)# interface Loopback35
R3(config-if)# ip address 192.168.35.1 255.255.255.0
R3(config-if)# interface Loopback40
R3(config-if)# ip address 192.168.40.1 255.255.255.0
R3(config-if)# interface Serial0/0/1
R3(config-if)# ip address 172.16.23.3 255.255.255.0
R3(config-if)# no shutdown
```

Be sure you can ping across the serial links when you are done.

TCL scripting is heavily used in the route optimization labs to show full or partial connectivity. If you are unfamiliar with TCL scripting or need a refresher, use the TCL reference document provided with the routing lab guide.

The TCL shell is only available on Cisco IOS release 12.3(2)T or later, and only in specific Cisco IOS feature sets. Refer to the Cisco.com Feature Navigator for more information at http://www.cisco.com/go/fn/.

You will be checking full and partial connectivity throughout this lab with the following TCL script:

```
foreach address {
172.16.1.1
192.168.48.1
192.168.49.1
192.168.50.1
192.168.51.1
192.168.70.1
172.16.12.1
172.16.2.1
172.16.12.2
172.16.23.2
172.16.3.1
192.168.20.1
192.168.25.1
192.168.30.1
192.168.35.1
192.168.40.1
172.16.23.3
} { ping $address }
```

At this point, the only pings you should receive back are those connected networks to the router from which you are pinging.

Step 2: Configure RIPv2

Configuring RIPv2 on a router is fairly simple:

- Type the global configuration command **router rip** to enter RIP configuration mode.
- Enable RIP version 2 with the **version 2** command.
- Enter the **no auto-summary** command to disable automatic summarization at classful network boundaries.
- Add the networks you want using the **network** *network* command.

Unlike EIGRP and OSPF, you cannot use a wildcard version of the **network** command, and you have to add the whole network. This is an inherited command from the classful protocol RIPv1. Classful protocols do not support subnets, so subnet or wildcard masks are unnecessary.

Which major networks do you need to add into RIP from the above diagram?

From which routers will these networks be advertised?

Apply the following commands to R1 and R2:

```
R1(config)# router rip
R1(config-router)# version 2
R1(config-router)# no auto-summary
R1(config-router)# network 172.16.0.0
R1(config-router)# network 192.168.48.0
R1(config-router)# network 192.168.49.0
R1(config-router)# network 192.168.50.0
R1(config-router)# network 192.168.51.0
R1(config-router)# network 192.168.70.0

R2(config)# router rip
R2(config-router)# version 2
R2(config-router)# no auto-summary
R2(config-router)# network 172.16.0.0
```

Verify RIP entering routes from the other routers into the routing table using the **show ip route rip** command on each router. You can also verify which routes this router learns by RIP advertisements with the **show ip rip database** command.

```
R1# show ip route rip
     172.16.0.0/24 is subnetted, 4 subnets
R       172.16.23.0 [120/1] via 172.16.12.2, 00:00:03, Serial0/0/0
R       172.16.2.0 [120/1] via 172.16.12.2, 00:00:03, Serial0/0/0
```



```
R2# show ip route rip
     172.16.0.0/24 is subnetted, 4 subnets
R       172.16.1.0 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
R    192.168.51.0/24 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
R    192.168.50.0/24 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
R    192.168.49.0/24 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
R    192.168.70.0/24 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
R    192.168.48.0/24 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
```

You can also verify which routes are coming in from RIP advertisements with the **show ip rip database** command.

```
R1# show ip rip database
172.16.0.0/16    auto-summary
172.16.1.0/24    directly connected, Loopback0
172.16.2.0/24
    [1] via 172.16.12.2, 00:00:06, Serial0/0/0
172.16.12.0/24    directly connected, Serial0/0/0
172.16.23.0/24
    [1] via 172.16.12.2, 00:00:06, Serial0/0/0
192.168.48.0/24    auto-summary
192.168.48.0/24    directly connected, Loopback48
192.168.49.0/24    auto-summary
192.168.49.0/24    directly connected, Loopback49
192.168.50.0/24    auto-summary
192.168.50.0/24    directly connected, Loopback50
192.168.51.0/24    auto-summary
192.168.51.0/24    directly connected, Loopback51
192.168.70.0/24    auto-summary
192.168.70.0/24    directly connected, Loopback70

R2# show ip rip database
172.16.0.0/16    auto-summary
172.16.1.0/24
    [1] via 172.16.12.1, 00:00:10, Serial0/0/0
172.16.2.0/24    directly connected, Loopback0
172.16.12.0/24    directly connected, Serial0/0/0
172.16.23.0/24    directly connected, Serial0/0/1
192.168.48.0/24    auto-summary
192.168.48.0/24
    [1] via 172.16.12.1, 00:00:10, Serial0/0/0
192.168.49.0/24    auto-summary
192.168.49.0/24
    [1] via 172.16.12.1, 00:00:10, Serial0/0/0
192.168.50.0/24    auto-summary
192.168.50.0/24
    [1] via 172.16.12.1, 00:00:10, Serial0/0/0
192.168.51.0/24    auto-summary
192.168.51.0/24
    [1] via 172.16.12.1, 00:00:10, Serial0/0/0
192.168.70.0/24    auto-summary
192.168.70.0/24
    [1] via 172.16.12.1, 00:00:10, Serial0/0/0
```

Step 3: Configure Passive Interfaces in RIP

Look again at the RIP routes in the routing table on R1. Notice that the serial interface of R2 connecting to R3 is there, even though you do not have a RIP neighbor on that interface.


```
R1# show ip route rip
     172.16.0.0/24 is subnetted, 4 subnets
R       172.16.23.0 [120/1] via 172.16.12.2, 00:00:03, Serial0/0/0
R       172.16.2.0 [120/1] via 172.16.12.2, 00:00:03, Serial0/0/0
```

This is because the entire class B network 172.16.0.0 /16 was added to RIP on R2. If you execute the **show ip protocols** command, you can see that RIP updates are being sent out both serial interfaces.

```
R2# show ip protocols
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 13 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    Serial0/0/0           2     2
    Serial0/0/1           2     2
    Loopback0             2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.12.1          120      00:00:26
  Distance: (default is 120)
```

You do not want to send RIP updates out that serial interface toward R3 for security reasons. You can disable updates being sent with the RIP configuration command **passive-interface** *interface_type interface_number*. Disable the serial interface to R3 on R2. Observe that that interface is no longer listed under **show ip protocols** for RIP.

```
R2(config)# router rip
R2(config-router)# passive-interface serial 0/0/1

R2# show ip protocols
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 23 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    Serial0/0/0           2     2
    Loopback0             2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
  Passive Interface(s):
    Serial0/0/1
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.12.1          120      00:00:17
  Distance: (default is 120)
```


Looking at R1's routing table, notice that the network is still there from RIP.

```
R1# show ip route rip
     172.16.0.0/24 is subnetted, 4 subnets
R       172.16.23.0 [120/1] via 172.16.12.2, 00:00:19, Serial0/0/0
R       172.16.2.0 [120/1] via 172.16.12.2, 00:00:19, Serial0/0/0
```

Making an interface in RIP passive only disables updates from being sent through RIP; it does not affect interfaces being received through it.

What are some reasons you would want to disable RIP sending updates out a particular interface?

Putting a RIPv2 interface in passive mode saves the router from sending multicast RIP packets out an interface that has no neighbors.

Does RIPv2 send advertisements out loopback interfaces?

If you are unsure, monitor the output of the **debug ip rip** command to verify your answer. Place any loopbacks out of which RIPv2 is sending advertisements in passive state with the **passive-interface** command, as described previously.

```
R1(config)# router rip
R1(config-router)# passive-interface loopback 0
R1(config-router)# passive-interface loopback 48
R1(config-router)# passive-interface loopback 49
R1(config-router)# passive-interface loopback 50
R1(config-router)# passive-interface loopback 51
R1(config-router)# passive-interface loopback 70

R2(config)#router rip
R2(config-router)#passive-interface loopback 0
```

If you are running RIPv2, you should implement the use of passive interfaces as a common practice to save CPU processor cycles and bandwidth on interfaces that do not have multicast RIPv2 neighbors.

Step 4: Summarize a Supernet with RIP

Notice that you can see all prefixes from R1 in R2's routing table.

```
R2# show ip route rip
     172.16.0.0/24 is subnetted, 4 subnets
R       172.16.1.0 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
R    192.168.51.0/24 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
```



```
R    192.168.50.0/24 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
R    192.168.49.0/24 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
R    192.168.70.0/24 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
R    192.168.48.0/24 [120/1] via 172.16.12.1, 00:00:29, Serial0/0/0
```

In preparing for redistribution, you want to redistribute the minimum number of destination prefixes into each of the routing protocols.

Which RIP routes should you summarize because they are contiguous, and which mask should you use?

Under normal circumstances, you could simply summarize the four consecutive class-C networks with the **ip summary address rip** command on R1's Serial0/0/0 interface. However, the Cisco IOS does not allow you to summarize to a mask length that is less than the classful network prefix (in this case, 24 bits). If you do, you receive the following error message:

```
R1(config-if)# ip summary-address rip 192.168.48.0 255.255.252.0
 Summary mask must be greater or equal to major net
```

Recall from the EIGRP labs that summary routes display in the summarizing device's routing table as having the next hop be the Null0 interface. The routing protocol advertises these routes as pointing toward the redistributing router.

To get around the **ip summary-address rip** message error, create a static route on R1 to summarize the networks of loopbacks 48 through 51. Then redistribute the route on R1.

```
R1(config)# ip route 192.168.48.0 255.255.252.0 null0
R1(config)# router rip
R1(config-router)# redistribute static
```

This solution might seem unusual, but in fact you are modeling the internal workings of other routing protocols like EIGRP or OSPF to overcome RIP's limitations. It is helpful to understand how EIGRP and OSPF handle summary routes internally, because it can apply to other applications.

Verify with the **show ip route** command on R1 and R2 that the RIP supernet has been added to the routing table:

```
R1# show ip route
<output omitted>

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 4 subnets
R       172.16.23.0 [120/1] via 172.16.12.2, 00:00:27, Serial0/0/0
C       172.16.12.0 is directly connected, Serial0/0/0
C       172.16.1.0 is directly connected, Loopback0
```



```
R       172.16.2.0 [120/1] via 172.16.12.2, 00:00:27, Serial0/0/0
C    192.168.51.0/24 is directly connected, Loopback51
C    192.168.50.0/24 is directly connected, Loopback50
C    192.168.49.0/24 is directly connected, Loopback49
C    192.168.70.0/24 is directly connected, Loopback70
C    192.168.48.0/24 is directly connected, Loopback48
S    192.168.48.0/22 is directly connected, Null0
```

```
R2# show ip route
<output omitted>

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 4 subnets
C       172.16.23.0 is directly connected, Serial0/0/1
C       172.16.12.0 is directly connected, Serial0/0/0
R       172.16.1.0 [120/1] via 172.16.12.1, 00:00:05, Serial0/0/0
C       172.16.2.0 is directly connected, Loopback0
R    192.168.51.0/24 [120/1] via 172.16.12.1, 00:00:05, Serial0/0/0
R    192.168.50.0/24 [120/1] via 172.16.12.1, 00:00:05, Serial0/0/0
R    192.168.49.0/24 [120/1] via 172.16.12.1, 00:00:05, Serial0/0/0
R    192.168.70.0/24 [120/1] via 172.16.12.1, 00:00:07, Serial0/0/0
R    192.168.48.0/24 [120/1] via 172.16.12.1, 00:00:07, Serial0/0/0
R    192.168.48.0/22 [120/1] via 172.16.12.1, 00:00:07, Serial0/0/0
```

Will this route to Null0 affect routing to prefixes with longer addresses on R1? Explain.

Step 5: Suppress Routes Using Prefix Lists

Sometimes you may not want to advertise certain networks out a particular interface, or you may want to filter updates as they come in. This is possible with some routing protocols, such as RIP or EIGRP. However, link-state protocols are less flexible, because every router in an area is required to have a synchronized database as a condition for full adjacency.

In this scenario, you want to filter updates from R1 to R2, allowing only the networks Loopback 0 and Loopback 70 and the summary route to be advertised. Suppress the more specific prefixes so that routing tables are kept small, and CPU processor cycles on the routers are not wasted.

Distribute lists use either access lists or prefix lists to filter routes by network address. They can also be configured to filter subnet masks. You can only use standard access lists to filter for the network address of the destination network without regard to subnet address. In this scenario, you have two networks with the same destination network address: 192.168.48.0. The 22-bit summary and the 24-bit major network address both have the same address, so standard access lists will not accomplish the filtering correctly. Prefix lists or extended access lists are appropriate workarounds. On R1, use a prefix list as a distribution filter to prevent the more specific routes to Loopbacks 48 through 51. Allow all other destination networks including the summary route.

```
Line 1:   ip prefix-list RIP-OUT permit 192.168.48.0/22
Line 2:   ip prefix-list RIP-OUT deny   192.168.48.0/22 le 24
Line 3:   ip prefix-list RIP-OUT permit 0.0.0.0/0 le 32
```

Line 1 of the prefix list permits the summary route and nothing else, because no other route can match that network address with a mask of exactly 22 bits. Line 2 denies all prefixes with a network address in the 192.168.48.0/22 block of addresses that have subnet masks from 22 bits to 24 bits. This removes exactly four network addresses matching both 22, 23, and 24 bits in length of the subnet mask. Line 2 would deny the 192.168.48.0/22 summary route you created if Line 1 did not explicitly permit the summary route. Line 3 allows all IPv4 prefixes that are not explicitly denied in previous statements of the prefix list.

Apply this access list with the **distribute-list** command from the RIP configuration prompt on R1.

```
R1(config)# router rip
R1(config-router)# distribute-list prefix RIP-OUT out serial0/0/0
```

Verify that the filtering has taken place using the **show ip route rip** and **show ip rip database** commands on R2.

```
R2# show ip route rip
     172.16.0.0/24 is subnetted, 4 subnets
R       172.16.1.0 [120/1] via 172.16.12.1, 00:00:12, Serial0/0/0
R    192.168.70.0/24 [120/1] via 172.16.12.1, 00:00:12, Serial0/0/0
R    192.168.48.0/22 [120/1] via 172.16.12.1, 00:00:12, Serial0/0/0


R2# show ip rip database
172.16.0.0/16    auto-summary
172.16.1.0/24
    [1] via 172.16.12.1, 00:00:11, Serial0/0/0
172.16.2.0/24    directly connected, Loopback0
172.16.12.0/24    directly connected, Serial0/0/0
172.16.23.0/24    directly connected, Serial0/0/1
192.168.48.0/22
    [1] via 172.16.12.1, 00:00:11, Serial0/0/0
192.168.70.0/24    auto-summary
192.168.70.0/24
    [1] via 172.16.12.1, 00:00:11, Serial0/0/0
```

Why would you want to filter updates getting sent out or coming in?

Step 6: Configure OSPF

Configure single-area OSPF between R2 and R3. On R2, include just the serial link connecting to R3. On R3, include the serial link and all loopback interfaces. Make sure that you change the network type for the loopback interfaces. Verify that your adjacencies come up with the **show ip ospf neighbors** command. Also make sure that you have routes from OSPF populating the routing tables with the **show ip route ospf** command.

```
R2(config)# router ospf 1
R2(config-router)# network 172.16.23.0 0.0.0.255 area 0
```

```
R3(config)# router ospf 1
R3(config-router)# network 172.16.0.0 0.0.255.255 area 0
R3(config-router)# network 192.168.0.0 0.0.255.255 area 0
```

```
R2# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.40.1      0   FULL/  -        00:00:37    172.16.23.3     Serial0/0/1
```

```
R3# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
172.16.2.1        0   FULL/  -        00:00:39    172.16.23.2     Serial0/0/1
```

```
R2# show ip route ospf
     192.168.30.0/32 is subnetted, 1 subnets
O       192.168.30.1 [110/65] via 172.16.23.3, 00:04:41, Serial0/0/1
     192.168.25.0/32 is subnetted, 1 subnets
O       192.168.25.1 [110/65] via 172.16.23.3, 00:04:41, Serial0/0/1
     192.168.40.0/32 is subnetted, 1 subnets
O       192.168.40.1 [110/65] via 172.16.23.3, 00:04:41, Serial0/0/1
     172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
O       172.16.3.1/32 [110/65] via 172.16.23.3, 00:00:20, Serial0/0/1
     192.168.20.0/32 is subnetted, 1 subnets
O       192.168.20.1 [110/65] via 172.16.23.3, 00:04:41, Serial0/0/1
     192.168.35.0/32 is subnetted, 1 subnets
O       192.168.35.1 [110/65] via 172.16.23.3, 00:04:41, Serial0/0/1
```

```
R3# show ip route ospf

R3# ! note that the above output is blank
```

The **network 192.168.0.0 0.0.255.255 area 0** command allows OSPF to involve interfaces that have IP addresses in that range.

A common misconception is that OSPF advertises the entire range of the network given in the router's network statement; it certainly does not. However, it does advertise any connected subnets in that entire range of addresses to adjacent routers. You can verify this by viewing the output of the **show ip route** command on R2. Do you see a 192.168.0.0/16 supernet?

R2 is the only router with all routes in the topology (except for those that were filtered out), because it is involved with both routing protocols.

Step 7: Configure Passive Interfaces in OSPF

As discussed before, passive interfaces save CPU cycles, router memory, and link bandwidth by preventing broadcast/multicast routing updates on interfaces that have no neighbors. In link-state protocols, adjacencies must be formed before routers exchange routing information. The **passive-interface** command in OSPF configuration mode prevents an interface from sending multicast Hello packets out that interface.

OSPF included R3's loopback interfaces in its network statements shown in Step 6.

On R3, configure Loopback0 as a passive interface in OSPF. At the OSPF configuration prompt, use the **passive-interface** *interface_type interface_number* command.

```
R3(config-router)# passive-interface loopback 0
```

How is this different from the RIP version of this command?

The Cisco IOS provides a quick way of selecting interfaces for passive mode. Use the **passive-interface default** command to make all interfaces passive. Then use the **no passive-interface** *interface interface_number* command to bring the Serial0/0/1 interface out of passive mode.

```
R3(config)# router ospf 1
R3(config-router)# passive-interface default
R3(config-router)#
*Oct 15 01:49:44.174: %OSPF-5-ADJCHG: Process 1, Nbr 172.16.2.1 on Serial0/0/1
from FULL to DOWN, Neighbor Down: Interface down or detached
R3(config-router)# no passive-interface serial 0/0/1
R3(config-router)#
*Oct 15 01:49:55.438: %OSPF-5-ADJCHG: Process 1, Nbr 172.16.2.1 on Serial0/0/1
from LOADING to FULL, Loading Done
```

You can verify the application of this command by issuing the **show ip protocols** command.

```
R3# show ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.40.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
```



```
    172.16.0.0 0.0.255.255 area 0
    192.168.0.0 0.0.255.255 area 0
 Reference bandwidth unit is 100 mbps
  Passive Interface(s):
    FastEthernet0/0
    FastEthernet0/1
    Serial0/0/0
    Serial0/1/0
    Serial0/1/1
    Loopback0
    Loopback20
    Loopback25
    Loopback30
    Loopback35
    Loopback40
    VoIP-Null0
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 110)
```

Step 8: Allow One-way Redistribution

On R2, configure OSPF to redistribute into RIP under the RIP configuration prompt with the **redistribute ospf** *process* **metric** *metric* command, where *process* is the OSPF process number, and *metric* is the default metric with which you want to originate the routes into RIP. If you do not specify a default metric in RIP, it gives routes an infinite metric and they are not advertised.

```
R2(config)# router rip
R2(config-router)# redistribute ospf 1 metric 4
```

Verify the redistribution with the **show ip protocols** command:

```
R2# show ip protocols
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 24 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip, ospf 1
  Default version control: send version 2, receive version 2
    Interface             Send  Recv  Triggered RIP  Key-chain
    Serial0/0/0           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
  Passive Interface(s):
    Serial0/0/1
    Loopback0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.12.1          120      00:00:19
  Distance: (default is 120)
...
<output omitted>
```

If you look at the routing table on R1 with the **show ip route** command, you see that it has all the routes in the topology. However, pinging a loopback on R3 from R1 shows that R1 has a route to R3, but R3 does not have a route back to R1. You can verify this with the **traceroute** command on R1.

```
R1# show ip route rip
     192.168.30.0/32 is subnetted, 1 subnets
R       192.168.30.1 [120/4] via 172.16.12.2, 00:00:02, Serial0/0/0
     192.168.25.0/32 is subnetted, 1 subnets
R       192.168.25.1 [120/4] via 172.16.12.2, 00:00:02, Serial0/0/0
     192.168.40.0/32 is subnetted, 1 subnets
R       192.168.40.1 [120/4] via 172.16.12.2, 00:00:02, Serial0/0/0
     172.16.0.0/24 is subnetted, 4 subnets
R       172.16.23.0 [120/1] via 172.16.12.2, 00:00:02, Serial0/0/0
R       172.16.2.0 [120/1] via 172.16.12.2, 00:00:02, Serial0/0/0
R       172.16.3.1/32 [120/4] via 172.16.12.2, 00:00:24, Serial0/0/0
     192.168.20.0/32 is subnetted, 1 subnets
R       192.168.20.1 [120/4] via 172.16.12.2, 00:00:02, Serial0/0/0
     192.168.35.0/32 is subnetted, 1 subnets
R       192.168.35.1 [120/4] via 172.16.12.2, 00:00:02, Serial0/0/0

R1# ping 192.168.30.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.30.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R1# traceroute 192.168.30.1

Type escape sequence to abort.
Tracing the route to 192.168.30.1

  1 172.16.12.2 12 msec 12 msec 16 msec
  2  *  *  *
  3  *  *  *
  4  *  *  *
<remaining output omitted>
```

To alleviate this problem, you can originate a default route into OSPF that points toward R2 so that the pings are routed back toward R2. R2 uses its information from RIPv2 to send pings back to R1.

Issue the **default-information originate always** command under the OSPF configuration prompt to force R2 to advertise a default route in OSPF. Verify that this route shows up in R3's routing table.

```
R2(config)# router ospf 1
R2(config-router)# default-information originate always

R3# show ip route ospf
O*E2 0.0.0.0/0 [110/1] via 172.16.23.2, 00:05:13, Serial0/0/1
```

You should now have full connectivity between all networks in the diagram. Try using the TCL script and comparing it with the output shown in Appendix A (all successful).

Step 9: Redistribute Between Two Routing Protocols

We can substitute this default route in with actual, more specific routes. First, take away the default route advertisement with the **no default-information originate always** command under the OSPF configuration prompt on R2. Next, use the **redistribute rip** command. You do not need to specify a default metric in OSPF. Notice the warning.

```
R2(config)# router ospf 1
R2(config-router)# no default-information originate always
R2(config-router)# redistribute rip
% Only classful networks will be redistributed
```

If you display the routing table on R3, the only external OSPF route that came in was the 192.168.70.0 /24 network.

```
R3# show ip route ospf
O E2 192.168.70.0/24 [110/20] via 172.16.23.2, 00:00:51, Serial0/0/1
O E2 192.168.48.0/22 [110/20] via 172.16.23.2, 00:00:51, Serial0/0/1
```

This is because, by default, OSPF only accepts classful networks when redistributing into it. The only classful network coming into R2 from RIP is the class C network 192.168.70.0. You can modify this behavior by adding the **subnets** keyword to the **redistribute** command. Verify this with the **show ip route ospf** command on R3.

```
R2(config)# router ospf 1
R2(config-router)# redistribute rip subnets

R3# show ip route ospf
     172.16.0.0/24 is subnetted, 5 subnets
O E2    172.16.12.0 [110/20] via 172.16.23.2, 00:00:01, Serial0/0/1
O E2    172.16.1.0 [110/20] via 172.16.23.2, 00:00:01, Serial0/0/1
O E2    172.16.2.0 [110/20] via 172.16.23.2, 00:00:01, Serial0/0/1
O E2 192.168.70.0/24 [110/20] via 172.16.23.2, 00:04:19, Serial0/0/1
O E2 192.168.48.0/22 [110/20] via 172.16.23.2, 00:04:19, Serial0/0/1
```

You should again have full connectivity between all networks in the diagram. Run the TCL script from each router. Verify your output against the output in Appendix A (all pings successful).

Step 10: Set a Default Seed Metric

Under any routing protocol, you can specify a default seed metric to be used for redistribution, instead of or in addition to setting metrics on a per-protocol basis. Seed metrics is a protocol-independent feature of the Cisco IOS software that is usually used when redistributing into distance-vector protocols.

Notice that the metric listed in the R3 routing table shown above is 20.

On R2, under the OSPF configuration prompt, issue the **default-metric** *metric* command to configure a default metric for redistributed routes. You can override the global creation of a default seed metric on a per-protocol basis by using the **metric** argument in a redistribution command. You can also use the **metric** command under other routing protocols. Verify the new metric in R3's routing table. It may take a little while for the new metric to propagate.

```
R2(config)# router ospf 1
R2(config-router)# default-metric 10000

R3# show ip route ospf
     172.16.0.0/24 is subnetted, 5 subnets
O E2    172.16.12.0 [110/10000] via 172.16.23.2, 00:02:56, Serial0/0/1
O E2    172.16.1.0 [110/10000] via 172.16.23.2, 00:02:56, Serial0/0/1
O E2    172.16.2.0 [110/10000] via 172.16.23.2, 00:02:56, Serial0/0/1
O E2 192.168.70.0/24 [110/10000] via 172.16.23.2, 00:02:56, Serial0/0/1
O E2 192.168.48.0/22 [110/10000] via 172.16.23.2, 00:02:56, Serial0/0/1
```

Step 11: Change the OSPF External Network Type

In this last step, take a look at R3's routing table. Notice that the external (redistributed) routes have O E2 as their type. Also notice that the metric is exactly the same as the seed metric given in the previous step. O means OSPF, and E2 means external, type 2. In OSPF, there are two external metric types, and E2 is the default. External type 1 metrics increase like a usual route, whereas external type 2 metrics do not increase as they get advertised through the OSPF domain.

Where would an external type 1 metric be useful?

Where would an external type 2 metric be useful?

You can change this type using the **metric-type** argument with the **redistribute** command. Change it to type 1 for RIP redistributed routes, and then display R3's routing table again.

```
R2(config)# router ospf 1
R2(config-router)# redistribute rip sub metric-type 1

R3# show ip route ospf
     172.16.0.0/24 is subnetted, 5 subnets
O E1    172.16.12.0 [110/10064] via 172.16.23.2, 00:03:05, Serial0/0/1
O E1    172.16.1.0 [110/10064] via 172.16.23.2, 00:03:05, Serial0/0/1
O E1    172.16.2.0 [110/10064] via 172.16.23.2, 00:03:05, Serial0/0/1
O E1 192.168.70.0/24 [110/10064] via 172.16.23.2, 00:03:05, Serial0/0/1
O E1 192.168.48.0/22 [110/10064] via 172.16.23.2, 00:03:05, Serial0/0/1
```

Which attributes of the routes changed?

Challenge: Use Extended Access Lists for Filtering

On R1, configure a distribute list to filter 192.168.20.0 /24 and 192.168.25.0 /27 from inbound updates from R2. Pay special attention to the subnet masks. Do not filter out 192.168.25.0 /24. Use an extended access list to accomplish this. Refer to Step 5 for more details.

Appendix A: TCL Script Output – Steps 8 and 9

```
R1# tclsh
R1(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#192.168.48.1
+>(tcl)#192.168.49.1
+>(tcl)#192.168.50.1
+>(tcl)#192.168.51.1
+>(tcl)#192.168.70.1
+>(tcl)#172.16.12.1
+>(tcl)#172.16.2.1
+>(tcl)#172.16.12.2
+>(tcl)#172.16.23.2
+>(tcl)#172.16.3.1
+>(tcl)#192.168.20.1
+>(tcl)#192.168.25.1
+>(tcl)#192.168.30.1
+>(tcl)#192.168.35.1
+>(tcl)#192.168.40.1
+>(tcl)#172.16.23.3
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.48.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.49.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.50.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.51.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.70.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.1, timeout is 2 seconds:
!!!!!
```



```
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.25.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.30.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.35.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.40.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
R1(tcl)# tclquit

R2# tclsh
R2(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#192.168.48.1
+>(tcl)#192.168.49.1
+>(tcl)#192.168.50.1
+>(tcl)#192.168.51.1
+>(tcl)#192.168.70.1
+>(tcl)#172.16.12.1
+>(tcl)#172.16.2.1
+>(tcl)#172.16.12.2
+>(tcl)#172.16.23.2
+>(tcl)#172.16.3.1
+>(tcl)#192.168.20.1
+>(tcl)#192.168.25.1
+>(tcl)#192.168.30.1
+>(tcl)#192.168.35.1
+>(tcl)#192.168.40.1
+>(tcl)#172.16.23.3
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
```



```
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.48.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.49.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.50.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.51.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.70.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/64 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.25.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.30.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.35.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.40.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
```



```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R2(tcl)# tclquit

R3# tclsh
R3(tcl)#foreach address {
+>(tcl)#172.16.1.1
+>(tcl)#192.168.48.1
+>(tcl)#192.168.49.1
+>(tcl)#192.168.50.1
+>(tcl)#192.168.51.1
+>(tcl)#192.168.70.1
+>(tcl)#172.16.12.1
+>(tcl)#172.16.2.1
+>(tcl)#172.16.12.2
+>(tcl)#172.16.23.2
+>(tcl)#172.16.3.1
+>(tcl)#192.168.20.1
+>(tcl)#192.168.25.1
+>(tcl)#192.168.30.1
+>(tcl)#192.168.35.1
+>(tcl)#192.168.40.1
+>(tcl)#172.16.23.3
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.48.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.49.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.50.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.51.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.70.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.2, timeout is 2 seconds:
```



```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.20.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.25.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.30.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.35.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.40.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
R3(tcl)# tclquit
```

Final Configurations

```
R1# show run
!
hostname R1
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
!
interface Loopback48
 ip address 192.168.48.1 255.255.255.0
!
interface Loopback49
 ip address 192.168.49.1 255.255.255.0
!
interface Loopback50
 ip address 192.168.50.1 255.255.255.0
!
interface Loopback51
 ip address 192.168.51.1 255.255.255.0
!
interface Loopback70
 ip address 192.168.70.1 255.255.255.0
!
interface Serial0/0/0
 ip address 172.16.12.1 255.255.255.0
 clock rate 64000
 no shutdown
!
router rip
```



```
 version 2
 redistribute static metric 1
 passive-interface Loopback0
 passive-interface Loopback48
 passive-interface Loopback49
 passive-interface Loopback50
 passive-interface Loopback51
 passive-interface Loopback70
 network 172.16.0.0
 network 192.168.48.0
 network 192.168.49.0
 network 192.168.50.0
 network 192.168.51.0
 network 192.168.70.0
 distribute-list prefix 100 out Serial0/0/0
!
ip route 192.168.48.0 255.255.252.0 Null0
!
ip prefix-list 100 seq 5 permit 192.168.48.0/22
ip prefix-list 100 seq 10 deny 192.168.48.0/22 le 32
ip prefix-list 100 seq 15 permit 0.0.0.0/0 le 32
!
end

R2# show run
!
hostname R2
!
interface Loopback0
 ip address 172.16.2.1 255.255.255.0
!
interface Serial0/0/0
 ip address 172.16.12.2 255.255.255.0
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.23.2 255.255.255.0
 clock rate 2000000
 no shutdown
!
router ospf 1
 redistribute rip metric-type 1 subnets
 network 172.16.23.0 0.0.0.255 area 0
 default-information originate
 default-metric 10000
!
router rip
 version 2
 redistribute ospf 1 metric 4
 passive-interface Serial0/0/1
 passive-interface Loopback0
 network 172.16.0.0
 no auto-summary
!
end

R3# show run
!
hostname R3
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
```



```
interface Loopback20
 ip address 192.168.20.1 255.255.255.0
!
interface Loopback25
 ip address 192.168.25.1 255.255.255.0
!
interface Loopback30
 ip address 192.168.30.1 255.255.255.0
!
interface Loopback35
 ip address 192.168.35.1 255.255.255.0
!
interface Loopback40
 ip address 192.168.40.1 255.255.255.0
!
interface Serial0/0/1
 ip address 172.16.23.3 255.255.255.0
 no shutdown
!
router ospf 1
 passive-interface default
 no passive-interface Serial0/0/1
 network 172.16.23.0 0.0.0.255 area 0
 network 172.16.0.0 0.0.255.255 area 0
 network 192.168.0.0 0.0.255.255 area 0
!
end
```

