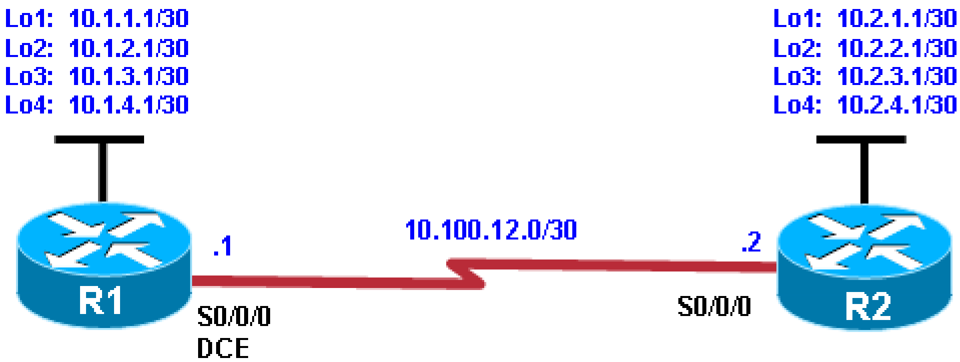
Cisco | Networking Academy®
Mind Wide Open™

CCNP ROUTE 6.0

Student Lab Manual

Chapter 1 Lab 1-1, Tcl Script Reference and Demonstration

Topology



Objectives

- Use Tcl scripts to verify full connectivity.
- Identify causes of failures.

Background

The Cisco IOS Scripting feature provides the ability to run Tool Command Language (Tcl) commands from the Cisco IOS command-line interface (CLI). Tcl scripts can be created to accomplish routine and repetitive functions with Cisco IOS-based networking devices. In this lab, you create and execute a Tcl script that sends pings to multiple IP addresses in the network to test overall network connectivity.

Note: Cisco IOS Release 12.3(2)T and later supports Tcl scripting.

Required Resources

- 2 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Service or comparable)
- Serial and console cables

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the advanced IP image c1841-advipservicesk9-mz.124-24.T1.bin. Other routers (such as a 2801 or 2811) and Cisco IOS Software versions can be used if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Step 1: Configure initial settings.

Copy and paste the following initial configurations for R1 and R2.

Router R1

```
hostname R1
!
interface loopback 1
 ip address 10.1.1.1 255.255.255.252
!
interface loopback 2
 ip address 10.1.2.1 255.255.255.252
!
interface loopback 3
 ip address 10.1.3.1 255.255.255.252
!
interface loopback 4
 ip address 10.1.4.1 255.255.255.252
!
interface serial 0/0/0
 ip address 10.100.12.1 255.255.255.252
 clock rate 64000
 bandwidth 64
 no shutdown
!
router rip
 version 2
 network 10.0.0.0
 no auto-summary
!
end
```

Note: A 30-bit subnet mask (255.255.255.252) is used for the serial links in this lab. However, starting with IOS 12.2(4)T, the 31-bit subnet mask (255.255.255.254) is supported on IPv4 point-to-point interfaces (per RFC 3021), requiring only 2 IP addresses per point-to-point link (.0 and .1). The IP Unnumbered feature can also be used to conserve IP addresses.

Router R2

```
hostname R2
!
interface loopback 1
 ip address 10.2.1.1 255.255.255.252
!
interface loopback 2
 ip address 10.2.2.1 255.255.255.252
!
interface loopback 3
 ip address 10.2.3.1 255.255.255.252
!
interface loopback 4
 ip address 10.2.4.1 255.255.255.252
!
interface serial 0/0/0
 bandwidth 64
```



```
 no shutdown
!
router rip
 version 2
 network 10.0.0.0
 no auto-summary
!
end
```

Do you think that these configurations will achieve full connectivity between R1 and R2? Explain.

____

____

____

____

____

____

Step 2: Verify connectivity.

The simplest way to verify OSI Layer 3 connectivity between two routers is to use ICMP. ICMP defines a number of message types in RFC 792 for IPv4 and RFC 4443 for IPv6. (See www.ietf.org and http://tools.ietf.org for more information.)

ICMP defines procedures for echo (ping), traceroute, and source notification of unreachable networks. Pinging an IP address can result in a variety of ICMP messages, but the only message indicating that a ping is successful is the ICMP echo reply message indicated by an exclamation point (!) in the output of the **ping** command. The following command on R1 pings its Lo1 interface. Loopback interfaces always have a status of UP/UP.

```
R1# ping 10.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

In Step 1, you might have noticed that the R2 configuration omits an IP address on serial 0/0/0. R2 does not exchange IP packets with R1 because the IP protocol is not running on the R2 serial interface until the IP address has been configured.

Without this IP address, for which addresses in the topology diagram do you expect the ping to fail?

____

____

____

____

____

Step 3: Create and execute a Tcl script.

Tcl scripts can be created to accomplish routine and repetitive functions with Cisco IOS-based networking devices. To construct a simple connectivity verification script, do the following.

a. Open a text editor and create a new text file. Using a text file saves time, especially if you are pasting the Tcl script into multiple devices.

b. Start with the **tclsh** command to enter Tcl shell mode in which you can use native Tcl instructions like **foreach** or issue EXEC mode commands. You can also access configuration mode from within the Tcl shell and issue configuration commands from their respective menus, although these features are not explored in this lab.

```
R1# tclsh
R1(tcl)#
```

c. Begin a loop using the **foreach** instruction. The loop iterates over a sequence of values, executing a defined sequence of instructions once *for each* value. Think of it as "for each *value* in *Values*, do each *instruction* in *Instructions*." For each iteration of the loop, *$identifier* reflects the current *value* in *Values*. The **foreach** instruction uses the following model.

```
foreach identifier {
value1
value2
.
.
.
valueX
} {
instruction1
instruction2
.
.
.
instructionY
}
```

d. To create a Tcl script that pings every IP address in the topology, enter each IP address in the value list. Issue the **ping $address** command as the only instruction in the instruction list.

```
foreach address {
 10.1.1.1
 10.1.2.1
 10.1.3.1
 10.1.4.1
 10.100.12.1
 10.2.1.1
 10.2.2.1
 10.2.3.1
 10.2.4.1
 10.100.12.2
} {
 ping $address
}
```

e. Enter Tcl mode with the **tclsh** command, and copy the Tcl script from the text file and paste it into R1.

```
R1# tclsh

R1(tcl)#foreach address {
```



```
+>(tcl)#10.1.1.1
+>(tcl)#10.1.2.1
+>(tcl)#10.1.3.1
+>(tcl)#10.1.4.1
+>(tcl)#10.100.12.1
+>(tcl)#10.2.1.1
+>(tcl)#10.2.2.1
+>(tcl)#10.2.3.1
+>(tcl)#10.2.4.1
+>(tcl)#10.100.12.2
+>(tcl)#} {
+>(tcl)#ping $address
+>(tcl)#}
```

Note: You might need to press Enter to execute the script.

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, 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 10.1.2.1, 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 10.1.3.1, 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 10.1.4.1, 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 10.100.12.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.4.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.12.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

f. Enter Tcl mode with the **`tclsh`** command, and copy the Tcl script from the text file and paste it into R2.


```
R2# tclsh

R2(tcl)#foreach address {
+>(tcl)#10.1.1.1
+>(tcl)#10.1.2.1
+>(tcl)#10.1.3.1
+>(tcl)#10.1.4.1
+>(tcl)#10.100.12.1
+>(tcl)#10.2.1.1
+>(tcl)#10.2.2.1
+>(tcl)#10.2.3.1
+>(tcl)#10.2.4.1
+>(tcl)#10.100.12.2
+>(tcl)#} {
+>(tcl)#ping $address
+>(tcl)#}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.2.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.3.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.4.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.100.12.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.1.1, 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 10.2.2.1, 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 10.2.3.1, 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 10.2.4.1, 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 10.100.12.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```


g. Exit Tcl mode using the **`tclquit`** command on each device.

```
R1(tcl)#tclquit
```

Note: You can also use the **`exit`** command to exit Tcl mode.

Notice that in the previous output, R1 and R2 could not route pings to the remote loopback networks for which they did not have routes installed in their routing tables.

You might have also noticed that R1 could not ping its local address on serial 0/0/0. This is because with PPP, HDLC, Frame Relay, and ATM serial technologies, all packets, including pings to the local interface, must be forwarded across the link.

For instance, R1 attempts to ping 10.100.12.1 and routes the packet out serial 0/0/0, even though the address is a local interface. Assume that an IP address of 10.100.12.2/30 is assigned to the serial 0/0/0 interface on R2. When a ping from R1 to 10.100.12.1 reaches R2, R2 evaluates that this is not its address on the 10.100.12.0/30 subnet and routes the packet back to R1 using its serial 0/0/0 interface. R1 receives the packet and evaluates that 10.100.12.1 is the address of the local interface. R1 opens this packet using ICMP, and responds to the ICMP echo request (ping) with an echo reply destined for 10.100.12.1. R1 encapsulates the echo reply at serial 0/0/0 and routes the packet to R2. R2 receives the packet and routes it back to R1, the originator of the ICMP echo. The ICMP protocol on R1 receives the echo reply, associates it with the ICMP echo that it sent, and displays the output in the form of an exclamation point.

Note: To understand this behavior, you can observe the output of the **`debug ip icmp`** and **`debug ip packet`** commands on R1 and R2 while pinging with the configurations provided in Step 1.

Step 4: Resolve connectivity issues.

a. On R2, assign the IP address 10.100.12.2/30 to serial 0/0/0.

```
R2# conf t
R2(config)# interface serial 0/0/0
R2(config-if)# ip address 10.100.12.2 255.255.255.252
```

b. On each router, verify the receipt of RIPv2 routing information with the **`show ip protocols`** command.

```
R1# 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 28 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
    Loopback1             2     2
    Loopback2             2     2
    Loopback3             2     2
    Loopback4             2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.100.12.2          120      00:00:13
  Distance: (default is 120)
```



```
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 26 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
    Loopback1             2     2
    Loopback2             2     2
    Loopback3             2     2
    Loopback4             2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.100.12.1          120      00:00:14
  Distance: (default is 120)
```

c. On each router, verify full connectivity to all subnets in the diagram by issuing the **tclsh** command and pasting the Tcl script on the command line in privileged EXEC mode.

```
R1# tclsh

R1(tcl)#foreach address {
+>(tcl)#10.1.1.1
+>(tcl)#10.1.2.1
+>(tcl)#10.1.3.1
+>(tcl)#10.1.4.1
+>(tcl)#10.100.12.1
+>(tcl)#10.2.1.1
+>(tcl)#10.2.2.1
+>(tcl)#10.2.3.1
+>(tcl)#10.2.4.1
+>(tcl)#10.100.12.2
+>(tcl)#} {
+>(tcl)#ping $address
+>(tcl)#}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, 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 10.1.2.1, 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 10.1.3.1, 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 10.1.4.1, 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 10.100.12.1, 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 10.2.1.1, 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 10.2.2.1, 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 10.2.3.1, 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 10.2.4.1, 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 10.100.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
R1(tcl)#tclquit
```

Notice that the average round-trip time for an ICMP packet from R1 to 10.100.12.1 is approximately twice that of a packet from R1 to loopback1 on R2. This verifies the conclusion reached in Step 3 that the ICMP echo request to 10.100.12.1 and the ICMP echo reply from 10.100.12.1 each traverse the link *twice* to verify full connectivity across the link.

```
R2# tclsh

R2(tcl)#foreach address {
+>(tcl)#10.1.1.1
+>(tcl)#10.1.2.1
+>(tcl)#10.1.3.1
+>(tcl)#10.1.4.1
+>(tcl)#10.100.12.1
+>(tcl)#10.2.1.1
+>(tcl)#10.2.2.1
+>(tcl)#10.2.3.1
+>(tcl)#10.2.4.1
+>(tcl)#10.100.12.2
+>(tcl)#} {
+>(tcl)#ping $address
+>(tcl)#}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, 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 10.1.2.1, 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 10.1.3.1, 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 10.1.4.1, 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 10.100.12.1, 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 10.2.1.1, 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 10.2.2.1, 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 10.2.3.1, 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 10.2.4.1, 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 10.100.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/68 ms
R2(tcl)#tclquit
```

Notice also that the average round-trip time for an ICMP packet from R2 to 10.100.12.2 is approximately twice that of a packet from R2 to loopback1 on R1.

Conclusion

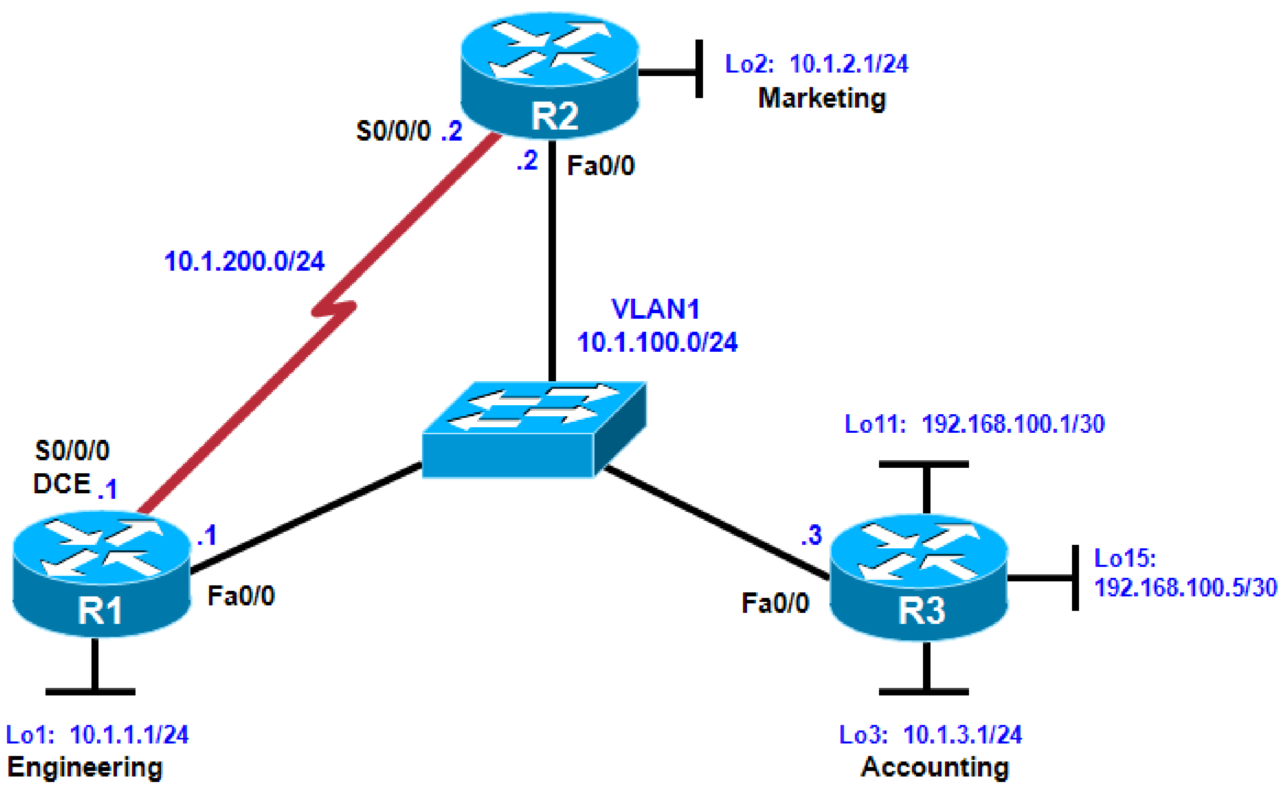
The creation of Tcl scripts takes a little extra time initially but can save considerable time during testing each time the script is executed. Use Tcl scripts to verify all your configurations in this course. If you verify your work, both academically and in production networks, you will gain knowledge and save time in troubleshooting.

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all the combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. An example of this is an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 2 Lab 2-1, EIGRP Configuration, Bandwidth, and Adjacencies

Topology



Objectives

- Configure EIGRP on multiple routers.
- Configure the **bandwidth** command to modify the EIGRP metric.
- Verify EIGRP adjacencies.
- Verify EIGRP routing information exchange.
- Use debugging commands for troubleshooting EIGRP.
- (Challenge) Test convergence for EIGRP when a topology change occurs.

Background

You are responsible for configuring a new network to connect your company's Engineering, Marketing, and Accounting departments, represented by the loopback interfaces on each of the three routers. The physical devices have just been installed and are connected by Fast Ethernet and serial interfaces. Your task is to configure EIGRP to enable full connectivity between all departments.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. The switch is a Cisco WS-C2960-24TT-L with the Cisco IOS

image c2960-lanbasek9-mz.122-46.SE.bin. You can use other routers (such as 2801 or 2811), switches (such as 2950), and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- 1 switch (Cisco 2960 with the Cisco IOS Release 12.2(46)SE C2960-LANBASEK9-M image or comparable)
- Serial and Ethernet cables

Step 1: Configure addressing and loopbacks.

a. Using the addressing scheme in the diagram, apply IP addresses to the Fast Ethernet interfaces on R1, R2, and R3. Then create Loopback1 on R1, Loopback2 on R2, and Loopback3 on R3 and address them according to the diagram.

```
R1# configure terminal
R1(config)# interface Loopback1
R1(config-if)# description Engineering Department
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# exit
R1(config)# interface FastEthernet0/0
R1(config-if)# ip address 10.1.100.1 255.255.255.0
R1(config-if)# no shutdown

R2# configure terminal
R2(config)# interface Loopback2
R2(config-if)# description Marketing Department
R2(config-if)# ip address 10.1.2.1 255.255.255.0
R2(config-if)# exit
R2(config)# interface FastEthernet0/0
R2(config-if)# ip address 10.1.100.2 255.255.255.0
R2(config-if)# no shutdown

R3# configure terminal
R3(config)# interface Loopback3
R3(config-if)# description Accounting Department
R3(config-if)# ip address 10.1.3.1 255.255.255.0
R3(config-if)# exit
R3(config)# interface FastEthernet0/0
R3(config-if)# ip address 10.1.100.3 255.255.255.0
R3(config-if)# no shutdown
```

Leave the switch in its default (blank) configuration. By default, all switch ports are in VLAN1 and are not administratively down.

Note: If the switch has been previously configured, erase the startup config, delete the vlan.dat file from flash memory, and reload the switch.

For now, also leave the serial interfaces in their default configuration. You will configure the serial link between R1 and R2 in Step 4.

b. Verify that the line protocol of each interface is up and that you can successfully ping across each link. You should see output similar to the following on each router.

```
R1# show ip interface brief
```



```
Interface              IP-Address      OK? Method Status
Protocol
FastEthernet0/0        10.1.100.1      YES manual up                    up
FastEthernet0/1        unassigned      YES unset  administratively down down
Serial0/0/0            unassigned      YES manual administratively down down
Serial0/0/1            unassigned      YES unset  administratively down down
Loopback1              10.1.1.1        YES manual up                    up
```

Step 2: Configure EIGRP on the Ethernet network.

a. After you have implemented your addressing scheme, create an EIGRP autonomous system (AS) on R1 using the following commands in global configuration mode.

```
R1(config)# router eigrp 1
R1(config-router)# network 10.0.0.0
R1(config-router)# no auto-summary
```

Using network statements with major networks causes EIGRP to begin sending EIGRP hello packets out all interfaces in that network (that is, subnets of the major network 10.0.0.0/8). In this case, EIGRP should start sending hello packets out of its FastEthernet0/0 and Loopback1 interfaces.

b. To check if this is occurring, use the **debug eigrp packets** command in privileged EXEC mode.

```
R1# debug eigrp packets
EIGRP Packets debugging is on
(UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB, SIAQUERY,
SIAREPLY)
R1#
*Feb  3 16:54:43.555: EIGRP: Sending HELLO on FastEthernet0/0
*Feb  3 16:54:43.555:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
*Feb  3 16:54:43.995: EIGRP: Sending HELLO on Loopback1
*Feb  3 16:54:43.995:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
*Feb  3 16:54:43.995: EIGRP: Received HELLO on Loopback1 nbr 10.1.1.1
*Feb  3 16:54:43.995:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0
*Feb  3 16:54:43.995: EIGRP: Packet from ourselves ignored
```

The hello packets are unanswered by the other routers because EIGRP is not yet running on R2 or R3. R1 ignores the hello packets from itself on Loopback1.

c. Use the **undebug all** command to stop the debug output.

```
R1# undebug all
```

d. Use the **show ip eigrp interfaces** command to display the interfaces that are participating in EIGRP.

```
R1# show ip eigrp interfaces
IP-EIGRP interfaces for process 1

                        Xmit Queue   Mean   Pacing Time   Multicast
Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Fa0/0              0        0/0         0       0/1            0           0
Lo1                0        0/0         0       0/1            0           0
```

Which interfaces are involved in the EIGRP routing process on this router?

____

To monitor the EIGRP adjacency forming between routers R1 and R2 in real time while you configure R2, issue the **debug eigrp packets** command on both routers before configuring router R2.

e. In global configuration mode on R2, issue the same set of commands that you issued on R1 to create EIGRP AS 1 and advertise the 10.0.0.0/8 network. You should see debug output similar to the following.

```
R2# debug eigrp packets
EIGRP Packets debugging is on
    (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB,
SIAQUERY, SIAREPLY)

R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

R2(config)# router eigrp 1
R2(config-router)# network 10.0.0.0
R2(config-router)#
*Feb  3 17:01:03.427: EIGRP: Sending HELLO on FastEthernet0/0
*Feb  3 17:01:03.427:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
*Feb  3 17:01:03.431: EIGRP: Received HELLO on FastEthernet0/0 nbr 10.1.100.1
*Feb  3 17:01:03.431:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0
*Feb  3 17:01:03.431: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.100.1
(FastEthernet0/0) is up: new adjacency
*Feb  3 17:01:03.431: EIGRP: Enqueueing UPDATE on FastEthernet0/0 nbr
10.1.100.1 iidbQ un/rely 0/1 peerQ un/rely 0/0
*Feb  3 17:01:03.435: EIGRP: Received UPDATE on FastEthernet0/0 nbr
10.1.100.1
*Feb  3 17:01:03.435:   AS 1, Flags 0x1, Seq 1/0 idbQ 0/0 iidbQ un/rely 0/1
peerQ un/rely 0/0
*Feb  3 17:01:03.435: EIGRP: Requeued unicast on FastEthernet0/0
*Feb  3 17:01:03.435: EIGRP: Sending HELLO on FastEthernet0/0
*Feb  3 17:01:03.435:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
*Feb  3 17:01:03.439: EIGRP: Sending UPDATE on FastEthernet0/0 nbr 10.1.100.1
*Feb  3 17:01:03.439:   AS 1, Flags 0x1, Seq 1/1 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 0/1
*Feb  3 17:01:03.443: EIGRP: Received UPDATE on FastEthernet0/0 nbr
10.1.100.1
*Feb  3 17:01:03.443:   AS 1, Flags 0x8, Seq 2/0 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 0/1
*Feb  3 17:01:03.447: EIGRP: Received ACK on FastEthernet0/0 nbr 10.1.100.1
*Feb  3 17:01:03.447:   AS 1, Flags 0x0, Seq 0/1 idbQ 0/0 iidbQ un/rely 0/0
un/rely 0/1
*Feb  3 17:01:03.447: EIGRP: Enqueueing UPDATE on FastEthernet0/0 nbr
10.1.100.1 iidbQ un/rely 0/1 peerQ un/rely 0/0 serno 1-2
*Feb  3 17:01:03.451: EIGRP: Requeued unicast on FastEthernet0/0
*Feb  3 17:01:03.455: EIGRP: Sending UPDATE on FastEthernet0/0 nbr 10.1.100.1
*Feb  3 17:01:03.455:   AS 1, Flags 0x8, Seq 2/2 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 0/1 serno 1-2
*Feb  3 17:01:03.455: EIGRP: Enqueueing UPDATE on FastEthernet0/0 iidbQ
un/rely 0/1 serno 3-3
*Feb  3 17:01:03.455: EIGRP: Received UPDATE on FastEthernet0/0 nbr
10.1.100.1
*Feb  3 17:01:03.455:   AS 1, Flags 0x8, Seq 3/1 idbQ 0/0 iidbQ un/rely 0/1
peerQ un/rely 0/1
*Feb  3 17:01:03.455: EIGRP: Enqueueing ACK on FastEthernet0/0 nbr 10.1.100.1
*Feb  3 17:01:03.455:   Ack seq 3 iidbQ un/rely 0/1 peerQ un/rely 1/1
*Feb  3 17:01:03.459: EIGRP: Received ACK on FastEthernet0/0 nbr 10.1.100.1
*Feb  3 17:01:03.459:   AS 1, Flags 0x0, Seq 0/2 idbQ 0/0 iidbQ un/rely 0/1
peerQ un/rely 1/1
*Feb  3 17:01:03.467: EIGRP: Forcing multicast xmit on FastEthernet0/0
*Feb  3 17:01:03.467: EIGRP: Sending UPDATE on FastEthernet0/0
```



```
*Feb  3 17:01:03.467:   AS 1, Flags 0x0, Seq 3/0 idbQ 0/0 iidbQ un/rely 0/0
serno 3-3
*Feb  3 17:01:03.471: EIGRP: Received ACK on FastEthernet0/0 nbr 10.1.100.1
*Feb  3 17:01:03.471:   AS 1, Flags 0x0, Seq 0/3 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 1/1
*Feb  3 17:01:03.471: EIGRP: FastEthernet0/0 multicast flow blocking cleared
*Feb  3 17:01:03.479: EIGRP: Sending ACK on FastEthernet0/0 nbr 10.1.100.1
*Feb  3 17:01:03.479:   AS 1, Flags 0x0, Seq 0/3 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 1/0
```

The debug output displays the EIGRP hello, update, and ACK packets. Because EIGRP uses Reliable Transport Protocol (RTP) for update packets, you see routers replying to update packets with the ACK packet. You can turn off debugging with the **undebug all** command.

f. Configure EIGRP on R3 using the same commands.

```
R3(config)# router eigrp 1
R3(config-router)# network 10.0.0.0

*Feb  3 17:16:05.415: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.100.2
(FastEthernet0/1) is up: new adjacency
*Feb  3 17:16:05.419: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.100.1
(FastEthernet0/1) is up: new adjacency
```

Step 3: Verify the EIGRP configuration.

a. When R3 is configured, issue the **show ip eigrp neighbors** command on each router. If you have configured each router successfully, each router has two adjacencies.

Note: In the output, the “H” column on the left lists the order in which a peering session was established with the specified neighbor. The order uses sequential numbering, starting with 0. The “H” stands for “handle,” which is an internal number used by the EIGRP implementation to refer to a particular neighbor.

```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   10.1.100.3              Fa0/0             10 00:00:17    1   200  0  7
0   10.1.100.2              Fa0/0             11 00:02:01    5   200  0  6

R2# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   10.1.100.3              Fa0/0             13 00:00:56    1   200  0  7
0   10.1.100.1              Fa0/0             12 00:02:40    1   200  0  47

R3# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   10.1.100.2              Fa0/0             11 00:01:21  819  4914  0  6
0   10.1.100.1              Fa0/0             11 00:01:21    2   200  0  47
```

b. Check whether the EIGRP routes are being exchanged between the routers using the **show ip eigrp topology** command.

```
R1# show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.1.1)
```



```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.0/24, 1 successors, FD is 156160
        via 10.1.100.3 (156160/128256), FastEthernet0/0
P 10.1.2.0/24, 1 successors, FD is 156160
        via 10.1.100.2 (156160/128256), FastEthernet0/0
P 10.1.1.0/24, 1 successors, FD is 128256
        via Connected, Loopback1
P 10.1.100.0/24, 1 successors, FD is 28160
        via Connected, FastEthernet0/0
```

You should see all the networks currently advertised by EIGRP on every router. You will explore the output of this command in the next lab. For now, verify that each loopback network exists in the EIGRP topology table.

c. Because EIGRP is the only routing protocol running and currently has routes to these networks, issuing the **show ip route eigrp** command displays the best route to the destination network.

```
R1# show ip route eigrp
     10.0.0.0/24 is subnetted, 4 subnets
D       10.1.3.0 [90/156160] via 10.1.100.3, 00:00:53, FastEthernet0/0
D       10.1.2.0 [90/156160] via 10.1.100.2, 00:00:53, FastEthernet0/0
```

d. To check whether you have full connectivity, ping the remote loopbacks from each router. If you have successfully pinged all the remote loopbacks, congratulations! You have configured EIGRP to route between these three remote networks.

Step 4: Configure EIGRP on the R1 and R2 serial interfaces.

a. Your serial interfaces are still in their default configuration. Specify the interface addresses according to the diagram, and set the clock rate to 64 kb/s for R1.

```
R1(config)# interface serial 0/0/0
R1(config-if)# ip address 10.1.200.1 255.255.255.0
R1(config-if)# clock rate 64000
R1(config-if)# no shut

R2(config)# interface serial 0/0/0
R2(config-if)# ip address 10.1.200.2 255.255.255.0
R2(config-if)# no shut
```

Notice that even though you have clocked the interface at 64 kb/s, issuing the **show interface serial 0/0/0** command reveals that the interface still shows the full T1 bandwidth of 1544 kb/s.

```
R1# show interfaces serial 0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 10.1.200.1/24
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255

<output omitted>
```

The bandwidth is set primarily to provide the correct composite metric factor and a realistic and true description of the available bandwidth on an interface. It is also set to prevent EIGRP from flooding the interface. By default, EIGRP uses up to 50 percent of the bandwidth that the interface reports to the Cisco IOS software. Suppose there was a significant routing instability in some other part of the EIGRP AS. If

EIGRP were to use 50 percent of 1544 kb/s for its own routing information traffic, EIGRP traffic would fully saturate the low-bandwidth 64 kb/s serial link.

Recall that EIGRP uses a composite metric in which one of the variables is the bandwidth of the interface. For EIGRP to make an accurate computation, it needs correct information about the bandwidth of the serial link. Therefore, you must manually configure the bandwidth variable to 64 kb/s.

b. Apply the **bandwidth 64** command to the R1 and R2 serial interfaces.

```
R1(config)# interface serial 0/0/0
R1(config-if)# bandwidth 64

R2(config)# interface serial 0/0/0
R2(config-if)# bandwidth 64
```

c. Verify that your bandwidth configuration is reflected in the output of the **show interface serial 0/0/0** command.

```
R1# show interfaces serial 0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 10.1.200.1/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
<output omitted>

R2# show interfaces serial 0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 10.1.200.2/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
<output omitted>
```

d. Issue the **show ip eigrp neighbors** command, which displays the following neighbor relationship between R1 and R2.

```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
2   10.1.200.2              Se0/0/0           10 00:03:03   24   200  0  53
1   10.1.100.3              Fa0/0             14 09:22:42  269  1614  0  54
0   10.1.100.2              Fa0/0             11 09:22:42  212  1272  0  59
```

Step 5: Configure network statement wildcard masks.

a. On R3, create Loopback11 with IP address 192.168.100.1/30, and Loopback15 with IP address 192.168.100.5/30.

```
R3(config)# interface Loopback11
R3(config-if)# ip address 192.168.100.1 255.255.255.252
R3(config-if)# exit
R3(config)# interface Loopback15
R3(config-if)# ip address 192.168.100.5 255.255.255.252
R3(config-if)# exit
```


How can you add the 192.168.100.0/30 network to EIGRP without involving the 192.168.100.4/30 network as well?

____

____

____

____

In Step 2, you looked at how network statements select networks for routing using major network boundaries. EIGRP also provides a way to select networks using wildcard masks. In a wildcard mask, bits that can vary are denoted by 1s in the binary bit values. If you wanted to route both Loopback11 and Loopback15 with EIGRP, you could use a wildcard mask that includes both of their network addresses, such as **network 192.168.100.0 0.0.0.7** or **network 192.168.100.0 0.0.0.255**. However, in this scenario, you want to select only the IP network for Loopback11.

b. On R3, issue the following commands:

```
R3(config)# router eigrp 1
R3(config-router)# network 192.168.100.0 0.0.0.3
```

c. Did this solution work? Check it with the **show ip eigrp interfaces** command. Notice that Loopback11 is involved in EIGRP, and Loopback15 is not.

```
R3# show ip eigrp interfaces
IP-EIGRP interfaces for process 1

                        Xmit Queue   Mean   Pacing Time   Multicast
Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Fa0/0              2        0/0        5        0/1           50          0
Lo3                0        0/0        0        0/1            0          0
Lo11               0        0/0        0        0/1            0          0
```

d. Which of these two IP networks can you see in the routing table on R1 after EIGRP converges with the new network? Look at the output of the **show ip route eigrp** command on R1.

```
R1# show ip route eigrp
     10.0.0.0/24 is subnetted, 5 subnets
D       10.1.3.0 [90/156160] via 10.1.100.3, 00:05:59, FastEthernet0/0
D       10.1.2.0 [90/156160] via 10.1.100.2, 00:12:16, FastEthernet0/0
D    192.168.100.0/24 [90/156160] via 10.1.100.3, 00:03:05, FastEthernet0/0
```

Notice that the subnet mask for the 192.168.100.0 network advertised by R3 is 24 bits. This will be examined more fully in the next lab. Which configuration command would allow R3 to advertise the proper subnet mask to its adjacent routers?

____

e. On R3, issue the **show ip protocols** command. Notice that automatic summarization is in effect. Also note the networks for which it is routing.

```
R3# show ip protocols
Routing Protocol is "eigrp 1"
  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 1
    EIGRP NSF-aware route hold timer is 240s
    Automatic network summarization is in effect
    Automatic address summarization:
      192.168.100.0/24 for Loopback11
        Summarizing with metric 128256
      10.0.0.0/8 for Loopback3, FastEthernet0/0
        Summarizing with metric 28160
    Maximum path: 4
    Routing for Networks:
      10.0.0.0
      192.168.100.0/30
    Routing Information Sources:
      Gateway         Distance      Last Update
      (this router)         90      00:22:13
      Gateway         Distance      Last Update
      10.1.100.2            90      00:22:15
      10.1.100.1            90      00:22:15
    Distance: internal 90 external 170
```

Challenge: Topology Change

You have been reading up about the advantages of different routing protocols. You noticed statements claiming that EIGRP converges faster than other routing protocols in a topology where there are multiple paths to the destination network. You are interested in testing this before you bring the network that you are designing online.

Verify the neighbor relationships and that the routing table of each router has the original loopback interfaces of the other routers, as described in the initial diagram. Make sure that you issue the **debug ip eigrp** command on all routers.

a. Issue the **show ip route** command on R2 and R3.

```
R2# show ip route eigrp
     10.0.0.0/24 is subnetted, 5 subnets
D       10.1.3.0 [90/156160] via 10.1.100.3, 00:05:22, FastEthernet0/0
D       10.1.1.0 [90/156160] via 10.1.100.1, 00:05:22, FastEthernet0/0
D    192.168.100.0/24 [90/156160] via 10.1.100.3, 00:14:30, FastEthernet0/0

R3# show ip route eigrp
     10.0.0.0/24 is subnetted, 5 subnets
D       10.1.2.0 [90/156160] via 10.1.100.2, 09:25:37, FastEthernet0/0
D       10.1.1.0 [90/156160] via 10.1.100.1, 09:25:37, FastEthernet0/0
D       10.0.0.0/8 is a summary, 09:25:37, Null0
D       10.1.200.0 [90/40514560] via 10.1.100.2, 00:03:01, FastEthernet0/0
                   [90/40514560] via 10.1.100.1, 00:03:01, FastEthernet0/0
     192.168.100.0/24 is variably subnetted, 3 subnets, 2 masks
D       192.168.100.0/24 is a summary, 00:18:15, Null0
```

b. From R3, trace the route to the Lo1 IP address on R1.

```
R3# traceroute 10.1.1.1

Type escape sequence to abort.
```

```
Tracing the route to 10.1.1.1

  1 10.1.100.1 4 msec *  0 msec
```

R3 is using R1 as the next hop to get to destination network 10.1.1.0/24 per the R3 routing table. However, R3 could potentially get to R1 through R2 via the serial link if the Fa0/0 interface on R1 was shut down.

c. From R3, issue a ping with a high repeat count to the destination address 10.1.1.1. You should see multiple exclamation points flooding the console output from R3.

```
R3# ping 10.1.1.1 repeat 10000
```

d. While the extended ping on R3 is running, shut down the Fa0/0 interface on R1. Allow the pings on R3 to complete.

```
R1(config)# interface FastEthernet0/0
R1(config-if)# shutdown


Type escape sequence to abort.
Sending 10000, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!.......!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

*Feb  4 13:35:55.311: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.100.1
(FastEthernet0/0) is down: holding time expired

<output omitted>

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Success rate is 99 percent (9992/10000), round-trip min/avg/max = 1/16/68 ms
```

From the perspective of R3, how many packets were dropped?

____

____

____

Which of the EIGRP timers causes this delay in the route recalculation?

____

e. Use the **traceroute** command to find the new route from R3 to R1.

```
R3# traceroute 10.1.1.1

Type escape sequence to abort.
Tracing the route to 10.1.1.1

  1 10.1.100.2 0 msec 0 msec 0 msec
  2 10.1.200.1 16 msec 12 msec *
```

f. Start the repeated ping again from R3, and administratively bring up the Fa0/0 interface on R1.

```
R3# ping 10.1.1.1 repeat 10000

R1(config)# interface FastEthernet0/0
R1(config-if)# no shutdown

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!............!!

*Feb  4 13:35:55.147: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.100.1
(FastEthernet0/0) is up: new adjacency!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

<output omitted>

Success rate is 99 percent (9983/10000), round-trip min/avg/max = 1/2/44 ms
```

From the perspective of R3, how many packets were dropped?

____

____

Note: The loss ICMP ECHO packets results in a significant delay, as many as 30 or more seconds. Why did it take so long for R3 to reestablish ping connectivity with R3 after the R1 Fa0/0 interface was re-enabled and what changes could be made to correct the problem? The answer lies with the switch itself.

The switch that connects the three routers together is in its default configuration, running STP on each port and requiring 30 seconds to proceed through Listening and Learning states until a port transitions to the Forwarding state. The 17 lost packets are caused by the 30 seconds required by STP to transition the port to Forwarding state plus a couple of seconds for DTP to determine the port mode and perhaps ARP to resolve R3's MAC address.

This issue can be addressed by configuring the switch with the **spanning-tree portfast default** command. In addition, all ports could be defined as static access ports using the **switchport mode access** command.

If you were using RIPv2 as your routing protocol instead of EIGRP, would fewer or more packets be dropped?

____

____

____

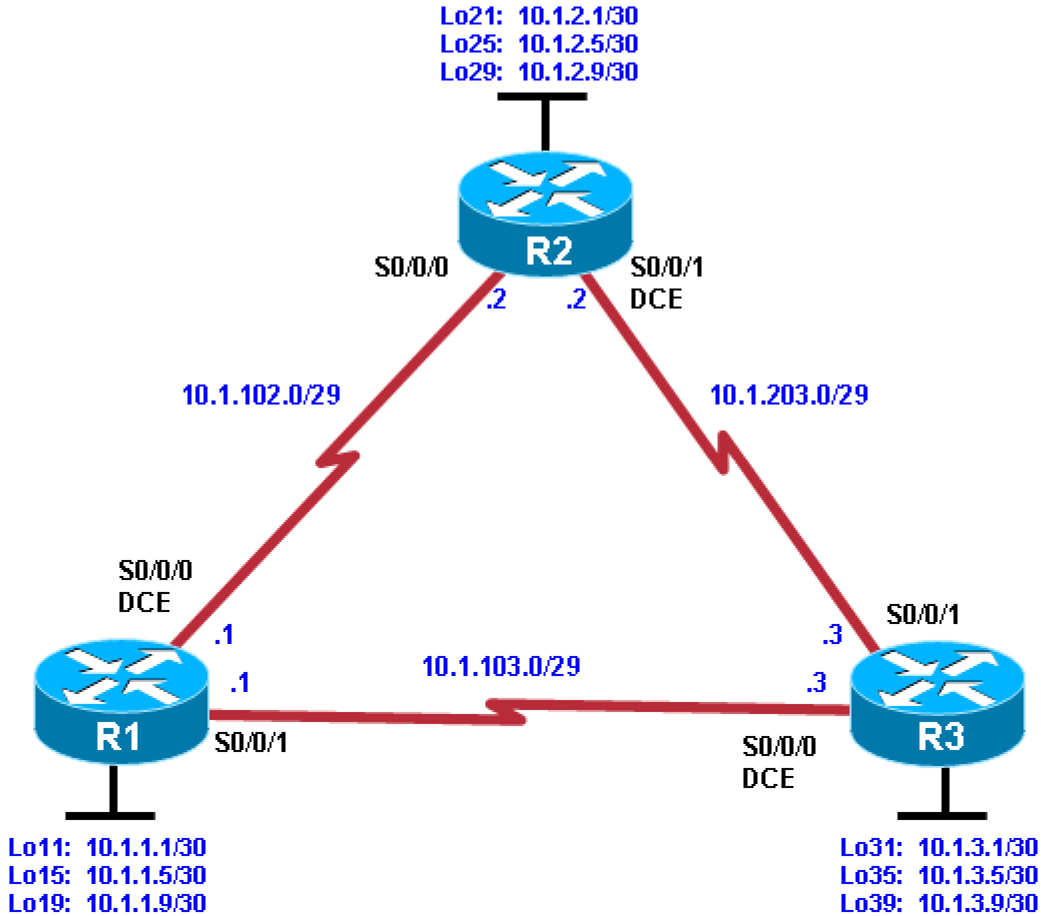
____

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 2 Lab 2-2, EIGRP Load Balancing

Topology



Objectives

- Review a basic EIGRP configuration.
- Explore the EIGRP topology table.
- Identify successors, feasible successors, and feasible distances.
- Use **show** and **debug** commands for the EIGRP topology table.
- Configure and verify equal-cost load balancing with EIGRP.
- Configure and verify unequal-cost load balancing with EIGRP.

Background

As a senior network engineer, you are considering deploying EIGRP in your corporation and want to evaluate its ability to converge quickly in a changing environment. You are also interested in equal-cost and unequal-cost load balancing because your network contains redundant links. These links are not often used by other link-state routing protocols because of high metrics. Because you are interested in testing the EIGRP claims that you have read about, you decide to implement and test on a set of three lab routers before deploying EIGRP throughout your corporate network.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the advanced IP services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Configure the addressing and serial links.

a. Create three loopback interfaces on each router and address them as 10.1.*X*.1/30, 10.1.*X*.5/30, and 10.1.*X*.9/30, where *X* is the number of the router. Use the following table or the initial configurations located at the end of the lab.

Router	Interface	IP Address/Mask
R1	Loopback11	10.1.1.1/30
R1	Loopback15	10.1.1.5/30
R1	Loopback19	10.1.1.9/30
R2	Loopback21	10.1.2.1/30
R2	Loopback25	10.1.2.5/30
R2	Loopback29	10.1.2.9/30
R3	Loopback31	10.1.3.1/30
R3	Loopback35	10.1.3.5/30
R3	Loopback39	10.1.3.9/30

```
R1(config)# interface Loopback 11
R1(config-if)# ip address 10.1.1.1 255.255.255.252
R1(config-if)# exit
R1(config)# interface Loopback 15
R1(config-if)# ip address 10.1.1.5 255.255.255.252
R1(config-if)# exit
R1(config)# interface Loopback 19
R1(config-if)# ip address 10.1.1.9 255.255.255.252
R1(config-if)# exit

R2(config)# interface Loopback 21
R2(config-if)# ip address 10.1.2.1 255.255.255.252
R2(config-if)# exit
R2(config)# interface Loopback 25
R2(config-if)# ip address 10.1.2.5 255.255.255.252
R2(config-if)# exit
R2(config)# interface Loopback 29
R2(config-if)# ip address 10.1.2.9 255.255.255.252
R2(config-if)# exit
```



```
R3(config)# interface Loopback 31
R3(config-if)# ip address 10.1.3.1 255.255.255.252
R3(config-if)# exit
R3(config)# interface Loopback 35
R3(config-if)# ip address 10.1.3.5 255.255.255.252
R3(config-if)# exit
R3(config)# interface Loopback 39
R3(config-if)# ip address 10.1.3.9 255.255.255.252
R3(config-if)# exit
```

b. Specify the addresses of the serial interfaces as shown in the topology diagram. Set the clock rate to 64 kb/s, and manually configure the interface bandwidth to 64 kb/s.

Note: If you have WIC-2A/S serial interfaces, the maximum clock rate is 128 kb/s. If you have WIC-2T serial interfaces, the maximum clock rate is much higher (2.048 Mb/s or higher depending on the hardware), which is more representative of a modern network WAN link. However, this lab uses 64 kb/s and 128 kb/s settings.

```
R1(config)# interface Serial 0/0/0
R1(config-if)# description R1-->R2
R1(config-if)# clock rate 64000
R1(config-if)# bandwidth 64
R1(config-if)# ip address 10.1.102.1 255.255.255.248
R1(config-if)# no shutdown
R1(config-if)# exit
R1(config)# interface Serial 0/0/1
R1(config-if)# description R1-->R3
R1(config-if)# bandwidth 64
R1(config-if)# ip address 10.1.103.1 255.255.255.248
R1(config-if)# no shutdown
R1(config-if)# exit

R2(config)# interface Serial 0/0/0
R2(config-if)# description R2-->R1
R2(config-if)# bandwidth 64
R2(config-if)# ip address 10.1.102.2 255.255.255.248
R2(config-if)# no shutdown
R2(config-if)# exit
R2(config)# interface Serial 0/0/1
R2(config-if)# description R2-->R3
R2(config-if)# clock rate 64000
R2(config-if)# bandwidth 64
R2(config-if)# ip address 10.1.203.2 255.255.255.248
R2(config-if)# no shutdown
R2(config-if)# exit

R3(config)# interface Serial 0/0/0
R3(config-if)# description R3-->R1
R3(config-if)# clock rate 64000
R3(config-if)# bandwidth 64
R3(config-if)# ip address 10.1.103.3 255.255.255.248
R3(config-if)# no shutdown
R3(config-if)# exit
R3(config)# interface Serial 0/0/1
R3(config-if)# description R3-->R2
R3(config-if)# bandwidth 64
R3(config-if)# ip address 10.1.203.3 255.255.255.248
R3(config-if)# no shutdown
```



```
R3(config-if)# exit
```

c. Verify connectivity by pinging across each of the local networks connected to each router.

d. Issue the **show interfaces description** command on each router. This command displays a brief listing of the interfaces, their status, and a description (if a description is configured). Router R1 is shown as an example.

```
R1# show interfaces description
Interface                      Status         Protocol Description
Fa0/0                          admin down     down
Fa0/1                          admin down     down
Se0/0/0                        up             up       R1-->R2
Se0/0/1                        up             up       R1-->R3
Vl1                            up             down
Lo11                           up             up
Lo15                           up             up
Lo19                           up             up
```

e. Issue the **show protocols** command on each router. This command displays a brief listing of the interfaces, their status, and the IP address and subnet mask configured (in prefix format /xx) for each interface. Router R1 is shown as an example.

```
R1# show protocols
Global values:
  Internet Protocol routing is enabled
FastEthernet0/0 is administratively down, line protocol is down
FastEthernet0/1 is administratively down, line protocol is down
Serial0/0/0 is up, line protocol is up
  Internet address is 10.1.102.1/29
Serial0/0/1 is up, line protocol is up
  Internet address is 10.1.103.1/29
Vlan1 is up, line protocol is down
Loopback11 is up, line protocol is up
  Internet address is 10.1.1.1/30
Loopback15 is up, line protocol is up
  Internet address is 10.1.1.5/30
Loopback19 is up, line protocol is up
  Internet address is 10.1.1.9/30
```

Step 2: Configure EIGRP.

a. Enable EIGRP AS 100 for all interfaces on R1 and R2 using the commands used in the previous EIGRP lab. Do not enable EIGRP yet on R3. For your reference, these are the commands which can be used:

```
R1(config)# router eigrp 100
R1(config-router)# network 10.0.0.0

R2(config)# router eigrp 100
R2(config-router)# network 10.0.0.0
```

b. Use the **debug ip eigrp 100** command to watch EIGRP install the routes in the routing table when your routers become adjacent. You get output similar to the following.

```
R3# debug ip eigrp 100
IP-EIGRP Route Events debugging is on
R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.

R3(config)# router eigrp 100
```

```
R3(config-router)# network 10.0.0.0

R3(config-router)#
*Feb  4 18:44:57.367: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.1.103.1
(Serial0/0/0) is up: new adjacency
*Feb  4 18:44:57.367: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.1.203.2
(Serial0/0/1) is up: new adjacency
*Feb  4 18:44:57.371: IP-EIGRP(Default-IP-Routing-Table:100): Processing
incoming UPDATE packet
*Feb  4 18:44:57.379: IP-EIGRP(Default-IP-Routing-Table:100): Processing
incoming UPDATE packet
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): Processing
incoming UPDATE packet
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): Int
10.1.102.0/29 M 41024000 - 40000000 1024000 SM 40512000 - 40000000 512000
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.102.0  ()
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): Int 10.1.1.0/30
M40640000 - 40000000 640000 SM 128256 - 256 128000
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.0  ()
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): Int 10.1.1.4/30
M 40640000 - 40000000 640000 SM 128256 - 256 128000
*Feb  4 18:44:57.427: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.4  ()
*Feb  4 18:44:57.431: IP-EIGRP(Default-IP-Routing-Table:100): Int 10.1.1.8/30
M40640000 - 40000000 640000 SM 128256 - 256 128000
*Feb  4 18:44:57.431: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.8  ()

<output omitted>
```

Essentially, the EIGRP DUAL state machine has just computed the topology table for these routes and installed them in the routing table.

c. Check to see that these routes exist in the routing table with the **show ip route** command.

```
R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

   10.0.0.0/8 is variably subnetted, 12 subnets, 2 masks
D     10.1.3.8/30 [90/40640000] via 10.1.103.3, 00:19:28, Serial0/0/1
D     10.1.2.8/30 [90/40640000] via 10.1.102.2, 00:21:59, Serial0/0/0
C     10.1.1.8/30 is directly connected, Loopback19
D     10.1.3.0/30 [90/40640000] via 10.1.103.3, 00:19:28, Serial0/0/1
D     10.1.2.0/30 [90/40640000] via 10.1.102.2, 00:21:59, Serial0/0/0
C     10.1.1.0/30 is directly connected, Loopback11
D     10.1.3.4/30 [90/40640000] via 10.1.103.3, 00:19:28, Serial0/0/1
D     10.1.2.4/30 [90/40640000] via 10.1.102.2, 00:21:59, Serial0/0/0
C     10.1.1.4/30 is directly connected, Loopback15
```



```
C       10.1.103.0/29 is directly connected, Serial0/0/1
C       10.1.102.0/29 is directly connected, Serial0/0/0
D       10.1.203.0/29 [90/41024000] via 10.1.103.3, 00:19:28, Serial0/0/1
                      [90/41024000] via 10.1.102.2, 00:19:28, Serial0/0/0
```

d. After you have full adjacency between the routers, ping all the remote loopbacks to ensure full connectivity or use the following Tcl script. If you have never used Tcl scripts or need a refresher, see Lab 1–1.

```
R1# tclsh

foreach address {
10.1.1.1
10.1.1.5
10.1.1.9
10.1.2.1
10.1.2.5
10.1.2.9
10.1.3.1
10.1.3.5
10.1.3.9
10.1.102.1
10.1.102.2
10.1.103.1
10.1.103.3
10.1.203.2
10.1.203.3
} { ping $address }
```

You should receive ICMP echo replies for each address pinged. Make sure that you run the Tcl script on each router and verify connectivity before you continue with the lab.

e. Verify the EIGRP neighbor relationships with the **show ip eigrp neighbors** command.

```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.102.2              Se0/0/0           10 00:00:22    1  5000  2  0
1   10.1.103.3              Se0/0/1           13 00:04:36   24  2280  0  14

R2# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.102.1              Se0/0/0           14 00:00:37    1  5000  1  22
1   10.1.203.3              Se0/0/1           11 00:03:29  143  2280  0  15

R3# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   10.1.203.2              Se0/0/1           14 00:03:43  241  2280  0  18
0   10.1.103.1              Se0/0/0           14 00:05:05   38  2280  0  17
```


Step 3: Examine the EIGRP topology table.

a. EIGRP builds a topology table containing all successor routes. The course content covered the vocabulary for EIGRP routes in the topology table. What is the feasible distance of route 10.1.1.0/30 in the R3 topology table in the following output?

____

____

```
R3# show ip eigrp topology
IP-EIGRP Topology Table for AS(100)/ID(10.1.3.9)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.8/30, 1 successors, FD is 128256
    via Connected, Loopback39
P 10.1.2.8/30, 1 successors, FD is 40640000
    via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.8/30, 1 successors, FD is 40640000
    via 10.1.103.1 (40640000/128256), Serial0/0/0
P 10.1.3.0/30, 1 successors, FD is 128256
    via Connected, Loopback31
P 10.1.2.0/30, 1 successors, FD is 40640000
    via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.0/30, 1 successors, FD is 40640000
    via 10.1.103.1 (40640000/128256), Serial0/0/0
P 10.1.3.4/30, 1 successors, FD is 128256
    via Connected, Loopback35
P 10.1.2.4/30, 1 successors, FD is 40640000
    via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.4/30, 1 successors, FD is 40640000
    via 10.1.103.1 (40640000/128256), Serial0/0/0
P 10.1.103.0/29, 1 successors, FD is 40512000
    via Connected, Serial0/0/0
P 10.1.102.0/29, 2 successors, FD is 41024000
    via 10.1.103.1 (41024000/40512000), Serial0/0/0
    via 10.1.203.2 (41024000/40512000), Serial0/0/1
P 10.1.203.0/29, 1 successors, FD is 40512000
    via Connected, Serial0/0/1
```

b. The most important thing is the two successor routes in the passive state on R3. R1 and R2 are both advertising their connected subnet of 10.1.102.0/30. Because both routes have the same feasible distance of 41024000, both are installed in the topology table. This distance of 41024000 reflects the composite metric of more granular properties about the path to the destination network. Can you view the metrics before the composite metric is computed?

____

____

____

____

c. Use the **show ip eigrp topology 10.1.102.0/29** command to view the information that EIGRP has received about the route from R1 and R2.

```
R3# show ip eigrp topology 10.1.102.0/29
IP-EIGRP (AS 100): Topology entry for 10.1.102.0/29
  State is Passive, Query origin flag is 1, 2 Successor(s), FD is 41024000
  Routing Descriptor Blocks:
  10.1.103.1 (Serial0/0/0), from 10.1.103.1, Send flag is 0x0
      Composite metric is (41024000/40512000), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 40000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
  10.1.203.2 (Serial0/0/1), from 10.1.203.2, Send flag is 0x0
      Composite metric is (41024000/40512000), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 40000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

The output of this command shows the following information regarding EIGRP:

- The bandwidth metric represents the *minimum* bandwidth among all links comprising the path to the destination network.
- The delay metric represents the *total* delay over the path.
- The minimum MTU represents the smallest MTU along the path.
- If you do not have full knowledge of your network, you can use the hop count information to check how many Layer 3 devices are between the router and the destination network.

Step 4: Observe equal-cost load balancing.

EIGRP produces equal-cost load balancing to the destination network 10.1.102.0/29 from R1. Two equal-cost paths are available to this destination per the topology table output above.

a. Use the **traceroute 10.1.102.1** command to view the hops from R3 to this R1 IP address. Notice that both R1 and R2 are listed as hops because there are two equal-cost paths and packets can reach this network via either link.

```
R3# traceroute 10.1.102.1

Type escape sequence to abort.
Tracing the route to 10.1.102.1

  1 10.1.203.2 12 msec
    10.1.103.1 12 msec
    10.1.203.2 12 msec
```

Recent Cisco IOS releases enable Cisco Express Forwarding (CEF), which, by default, performs per-destination load balancing. CEF allows for very rapid switching without the need for route processing. However, if you were to ping the destination network, you would not see load balancing occurring on a packet level because CEF treats the entire series of pings as one flow.

CEF on R3 overrides the per-packet balancing behavior of process switching with per-destination load balancing.

b. To see the full effect of EIGRP equal-cost load balancing, temporarily disable CEF and route caching so that all IP packets are processed individually and not fast-switched by CEF.

```
R3(config)# no ip cef

R3(config)# interface S0/0/0
R3(config-if)# no ip route-cache
R3(config-if)# interface S0/0/1
R3(config-if)# no ip route-cache
```

Note: Typically, you would not disable CEF in a production network. It is done here only to illustrate load balancing. Another way to demonstrate per-packet load balancing, that does not disable CEF, is to use the per-packet load balancing command **ip load-share per-packet** on outgoing interfaces S0/0/0 and S0/0/1.

c. Verify load balancing with the **debug ip packet** command, and then ping 10.1.102.1. You see output similar to the following:

```
R3# debug ip packet
IP packet debugging is on

R3# ping 10.1.102.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.102.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
R3#
*Feb  5 12:58:27.943: IP: tableid=0, s=10.1.103.3 (local), d=10.1.102.1
(Serial0/0/0), routed via RIB
*Feb  5 12:58:27.943: IP: s=10.1.103.3 (local), d=10.1.102.1 (Serial0/0/0),
len 100, sending
*Feb  5 12:58:27.947: IP: tableid=0, s=10.1.102.1 (Serial0/0/0), d=10.1.103.3
(Serial0/0/0), routed via RIB
*Feb  5 12:58:27.947: IP: s=10.1.102.1 (Serial0/0/0), d=10.1.103.3
(Serial0/0/0), len 100, rcvd 3
*Feb  5 12:58:27.947: IP: tableid=0, s=10.1.203.3 (local), d=10.1.102.1
(Serial0/0/1), routed via RIB
*Feb  5 12:58:27.947: IP: s=10.1.203.3 (local), d=10.1.102.1 (Serial0/0/1),
len 100, sending

<output omitted>
```

Notice that EIGRP load-balances between Serial0/0/0 (s=10.1.103.3) and Serial0/0/1 (s=10.1.203.3). This behavior is part of EIGRP. It can help utilize underused links in a network, especially during periods of congestion.

Step 5: Analyze alternate EIGRP paths not in the topology table.

a. Perhaps you expected to see more paths to the R1 and R2 loopback networks in the R3 topology table. Why are these routes not shown in the topology table?

b. Issue the **show ip eigrp topology all-links** command to see all routes that R3 has learned through EIGRP. This command shows all entries that EIGRP holds on this router for networks in the topology, including the exit serial interface and IP address of the next hop to each destination network, and the serial number (serno) that uniquely identifies a destination network in EIGRP.

```
R3# show ip eigrp topology all-links
IP-EIGRP Topology Table for AS(100)/ID(10.1.3.9)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.0/30, 1 successors, FD is 128256, serno 1
        via Connected, Loopback31
P 10.1.3.4/30, 1 successors, FD is 128256, serno 2
        via Connected, Loopback35
P 10.1.3.8/30, 1 successors, FD is 128256, serno 3
        via Connected, Loopback39
P 10.1.2.8/30, 1 successors, FD is 40640000, serno 24
        via 10.1.203.2 (40640000/128256), Serial0/0/1
        via 10.1.103.1 (41152000/40640000), Serial0/0/0
P 10.1.1.8/30, 1 successors, FD is 40640000, serno 17
        via 10.1.103.1 (40640000/128256), Serial0/0/0
        via 10.1.203.2 (41152000/40640000), Serial0/0/1
P 10.1.2.0/30, 1 successors, FD is 40640000, serno 22
        via 10.1.203.2 (40640000/128256), Serial0/0/1
        via 10.1.103.1 (41152000/40640000), Serial0/0/0
P 10.1.1.0/30, 1 successors, FD is 40640000, serno 15
        via 10.1.103.1 (40640000/128256), Serial0/0/0
        via 10.1.203.2 (41152000/40640000), Serial0/0/1
P 10.1.2.4/30, 1 successors, FD is 40640000, serno 23
        via 10.1.203.2 (40640000/128256), Serial0/0/1
        via 10.1.103.1 (41152000/40640000), Serial0/0/0
P 10.1.1.4/30, 1 successors, FD is 40640000, serno 16
        via 10.1.103.1 (40640000/128256), Serial0/0/0
        via 10.1.203.2 (41152000/40640000), Serial0/0/1
P 10.1.103.0/29, 1 successors, FD is 40512000, serno 13
        via Connected, Serial0/0/0
P 10.1.102.0/29, 2 successors, FD is 41024000, serno 42
        via 10.1.103.1 (41024000/40512000), Serial0/0/0
```

```
        via 10.1.203.2 (41024000/40512000), Serial0/0/1
P 10.1.203.0/29, 1 successors, FD is 40512000, serno 12
        via Connected, Serial0/0/1
```

What is the advertised distance of the R1 loopback network routes from R1 and R2?

____

____

____

____

____

c. Use the **show ip eigrp topology 10.1.2.0/30** command to see the granular view of the alternate paths to 10.1.2.0, including ones with a higher reported distance than the feasible distance.

```
R3# show ip eigrp topology 10.1.2.0/30
IP-EIGRP (AS 100): Topology entry for 10.1.2.0/30
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 40640000
  Routing Descriptor Blocks:
  10.1.203.2 (Serial0/0/1), from 10.1.203.2, Send flag is 0x0
      Composite metric is (40640000/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
  10.1.103.1 (Serial0/0/0), from 10.1.103.1, Send flag is 0x0
      Composite metric is (41152000/40640000), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 45000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
```

When using the **show ip eigrp topology** command, why is the route to 10.1.2.1 through R1 not listed in the topology table?

____

____

____

____

____

What is its advertised distance from R1?

____

What is its feasible distance?

____

If the R2 Serial0/0/1 interface were shut down, would EIGRP route through R1 to get to 10.1.2.0/30? Would the switch be immediate?

____

____

____

____

____

____

____

____

Record your answer, and then experiment by shutting down the R1 s0/0/01 interface while an extended ping is running as described below.

d. Start a ping with a high repeat count on R3 to the R1 Serial0/0/0 interface 10.1.102.1.

```
R3# ping 10.1.102.1 repeat 10000
```

e. Enter interface configuration mode on R1 and shut down port Serial0/0/1, which is the direct link from R1 to R3.

```
R1(config)# interface serial 0/0/1
R1(config-if)# shutdown
```

f. When the adjacency between R1 and R3 goes down, some pings will be lost. After pings are again being successfully received, stop the ping using Ctrl+Shift+^.

```
R3# ping 10.1.102.1 repeat 10000

Type escape sequence to abort.
Sending 10000, 100-byte ICMP Echos to 10.1.102.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!.
*Dec 11 18:41:55.843: %LINK-3-UPDOWN: Interface Serial0/0/0, changed state to
down
*Dec 11 18:41:55.847: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.1.103.1
(Serial0/0/0) is down: interface down
*Dec 11 18:41:56.843: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Serial0/0/0, changed state to down
.!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 99 percent (374/376), round-trip min/avg/max = 28/39/96 ms
R3#
```

How many packets were dropped?

____

Note: When examining the EIGRP reconvergence speed after deactivating the serial link between R1 and R3, the focus should not be on the count of lost ping packets but rather on the duration of connectivity loss or how long it took to perform a successful cutover. The router waits for up to two seconds for each sent ICMP ECHO request to receive a reply and only then does it send another ECHO request. If the router did not wait for the reply, the count of lost packets would be much higher. Because two packets were lost, the cutover took approximately 4 seconds.

Another factor to consider is that an interface deliberately delays the information about loss of connectivity for 2 seconds to prevent transient link flaps (link going up and down) from introducing instability into the network. If the real speed of EIGRP is to be observed, this delay can be made as short as possible using the command **carrier-delay msec 0** on all serial interfaces.

g. Issue the **no shutdown** command on the R1 Serial0/0/1 interface before continuing to the next step.

Step 6: Observe unequal-cost load balancing.

a. Review the composite metrics advertised by EIGRP using the **show ip eigrp topology 10.1.2.0/30** command,.

```
R3# show ip eigrp topology 10.1.2.0/30
IP-EIGRP (AS 100): Topology entry for 10.1.2.0/30
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 40640000
  Routing Descriptor Blocks:
  10.1.203.2 (Serial0/0/1), from 10.1.203.2, Send flag is 0x0
      Composite metric is (40640000/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
  10.1.103.1 (Serial0/0/0), from 10.1.103.1, Send flag is 0x0
      Composite metric is (41152000/40640000), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 45000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
```

The reported distance for a loopback network is higher than the feasible distance, so DUAL does not consider it a feasible successor route.

b. To demonstrate unequal-cost load balancing in your internetwork, upgrade the path to the destination network through R1 with a higher bandwidth. Change the clock rate and bandwidth on the R1, R2, and R3 serial interfaces to 128 kb/s.

```
R1(config)# interface serial 0/0/0
R1(config-if)# bandwidth 128
R1(config-if)# clock rate 128000
R1(config-if)# interface serial 0/0/1
R1(config-if)# bandwidth 128

R2(config)# interface serial 0/0/0
R2(config-if)# bandwidth 128
```



```
R3(config)# interface serial 0/0/0
R3(config-if)# clock rate 128000
R3(config-if)# bandwidth 128
```

c. Issue the **show ip eigrp topology 10.1.2.0/30** command again on R3 to see what has changed.

```
R3# show ip eigrp topology 10.1.2.0/30
IP-EIGRP (AS 100): Topology entry for 10.1.2.0/30
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 21152000
  Routing Descriptor Blocks:
  10.1.103.1 (Serial0/0/0), from 10.1.103.1, Send flag is 0x0
      Composite metric is (21152000/20640000), Route is Internal
      Vector metric:
        Minimum bandwidth is 128 Kbit
        Total delay is 45000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2

 10.1.203.2 (Serial0/0/1), from 10.1.203.2, Send flag is 0x0
      Composite metric is (40640000/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

After manipulating the bandwidth parameter, the preferred path for R3 to the loopback interfaces of R2 is now through R1. Even though the hop count is two and the delay through R1 is nearly twice that of the R2 path, the higher bandwidth and lower FD results in this being the preferred route.

d. Issue the **show ip route** command to verify that the preferred route to network 10.1.2.0 is through R1 via Serial0/0/0 to next hop 10.1.103.1. There is only one route to this network due to the difference in bandwidth.

```
R3# show ip route eigrp
     10.0.0.0/8 is variably subnetted, 12 subnets, 2 masks
D       10.1.2.8/30 [90/21152000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.1.8/30 [90/20640000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.2.0/30 [90/21152000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.1.0/30 [90/20640000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.2.4/30 [90/21152000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.1.4/30 [90/20640000] via 10.1.103.1, 00:16:52, Serial0/0/0
D       10.1.102.0/29 [90/21024000] via 10.1.103.1, 00:16:52, Serial0/0/0
```

e. Issue the **debug ip eigrp 100** command on R3 to show route events changing in real time. Then, under the EIGRP router configuration on R3, issue the **variance 2** command, which allows unequal-cost load balancing bounded by a maximum distance of (2) × (FD), where FD represents the feasible distance for each route in the routing table.

```
R3# debug ip eigrp 100
IP-EIGRP Route Events debugging is on

R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
```



```
R3(config)# router eigrp 100
R3(config-router)# variance 2
R3(config-router)#
*Feb  5 15:11:45.195: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.3.8/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.195: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.8  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.8  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.8  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.1.8/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.3.0/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.0  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.0  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.0  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.1.0/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.3.4/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.4  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.2.4  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.1.4  ()
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.1.4/30
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): 10.1.103.0/29
routing table not updated thru 10.1.203.2
*Feb  5 15:11:45.199: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.102.0  ()
*Feb  5 15:11:45.203: IP-EIGRP(Default-IP-Routing-Table:100): route installed
for 10.1.102.0  ()
```

f. Issue the **show ip route** command again to verify that there are now two routes to network 10.1.2.0.

```
R3# show ip route eigrp

     10.0.0.0/8 is variably subnetted, 12 subnets, 2 masks
D       10.1.2.8/30 [90/40640000] via 10.1.203.2, 00:02:27, Serial0/0/1
                    [90/21152000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.1.8/30 [90/20640000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.2.0/30 [90/40640000] via 10.1.203.2, 00:02:27, Serial0/0/1
                    [90/21152000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.1.0/30 [90/20640000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.2.4/30 [90/40640000] via 10.1.203.2, 00:02:27, Serial0/0/1
                    [90/21152000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.1.4/30 [90/20640000] via 10.1.103.1, 00:02:27, Serial0/0/0
D       10.1.102.0/29 [90/41024000] via 10.1.203.2, 00:02:27, Serial0/0/1
                      [90/21024000] via 10.1.103.1, 00:02:27, Serial0/0/0
```


g. These unequal-cost routes also show up in the EIGRP topology table, even though they are not considered feasible successor routes. Use the **show ip eigrp topology** command to verify this.

```
R3# show ip eigrp topology
IP-EIGRP Topology Table for AS(100)/ID(10.1.3.9)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.8/30, 1 successors, FD is 128256
        via Connected, Loopback39
P 10.1.2.8/30, 1 successors, FD is 21152000
        via 10.1.103.1 (21152000/20640000), Serial0/0/0
        via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.8/30, 1 successors, FD is 20640000
        via 10.1.103.1 (20640000/128256), Serial0/0/0
P 10.1.3.0/30, 1 successors, FD is 128256
        via Connected, Loopback31
P 10.1.2.0/30, 1 successors, FD is 21152000
        via 10.1.103.1 (21152000/20640000), Serial0/0/0
        via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.0/30, 1 successors, FD is 20640000
        via 10.1.103.1 (20640000/128256), Serial0/0/0
P 10.1.3.4/30, 1 successors, FD is 128256
        via Connected, Loopback35
P 10.1.2.4/30, 1 successors, FD is 21152000
        via 10.1.103.1 (21152000/20640000), Serial0/0/0
        via 10.1.203.2 (40640000/128256), Serial0/0/1
P 10.1.1.4/30, 1 successors, FD is 20640000
        via 10.1.103.1 (20640000/128256), Serial0/0/0
P 10.1.103.0/29, 1 successors, FD is 20512000
        via Connected, Serial0/0/0
P 10.1.102.0/29, 1 successors, FD is 21024000
        via 10.1.103.1 (21024000/20512000), Serial0/0/0
        via 10.1.203.2 (41024000/20512000), Serial0/0/1
P 10.1.203.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/1
```

h. Load balancing over serial links occurs in blocks of packets, the number of which are recorded in the routing table's detailed routing information. Use the **show ip route 10.1.2.0** command to get a detailed view of how traffic is shared between the two links.

```
R3# show ip route 10.1.2.0
Routing entry for 10.1.2.0/30
  Known via "eigrp 100", distance 90, metric 21152000, type internal
  Redistributing via eigrp 100
  Last update from 10.1.203.2 on Serial0/0/1, 00:05:41 ago
  Routing Descriptor Blocks:
    10.1.203.2, from 10.1.203.2, 00:05:41 ago, via Serial0/0/1
      Route metric is 40640000, traffic share count is 25
      Total delay is 25000 microseconds, minimum bandwidth is 64 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
  * 10.1.103.1, from 10.1.103.1, 00:05:41 ago, via Serial0/0/0
      Route metric is 21152000, traffic share count is 48
      Total delay is 45000 microseconds, minimum bandwidth is 128 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
```

i. Check the actual load balancing using the **debug ip packet** command. Ping from R3 to 10.1.2.1 with a high enough repeat count to view the load balancing over both paths. In the case above, the traffic share is 25 packets routed to R2 to every 48 packets routed to R1.

j. To filter the debug output to make it more useful, use the following extended access list.

```
R3(config)# access-list 100 permit icmp any any echo
R3(config)# end

R3# debug ip packet 100
IP packet debugging is on for access list 100

R3# ping 10.1.2.1 repeat 50

Type escape sequence to abort.
Sending 50, 100-byte ICMP Echos to 10.1.2.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*Feb  5 15:20:54.215: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.215: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.231: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.231: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.247: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.247: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.263: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.263: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.279: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.279: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.295: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*Feb  5 15:20:54.295: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
*Feb  5 15:20:54.311: IP: tableid=0, s=10.1.103.3 (local), d=10.1.2.1
(Serial0/0/0), routed via RIB
*Feb  5 15:20:54.311: IP: s=10.1.103.3 (local), d=10.1.2.1 (Serial0/0/0), len
100, sending
!
<output omitted until the switch to the other path takes place>
!
*Feb  5 15:20:55.395: IP: tableid=0, s=10.1.203.3 (local), d=10.1.2.1
(Serial0/0/1), routed via RIB
!
```

R3 just switched to load-share the outbound ICMP packets to Serial0/0/1.

```
!
*Feb  5 15:20:55.395: IP: s=10.1.203.3 (local), d=10.1.2.1 (Serial0/0/1), len
100, sending
```



```
*Feb  5 15:20:55.423: IP: tableid=0, s=10.1.203.3 (local), d=10.1.2.1
(Serial0/0/1), routed via RIB
*Feb  5 15:20:55.423: IP: s=10.1.203.3 (local), d=10.1.2.1 (Serial0/0/1), len
100, sending
*Feb  5 15:20:55.451: IP: tableid=0, s=10.1.203.3 (local), d=10.1.2.1
(Serial0/0/1), routed via RIB
*Feb  5 15:20:55.451: IP: s=10.1.203.3 (local), d=10.1.2.1 (Serial0/0/1), len
100, sending
*Feb  5 15:20:55.483: IP: tableid=0, s=10.1.203.3 (local), d=10.1.2.1
(Serial0/0/1), routed via RIB
*Feb  5 15:20:55.483: IP: s=10.1.203.3 (local), d=10.1.2.1 (Serial0/0/1), len
100, sending

<output omitted>
```

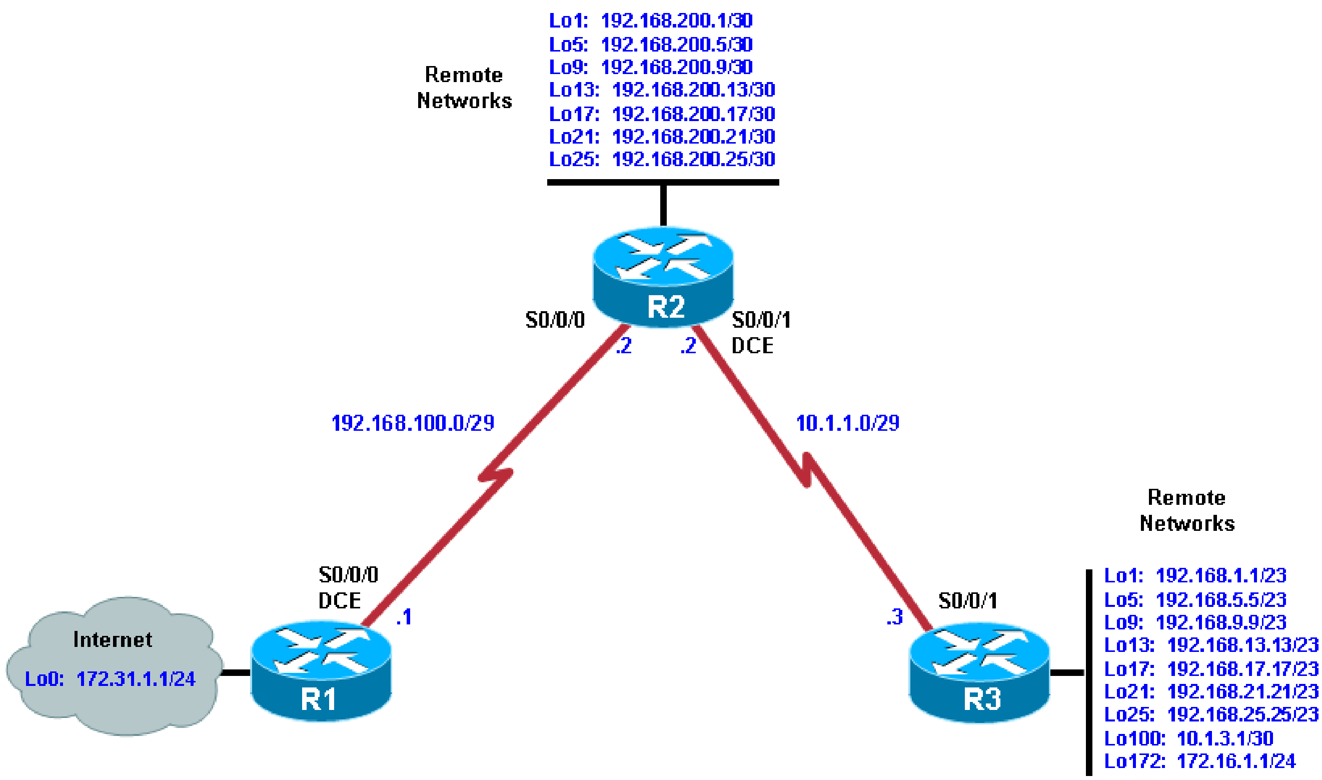
Note: If a deliberate metric manipulation is necessary on a router to force it to prefer one interface over another for EIGRP-discovered routes, it is recommended to use the interface-level command "delay" for these purposes. While the "bandwidth" command can also be used to influence the metrics of EIGRP-discovered routes through a particular interface, it is discouraged because the "bandwidth" will also influence the amount of bandwidth reserved for EIGRP packets and other IOS subsystems as well. The "delay" parameter specifies the value of the interface delay that is used exclusively by EIGRP to perform metric calculations and does not influence any other area of IOS operation.

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 2 Lab 2-3, EIGRP Summarization and Default Network Advertisement

Topology



Objectives

- Review a basic EIGRP configuration.
- Configure and verify EIGRP auto-summarization.
- Configure and verify EIGRP manual summarization.
- Use **show** and **debug** commands for EIGRP summarization.
- Configure default network advertisement.
- Consider the effects of summarization and default routes in a large internetwork.

Background

A network engineer has been having trouble with high memory, bandwidth, and CPU utilization on routers that are running EIGRP. Over lunch, the engineer mentions to you that routes in remote parts of the EIGRP autonomous system are flapping, indicating a performance impediment. The engineer's network has only one path out to the Internet, and the ISP has mandated that 172.31.1.1/24 be used on the end of the backbone connection.

After asking if you could take a look at the network, you discover that the routing tables are filled with 29-bit and 30-bit IP network prefixes, some of which are unstable and flapping. You observe that summarization would result in a dramatic improvement in network performance and volunteer to implement it.

The engineer asks you to show proof-of-concept in the lab first, so you copy the configuration files to paste into your lab routers.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Configure the addressing and serial links.

a. Paste the following configurations into your routers to simulate this network. Save the configurations.

Router R1

```
hostname R1
!
interface Loopback0
 ip address 172.31.1.1 255.255.255.0
!
interface Serial0/0/0
 bandwidth 64
 ip address 192.168.100.1 255.255.255.248
 clock rate 64000
 no shutdown
!
router eigrp 100
 network 172.31.0.0
 network 192.168.100.0
 no auto-summary
!
end
```

Router R2

```
hostname R2
!
interface Loopback1
 ip address 192.168.200.1 255.255.255.252
!
interface Loopback5
 ip address 192.168.200.5 255.255.255.252
!
interface Loopback9
 ip address 192.168.200.9 255.255.255.252
!
interface Loopback13
 ip address 192.168.200.13 255.255.255.252
!
interface Loopback17
```



```
 ip address 192.168.200.17 255.255.255.252
!
interface Loopback21
 ip address 192.168.200.21 255.255.255.252
!
interface Loopback25
 ip address 192.168.200.25 255.255.255.252
!
interface Serial0/0/0
 bandwidth 64
 ip address 192.168.100.2 255.255.255.248
 no shutdown
!
interface Serial0/0/1
 bandwidth 64
 ip address 10.1.1.2 255.255.255.248
 clock rate 64000
 no shutdown
!
router eigrp 100
 network 10.0.0.0
 network 192.168.100.0
 network 192.168.200.0
 no auto-summary
!
end
```

Router R3

```
hostname R3
!
interface Loopback1
 ip address 192.168.1.1 255.255.254.0
!
interface Loopback5
 ip address 192.168.5.5 255.255.254.0
!
interface Loopback9
 ip address 192.168.9.9 255.255.254.0
!
interface Loopback13
 ip address 192.168.13.13 255.255.254.0
!
interface Loopback17
 ip address 192.168.17.17 255.255.254.0
!
interface Loopback21
 ip address 192.168.21.21 255.255.254.0
!
interface Loopback25
 ip address 192.168.25.25 255.255.254.0
!
interface Loopback100
 ip address 10.1.3.1 255.255.255.252
!
interface Loopback172
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0/0/1
```



```
 bandwidth 64
 ip address 10.1.1.3 255.255.255.248
 no shutdown
!
router eigrp 100
 network 10.0.0.0
 network 172.16.0.0
 network 192.168.0.0 0.0.31.255
 no auto-summary
!
end
```

b. Verify that you have full EIGRP adjacency between routers R1 and R2 and between R2 and R3 using the **show ip eigrp neighbors** command.

```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   192.168.100.2           Se0/0/0           10 00:00:13   40  2280  0  38

R2# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   10.1.1.3                Se0/0/1           14 00:00:33    6  2280  0  28
0   192.168.100.1           Se0/0/0           10 00:00:40   21  2280  0  21

R3# show ip eigrp neighbors
IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.1.2                Se0/0/1           13 00:00:52   13  2280  0  37
```

c. Ping all the IP addresses to ensure full connectivity, or use the following Tcl script. If you have never used Tcl scripts or need a refresher, see Lab 1-1.

```
R1# tclsh

foreach address {
10.1.1.2
10.1.1.3
10.1.3.1
172.16.1.1
172.31.1.1
192.168.1.1
192.168.5.5
192.168.9.9
192.168.13.13
192.168.17.17
192.168.21.21
192.168.25.25
192.168.100.1
192.168.200.1
192.168.200.5
192.168.200.9
192.168.200.13
192.168.200.17
192.168.200.21
```



```
192.168.200.25
192.168.100.2
} { ping $address }
```

You should receive ICMP echo replies for each address pinged. Make sure that you run the Tcl script on each router and verify connectivity before you continue with the lab.

Step 2: Analyze summarization options.

Currently, the engineer has the following networks configured within the network:

Router	Interface	IP Address/Mask
R1	Loopback0	172.31.1.1/24
R1	Serial0/0/0	192.168.100.1/29
R2	Loopback1	192.168.200.1/30
R2	Loopback5	192.168.200.5/30
R2	Loopback9	192.168.200.9/30
R2	Loopback13	192.168.200.13/30
R2	Loopback17	192.168.200.17/30
R2	Loopback21	192.168.200.21/30
R2	Loopback25	192.168.200.25/30
R2	Serial0/0/0	192.168.100.2/29
R2	Serial0/0/1	10.1.1.2/29
R3	Loopback1	192.168.1.1/23
R3	Loopback5	192.168.5.5/23
R3	Loopback9	192.168.9.9/23
R3	Loopback13	192.168.13.13/23
R3	Loopback17	192.168.17.17/23
R3	Loopback21	192.168.21.21/23
R3	Loopback25	192.168.25.25/23
R3	Loopback100	10.1.3.1/30
R3	Loopback172	172.16.1.1/24
R3	Serial 0/0/1	10.1.1.3/29

a. Given this addressing scheme, how many major networks are involved in this simulation? What are they?

____
____
____
____
____
____

Note: If you are unsure, use the **show ip route** command on R1 and look at the analysis of the output in Appendix A.

b. The engineer has not configured any automatic or manual EIGRP summarization in the network. How would summarization benefit the network, especially in light of the fact that outlying routes are flapping? List at least two reasons.

____

____

____

____

____

____

____

____

____

c. For the following networks, which router should you summarize to minimize the size of the routing table for all the involved routers? Which summary should you use?

- 10.0.0.0/8 –______________________________
- 172.16.0.0/16 –____________________________
- 172.31.0.0/16 –____________________________
- 192.168.100.0/24 –__________________________
- 192.168.200.0/24 –__________________________
- 192.168.0.0/23 through 192.168.24.0/23 –___________

If EIGRP auto-summarization is turned on in this topology, will 192.168.0.0/23 through 192.168.24.0/23 be summarized?

____

____

d. Because all routes involved in this lab, including later summary routes, will be installed in the routing table by EIGRP, observe the routing table on each router with the **show ip route eigrp** command. You will use this command throughout the lab to periodically observe the routing table.

```
R1# show ip route eigrp
     172.16.0.0/24 is subnetted, 1 subnets
D       172.16.1.0 [90/41152000] via 192.168.100.2, 00:01:14, Serial0/0/0
     192.168.200.0/30 is subnetted, 7 subnets
D       192.168.200.0 [90/40640000] via 192.168.100.2, 00:03:09, Serial0/0/0
D       192.168.200.4 [90/40640000] via 192.168.100.2, 00:03:09, Serial0/0/0
D       192.168.200.8 [90/40640000] via 192.168.100.2, 00:03:09, Serial0/0/0
D       192.168.200.12 [90/40640000] via 192.168.100.2, 00:03:09, Serial0/0/0
D       192.168.200.16 [90/40640000] via 192.168.100.2, 00:03:09, Serial0/0/0
D       192.168.200.20 [90/40640000] via 192.168.100.2, 00:03:09, Serial0/0/0
D       192.168.200.24 [90/40640000] via 192.168.100.2, 00:03:09, Serial0/0/0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.1.3.0/30 [90/41152000] via 192.168.100.2, 00:03:09, Serial0/0/0
D       10.1.1.0/29 [90/41024000] via 192.168.100.2, 00:03:09, Serial0/0/0
```



```
D    192.168.12.0/23 [90/41152000] via 192.168.100.2, 00:03:09, Serial0/0/0
D    192.168.8.0/23 [90/41152000] via 192.168.100.2, 00:03:11, Serial0/0/0
D    192.168.24.0/23 [90/41152000] via 192.168.100.2, 00:03:11, Serial0/0/0
D    192.168.4.0/23 [90/41152000] via 192.168.100.2, 00:03:11, Serial0/0/0
D    192.168.20.0/23 [90/41152000] via 192.168.100.2, 00:03:11, Serial0/0/0
D    192.168.0.0/23 [90/41152000] via 192.168.100.2, 00:03:11, Serial0/0/0
D    192.168.16.0/23 [90/41152000] via 192.168.100.2, 00:03:11, Serial0/0/0

R2# show ip route eigrp
     172.16.0.0/24 is subnetted, 1 subnets
D       172.16.1.0 [90/40640000] via 10.1.1.3, 00:01:40, Serial0/0/1
     172.31.0.0/24 is subnetted, 1 subnets
D       172.31.1.0 [90/40640000] via 192.168.100.1, 00:03:35, Serial0/0/0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.1.3.0/30 [90/40640000] via 10.1.1.3, 00:06:21, Serial0/0/1
D    192.168.12.0/23 [90/40640000] via 10.1.1.3, 00:04:04, Serial0/0/1
D    192.168.8.0/23 [90/40640000] via 10.1.1.3, 00:04:04, Serial0/0/1
D    192.168.24.0/23 [90/40640000] via 10.1.1.3, 00:04:04, Serial0/0/1
D    192.168.4.0/23 [90/40640000] via 10.1.1.3, 00:04:05, Serial0/0/1
D    192.168.20.0/23 [90/40640000] via 10.1.1.3, 00:04:04, Serial0/0/1
D    192.168.0.0/23 [90/40640000] via 10.1.1.3, 00:04:05, Serial0/0/1
D    192.168.16.0/23 [90/40640000] via 10.1.1.3, 00:04:04, Serial0/0/1

R3# show ip route eigrp
     172.31.0.0/24 is subnetted, 1 subnets
D       172.31.1.0 [90/41152000] via 10.1.1.2, 00:04:12, Serial0/0/1
     192.168.200.0/30 is subnetted, 7 subnets
D       192.168.200.0 [90/40640000] via 10.1.1.2, 00:06:58, Serial0/0/1
D       192.168.200.4 [90/40640000] via 10.1.1.2, 00:06:58, Serial0/0/1
D       192.168.200.8 [90/40640000] via 10.1.1.2, 00:06:58, Serial0/0/1
D       192.168.200.12 [90/40640000] via 10.1.1.2, 00:06:58, Serial0/0/1
D       192.168.200.16 [90/40640000] via 10.1.1.2, 00:06:58, Serial0/0/1
D       192.168.200.20 [90/40640000] via 10.1.1.2, 00:06:58, Serial0/0/1
D       192.168.200.24 [90/40640000] via 10.1.1.2, 00:06:58, Serial0/0/1
     192.168.100.0/29 is subnetted, 1 subnets
D       192.168.100.0 [90/41024000] via 10.1.1.2, 00:06:58, Serial0/0/1
```

How do you expect the output of this command to change if you implement the summarization you described above? Record your answer and compare it with the results you observe later.

____

____

____

e. You can also look at the size of each router's routing table with the **show ip route summary** command.

```
R1# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Overhead    Memory (bytes)
connected       0           2           128         304
static          0           0           0           0
eigrp 100       7           10          1088        2584
internal        5                                   5860
Total           12          12          1216        8748

R2# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
```



```
IP routing table maximum-paths is 32
Route Source     Networks     Subnets      Overhead     Memory (bytes)
connected        0            9            576          1368
static           0            0            0            0
eigrp 100        7            3            640          1520
internal         5                                      5860
Total            12           12           1216         8748

R3# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 32
Route Source     Networks     Subnets      Overhead     Memory (bytes)
connected        7            3            640          1520
static           0            0            0            0
eigrp 100        0            9            576          1368
internal         5                                      5860
Total            12           12           1216         8748
```

Step 3: Configure EIGRP auto-summarization.

The network engineer reminds you that EIGRP auto-summarization is turned on by default, but that it was turned off because of discontiguous networks that were later removed. It is now safe to begin using auto-summarization again.

a. Verify that EIGRP AS 100 is not using auto-summarization on R1 with the **show ip protocols** command.

```
R1# show ip protocols
Routing Protocol is "eigrp 100"
  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 100
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.31.0.0
    192.168.100.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.100.2         90      00:04:31
  Distance: internal 90 external 170
```

You will use this command to check whether the following is occurring:

- EIGRP is flagging default networks sent to other routers.
- EIGRP is accepting default networks advertised to this router.
- Auto-summarization is turned on.

b. You can enable EIGRP route and summary route debugging on each router, which allows you to observe when summary routes are advertised from the router, with the **debug ip eigrp 100** and **debug ip eigrp summary** commands.

```
R1# debug ip eigrp 100
R1# debug ip eigrp summary
```

```
R2# debug ip eigrp 100
R2# debug ip eigrp summary

R3# debug ip eigrp 100
R3# debug ip eigrp summary
```

c. On R3, issue the **auto-summary** command in the EIGRP configuration menu. This command produces system logging messages on both routers and debug output on R3.

```
R3(config)# router eigrp 100
R3(config-router)# auto-summary
```

You should see the following types of log messages.

On R3:

```
*Feb 6 16:55:03.035: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.1.1.2
(Serial0/0/1) is resync: summary configured
```

On R2:

```
*Feb 6 16:56:54.539: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.1.1.3
(Serial0/0/1) is resync: peer graceful-restart
```

Your router issues a notification similar to the message on R3 when you either configure or disable auto-summary on the local router. You receive a notification similar to the message on R2 when you configure auto-summary on an adjacent router. The adjacency must be resynchronized so that EIGRP update packets advertising the new summary routing information are sent.

Following the log messages, you get a flood of debug output on R3 as it searches its topology table for routes that can be summarized. EIGRP attempts to automatically summarize both 172.16.0.0/16 and 10.0.0.0/8 on R3 because it hosts the classful boundary between those networks. However, the output has been limited to only the debug messages concerning the 172.16.0.0/16 network. You should receive the same messages for 10.0.0.0/8, with the exception of the addition of the Serial0/0/1 interface. The reason for this exception is explained later.

```
<Output regarding network 10.0.0.0/8 is omitted.>

*Feb  6 19:23:37.811: IP-EIGRP: add_auto_summary: Serial0/0/1 172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP: find_summary: add new sum: 172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP: find_summary: add new if: Serial0/0/1 to
172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP(Default-IP-Routing-Table:100):
process_summary: 172.16.0.0/16 1
*Feb  6 19:23:37.811: IP-EIGRP: add_auto_summary: Loopback100 172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP: find_summary: add new if: Loopback100 to
172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP(Default-IP-Routing-Table:100):
process_summary: 172.16.0.0/16 1
*Feb  6 19:23:37.811: IP-EIGRP: add_auto_summary: Loopback1 172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP: find_summary: add new if: Loopback1 to
172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP(Default-IP-Routing-Table:100):
process_summary: 172.16.0.0/16 1
*Feb  6 19:23:37.811: IP-EIGRP: add_auto_summary: Loopback5 172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP: find_summary: add new if: Loopback5 to
172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP(Default-IP-Routing-Table:100):
process_summary: 172.16.0.0/16 1
```



```
*Feb  6 19:23:37.811: IP-EIGRP: add_auto_summary: Loopback9 172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP: find_summary: add new if: Loopback9 to
172.16.0.0/16 5
*Feb  6 19:23:37.811: IP-EIGRP(Default-IP-Routing-Table:100):
process_summary: 172.16.0.0/16 1
*Feb  6 19:23:37.811: IP-EIGRP: add_auto_summary: Loopback13 172.16.0.0/16 5
*Feb  6 19:23:37.815: IP-EIGRP: find_summary: add new if: Loopback13 to
172.16.0.0/16 5
*Feb  6 19:23:37.815: IP-EIGRP(Default-IP-Routing-Table:100):
process_summary: 172.16.0.0/16 1
*Feb  6 19:23:37.815: IP-EIGRP: add_auto_summary: Loopback17 172.16.0.0/16 5
*Feb  6 19:23:37.815: IP-EIGRP: find_summary: add new if: Loopback17 to
172.16.0.0/16 5
*Feb  6 19:23:37.815: IP-EIGRP(Default-IP-Routing-Table:100):
process_summary: 172.16.0.0/16 1
*Feb  6 19:23:37.815: IP-EIGRP: add_auto_summary: Loopback21 172.16.0.0/16 5
*Feb  6 19:23:37.815: IP-EIGRP: find_summary: add new if: Loopback21 to
172.16.0.0/16 5
*Feb  6 19:23:37.815: IP-EIGRP(Default-IP-Routing-Table:100):
process_summary: 172.16.0.0/16 1
*Feb  6 19:23:37.815: IP-EIGRP: add_auto_summary: Loopback25 172.16.0.0/16 5
*Feb  6 19:23:37.815: IP-EIGRP: find_summary: add new if: Loopback25 to
172.16.0.0/16 5
*Feb  6 19:23:37.815: IP-EIGRP(Default-IP-Routing-Table:100):
process_summary: 172.16.0.0/16 1
*Feb  6 19:23:37.815: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.1.1.2
(Serial0/0/1) is resync: summary configured
*Feb  6 19:23:37.815: IP-EIGRP(Default-IP-Routing-Table:100):
get_summary_metric: 172.16.0.0/16
*Feb  6 19:23:37.819: IP-EIGRP(Default-IP-Routing-Table:100):
get_summary_metric: 172.16.0.0/16
*Feb  6 19:23:37.819: IP-EIGRP(Default-IP-Routing-Table:100):
get_summary_metric: 172.16.0.0/16
*Feb  6 19:23:37.823: IP-EIGRP(Default-IP-Routing-Table:100):
get_summary_metric: 172.16.0.0/16
*Feb  6 19:23:37.823: IP-EIGRP(Default-IP-Routing-Table:100):
get_summary_metric: 172.16.0.0/16
*Feb  6 19:23:37.823: IP-EIGRP(Default-IP-Routing-Table:100):
get_summary_metric: 172.16.0.0/16
*Feb  6 19:23:37.827: IP-EIGRP(Default-IP-Routing-Table:100):
get_summary_metric: 172.16.0.0/16
*Feb  6 19:23:37.827: IP-EIGRP(Default-IP-Routing-Table:100):
get_summary_metric: 172.16.0.0/16
*Feb  6 19:23:37.831: IP-EIGRP(Default-IP-Routing-Table:100):
get_summary_metric: 172.16.0.0/16
```

Each get_summary_metric message at the end represents a function call to create a composite metric for the summary route for each outbound interface.

Imagine that you have EIGRP neighbors out each loopback interface connected to R3. How many interfaces will receive the 172.16.0.0/16 summary route?

____

____

____

____

____

Which summary routes are sent to R2 from R3?

____

d. Check which summary routes are sent with the **show ip route eigrp** command.

```
R2# show ip route eigrp
D    172.16.0.0/16 [90/40640000] via 10.1.1.3, 00:38:38, Serial0/0/1
     172.31.0.0/24 is subnetted, 1 subnets
D       172.31.1.0 [90/40640000] via 192.168.100.1, 00:47:51, Serial0/0/0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.1.3.0/30 [90/40640000] via 10.1.1.3, 00:50:36, Serial0/0/1
D    192.168.12.0/23 [90/40640000] via 10.1.1.3, 00:48:20, Serial0/0/1
D    192.168.8.0/23 [90/40640000] via 10.1.1.3, 00:48:20, Serial0/0/1
D    192.168.24.0/23 [90/40640000] via 10.1.1.3, 00:48:19, Serial0/0/1
D    192.168.4.0/23 [90/40640000] via 10.1.1.3, 00:48:20, Serial0/0/1
D    192.168.20.0/23 [90/40640000] via 10.1.1.3, 00:48:19, Serial0/0/1
D    192.168.0.0/23 [90/40640000] via 10.1.1.3, 00:48:20, Serial0/0/1
D    192.168.16.0/23 [90/40640000] via 10.1.1.3, 00:48:20, Serial0/0/1
```

Notice that the summary route has the same composite metric as the previous single route to 172.16.1.0/30.

When the summary route is generated, what happens in the R3 routing table?

____

____

e. Issue the **show ip route eigrp** command to check for the summary routes to null0.

```
R3# show ip route eigrp
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
D       172.16.0.0/16 is a summary, 00:14:57, Null0
     172.31.0.0/24 is subnetted, 1 subnets
D       172.31.1.0 [90/41152000] via 10.1.1.2, 00:15:24, Serial0/0/1
     192.168.200.0/30 is subnetted, 7 subnets
D       192.168.200.0 [90/40640000] via 10.1.1.2, 00:15:24, Serial0/0/1
D       192.168.200.4 [90/40640000] via 10.1.1.2, 00:15:24, Serial0/0/1
D       192.168.200.8 [90/40640000] via 10.1.1.2, 00:15:24, Serial0/0/1
D       192.168.200.12 [90/40640000] via 10.1.1.2, 00:15:24, Serial0/0/1
D       192.168.200.16 [90/40640000] via 10.1.1.2, 00:15:24, Serial0/0/1
D       192.168.200.20 [90/40640000] via 10.1.1.2, 00:15:24, Serial0/0/1
D       192.168.200.24 [90/40640000] via 10.1.1.2, 00:15:24, Serial0/0/1
     10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
D       10.0.0.0/8 is a summary, 00:14:57, Null0
     192.168.100.0/29 is subnetted, 1 subnets
D       192.168.100.0 [90/41024000] via 10.1.1.2, 00:15:24, Serial0/0/1
```

The output of the **debug ip eigrp summary** command also contained messages pertaining to 10.0.0.0/8. Although R3 has a summary route for 10.0.0.0/8 installed in its routing table to Null0, why did R3 not send the summary route for 10.0.0.0/8 to R2?

The 10.0.0.0/8 summary will not be sent out to a connected subnet within that major network. Automatic summarization takes place at the classful boundary by sending a classful network summary to all local EIGRP interfaces not in the summarized network. The automatic summarization takes place only if a subnet of a particular major network is going to be advertised through an interface that is itself in a different major network. Because Serial0/0/1 has an IP address that is part of the 10.0.0.0/8 network, R3 does not send that summary to R2 through the Serial0/0/1 interface. Notice that it is not in the EIGRP topology table on R2.

```
R2# show ip eigrp topology
IP-EIGRP Topology Table for AS(100)/ID(192.168.200.25)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.0/30, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 10.1.1.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/1
P 192.168.100.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/0
P 192.168.8.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.12.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.0.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.4.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.24.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.16.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.20.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.200.0/30, 1 successors, FD is 128256
        via Connected, Loopback1
P 192.168.200.4/30, 1 successors, FD is 128256
        via Connected, Loopback5
P 192.168.200.8/30, 1 successors, FD is 128256
        via Connected, Loopback9
P 192.168.200.12/30, 1 successors, FD is 128256
        via Connected, Loopback13
P 192.168.200.16/30, 1 successors, FD is 128256
        via Connected, Loopback17
P 172.31.1.0/24, 1 successors, FD is 40640000
        via 192.168.100.1 (40640000/128256), Serial0/0/0
P 192.168.200.20/30, 1 successors, FD is 128256
        via Connected, Loopback21
P 192.168.200.24/30, 1 successors, FD is 128256
        via Connected, Loopback25
P 172.16.0.0/16, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
```

Which of the R3 connected networks are not being summarized?

Review your answers to the questions at the end of Step 2. Why is this summarization not occurring?

____

____

____

f. Because the engineer has no discontiguous networks in the internetwork, you decide to enable EIGRP auto-summary on all routers.

```
R1(config)# router eigrp 100
R1(config-router)# auto-summary

R2(config)# router eigrp 100
R2(config-router)# auto-summary
```

g. Verify that the summaries are shown by issuing the **show ip eigrp topology** command on each router. You should see summary routes on each router for each major network that is not part of the /23 supernet. Supernets are not included in auto-summary routes because EIGRP automatically summarizes only to the classful boundary and no further. Compare your output with the output below.

```
R1# show ip eigrp topology
IP-EIGRP Topology Table for AS(100)/ID(172.31.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.0.0.0/8, 1 successors, FD is 41024000
        via 192.168.100.2 (41024000/40512000), Serial0/0/0
P 192.168.100.0/24, 1 successors, FD is 40512000
        via Summary (40512000/0), Null0
P 192.168.100.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/0
P 192.168.8.0/23, 1 successors, FD is 41152000
        via 192.168.100.2 (41152000/40640000), Serial0/0/0
P 192.168.12.0/23, 1 successors, FD is 41152000
        via 192.168.100.2 (41152000/40640000), Serial0/0/0
P 192.168.0.0/23, 1 successors, FD is 41152000
        via 192.168.100.2 (41152000/40640000), Serial0/0/0
P 192.168.4.0/23, 1 successors, FD is 41152000
        via 192.168.100.2 (41152000/40640000), Serial0/0/0
P 192.168.24.0/23, 1 successors, FD is 41152000
        via 192.168.100.2 (41152000/40640000), Serial0/0/0
P 192.168.16.0/23, 1 successors, FD is 41152000
        via 192.168.100.2 (41152000/40640000), Serial0/0/0
P 192.168.20.0/23, 1 successors, FD is 41152000
        via 192.168.100.2 (41152000/40640000), Serial0/0/0
P 192.168.200.0/24, 1 successors, FD is 40640000
        via 192.168.100.2 (40640000/128256), Serial0/0/0
P 172.31.1.0/24, 1 successors, FD is 128256
        via Connected, Loopback0
P 172.31.0.0/16, 1 successors, FD is 128256
        via Summary (128256/0), Null0
P 172.16.0.0/16, 1 successors, FD is 41152000
        via 192.168.100.2 (41152000/40640000), Serial0/0/0

R2# show ip eigrp topology
IP-EIGRP Topology Table for AS(100)/ID(192.168.200.25)
```



```
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.0/30, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 10.0.0.0/8, 1 successors, FD is 40512000
        via Summary (40512000/0), Null0
P 10.1.1.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/1
P 192.168.100.0/24, 1 successors, FD is 40512000
        via Summary (40512000/0), Null0
P 192.168.100.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/0
P 192.168.8.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.12.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.0.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.4.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.24.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.16.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.20.0/23, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1
P 192.168.200.0/24, 1 successors, FD is 128256
        via Summary (128256/0), Null0
P 192.168.200.0/30, 1 successors, FD is 128256
        via Connected, Loopback1
P 192.168.200.4/30, 1 successors, FD is 128256
        via Connected, Loopback5
P 192.168.200.8/30, 1 successors, FD is 128256
        via Connected, Loopback9
P 192.168.200.12/30, 1 successors, FD is 128256
        via Connected, Loopback13
P 192.168.200.16/30, 1 successors, FD is 128256
        via Connected, Loopback17
P 172.31.0.0/16, 1 successors, FD is 40640000
        via 192.168.100.1 (40640000/128256), Serial0/0/0
P 192.168.200.20/30, 1 successors, FD is 128256
        via Connected, Loopback21
P 192.168.200.24/30, 1 successors, FD is 128256
        via Connected, Loopback25
P 172.16.0.0/16, 1 successors, FD is 40640000
        via 10.1.1.3 (40640000/128256), Serial0/0/1

R3# show ip eigrp topology
IP-EIGRP Topology Table for AS(100)/ID(192.168.25.25)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.1.3.0/30, 1 successors, FD is 128256
        via Connected, Loopback100
P 10.0.0.0/8, 1 successors, FD is 128256
        via Summary (128256/0), Null0
```



```
P 10.1.1.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/1
P 192.168.100.0/24, 1 successors, FD is 41024000
        via 10.1.1.2 (41024000/40512000), Serial0/0/1
P 192.168.8.0/23, 1 successors, FD is 128256
        via Connected, Loopback9
P 192.168.12.0/23, 1 successors, FD is 128256
        via Connected, Loopback13
P 192.168.0.0/23, 1 successors, FD is 128256
        via Connected, Loopback1
P 192.168.4.0/23, 1 successors, FD is 128256
        via Connected, Loopback5
P 192.168.24.0/23, 1 successors, FD is 128256
        via Connected, Loopback25
P 192.168.16.0/23, 1 successors, FD is 128256
        via Connected, Loopback17
P 192.168.20.0/23, 1 successors, FD is 128256
        via Connected, Loopback21
P 192.168.200.0/24, 1 successors, FD is 40640000
        via 10.1.1.2 (40640000/128256), Serial0/0/1
P 172.31.0.0/16, 1 successors, FD is 41152000
        via 10.1.1.2 (41152000/40640000), Serial0/0/1
P 172.16.0.0/16, 1 successors, FD is 128256
        via Summary (128256/0), Null0
P 172.16.1.0/24, 1 successors, FD is 128256
        via Connected, Loopback172
```

Step 4: Configure EIGRP manual summarization.

EIGRP calculates summaries, whether manually or automatically, on a per-interface basis. Recall that when you configured auto-summary, the debug output showed that EIGRP summary routes were generated on a per-interface basis. The EIGRP **auto-summary** command turns auto-summarization on globally on a router, but you can also configure summary routes manually with the interface-level command **ip summary-address eigrp** *as* ***network*** *mask*.

Note: Combining manual and automatic summarization is not a best practice. If both manual and automatic summarization are activated, EIGRP sends both the automatic and the manual summary route out an interface. Normally, you need to leave EIGRP auto-summarization off, especially in topologies with discontiguous networks, and create manual summary routes instead. For this scenario, you enable manual summarization on the R3 Serial0/0/1 interface to show the engineer how summarization can further benefit the network. R3 should advertise the /23 subnets to R2.

a. What is the most efficient mask to summarize these routes?

____

____

b. Implement the summarization on R3.

```
R3(config)# interface Serial 0/0/1
R3(config-if)# ip summary-address eigrp 100 192.168.0.0 255.255.224.0
```

The **100** parameter specifies that the summarization be sent out only to neighbors in EIGRP AS 100.

Note: If you are unfamiliar with the parameters of this command, use the **?** for the inline Cisco IOS help system. It is recommended that you use the help system to familiarize yourself with parameters when working through these labs.

The adjacency between R2 and R3 resynchronizes after the summary is configured, as indicated by the debug messages. The routing tables should appear similar to the following.

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

Gateway of last resort is not set

D    172.16.0.0/16 [90/41152000] via 192.168.100.2, 04:04:11, Serial0/0/0
     172.31.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.31.1.0/24 is directly connected, Loopback0
D       172.31.0.0/16 is a summary, 02:47:43, Null0
D    192.168.200.0/24 [90/40640000] via 192.168.100.2, 02:47:34, Serial0/0/0
D    10.0.0.0/8 [90/41024000] via 192.168.100.2, 02:47:34, Serial0/0/0
     192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.100.0/29 is directly connected, Serial0/0/0
D       192.168.100.0/24 is a summary, 02:47:44, Null0
D    192.168.0.0/19 [90/41152000] via 192.168.100.2, 02:32:07, Serial0/0/0

R2# show ip route
<output omitted>

Gateway of last resort is not set

D    172.16.0.0/16 [90/40640000] via 10.1.1.3, 02:33:29, Serial0/0/1
D    172.31.0.0/16 [90/40640000] via 192.168.100.1, 02:48:58, Serial0/0/0
     192.168.200.0/24 is variably subnetted, 8 subnets, 2 masks
C       192.168.200.0/30 is directly connected, Loopback1
D       192.168.200.0/24 is a summary, 02:48:58, Null0
C       192.168.200.4/30 is directly connected, Loopback5
C       192.168.200.8/30 is directly connected, Loopback9
C       192.168.200.12/30 is directly connected, Loopback13
C       192.168.200.16/30 is directly connected, Loopback17
C       192.168.200.20/30 is directly connected, Loopback21
C       192.168.200.24/30 is directly connected, Loopback25
     10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
D       10.1.3.0/30 [90/40640000] via 10.1.1.3, 02:33:30, Serial0/0/1
C       10.1.1.0/29 is directly connected, Serial0/0/1
D       10.0.0.0/8 is a summary, 02:49:00, Null0
     192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.100.0/29 is directly connected, Serial0/0/0
D       192.168.100.0/24 is a summary, 02:49:00, Null0
D    192.168.0.0/19 [90/40640000] via 10.1.1.3, 02:33:31, Serial0/0/1

R3# show ip route
<output omitted>

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
D       172.16.0.0/16 is a summary, 04:07:05, Null0
C       172.16.1.0/24 is directly connected, Loopback172
     172.31.0.0/16 is subnetted, 1 subnets
D       172.31.0.0 [90/41152000] via 10.1.1.2, 02:35:00, Serial0/0/1
D    192.168.200.0/24 [90/40640000] via 10.1.1.2, 02:50:28, Serial0/0/1
     10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C       10.1.3.0/30 is directly connected, Loopback100
C       10.1.1.0/29 is directly connected, Serial0/0/1
```



```
D       10.0.0.0/8 is a summary, 04:07:06, Null0
D    192.168.100.0/24 [90/41024000] via 10.1.1.2, 02:50:29, Serial0/0/1
C    192.168.12.0/23 is directly connected, Loopback13
C    192.168.8.0/23 is directly connected, Loopback9
C    192.168.24.0/23 is directly connected, Loopback25
C    192.168.4.0/23 is directly connected, Loopback5
C    192.168.20.0/23 is directly connected, Loopback21
C    192.168.0.0/23 is directly connected, Loopback1
D    192.168.0.0/19 is a summary, 02:35:02, Null0
C    192.168.16.0/23 is directly connected, Loopback17
```

Notice that on each router the only EIGRP routes (marked as D) are summary routes to locally connected networks (Null0) or to remote networks, both of which reduce the number of advertised networks.

At this point, you have efficiently summarized the network. Based on your knowledge of routing protocols and techniques, are there any other ways to minimize the routing table even further for this topology without filtering routes?

____

____

____

____

Step 5: Configure default network advertisement.

Suppose this engineer has another branch office of the core network that is also running EIGRP in a different autonomous system, AS 200, connected to the FastEthernet0/0 interface on R1. However, the branch you are modeling is completely independent of that topology and vice versa.

Based on this corporation's new routing policies, EIGRP AS 100 only needs to know that all traffic out of its network is forwarded to R1. The engineer queries you as to how connectivity can be preserved to AS 200 networks, while minimizing routing tables within AS 100.

a. What solutions would you propose?

____

____

____

You decide that this company's policies are in line with the use of a default route out of the system. The default network that you will configure is 172.31.0.0/16, because this is the path to the Internet.

The IP network 0.0.0.0/0 matches all unknown destination prefixes because the routing table acts in a classless manner. Classless routing tables use the first match based on the longest IP subnet mask for that destination network. Therefore, if the routing table has no matches for a subnet mask greater than 0 bits for a given destination network, the shortest subnet mask (/0) matches any of the 32 bits of a destination network.

For instance, if the router does not have a route to 192.168.7.0/24, it tries to match against any routes it has to 192.168.6.0/23, 192.168.4.0/22, 192.168.0.0/21, and so on. If it does not find any matching routes, it eventually gets to the 0.0.0.0/0 network, which matches all destination IP addresses, and sends the packet to its "gateway of last resort."

b. The **ip default-network** command propagates through the EIGRP system so that each router sees its candidate default network as the path with the shortest feasible distance to the default network (172.31.0.0/16). Issue this command on R1.

   Note: There are different methods to propagate a default route in EIGRP. Because EIGRP does not have the **default-information originate** command, this example uses the **ip default-network** command.

```
R1(config)# ip default-network 172.31.0.0
```

   This command routes all traffic through R1 to destination networks not matching any other networks or subnets in the routing table to the 172.31.0.0 network. EIGRP flags this route as the default route in advertisements to other routers.

c. Verify that the flag is set on updates to R2 using the **show ip eigrp topology 172.31.0.0/16** command.

```
R2# show ip eigrp topology 172.31.0.0/16
IP-EIGRP (AS 100): Topology entry for 172.31.0.0/16
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 40640000
  Routing Descriptor Blocks:
  192.168.100.1 (Serial0/0/0), from 192.168.100.1, Send flag is 0x0
      Composite metric is (40640000/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
      Exterior flag is set
```

d. Use the **show ip route** command to view how the routing table has changed on each router.

```
R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 172.31.0.0

D    172.16.0.0/16 [90/41152000] via 192.168.100.2, 06:32:23, Serial0/0/0
*   172.31.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.31.1.0/24 is directly connected, Loopback0
D*      172.31.0.0/16 is a summary, 00:02:04, Null0
D    192.168.200.0/24 [90/40640000] via 192.168.100.2, 05:15:46, Serial0/0/0
D    10.0.0.0/8 [90/41024000] via 192.168.100.2, 05:15:46, Serial0/0/0
     192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.100.0/29 is directly connected, Serial0/0/0
D       192.168.100.0/24 is a summary, 05:15:56, Null0
D    192.168.0.0/19 [90/41152000] via 192.168.100.2, 05:00:19, Serial0/0/0

R2# show ip route
Codes: C - connected, S - static, 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
```



```
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.100.1 to network 172.31.0.0

D    172.16.0.0/16 [90/40640000] via 10.1.1.3, 04:58:38, Serial0/0/1
D*   172.31.0.0/16 [90/40640000] via 192.168.100.1, 00:00:09, Serial0/0/0
     192.168.200.0/24 is variably subnetted, 8 subnets, 2 masks
C       192.168.200.0/30 is directly connected, Loopback1
D       192.168.200.0/24 is a summary, 05:14:07, Null0
C       192.168.200.4/30 is directly connected, Loopback5
C       192.168.200.8/30 is directly connected, Loopback9
C       192.168.200.12/30 is directly connected, Loopback13
C       192.168.200.16/30 is directly connected, Loopback17
C       192.168.200.20/30 is directly connected, Loopback21
C       192.168.200.24/30 is directly connected, Loopback25
     10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
D       10.1.3.0/30 [90/40640000] via 10.1.1.3, 04:58:39, Serial0/0/1
C       10.1.1.0/29 is directly connected, Serial0/0/1
D       10.0.0.0/8 is a summary, 05:14:09, Null0
     192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.100.0/29 is directly connected, Serial0/0/0
D       192.168.100.0/24 is a summary, 05:14:09, Null0
D    192.168.0.0/19 [90/40640000] via 10.1.1.3, 04:58:40, Serial0/0/1

R3# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.1.2 to network 172.31.0.0

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
D       172.16.0.0/16 is a summary, 06:37:06, Null0
C       172.16.1.0/24 is directly connected, Loopback172
D*   172.31.0.0/16 [90/41152000] via 10.1.1.2, 00:06:32, Serial0/0/1
D    192.168.200.0/24 [90/40640000] via 10.1.1.2, 05:20:29, Serial0/0/1
     10.0.0.0/8 is variably subnetted, 3 subnets, 3 masks
C       10.1.3.0/30 is directly connected, Loopback100
C       10.1.1.0/29 is directly connected, Serial0/0/1
D       10.0.0.0/8 is a summary, 06:37:07, Null0
D    192.168.100.0/24 [90/41024000] via 10.1.1.2, 05:20:31, Serial0/0/1
C    192.168.12.0/23 is directly connected, Loopback13
C    192.168.8.0/23 is directly connected, Loopback9
C    192.168.24.0/23 is directly connected, Loopback25
C    192.168.4.0/23 is directly connected, Loopback5
C    192.168.20.0/23 is directly connected, Loopback21
C    192.168.0.0/23 is directly connected, Loopback1
D    192.168.0.0/19 is a summary, 05:05:22, Null0
C    192.168.16.0/23 is directly connected, Loopback17
```


e. On R1, the gateway of last resort is designated as 172.31.0.0. What is the IP address of the gateway of last resort on R2 and R3?

f. What are the benefits of introducing the routing information of the other autonomous system into EIGRP AS 100?

g. What are the drawbacks of configuring the default network to propagate from R1?

h. If R3 were to ping a destination network that is not reachable from this internetwork, how far would the data travel?

If the packets must travel to R1 before being dropped, does this make the network more or less susceptible to denial of service (DoS) attacks from within?

Which routers in this scenario could be overloaded by such unreachable traffic?

i. Always consider the benefits and drawbacks in summarization and using default routing techniques before implementing them in an internetwork. These tools are useful in decreasing the size of a routing table, but might have drawbacks as well based on your topology. For instance, auto-summarization should not be used in topologies with discontiguous networks.

What would happen if the connection to the Internet on R1 were a subnet of the 172.16.0.0/16 network?

____

____

____

____

____

____

____

Step 6: Verify summarization and routing table efficiencies achieved.

a. Issue the **show ip protocols** command again. How has the output changed?

```
R1# show ip protocols
Routing Protocol is "eigrp 100"
  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 100
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Automatic address summarization:
    192.168.100.0/24 for Loopback0
      Summarizing with metric 40512000
    172.31.0.0/16 for Serial0/0/0
      Summarizing with metric 128256
  Maximum path: 4
  Routing for Networks:
    172.31.0.0
    192.168.100.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)         90      00:23:10
    Gateway         Distance      Last Update
    192.168.100.2         90      00:30:32
  Distance: internal 90 external 170
```

b. Run the Tcl script from Step 1 again. The pings should be successful.

When configuring a major network change such as summarization and default network, always test to see whether you have achieved the desired effect within the core paths and the outlying branches.

c. The engineer still wants to know if all of these solutions decreased the size of the routing table as you claimed. Display the size of the routing table on R1, R2, and R3 with the **show ip route summary** command you used at the end of Step 2.

Before snapshot (initial configuration from Step 1):


```
R1# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Overhead    Memory (bytes)
connected       0           2           128         304
static          0           0           0           0
eigrp 100       7           10          1088        2584
internal        5                                   5860
Total           12          12          1216        8748

R2# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Overhead    Memory (bytes)
connected       0           9           576         1368
static          0           0           0           0
eigrp 100       7           3           640         1520
internal        5                                   5860
Total           12          12          1216        8748

R3# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Overhead    Memory (bytes)
connected       7           3           640         1520
static          0           0           0           0
eigrp 100       0           9           576         1368
internal        5                                   5860
Total           12          12          1216        8748
```

After snapshot (after configuring auto-summary, R3 summary address, and default network on R1)

```
R1# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Overhead    Memory (bytes)
connected       0           2           128         304
static          0           0           0           0
eigrp 100       4           2           384         2952
internal        2                                   2344
Total           6           4           512         5600

R2# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Overhead    Memory (bytes)
connected       0           9           576         1368
static          0           0           0           0
eigrp 100       3           4           448         3104
internal        3                                   3516
Total           6           13          1024        7988

R3# show ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 32
Route Source    Networks    Subnets     Overhead    Memory (bytes)
connected       7           3           640         1520
static          0           0           0           0
eigrp 100       3           3           384         3972
```



```
internal        3                                        3516
Total           13          6            1024            9008
```

d. By what amount has the total routing table size decreased on each router? Depending on the equipment in your lab, your answers may vary.

____

____

____

With the equipment used in this lab, the most significant change is on R1. On R1, the routing table has decreased by 3148 bytes, which is a 36 percent decrease from its initial size. On R2, the routing table has decreased by 760 bytes, which is a 9 percent decrease. On R3, the routing table has actually increased slightly by 260 bytes, which is a 3 percent increase. This increase is due to the increase in the memory usage by the major network entries in the routing table learned via EIGRP, as compared to the base configuration.

Although this may seem like a trivial amount in terms of bytes, it is important to understand the principles involved and the outcome of a much more converged, scalable routing table. Consider also that summaries cause less EIGRP query, reply, update, and ACK packets to be sent to neighbors every time an EIGRP interface flaps. Queries can be propagated far beyond the local link and, by default, EIGRP might consume up to 50 percent of the bandwidth with its traffic. This amount could have severe repercussions on bandwidth consumption on a link.

Consider also the routing table of the Internet and how candidate default routing within an enterprise network can help minimize routing tables by routing traffic to a dynamically identified outbound path from a network. For enterprise-level networks, the amount of space and CPU utilization saved in storing topology and routing tables and maintaining routing tables with constant changes can be an important method for developing a faster and more converged network.

Appendix A: Analyzing Major Networks

The output of the **show ip route** command in this scenario is somewhat complicated but useful to understand because you will see similar output in production networks. This output involves both subnets and supernets as well as the major networks themselves as group headings.

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

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
D       172.16.1.0 [90/41152000] via 192.168.100.2, 00:10:31, Serial0/0/0
     172.31.0.0/24 is subnetted, 1 subnets
C       172.31.1.0 is directly connected, Loopback0
     192.168.200.0/30 is subnetted, 7 subnets
D       192.168.200.0 [90/40640000] via 192.168.100.2, 00:11:14, Serial0/0/0
D       192.168.200.4 [90/40640000] via 192.168.100.2, 00:11:14, Serial0/0/0
D       192.168.200.8 [90/40640000] via 192.168.100.2, 00:11:14, Serial0/0/0
D       192.168.200.12 [90/40640000] via 192.168.100.2, 00:11:15, Serial0/0/0
D       192.168.200.16 [90/40640000] via 192.168.100.2, 00:11:15, Serial0/0/0
D       192.168.200.20 [90/40640000] via 192.168.100.2, 00:11:15, Serial0/0/0
D       192.168.200.24 [90/40640000] via 192.168.100.2, 00:11:15, Serial0/0/0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.1.3.0/30 [90/41152000] via 192.168.100.2, 00:10:32, Serial0/0/0
D       10.1.1.0/29 [90/41024000] via 192.168.100.2, 00:10:39, Serial0/0/0
     192.168.100.0/29 is subnetted, 1 subnets
C       192.168.100.0 is directly connected, Serial0/0/0
D    192.168.12.0/23 [90/41152000] via 192.168.100.2, 00:10:32, Serial0/0/0
D    192.168.8.0/23 [90/41152000] via 192.168.100.2, 00:10:32, Serial0/0/0
D    192.168.24.0/23 [90/41152000] via 192.168.100.2, 00:10:32, Serial0/0/0
D    192.168.4.0/23 [90/41152000] via 192.168.100.2, 00:10:32, Serial0/0/0
D    192.168.20.0/23 [90/41152000] via 192.168.100.2, 00:10:32, Serial0/0/0
D    192.168.0.0/23 [90/41152000] via 192.168.100.2, 00:10:33, Serial0/0/0
D    192.168.16.0/23 [90/41152000] via 192.168.100.2, 00:10:33, Serial0/0/0
R1#
```

Notice that the output of the **show ip route** command displays all subnets of a given major network grouped by major network:

- 10.0.0.0/8
- 172.16.0.0/16
- 172.31.0.0/16
- 192.168.100.0/24
- 192.168.200.0/24

Each /23 supernet consists of two major networks combined into one /23. For example, the 192.168.0.0/23 network covers the major networks 192.168.0.0/24 and 192.168.1.0/24.

Why do 172.16.0.0/24, 172.31.0.0/24, 192.168.100.0/30, and 192.168.200.0/29 appear as group headings with longer masks than the classful mask?

When you subnet a major network into subnets that all have the same mask and advertise those networks to a router, the routing table simply decides that it will do all lookups for that major network in a classless way using the mask provided. The routing table is not expecting any variable-length subnet masks (VLSMs) for those major networks because it has not yet learned of any. Therefore, the headings listed above display as the headings in the routing table.

Analyze the output of the **show ip route** command as follows:

- 172.16.0.0/24 indicates that the 172.16.0.0/16 major network is only divided into subnets of 24-bit masks.
- 172.31.0.0/24 indicates that the 172.31.0.0/16 major network is only divided into subnets of 24-bit masks.
- 192.168.100.0/30 indicates that the 192.168.100.0/24 major network is only divided into subnets of 30-bit masks.
- 192.168.200.0/29 indicates that the 192.168.200.0/24 major network is only divided into subnets of 29-bit masks.

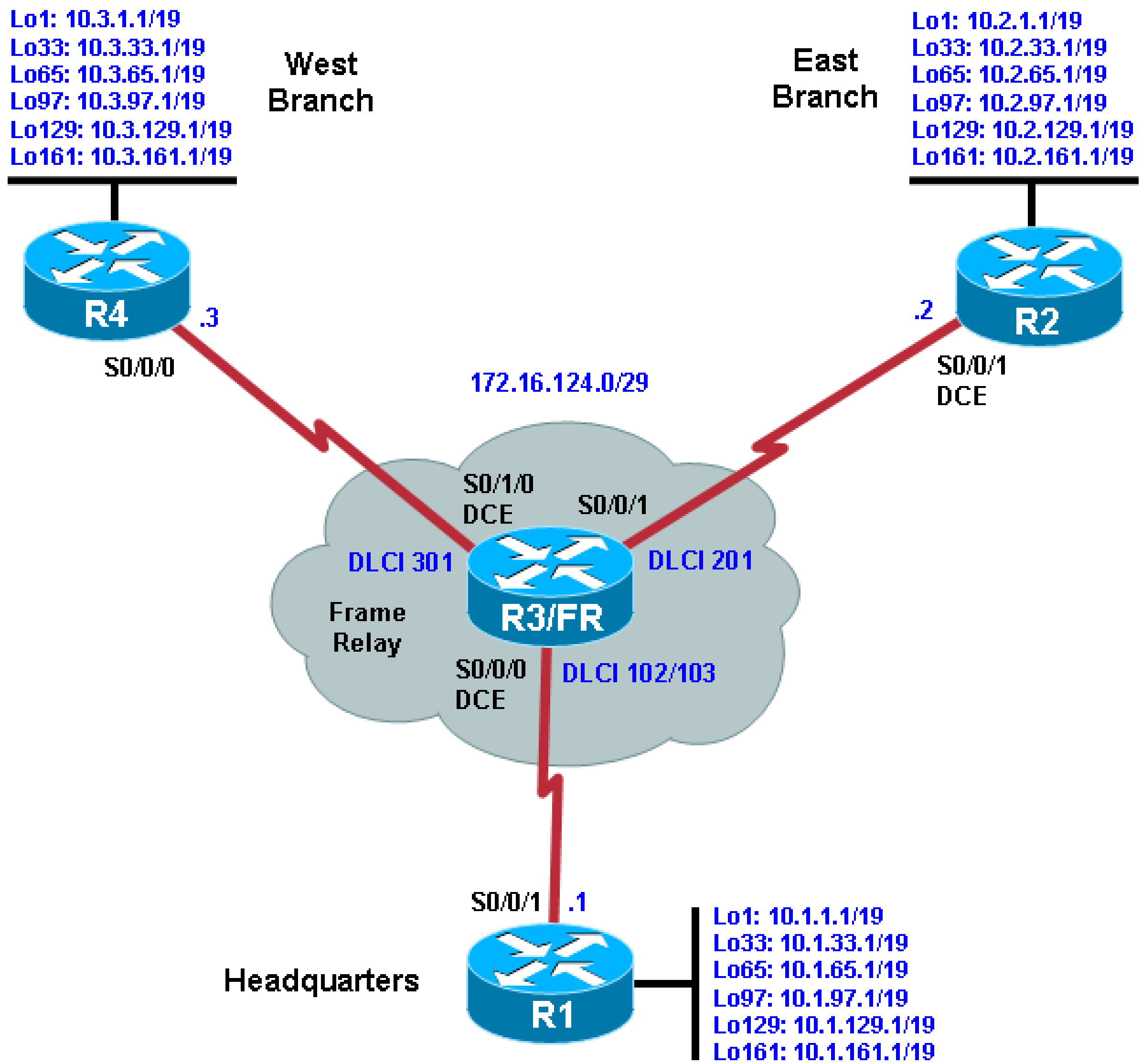
You should not observe this behavior with the 10.0.0.0/8 network because the R1 routing table has had subnets installed with VLSMs within that major network. Because R1 cannot generalize its destination prefixes for the 10.0.0.0/8 network, it forces the subnet into VLSM mode and shows it as “variably subnetted.”

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 2 Lab 2-4, EIGRP Frame Relay Hub-and-Spoke: Router Used as a Frame Relay Switch

Topology



Objectives

- Review a basic configuration of EIGRP on a serial interface.
- Configure EIGRP over Frame Relay hub-and-spoke.
- Configure a router as a Frame Relay switch.
- Configure the **ip bandwidth-percent** command.
- Disable split horizon.

- Use EIGRP in non-broadcast mode.
- Enable EIGRP manual summarization in topologies with discontiguous major networks.

Background

You are responsible for configuring and testing the new network that connects your company's headquarters (HQ) and EAST and WEST branches. The three locations are connected over hub-and-spoke Frame Relay, using the company headquarters as the hub. In this lab, you model each branch office's network with multiple loopback interfaces on each router and configure EIGRP to allow full connectivity between all departments.

To simulate the Frame Relay WAN connections, use a router with three serial ports to act as a Frame Relay switch. The configuration of the router as a Frame Relay switch is described in Step 2.

When accessing a Frame Relay service, a modem or a CSU/DSU is used at the customer premises to connect the router to the local loop and over the local loop to the Frame Relay switch. The modem or CSU/DSU provides the Layer 1 DCE (clocking) functions to the router. However, in this lab, R2 interface serial 0/0/1 is configured as the DCE for compatibility with other labs. If you are uncertain which side of the connection is the DCE, use the **show controllers serial** *interface-number* command:

```
FRS# show controllers serial0/0/0
Interface Serial0/0/0
Hardware is GT96K
DCE V.35, clock rate 64000
```

Note: In this lab, Router R3 acts as the Frame Relay switch and requires two serial interface cards. If you are using an Adtran as a Frame Relay switch, see Appendix A for the Adtran configuration. When using the Adtran, the clock (DCE) is provided for each serial link.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- 1 router acting as a Frame Relay switch (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Configure loopback addressing.

Using the addressing scheme in the diagram, apply IP addresses to the loopback interfaces on HQ, EAST, and WEST. You can paste the following configurations into your routers to begin.

Router R1 (HQ)

```
hostname HQ
!
interface Loopback1
 ip address 10.1.1.1 255.255.224.0
interface Loopback33
 ip address 10.1.33.1 255.255.224.0
interface Loopback65
 ip address 10.1.65.1 255.255.224.0
interface Loopback97
```



```
 ip address 10.1.97.1 255.255.224.0
interface Loopback129
 ip address 10.1.129.1 255.255.224.0
interface Loopback161
 ip address 10.1.161.1 255.255.224.0
!
```

Router R2 (EAST)

```
hostname EAST
!
interface Loopback1
 ip address 10.2.1.1 255.255.224.0
interface Loopback33
 ip address 10.2.33.1 255.255.224.0
interface Loopback65
 ip address 10.2.65.1 255.255.224.0
interface Loopback97
 ip address 10.2.97.1 255.255.224.0
interface Loopback129
 ip address 10.2.129.1 255.255.224.0
interface Loopback161
 ip address 10.2.161.1 255.255.224.0
!
end
```

Router R4 (WEST)

```
hostname WEST
!
interface Loopback1
 ip address 10.3.1.1 255.255.224.0
interface Loopback33
 ip address 10.3.33.1 255.255.224.0
interface Loopback65
 ip address 10.3.65.1 255.255.224.0
interface Loopback97
 ip address 10.3.97.1 255.255.224.0
interface Loopback129
 ip address 10.3.129.1 255.255.224.0
interface Loopback161
 ip address 10.3.161.1 255.255.224.0
!
end
```

Step 2: Configure the Frame Relay switch.

Use a fourth Cisco router with three serial interfaces as a Frame Relay switch, and cable the routers according to the diagram. Paste the following configuration into the router that is simulating the Frame Relay switch.

Note: If you are using an Adtran as a Frame Relay switch, see Appendix A for the Adtran configuration and which cables to use between the routers and the Adtran.

Router R3 (FRS)

```
hostname FRS
!
frame-relay switching
!
```

```
interface Serial0/0/0
 description FR to HQ
 no ip address
 encapsulation frame-relay ietf
 clock rate 128000
 frame-relay lmi-type cisco
 frame-relay intf-type dce
 frame-relay route 102 interface Serial0/0/1 201
 frame-relay route 103 interface Serial0/1/0 301
 no shutdown
!
interface Serial0/0/1
 description FR to EAST
 no ip address
 encapsulation frame-relay ietf
 frame-relay lmi-type cisco
 frame-relay intf-type dce
 frame-relay route 201 interface Serial0/0/0 102
 no shutdown
!
interface Serial0/1/0
 description FR to WEST
 no ip address
 encapsulation frame-relay ietf
 clock rate 64000
 frame-relay lmi-type cisco
 frame-relay intf-type dce
 frame-relay route 301 interface Serial0/0/0 103
 no shutdown
!
End
```

Note: You do not need to configure the LMI type as **cisco** because it is the default. In addition, the HQ, EAST, and WEST routers are able to automatically determine the LMI type. However, you could configure the Frame Relay switch with a different LMI type on each of its interfaces to demonstrate support for the existing types (cisco, ansi, q933a) and to show that they interoperate cleanly without requiring any particular configuration on the Frame Relay switch or end routers. Additionally, the **ietf** keyword is meaningful only on Frame Relay end devices, not on Frame Relay switches. It is configured on R3 here for clarity.

Step 3: Configure the Frame Relay endpoints.

You will be configuring HQ to be the Frame Relay hub, with EAST and WEST as the spokes. Check the topology diagram for the data-link connection identifiers (DLCIs) to use in the Frame Relay maps. You will also be turning off Frame Relay Inverse Address Resolution Protocol (InARP) for all interfaces. Inverse ARP allows a Frame Relay network to discover the IP address associated with the virtual circuit. This is sometimes a desirable trait in a production network. However, in the lab, Inverse ARP is turned off because you are using static maps.

a. Assign the Frame Relay interface of each router an IP address in the Frame Relay subnet 172.16.124.0 /29, as indicated in the topology diagram.

b. Enable Frame Relay encapsulation, disable Frame Relay Inverse ARP, and map the other IPs in the subnet to DLCIs using the **frame-relay map ip** *address dlci* **broadcast** command. The **broadcast** keyword is important because without it, no broadcast or multicast packets including EIGRP messages are sent through the Frame Relay cloud. Bring up the interfaces with the **no shutdown** command.

Note: It is unnecessary and actually undesirable to specify the mappings from EAST to WEST and from WEST to EAST with the **broadcast** option. Specifying this option duplicates all broadcasts and multicasts sent from EAST or WEST routers. Because of the hub-and-spoke topology, HQ will also receive all broadcasts and multicasts twice. Specify the **broadcast** option if a direct PVC is going toward the mapped IP, as is the case between EAST and HQ and between WEST and HQ.

Note: It is good practice to specify the IETF encapsulation format, which is an open format. Using the IETF frame format helps ensure interoperability between different vendors.

```
HQ# conf t
HQ(config)# interface serial 0/0/1
HQ(config-if)# ip address 172.16.124.1 255.255.255.248
HQ(config-if)# encapsulation frame-relay ietf
HQ(config-if)# no frame-relay inverse-arp
HQ(config-if)# frame-relay map ip 172.16.124.2 102 broadcast
HQ(config-if)# frame-relay map ip 172.16.124.3 103 broadcast
HQ(config-if)# no shutdown

EAST# conf t
EAST(config)# interface serial 0/0/1
EAST(config-if)# ip address 172.16.124.2 255.255.255.248
EAST(config-if)# clock rate 64000
EAST(config-if)# encapsulation frame-relay ietf
EAST(config-if)# no frame-relay inverse-arp
EAST(config-if)# frame-relay map ip 172.16.124.1 201 broadcast
EAST(config-if)# frame-relay map ip 172.16.124.3 201
EAST(config-if)# no shutdown

WEST# conf t
WEST(config)# interface serial 0/0/0
WEST(config-if)# ip address 172.16.124.3 255.255.255.248
WEST(config-if)# encapsulation frame-relay ietf
WEST(config-if)# no frame-relay inverse-arp
WEST(config-if)# frame-relay map ip 172.16.124.1 301 broadcast
WEST(config-if)# frame-relay map ip 172.16.124.2 301
WEST(config-if)# no shutdown
```

You will configure the bandwidth for the serial links in Step 4.

c. Verify that you have connectivity across the Frame Relay network by pinging the remote routers from each of the Frame Relay endpoints.

```
HQ# ping 172.16.124.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.124.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

The only interface that the Frame Relay interface is unable to communicate with is itself. This is not a significant problem in Frame Relay networks. To allow the Frame Relay interface to ping itself, you can map the local IP address to be forwarded out a PVC. The remote router at the other end of the PVC can then forward the IP address back based on its Frame Relay map statements. This solution is so that the Tcl scripts used for testing return successful echo replies under all circumstances. You do not need the **broadcast** keyword on this DLCI, because it is not important to forward broadcast and multicast packets (such as EIGRP hellos) to your own interface.

d. Configure the local mappings.

```
HQ(config)# interface serial 0/0/1
```



```
HQ(config-if)# frame-relay map ip 172.16.124.1 102

EAST(config)# interface serial 0/0/1
EAST(config-if)# frame-relay map ip 172.16.124.2 201

WEST(config)# interface serial 0/0/0
WEST(config-if)# frame-relay map ip 172.16.124.3 301
```

HQ now forwards packets destined for its own serial interface (172.16.124.1) to the EAST serial interface (172.16.124.2) and then back. This allows HQ to ping its own serial interface in the lab Frame Relay network. In a production network in which a company is billed based on per-PVC usage, this is not a preferred configuration. However, in this lab network, it helps ensure full ICMP connectivity in the Tcl scripts.

Step 4: Set interface-level bandwidth.

a. On the three routers, set the Frame Relay serial interface bandwidth with the **bandwidth** *bandwidth* command in interface configuration mode. Specify the bandwidth in kilobits per second (kb/s). For HQ, use 128 kb/s. On EAST and WEST, use 64 kb/s.

Recall from Lab 2-1 that, by default, EIGRP limits its bandwidth usage to 50 percent of the value specified by the **bandwidth** parameter. The default bandwidth for a serial interface is 1544 kb/s. This means that each neighbor for which this is an outbound interface has a traffic limit of a fraction of that 50 percent, represented by 1/*N*, where *N* is the number of neighbors out that interface.

```
HQ(config)# interface serial 0/0/1
HQ(config-if)# bandwidth 128

EAST(config)# interface serial 0/0/1
EAST(config-if)# bandwidth 64

WEST(config)# interface serial 0/0/0
WEST(config-if)# bandwidth 64
```

The HQ serial interface divides its total EIGRP bandwidth into fractional amounts according to the number of neighbors out that interface.

How much bandwidth on HQ serial 0/0/1 is reserved for EIGRP traffic to EAST?

____

____

b. You can control both the bandwidth and the EIGRP bandwidth percentage on a per-interface basis. On HQ, limit the bandwidth used by EIGRP to 40 percent without changing the **bandwidth** parameter on the interface using the interface-level command **ip bandwidth-percent eigrp** *as_number percent*.

```
HQ(config-if)# ip bandwidth-percent eigrp 1 40
```

Step 5: Configure EIGRP routing.

a. Configure EIGRP AS 1 on HQ, EAST, and WEST.

The network represented in the diagram is a discontiguous network (10.0.0.0/8) configured on all three routers. If you enabled auto-summarization, HQ sends and receives summaries for 10.0.0.0/8 with both EAST and WEST. Auto-summarization causes considerable routing disruption in this network, because HQ does not know which of the two spokes is the correct destination for subnets of 10.0.0.0/8. For this reason, turn off auto-summarization on each router.

b. Add your network statements to EIGRP on all three routers. The two major networks being used are 10.0.0.0 for the loopbacks and 172.16.0.0 for the Frame Relay cloud.

```
HQ(config)# router eigrp 1
HQ(config-router)# network 10.0.0.0
HQ(config-router)# network 172.16.0.0
HQ(config-router)# no auto-summary

EAST(config)# router eigrp 1
EAST(config-router)# network 10.0.0.0
EAST(config-router)# network 172.16.0.0
EAST(config-router)# no auto-summary

WEST(config)# router eigrp 1
WEST(config-router)# network 10.0.0.0
WEST(config-router)# network 172.16.0.0
WEST(config-router)# no auto-summary
```

c. Issue the **show ip eigrp topology** command on EAST.

```
EAST# show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(172.16.124.2)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.2.0.0/19, 1 successors, FD is 128256
        via Connected, Loopback1
P 10.1.0.0/19, 1 successors, FD is 40640000
        via 172.16.124.1 (40640000/128256), Serial0/0/1
P 10.2.32.0/19, 1 successors, FD is 128256
        via Connected, Loopback33
P 10.1.32.0/19, 1 successors, FD is 40640000
        via 172.16.124.1 (40640000/128256), Serial0/0/1
P 10.2.64.0/19, 1 successors, FD is 128256
        via Connected, Loopback65
P 10.1.64.0/19, 1 successors, FD is 40640000
        via 172.16.124.1 (40640000/128256), Serial0/0/1
P 10.2.96.0/19, 1 successors, FD is 128256
        via Connected, Loopback97
P 10.1.96.0/19, 1 successors, FD is 40640000
        via 172.16.124.1 (40640000/128256), Serial0/0/1
P 10.2.128.0/19, 1 successors, FD is 128256
        via Connected, Loopback129
P 10.1.128.0/19, 1 successors, FD is 40640000
        via 172.16.124.1 (40640000/128256), Serial0/0/1
P 10.2.160.0/19, 1 successors, FD is 128256
        via Connected, Loopback161
P 10.1.160.0/19, 1 successors, FD is 40640000
        via 172.16.124.1 (40640000/128256), Serial0/0/1
P 172.16.124.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/1
EAST#
```


Which networks are missing from the topology database?

____

____

____

What do you think is responsible for this problem?

____

____

____

d. HQ needs the **no ip split-horizon eigrp** *as_number* command on its serial Frame Relay interface. This command disables split horizon for an EIGRP autonomous system. If split horizon is enabled (the default), route advertisements from EAST to HQ do not travel to WEST and vice versa, as shown in the above output.

```
HQ(config)# interface serial 0/0/1
HQ(config-if)# no ip split-horizon eigrp 1
```

e. Verify that you see the correct EIGRP adjacencies with the **show ip eigrp neighbors** command.

```
HQ# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address             Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
1   172.16.124.2        Se0/0/1          176 00:00:05 1588  5000  0  6
0   172.16.124.3        Se0/0/1          176 00:00:05   23  1140  0  6

EAST# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address             Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
0   172.16.124.1        Se0/0/1          129 00:00:52   20  2280  0  20

WEST# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address             Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                        (sec)         (ms)       Cnt Num
0   172.16.124.1        Se0/0/0          176 00:00:55   20  2280  0  13
```

f. Verify that you have IP routes on all three routers for the entire topology with the **show ip route** command.

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

    172.16.0.0/29 is subnetted, 1 subnets
C      172.16.124.0 is directly connected, Serial0/0/1
    10.0.0.0/19 is subnetted, 18 subnets
D      10.2.0.0 [90/20640000] via 172.16.124.2, 00:04:36, Serial0/0/1
D      10.3.0.0 [90/20640000] via 172.16.124.3, 00:04:20, Serial0/0/1
C      10.1.0.0 is directly connected, Loopback1
D      10.2.32.0 [90/20640000] via 172.16.124.2, 00:04:36, Serial0/0/1
D      10.3.32.0 [90/20640000] via 172.16.124.3, 00:04:20, Serial0/0/1
C      10.1.32.0 is directly connected, Loopback33
D      10.2.64.0 [90/20640000] via 172.16.124.2, 00:04:37, Serial0/0/1
```



```
D       10.3.64.0 [90/20640000] via 172.16.124.3, 00:04:21, Serial0/0/1
C       10.1.64.0 is directly connected, Loopback65
D       10.2.96.0 [90/20640000] via 172.16.124.2, 00:04:37, Serial0/0/1
D       10.3.96.0 [90/20640000] via 172.16.124.3, 00:04:21, Serial0/0/1
C       10.1.96.0 is directly connected, Loopback97
D       10.2.128.0 [90/20640000] via 172.16.124.2, 00:04:37, Serial0/0/1
D       10.3.128.0 [90/20640000] via 172.16.124.3, 00:04:21, Serial0/0/1
C       10.1.128.0 is directly connected, Loopback129
D       10.2.160.0 [90/20640000] via 172.16.124.2, 00:04:37, Serial0/0/1
D       10.3.160.0 [90/20640000] via 172.16.124.3, 00:04:21, Serial0/0/1
C       10.1.160.0 is directly connected, Loopback161

EAST# show ip route
<output omitted>

    172.16.0.0/29 is subnetted, 1 subnets
C       172.16.124.0 is directly connected, Serial0/0/1
    10.0.0.0/19 is subnetted, 18 subnets
C       10.2.0.0 is directly connected, Loopback1
D       10.3.0.0 [90/41152000] via 172.16.124.1, 00:01:31, Serial0/0/1
D       10.1.0.0 [90/40640000] via 172.16.124.1, 00:07:12, Serial0/0/1
C       10.2.32.0 is directly connected, Loopback33
D       10.3.32.0 [90/41152000] via 172.16.124.1, 00:01:31, Serial0/0/1
D       10.1.32.0 [90/40640000] via 172.16.124.1, 00:07:13, Serial0/0/1
C       10.2.64.0 is directly connected, Loopback65
D       10.3.64.0 [90/41152000] via 172.16.124.1, 00:01:32, Serial0/0/1
D       10.1.64.0 [90/40640000] via 172.16.124.1, 00:07:13, Serial0/0/1
C       10.2.96.0 is directly connected, Loopback97
D       10.3.96.0 [90/41152000] via 172.16.124.1, 00:01:32, Serial0/0/1
D       10.1.96.0 [90/40640000] via 172.16.124.1, 00:07:13, Serial0/0/1
C       10.2.128.0 is directly connected, Loopback129
D       10.3.128.0 [90/41152000] via 172.16.124.1, 00:01:32, Serial0/0/1
D       10.1.128.0 [90/40640000] via 172.16.124.1, 00:07:13, Serial0/0/1
C       10.2.160.0 is directly connected, Loopback161
D       10.3.160.0 [90/41152000] via 172.16.124.1, 00:01:32, Serial0/0/1
D       10.1.160.0 [90/40640000] via 172.16.124.1, 00:07:13, Serial0/0/1

WEST# show ip route
<output omitted>

    172.16.0.0/29 is subnetted, 1 subnets
C       172.16.124.0 is directly connected, Serial0/0/0
    10.0.0.0/19 is subnetted, 18 subnets
D       10.2.0.0 [90/41152000] via 172.16.124.1, 00:02:00, Serial0/0/0
C       10.3.0.0 is directly connected, Loopback1
D       10.1.0.0 [90/40640000] via 172.16.124.1, 00:07:41, Serial0/0/0
D       10.2.32.0 [90/41152000] via 172.16.124.1, 00:02:00, Serial0/0/0
C       10.3.32.0 is directly connected, Loopback33
D       10.1.32.0 [90/40640000] via 172.16.124.1, 00:07:43, Serial0/0/0
D       10.2.64.0 [90/41152000] via 172.16.124.1, 00:02:01, Serial0/0/0
C       10.3.64.0 is directly connected, Loopback65
D       10.1.64.0 [90/40640000] via 172.16.124.1, 00:07:43, Serial0/0/0
D       10.2.96.0 [90/41152000] via 172.16.124.1, 00:02:01, Serial0/0/0
C       10.3.96.0 is directly connected, Loopback97
D       10.1.96.0 [90/40640000] via 172.16.124.1, 00:07:43, Serial0/0/0
D       10.2.128.0 [90/41152000] via 172.16.124.1, 00:02:01, Serial0/0/0
C       10.3.128.0 is directly connected, Loopback129
D       10.1.128.0 [90/40640000] via 172.16.124.1, 00:07:43, Serial0/0/0
```



```
D       10.2.160.0 [90/41152000] via 172.16.124.1, 00:02:01, Serial0/0/0
C       10.3.160.0 is directly connected, Loopback161
D       10.1.160.0 [90/40640000] via 172.16.124.1, 00:07:43, Serial0/0/0
```

g. Run the following Tcl script on all routers to verify full connectivity.

```
HQ# tclsh

foreach address {
10.1.1.1
10.1.33.1
10.1.65.1
10.1.97.1
10.1.129.1
10.1.161.1
172.16.124.1
10.2.1.1
10.2.33.1
10.2.65.1
10.2.97.1
10.2.129.1
10.2.161.1
172.16.124.2
10.3.1.1
10.3.33.1
10.3.65.1
10.3.97.1
10.3.129.1
10.3.161.1
172.16.124.3
} { ping $address }
```

You should get ICMP echo replies for every address pinged.

Step 6: Configure non-broadcast EIGRP mode.

Currently, you are using EIGRP in its default mode, which multicasts packets to the link-local address 224.0.0.10. However, not all Frame Relay configurations support multicast. EIGRP supports unicasts to remote destinations using non-broadcast mode on a per-interface basis. This mode is analogous to configuring RIPv2 with a passive interface and statically configuring neighbors out that interface.

a. Implement packet unicasts to neighbors.

```
HQ(config)# router eigrp 1
HQ(config-router)# neighbor 172.16.124.2 serial 0/0/1
HQ(config-router)# neighbor 172.16.124.3 serial 0/0/1

EAST(config)# router eigrp 1
EAST(config-router)# neighbor 172.16.124.1 serial 0/0/1

WEST(config)# router eigrp 1
WEST(config-router)# neighbor 172.16.124.1 serial 0/0/0
```

b. HQ now has two neighbor statements, and the other two routers have one. After you configure neighbor statements for an interface, EIGRP automatically stops multicasting packets out that interface and starts unicasting packets instead. Use the **show ip eigrp neighbors** command to verify the changes.

```
HQ# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address            Interface       Hold Uptime   SRTT   RTO  Q  Seq
```



```
                                          (sec)         (ms)       Cnt Num
1   172.16.124.2        Se0/0/1            153 00:00:28   65   390  0  9
0   172.16.124.3        Se0/0/1            158 00:00:28 1295  5000  0  9

EAST# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address             Interface         Hold Uptime   SRTT   RTO  Q  Seq
                                          (sec)         (ms)       Cnt Num
0   172.16.124.1        Se0/0/1            146 00:02:19   93   558  0  15

WEST# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address             Interface         Hold Uptime   SRTT   RTO  Q  Seq
                                          (sec)         (ms)       Cnt Num
0   172.16.124.1        Se0/0/0            160 00:03:00   59   354  0  15
```

Step 7: Implement EIGRP manual summarization.

a. Implement EIGRP manual summarization on each router. Each router should advertise only one network summarizing all of its loopbacks. Using the commands you learned in EIGRP Lab 2-3, configure the summary address on the serial interfaces.

   What is the length of the network mask that is used to summarize all the loopbacks on each router?

   ____

b. Look at the simplified EIGRP topology table on each router using the **show ip eigrp topology** command.

```
HQ# show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.161.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.2.0.0/16, 1 successors, FD is 20640000
        via 172.16.124.2 (20640000/128256), Serial0/0/1
P 10.3.0.0/16, 1 successors, FD is 20640000
        via 172.16.124.3 (20640000/128256), Serial0/0/1
P 10.1.0.0/16, 1 successors, FD is 128256
        via Summary (128256/0), Null0
P 10.1.0.0/19, 1 successors, FD is 128256
        via Connected, Loopback1
P 10.1.32.0/19, 1 successors, FD is 128256
        via Connected, Loopback33
P 10.1.64.0/19, 1 successors, FD is 128256
        via Connected, Loopback65
P 10.1.96.0/19, 1 successors, FD is 128256
        via Connected, Loopback97
P 10.1.128.0/19, 1 successors, FD is 128256
        via Connected, Loopback129
P 10.1.160.0/19, 1 successors, FD is 128256
        via Connected, Loopback161

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 172.16.124.0/29, 1 successors, FD is 20512000
        via Connected, Serial0/0/1
```



```
EAST# show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.2.161.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.2.0.0/16, 1 successors, FD is 128256
        via Summary (128256/0), Null0
P 10.2.0.0/19, 1 successors, FD is 128256
        via Connected, Loopback1
P 10.3.0.0/16, 1 successors, FD is 41152000
        via 172.16.124.1 (41152000/20640000), Serial0/0/1
P 10.1.0.0/16, 1 successors, FD is 40640000
        via 172.16.124.1 (40640000/128256), Serial0/0/1
P 10.2.32.0/19, 1 successors, FD is 128256
        via Connected, Loopback33
P 10.2.64.0/19, 1 successors, FD is 128256
        via Connected, Loopback65
P 10.2.96.0/19, 1 successors, FD is 128256
        via Connected, Loopback97
P 10.2.128.0/19, 1 successors, FD is 128256
        via Connected, Loopback129
P 10.2.160.0/19, 1 successors, FD is 128256
        via Connected, Loopback161
P 172.16.124.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/1
        via 172.16.124.1 (41024000/20512000), Serial0/0/1

WEST# show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(172.16.124.3)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.2.0.0/16, 1 successors, FD is 41152000
        via 172.16.124.1 (41152000/20640000), Serial0/0/0
P 10.3.0.0/16, 1 successors, FD is 128256
        via Summary (128256/0), Null0
P 10.3.0.0/19, 1 successors, FD is 128256
        via Connected, Loopback1
P 10.1.0.0/16, 1 successors, FD is 40640000
        via 172.16.124.1 (40640000/128256), Serial0/0/0
P 10.3.32.0/19, 1 successors, FD is 128256
        via Connected, Loopback33
P 10.3.64.0/19, 1 successors, FD is 128256
        via Connected, Loopback65
P 10.3.96.0/19, 1 successors, FD is 128256
        via Connected, Loopback97
P 10.3.128.0/19, 1 successors, FD is 128256
        via Connected, Loopback129
P 10.3.160.0/19, 1 successors, FD is 128256
        via Connected, Loopback161
P 172.16.124.0/29, 1 successors, FD is 40512000
        via Connected, Serial0/0/0
        via 172.16.124.1 (41024000/20512000), Serial0/0/0
```


Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Appendix A: Adtran Frame Relay Switch Configuration

If an Adtran Atlas 550 is used for the Frame Relay switch, connect the serial cable from each router interface in the topology diagram to the Adtran interface indicated in the table below. The Adtran is preconfigured to simulate a Frame Relay service that provides the following PVCs.

Connected Router	Router Interface	Adtran Interface	Ingress DLCI	Egress DLCI	Egress Router
HQ	S0/0/1 DTE	port 1/1	102	201	East
HQ	S0/0/1 DTE	port 1/1	103	301	West
East	S0/0/1 DTE	port 1/2	201	102	HQ
West	S0/0/0 DTE	port 2/1	301	103	HQ

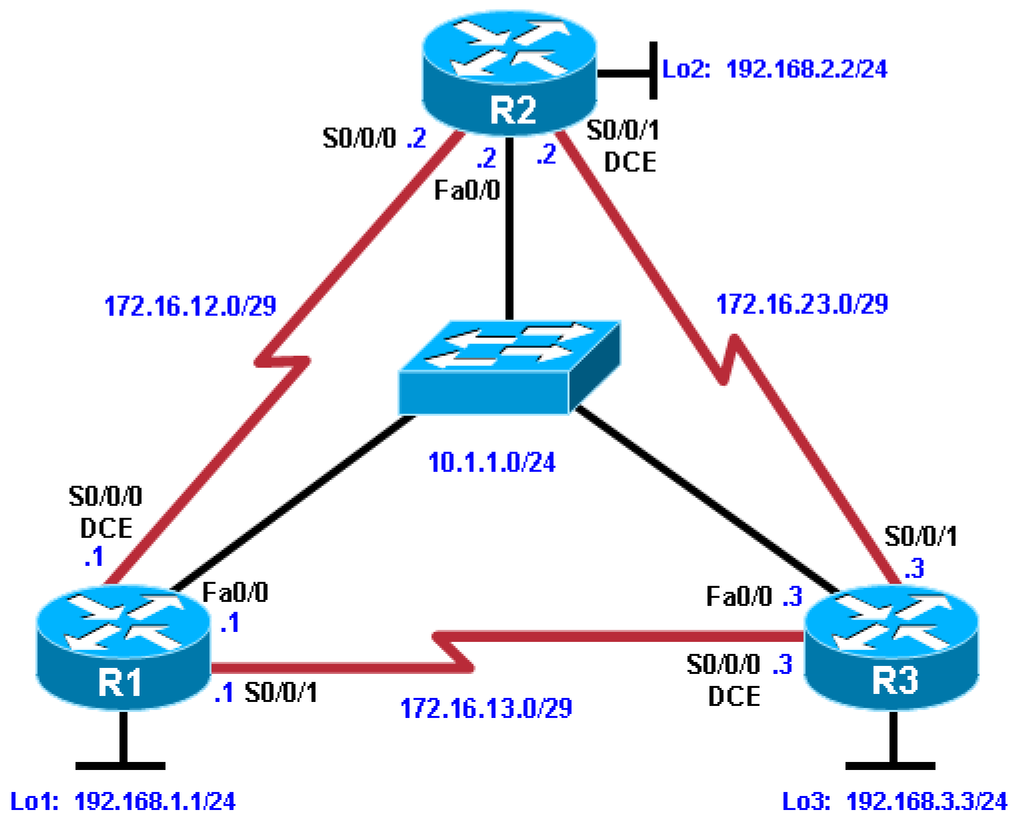
Frame Relay Switching Configuration

The Adtran Frame Relay switch interfaces all provide the DCE clock. Be sure to use the appropriate cable between each router and the Adtran. All the router interfaces are DTE, and the cable to the Adtran interface should be serial to V.35 DCE. Use the **show controllers** command to verify which cable type is connected to a given router interface.

```
HQ# show controllers s0/0/1
Interface Serial0/0/1
Hardware is GT96K
DTE V.35 TX and RX clocks detected.

<output omitted>
```


Chapter 2 Lab 2-5, EIGRP Authentication and Timers

Topology



Objectives

- Review a basic configuration of EIGRP.
- Configure and verify EIGRP authentication parameters.
- Configure EIGRP hello interval and hold time.
- Verify the hello interval and hold time.

Background

As a network engineer, you have weighed the benefits of routing protocols and deployed EIGRP in your corporation's network. Recently, a new Chief Information Officer replaced the previous CIO and outlined a new network policy detailing more robust security measures. The CIO has also drawn up specifications to allow more frequent checking between neighboring routers so that fewer packets are lost in transit during times of instability. In this lab, you implement the CIO's specifications on the network.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the advanced IP services image c1841-advipservicesk9-mz.124-24.T1.bin. The switch is a Cisco WS-C2960-24TT-L with the Cisco IOS image c2960-lanbasek9-mz.122-46.SE.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router or switch

model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- 1 switch (Cisco 2960 with the Cisco IOS Release 12.2(46)SE C2960-LANBASEK9-M image or comparable)
- Serial and Ethernet cables

Step 1: Configure the hostname and interface addresses.

Using the addressing scheme in the diagram, apply IP addresses to the loopback, serial, and Fast Ethernet interfaces on R1, R2, and R3. Set the serial interface bandwidth on each router with the interface-level **bandwidth** *bandwidth* command. Specify the bandwidth as 64 kb/s on each serial interface. Specify the clock rate on the DCE end of each serial link using the **clock rate 64000** command.

Note: If you have WIC-2A/S serial interfaces, the maximum clock rate is 128 kb/s. If you have WIC-2T serial interfaces, the maximum clock rate is much higher (2.048 Mb/s or higher depending on hardware), which is more representative of a modern network WAN link. However, this lab uses 64 kb/s and 128 kb/s settings.

You can copy and paste the following configurations into your routers to begin.

Note: Depending on the router model, the interfaces might be numbered differently than those listed and might require you to alter the interface designation accordingly.

Router R1

```
hostname R1
!
interface Loopback1
 ip address 192.168.1.1 255.255.255.0
!
interface FastEthernet0/0
 ip address 10.1.1.1 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 ip address 172.16.12.1 255.255.255.248
 clock rate 64000
 bandwidth 64
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.13.1 255.255.255.248
 bandwidth 64
 no shutdown
!
end
```

Router R2

```
hostname R2
!
interface Loopback2
 ip address 192.168.2.2 255.255.255.0
!
interface FastEthernet0/0
```



```
 ip address 10.1.1.2 255.255.255.0
 no shutdown
!
interface Serial0/0/0
 ip address 172.16.12.2 255.255.255.248
 bandwidth 64
 no shutdown
!
interface Serial0/0/1
ip address 172.16.23.2 255.255.255.248
 clock rate 64000
 bandwidth 64
 no shutdown
!
end
```

Router R3

```
hostname R3
!
interface Loopback3
 ip address 192.168.3.3 255.255.255.0
!
interface FastEthernet0/0
 ip address 10.1.1.3 255.255.255.0
 no shutdown
!
interface Serial0/0/0
ip address 172.16.13.3 255.255.255.248
 clock rate 64000
 bandwidth 64
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.23.3 255.255.255.248
 bandwidth 64
 no shutdown
!
end
```

Step 2: Configure basic EIGRP.

a. Configure EIGRP AS 1 as in the previous EIGRP labs. Run EIGRP on all connections in the lab, and leave auto-summarization on. Advertise networks 10.0.0.0/8, 172.16.0.0/16, 192.168.1.0/24, 192.168.2.0/24, and 192.168.3.0/24 from their respective routers.

b. Use the **show ip eigrp neighbors** command to check which routers have EIGRP adjacencies.

```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
3   10.1.1.2                Fa0/0             11 00:00:54    4   200  0  36
2   10.1.1.3                Fa0/0             11 00:00:54   13   200  0  39
1   172.16.12.2             Se0/0/0           14 00:14:18   27  2280  0  32
0   172.16.13.3             Se0/0/1           13 00:14:23   25  2280  0  37

R2# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
```



```
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
3   10.1.1.1                Fa0/0             10 00:02:05 1020  5000  0  35
2   10.1.1.3                Fa0/0             14 00:02:05   11   200  0  39
1   172.16.12.1             Se0/0/0           14 00:15:25  106  2280  0  32
0   172.16.23.3             Se0/0/1           13 00:16:59    1  2280  0  38

R3# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
3   10.1.1.1                Fa0/0             12 00:03:18  816  4896  0  34
2   10.1.1.2                Fa0/0             11 00:03:18  822  4932  0  35
1   172.16.13.1             Se0/0/0           14 00:16:47   22  2280  0  31
0   172.16.23.2             Se0/0/1           14 00:18:12    4  2280  0  33
```

Did you receive the output that you expected?

____

c. Run the following Tcl script on all routers to verify full connectivity.

```
R1# tclsh

foreach address {
10.1.1.1
172.16.12.1
172.16.13.1
192.168.1.1
10.1.1.2
172.16.12.2
172.16.23.2
192.168.2.2
10.1.1.3
172.16.13.3
172.16.23.3
192.168.3.3
} { ping $address }
```

You should get ICMP echo replies for every address pinged.

Step 3: Configure authentication keys.

Before you configure a link to authenticate the EIGRP adjacencies, you must configure the keys that are used for the authentication. EIGRP uses Cisco IOS generic router key chains as storage locations for keys. These key chains classify keys into groups, enabling keys to be easily changed periodically without bringing down adjacencies.

a. Use the **key chain** *name* command in global configuration mode to create a chain of keys with the label EIGRP-KEYS.

```
R1# conf t
R1(config)# key chain EIGRP-KEYS
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string cisco

R2# conf t
R2(config)# key chain EIGRP-KEYS
```

```
R2(config-keychain)# key 1
R2(config-keychain-key)# key-string cisco

R3# conf t
R3(config)# key chain EIGRP-KEYS
R3(config-keychain)# key 1
R3(config-keychain-key)# key-string cisco
```

b. Issue the **show key chain** command. You should have the same output on every router.

```
R1# show key chain
Key-chain EIGRP-KEYS:
    key 1 -- text "cisco"
        accept lifetime (always valid) - (always valid) [valid now]
        send lifetime (always valid) - (always valid) [valid now]
```

You can set a time span for sending a key to other routers and during which a key is accepted from other routers. Although lifetime values are not explored in the route labs, you should keep it in mind for production networks when you are rolling from one set of authentication strings to another. For now, you simply want to authenticate the EIGRP adjacencies for security reasons.

Step 4: Configure EIGRP link authentication.

When configuring EIGRP link authentication, you must first associate the key chain with a particular EIGRP process (or autonomous system) running on the interface using the **ip authentication key-chain** *eigrp as_number* **key** *key_chain_label* command. Then you activate the MD5 authentication for that EIGRP process using the **ip authentication mode eigrp** *as_number* **md5** command.

a. Apply the following commands on all active EIGRP interfaces.

```
R1# conf t
R1(config)# interface serial 0/0/0
R1(config-if)# ip authentication key-chain eigrp 1 EIGRP-KEYS
R1(config-if)# ip authentication mode eigrp 1 md5
R1(config-if)# interface serial 0/0/1
R1(config-if)# ip authentication key-chain eigrp 1 EIGRP-KEYS
R1(config-if)# ip authentication mode eigrp 1 md5
R1(config-if)# interface fastethernet 0/0
R1(config-if)# ip authentication key-chain eigrp 1 EIGRP-KEYS
R1(config-if)# ip authentication mode eigrp 1 md5

R2# conf t
R2(config)# interface serial 0/0/0
R2(config-if)# ip authentication key-chain eigrp 1 EIGRP-KEYS
R2(config-if)# ip authentication mode eigrp 1 md5
R2(config-if)# interface serial 0/0/1
R2(config-if)# ip authentication key-chain eigrp 1 EIGRP-KEYS
R2(config-if)# ip authentication mode eigrp 1 md5
R2(config-if)# interface fastethernet 0/0
R2(config-if)# ip authentication key-chain eigrp 1 EIGRP-KEYS
R2(config-if)# ip authentication mode eigrp 1 md5

R3# conf t
R3(config)# interface serial 0/0/0
R3(config-if)# ip authentication key-chain eigrp 1 EIGRP-KEYS
R3(config-if)# ip authentication mode eigrp 1 md5
R3(config-if)# interface serial 0/0/1
R3(config-if)# ip authentication key-chain eigrp 1 EIGRP-KEYS
R3(config-if)# ip authentication mode eigrp 1 md5
```



```
R3(config-if)# interface fastethernet 0/0
R3(config-if)# ip authentication key-chain eigrp 1 EIGRP-KEYS
R3(config-if)# ip authentication mode eigrp 1 md5
```

Each EIGRP adjacency should flap (go down and come back up) when you implement MD5 authentication on one side of the link before the other side has been configured. In a production network, flapping causes some instability during a configuration, so make sure you implement MD5 outside of peak usage times.

b. Check the configuration with the **show ip eigrp interfaces detail** command.

```
R1# show ip eigrp interfaces detail
IP-EIGRP interfaces for process 1

                 Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Fa0/0       2        0/0         3       0/1           50           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/14  Un/reliable ucasts: 26/21
  Mcast exceptions: 3  CR packets: 3  ACKs suppressed: 3
  Retransmissions sent: 1  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use multicast
Se0/0/0     1        0/0         4       0/12          50           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 10/28
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 5
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use unicast
Se0/0/1     1        0/0         1       0/12          50           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 10/22
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 8
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use unicast
Lo1         0        0/0         0       0/1            0           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 0/0
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is not set
  Use multicast

R2# show ip eigrp interfaces detail
IP-EIGRP interfaces for process 1

                 Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Fa0/0       2        0/0         4       0/10          50           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/7  Un/reliable ucasts: 34/15
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 7
```



```
  Retransmissions sent: 1  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
Se0/0/0       1         0/0         1       0/12           50            0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 19/17
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 7
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
Se0/0/1       1         0/0         3       0/12           50            0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 11/9
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 4
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
Lo2           0         0/0         0       0/1             0            0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 0/0
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is not set
  Use multicast

R3# show ip eigrp interfaces detail
IP-EIGRP interfaces for process 1

                     Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface     Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Fa0/0           2        0/0         4       0/1           50            0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/3  Un/reliable ucasts: 6/7
  Mcast exceptions: 1  CR packets: 1  ACKs suppressed: 0
  Retransmissions sent: 2  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use multicast
Se0/0/0         1        0/0       482      10/380        2732           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 3/7
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 2
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use unicast
Se0/0/1         1        0/0       109      10/380         904           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 4/7
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 2
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use unicast
Lo3             0        0/0         0       0/1             0           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 0/0
```



```
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is not set
  Use multicast
```

At this point, the interfaces are authenticating each adjacency with the EIGRP-KEYS key chain. Make sure that you verify the number of neighbors out each interface in the above output. Notice that the number of peers is the number of adjacencies established out that interface.

When EIGRP has a key chain associated with an autonomous system on a given interface and EIGRP is authenticating its adjacencies, you have successfully completed the initial work.

c. Use the **debug eigrp packets** command to see the authenticated hellos.

```
R1# debug eigrp packets
EIGRP Packets debugging is on
    (UPDATE, REQUEST, QUERY, REPLY, HELLO, IPXSAP, PROBE, ACK, STUB,
SIAQUERY, SIAREPLY)
R1#
* Feb 9 19:10:51.090: EIGRP: Sending HELLO on Serial0/0/1
* Feb 9 19:10:51.090:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
* Feb 9 19:10:51.190: EIGRP: received packet with MD5 authentication, key id
= 1
* Feb 9 19:10:51.190: EIGRP: Received HELLO on Serial0/0/1 nbr 172.16.13.3
* Feb 9 19:10:51.190:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 0/0
* Feb 9 19:10:51.854: EIGRP: received packet with MD5 authentication, key id
= 1
* Feb 9 19:10:51.854: EIGRP: Received HELLO on FastEthernet0/0 nbr 10.1.1.2
* Feb 9 19:10:51.854:   AS 1, Flags 0x0, Seq 0/0 idbQ 0/0 iidbQ un/rely 0/0
peerQ un/rely 0/0
* Feb 9 19:10:53.046: EIGRP: received packet with MD5 authentication, key id
= 1

<output omitted>
```

d. Issue the **undebug all** command to stop the debugging output.

Step 5: Manipulate EIGRP timers.

The CIO also ordered you to change the hello and dead intervals on point-to-point serial interfaces so that dead neighbors are detected in roughly half the time that they are detected by default.

a. To view the default timers, use the **show ip eigrp interfaces detail** command.

```
R1# show ip eigrp interfaces detail
IP-EIGRP interfaces for process 1

                     Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface     Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Fa0/0           2       0/0          4       0/1           50           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/3  Un/reliable ucasts: 6/7
  Mcast exceptions: 1  CR packets: 1  ACKs suppressed: 0
  Retransmissions sent: 2  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use multicast
Se0/0/0         1        0/0        482      10/380       2732           0
  Hello interval is 5 sec
```



```
    Next xmit serial <none>
    Un/reliable mcasts: 0/0  Un/reliable ucasts: 3/7
    Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 2
    Retransmissions sent: 0  Out-of-sequence rcvd: 0
    Authentication mode is md5,  key-chain is "EIGRP-KEYS"
    Use unicast
Se0/0/1          1        0/0       109      10/380        904          0
  Hello interval is 5 sec
    Next xmit serial <none>
    Un/reliable mcasts: 0/0  Un/reliable ucasts: 4/7
    Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 2
    Retransmissions sent: 0  Out-of-sequence rcvd: 0
    Authentication mode is md5,  key-chain is "EIGRP-KEYS"
    Use unicast

<output omitted>
```

The default hello interval for point-to-point serial links is 5 seconds, regardless of the bandwidth, and 5 seconds for LAN interfaces. The default hold time is three times the length of the hello interval.

The hello interval determines how often *outgoing* EIGRP hellos are sent, while the hold time defines how long other neighbors tolerate the loss of the hello packets You are more concerned with the hold time than the hello interval, because the hold time detects a dead neighbor. However, you also want the neighbors to send the same number of hellos as under normal circumstances before declaring a neighbor dead.

The requirements from the CIO specify that the hold time should be roughly half of the default, which is 15 seconds, so a new hold time of 7 or 8 seconds would be appropriate. A shorter hold time allows a dead neighbor to be detected more quickly. A hello interval of 2 seconds results in detecting new neighbors more rapidly.

b. Change both the hello interval and the hold time for AS 1 for serial 0/0/0 on R1 and R2 using the **ip hello-interval eigrp 1 2** and **ip hold-time eigrp 1 8** commands. If necessary, use the **?** to investigate what each parameter does.

```
R1# conf t
R1(config)# interface serial 0/0/0
R1(config-if)# ip hello-interval eigrp 1 2
R1(config-if)# ip hold-time eigrp 1 8

R2# conf t
R2(config)# interface serial 0/0/0
R2(config-if)# ip hello-interval eigrp 1 2
R2(config-if)# ip hold-time eigrp 1 8
```

c. Verify that the hello interval has been successfully changed on routers R1 and R2 using the **show ip eigrp 1 interfaces detail serial 0/0/0** command.

```
R1# show ip eigrp 1 interfaces detail serial 0/0/0
IP-EIGRP interfaces for process 1

                        Xmit Queue   Mean   Pacing Time   Multicast
Pending
Interface      Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Se0/0/0          1        0/0       482      10/380       2732          0
  Hello interval is 2 sec
    Next xmit serial <none>
    Un/reliable mcasts: 0/0  Un/reliable ucasts: 3/7
    Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 2
```



```
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use unicast

R2# show ip eigrp 1 interfaces detail serial 0/0/0
IP-EIGRP interfaces for process 1
                        Xmit Queue   Mean   Pacing Time   Multicast
Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Se0/0/0            1        0/0       190      10/380        1300           0
  Hello interval is 2 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 4/5
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 2
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use unicast
```

d. Verify that the hold time has been successfully changed with the **show ip eigrp neighbors** command.

```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
3   10.1.1.2                Fa0/0             11 01:32:00    7   200  0  19
2   10.1.1.3                Fa0/0             12 01:32:03    1   200  0  18
1   172.16.12.2             Se0/0/0            7 01:32:27  482  2892  0  17
0   172.16.13.3             Se0/0/1           11 01:32:28  109  2280  0  19

R2# show ip eigrp neighbors
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
3   10.1.1.1                Fa0/0             14 01:30:33  816  4896  0  19
2   10.1.1.3                Fa0/0             12 01:30:33  819  4914  0  21
1   172.16.12.1             Se0/0/0            7 01:30:58  190  2280  0  21
0   172.16.23.3             Se0/0/1           13 01:30:59   80  2280  0  20
```

e. Configure the same hello interval and hold time on each active serial interface in the topology.

```
R1# conf t
R1(config)# interface serial 0/0/1
R1(config-if)# ip hello-interval eigrp 1 2
R1(config-if)# ip hold-time eigrp 1 8

R2# conf t
R2(config)# interface serial 0/0/1
R2(config-if)# ip hello-interval eigrp 1 2
R2(config-if)# ip hold-time eigrp 1 8

R3# conf t
R3(config)# interface serial 0/0/0
R3(config-if)# ip hello-interval eigrp 1 2
R3(config-if)# ip hold-time eigrp 1 8
R3(config-if)# interface serial 0/0/1
R3(config-if)# ip hello-interval eigrp 1 2
R3(config-if)# ip hold-time eigrp 1 8
```

f. Make sure that all of the EIGRP neighbor relationships remain up during the configuration process. Use the **show ip eigrp neighbors** command to verify the hold time, and the **show ip eigrp interfaces detail** command to verify the hello interval.


```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
3   10.1.1.2                Fa0/0             14 01:35:15    7   200  0  19
2   10.1.1.3                Fa0/0             12 01:35:18    1   200  0  18
1   172.16.12.2             Se0/0/0            7 01:35:43  482  2892  0  17
0   172.16.13.3             Se0/0/1            6 01:35:43  109  2280  0  19

R1# show ip eigrp interfaces detail
IP-EIGRP interfaces for process 1

                     Xmit Queue   Mean   Pacing Time   Multicast    Pending
Interface     Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Fa0/0           2        0/0        4        0/1           50           0
  Hello interval is 5 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/3  Un/reliable ucasts: 6/7
  Mcast exceptions: 1  CR packets: 1  ACKs suppressed: 0
  Retransmissions sent: 2  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use multicast
Se0/0/0         1        0/0      482      10/380       2732            0
  Hello interval is 2 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 3/7
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 2
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use unicast
Se0/0/1         1        0/0      109      10/380        904            0
  Hello interval is 2 sec
  Next xmit serial <none>
  Un/reliable mcasts: 0/0  Un/reliable ucasts: 4/7
  Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 2
  Retransmissions sent: 0  Out-of-sequence rcvd: 0
  Authentication mode is md5,  key-chain is "EIGRP-KEYS"
  Use unicast

<output omitted>
```

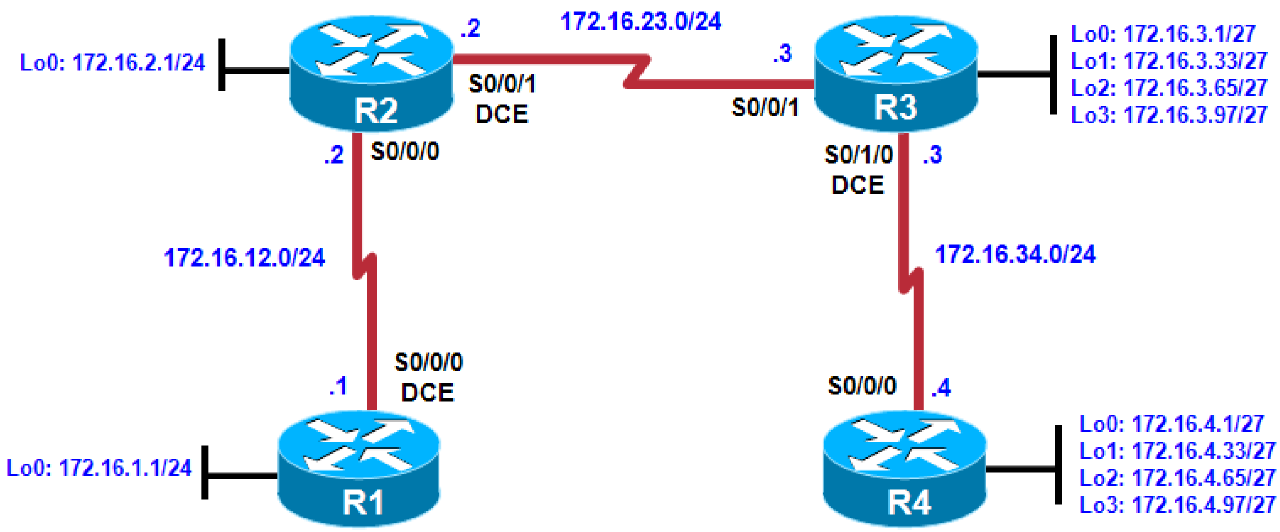
g. Run the Tcl script again to make sure you still have full connectivity after making the changes to the EIGRP default configuration. You should receive all ICMP echo replies back successfully.

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 2 Lab 2-6, EIGRP Challenge Lab

Topology



Objectives

- Implement a topology and EIGRP routing.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Challenge Steps

1. Configure all interfaces in the topology diagram with the IP addresses shown. Configure a bandwidth of 64 on all serial interfaces.

2. Configure EIGRP AS 1 to route all networks shown in the diagram.

3. Disable auto-summarization.

4. Configure R4 to summarize its loopback addresses to the most specific summary possible.

5. Do not multicast EIGRP hellos on the network between R1 and R2.

6. Modify the hello timers on the link between R2 and R3 to send a hello packet every 2 seconds.
7. Use MD5 authentication with the key “cisco” over the link between R3 and R4.
8. Run a Tcl script on all routers to verify that there is connectivity between the correct routers and IP addresses.

As you work through the challenge steps, you can document commands used and ideas for satisfying the requirements specified in the Notes section below.

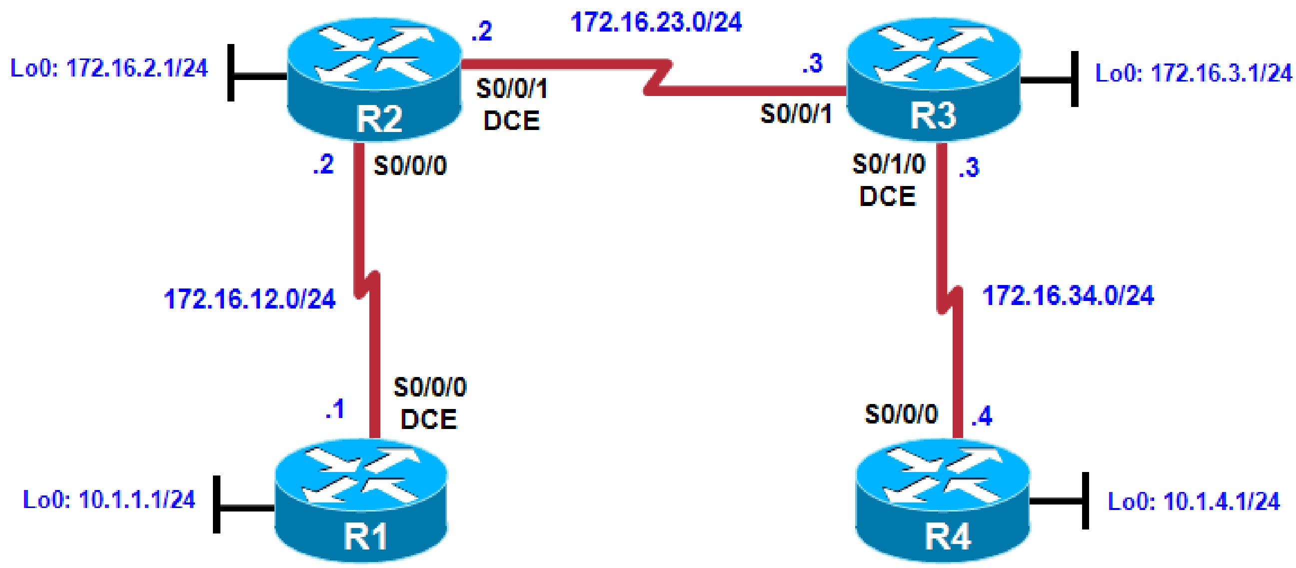
Notes: ____

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 2 Lab 2-7, Troubleshooting EIGRP

Topology



Objective

- Troubleshoot EIGRP configurations.

Background

In this lab, you troubleshoot existing configurations to achieve a working topology. You use troubleshooting techniques to correct anything in the scenario that prevents full IP connectivity. Full IP connectivity means that every IP address in the scenario should be reachable from every router. If you do not know where to start, try pinging remote addresses and see which ones are reachable (either manually performing pings or using a Tcl script).

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the advanced IP services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Requirements

- Use the IP addressing scheme shown in the diagram.
- All routers must participate in EIGRP AS 1.
- All networks in the diagram must be in EIGRP AS 1.
- Do not use any static routes, default routes, or other routing protocols.
- All IP addresses in the topology must be reachable from all routers.

Initial Configurations

Copy and paste the initial configurations into your routers.

Router R1

```
hostname R1
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.0
!
interface Serial0/0/0
 ip address 172.16.21.1 255.255.255.0
 clock rate 64000
 bandwidth 64
 no shutdown
!
router eigrp 1
 network 10.1.1.0 0.0.0.255
 network 172.16.12.0 0.0.0.255
 auto-summary
end
```

Router R2

```
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
 bandwidth 64
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.23.2 255.255.255.0
 clock rate 64000
 bandwidth 64
 no shutdown
!
router eigrp 1
 network 172.16.2.0 0.0.0.255
 network 172.16.12.0 0.0.0.255
 network 172.16.23.0 0.0.0.255
 no auto-summary
end
```


Router R3

```
hostname R3
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Serial0/0/1
 ip address 172.16.23.3 255.255.255.0
 bandwidth 64
 no shutdown
!
interface Serial0/1/0
 ip address 172.16.34.3 255.255.255.0
 clock rate 64000
 bandwidth 64
 no shutdown
!
router eigrp 1
 network 172.16.23.0 0.0.0.255
 network 172.16.30.0 0.0.0.255
 network 172.16.34.0 0.0.0.255
 no auto-summary
end
```

Router R4

```
hostname R4
!
interface Loopback0
 ip address 10.1.4.1 255.255.255.0
!
interface Serial0/0/0
 ip address 172.16.34.4 255.255.255.0
 bandwidth 64
 no shutdown
!
router eigrp 100
 network 10.1.4.0 0.0.0.255
 network 172.16.34.0 0.0.0.255
 auto-summary
end
```

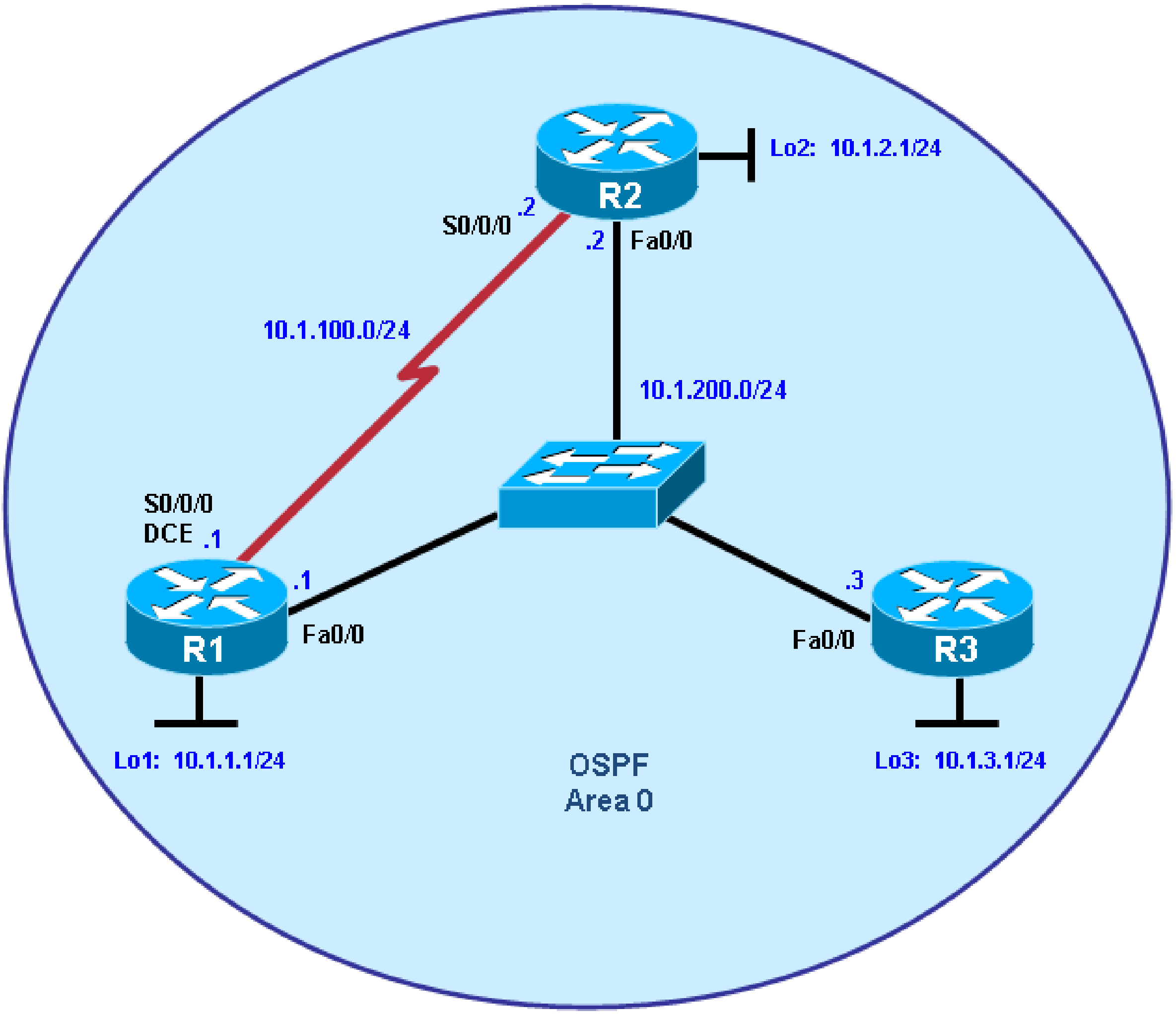
Notes:

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 3 Lab 3-1, Single-Area OSPF Link Costs and Interface Priorities

Topology



Objectives

- Configure single-area OSPF on a router.
- Advertise loopback interfaces into OSPF.
- Verify OSPF adjacencies.
- Verify OSPF routing information exchange.
- Modify OSPF link costs.
- Change interface priorities.
- Utilize debugging commands for troubleshooting OSPF.

Background

You are responsible for configuring the new network to connect your company's engineering, marketing, and accounting departments, represented by the loopback interfaces on each of the three routers. The physical devices have just been installed and connected by Fast Ethernet and serial cables. Configure OSPF to allow full connectivity between all departments.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. The switch is a Cisco WS-C2960-24TT-L with the Cisco IOS image c2960-lanbasek9-mz.122-46.SE.bin. You can use other routers (such as a 2801 or 2811), switches (such as a 2950), and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- 1 switch (Cisco 2960 with the Cisco IOS Release 12.2(46)SE C2960-LANBASEK9-M image or comparable)
- Serial and Ethernet cables

Step 1: Configure addressing and loopbacks.

a. Using the addressing scheme in the diagram, apply IP addresses to the Fast Ethernet interfaces on R1, R2, and R3. Create Loopback1 on R1, Loopback2 on R2, and Loopback3 on R3, and address them according to the diagram.

   Note: Depending on the router models you have, you might need to add clock rates to the DCE end of each connection (newer equipment adds this automatically). Verify connectivity across each serial link.

```
R1# configure terminal
R1(config)# interface Loopback1
R1(config-if)# description Engineering Department
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# exit
R1(config)# interface FastEthernet0/0
R1(config-if)# ip address 10.1.200.1 255.255.255.0
R1(config-if)# no shutdown

R2# configure terminal
R2(config)# interface Loopback2
R2(config-if)# description Marketing Department
R2(config-if)# ip address 10.1.2.1 255.255.255.0
R2(config-if)# exit
R2(config)# interface FastEthernet0/0
R2(config-if)# ip address 10.1.200.2 255.255.255.0
R2(config-if)# no shutdown

R3# configure terminal
R3(config)# interface Loopback3
R3(config-if)# description Accounting Department
R3(config-if)# ip address 10.1.3.1 255.255.255.0
R3(config-if)# exit
R3(config)# interface FastEthernet0/0
R3(config-if)# ip address 10.1.200.3 255.255.255.0
R3(config-if)# no shutdown
```


Leave the switch in its default (blank) configuration. By default, all switch ports are in VLAN1 and are not administratively down.

b. Configure the serial interfaces on R1 and R2 with the IP addresses shown in the diagram. Add the **clockrate** command where needed.

```
R1(config)# interface Serial 0/0/0
R1(config-if)# ip address 10.1.100.1 255.255.255.0
R1(config-if)# clockrate 64000
R1(config-if)# bandwidth 64
R1(config-if)# no shutdown

R2(config)# interface Serial 0/0/0
R2(config-if)# ip address 10.1.100.2 255.255.255.0
R2(config-if)# bandwidth 64
R2(config-if)# no shutdown
```

Note: The **bandwidth** command on the serial interfaces is used to match the actual bandwidth of the link. By default, OSPF calculates the cost of links based on the default interface bandwidth which may be either 128 or 1544 Kb/s, depending on the WIC type. In this case the **bandwidth 64** command is used because the real bandwidth of the serial interfaces is set to 64 Kbps. Refer to Step 5 for information on modifying OSPF link costs.

c. Verify that the appropriate interfaces are up and that you can ping across each link.

Step 2: Add physical interfaces to OSPF.

a. Enter the OSPF configuration prompt using the **router ospf** *process_number* command. The process number is a locally significant number that does not affect how OSPF works. For this lab, use process number 1 on all the routers.

b. Add interfaces with the **network** *address wildcard_mask* **area** *area* command. The address is an IP address. The mask is an inverse mask, similar to the kind used in an access list. The area is the OSPF area to put the interface. For this lab, use area 0, the backbone area, for all interfaces.

This command can be confusing at first. What it means is that any interface with an IP address that matches the address and wildcard mask combination in the network statement is added to the OSPF process in that area. The wildcard mask used in the **network** command has no influence on the actual IP subnet mask that is advertised with a network on an interface. The **network** command selects interfaces to be included into OSPF, but OSPF advertises the real subnet mask of the network attached to that interface (with the only exception being loopback interfaces).

For example, the command **network 10.1.200.1 0.0.0.0 area 0** adds the interface with the IP address of 10.1.200.1 and its network to the OSPF process into area 0. The wildcard mask of 0.0.0.0 means that all 32 bits of the IP address have to be an exact match. A 0 bit in the wildcard mask means that portion of the interface IP must match the address. A 1 bit means that the bit in the interface IP does not have to match that portion of the IP address.

The command **network 10.1.100.0 0.0.0.255 area 0** means that any interface whose IP address matches 10.1.100.0 for the first 3 octets will match the command and add it to area 0. The last octet is all 1s, because in the wildcard mask it is 255. This means that an interface with an IP of 10.1.100.1, 10.1.100.2, or 10.1.100.250 would match this address and wildcard combination and get added to OSPF.

Instead of using wildcard masks in the **network** command, it is possible to use subnet masks. The router converts the subnet masks to the wildcard format automatically. An easy way to calculate a wildcard mask from the subnet mask is to subtract the octet value for each octet from 255. For example, a subnet mask of 255.255.255.252 (/30) becomes 0.0.0.3 to capture all interfaces on that subnet:

```
  255.255.255.255
– 255.255.255.252
=   0.   0.   0.   3
```

Note: Another option for adding individual directly connected networks into the OSPF process is to use the **ip ospf** *process-id* **area** *area-id* interface command that is available with Cisco IOS version 12.3(11)T and later.

c. Enter the commands on R1. Exit to privileged EXEC mode and type **debug ip ospf adj**. The **debug** command lets you watch OSPF neighbors come up and see neighbor relationships.

```
R1(config)# router ospf 1
R1(config-router)# network 10.1.100.0 0.0.0.255 area 0
R1(config-router)# network 10.1.200.0 0.0.0.255 area 0
R1(config-router)# end
R1#

R1# debug ip ospf adj
OSPF adjacency events debugging is on
```

d. Add network statements to the other two routers.

```
R2(config)# router ospf 1
R2(config-router)# network 10.1.100.0 0.0.0.255 area 0
R2(config-router)# network 10.1.200.0 0.0.0.255 area 0

R3(config)# router ospf 1
R3(config-router)# network 10.1.200.0 0.0.0.255 area 0
```

e. Observe the debug output on R1. When you are finished, turn off debugging on R1 with the **undebug all** command.

f. What is the advantage of adding networks with a wildcard mask instead of using classful network addresses?

____

____

____

____

____

____

____

Step 3: Use OSPF show commands.

a. The **show ip protocols** command displays basic high-level routing protocol information. The output lists each OSPF process, the router ID, and which networks OSPF is routing for in each area. This information can be useful in debugging routing operations.

```
R1# 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 10.1.1.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.100.0 0.0.0.255 area 0
    10.1.200.1 0.0.0.0 area 0
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 110)
```

b. The **show ip ospf** command displays the OSPF process ID and router ID.

```
R1# show ip ospf
 Routing Process "ospf 1" with ID 10.1.1.1
 Start time: 00:17:44.612, Time elapsed: 00:10:51.408
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 Router is not originating router-LSAs with maximum metric
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 1. 1 normal 0 stub 0 nssa
 Number of areas transit capable is 0
 External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 2
        Area has no authentication
        SPF algorithm last executed 00:03:21.132 ago
        SPF algorithm executed 5 times
        Area ranges are
        Number of LSA 4. Checksum Sum 0x021A30
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

Notice the router ID listed in the output. The R1 ID is 10.1.1.1, even though you have not added this loopback into the OSPF process. The router chooses the router ID using the highest IP on a loopback interface when OSPF is configured. If an additional loopback interface with a higher IP address is added after OSPF is turned on, it does not become the router ID unless the router is reloaded, the OSPF configuration is removed and reentered, or the OSPF-level command **router-id** is used to modify the RID manually and the **clear ip ospf process** command is subsequently entered. If no loopback interfaces are present on the router, the router selects the highest available IP address among interfaces that are activated using the **no shutdown** command. If no IP addresses are assigned to interfaces, the OSPF process does not start.

c. The **show ip ospf neighbor** command displays important neighbor status, including the adjacency state, address, router ID, and connected interface.

```
R1# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.2.1          1   FULL/BDR        00:00:36    10.1.200.2      FastEthernet0/0
10.1.3.1          1   FULL/DR         00:00:35    10.1.200.3      FastEthernet0/0
10.1.2.1          0   FULL/  -        00:00:36    10.1.100.2      Serial0/0/0
```

If you need more detail than the standard one-line summaries of neighbors, use the **show ip ospf neighbor detail** command. However, generally, the regular command gives you all that you need.

d. The **show ip ospf interface** *interface_type number* command shows interface timers and network types.

```
R1# show ip ospf interface FastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 10.1.200.1/24, Area 0
  Process ID 1, Router ID 10.1.1.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DROTHER, Priority 1
  Designated Router (ID) 10.1.3.1, Interface address 10.1.200.3
  Backup Designated router (ID) 10.1.2.1, Interface address 10.1.200.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:09
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 2, Adjacent neighbor count is 2
     Adjacent with neighbor 10.1.3.1  (Designated Router)
     Adjacent with neighbor 10.1.2.1
  Suppress hello for 0 neighbor(s)
```

e. A variation of the previous command is the **show ip ospf interface brief** command, which displays each interface that is participating in the OSPF process on the router, the area it is in, its IP address, cost, state, and number of neighbors.

```
R1# show ip ospf interface brief
Interface    PID   Area        IP Address/Mask    Cost  State  Nbrs F/C
Fa0/0        1     0           10.1.200.1/24      1     DROTH  2/2
Se0/0/0      1     0           10.1.100.1/24      1     P2P    1/1
```

f. The **show ip ospf database** command displays the various LSAs in the OSPF database, organized by area and type.

```
R1# show ip ospf database

OSPF Router with ID (10.1.1.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.1.1.1        10.1.1.1        1782        0x80000002 0x001AC7 3
```



```
10.1.2.1        10.1.2.1        1783        0x80000001 0x001DC2 3
10.1.3.1        10.1.3.1        1720        0x80000002 0x00F077 1

                Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
10.1.200.1      10.1.1.1        1719        0x80000002 0x00EC3C

            OSPF Router with ID (10.1.1.1) (Process ID 1)
```

Step 4: Add loopback interfaces to OSPF.

a. All three routers have loopback interfaces, but they are not yet advertised in the routing process. You can verify this with the **show ip route** command on the three routers.

```
R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
C       10.1.1.0 is directly connected, Loopback1
C       10.1.100.0 is directly connected, Serial0/0/0
C       10.1.200.0 is directly connected, FastEthernet0/0

R2# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
C       10.1.2.0 is directly connected, Loopback2
C       10.1.100.0 is directly connected, Serial0/0/0
C       10.1.200.0 is directly connected, FastEthernet0/0

R3# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route
```



```
Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
C       10.1.3.0 is directly connected, Loopback3
O       10.1.100.0 [110/65] via 10.1.200.2, 00:06:39, FastEthernet0/0
                   [110/65] via 10.1.200.1, 00:06:39, FastEthernet0/0
C       10.1.200.0 is directly connected, FastEthernet0/0
```

b. For each router, the only loopback address displayed is the locally connected one. Add the loopbacks into the routing process for each router using the same **network** command previously used to add the physical interfaces.

```
R1(config)# router ospf 1
R1(config-router)# network 10.1.1.0 0.0.0.255 area 0

R2(config)# router ospf 1
R2(config-router)# network 10.1.2.0 0.0.0.255 area 0

R3(config)# router ospf 1
R3(config-router)# network 10.1.3.0 0.0.0.255 area 0
```

c. Verify that these networks have been added to the routing table using the **show ip route** command.

```
R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O       10.1.2.1/32 [110/2] via 10.1.200.2, 00:00:03, FastEthernet0/0
O       10.1.3.1/32 [110/2] via 10.1.200.3, 00:00:03, FastEthernet0/0
C       10.1.1.0/24 is directly connected, Loopback1
C       10.1.100.0/24 is directly connected, Serial0/0/0
C       10.1.200.0/24 is directly connected, FastEthernet0/0
```

Now you can see the loopbacks of the other routers, but their subnet mask is incorrect, because the default network type on loopback interfaces advertises them as /32 (host) routes. As you can see in the output of the **show ip ospf interface Lo1** command, the default OSPF network type for a loopback interface is LOOPBACK, causing the OSPF to advertise host routes instead of actual network masks.

```
R1# show ip ospf interface Lo1
Loopback1 is up, line protocol is up
  Internet Address 10.1.1.1/24, Area 0
  Process ID 1, Router ID 10.1.1.1, Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host
```

Note: The OSPF network type of LOOPBACK is a Cisco-proprietary extension that is not configurable but that is present on loopback interfaces by default. In some applications such as MPLS, the possible discrepancy between the real loopback interface mask and the advertised address/mask can lead to reachability or functionality issues, and care must be taken to either use /32 mask on loopbacks, or whenever a different mask is used, the OSPF network type must be changed to point-to-point.

d. To change this default behavior use the **ip ospf network point-to-point** command in interface configuration mode for each loopback. After the routes propagate, you see the correct subnet masks associated with those loopback interfaces.

```
R1(config)# interface loopback1
R1(config-if)# ip ospf network point-to-point

R2(config)# interface loopback2
R2(config-if)# ip ospf network point-to-point

R3(config)# interface loopback3
R3(config-if)# ip ospf network point-to-point

R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 5 subnets
O       10.1.3.0 [110/2] via 10.1.200.3, 00:00:01, FastEthernet0/0
O       10.1.2.0 [110/2] via 10.1.200.2, 00:00:01, FastEthernet0/0
C       10.1.1.0 is directly connected, Loopback1
C       10.1.100.0 is directly connected, Serial0/0/0
C       10.1.200.0 is directly connected, FastEthernet0/0
```

e. Use the following Tcl script to verify connectivity to all addresses in the topology.

```
R1# tclsh

foreach address {
10.1.1.1
10.1.2.1
10.1.3.1
10.1.100.1
10.1.100.2
10.1.200.1
10.1.200.2
10.1.200.3
} {
ping $address }
```

Step 5: Modify OSPF link costs.

When you use the **show ip route** command on R1, you see that the most direct route to the R2 loopback is through its Ethernet connection. Next to this route is a pair in the form [administrative distance / metric]. The default administrative distance of OSPF on Cisco routers is 110. The metric depends on the link type. OSPF always chooses the route with the lowest metric, which is a sum of all link costs.

You can modify a single link cost by using the interface command **ip ospf cost** *cost*. Use this command on both ends of the link. In the following commands, the link cost of the Fast Ethernet connection between the three routers is changed to a cost of 50. Notice the change in the metrics in the routing table.

```
R1(config)# interface FastEthernet 0/0
R1(config-if)# ip ospf cost 50

R2(config)# interface FastEthernet 0/0
R2(config-if)# ip ospf cost 50

R3(config)# interface FastEthernet 0/0
R3(config-if)# ip ospf cost 50

R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 5 subnets
O       10.1.3.0 [110/51] via 10.1.200.3, 00:01:40, FastEthernet0/0
O       10.1.2.0 [110/51] via 10.1.200.2, 00:01:40, FastEthernet0/0
C       10.1.1.0 is directly connected, Loopback1
C       10.1.100.0 is directly connected, Serial0/0/0
C       10.1.200.0 is directly connected, FastEthernet0/0
```

For reference, here are some default link costs (taken from Cisco.com):

- 64-kb/s serial link: 1562
- T1 (1.544-Mb/s serial link): 64
- E1 (2.048-Mb/s serial link): 48
- Ethernet: 10
- Fast Ethernet: 1
- FDDI: 1
- X25: 5208
- ATM: 1

OSPF uses a reference bandwidth of 100 Mb/s for cost calculation. The formula to calculate the cost is the reference bandwidth divided by the interface bandwidth. For example, in the case of Ethernet, is the cost is 100 Mb/s / 10 Mb/s = 10.

The above link costs do not include Gigabit Ethernet, which is significantly faster than Fast Ethernet, but would still have a cost of 1 using the default reference bandwidth of 100 Mb/s.

The cost calculation can be adjusted to account for network links that are faster than 100 Mb/s by using the **auto-cost reference-bandwidth** command to change the reference bandwidth. For example, to change the reference bandwidth to 1000 Mb/s (Gigabit Ethernet), use the following commands:

```
R1(config)# router ospf 1
R1(config-router)# auto-cost reference-bandwidth 1000
```



```
% OSPF: Reference bandwidth is changed.
    Please ensure reference bandwidth is consistent across all routers.
```

Note: If the **ip ospf cost** *cost* command is used on the interface, as is the case here, it overrides this formulated cost.

Note: The above example is for reference only and should not be entered on R1.

Step 6: Modify interface priorities to control the DR and BDR election.

If you use the **show ip ospf neighbor detail** command on any of the routers, you see that for the Ethernet network, R3 is the DR (designated router) and R2 is the BDR (backup designated router). These designations are determined by the interface priority for all routers in that network, which you see in the **show** output.

The default priority is 1. If all the priorities are the same (which happens by default), the DR election is then based on router IDs. The highest router ID router becomes the DR, and the second highest becomes the BDR. All other routers become DROTHERs.

Note: If your routers do not have this exact behavior, it might be because of the order in which the routers came up. Sometimes a router does not leave the DR position unless its interface goes down and another router takes over. Your routers might not behave exactly like the example.

Use the **ip ospf priority** *number* interface command to change the OSPF priorities on R1 and R2 to make R1 the DR and R2 the BDR. After changing the priority on both interfaces, look at the output of the **show ip ospf neighbor detail** command. You can also see the change with the **show ip ospf neighbor** command, but it requires more interpretation because it comes up with states per neighbor, rather than stating the DR and BDR on a neighbor adjacency network.

```
R1(config)# interface FastEthernet 0/0
R1(config-if)# ip ospf priority 10

R2(config)# interface FastEthernet 0/0
R2(config-if)# ip ospf priority 5

R1# show ip ospf neighbor detail
 Neighbor 10.1.2.1, interface address 10.1.200.2
    In the area 0 via interface FastEthernet0/0
    Neighbor priority is 5, State is FULL, 12 state changes
    DR is 10.1.200.1 BDR is 10.1.200.2
    Options is 0x52
    LLS Options is 0x1 (LR)
    Dead timer due in 00:00:37
    Neighbor is up for 00:01:32
    Index 3/3, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
 Neighbor 10.1.3.1, interface address 10.1.200.3
    In the area 0 via interface FastEthernet0/0
    Neighbor priority is 1, State is FULL, 12 state changes
    DR is 10.1.200.1 BDR is 10.1.200.2
    Options is 0x52
    LLS Options is 0x1 (LR)
    Dead timer due in 00:00:30
    Neighbor is up for 00:01:12
    Index 1/1, retransmission queue length 0, number of retransmission 3
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
```



```
      Last retransmission scan time is 0 msec, maximum is 0 msec
   Neighbor 10.1.2.1, interface address 10.1.100.2
      In the area 0 via interface Serial0/0/0
      Neighbor priority is 0, State is FULL, 12 state changes
      DR is 0.0.0.0 BDR is 0.0.0.0
      Options is 0x52
      LLS Options is 0x1 (LR)
      Dead timer due in 00:00:35
      Neighbor is up for 00:01:44
      Index 2/2, retransmission queue length 0, number of retransmission 2
      First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
      Last retransmission scan length is 2, maximum is 2
      Last retransmission scan time is 0 msec, maximum is 0 msec
```

Note: To make a router take over as DR, use the **clear ip ospf process** command on all the routers after changing the priorities. Another method of demonstrating the election process and priorities is to shutdown and reactivate all ports on the switch simultaneously. The switch can be configured with **spanning-tree portfast default** and all ports can be shutdown and reactivated using the following commands.

```
interface range fa0/1 - 24
 shutdown
 no shutdown
```

What is the purpose of a DR in OSPF?

____

____

____

____

____

____

____

____

____

What is the purpose of a BDR in OSPF?

____

____

____

____

____

Challenge: Topology Change

OSPF, like many link-state routing protocols, is reasonably fast when it comes to convergence. To test this, have R3 send a large number of pings to the R1 loopback. By default, the pings take the path from R3 to R1 over Fast Ethernet because it has the lowest total path cost.

a. Check the path from R3 to R1 by performing a traceroute on R3 to the loopback of R1.

```
R3# traceroute 10.1.1.1
```



```
Type escape sequence to abort.
Tracing the route to 10.1.1.1

  1 10.1.200.1 0 msec 0 msec *
```

Note: Read the next substep carefully before trying out the commands on routers.

b. Initiate a ping from R3 to the R1 loopback with a high repeat number using the command **ping** *ip* **repeat** *number* command. While this ping is going on, shut down the R1 Fa0/0 interface.

```
R3# ping 10.1.1.1 repeat 10000

R1(config)# interface FastEthernet 0/0
R1(config-if)# shutdown
```

Did you notice that some packets were dropped but then the pings started returning again?

____

____

____

How do you think OSPF convergence compares to other routing protocols, such as RIP? What about EIGRP?

____

____

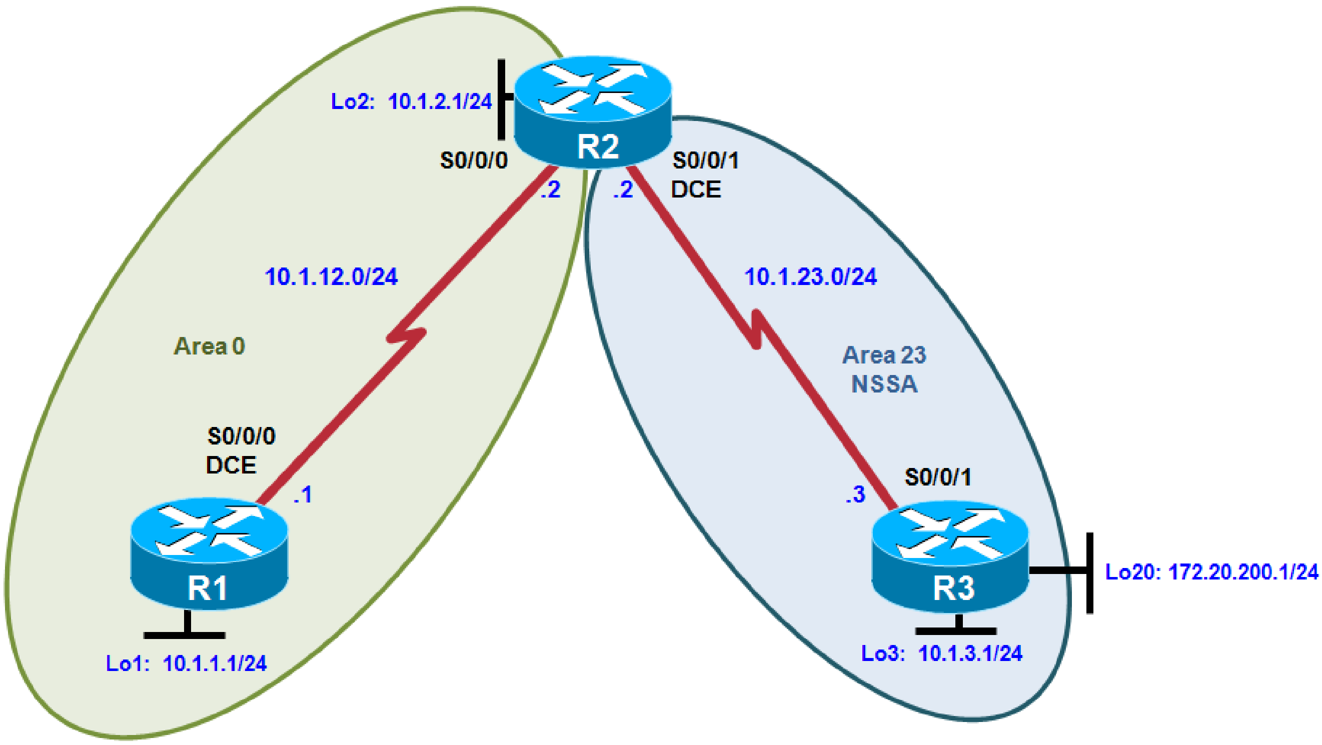
____

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 3 Lab 3-2, Multi-Area OSPF with Stub Areas and Authentication

Topology



Objectives

- Configure multiple-area OSPF on a router.
- Verify multiple-area behavior.
- Configure OSPF stub, totally stubby, and not-so-stubby areas.
- Configure OSPF authentication.

Background

You are responsible for configuring the new network to connect your company's engineering, marketing, and accounting departments, represented by loopback interfaces on each of the three routers. The physical devices have just been installed and connected by serial cables. Configure multiple-area OSPF to allow full connectivity between all departments.

R3 also has a loopback representing a connection to another autonomous system that is not part of OSPF.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router

model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Configure addressing and loopbacks.

a. Using the addressing scheme in the diagram, apply IP addresses to the serial interfaces on R1, R2, and R3. Create loopbacks on R1, R2, and R3, and address them according to the diagram.

Note: Depending on the router models you have, you might need to add clock rates to the DCE end of each connection (newer equipment adds this automatically). Verify connectivity across each serial link.

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# interface loopback 1
R1(config-if)# description Engineering Department
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# interface serial 0/0/0
R1(config-if)# ip address 10.1.12.1 255.255.255.0
R1(config-if)# clockrate 64000
R1(config-if)# no shutdown

R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# interface loopback 2
R2(config-if)# description Marketing Department
R2(config-if)# ip address 10.1.2.1 255.255.255.0
R2(config-if)# interface serial 0/0/0
R2(config-if)# ip address 10.1.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial 0/0/1
R2(config-if)# ip address 10.1.23.2 255.255.255.0
R2(config-if)# clockrate 64000
R2(config-if)# no shutdown

R3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# interface loopback 3
R3(config-if)# description Accounting Department
R3(config-if)# ip address 10.1.3.1 255.255.255.0
R3(config-if)# interface loopback 20
R3(config-if)# description Connection to another AS
R3(config-if)# ip address 172.20.200.1 255.255.255.0
R3(config-if)# interface serial 0/0/1
R3(config-if)# ip address 10.1.23.3 255.255.255.0
R3(config-if)# no shutdown
```

Step 2: Add interfaces into OSPF.

a. Create OSPF process 1 on routers R1 and R2. Configure the subnet of the serial link between R1 and R2 to be in OSPF area 0 using the **network** command. Add loopback 1 on R1 and loopback 2 on R2 into OSPF area 0. Change the network type on the loopback interfaces so that they are advertised with the correct subnet.


```
R1(config)# router ospf 1
R1(config-router)# network 10.1.12.0 0.0.0.255 area 0
R1(config-router)# network 10.1.1.0 0.0.0.255 area 0
R1(config-router)# exit
R1(config)# interface loopback 1
R1(config-if)# ip ospf network point-to-point

R2(config)# router ospf 1
R2(config-router)# network 10.1.12.0 0.0.0.255 area 0
R2(config-router)# network 10.1.2.0 0.0.0.255 area 0
R2(config-router)# exit
R2(config)# interface loopback 2
R2(config-if)# ip ospf network point-to-point
```

Note: Another option for adding individual directly connected networks into the OSPF process is to use the **ip ospf** *process-id* **area** *area-id* interface command that is available with Cisco IOS version 12.3(11)T and later.

b. Verify that both routers have OSPF neighbors using the **show ip ospf neighbors** command.

```
R1# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.2.1          0   FULL/  -        00:00:38    10.1.12.2       Serial0/0/0

R2# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.1.1          0   FULL/  -        00:00:35    10.1.12.1       Serial0/0/0
```

c. Verify that the routers can see each other's loopback with the **show ip route** command.

```
R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
O       10.1.2.0 [110/65] via 10.1.12.2, 00:00:10, Serial0/0/0
C       10.1.1.0 is directly connected, Loopback1

R2# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```



```
     10.0.0.0/24 is subnetted, 4 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
C       10.1.2.0 is directly connected, Loopback2
O       10.1.1.0 [110/65] via 10.1.12.1, 00:00:30, Serial0/0/0
C       10.1.23.0 is directly connected, Serial0/0/1
```

d. Add the subnet between R2 and R3 into OSPF area 23 using the **network** command. Add loopback 3 on R3 into area 23.

```
R2(config)# router ospf 1
R2(config-router)# network 10.1.23.0 0.0.0.255 area 23

R3(config)# router ospf 1
R3(config-router)# network 10.1.23.0 0.0.0.255 area 23
R3(config-router)# network 10.1.3.0 0.0.0.255 area 23
R3(config-router)# exit
R3(config)# interface loopback 3
R3(config-if)# ip ospf network point-to-point
```

e. Verify that this neighbor relationship comes up using the **show ip ospf neighbors** command.

```
R2# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.1.1          0   FULL/  -        00:00:36    10.1.12.1       Serial0/0/0
10.1.3.1          0   FULL/  -        00:00:36    10.1.23.3       Serial0/0/1
```

f. If you look at the output of the **show ip route** command on R1, you see a route to the R3 loopback. Notice that it is identified as an inter-area route.

```
R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 5 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
O IA    10.1.3.0 [110/129] via 10.1.12.2, 00:00:28, Serial0/0/0
O       10.1.2.0 [110/65] via 10.1.12.2, 00:01:38, Serial0/0/0
C       10.1.1.0 is directly connected, Loopback1
O IA    10.1.23.0 [110/128] via 10.1.12.2, 00:01:38, Serial0/0/0
```

g. Issue the **show ip route** command on R2. Notice that R2 has no inter-area routes because R2 is in both areas. It is an ABR, or area border router.

```
R2# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
```



```
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 5 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
O       10.1.3.0 [110/65] via 10.1.23.3, 00:00:50, Serial0/0/1
C       10.1.2.0 is directly connected, Loopback2
O       10.1.1.0 [110/65] via 10.1.12.1, 00:02:00, Serial0/0/0
C       10.1.23.0 is directly connected, Serial0/0/1
```

h. Using a Tcl script, verify connectivity to all interfaces from any router, with the exception of loopback 20 on R3 (172.20.200.1), which has not yet been configured as part of OSPF.

i. Use the following Tcl script to verify that you can ping all addresses in the topology.

```
R1# tclsh
R1(tcl)#

foreach address {
10.1.1.1
10.1.2.1
10.1.3.1
10.1.12.1
10.1.12.2
10.1.23.2
10.1.23.3
172.20.200.1
} {
ping $address }
```

Step 3: Configure a stub area.

a. Under the OSPF process on R2 and R3, make area 23 the stub area using the **area** *area* **stub** command. The adjacency between the two routers might go down during the transition period, but it should come back up afterwards.

```
R2(config)# router ospf 1
R2(config-router)# area 23 stub

R3(config)# router ospf 1
R3(config-router)# area 23 stub
```

b. Confirm that it comes up by using the **show ip ospf neighbors** command.

```
R2# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.1.1          0   FULL/  -        00:00:36    10.1.12.1       Serial0/0/0
10.1.3.1          0   FULL/  -        00:00:36    10.1.23.3       Serial0/0/1

R3# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.2.1          0   FULL/  -        00:00:31    10.1.23.2       Serial0/0/1
```

c. Using the **show ip route** command, you can see that R3 now has a default route pointing toward R2. A stub area does not receive any external routes. It receives a default route and OSPF inter-area routes.

```
R3# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.23.2 to network 0.0.0.0

     172.20.0.0/24 is subnetted, 1 subnets
C       172.20.200.0 is directly connected, Loopback20
     10.0.0.0/24 is subnetted, 5 subnets
O IA    10.1.12.0 [110/128] via 10.1.23.2, 00:00:56, Serial0/0/1
C       10.1.3.0 is directly connected, Loopback3
O IA    10.1.2.0 [110/65] via 10.1.23.2, 00:00:56, Serial0/0/1
O IA    10.1.1.0 [110/129] via 10.1.23.2, 00:00:56, Serial0/0/1
C       10.1.23.0 is directly connected, Serial0/0/1
O*IA 0.0.0.0/0 [110/65] via 10.1.23.2, 00:00:56, Serial0/0/1
```

d. Look at the output of the **show ip ospf** command to see what type each area is.

```
R2# show ip ospf
 Routing Process "ospf 1" with ID 10.1.2.1
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 It is an area border router
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 0. Checksum Sum 0x000000
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 2. 1 normal 1 stub 0 nssa
 Number of areas transit capable is 0
 External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 2
        Area has no authentication
        SPF algorithm last executed 00:02:11.680 ago
        SPF algorithm executed 5 times
        Area ranges are
        Number of LSA 4. Checksum Sum 0x01A85A
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
    Area 23
        Number of interfaces in this area is 1
```



```
        It is a stub area
          generates stub default route with cost 1
        Area has no authentication
        SPF algorithm last executed 00:01:38.276 ago
        SPF algorithm executed 8 times
        Area ranges are
        Number of LSA 6. Checksum Sum 0x027269
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

What are the advantages of having a router receive a default route rather than a more specific route?

____

____

Why do all routers in a stub area need to know that the area is a stub?

____

____

____

Step 4: Configure a totally stubby area.

A modified version of a stubby area is a totally stubby area. A totally stubby area ABR only allows in a single, default route from the backbone. To configure a totally stubby area, you only need to change a command at the ABR, R2 in this scenario. Under the router OSPF process, you will enter the **area 23 stub no-summary** command to replace the existing stub command for area 23. The **no-summary** option tells the router that this area will not receive summary (inter-area) routes.

a. To see how this works, issue the **show ip route** command on R3. Notice the inter-area routes, in addition to the default route generated by R2.

```
R3# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.23.2 to network 0.0.0.0

     172.20.0.0/24 is subnetted, 1 subnets
C       172.20.200.0 is directly connected, Loopback20
     10.0.0.0/24 is subnetted, 5 subnets
O IA    10.1.12.0 [110/128] via 10.1.23.2, 00:00:56, Serial0/0/1
C       10.1.3.0 is directly connected, Loopback3
O IA    10.1.2.0 [110/65] via 10.1.23.2, 00:00:56, Serial0/0/1
O IA    10.1.1.0 [110/129] via 10.1.23.2, 00:00:56, Serial0/0/1
C       10.1.23.0 is directly connected, Serial0/0/1
O*IA 0.0.0.0/0 [110/65] via 10.1.23.2, 00:00:56, Serial0/0/1
```


b. Look at the output of the **show ip ospf database** command on R2 to see which LSAs are in its OSPF database.

```
R2# show ip ospf database

            OSPF Router with ID (10.1.2.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.1.1.1        10.1.1.1        435         0x80000004 0x0056D6 3
10.1.2.1        10.1.2.1        358         0x80000003 0x0057D2 3

                Summary Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
10.1.3.0        10.1.2.1        174         0x80000001 0x00EFEF
10.1.23.0       10.1.2.1        354         0x80000001 0x0009C3

                Router Link States (Area 23)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.1.2.1        10.1.2.1        188         0x80000004 0x00298C 2
10.1.3.1        10.1.3.1        188         0x80000004 0x00B762 3

                Summary Net Link States (Area 23)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.0         10.1.2.1        207         0x80000001 0x003BF4
10.1.1.0        10.1.2.1        209         0x80000002 0x0022C0
10.1.2.0        10.1.2.1        209         0x80000002 0x00948D
10.1.12.0       10.1.2.1        209         0x80000002 0x009E3A
```

c. Enter the **stub no-summary** command on R2 (the ABR) under the OSPF process.

```
R2(config)# router ospf 1
R2(config-router)# area 23 stub no-summary
```

d. Go back to R3 and issue the **show ip route** command again. Notice that it shows only one incoming route from OSPF.

```
R3# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.23.2 to network 0.0.0.0

     172.20.0.0/24 is subnetted, 1 subnets
C       172.20.200.0 is directly connected, Loopback20
     10.0.0.0/24 is subnetted, 2 subnets
C       10.1.3.0 is directly connected, Loopback3
C       10.1.23.0 is directly connected, Serial0/0/1
O*IA 0.0.0.0/0 [110/65] via 10.1.23.2, 00:00:10, Serial0/0/1
```


e. Look at the **show ip ospf database** output to see which routes are in area 23.

```
R3# show ip ospf database

            OSPF Router with ID (10.1.3.1) (Process ID 1)

                Router Link States (Area 23)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.1.2.1        10.1.2.1        275         0x80000004 0x00298C 2
10.1.3.1        10.1.3.1        276         0x80000004 0x00B762 3

                Summary Net Link States (Area 23)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.0         10.1.2.1        68          0x80000002 0x0039F5
```

What are the advantages of making an area totally stubby instead of a regular stub area? What are the disadvantages?

____

____

____

____

Why did only the ABR need to know that the area was totally stubby rather than all routers in the area?

____

____

____

Step 5: Configure a not-so-stubby area.

Not-so-stubby areas (NSSAs) are similar to regular stub areas, except that they allow routes to be redistributed from an ASBR into that area with a special LSA type, which gets converted to a normal external route at the ABR.

a. Change area 23 into an NSSA. NSSAs are not compatible with stub areas, so the first thing to do is issue the **no area 23 stub** command on routers R2 and R3. Next, issue the **area** *area* **nssa** command on routers R2 and R3 to change area 23 to an NSSA. To generate an external route into the NSSA, use the **redistribute connected subnets** command on R3. This adds the previously unreachable loopback 20 into OSPF. Be sure to include the **subnets** keyword; otherwise, only classful networks are redistributed.

```
R2(config)# router ospf 1
R2(config-router)# no area 23 stub
R2(config-router)# area 23 nssa

R3(config)# router ospf 1
R3(config-router)# no area 23 stub
R3(config-router)# area 23 nssa
R3(config-router)# redistribute connected subnets
```

b. In the output of the **show ip ospf** command on R2, notice that area 23 is an NSSA and that R2 is performing the LSA type 7 to type 5 translation. If there are multiple ABRs to an NSSA, the ABR with the highest router ID performs the translation.


```
R2# show ip ospf
 Routing Process "ospf 1" with ID 10.1.2.1
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Supports Link-local Signaling (LLS)
 Supports area transit capability
 It is an area border and autonomous system boundary router
 Redistributing External Routes from,
 Initial SPF schedule delay 5000 msecs
 Minimum hold time between two consecutive SPFs 10000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Incremental-SPF disabled
 Minimum LSA interval 5 secs
 Minimum LSA arrival 1000 msecs
 LSA group pacing timer 240 secs
 Interface flood pacing timer 33 msecs
 Retransmission pacing timer 66 msecs
 Number of external LSA 1. Checksum Sum 0x00CA2F
 Number of opaque AS LSA 0. Checksum Sum 0x000000
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 2. 1 normal 0 stub 1 nssa
 Number of areas transit capable is 0
 External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 2
        Area has no authentication
        SPF algorithm last executed 00:03:11.636 ago
        SPF algorithm executed 9 times
        Area ranges are
        Number of LSA 4. Checksum Sum 0x01AC53
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
    Area 23
        Number of interfaces in this area is 1
        It is a NSSA area
        Perform type-7/type-5 LSA translation
        Area has no authentication
        SPF algorithm last executed 00:00:16.408 ago
        SPF algorithm executed 16 times
        Area ranges are
        Number of LSA 6. Checksum Sum 0x025498
        Number of opaque link LSA 0. Checksum Sum 0x000000
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

c. Look at the **show ip route** output on R2. Notice that the external route comes in as type N2 from R3. This is because it is a special NSSA external route.

```
R2# show ip route
Codes: C - connected, S - static, 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
```



```
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.20.0.0/24 is subnetted, 1 subnets
O N2    172.20.200.0 [110/20] via 10.1.23.3, 00:00:41, Serial0/0/1
     10.0.0.0/24 is subnetted, 5 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
O       10.1.3.0 [110/65] via 10.1.23.3, 00:00:47, Serial0/0/1
C       10.1.2.0 is directly connected, Loopback2
O       10.1.1.0 [110/65] via 10.1.12.1, 00:03:42, Serial0/0/0
C       10.1.23.0 is directly connected, Serial0/0/1
```

d. Look at the **show ip route** output on R1. Notice that the route is now a regular E2 external route, because R2 has performed the type 7 to type 5 translation.

```
R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.20.0.0/24 is subnetted, 1 subnets
O E2    172.20.200.0 [110/20] via 10.1.12.2, 00:01:22, Serial0/0/0
     10.0.0.0/24 is subnetted, 5 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
O IA    10.1.3.0 [110/129] via 10.1.12.2, 00:02:06, Serial0/0/0
O       10.1.2.0 [110/65] via 10.1.12.2, 00:04:22, Serial0/0/0
C       10.1.1.0 is directly connected, Loopback1
O IA    10.1.23.0 [110/128] via 10.1.12.2, 00:04:22, Serial0/0/0
```

e. Look at the **show ip route** output on R3. Notice that it no longer has a default route in it, but inter-area routes are coming in.

Note: An NSSA does not have the default route injected by the ABR (R2) automatically. It is possible to make the ABR inject the default route into the NSSA using the **area 23 nssa default-information-originate** command on R2.

```
R3# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```



```
     172.20.0.0/24 is subnetted, 1 subnets
C       172.20.200.0 is directly connected, Loopback20
     10.0.0.0/24 is subnetted, 5 subnets
O IA    10.1.12.0 [110/128] via 10.1.23.2, 00:02:11, Serial0/0/1
C       10.1.3.0 is directly connected, Loopback3
O IA    10.1.2.0 [110/65] via 10.1.23.2, 00:02:11, Serial0/0/1
O IA    10.1.1.0 [110/129] via 10.1.23.2, 00:02:11, Serial0/0/1
C       10.1.23.0 is directly connected, Serial0/0/1
```

f. Yet another type of area is a totally-stubby NSSA that combines the property of an NSSA area (injecting external routing information into OSPF) with a totally stubby behavior (accepting only default route from the backbone). Issue the **area 23 nssa no-summary** command on R2, similar to converting a stub area into a totally stubby area.

```
R2(config)# router ospf 1
R2(config-router)# area 23 nssa no-summary
```

g. Check the routing table on R3. Notice that the inter-area routes have been replaced by a single default route.

```
R3# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.23.2 to network 0.0.0.0

     172.20.0.0/24 is subnetted, 1 subnets
C       172.20.200.0 is directly connected, Loopback20
     10.0.0.0/24 is subnetted, 2 subnets
C       10.1.3.0 is directly connected, Loopback3
C       10.1.23.0 is directly connected, Serial0/0/1
O*IA 0.0.0.0/0 [110/65] via 10.1.23.2, 00:00:20, Serial0/0/1
```

h. On R2, look at the **show ip ospf database** output to see the various LSA types.

```
R2# show ip ospf database

            OSPF Router with ID (10.1.2.1) (Process ID 1)

                Router Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.1.1.1        10.1.1.1        944         0x80000004 0x0056D6 3
10.1.2.1        10.1.2.1        383         0x80000004 0x005BCB 3

                Summary Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
10.1.3.0        10.1.2.1        242         0x80000001 0x00EFEF
10.1.23.0       10.1.2.1        862         0x80000001 0x0009C3

                Router Link States (Area 23)

Link ID         ADV Router      Age         Seq#       Checksum Link count
```



```
10.1.2.1        10.1.2.1        257         0x80000007 0x00B0F7 2
10.1.3.1        10.1.3.1        209         0x80000007 0x003FCD 3

                Summary Net Link States (Area 23)

Link ID         ADV Router      Age         Seq#       Checksum
0.0.0.0         10.1.2.1        34          0x80000001 0x00C265

                Type-7 AS External Link States (Area 23)

Link ID         ADV Router      Age         Seq#       Checksum Tag
10.1.3.0        10.1.3.1        200         0x80000001 0x0076FC 0

                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
10.1.3.0        10.1.2.1        199         0x80000001 0x00CA2F 0
```

Where would it be useful to make an area into an NSSA?

____

____

____

Step 6: Configure OSPF interface authentication.

For security purposes, you can configure OSPF interfaces to use authentication.

a. Configure the link between R2 and R3 for plaintext authentication. To set up plaintext authentication on an interface, type **ip ospf authentication** at the interface command prompt. Then set the password to **cisco** with the **ip ospf authentication-key** *key-string* command.

```
R2(config)# interface serial 0/0/1
R2(config-if)# ip ospf authentication
R2(config-if)# ip ospf authentication-key cisco

R3(config)# interface serial 0/0/1
R3(config-if)# ip ospf authentication
R3(config-if)# ip ospf authentication-key cisco
```

Note: While configuring the authentication, the adjacency might go down if the dead timer expires on one of the routers. The relationship should be reestablished once authentication is configured on both sides.

b. Verify the authentication using the **show ip ospf interface** *interface* command.

```
R2# show ip ospf interface serial 0/0/1
Serial0/0/1 is up, line protocol is up
  Internet Address 10.1.23.2/24, Area 23
  Process ID 1, Router ID 10.1.2.1, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:09
  Supports Link-local Signaling (LLS)
  Index 1/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 4
  Last flood scan time is 0 msec, maximum is 0 msec
```



```
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.3.1
  Suppress hello for 0 neighbor(s)
  Simple password authentication enabled
```

c. MD5 authentication encrypts the password for stronger security. Configure the link between R1 and R2 for MD5 authentication using the **ip ospf authentication message-digest** interface command. Then set the password to **cisco** with the **ip ospf message-digest-key** *key_number* **md5** *key-string* command. Make sure that the key number is the same on both routers. In this case, use 1 for simplicity.

```
R1(config)# interface serial 0/0/0
R1(config-if)# ip ospf authentication message-digest
R1(config-if)# ip ospf message-digest-key 1 md5 cisco

R2(config)# interface serial 0/0/0
R2(config-if)# ip ospf authentication message-digest
R2(config-if)# ip ospf message-digest-key 1 md5 cisco
```

Note: The MD5 key number works differently than key chains. The router uses the most recently added key for authenticating sent packets. The key number does not have a direct influence on this behavior, that is, if the interface was configured with the MD5 key number 10 and later the key with number 5 was added, the router would use the key number 5 to digitally sign outbound sent packets. If a router having several MD5 keys on an interface detects that at least one of its neighbors has not yet started using the most recently added key, it engages in a simple key migration procedure: it sends each OSPF packet multiple times, with each instance of the packet authenticated by a particular MD5 key configured on the interface, one instance for each key. This ensures a smooth, gradual migration.

d. Verify the configuration using the **show ip ospf interface** *interface* command.

```
R1# show ip ospf interface serial 0/0/0
Serial0/0/0 is up, line protocol is up
  Internet Address 10.1.12.1/24, Area 0
  Process ID 1, Router ID 10.1.1.1, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:08
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 1
```

Why is configuring authentication for OSPF, or any routing protocol, a good idea?

____

____

____

e. Use the following Tcl script to verify connectivity to all addresses in the topology.

```
R1# tclsh
R1(tcl)#
```



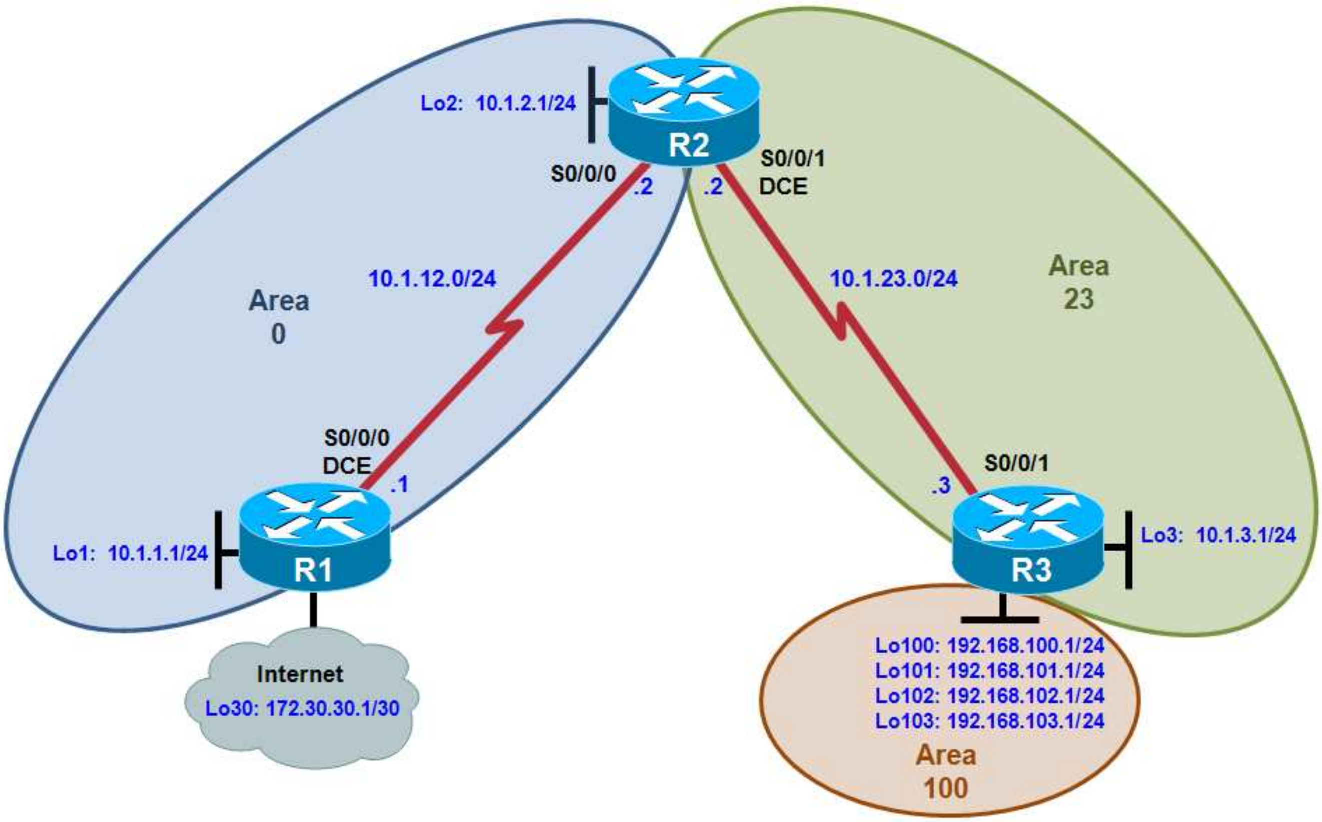
```
foreach address {
10.1.1.1
10.1.2.1
10.1.3.1
10.1.12.1
10.1.12.2
10.1.23.2
10.1.23.3
172.20.200.1
} {
ping $address }
```


Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 3 Lab 3-3, OSPF Virtual Links and Area Summarization

Topology



Objectives

- Configure multi-area OSPF on a router.
- Verify multi-area behavior.
- Create an OSPF virtual link.
- Summarize an area.
- Generate a default route into OSPF.

Background

You are responsible for configuring the new network to connect your company's engineering, marketing, and accounting departments, represented by loopback interfaces on each of the three routers. The physical devices have just been installed and connected by serial cables. Configure multiple-area OSPF to allow full connectivity between all departments.

In addition, R1 has a loopback interface representing a connection to the Internet. This connection will not be added into OSPF. R3 will have four additional loopback interfaces representing connections to branch offices.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Configure addressing and loopbacks.

Using the addressing scheme in the diagram, apply IP addresses to the serial interfaces on R1, R2, and R3. Create loopbacks on R1, R2, and R3, and address them according to the diagram.

```
R1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# interface loopback 1
R1(config-if)# description Engineering Department
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# interface loopback 30
R1(config-if)# ip address 172.30.30.1 255.255.255.252
R1(config-if)# interface serial 0/0/0
R1(config-if)# ip address 10.1.12.1 255.255.255.0
R1(config-if)# clockrate 64000
R1(config-if)# no shutdown

R2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)# interface loopback 2
R2(config-if)# description Marketing Department
R2(config-if)# ip address 10.1.2.1 255.255.255.0
R2(config-if)# interface serial 0/0/0
R2(config-if)# ip address 10.1.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial 0/0/1
R2(config-if)# ip address 10.1.23.2 255.255.255.0
R2(config-if)# clockrate 64000
R2(config-if)# no shutdown

R3# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# interface loopback 3
R3(config-if)# description Accounting Department
R3(config-if)# ip address 10.1.3.1 255.255.255.0
R3(config-if)# interface loopback 100
R3(config-if)# ip address 192.168.100.1 255.255.255.0
R3(config-if)# interface loopback 101
R3(config-if)# ip address 192.168.101.1 255.255.255.0
R3(config-if)# interface loopback 102
R3(config-if)# ip address 192.168.102.1 255.255.255.0
R3(config-if)# interface loopback 103
R3(config-if)# ip address 192.168.103.1 255.255.255.0
R3(config-if)# interface serial 0/0/1
R3(config-if)# ip address 10.1.23.3 255.255.255.0
R3(config-if)# no shutdown
```


Step 2: Add interfaces into OSPF.

a. Create OSPF process 1 on all three routers. Using the **network** command, configure the subnet of the serial link between R1 and R2 to be in OSPF area 0. Add loopback 1 on R1 and loopback 2 on R2 into OSPF area 0.

   Note: The default behavior of OSPF for loopback interfaces is to advertise a 32-bit host route. To ensure that the full /24 network is advertised, use the **ip ospf network point-to-point** command. Change the network type on the loopback interfaces so that they are advertised with the correct subnet.

```
R1(config)# router ospf 1
R1(config-router)# network 10.1.12.0 0.0.0.255 area 0
R1(config-router)# network 10.1.1.0 0.0.0.255 area 0
R1(config-router)# exit
R1(config)# interface loopback 1
R1(config-if)# ip ospf network point-to-point

R2(config)# router ospf 1
R2(config-router)# network 10.1.12.0 0.0.0.255 area 0
R2(config-router)# network 10.1.2.0 0.0.0.255 area 0
R2(config-router)# exit
R2(config)# interface loopback 2
R2(config-if)# ip ospf network point-to-point
```

b. Verify that you can see OSPF neighbors in the **show ip ospf neighbors** output on both routers. Verify that the routers can see each other's loopback with the **show ip route** command.

```
R1# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.2.1          0   FULL/  -        00:00:38    10.1.12.2       Serial0/0/0

R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
O       10.1.2.0 [110/65] via 10.1.12.2, 00:00:10, Serial0/0/0
C       10.1.1.0 is directly connected, Loopback1
     172.30.0.0/30 is subnetted, 1 subnets
C       172.30.30.0 is directly connected, Loopback30

R2# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
172.30.30.1       0   FULL/  -        00:00:35    10.1.12.1       Serial0/0/0

R2# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 4 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
C       10.1.2.0 is directly connected, Loopback2
O       10.1.1.0 [110/65] via 10.1.12.1, 00:00:30, Serial0/0/0
C       10.1.23.0 is directly connected, Serial0/0/1
```

c. Add the subnet between R2 and R3 into OSPF area 23 using the **network** command. Add loopback 3 on R3 into area 23.

```
R2(config)# router ospf 1
R2(config-router)# network 10.1.23.0 0.0.0.255 area 23

R3(config)# router ospf 1
R3(config-router)# network 10.1.23.0 0.0.0.255 area 23
R3(config-router)# network 10.1.3.0 0.0.0.255 area 23
R3(config-router)# exit
R3(config)# interface loopback 3
R3(config-if)# ip ospf network point-to-point
```

d. Verify that this neighbor relationship comes up with the **show ip ospf neighbors** command.

```
R2# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
172.30.30.1       0   FULL/  -        00:00:36    10.1.12.1       Serial0/0/0
192.168.103.1     0   FULL/  -        00:00:36    10.1.23.3       Serial0/0/1
```

e. Using a Tcl script, verify connectivity to all interfaces from any router, with the exception of loopback 30 on R1, and R3 loopbacks 100 through 103.

```
R1# tclsh
R1(tcl)#

foreach address {
10.1.1.1
10.1.2.1
10.1.3.1
10.1.12.1
10.1.12.2
10.1.23.2
10.1.23.3
172.30.30.1
192.168.100.1
192.168.101.1
192.168.102.1
192.168.103.1
```

Step 3: Create a virtual link.

a. Add loopbacks 100 through 103 on R3 to the OSPF process in area 100 using the **network** command. Change the network type to advertise the correct subnet mask.


```
R3(config)# router ospf 1
R3(config-router)# network 192.168.100.0 0.0.3.255 area 100
R3(config-router)# exit
R3(config)# interface loopback 100
R3(config-if)# ip ospf network point-to-point
R3(config-if)# interface loopback 101
R3(config-if)# ip ospf network point-to-point
R3(config-if)# interface loopback 102
R3(config-if)# ip ospf network point-to-point
R3(config-if)# interface loopback 103
R3(config-if)# ip ospf network point-to-point
```

b. Look at the output of the **show ip route** command on R2. Notice that the routes to those networks do not appear. The reason for this behavior is that area 100 on R3 is not connected to the backbone. It is only connected to area 23. If an area is not connected to the backbone, its routes are not advertised outside of its area.

```
R2# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 5 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
O       10.1.3.0 [110/65] via 10.1.23.3, 00:01:00, Serial0/0/1
C       10.1.2.0 is directly connected, Loopback2
O       10.1.1.0 [110/65] via 10.1.12.1, 00:03:10, Serial0/0/0
C       10.1.23.0 is directly connected, Serial0/0/1
```

What would happen if routes could pass between areas without going through the backbone?

____

____

____

You can get around this situation by creating a virtual link. A virtual link is an OSPF feature that creates a logical extension of the backbone area across a regular area, without actually adding any physical interfaces into area 0.

Note: Prior to creating a virtual link you need to identify the OSPF router ID for the routers involved (R2 and R3), using a command such as **show ip ospf**, **show ip protocols** or **show ip ospf interface**. The output for the **show ip ospf** command on R1 and R3 is shown below.

```
R2# show ip ospf
 Routing Process "ospf 1" with ID 10.1.2.1
<output omitted>

R3# show ip ospf
 Routing Process "ospf 1" with ID 192.168.103.1
<output omitted>
```


c. Create a virtual link using the **area** *transit_area* **virtual-link** *router-id* OSPF configuration command on both R2 and R3.

```
R2(config)# router ospf 1
R2(config-router)# area 23 virtual-link 192.168.103.1

R3(config)# router ospf 1
R3(config-router)# area 23 virtual-link 10.1.2.1
```

Note: To ensure that the router ID of the virtual link endpoints remains constant, you can statically configure the OSPF router ID of the virtual link endpoints using the **router-id** command.

d. After you see the adjacency over the virtual interface come up, issue the **show ip route** command on R2 and see the routes from area 100. You can verify the virtual link with the **show ip ospf neighbor** and **show ip ospf interface** commands.

```
R2# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 5 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
O       10.1.3.0 [110/65] via 10.1.23.3, 00:01:35, Serial0/0/1
C       10.1.2.0 is directly connected, Loopback2
O       10.1.1.0 [110/65] via 10.1.12.1, 00:01:35, Serial0/0/0
C       10.1.23.0 is directly connected, Serial0/0/1
O IA 192.168.102.0/24 [110/65] via 10.1.23.3, 00:00:05, Serial0/0/1
O IA 192.168.103.0/24 [110/65] via 10.1.23.3, 00:00:05, Serial0/0/1
O IA 192.168.100.0/24 [110/65] via 10.1.23.3, 00:00:57, Serial0/0/1
O IA 192.168.101.0/24 [110/65] via 10.1.23.3, 00:00:16, Serial0/0/1

R2# show ip ospf neighbor
Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.103.1     0   FULL/  -           -        10.1.23.3       OSPF_VL0
172.30.30.1       0   FULL/  -        00:00:30    10.1.12.1       Serial0/0/0
192.168.103.1     0   FULL/  -        00:00:30    10.1.23.3       Serial0/0/1

R2# show ip ospf interface
OSPF_VL0 is up, line protocol is up
  Internet Address 10.1.23.2/24, Area 0
  Process ID 1, Router ID 10.1.2.1, Network Type VIRTUAL_LINK, Cost: 64
  Configured as demand circuit.
  Run as demand circuit.
  DoNotAge LSA allowed.
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:03
  Supports Link-local Signaling (LLS)
```



```
  Index 3/4, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 192.168.103.1  (Hello suppressed)
  Suppress hello for 1 neighbor(s)
<output omitted>
```

When are virtual links useful?

____

____

____

Why are virtual links a poor long-term solution?

____

____

____

Step 4: Summarize an area.

Loopbacks 100 through 103 can be summarized into one supernet of 192.168.100.0 /22. You can configure area 100 to be represented by this single summary route.

a. Configure R3 (the ABR) to summarize this area using the **area** *area* **range** *network mask* command.

```
R3(config)# router ospf 1
R3(config-router)# area 100 range 192.168.100.0 255.255.252.0
```

b. You can see the summary route on R2 with the **show ip route** and **show ip ospf database** commands.

```
R2# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 5 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
O       10.1.3.0 [110/65] via 10.1.23.3, 00:07:25, Serial0/0/1
C       10.1.2.0 is directly connected, Loopback2
O       10.1.1.0 [110/65] via 10.1.12.1, 00:07:25, Serial0/0/0
C       10.1.23.0 is directly connected, Serial0/0/1
O IA 192.168.100.0/22 [110/65] via 10.1.23.3, 00:00:01, Serial0/0/1

R2# show ip ospf database

            OSPF Router with ID (10.1.2.1) (Process ID 1)

                Router Link States (Area 0)
```

```
Link ID         ADV Router      Age         Seq#       Checksum Link count
10.1.2.1        10.1.2.1        341         0x80000003 0x0028DD 4
172.30.30.1     172.30.30.1     1665        0x80000002 0x000E67 3
192.168.103.1   192.168.103.1   1     (DNA) 0x80000003 0x00A374 1

                Summary Net Link States (Area 0)

Link ID         ADV Router      Age         Seq#       Checksum
10.1.3.0        10.1.2.1        1268        0x80000001 0x00EFEF
10.1.3.0        192.168.103.1   6     (DNA) 0x80000001 0x00FD5E
10.1.23.0       10.1.2.1        1311        0x80000001 0x0009C3
10.1.23.0       192.168.103.1   6     (DNA) 0x80000001 0x00996F
192.168.100.0   192.168.103.1   1     (DNA) 0x80000002 0x009A04

                Router Link States (Area 23)

Link ID         ADV Router      Age         Seq#       Checksum Link count
10.1.2.1        10.1.2.1        341         0x80000003 0x00DD8B 2
192.168.103.1   192.168.103.1   342         0x80000003 0x002E57 3

                Summary Net Link States (Area 23)

Link ID         ADV Router      Age         Seq#       Checksum
10.1.1.0        10.1.2.1        1321        0x80000001 0x0006DB
10.1.2.0        10.1.2.1        1321        0x80000001 0x0078A8
10.1.12.0       10.1.2.1        1321        0x80000001 0x008255
192.168.100.0   192.168.103.1   157         0x80000002 0x009A04
```

c. Notice on R3 that OSPF has generated a summary route pointing toward Null0.

```
R3# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 5 subnets
O       10.1.12.0 [110/128] via 10.1.23.2, 00:01:18, Serial0/0/1
C       10.1.3.0 is directly connected, Loopback3
O       10.1.2.0 [110/65] via 10.1.23.2, 00:01:18, Serial0/0/1
O       10.1.1.0 [110/129] via 10.1.23.2, 00:01:18, Serial0/0/1
C       10.1.23.0 is directly connected, Serial0/0/1
C    192.168.102.0/24 is directly connected, Loopback102
C    192.168.103.0/24 is directly connected, Loopback103
C    192.168.100.0/24 is directly connected, Loopback100
C    192.168.101.0/24 is directly connected, Loopback101
O    192.168.100.0/22 is a summary, 00:01:19, Null0
```

This behavior is known as sending unknown traffic to the “bit bucket.” This means that if the router advertising the summary route receives a packet destined for something covered by that summary but not in the routing table, it drops it.

What is the reasoning behind this behavior?

____

____

____

____

____

____

Step 5: Generate a default route into OSPF.

You can simulate loopback 30 on R1 to be a connection to the Internet. You do not need to advertise this specific network to the rest of the network. Instead, you can just have a default route for all unknown traffic to go to R1.

a. To have R1 generate a default route, use the OSPF configuration command **default-information originate always**. The **always** keyword is necessary for generating a default route in this scenario. Without this keyword, a default route is generated only into OSPF if one exists in the routing table.

```
R1(config)# router ospf 1
R1(config-router)# default-information originate always
```

b. Verify that the default route appears on R2 and R3 with the **show ip route** command.

```
R2# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.12.1 to network 0.0.0.0

     10.0.0.0/24 is subnetted, 5 subnets
C       10.1.12.0 is directly connected, Serial0/0/0
O       10.1.3.0 [110/65] via 10.1.23.3, 00:10:36, Serial0/0/1
C       10.1.2.0 is directly connected, Loopback2
O       10.1.1.0 [110/65] via 10.1.12.1, 00:00:19, Serial0/0/0
C       10.1.23.0 is directly connected, Serial0/0/1
O*E2 0.0.0.0/0 [110/1] via 10.1.12.1, 00:00:09, Serial0/0/0
O IA 192.168.100.0/22 [110/65] via 10.1.23.3, 00:00:19, Serial0/0/1

R3# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 10.1.23.2 to network 0.0.0.0
```



```
     10.0.0.0/24 is subnetted, 5 subnets
O       10.1.12.0 [110/128] via 10.1.23.2, 00:00:35, Serial0/0/1
C       10.1.3.0 is directly connected, Loopback3
O       10.1.2.0 [110/65] via 10.1.23.2, 00:00:35, Serial0/0/1
O       10.1.1.0 [110/129] via 10.1.23.2, 00:00:35, Serial0/0/1
C       10.1.23.0 is directly connected, Serial0/0/1
C    192.168.102.0/24 is directly connected, Loopback102
C    192.168.103.0/24 is directly connected, Loopback103
C    192.168.100.0/24 is directly connected, Loopback100
C    192.168.101.0/24 is directly connected, Loopback101
O*E2 0.0.0.0/0 [110/1] via 10.1.23.2, 00:00:26, Serial0/0/1
O    192.168.100.0/22 is a summary, 00:03:28, Null0
```

c. You should be able to ping the interface connecting to the Internet from R2 or R3, despite never being advertised into OSPF.

```
R3# ping 172.30.30.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.30.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/32 ms
```

d. Use the following Tcl script to verify connectivity to all addresses in the topology.

```
R1# tclsh
R1(tcl)#

foreach address {
10.1.1.1
10.1.2.1
10.1.3.1
10.1.12.1
10.1.12.2
10.1.23.2
10.1.23.3
172.30.30.1
192.168.100.1
192.168.101.1
192.168.102.1
192.168.103.1
} {
ping $address }
```

Challenge: Configure OSPF Authentication

Configure OSPF MD5 authentication on the link between R2 and R3, using key ID 1 and the password cisco. Record the commands used below.

____

____

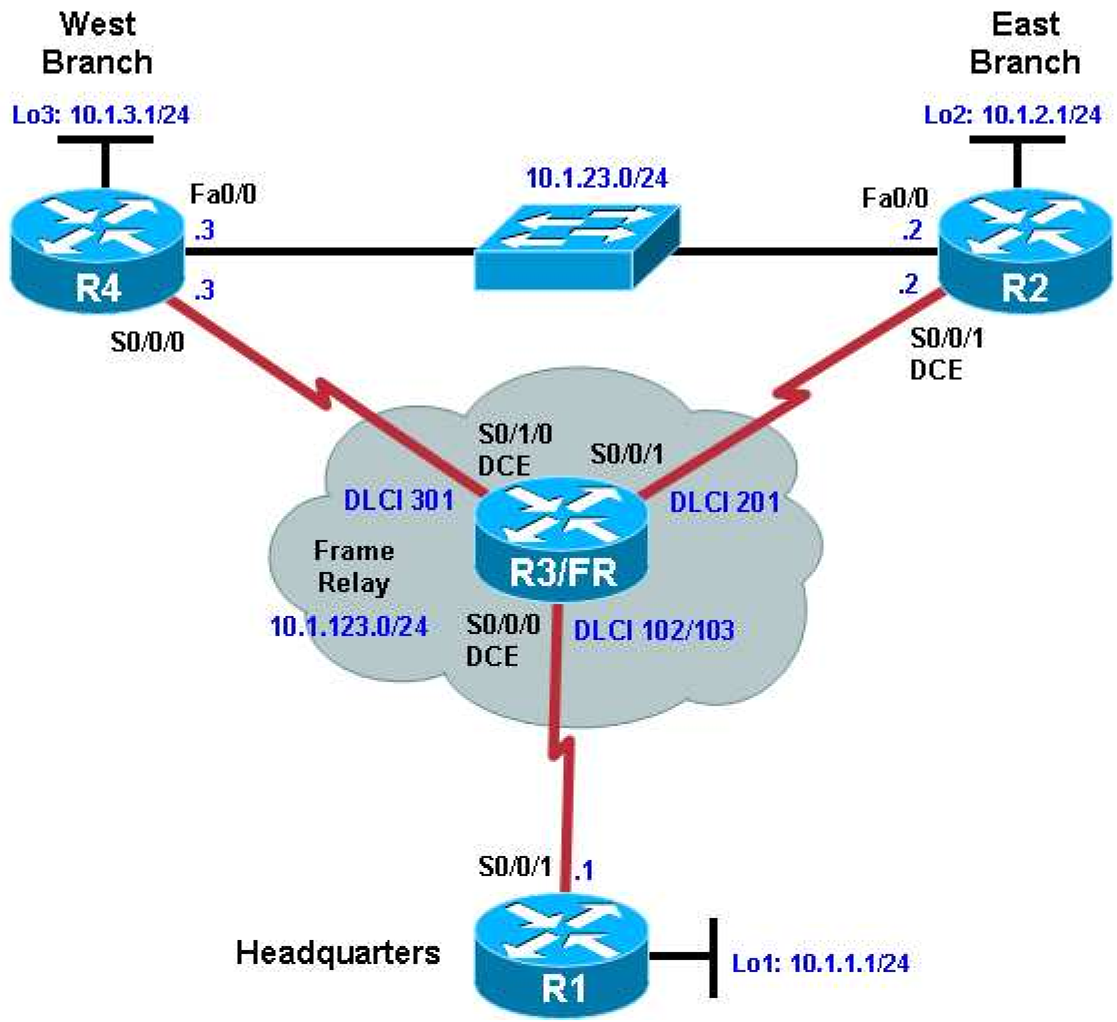
____

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 3 Lab 3-4, OSPF over Frame Relay

Topology



Objectives

- Configure OSPF over Frame Relay.
- Use non-broadcast and point-to-multipoint OSPF network types.
- Modify default OSPF timers.

Background

You are responsible for configuring the new network to connect your company's East branch and West branch through the company headquarters represented by loopback interfaces on each of the three routers. The physical devices have just been installed and connected over Frame Relay hub-and-spoke topology. Configure OSPF to allow full connectivity between all departments.

To simulate the Frame Relay WAN connections, use a router with three serial ports to act as a Frame Relay switch. The configuration of the router as a Frame Relay switch is described in Step 2. The Fast Ethernet connection between router EAST and router WEST represents a backup link from a service provider.

In real-world Frame Relay deployments, a modem or CSU/DSU normally provides the clocking functions (DCE) for each serial link. However, in this lab, R2 interface Serial0/0/1 is configured as the DCE for compatibility with other labs. If you are uncertain which side of the connection is the DCE, use the **show controllers serial** *interface-number* command:

```
FRS# show controllers serial0/0/0
Interface Serial0/0/0
Hardware is GT96K
DCE V.35, clock rate 64000
```

Note: In this lab, router R3 acts as the Frame Relay switch and requires two serial interface cards. If you are using an Adtran as a Frame Relay switch, see Appendix A for the Adtran configuration. When using the Adtran as the Frame Relay switch, the clock (DCE) is provided for each serial link.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. The switch is a Cisco WS-C2960-24TT-L with the Cisco IOS image c2960-lanbasek9-mz.122-46.SE.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- 1 switch (Cisco 2960 with the Cisco IOS Release 12.2(46)SE C2960-LANBASEK9-M image or comparable)
- Serial and Ethernet cables

Step 1: Configure Frame Relay and addressing on the HQ, EAST, and WEST routers.

a. Configure the router physical interfaces with IP addresses. On the interfaces connected to the Frame Relay switch, configure Frame Relay encapsulation with Inverse ARP disabled. Inverse ARP is disabled in this lab so that you have exclusive control over the IP/DLCI mappings. Use **frame relay map** statements to configure local Frame Relay maps so that you can ping the router's interface. Also set up the loopback interfaces.

Router R1 (Hostname HQ)

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# hostname HQ
HQ(config)# interface loopback 1
HQ(config-if)# ip address 10.1.1.1 255.255.255.0
HQ(config-if)# interface serial 0/0/1
HQ(config-if)# ip address 10.1.123.1 255.255.255.0
HQ(config-if)# encapsulation frame-relay ietf
HQ(config-if)# no frame-relay inverse-arp
HQ(config-if)# frame-relay map ip 10.1.123.1 102
HQ(config-if)# frame-relay map ip 10.1.123.2 102
HQ(config-if)# frame-relay map ip 10.1.123.3 103
HQ(config-if)# no shutdown
```


Router R2 (Hostname EAST)

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# hostname EAST
EAST(config)# interface loopback 2
EAST(config-if)# ip address 10.1.2.1 255.255.255.0
EAST(config-if)# interface serial 0/0/1
EAST(config-if)# ip address 10.1.123.2 255.255.255.0
EAST(config-if)# clock rate 64000
EAST(config-if)# encapsulation frame-relay ietf
EAST(config-if)# no frame-relay inverse-arp
EAST(config-if)# frame-relay map ip 10.1.123.1 201
EAST(config-if)# frame-relay map ip 10.1.123.2 201
EAST(config-if)# frame-relay map ip 10.1.123.3 201
EAST(config-if)# no shutdown
EAST(config-if)# interface FastEthernet 0/0
EAST(config-if)# ip address 10.1.23.2 255.255.255.0
EAST(config-if)# no shutdown
```

Router R4 (Hostname WEST)

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# hostname WEST
WEST(config)# interface loopback 3
WEST(config-if)# ip address 10.1.3.1 255.255.255.0
WEST(config-if)# interface serial 0/0/0
WEST(config-if)# ip address 10.1.123.3 255.255.255.0
WEST(config-if)# encapsulation frame-relay ietf
WEST(config-if)# no frame-relay inverse-arp
WEST(config-if)# frame-relay map ip 10.1.123.1 301
WEST(config-if)# frame-relay map ip 10.1.123.2 301
WEST(config-if)# frame-relay map ip 10.1.123.3 301
WEST(config-if)# no shutdown
WEST(config-if)# interface FastEthernet 0/0
WEST(config-if)# ip address 10.1.23.3 255.255.255.0
WEST(config-if)# no shutdown
```

b. Verify that you have local subnet connectivity with **ping**.

Step 2: Configure the Frame Relay switch.

Use a fourth Cisco router with three serial interfaces as a Frame Relay switch, and cable the routers according to the diagram. Configure the Frame Relay switch to have the DLCIs indicated in the diagram between HQ and EAST and HQ and WEST.

Router R3 (FRS)

```
hostname FRS
!
frame-relay switching
interface Serial0/0/0
 no ip address
 encapsulation frame-relay ietf
 no ip route-cache
 clock rate 64000
 frame-relay intf-type dce
```



```
 frame-relay route 102 interface Serial0/0/1 201
 frame-relay route 103 interface Serial0/1/0 301
 no shutdown
!
interface Serial0/0/1
 no ip address
 encapsulation frame-relay ietf
 frame-relay intf-type dce
 frame-relay route 201 interface Serial0/0/0 102
 no shutdown
!
interface Serial0/1/0
 no ip address
 encapsulation frame-relay ietf
 no ip route-cache
 frame-relay intf-type dce
 clock rate 64000
 frame-relay route 301 interface Serial0/0/0 103
 no shutdown
```

Step 3: Configure OSPF network type NBMA.

Frame Relay is inherently a non-broadcast multi-access (NBMA) network. In this step, you configure OSPF for hub-and-spoke over Frame Relay using the NBMA OSPF network type, which is the default for Frame Relay physical interfaces and multipoint-type subinterfaces. HQ is the hub; EAST and WEST are the spokes.

a. Create OSPF process 1. Add the Frame Relay interfaces on each router into area 0 with the **network** command. Add the loopback interfaces on each router into area 0, and then change the network type to allow the correct subnet mask to be advertised.

```
HQ(config)# router ospf 1
HQ(config-router)# network 10.1.123.0 0.0.0.255 area 0
HQ(config-router)# network 10.1.1.0 0.0.0.255 area 0
HQ(config-router)# exit
HQ(config)# interface loopback 1
HQ(config-if)# ip ospf network point-to-point

EAST(config)# router ospf 1
EAST(config-router)# network 10.1.123.0 0.0.0.255 area 0
EAST(config-router)# network 10.1.2.0 0.0.0.255 area 0
EAST(config-router)# exit
EAST(config)# interface loopback 2
EAST(config-if)# ip ospf network point-to-point

WEST(config)# router ospf 1
WEST(config-router)# network 10.1.123.0 0.0.0.255 area 0
WEST(config-router)# network 10.1.3.0 0.0.0.255 area 0
WEST(config-router)# exit
WEST(config)# interface loopback 3
WEST(config-if)# ip ospf network point-to-point
```

b. On EAST and WEST, change the Frame Relay interfaces to have OSPF priority 0. This priority ensures that HQ becomes the DR.

```
EAST(config)# interface serial 0/0/1
EAST(config-if)# ip ospf priority 0

WEST(config)# interface serial 0/0/0
WEST(config-if)# ip ospf priority 0
```


c. No OSPF Frame Relay adjacencies will be established yet because the default network type is nonbroadcast. You can change this by adding **neighbor** statements. Configure **neighbor** statements on HQ pointing toward EAST and WEST. Only the router starting the exchange needs the statements (HQ in this case). However, it is considered best practice to also specify HQ as a neighbor on the EAST and WEST routers. Because the hello timers are longer on serial nonbroadcast links, the neighbor adjacencies might take longer to come up.

```
HQ(config)# router ospf 1
HQ(config-router)# neighbor 10.1.123.2
HQ(config-router)# neighbor 10.1.123.3

EAST(config)# router ospf 1
EAST(config-router)# neighbor 10.1.123.1

WEST(config)# router ospf 1
WEST(config-router)# neighbor 10.1.123.1
```

Note: Neighbor commands shown for the EAST and WEST routers will not appear in the running config. This is because the local router's OSPF priority has been set to 0 on the interface which would be used to communicate with the designated neighbor.

d. You can verify adjacency states with the **show ip ospf neighbor** command.

```
HQ# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.2.1          0   FULL/DROTHER    00:01:57    10.1.123.2      Serial0/0/1
10.1.3.1          0   FULL/DROTHER    00:01:57    10.1.123.3      Serial0/0/1
```

Step 4: Change the network type to point-to-multipoint.

Point-to-multipoint is an OSPF network type that lends itself well to a hub-and-spoke topology. Point-to-multipoint does not elect DRs or BDRs, so it does not need interface priorities. Instead, it treats the network as a collection of point-to-point networks and advertises host routes for any neighbors that it has.

a. To configure point-to-multipoint links, remove the previously configured **neighbor** statements and interface priorities..

```
HQ(config)# router ospf 1
HQ(config-router)# no neighbor 10.1.123.2
HQ(config-router)# no neighbor 10.1.123.3

EAST(config)# interface serial 0/0/1
EAST(config-if)# no ip ospf priority 0
EAST(config-if)# exit
EAST(config)# router ospf 1
EAST(config-router)# no neighbor 10.1.123.1

WEST(config)# interface serial 0/0/0
WEST(config-if)# no ip ospf priority 0
WEST(config-if)# exit
WEST(config)# router ospf 1
WEST(config-router)# no neighbor 10.1.123.1
```

b. Use the interface command **ip ospf network point-to-multipoint** and reapply the Frame Relay maps using the **broadcast** option. Reset the OSPF process using the **clear ip ospf process** command. Verify that the adjacencies are active with the **show ip ospf neighbor** command.


```
HQ(config)# interface serial 0/0/1
HQ(config-if)# ip ospf network point-to-multipoint
HQ(config-if)# frame-relay map ip 10.1.123.2 102 broadcast
HQ(config-if)# frame-relay map ip 10.1.123.3 103 broadcast

EAST(config)# interface serial 0/0/1
EAST(config-if)# ip ospf network point-to-multipoint
EAST(config-if)# frame-relay map ip 10.1.123.1 201 broadcast
EAST(config-if)# frame-relay map ip 10.1.123.3 201 broadcast

WEST(config)# interface serial 0/0/0
WEST(config-if)# ip ospf network point-to-multipoint
WEST(config-if)# frame-relay map ip 10.1.123.1 301 broadcast
WEST(config-if)# frame-relay map ip 10.1.123.2 301 broadcast

HQ# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.3.1          0   FULL/  -        00:01:34    10.1.123.3      Serial0/0/1
10.1.2.1          0   FULL/  -        00:01:45    10.1.123.2      Serial0/0/1
```

c. Observe the routing table on one of the spoke routers. Notice how the routing table has host routes in it. This is part of point-to-multipoint behavior.

```
EAST# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
O       10.1.3.0/24 [110/129] via 10.1.123.1, 00:01:07, Serial0/0/1
C       10.1.2.0/24 is directly connected, Loopback2
O       10.1.1.0/24 [110/65] via 10.1.123.1, 00:01:07, Serial0/0/1
C       10.1.23.0/24 is directly connected, FastEthernet0/0
C       10.1.123.0/24 is directly connected, Serial0/0/1
O       10.1.123.1/32 [110/64] via 10.1.123.1, 00:01:07, Serial0/0/1
O       10.1.123.3/32 [110/128] via 10.1.123.1, 00:01:07, Serial0/0/1
```

d. Look at the output of the **show ip ospf interface** *interface* command on your routers. Notice that the interface type is point-to-multipoint.

```
EAST# show ip ospf interface serial 0/0/1
Serial0/0/1 is up, line protocol is up
  Internet Address 10.1.123.2/24, Area 0
  Process ID 1, Router ID 10.1.2.1, Network Type POINT_TO_MULTIPOINT, Cost:
64
  Transmit Delay is 1 sec, State POINT_TO_MULTIPOINT,
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    oob-resync timeout 120
    Hello due in 00:00:16
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
```

```
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.1.1
  Suppress hello for 0 neighbor(s)
```

Step 5: Change OSPF timers.

a. Add the Ethernet link connecting EAST and WEST to the OSPF process using the **network** command.

```
EAST(config)# router ospf 1
EAST(config-router)# network 10.1.23.0 0.0.0.255 area 0

WEST(config)# router ospf 1
WEST(config-router)# network 10.1.23.0 0.0.0.255 area 0
```

b. Look at the interface OSPF properties with the **show ip ospf interface** *interface* command.

```
EAST# show ip ospf interface FastEthernet 0/0
  FastEthernet0/0 is up, line protocol is up
  Internet Address 10.1.23.2/24, Area 0
  Process ID 1, Router ID 10.1.2.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 10.1.3.1, Interface address 10.1.23.3
  Backup Designated router (ID) 10.1.2.1, Interface address 10.1.23.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:00
  Supports Link-local Signaling (LLS)
  Index 3/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.3.1  (Designated Router)
  Suppress hello for 0 neighbor(s)
```

Because it is a Fast Ethernet link, the default network type is broadcast and the default network timers are those associated with a broadcast network. You might want to change the default timers to allow for better network convergence. Neighbors that go down are detected more quickly with lower dead timers. The disadvantage of lower dead timers is higher router CPU utilization and more bandwidth being consumed by hello packets.

c. Use the **ip ospf hello-interval** *seconds* command to change the default hello timer interval to 5 seconds on both sides. Change the dead timer to 15 seconds with the **ip ospf dead-interval** *seconds* command.

```
EAST(config)# interface FastEthernet 0/0
EAST(config-if)# ip ospf hello-interval 5
EAST(config-if)# ip ospf dead-interval 15

WEST(config)# interface FastEthernet 0/0
WEST(config-if)# ip ospf hello-interval 5
WEST(config-if)# ip ospf dead-interval 15
```

d. Verify the changes with the **show ip ospf interface** *interface* command.

```
EAST# show ip ospf int Fa0/0
  FastEthernet0/0 is up, line protocol is up
  Internet Address 10.1.23.2/24, Area 0
```



```
  Process ID 1, Router ID 10.1.2.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 10.1.3.1, Interface address 10.1.23.3
  Backup Designated router (ID) 10.1.2.1, Interface address 10.1.23.2
  Timer intervals configured, Hello 5, Dead 15, Wait 15, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:01
  Supports Link-local Signaling (LLS)
  Index 3/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.3.1  (Designated Router)
  Suppress hello for 0 neighbor(s)
```

What are some disadvantages to changing the timers if they are not tuned correctly?

____

____

____

____

____

____

e. Use the following Tcl script to verify connectivity to all addresses in the topology.

```
HQ# tclsh
HQ(tcl)#

foreach address {
10.1.1.1
10.1.2.1
10.1.3.1
10.1.123.1
10.1.123.2
10.1.123.3
10.1.23.2
10.1.23.3
} {
ping $address }
```

Challenge: Minimal Hello Intervals

Configure the Fast Ethernet link between EAST and WEST to lower the convergence time using the **ip ospf dead-interval minimal hello-multiplier** *multiplie*r command. This command sets the dead interval to 1 second. Hellos will be sent at the rate of *multiplier* per second.

Note: The use of this command overrides hello and dead intervals configured on EAST and WEST Fa0/0 in Step 5.

a. Configure the routers to send five hellos a second.

____

____

Note: Although you could use this command on the serial links, it would create additional OSPF overhead on these relatively slow (64 Kb/s) links and could result in flapping adjacencies if a link begins to experience congestion. Using this command with high-speed serial links would be less of an issue.

b. Look at the dead time column of the **show ip ospf neighbor** command. Is it a different format than before for that connection?

____

c. Display the OSPF information for Fa0/0 on EAST or WEST using the **show ip ospf interface fastEthernet 0/0** command. What are the Hello and Dead intervals now?

____

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Appendix A: Adtran Frame Relay Switch Configuration

If an Adtran Atlas 550 is used for the Frame Relay switch, connect the serial cable from each router interface in the topology diagram to the Adtran interface indicated in the table below. The Adtran is preconfigured to simulate a Frame Relay service that provides the following PVCs.

Frame Relay Switching Configuration

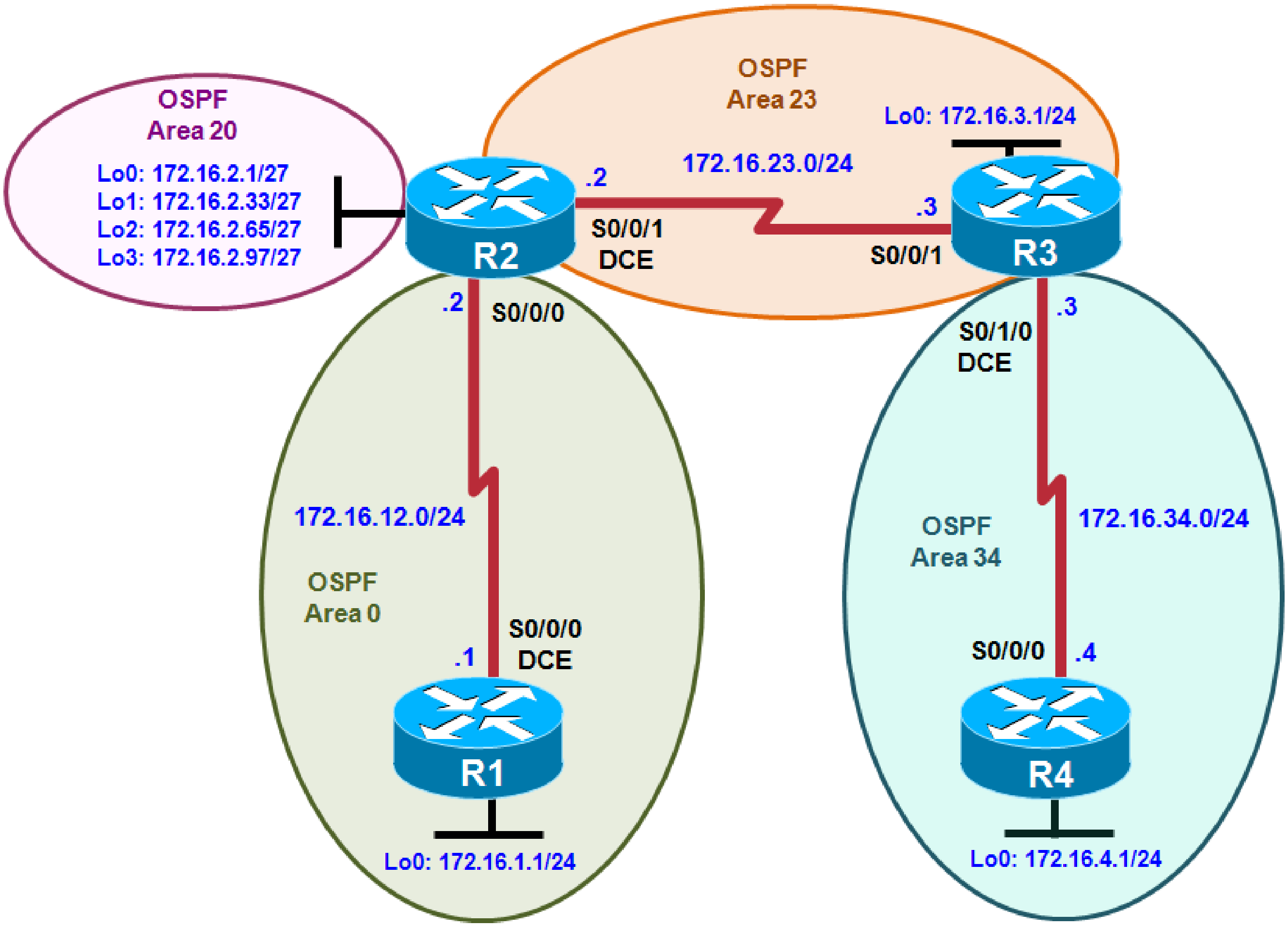
Connected Router	Router Interface	Adtran Interface	Ingress DLCI	Egress DLCI	Egress Router
HQ	S0/0/1 DTE	port 1/1	102	201	EAST
HQ	S0/0/1 DTE	port 1/1	103	301	WEST
EAST	S0/0/1 DTE	port 1/2	201	102	HQ
WEST	S0/0/0 DTE	port 2/1	301	103	HQ

The Adtran Frame Relay switch interfaces all provide the DCE clock. Be sure to use the appropriate cable between each router and the Adtran. All the router interfaces are DTE, and the cable to the Adtran interface should be serial to V.35 DCE. Use the **show controllers** command to verify which cable type is connected to a given router interface.

```
HQ# show controllers s0/0/1
Interface Serial0/0/1
Hardware is GT96K
DTE V.35 TX and RX clocks detected.

<output omitted>
```


Chapter 3 Lab 3-5, OSPF Challenge Lab

Topology



Objectives

- Implement the topology diagram following the instructions in the Configuration Requirements section.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Configuration Requirements

1. Configure the interfaces in the diagram with the IP addresses shown.
2. Configure the bandwidth to reflect the actual bandwidth of all serial links.
3. Configure OSPF with interfaces in the areas shown in the diagram.
4. Configure R2 to summarize area 20 with the most specific mask possible.
5. Make the link between R1 and R2 have the OSPF network type of broadcast, with R1 as the DR.
6. Configure R1 to always originate a default route.
7. Modify the link between R2 and R3 to have hello timers and dead timers that are double the default values.
8. Make the link between R2 and R3 have a cost of 500.
9. Configure area 34 to be a totally stubby area.
10. Use MD5 authentication with the keyword “cisco” over the link between R3 and R4.
11. Figure out the hidden issue in the topology that you need to address to have full connectivity.
12. Run a Tcl script on all routers to verify that there is connectivity between the IP addresses in the topology.

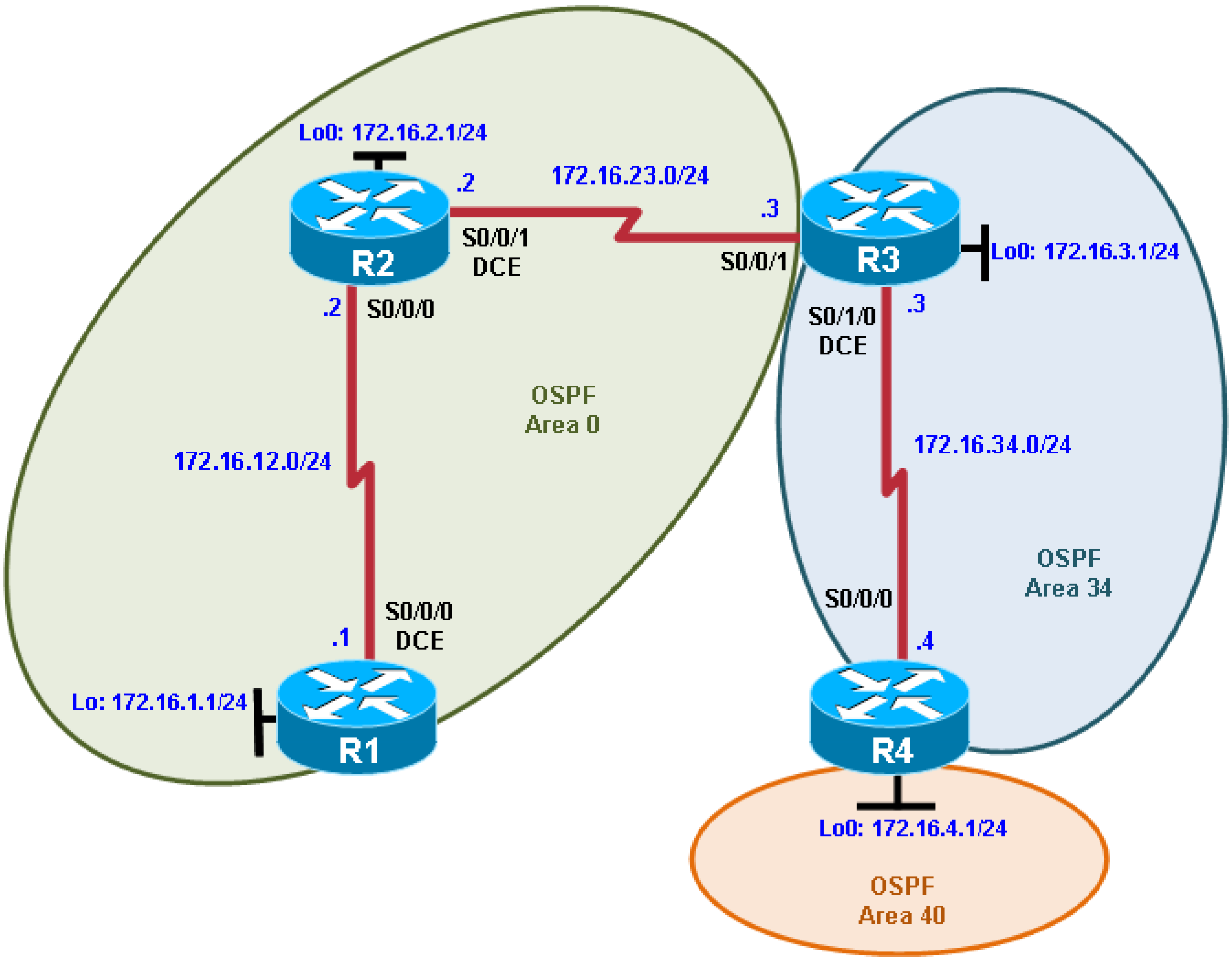
Notes: ____

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 3 Lab 3-6, OSPF Troubleshooting Lab

Topology



Objectives

- Troubleshoot OSPF operation and configuration.

Background

In this lab, you troubleshoot existing configurations to get a working topology. Some of these configurations are correct, and some are intentionally wrong. Your goal is to use troubleshooting techniques to fix anything in the scenario that prevents full IP connectivity. Full IP connectivity means every address in the scenario should be reachable from every router. If you do not know where to start, try pinging remote addresses and see which ones are reachable (either manually performing pings or using a Tcl script).

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Requirements

- Cut and paste the initial configurations from this lab into the respective routers.
- Use the IP addressing scheme shown in the diagram.
- All routers must participate in OSPF.
- All interfaces must be in the OSPF areas shown in the diagram.
- Do not use static routes, default routes, or other routing protocols.
- All IP addresses in the topology must be reachable from all routers.
- The OSPF network type for the link between R2 and R3 is nonbroadcast.

Initial Configurations

Router R1

```
hostname R1
!
interface Loopback0
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0/0/0
 ip address 172.16.12.1 255.255.255.0
 clock rate 64000
 bandwidth 64
 no shutdown
!
router ospf 1
 network 172.16.1.0 0.0.0.255 area 0
 network 172.16.12.2 0.0.0.0 area 0
end
```

Router R2

```
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
 bandwidth 64
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.23.2 255.255.255.0
```

```
 ip ospf network non-broadcast
 clock rate 64000
 bandwidth 64
 no shutdown
!
router ospf 1
 network 172.16.2.0 0.0.0.255 area 0
 network 172.16.12.0 0.0.0.255 area 0
 network 172.16.23.0 0.0.0.255 area 0
end
```

Router R3

```
hostname R3
!
interface Loopback0
 ip address 172.16.3.1 255.255.255.0
!
interface Serial0/0/1
 ip address 172.16.23.3 255.255.255.0
 ip ospf network non-broadcast
 bandwidth 64
 no shutdown
!
interface Serial0/1/0
 ip address 172.16.34.3 255.255.255.0
 clock rate 64000
 bandwidth 64
 no shutdown
!
router ospf 1
 area 34 virtual-link 172.16.4.1
 network 172.16.3.0 0.0.0.255 area 34
 network 172.16.23.0 0.0.0.255 area 0
 network 172.16.34.0 0.0.0.255 area 34
end
```

Router R4

```
hostname R4
!
interface Loopback0
 ip address 172.16.4.1 255.255.255.0
!
interface Serial0/0/0
 ip address 172.16.34.4 255.255.255.0
 bandwidth 64
 no shutdown
!
router ospf 1
 area 34 virtual-link 172.16.34.3
 network 172.16.4.0 0.0.0.255 area 40
 network 172.16.34.0 0.0.0.255 area 34
end
```

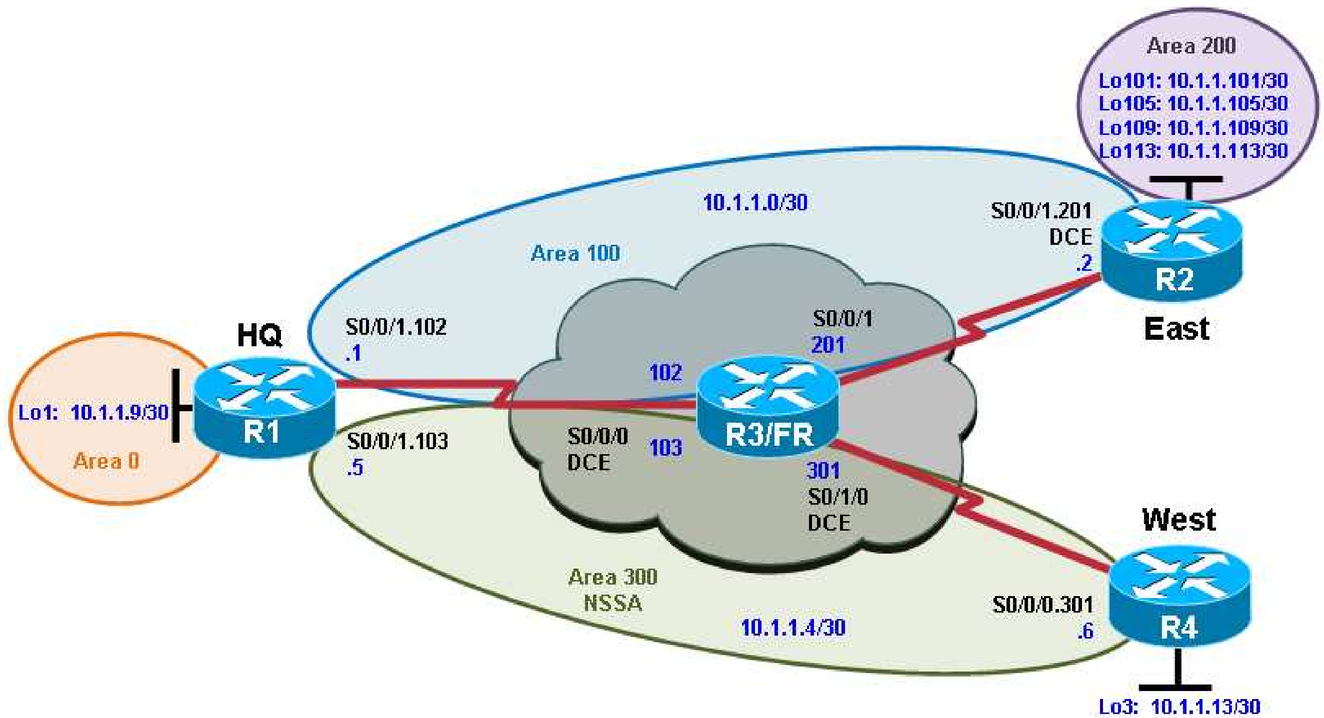

Notes:

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 3 Lab 3-7, OSPF Case Study

Topology



Objectives

- Plan, design, and implement the International Travel Agency network shown in the diagram and described below.
- Verify that all configurations are operational and functioning according to the guidelines.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Service or comparable)
- Serial and console cables

Requirements

The International Travel Agency needs its core network set up for OSPF with the specifications indicated in the diagram and listed below. Design, configure, and test a network that meets all the following requirements:

1. Use the addressing scheme shown in the diagram.
2. Configure OSPF with the networks shown in the diagram.
3. Configure the OSPF backbone area to be on Loopback0 on HQ.
4. Configure the Frame Relay subnets as point-to-point subinterfaces, with the link between HQ and East in area 100, and the link between HQ and West in area 300.
5. Configure area 300 as an NSSA.
6. Configure the router East loopback interfaces to be in area 200. Summarize this area with the most efficient summary.
7. Redistribute the loopback network on router West into OSPF.
8. Create virtual links as necessary for full connectivity.
9. Make sure that all loopback interfaces are advertised with the correct subnet mask.

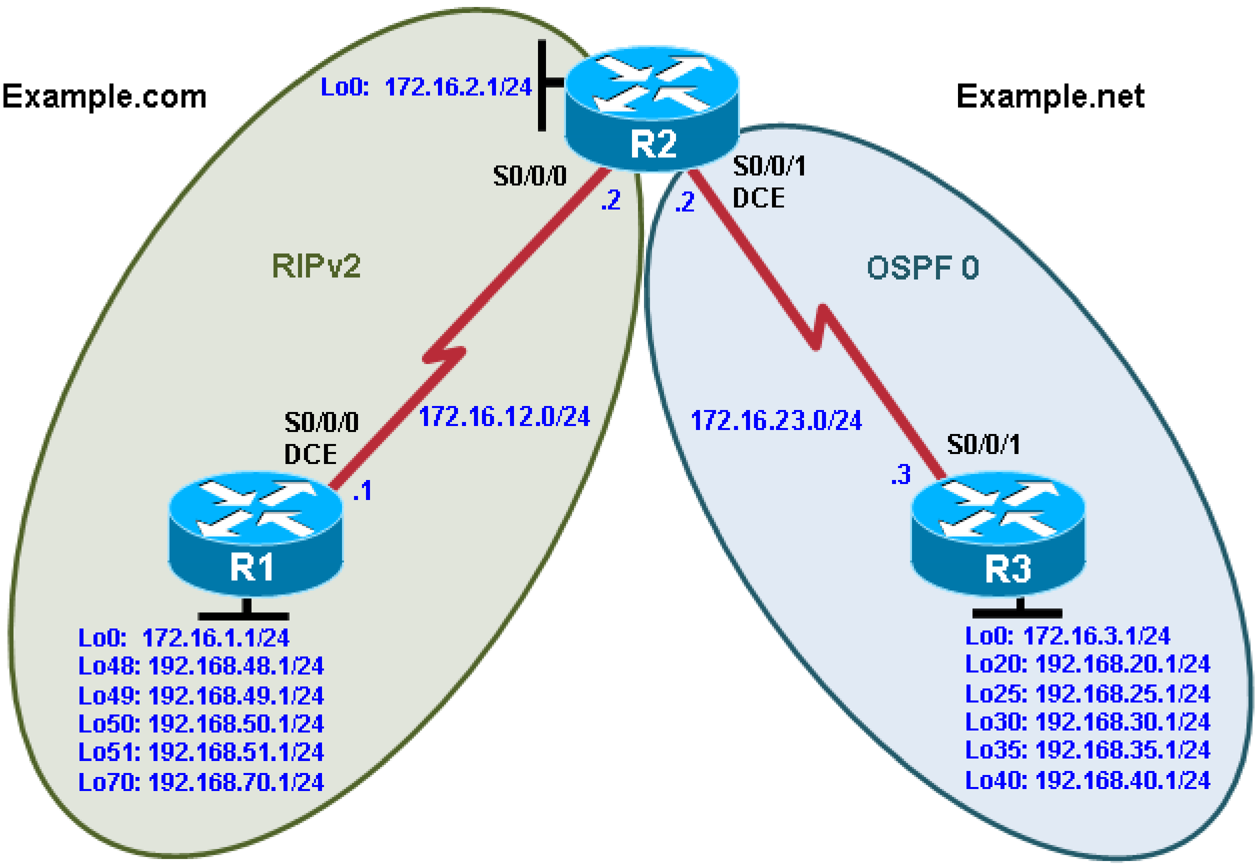
Notes:

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 4 Lab 4-1, Redistribution Between RIP and OSPF

Topology



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.

Background

Two online booksellers, Example.com and Example.net, have merged and now need a short-term solution to inter-domain 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. Because 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 shown 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 be a 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.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Configure loopbacks and assign addresses.

a. 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)# clock rate 64000
R1(config-if)# bandwidth 64
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)# bandwidth 64
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 64000
R2(config-if)# bandwidth 64
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)# bandwidth 64
R3(config-if)# no shutdown
```

b. (Optional) On each router, create an enable secret password. Configure the console line for synchronous logging and no timeout. Configure the vty lines to allow Telnet to and remote configuration of network devices.

R1 example:

```
R1(config)# enable secret cisco

R1(config)# line con 0
R1(config-line)# logging synchronous
R1(config-line)# exec-timeout 0 0

R1(config)# line vty 0 4
R1(config-line)# password cisco
R1(config-line)# login
```

c. Verify that you can ping across the serial links when you are finished. Use the following Tcl script to check full and partial connectivity throughout this lab.

```
R1# tclsh

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 that you should receive back are those of the connected networks of 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 RIPv2 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, the RIP **network** command only requires the classful network address to be entered and does not support a wildcard mask. This behavior is inherited from the classful RIPv1 protocol configuration and is kept for backward compatibility with older Cisco IOS versions that would not otherwise be able to process **network** commands with wildcard masks. Classful protocols do not support subnets; therefore, subnet or wildcard masks are unnecessary.

Based on the topology diagram, which major networks need to be advertised into RIP for R1?

____

____

____

____

____

____

Which major networks need to be advertised into RIP for R2?

____

____

a. 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
```

b. Verify that the RIP routes were learned from the other routers using the **show ip route rip** command on each router.

```
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
```

c. 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.

a. On R1, use the **show ip route rip** command to view the RIP routes in the routing table. Notice that the network for the serial interface of R2 that connects to R3 is present, even though you do not have a RIP neighbor on that interface. This is because the entire class B network 172.16.0.0 /16 was added to RIP on R2.

```
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
```

b. Issue the **show ip protocols** command to verify that RIPv2 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)
```

For security reasons and to reduce unnecessary traffic, RIP updates should not be propagated into the OSPF domain. You can disable sending updates with the **passive-interface** *interface_type interface_number* router configuration command

c. On R2, configure the serial interface connecting to R3 as passive. Notice that the interface is no longer listed in the output of the **show ip protocols** command.

```
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)
```

d. On R1, issue the **show ip route rip** command. Notice that the 172.16.23.0 network is still in the routing table and being sourced 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 routes being received through it.

What are some reasons to prevent RIP from 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?

____

____

____

____

e. If you are unsure, monitor the output of the **debug ip rip** command to verify your answer. On R1 and R2, configure all loopbacks from which RIPv2 is sending advertisements in passive state with the **passive-interface** command.

```
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
```

When running RIPv2, implement passive interfaces as a common practice to save CPU processor cycles and bandwidth on interfaces that do not have multicast RIPv2 neighbors.

Note: An alternative to making each loopback interface on R1 passive is to make all interfaces passive with the **passive-interface default** command in router configuration mode. Then make any interfaces that need to send updates, such as S0/0/0, nonpassive.

```
R1(config)# router rip
R1(config-router)# passive-interface default
R1(config-router)# no passive-interface Serial0/0/0
```

Step 4: Summarize a supernet with RIP.

a. On R2, issue the **show ip route rip** command. Notice that you can see all prefixes from R1 in the R2 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 the R1 serial 0/0/0 interface. However, the RIP implementation in the Cisco IOS Software does not allow summarizing to a mask length that is less than the classful network prefix (in this case, 24 bits). This limitation does not affect other routing protocols. If you do try, you receive the following error message:

```
R1(config)# interface serial 0/0/0
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 being the Null0 interface. You can create an entry manually using the **ip route** command and redistribute it into RIP, thereby emulating the approach of EIGRP to a certain extent.

b. 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 for RIPv2, it resembles many effects of summarization as performed in other routing protocols like EIGRP or OSPF. Again, this is not a limitation of RIPv2, but rather a Cisco IOS implementation issue.

c. On R1 and R2, verify that the RIP supernet has been added to the routing table with the **show ip route** command.

```
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 might not want to advertise certain networks out a particular interface, or you might want to filter updates as they come in. This is possible with distance-vector 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.

Distribute lists can be used with either access lists or prefix lists to filter routes by network address. With prefix lists, they can also be configured to filter routes by subnet masks.

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

The 22-bit summary and the 24-bit major network address both have the same address, so access lists will not accomplish the filtering correctly. Therefore, it is necessary to use prefix lists.

To create a prefix list or add a prefix list entry, use the **ip prefix-list** command in global configuration mode.

```
ip prefix-list {list-name | list-number} {deny network/length | permit
network/length} [ge ge-length] [le le-length]
```

The **ge** keyword represents the “greater than or equal to” operator. The **le** keyword represents the “less than or equal to” operator. If both the **ge** and **le** keywords are omitted, the prefix list is processed using an exact match.

a. On R1, use a prefix list as a distribution filter to prevent the more specific routes to loopbacks 48 through 51 from being advertised. Allow all other destination networks, including the summary route.

```
R1(config)# ip prefix-list RIP-OUT permit 192.168.48.0/22
R1(config)# ip prefix-list RIP-OUT deny 192.168.48.0/22 le 24
R1(config)# 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 the 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.

b. From the RIP configuration prompt on R1, apply this access list with the **distribute-list** command.

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

c. On R2, verify that the filtering has taken place using the **show ip route rip** and **show ip rip database** commands.

```
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
```

Note: You might need to issue the **clear ip route *** command on R2 to see the removal of the more specific R1 prefixes. Also, if the network 192.168.48.0/22 does not appear on R2, this is incorrect behavior and might be corrected in recent versions of Cisco IOS Software. A workaround is to remove the **network 192.168.48.0** command from RIP and issue the **clear ip route *** command on R1.

```
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 being sent out or coming in?

____

____

____

____

____

____

Step 6: Configure OSPF.

a. 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.

```
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

15:01:37.047: %OSPF-5-ADJCHG: Process 1, Nbr 172.16.2.1 on Serial0/0/1 from
LOADING to FULL, Loading Done
```

b. On R3, change the network type for the loopback interfaces to point-to-point so that they are advertised with the correct subnet mask (/24 instead of /32).

```
R3(config)# interface Loopback0
R3(config-if)# ip ospf network point-to-point
R3(config-if)# interface Loopback20
R3(config-if)# ip ospf network point-to-point
R3(config-if)# interface Loopback25
R3(config-if)# ip ospf network point-to-point
R3(config-if)# interface Loopback30
R3(config-if)# ip ospf network point-to-point
R3(config-if)# interface Loopback35
R3(config-if)# ip ospf network point-to-point
R3(config-if)# interface Loopback40
R3(config-if)# ip ospf network point-to-point
```

c. Verify the OSPF adjacencies on R2 and R3 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# 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
O    192.168.30.0/24 [110/1563] via 172.16.23.3, 00:01:23, Serial0/0/1
O    192.168.25.0/24 [110/1563] via 172.16.23.3, 00:01:23, Serial0/0/1
O    192.168.40.0/24 [110/1563] via 172.16.23.3, 00:01:23, Serial0/0/1
     172.16.0.0/24 is subnetted, 5 subnets
O       172.16.3.0 [110/1563] via 172.16.23.3, 00:01:23, Serial0/0/1
O    192.168.20.0/24 [110/1563] via 172.16.23.3, 00:01:23, Serial0/0/1
O    192.168.35.0/24 [110/1563] via 172.16.23.3, 00:01:23, Serial0/0/1
```

```
R3# show ip route ospf

R3#
```

Note that output of the **show ip route ospf** command on R3 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 does not. However, it does advertise any connected subnets in that 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.

Passive interfaces save CPU cycles, router memory, and link bandwidth by preventing broadcast and 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 or processing OSPF packets on that interface.

OSPF included the R3 loopback interfaces in its **network** statements shown in Step 6.

a. On R3, configure Loopback0 as a passive interface in OSPF. At the OSPF router 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?

____

____

____

____

____

____

b. Cisco IOS Software 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
```

c. 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
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.2.1           110      00:03:04
  Distance: (default is 110)
```

Step 8: Allow one-way redistribution.

a. 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
```

b. 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>
```

c. On R1, look at the routing table with the **show ip route rip** command. It has all the routes in the topology.

```
R1# show ip route rip
R       192.168.30.0 [120/4] via 172.16.12.2, 00:00:11, Serial0/0/0
R       192.168.25.0 [120/4] via 172.16.12.2, 00:00:11, Serial0/0/0
R       192.168.40.0 [120/4] via 172.16.12.2, 00:00:11, Serial0/0/0
     172.16.0.0/24 is subnetted, 5 subnets
R       172.16.23.0 [120/1] via 172.16.12.2, 00:00:11, Serial0/0/0
R       172.16.2.0 [120/1] via 172.16.12.2, 00:00:11, Serial0/0/0
R       172.16.3.0 [120/4] via 172.16.12.2, 00:00:11, Serial0/0/0
R       192.168.20.0 [120/4] via 172.16.12.2, 00:00:11, Serial0/0/0
R       192.168.35.0 [120/4] via 172.16.12.2, 00:00:11, Serial0/0/0
```

d. On R1, ping a loopback on R3. Notice that it shows that R1 has a route to R3, but R3 does not have a route back to R1.

```
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)
```

e. On R1, verify that R3 does not have a route back with the **traceroute** command.

```
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  *   *   *

<output omitted>
```

To address 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.

f. From the OSPF configuration prompt, issue the **default-information originate always** command to force R2 to advertise a default route in OSPF.

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

g. Verify that this route is present in the R3 routing table.

```
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.

h. Use the Tcl script from Step 1 to verify full connectivity.

Step 9: Redistribute between two routing protocols.

You can substitute this default route with actual, more specific routes.

a. On R2, under the OSPF router configuration prompt, remove the default route advertisement with the **no default-information originate always** command. 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
```

b. If you display the routing table on R3, the external OSPF routes that were added are the 192.168.70.0/24 and 192.168.48.0/22 networks.

```
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 and supernets when redistributing into it. The only classful network coming into R2 from RIP is the class C network 192.168.70.0, and the only supernet is the 192.168.48.0/22.

c. You can modify this behavior by adding the **subnets** keyword to the **redistribute** command.

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

d. On R3, verify the configuration with the **show ip route ospf** command.

```
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.

e. Run the Tcl script on each router to verify full connectivity.

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. A seed metric is a protocol-independent feature of the Cisco IOS Software that is usually configured when redistributing into distance-vector protocols.

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

```
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 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.

a. On R2, in OSPF configuration mode, issue the **default-metric** *metric* command to configure a default metric for redistributed routes. The default metric for all OSPF redistributed routes is 20, except for BGP, which is 1. Setting the metric for RIP to a higher number makes it less preferable to routes redistributed from other routing protocols.

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

b. Verify the new metric in the R3 routing table. It might take some time for the new metric to propagate.

```
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.

Look at the R3 routing table. Notice that the external (redistributed) routes have O E2 as their type. In the output, O means OSPF, and E2 means external, type 2. OSPF has 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. Also notice that the metric is exactly the same as the seed metric in the previous step.

Where would an external type 1 metric be useful?

____

____

____

____

Where would an external type 2 metric be useful?

____

____

____

a. You can change the external type using the **metric-type** argument with the **redistribute** command. Change the type to E1 for RIP redistributed routes.

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


b. Display the R3 routing table again.

```
R3# show ip route ospf
     172.16.0.0/24 is subnetted, 5 subnets
O E1    172.16.12.0 [110/11562] via 172.16.23.2, 00:03:05, Serial0/0/1
O E1    172.16.1.0 [110/11562] via 172.16.23.2, 00:03:05, Serial0/0/1
O E1    172.16.2.0 [110/11562] via 172.16.23.2, 00:03:05, Serial0/0/1
O E1 192.168.70.0/24 [110/11562] via 172.16.23.2, 00:03:05, Serial0/0/1
O E1 192.168.48.0/22 [110/11562] via 172.16.23.2, 00:03:05, Serial0/0/1
```

Which attributes of the routes changed?

____

____

____

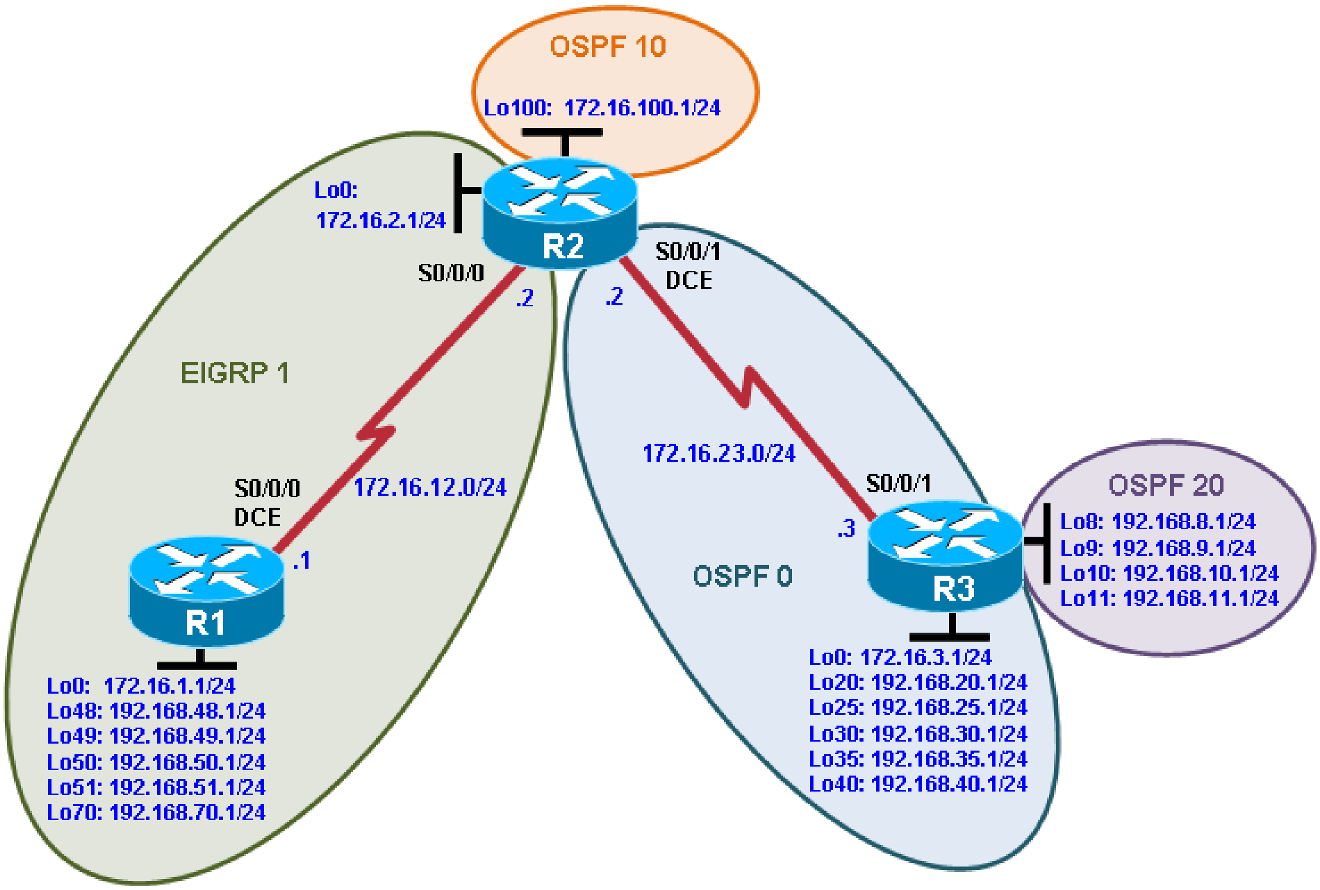
Note: Be sure to save your final configurations through Step 11 for use in Lab 4-2, "Redistribution Between EIGRP and OSPF."

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 4 Lab 4-2, Redistribution Between EIGRP and OSPF

Topology



Objectives

- Review EIGRP and OSPF configuration.
- Redistribute into EIGRP.
- Redistribute into OSPF.
- Summarize routes in EIGRP.
- Filter routes using route maps.
- Modify EIGRP distances.
- Modify OSPF distances.
- Create passive interfaces in EIGRP.
- Summarize in OSPF at an ABR and an ASBR.

Background

R1 is running EIGRP, and R3 is running multi-area OSPF. In this lab, you configure redistribution on R2 to enable these two routing protocols to interact, allowing full connectivity between all networks. In Appendix A of this lab, you explore black hole operation.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Configure loopbacks and additional addressing.

a. Start with the final configurations of Lab 4.1, "Redistribution Between RIP and OSPF." On R1 and R2, remove the RIPv2 configuration and the static route with the following commands.

```
R1(config)# no router rip
R1(config)# no ip route 192.168.48.0 255.255.252.0 null0
R1(config)# no ip prefix-list RIP-OUT

R2(config)# no router rip
R2(config)# router ospf 1
R2(config-router)# no default-information originate
R2(config-router)# no redistribute rip
R2(config-router)# no default-metric 10000
```

b. Configure the additional loopback interfaces on R2 and R3, as shown in the diagram.

```
R2(config)# interface loopback 100
R2(config-if)# ip address 172.16.100.1 255.255.255.0

R3(config)# interface loopback 8
R3(config-if)# ip address 192.168.8.1 255.255.255.0
R3(config-if)# interface loopback 9
R3(config-if)# ip address 192.168.9.1 255.255.255.0
R3(config-if)# interface loopback 10
R3(config-if)# ip address 192.168.10.1 255.255.255.0
R3(config-if)# interface loopback 11
R3(config-if)# ip address 192.168.11.1 255.255.255.0
```

Step 2: Configure EIGRP.

a. Configure R1 and R2 to run EIGRP in autonomous system 1. On R1, add in all connected interfaces either with classful **network** commands or with wildcard masks. Use a classful **network** statement on R2 and disable automatic summarization.

```
R1(config)# router eigrp 1
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
```

or

```
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# network 172.16.0.0
R1(config-router)# network 192.168.0.0 0.0.255.255

R2(config)# router eigrp 1
R2(config-router)# no auto-summary
R2(config-router)# network 172.16.0.0
```

b. Verify the configuration with the **show ip eigrp neighbors** and **show ip route eigrp** commands on both routers.

```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   172.16.12.2             Se0/0/0           11 00:00:30   36   216  0   3

R1# show ip route eigrp
     172.16.0.0/24 is subnetted, 5 subnets
D       172.16.23.0 [90/41024000] via 172.16.12.2, 00:01:38, Serial0/0/0
D       172.16.2.0 [90/40640000] via 172.16.12.2, 00:01:16, Serial0/0/0
D       172.16.100.0 [90/40640000] via 172.16.12.2, 00:02:13, Serial0/0/0

R2# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   172.16.12.1             Se0/0/0           11 00:01:53 1604  5000  0   2

R2# show ip route eigrp
     172.16.0.0/24 is subnetted, 6 subnets
D       172.16.1.0 [90/40640000] via 172.16.12.1, 00:01:08, Serial0/0/0
D    192.168.70.0/24 [90/40640000] via 172.16.12.1, 00:01:08, Serial0/0/0
D    192.168.51.0/24 [90/40640000] via 172.16.12.1, 00:01:08, Serial0/0/0
D    192.168.50.0/24 [90/40640000] via 172.16.12.1, 00:01:08, Serial0/0/0
D    192.168.49.0/24 [90/40640000] via 172.16.12.1, 00:01:08, Serial0/0/0
D    192.168.48.0/24 [90/40640000] via 172.16.12.1, 00:01:08, Serial0/0/0
```

Step 3: Create passive interfaces in EIGRP.

a. Issue the **show ip eigrp interfaces** command on R2.

```
R2# show ip eigrp interfaces
IP-EIGRP interfaces for process 1

                        Xmit Queue   Mean   Pacing Time   Multicast
Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer    Routes
Se0/0/0            1        0/0       32       10/380         496           0
Se0/0/1            0        0/0        0        0/1             0           0
Lo0                0        0/0        0        0/1             0           0
Lo100              0        0/0        0        0/1             0           0
```


Because you used the classful **network** command, both serial interfaces are involved with EIGRP.

b. To stop EIGRP from sending or processing received EIGRP packets on the serial interface going to R3, use the **passive-interface** *interface_type interface_number* command.

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

c. Verify the change with the **show ip eigrp interfaces** and **show ip protocols** commands.

```
R2# show ip eigrp interfaces
IP-EIGRP interfaces for process 1

                        Xmit Queue   Mean   Pacing Time   Multicast
Pending
Interface        Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Se0/0/0            1        0/0       32      10/380         496            0
Lo0                0        0/0        0       0/1             0            0
Lo100              0        0/0        0       0/1             0            0

R2# show ip protocols
Routing Protocol is "ospf 1"
<output omitted>

Routing Protocol is "eigrp 1"
  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 1
  EIGRP NSF-aware route hold timer is 240s
  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            90      00:27:57
  Distance: internal 90 external 170
```

How does preventing hello packets out of an interface affect the update capabilities of EIGRP out that interface?

____

____

____

____

Is this behavior more like RIP or like OSPF in regard to the **passive-interface** command?

____

____

____

Step 4: Manually summarize with EIGRP.

You can have EIGRP summarize routes sent out an interface to make routing updates more efficient by using the **ip summary-address eigrp** *as network mask* command.

a. Have R1 advertise one supernet for loopbacks 48 and 49 to R2. Do not summarize loopbacks 50 and 51 in this statement, because these will be summarized in Step 9.

```
R1(config)# interface serial 0/0/0
R1(config-if)# ip summary-address eigrp 1 192.168.48.0 255.255.254.0

*Feb 23 18:20:21.219: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.12.2
(Serial0/0/0) is resync: summary configured

R1# show ip route eigrp
     172.16.0.0/24 is subnetted, 6 subnets
D       172.16.23.0 [90/41024000] via 172.16.12.2, 00:45:21, Serial0/0/0
D       172.16.2.0 [90/40640000] via 172.16.12.2, 00:45:21, Serial0/0/0
D       172.16.100.0 [90/40640000] via 172.16.12.2, 00:08:12, Serial0/0/0
D    192.168.48.0/23 is a summary, 04:27:07, Null0
```

b. Verify the configuration with the **show ip route eigrp** and **show ip route 192.168.48.0 255.255.254.0** commands on R1. Notice the administrative distance for this route.

```
R1# show ip route 192.168.48.0 255.255.254.0
Routing entry for 192.168.48.0/23, supernet
  Known via "eigrp 1", distance 5, metric 128256, type internal
  Redistributing via eigrp 1
  Routing Descriptor Blocks:
  * directly connected, via Null0
      Route metric is 128256, traffic share count is 1
      Total delay is 5000 microseconds, minimum bandwidth is 10000000 Kbit
      Reliability 255/255, minimum MTU 1514 bytes
      Loading 1/255, Hops 0
```

Why does EIGRP make the administrative distance different for summary routes?

____

____

____

____

Step 5: Additional OSPF configuration.

OSPF is already partially configured on R2 and R3.

a. You need to add the area 10 configuration to R2 and the area 20 configuration to R3 to complete the configuration.

```
R2(config)# router ospf 1
R2(config-router)# network 172.16.100.0 0.0.0.255 area 10
```



```
R3(config)# router ospf 1
R3(config-router)# network 192.168.8.0 0.0.3.255 area 20
```

b. Verify that your adjacencies come up with the **show ip ospf neighbor** command, and make sure that you have routes from OSPF populating the R2 routing table using the **show ip route ospf** command.

```
R2# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
192.168.40.1      0   FULL/  -        00:00:35    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/  -        00:00:35    172.16.23.2     Serial0/0/1

R2# show ip route ospf
O    192.168.30.0/24 [110/1563] via 172.16.23.3, 00:12:10, Serial0/0/1
     192.168.8.0/32 is subnetted, 1 subnets
O IA    192.168.8.1 [110/1563] via 172.16.23.3, 00:20:48, Serial0/0/1
O    192.168.25.0/24 [110/1563] via 172.16.23.3, 00:12:10, Serial0/0/1
     192.168.9.0/32 is subnetted, 1 subnets
O IA    192.168.9.1 [110/1563] via 172.16.23.3, 00:20:48, Serial0/0/1
     192.168.10.0/32 is subnetted, 1 subnets
O IA    192.168.10.1 [110/1563] via 172.16.23.3, 00:20:48, Serial0/0/1
O    192.168.40.0/24 [110/1563] via 172.16.23.3, 00:12:10, Serial0/0/1
     172.16.0.0/24 is subnetted, 6 subnets
O       172.16.3.0 [110/1563] via 172.16.23.3, 00:21:26, Serial0/0/1
     192.168.11.0/32 is subnetted, 1 subnets
O IA    192.168.11.1 [110/1563] via 172.16.23.3, 00:20:48, Serial0/0/1
O    192.168.20.0/24 [110/1563] via 172.16.23.3, 00:12:10, Serial0/0/1
O    192.168.35.0/24 [110/1563] via 172.16.23.3, 00:12:10, Serial0/0/1

R3# show ip route ospf
     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
O IA    172.16.100.1/32 [110/1563] via 172.16.23.2, 00:00:15, Serial0/0/1
```

Notice that for the newly added loopback interfaces, OSPF advertised /32 destination prefixes (for example, R2 has a route to 192.168.8.1/32 in its routing table).

c. Override this default behavior by using the **ip ospf network point-to-point** command on the OSPF loopback interfaces on R2 and R3. You can copy and paste the following configurations to save time.

Router R2:

```
interface loopback 100
 ip ospf network point-to-point
```

Router R3:
(Only configure the point-to-point network type for the newly added loopbacks in area 20. The area 0 loopbacks were configured in Lab 4-1.)

```
interface loopback 8
 ip ospf network point-to-point

interface loopback 9
 ip ospf network point-to-point

interface loopback 10
```



```
 ip ospf network point-to-point

interface loopback 11
 ip ospf network point-to-point
```

Note: You can also use the **interface range** command to configure multiple interfaces simultaneously, as shown below.

```
interface range lo 8 - 11
 ip ospf network point-to-point
```

d. Verify the configuration with the **show ip route** command on R2. Notice that the routes now each show on one line with the /24 major network mask.

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

Gateway of last resort is not set

O    192.168.30.0/24 [110/1563] via 172.16.23.3, 00:27:11, Serial0/0/1
O IA 192.168.8.0/24 [110/1563] via 172.16.23.3, 00:08:39, Serial0/0/1
O    192.168.25.0/24 [110/1563] via 172.16.23.3, 00:27:11, Serial0/0/1
O IA 192.168.9.0/24 [110/1563] via 172.16.23.3, 00:08:39, Serial0/0/1
O IA 192.168.10.0/24 [110/1563] via 172.16.23.3, 00:08:39, Serial0/0/1
O    192.168.40.0/24 [110/1563] via 172.16.23.3, 00:27:11, Serial0/0/1
     172.16.0.0/24 is subnetted, 6 subnets
C       172.16.23.0 is directly connected, Serial0/0/1
C       172.16.12.0 is directly connected, Serial0/0/0
D       172.16.1.0 [90/40640000] via 172.16.12.1, 00:47:33, Serial0/0/0
C       172.16.2.0 is directly connected, Loopback0
O       172.16.3.0 [110/1563] via 172.16.23.3, 00:36:27, Serial0/0/1
C       172.16.100.0 is directly connected, Loopback100
O IA 192.168.11.0/24 [110/1563] via 172.16.23.3, 00:08:41, Serial0/0/1
O    192.168.20.0/24 [110/1563] via 172.16.23.3, 00:27:13, Serial0/0/1
D    192.168.51.0/24 [90/40640000] via 172.16.12.1, 00:47:36, Serial0/0/0
D    192.168.50.0/24 [90/40640000] via 172.16.12.1, 00:47:36, Serial0/0/0
O    192.168.35.0/24 [110/1563] via 172.16.23.3, 00:27:13, Serial0/0/1
D    192.168.70.0/24 [90/40640000] via 172.16.12.1, 00:47:36, Serial0/0/0
D    192.168.48.0/23 [90/40640000] via 172.16.12.1, 00:40:01, Serial0/0/0
```

Notice that R2 is the only router with knowledge of all routes in the topology at this point, because it is involved with both routing protocols.

Step 6: Summarize OSPF areas at the ABR.

Review the R2 routing table. Notice the inter-area routes for the R3 loopbacks in area 20.

a. Summarize the areas into a single inter-area route using the **area** *area* **range** *network mask* command on R3.

```
R3(config)# router ospf 1
R3(config-router)# area 20 range 192.168.8.0 255.255.252.0
```

b. On R2, verify the summarization with the **show ip route ospf** command on R2.

```
R2# show ip route ospf
O    192.168.30.0/24 [110/1563] via 172.16.23.3, 02:38:46, Serial0/0/1
O    192.168.25.0/24 [110/1563] via 172.16.23.3, 02:38:46, Serial0/0/1
O    192.168.40.0/24 [110/1563] via 172.16.23.3, 02:38:46, Serial0/0/1
     172.16.0.0/24 is subnetted, 6 subnets
O       172.16.3.0 [110/1563] via 172.16.23.3, 02:38:46, Serial0/0/1
```



```
O    192.168.20.0/24 [110/1563] via 172.16.23.3, 02:38:46, Serial0/0/1
O    192.168.35.0/24 [110/1563] via 172.16.23.3, 02:38:46, Serial0/0/1
O IA 192.168.8.0/22 [110/1563] via 172.16.23.3, 00:00:07, Serial0/0/1
```

Where can you summarize in OSPF?

____

____

____

Compare and contrast OSPF and EIGRP in terms of where summarization takes place.

____

____

____

Explain the synchronization requirement in OSPF that eliminates other routers as points of summarization.

____

____

____

____

____

Why or why not does EIGRP have this requirement?

____

____

____

____

____

____

Step 7: Configure mutual redistribution between OSPF and EIGRP.

a. Under the OSPF process on R2, issue the **redistribute eigrp 1 subnets** command. The **subnets** command is necessary because, by default, OSPF only redistributes classful networks and supernets. A default seed metric is not required for OSPF. Under the EIGRP process, issue the **redistribute ospf 1 metric 10000 100 255 1 1500** command, which tells EIGRP to redistribute OSPF process 1 with these metrics: bandwidth of 10000, delay of 100, reliability of 255/255, load of 1/255, and a MTU of 1500. Like RIP, EIGRP requires a seed metric. You can also set a default seed metric with the **default-metric** command.

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

R2(config)# router eigrp 1
R2(config-router)# redistribute ospf 1 metric 10000 100 255 1 1500
```


or

```
R2(config-router)# default-metric 10000 100 255 1 1500
R2(config-router)# redistribute ospf 1
```

b. Issue the **show ip protocols** command on the redistributing router, R2. Compare your output with the following output.

```
R2# 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 172.16.2.1
  It is an area border and autonomous system boundary router
  Redistributing External Routes from,
    eigrp 1, includes subnets in redistribution
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    172.16.23.0 0.0.0.255 area 0
    172.16.100.0 0.0.0.255 area 10
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.40.1         110      00:00:33
  Distance: (default is 110)

Routing Protocol is "eigrp 1"
  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: ospf 1, eigrp 1
  EIGRP NSF-aware route hold timer is 240s
  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           90      00:00:49
  Distance: internal 90 external 170
```

c. Display the routing tables on R1 and R3 so that you can see the redistributed routes. Redistributed OSPF routes display on R1 as D EX, which means that they are external EIGRP routes. Redistributed EIGRP routes are tagged in the R3 routing table as O E2, which means that they are OSPF external type 2. Type 2 is the default OSPF external type.

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

Gateway of last resort is not set
```

```
D EX 192.168.30.0/24 [170/40537600] via 172.16.12.2, 00:00:05, Serial0/0/0
D EX 192.168.25.0/24 [170/40537600] via 172.16.12.2, 00:00:05, Serial0/0/0
D EX 192.168.40.0/24 [170/40537600] via 172.16.12.2, 00:00:05, Serial0/0/0
     172.16.0.0/24 is subnetted, 6 subnets
D       172.16.100.0 [90/40640000] via 172.16.12.2, 00:38:02, Serial0/0/0
D       172.16.23.0 [90/41024000] via 172.16.12.2, 00:38:02, Serial0/0/0
C       172.16.12.0 is directly connected, Serial0/0/0
C       172.16.1.0 is directly connected, Loopback0
D       172.16.2.0 [90/40640000] via 172.16.12.2, 00:38:02, Serial0/0/0
D EX    172.16.3.0 [170/40537600] via 172.16.12.2, 00:00:06, Serial0/0/0
D EX 192.168.20.0/24 [170/40537600] via 172.16.12.2, 00:00:06, Serial0/0/0
C    192.168.51.0/24 is directly connected, Loopback51
C    192.168.50.0/24 is directly connected, Loopback50
D EX 192.168.35.0/24 [170/40537600] via 172.16.12.2, 00:00:06, Serial0/0/0
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
D EX 192.168.8.0/22 [170/40537600] via 172.16.12.2, 00:00:07, Serial0/0/0
D    192.168.48.0/23 is a summary, 04:19:50, Null0

R3# show ip route
<output omitted>

Gateway of last resort is not set

C    192.168.30.0/24 is directly connected, Loopback30
C    192.168.8.0/24 is directly connected, Loopback8
C    192.168.25.0/24 is directly connected, Loopback25
C    192.168.9.0/24 is directly connected, Loopback9
C    192.168.10.0/24 is directly connected, Loopback10
C    192.168.40.0/24 is directly connected, Loopback40
     172.16.0.0/24 is subnetted, 6 subnets
C       172.16.23.0 is directly connected, Serial0/0/1
O E2    172.16.12.0 [110/20] via 172.16.23.2, 00:41:48, Serial0/0/1
O E2    172.16.1.0 [110/20] via 172.16.23.2, 00:41:48, Serial0/0/1
O E2    172.16.2.0 [110/20] via 172.16.23.2, 00:41:48, Serial0/0/1
O IA    172.16.100.0 [110/1563] via 172.16.23.2, 00:41:48, Serial0/0/1
C       172.16.3.0 is directly connected, Loopback0
C    192.168.11.0/24 is directly connected, Loopback11
C    192.168.20.0/24 is directly connected, Loopback20
O E2 192.168.51.0/24 [110/20] via 172.16.23.2, 00:41:48, Serial0/0/1
O E2 192.168.50.0/24 [110/20] via 172.16.23.2, 00:41:48, Serial0/0/1
C    192.168.35.0/24 is directly connected, Loopback35
O E2 192.168.70.0/24 [110/20] via 172.16.23.2, 00:41:48, Serial0/0/1
O    192.168.8.0/22 is a summary, 01:34:48, Null0
O E2 192.168.48.0/23 [110/20] via 172.16.23.2, 00:41:48, Serial0/0/1
```

d. Verify full connectivity with the following Tcl script:

```
R1# tclsh

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
172.16.100.1
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
192.168.8.1
192.168.9.1
192.168.10.1
192.168.11.1
172.16.23.3
} { ping $address }
```

Step 8: Filter redistribution with route maps.

One way to filter prefixes is with a route map. When used for filtering prefixes, a route map works like an access list. It has multiple statements that are read in a sequential order. Each statement can be a deny or permit and can have a match clause for a variety of attributes, such as the route or a route tag. You can also include route attributes in each statement that will be set if the match clause is met.

a. Before filtering the R3 loopback 25 and 30 networks from being redistributed into EIGRP on R2, display the R1 routing table and verify that those two routes currently appear there.

```
R1# show ip route eigrp
D EX 192.168.30.0/24 [170/40537600] via 172.16.12.2, 00:04:28, Serial0/0/0
D EX 192.168.25.0/24 [170/40537600] via 172.16.12.2, 00:04:28, Serial0/0/0
D EX 192.168.40.0/24 [170/40537600] via 172.16.12.2, 00:04:28, Serial0/0/0
     172.16.0.0/24 is subnetted, 6 subnets
D       172.16.23.0 [90/41024000] via 172.16.12.2, 00:42:25, Serial0/0/0
D       172.16.2.0 [90/40640000] via 172.16.12.2, 00:42:25, Serial0/0/0
D EX    172.16.3.0 [170/40537600] via 172.16.12.2, 00:04:28, Serial0/0/0
D       172.16.100.0 [90/40640000] via 172.16.12.2, 01:34:26, Serial0/0/0
D EX 192.168.20.0/24 [170/40537600] via 172.16.12.2, 00:04:28, Serial0/0/0
D EX 192.168.35.0/24 [170/40537600] via 172.16.12.2, 00:04:28, Serial0/0/0
D EX 192.168.8.0/22 [170/40537600] via 172.16.12.2, 00:04:28, Serial0/0/0
D    192.168.48.0/23 is a summary, 04:24:12, Null0
```

There are multiple ways to configure this filtering. For this exercise, configure an access list that matches these two network addresses and a route map that denies based on a match for that access list.

b. Configure the access list as follows:

```
R2(config)# access-list 1 permit 192.168.25.0
R2(config)# access-list 1 permit 192.168.30.0
```

c. Configure a route map with a statement that denies based on a match with this access list. Then add a permit statement without a match statement, which acts as an explicit permit all.

```
R2(config)# route-map SELECTED-DENY deny 10
R2(config-route-map)# match ip address 1
R2(config-route-map)# route-map SELECTED-DENY permit 20
```

d. Apply this route map by redoing the **redistribute** command with the route map under the EIGRP process.

```
R2(config)# router eigrp 1
R2(config-router)# redistribute ospf 1 route-map SELECTED-DENY metric 64 100
255 1 1500
```


e. As an alternative, if you previously configured a default metric under EIGRP, you can simply use the following command.

```
R2(config-router)# redistribute ospf 1 route-map SELECTED-DENY
```

f. Verify that these routes are filtered out in the R1 routing table.

```
R1# show ip route eigrp
D EX 192.168.40.0/24 [170/40537600] via 172.16.12.2, 00:07:24, Serial0/0/0
     172.16.0.0/24 is subnetted, 6 subnets
D       172.16.23.0 [90/41024000] via 172.16.12.2, 00:45:21, Serial0/0/0
D       172.16.2.0 [90/40640000] via 172.16.12.2, 00:45:21, Serial0/0/0
D EX    172.16.3.0 [170/40537600] via 172.16.12.2, 00:07:24, Serial0/0/0
D       172.16.100.0 [90/40640000] via 172.16.12.2, 00:45:21, Serial0/0/0
D EX 192.168.20.0/24 [170/40537600] via 172.16.12.2, 00:07:24, Serial0/0/0
D EX 192.168.35.0/24 [170/40537600] via 172.16.12.2, 00:07:24, Serial0/0/0
D EX 192.168.8.0/22 [170/40537600] via 172.16.12.2, 00:07:24, Serial0/0/0
D    192.168.48.0/23 is a summary, 04:27:07, Null0
```

Step 9: Summarize external routes into OSPF at the ASBR.

You cannot summarize routes redistributed into OSPF using the **area range** command. This command is effective only on routes internal to the specified area. Instead, use the OSPF **summary-address** *network mask* command.

a. Before you make any changes, display the R3 routing table.

```
R3# show ip route ospf
     172.16.0.0/24 is subnetted, 6 subnets
O E2    172.16.12.0 [110/20] via 172.16.23.2, 00:00:07, Serial0/0/1
O E2    172.16.1.0 [110/20] via 172.16.23.2, 00:00:07, Serial0/0/1
O E2    172.16.2.0 [110/20] via 172.16.23.2, 00:00:07, Serial0/0/1
O IA    172.16.100.0 [110/1563] via 172.16.23.2, 00:00:07, Serial0/0/1
O E2 192.168.70.0/24 [110/20] via 172.16.23.2, 00:00:07, Serial0/0/1
O    192.168.8.0/22 is a summary, 00:00:07, Null0
O E2 192.168.51.0/24 [110/20] via 172.16.23.2, 00:00:07, Serial0/0/1
O E2 192.168.50.0/24 [110/20] via 172.16.23.2, 00:00:07, Serial0/0/1
O E2 192.168.48.0/23 [110/20] via 172.16.23.2, 00:00:07, Serial0/0/1
```

Notice the three external routes for the R1 loopback interfaces 48 through 51. Two of the loopbacks are already summarized to one /23.

Which mask should you use to summarize all four of the loopbacks to one prefix?

____

b. You can summarize this all into one supernet on R2 using the following commands.

```
R2(config)# router ospf 1
R2(config-router)# summary-address 192.168.48.0 255.255.252.0
```

c. Verify this action in the R3 routing table.

```
R3# show ip route ospf
     172.16.0.0/24 is subnetted, 6 subnets
O E2    172.16.12.0 [110/20] via 172.16.23.2, 01:40:45, Serial0/0/1
O E2    172.16.1.0 [110/20] via 172.16.23.2, 00:48:54, Serial0/0/1
O E2    172.16.2.0 [110/20] via 172.16.23.2, 01:40:45, Serial0/0/1
O IA    172.16.100.0 [110/1563] via 172.16.23.2, 01:40:45, Serial0/0/1
O E2 192.168.70.0/24 [110/20] via 172.16.23.2, 00:48:54, Serial0/0/1
O    192.168.8.0/22 is a summary, 01:41:55, Null0
O E2 192.168.48.0/22 [110/20] via 172.16.23.2, 00:00:08, Serial0/0/1
```

What would happen if loopback 50 on R1 were to become unreachable by R2?

____

____

Would data destined for 192.168.50.0/24 from R3 still be sent to R2?

____

____

Would data destined for 192.168.50.0/24 from R2 continue to be sent to R1?

____

____

d. If you are unsure of the outcome, shut down the interface on R1. Issue the ICMP **traceroute** command to 192.168.50.1 from R3 and then from R2. Check your output against the output and analysis in Appendix A. Remember to issue the **no shutdown** command when you are finished checking.

Is this a desirable outcome? Explain.

____

____

____

____

Step 10: Modify EIGRP distances.

a. By default, EIGRP uses an administrative distance of 90 for internal routes and 170 for external routes. You can see this in the R1 routing table and in the output of the **show ip protocols** command.

```
R1# show ip route eigrp
D EX 192.168.40.0/24 [170/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0
     172.16.0.0/24 is subnetted, 6 subnets
D       172.16.23.0 [90/41024000] via 172.16.12.2, 00:04:03, Serial0/0/0
D       172.16.2.0 [90/40640000] via 172.16.12.2, 00:04:03, Serial0/0/0
D EX    172.16.3.0 [170/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0
D       172.16.100.0 [90/40640000] via 172.16.12.2, 00:04:03, Serial0/0/0
D EX 192.168.20.0/24 [170/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0
D EX 192.168.35.0/24 [170/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0
D EX 192.168.8.0/22 [170/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0
D    192.168.48.0/23 is a summary, 3d17h, Null0
D EX 192.168.48.0/22 [170/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0

R1# show ip protocols
<output omitted>

  Address Summarization:
    192.168.48.0/23 for Serial0/0/0
      Summarizing with metric 128256
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.0.0/16
  Routing Information Sources:
    Gateway         Distance      Last Update
```



```
    (this router)         90      00:34:33
    172.16.12.2           90      00:16:35
  Distance: internal 90 external 170
```

b. You can change the administrative distance with the **distance eigrp** *internal external* command. This command is only applicable locally. Change the distance to 95 for internal routes and 165 for external routes.

```
R1(config)# router eigrp 1
R1(config-router)# distance eigrp 95 165
```

Note: The EIGRP neighbor adjacency will be re-negotiated:

```
R1#
*May  3 00:28:38.379: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.12.2
(Serial0/0/0) is down: route configuration changed
*May  3 00:28:41.503: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.12.2
(Serial0/0/0) is up: new adjacency
```

c. Verify the change in the routing table with the **show ip route eigrp** and **show ip protocols** commands.

```
R1# show ip route eigrp
D EX 192.168.40.0/24 [165/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0
     172.16.0.0/24 is subnetted, 6 subnets
D       172.16.23.0 [95/41024000] via 172.16.12.2, 00:04:03, Serial0/0/0
D       172.16.2.0 [95/40640000] via 172.16.12.2, 00:04:03, Serial0/0/0
D EX    172.16.3.0 [165/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0
D       172.16.100.0 [95/40640000] via 172.16.12.2, 00:04:03, Serial0/0/0
D EX 192.168.20.0/24 [165/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0
D EX 192.168.35.0/24 [165/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0
D EX 192.168.8.0/22 [165/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0
D    192.168.48.0/23 is a summary, 3d17h, Null0
D EX 192.168.48.0/22 [165/40537600] via 172.16.12.2, 00:04:03, Serial0/0/0

R1# show ip protocols
Routing Protocol is "eigrp 1"
<output omitted>

  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.12.2           95      00:00:00
  Distance: internal 95 external 165
```

Step 11: Modify OSPF distances.

You can also modify individual OSPF distances. By default, all OSPF distances are 110, but you can change the intra-area, inter-area, and external route distances using the **distance ospf intra-area** *distance* **inter-area** *distance* **external** *distance* command. All the command arguments are optional, so you can change only what you need to.

a. Before changing anything, display the R3 routing table.

```
R3# show ip route ospf
     172.16.0.0/24 is subnetted, 6 subnets
O E2    172.16.12.0 [110/20] via 172.16.23.2, 01:40:45, Serial0/0/1
O E2    172.16.1.0 [110/20] via 172.16.23.2, 00:48:54, Serial0/0/1
O E2    172.16.2.0 [110/20] via 172.16.23.2, 01:40:45, Serial0/0/1
O IA    172.16.100.0 [110/1563] via 172.16.23.2, 01:40:45, Serial0/0/1
O E2 192.168.70.0/24 [110/20] via 172.16.23.2, 00:48:54, Serial0/0/1
O    192.168.8.0/22 is a summary, 01:41:55, Null0
```



```
O E2 192.168.48.0/22 [110/20] via 172.16.23.2, 00:00:08, Serial0/0/1
```

b. Change the intra-area distance to 105, inter-area distance to 115, and external routes to 175 on R3.

```
R3(config)# router ospf 1
R3(config-router)# distance ospf intra-area 105 inter-area 115 external 175
```

c. Verify the change in the routing table. Unfortunately, the only information that you can get from the output of the **show ip protocols** command is the default distance, which is the intra-area distance.

```
R3# show ip route ospf
     172.16.0.0/24 is subnetted, 6 subnets
O E2    172.16.12.0 [175/20] via 172.16.23.2, 00:00:05, Serial0/0/1
O E2    172.16.1.0 [175/20] via 172.16.23.2, 00:00:05, Serial0/0/1
O E2    172.16.2.0 [175/20] via 172.16.23.2, 00:00:05, Serial0/0/1
O IA    172.16.100.0 [115/1563] via 172.16.23.2, 00:00:05, Serial0/0/1
O E2 192.168.70.0/24 [175/20] via 172.16.23.2, 00:00:05, Serial0/0/1
O    192.168.8.0/22 is a summary, 00:00:05, Null0
O E2 192.168.48.0/22 [175/20] via 172.16.23.2, 00:00:05, Serial0/0/1

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
  It is an area border router
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    172.16.0.0 0.0.255.255 area 0
    192.168.8.0 0.0.3.255 area 20
    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
    Loopback8
    Loopback9
    Loopback10
    Loopback11
  Passive Interface(s):
    Loopback20
    Loopback25
    Loopback30
    Loopback35
    Loopback40
    VoIP-Null0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)        110      00:03:04
    172.16.2.1           110      00:03:04
  Distance: (default is 105)
```


Challenge: Change the Administrative Distance on R2

The previous two steps demonstrated using the **distance** command in a fairly inconsequential environment. In which types of scenarios would the **distance** command be more valuable?

____

____

____

____

On R2, you are running both EIGRP and OSPF. Imagine a fourth router, R4, connected to both R1 and R3. R4 is redistributing between the two routing protocols.

Using the default administrative distances for EIGRP and OSPF, which protocol would be preferred in the routing table for destination prefixes in native OSPF networks and why?

____

____

____

Which protocol would be preferred in the routing table for destination prefixes for native EIGRP networks?

____

____

____

Instead of adding the 172.16.1.0/24 networks natively to EIGRP using a **network** statement, add the networks using the **redistribute connected** command in EIGRP configuration mode on R1.

____

____

With the default administrative distances set, what would the administrative distance be for that prefix on R2 in EIGRP and in OSPF? Explain why.

____

____

____

How could you make the EIGRP path prefer this route? Is there more than one way?

____

____

____

____

____

____

Could using the **distance** command in this situation cause asymmetric routing? Explain.

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Appendix A: Exploring Black Hole Operations

a. Configure R1 and shut down the loopback 50 interface:

```
R1(config)# interface loopback 50
R1(config-if)# shutdown
```

b. On R2, you should see the following output.

```
R2# show ip route

Gateway of last resort is not set

O    192.168.30.0/24 [110/1563] via 172.16.23.3, 18:53:52, Serial0/0/1
O    192.168.25.0/24 [110/1563] via 172.16.23.3, 18:53:52, Serial0/0/1
O    192.168.40.0/24 [110/1563] via 172.16.23.3, 18:53:52, Serial0/0/1
     172.16.0.0/24 is subnetted, 6 subnets
C       172.16.100.0 is directly connected, Loopback100
C       172.16.23.0 is directly connected, Serial0/0/1
C       172.16.12.0 is directly connected, Serial0/0/0
D       172.16.1.0 [90/40640000] via 172.16.12.1, 18:54:06, Serial0/0/0
C       172.16.2.0 is directly connected, Loopback0
O       172.16.3.0 [110/1563] via 172.16.23.3, 18:53:53, Serial0/0/1
O    192.168.20.0/24 [110/1563] via 172.16.23.3, 18:53:53, Serial0/0/1
D    192.168.51.0/24 [90/40640000] via 172.16.12.1, 18:54:07, Serial0/0/0
O    192.168.35.0/24 [110/1563] via 172.16.23.3, 18:53:53, Serial0/0/1
D    192.168.70.0/24 [90/40640000] via 172.16.12.1, 18:54:07, Serial0/0/0
O IA 192.168.8.0/22 [110/1563] via 172.16.23.3, 18:53:54, Serial0/0/1
D    192.168.48.0/23 [90/40640000] via 172.16.12.1, 18:54:08, Serial0/0/0
O    192.168.48.0/22 is a summary, 17:16:44, Null0
```

Notice the absence of 192.168.50.0/24 in a specific route in the R2 routing table.

c. Begin debugging all incoming IP packets on R2, and then issue the **ping 192.168.50.1** command.

```
R2# debug ip packet
```

```
R2# ping 192.168.50.1
(debug output cleaned up so as to be readable)

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.50.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

*Oct 17 16:39:14.147: IP: s=172.16.2.1 (local), d=192.168.50.1 (Null0), len
100, sending
...

R2# undebug all

R2# traceroute 192.168.50.1

Type escape sequence to abort.
Tracing the route to 192.168.50.1

  1  *  *  *
  2  *  *  *
  3  *  *  *
  4  *  *  *
  5  *  *  *
  6  *  *  *
  7  *  *  *
  8  *  *  *
  9  *  *  *

<output omitted>
```

The summary route, pointing to the Null0 interface as the next hop, acts as a “catch all” for any traffic generated by R2 or forwarded to R2 with the destination network 192.168.48.0/24. R2 sends traffic to the Null0 virtual interface, as shown by the IP packet debugging output highlighted above.

R2 is not able to ping the R1 shutdown loopback interface because the 192.168.50.0/24 route no longer exists in the routing table.

Check to see if network 192.168.50.0/24, or a supernet of it, is in the routing table of R3.

```
R3# show ip route 192.168.50.1
Routing entry for 192.168.48.0/22, supernet
  Known via "ospf 1", distance 110, metric 20, type extern 2, forward metric
1562
  Last update from 172.16.23.2 on Serial0/0/1, 00:39:17 ago
  Routing Descriptor Blocks:
  * 172.16.23.2, from 172.16.2.1, 00:39:17 ago, via Serial0/0/1
      Route metric is 20, traffic share count is 1
```

d. Begin debugging all IP and ICMP packets on R3. Ping the address 192.168.50.1 from R3. Try to trace the route from R3 to 192.168.50.1.

```
R3# debug ip packet
R3# debug ip icmp

R3# ping 192.168.50.1
<output omitted>

Type escape sequence to abort.
```



```
Sending 5, 100-byte ICMP Echos to 192.168.50.1, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

*Oct 17 16:49:21.023: IP: tableid=0, s=172.16.23.3 (local), d=192.168.50.1
(Serial0/0/1), routed via FIB

*Oct 17 16:49:21.047: ICMP: dst (172.16.23.3) host unreachable rcv from
172.16.23.2

R3# undebug all

R3# traceroute 192.168.50.1

Type escape sequence to abort.
Tracing the route to 192.168.50.1

  1 172.16.23.2 12 msec 12 msec 16 msec
  2 172.16.23.2 !H  !H  *
```

Analyze the process indicated by the ICMP responses. You might also want to refer to debugging messages for ICMP and IP packets on R2.

1. R3 generates an ICMP echo request (ping) to 192.168.50.1.
2. R3 looks up the (next-hop address, outgoing interface) pair for the longest matching prefix containing 192.168.50.1 in the IP routing table. It finds (172.16.23.2, Serial0/0/1).
3. R3 routes the IP packet to (172.16.23.2, Serial0/0/1).
4. R2 receives the IP packet from R3 on interface Serial0/0/1.
5. R2 looks up the (next-hop address, outgoing interface) pair for the longest prefix matching containing 192.168.50.1 in the IP routing table. The longest matching prefix that the routing table returns is 192.168.48.0/22, for which the routing table responds with (null, Null0) because it has no next-hop address or physical outgoing interface.
6. R2 realizes that this packet was routed remotely to it but that it has no route, so it sends an ICMP Type 3, Code 1 (host unreachable) packet to the source address of the packet, 172.16.23.3.
7. R2 looks up the (next-hop address, outgoing interface) pair for 172.16.23.3 and resolves it to (172.16.23.3, Serial0/0/1).
8. R2 then routes the ICMP packet for destination 172.16.23.3, normally 172.16.23.3 through Serial0/0/1.
9. R3 receives a packet destined for its local address 172.16.23.3 and reads the packet, sending the ICMP “Host Unreachable” message to the ping output.

Note: For more information about how routers respond to unreachable hosts, see RFC 792 (ICMP) at http://tools.ietf.org/html/rfc792 and RFC 4443 (ICMPv6) at http://tools.ietf.org/html/rfc4443.

Notice that R2 sends R3 an ICMP Type 3, Code 1 reply indicating that it does not have a route to the host 192.168.50.1. This ICMP “Host Unreachable” message is not only sent in response to pings or traceroutes (also a form of ICMP) but for all IP traffic.

e. If you were to use Telnet to 192.168.50.1, you would receive the following message based on the ICMP response from R2:

```
R3#telnet 192.168.50.1
Trying 192.168.50.1 ...
```



```
% Destination unreachable; gateway or host down

R3#
```

This is not an example of Telnet timing out, but of intelligent network protocols responding to routing issues in the network.

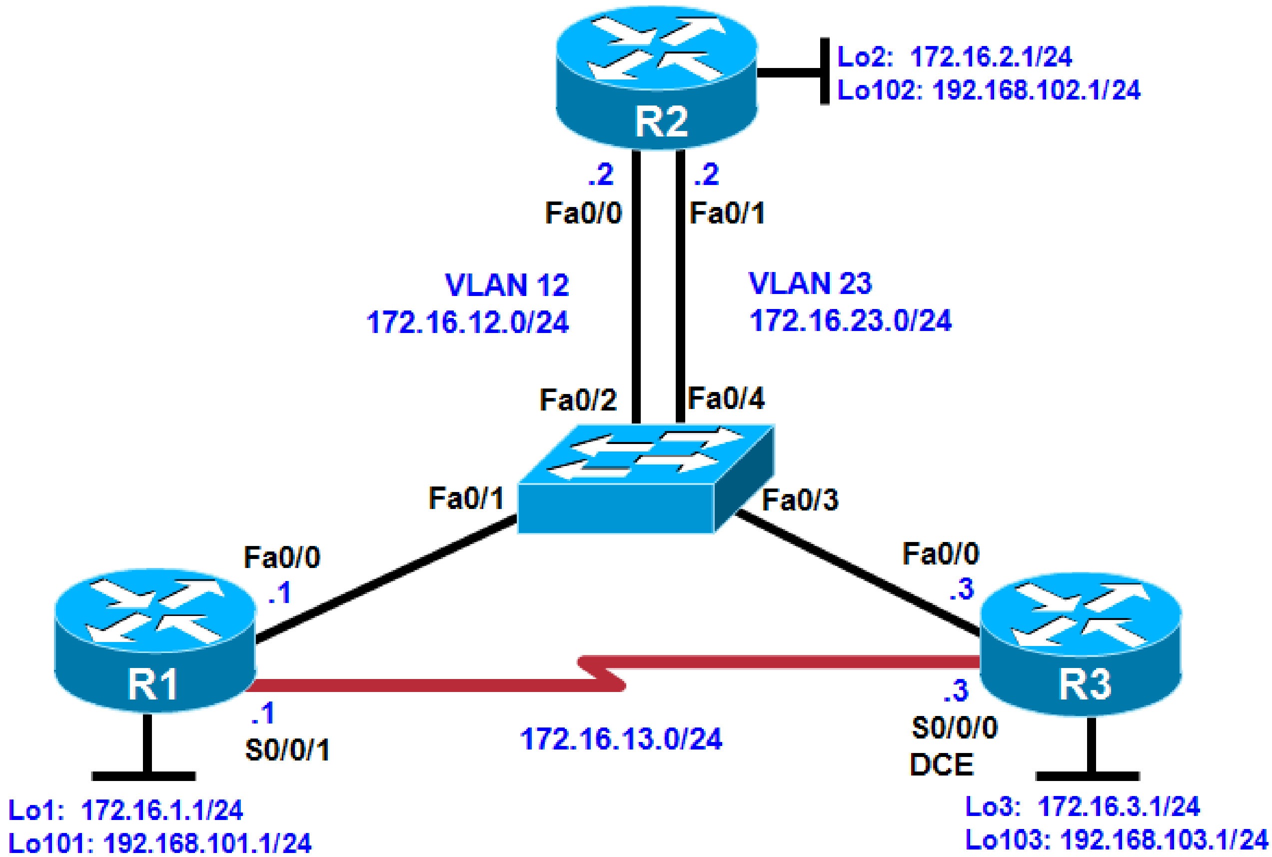
This summarization problem is a classic example of a “black hole” in a domain, which simply means traffic passing through the network destined for that subnet is discarded at some point along the way. Thankfully, ICMP informs sources of when their traffic is being discarded.

f. Do not forget to issue the **no shutdown** command on the R1 loopback 50 interface to re-enable routing to this network.

```
R1(config)# interface loopback 50
R1(config-if)# no shutdown
```

Chapter 4 Lab 4-3, Manipulating Administrative Distances

Topology



Objectives

- Configure RIP on a router.
- Configure OSPF on a router.
- Manipulate administrative distances.
- Compare routing protocol behavior.

Background

In this lab, you will compare the RIP and OSPF routing protocols based on how efficient they are at selecting routes, as well as what happens when you manipulate administrative distances in the routing table.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. The switch is a Cisco WS-C2960-24TT-L with the Cisco IOS image c2960-lanbasek9-mz.122-46.SE.bin. You can use other routers (such as a 2801 or 2811), switches (such as 2950), and Cisco IOS Software versions if they have comparable capabilities and features.

Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- 1 switch (Cisco 2960 with the Cisco IOS Release 12.2(46)SE C2960-LANBASEK9-M image or comparable)
- Serial and Ethernet cables

Step 1: Review default administrative distances.

Fill in the following table with all the administrative distances you can recall from your reading.

Protocol	Administrative Distance
Connected	
Static	
EIGRP Summary Route	
External BGP	
EIGRP	
IGRP	
OSPF	
IS-IS	
RIP	
EGP	
On-Demand Routing (ODR)	
External EIGRP	
Internal BGP	
Unknown	

Of the interior gateway protocols (IGPs) that you have studied, which one is considered the most trusted on a Cisco router and why?

____

____

Step 2: Configure router loopbacks and addressing.

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

```
R1# conf t
R1(config)# interface loopback 1
```



```
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# interface loopback 101
R1(config-if)# ip address 192.168.101.1 255.255.255.0
R1(config-if)# interface fastethernet 0/0
R1(config-if)# ip address 172.16.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface serial 0/0/1
R1(config-if)# bandwidth 64
R1(config-if)# ip address 172.16.13.1 255.255.255.0
R1(config-if)# no shutdown

R2# conf t
R2(config)# interface loopback 2
R2(config-if)# ip address 172.16.2.1 255.255.255.0
R2(config-if)# interface loopback 102
R2(config-if)# ip address 192.168.102.1 255.255.255.0
R2(config-if)# interface fastethernet 0/0
R2(config-if)# ip address 172.16.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface fastethernet 0/1
R2(config-if)# ip address 172.16.23.2 255.255.255.0
R2(config-if)# no shutdown

R3# conf t
R3(config)# interface loopback 3
R3(config-if)# ip address 172.16.3.1 255.255.255.0
R3(config-if)# interface loopback 103
R3(config-if)# ip address 192.168.103.1 255.255.255.0
R3(config-if)# interface fastethernet 0/0
R3(config-if)# ip address 172.16.23.3 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# interface serial 0/0/0
R3(config-if)# bandwidth 64
R3(config-if)# ip address 172.16.13.3 255.255.255.0
R3(config-if)# clock rate 64000
R3(config-if)# no shutdown
```

Step 3: Configure switch VLANs.

a. Configure the switch VLANs, and place the correct access ports in each VLAN.

Note: The switch ports used are not important as long as the ports connecting to R1 Fa0/0 and R2 Fa0/0 are in VLAN 12 and the ports connecting to R3 Fa0/0 and R2 Fa0/1 are in VLAN 23.

```
Switch(config)# vlan 12
Switch(config-vlan)# name R1-R2
Switch(config-vlan)# vlan 23
Switch(config-vlan)# name R2-R3
Switch(config-vlan)# exit

Switch(config)# interface fastEthernet 0/1
Switch(config-if)# description To R1 Fa0/0
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 12

Switch(config-if)# interface fastEthernet 0/2
Switch(config-if)# description To R2 Fa0/0
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 12
```

```
Switch(config-if)# interface fastEthernet 0/3
Switch(config-if)# description To R3 Fa0/0
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 23

Switch(config-if)# interface fastEthernet 0/4
Switch(config-if)# description To R2 Fa0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 23
```

b. Verify that you can ping across the local subnets.

Step 4: Configure RIP.

a. Configure RIPv2 on all three routers for the major networks. Disable automatic summarization.

```
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 192.168.101.0

R2(config)# router rip
R2(config-router)# version 2
R2(config-router)# no auto-summary
R2(config-router)# network 172.16.0.0
R2(config-router)# network 192.168.102.0

R3(config)# router rip
R3(config-router)# version 2
R3(config-router)# no auto-summary
R3(config-router)# network 172.16.0.0
R3(config-router)# network 192.168.103.0
```

b. Verify the configuration using the **show ip route rip** command on each router.

```
R1# show ip route rip
     172.16.0.0/24 is subnetted, 6 subnets
R       172.16.23.0 [120/1] via 172.16.13.3, 00:02:29, Serial0/0/1
                    [120/1] via 172.16.12.2, 00:02:15, FastEthernet0/0
R       172.16.2.0 [120/1] via 172.16.12.2, 00:02:15, FastEthernet0/0
R       172.16.3.0 [120/1] via 172.16.13.3, 00:02:29, Serial0/0/1
R    192.168.102.0/24 [120/1] via 172.16.12.2, 00:02:15, FastEthernet0/0
R    192.168.103.0/24 [120/1] via 172.16.13.3, 00:02:29, Serial0/0/1

R2# show ip route rip
     172.16.0.0/24 is subnetted, 6 subnets
R       172.16.13.0 [120/1] via 172.16.23.3, 00:02:18, FastEthernet0/1
                    [120/1] via 172.16.12.1, 00:02:20, FastEthernet0/0
R       172.16.1.0 [120/1] via 172.16.12.1, 00:02:20, FastEthernet0/0
R       172.16.3.0 [120/1] via 172.16.23.3, 00:02:18, FastEthernet0/1
R    192.168.103.0/24 [120/1] via 172.16.23.3, 00:02:18, FastEthernet0/1
R    192.168.101.0/24 [120/1] via 172.16.12.1, 00:02:20, FastEthernet0/0

R3# show ip route rip
     172.16.0.0/24 is subnetted, 6 subnets
R       172.16.12.0 [120/1] via 172.16.23.2, 00:02:32, FastEthernet0/0
                    [120/1] via 172.16.13.1, 00:02:47, Serial0/0/0
R       172.16.1.0 [120/1] via 172.16.13.1, 00:02:47, Serial0/0/0
```



```
R       172.16.2.0 [120/1] via 172.16.23.2, 00:02:32, FastEthernet0/0
R    192.168.102.0/24 [120/1] via 172.16.23.2, 00:02:32, FastEthernet0/0
R    192.168.101.0/24 [120/1] via 172.16.13.1, 00:02:47, Serial0/0/0
```

Notice that on R1, RIP chooses the serial interface as the best next hop for the R3 loopback interface.

c. Verify that each router is receiving RIP routes from other routers using the **show ip protocols** command.

```
R1# 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 26 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
    FastEthernet0/0       2     2
    Serial0/0/1           2     2
    Loopback1             2     2
    Loopback101           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.101.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.12.2          120      00:00:21
    172.16.13.3          120      00:00:03
  Distance: (default is 120)

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
    FastEthernet0/0       2     2
    FastEthernet0/1       2     2
    Loopback2             2     2
    Loopback102           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.102.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.23.3          120      00:00:02
    172.16.12.1          120      00:00:24
  Distance: (default is 120)

R3# 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 22 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
    FastEthernet0/0       2     2
    Serial0/0/0           2     2
    Loopback3             2     2
    Loopback103           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.103.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.23.2          120      00:00:06
    172.16.13.1          120      00:00:17
  Distance: (default is 120)
```

Step 5: Configure OSPF.

a. Configure OSPF on all routers. Include the entire major network in area 0 on all three routers. Remember to change the network type on the loopback interfaces.

```
R1(config)# interface loopback 1
R1(config-if)# ip ospf network point-to-point
R1(config-if)# interface loopback 101
R1(config-if)# ip ospf network point-to-point
R1(config-if)# router ospf 1
R1(config-router)# network 172.16.0.0 0.0.255.255 area 0
R1(config-router)# network 192.168.101.0 0.0.0.255 area 0

R2(config)# interface loopback 2
R2(config-if)# ip ospf network point-to-point
R2(config-if)# interface loopback 102
R2(config-if)# ip ospf network point-to-point
R2(config-if)# router ospf 1
R2(config-router)# network 172.16.0.0 0.0.255.255 area 0
R2(config-router)# network 192.168.102.0 0.0.0.255 area 0

R3(config)# interface loopback 3
R3(config-if)# ip ospf network point-to-point
R3(config-if)# interface loopback 103
R3(config-if)# ip ospf network point-to-point
R3(config-if)# router ospf 1
R3(config-router)# network 172.16.0.0 0.0.255.255 area 0
R3(config-router)# network 192.168.103.0 0.0.0.255 area 0
```

b. Verify the configuration using the **show ip ospf neighbors** and **show ip route** commands on each router.

```
R1# show ip ospf neighbor

Neighbor ID     Pri   State       Dead Time   Address        Interface
192.168.103.1     0   FULL/  -    00:00:39    172.16.13.3    Serial0/0/1
192.168.102.1     1   FULL/DR     00:00:39    172.16.12.2    FastEthernet0/0
```



```
R1# show ip route
<output omitted>
     172.16.0.0/24 is subnetted, 6 subnets
O       172.16.23.0 [110/2] via 172.16.12.2, 00:00:48, FastEthernet0/0
C       172.16.12.0 is directly connected, FastEthernet0/0
C       172.16.13.0 is directly connected, Serial0/0/1
C       172.16.1.0 is directly connected, Loopback1
O       172.16.2.0 [110/2] via 172.16.12.2, 00:00:48, FastEthernet0/0
O       172.16.3.0 [110/3] via 172.16.12.2, 00:00:48, FastEthernet0/0
O    192.168.102.0/24 [110/2] via 172.16.12.2, 00:00:48, FastEthernet0/0
O    192.168.103.0/24 [110/3] via 172.16.12.2, 00:00:49, FastEthernet0/0
C    192.168.101.0/24 is directly connected, Loopback101

R2# show ip ospf neighbor

Neighbor ID     Pri   State      Dead Time   Address        Interface
192.168.103.1     1   FULL/DR    00:00:31    172.16.23.3   FastEthernet0/1
192.168.101.1     1   FULL/BDR   00:00:34    172.16.12.1   FastEthernet0/0

R2# show ip route
<output omitted>
     172.16.0.0/24 is subnetted, 6 subnets
C       172.16.23.0 is directly connected, FastEthernet0/1
C       172.16.12.0 is directly connected, FastEthernet0/0
O       172.16.13.0 [110/1563] via 172.16.23.3, 00:01:19, FastEthernet0/1
                    [110/1563] via 172.16.12.1, 00:01:19, FastEthernet0/0
O       172.16.1.0 [110/2] via 172.16.12.1, 00:01:19, FastEthernet0/0
C       172.16.2.0 is directly connected, Loopback2
O       172.16.3.0 [110/2] via 172.16.23.3, 00:01:19, FastEthernet0/1
C    192.168.102.0/24 is directly connected, Loopback102
O    192.168.103.0/24 [110/2] via 172.16.23.3, 00:01:20, FastEthernet0/1
O    192.168.101.0/24 [110/2] via 172.16.12.1, 00:01:20, FastEthernet0/0

R3# show ip ospf neighbor

Neighbor ID     Pri   State      Dead Time   Address        Interface
192.168.101.1     0   FULL/  -   00:00:36    172.16.13.1   Serial0/0/0
192.168.102.1     1   FULL/BDR   00:00:33    172.16.23.2   FastEthernet0/0

R3# show ip route
<output omitted>
     172.16.0.0/24 is subnetted, 6 subnets
C       172.16.23.0 is directly connected, FastEthernet0/0
O       172.16.12.0 [110/2] via 172.16.23.2, 00:02:10, FastEthernet0/0
C       172.16.13.0 is directly connected, Serial0/0/0
O       172.16.1.0 [110/3] via 172.16.23.2, 00:02:10, FastEthernet0/0
O       172.16.2.0 [110/2] via 172.16.23.2, 00:02:10, FastEthernet0/0
C       172.16.3.0 is directly connected, Loopback3
O    192.168.102.0/24 [110/2] via 172.16.23.2, 00:02:10, FastEthernet0/0
C    192.168.103.0/24 is directly connected, Loopback103
O    192.168.101.0/24 [110/3] via 172.16.23.2, 00:02:11, FastEthernet0/0
```

Notice that all the OSPF routes have replaced the RIP routes in the routing table. This is because OSPF has an administrative distance of 110, and RIP has an administrative distance of 120.

What is the best next hop on R1 for 172.16.3.1 with only RIP running?

____

____

What is the best next hop on R1 for 172.16.3.1 with OSPF running?

____

____

____

On R1, the best next hop for the R3 loopback is now through the VLAN between R1 and R2. This is because the sum of the costs for the two Ethernet links is still less than that of the single low-bandwidth (64 kb/s) serial link. This is one of the reasons why RIP's metric of a hop count is not very effective.

Which metric does R1 use to make routing decisions about whether to cross the serial link to R3 to reach R3's 172.16.3.1?

____

____

____

Use the following information for your answer.

```
R1# show ip ospf database router adv-router 192.168.103.1

            OSPF Router with ID (192.168.101.1) (Process ID 1)

                Router Link States (Area 0)

  LS age: 433
  Options: (No TOS-capability, DC)
  LS Type: Router Links
  Link State ID: 192.168.103.1
  Advertising Router: 192.168.103.1
  LS Seq Number: 80000003
  Checksum: 0xE87F
  Length: 84
  Number of Links: 5

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 192.168.103.0
     (Link Data) Network Mask: 255.255.255.0
      Number of TOS metrics: 0
       TOS 0 Metrics: 1

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 172.16.3.0
     (Link Data) Network Mask: 255.255.255.0
      Number of TOS metrics: 0
       TOS 0 Metrics: 1

    Link connected to: another Router (point-to-point)
     (Link ID) Neighboring Router ID: 192.168.101.1
     (Link Data) Router Interface address: 172.16.13.3
      Number of TOS metrics: 0
       TOS 0 Metrics: 1562

    Link connected to: a Stub Network
     (Link ID) Network/subnet number: 172.16.13.0
```



```
     (Link Data) Network Mask: 255.255.255.0
      Number of TOS metrics: 0
       TOS 0 Metrics: 1562

    Link connected to: a Transit Network
     (Link ID) Designated Router address: 172.16.23.3
     (Link Data) Router Interface address: 172.16.23.3
      Number of TOS metrics: 0
       TOS 0 Metrics: 1
```

Step 6: Modify the routing protocol distance.

The **distance** command is a protocol-independent way to manipulate routing protocol distances. This command is different from the routing protocol-specific commands such as **distance ospf** and **distance eigrp**. This command lets you globally change a routing protocol's distances, change only routes from a certain neighbor or those matching an access list, or a combination of any two of these three options.

Try applying the **distance** *distance* command, which changes the distance of every route. The previous output of the **show ip route** command shows that OSPF marks routes it injects into the routing table with a default administrative distance of 110. RIP injects routes into the routing table with a default administrative distance of 120.

What would happen if the administrative distance on each router for RIP were set to 100?

____

____

a. On all three routers, change the distance of RIP to 100.

```
R1(config)# router rip
R1(config-router)# distance 100

R2(config)# router rip
R2(config-router)# distance 100

R3(config)# router rip
R3(config-router)# distance 100
```

b. Examine the output of the **show ip route** command. Notice that *all* the routes have become RIP routes because RIP now has a lower distance than OSPF.

```
R1# show ip route
<output omitted>
     172.16.0.0/24 is subnetted, 6 subnets
R       172.16.23.0 [100/1] via 172.16.13.3, 00:00:17, Serial0/0/1
                    [100/1] via 172.16.12.2, 00:00:09, FastEthernet0/0
C       172.16.12.0 is directly connected, FastEthernet0/0
C       172.16.13.0 is directly connected, Serial0/0/1
C       172.16.1.0 is directly connected, Loopback1
R       172.16.2.0 [100/1] via 172.16.12.2, 00:00:09, FastEthernet0/0
R       172.16.3.0 [100/1] via 172.16.13.3, 00:00:17, Serial0/0/1
R    192.168.102.0/24 [100/1] via 172.16.12.2, 00:00:10, FastEthernet0/0
R    192.168.103.0/24 [100/1] via 172.16.13.3, 00:00:18, Serial0/0/1
C    192.168.101.0/24 is directly connected, Loopback101

R2# show ip route
<output omitted>
     172.16.0.0/24 is subnetted, 6 subnets
C       172.16.23.0 is directly connected, FastEthernet0/1
```



```
C       172.16.12.0 is directly connected, FastEthernet0/0
R       172.16.13.0 [100/1] via 172.16.23.3, 00:00:07, FastEthernet0/1
                    [100/1] via 172.16.12.1, 00:00:07, FastEthernet0/0
R       172.16.1.0 [100/1] via 172.16.12.1, 00:00:07, FastEthernet0/0
C       172.16.2.0 is directly connected, Loopback2
R       172.16.3.0 [100/1] via 172.16.23.3, 00:00:07, FastEthernet0/1
C    192.168.102.0/24 is directly connected, Loopback102
R    192.168.103.0/24 [100/1] via 172.16.23.3, 00:00:08, FastEthernet0/1
R    192.168.101.0/24 [100/1] via 172.16.12.1, 00:00:08, FastEthernet0/0

R3# show ip route
<output omitted>
     172.16.0.0/24 is subnetted, 6 subnets
C       172.16.23.0 is directly connected, FastEthernet0/0
R       172.16.12.0 [100/1] via 172.16.23.2, 00:00:07, FastEthernet0/0
                    [100/1] via 172.16.13.1, 00:00:02, Serial0/0/0
C       172.16.13.0 is directly connected, Serial0/0/0
R       172.16.1.0 [100/1] via 172.16.13.1, 00:00:02, Serial0/0/0
R       172.16.2.0 [100/1] via 172.16.23.2, 00:00:07, FastEthernet0/0
C       172.16.3.0 is directly connected, Loopback3
R    192.168.102.0/24 [100/1] via 172.16.23.2, 00:00:08, FastEthernet0/0
C    192.168.103.0/24 is directly connected, Loopback103
R    192.168.101.0/24 [100/1] via 172.16.13.1, 00:00:03, Serial0/0/0
```

c. You can display the new default distance for RIP using the **show ip protocols** command.

```
R1# 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 11 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
    FastEthernet0/0       2     2
    Serial0/0/1           2     2
    Loopback1             2     2
    Loopback101           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.101.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.13.3          100      00:00:14
    172.16.12.2          100      00:00:22
  Distance: (default is 100)
<output omitted>
```

Step 7: Modify distance based on route source.

You can also modify administrative distance based on route source using the **distance** *distance address wildcard* command, where *address* and *wildcard* represent the peer advertising the route. For OSPF, the address is the router ID.

a. On all three routers, change the OSPF administrative distance to 85 for any routes being advertised from routers with IDs in the range of 192.168.100.0/21.


```
R1(config)# router ospf 1
R1(config-router)# distance 85 192.168.100.0 0.0.3.255

R2(config)# router ospf 1
R2(config-router)# distance 85 192.168.100.0 0.0.3.255

R3(config)# router ospf 1
R3(config-router)# distance 85 192.168.100.0 0.0.3.255
```

b. Verify the change with the **show ip protocols** and **show ip route** commands.

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

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 6 subnets
O       172.16.23.0 [85/2] via 172.16.12.2, 00:00:31, FastEthernet0/0
C       172.16.12.0 is directly connected, FastEthernet0/0
C       172.16.13.0 is directly connected, Serial0/0/1
C       172.16.1.0 is directly connected, Loopback1
O       172.16.2.0 [85/2] via 172.16.12.2, 00:00:31, FastEthernet0/0
O       172.16.3.0 [85/3] via 172.16.12.2, 00:00:31, FastEthernet0/0
O    192.168.102.0/24 [85/2] via 172.16.12.2, 00:00:31, FastEthernet0/0
O    192.168.103.0/24 [85/3] via 172.16.12.2, 00:00:32, FastEthernet0/0
C    192.168.101.0/24 is directly connected, Loopback101

R2# show ip route
<output omitted>

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 6 subnets
C       172.16.23.0 is directly connected, FastEthernet0/1
C       172.16.12.0 is directly connected, FastEthernet0/0
O       172.16.13.0 [85/1563] via 172.16.23.3, 00:00:53, FastEthernet0/1
                    [85/1563] via 172.16.12.1, 00:00:53, FastEthernet0/0
O       172.16.1.0 [85/2] via 172.16.12.1, 00:00:53, FastEthernet0/0
C       172.16.2.0 is directly connected, Loopback2
O       172.16.3.0 [85/2] via 172.16.23.3, 00:00:53, FastEthernet0/1
C    192.168.102.0/24 is directly connected, Loopback102
O    192.168.103.0/24 [85/2] via 172.16.23.3, 00:00:54, FastEthernet0/1
O    192.168.101.0/24 [85/2] via 172.16.12.1, 00:00:54, FastEthernet0/0
R3# show ip route
<output omitted>

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 6 subnets
C       172.16.23.0 is directly connected, FastEthernet0/0
O       172.16.12.0 [85/2] via 172.16.23.2, 00:01:15, FastEthernet0/0
C       172.16.13.0 is directly connected, Serial0/0/0
O       172.16.1.0 [85/3] via 172.16.23.2, 00:01:15, FastEthernet0/0
O       172.16.2.0 [85/2] via 172.16.23.2, 00:01:15, FastEthernet0/0
C       172.16.3.0 is directly connected, Loopback3
O    192.168.102.0/24 [85/2] via 172.16.23.2, 00:01:15, FastEthernet0/0
C    192.168.103.0/24 is directly connected, Loopback103
O    192.168.101.0/24 [85/3] via 172.16.23.2, 00:01:16, FastEthernet0/0
```



```
R1# 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 192.168.101.1
  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
    192.168.101.0 0.0.0.255 area 0
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.103.1         85      00:05:47
    192.168.102.1         85      00:05:47
  Distance: (default is 110)
    Address         Wild mask       Distance  List
    192.168.100.0         0.0.3.255       85
```

Each of the routers should have an entry similar to the one highlighted above.

Step 8: Modify distance based on an access list.

You can also modify administrative distance based on which routes match an access list using the **distance** *distance address wildcard acl* command. The way you list routes in an access list which will be used to modify distance is similar to how you list them when the access list is used to filter routes. For this lab, create an access list containing all the subnets of 172.16.0.0/16. Then associate the access list with the **distance** command, setting the address and wildcard to be any IP address (i.e., any route source).

a. On all three routers, change the distances of the affected routes to 65.

```
R1(config)# access-list 1 permit 172.16.0.0 0.0.255.255
R1(config)# router rip
R1(config-router)# distance 65 0.0.0.0 255.255.255.255 1

R2(config)# access-list 1 permit 172.16.0.0 0.0.255.255
R2(config)# router rip
R2(config-router)# distance 65 0.0.0.0 255.255.255.255 1

R3(config)# access-list 1 permit 172.16.0.0 0.0.255.255
R3(config)# router rip
R3(config-router)# distance 65 0.0.0.0 255.255.255.255 1
```

b. Verify the change with the **show ip protocols** and **show ip route** commands.

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

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 22 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
    FastEthernet0/0       2     2
    Serial0/0/1           2     2
    Loopback1             2     2
```



```
    Loopback101           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.101.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.12.2           64      00:00:11
    172.16.13.3           64      00:00:12
  Distance: (default is 100)
    Address         Wild mask       Distance  List
    0.0.0.0         255.255.255.255       65  1

R1# show ip route
<output omitted>
     172.16.0.0/24 is subnetted, 6 subnets
R       172.16.23.0 [65/1] via 172.16.13.3, 00:00:20, Serial0/0/1
                    [65/1] via 172.16.12.2, 00:00:19, FastEthernet0/0
C       172.16.12.0 is directly connected, FastEthernet0/0
C       172.16.13.0 is directly connected, Serial0/0/1
C       172.16.1.0 is directly connected, Loopback1
R       172.16.2.0 [65/1] via 172.16.12.2, 00:00:19, FastEthernet0/0
R       172.16.3.0 [65/1] via 172.16.13.3, 00:00:20, Serial0/0/1
O    192.168.102.0/24 [85/2] via 172.16.12.2, 00:09:09, FastEthernet0/0
O    192.168.103.0/24 [85/3] via 172.16.12.2, 00:09:09, FastEthernet0/0
C    192.168.101.0/24 is directly connected, Loopback101

R2# show ip protocols
<output omitted>
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 27 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
    FastEthernet0/0       2     2
    FastEthernet0/1       2     2
    Loopback2             2     2
    Loopback102           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.102.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.23.3           65      00:00:06
    172.16.12.1           65      00:00:22
  Distance: (default is 100)
    Address         Wild mask       Distance  List
    0.0.0.0         255.255.255.255       65  1

R2# show ip route
<output omitted>
     172.16.0.0/24 is subnetted, 6 subnets
```



```
C       172.16.23.0 is directly connected, FastEthernet0/1
C       172.16.12.0 is directly connected, FastEthernet0/0
R       172.16.13.0 [65/1] via 172.16.23.3, 00:00:10, FastEthernet0/1
                    [65/1] via 172.16.12.1, 00:00:00, FastEthernet0/0
R       172.16.1.0 [65/1] via 172.16.12.1, 00:00:00, FastEthernet0/0
C       172.16.2.0 is directly connected, Loopback2
R       172.16.3.0 [65/1] via 172.16.23.3, 00:00:10, FastEthernet0/1
C    192.168.102.0/24 is directly connected, Loopback102
O    192.168.103.0/24 [85/2] via 172.16.23.3, 00:09:35, FastEthernet0/1
O    192.168.101.0/24 [85/2] via 172.16.12.1, 00:09:35, FastEthernet0/0

R3# show ip protocols
<output omitted>
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 15 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
    FastEthernet0/0       2     2
    Serial0/0/0           2     2
    Loopback3             2     2
    Loopback103           2     2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.103.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.23.2           65      00:00:24
    172.16.13.1           65      00:00:16
  Distance: (default is 100)
    Address         Wild mask       Distance  List
    0.0.0.0         255.255.255.255       65  1

R3# show ip route
<output omitted>
     172.16.0.0/24 is subnetted, 6 subnets
C       172.16.23.0 is directly connected, FastEthernet0/1
R       172.16.12.0 [65/1] via 172.16.23.2, 00:00:00, FastEthernet0/1
                    [65/1] via 172.16.13.1, 00:00:19, Serial0/0/0
C       172.16.13.0 is directly connected, Serial0/0/0
R       172.16.1.0 [65/1] via 172.16.13.1, 00:00:19, Serial0/0/0
R       172.16.2.0 [65/1] via 172.16.23.2, 00:00:00, FastEthernet0/1
C       172.16.3.0 is directly connected, Loopback3
O    192.168.102.0/24 [85/2] via 172.16.23.2, 00:09:43, FastEthernet0/1
C    192.168.103.0/24 is directly connected, Loopback103
O    192.168.101.0/24 [85/3] via 172.16.23.2, 00:09:43, FastEthernet0/1
```

c. Verify full connectivity with the following Tcl script.

```
R1# tclsh

foreach address {
172.16.1.1
172.16.2.1
```



```
172.16.3.1
172.16.12.1
172.16.12.2
172.16.13.1
172.16.13.3
172.16.23.2
172.16.23.3
192.168.101.1
192.168.102.1
192.168.103.1
} { ping $address }
```

Challenge

Attempt this exercise based on what you know about OSPF, Dijkstra's algorithm, and the **distance** command. Using only the **distance** command, write out the commands necessary to confuse the routers in this topology so that packets destined for 172.16.3.1 would continually bounce between R1 to R2?

____

____

____

Because it is possible to intentionally break routing in this way, what degree of caution should be exercised when manipulating administrative distances in a production network?

____

____

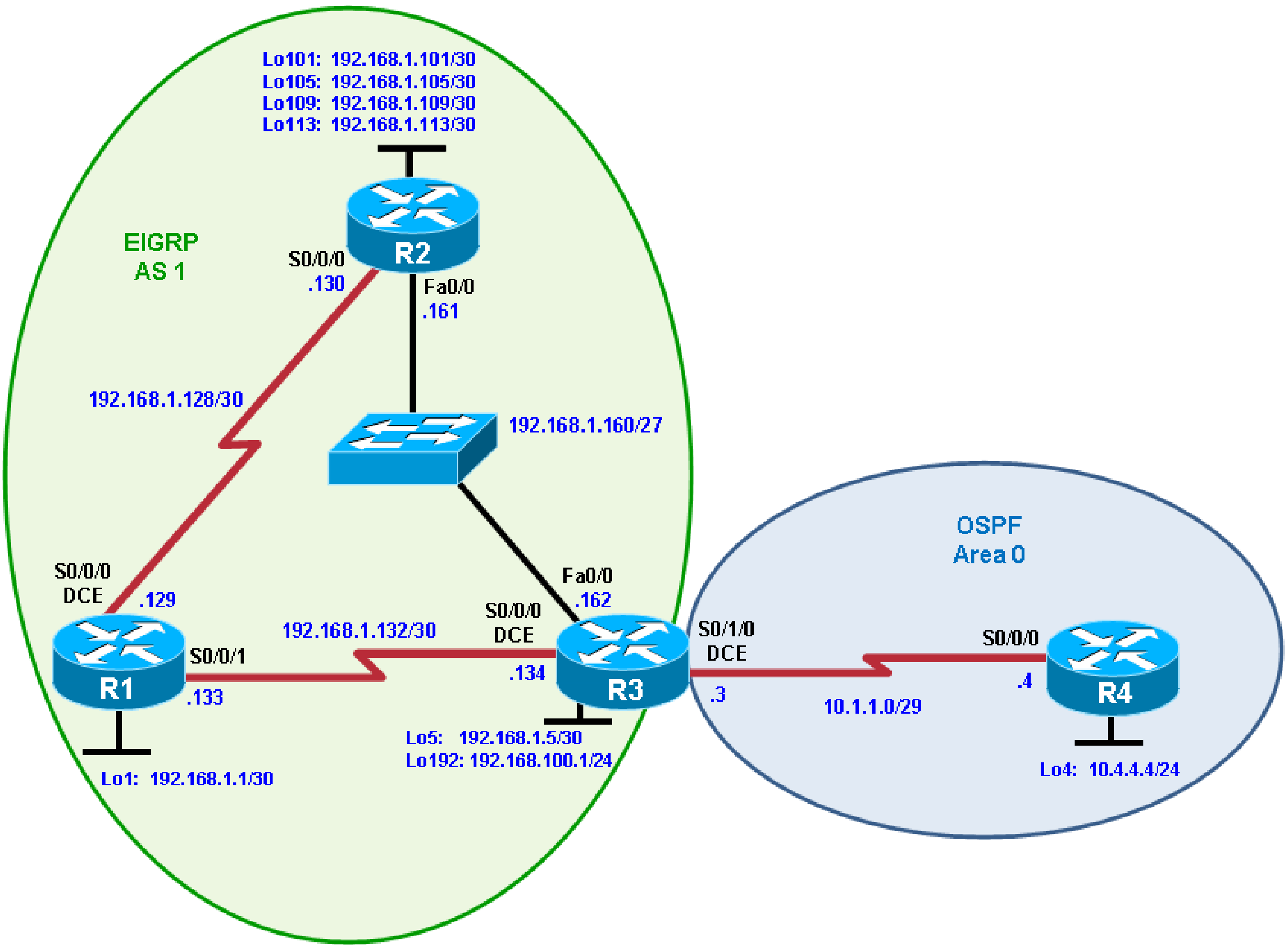
____

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 4 Lab 4-4, EIGRP and OSPF Case Study

Topology



Objectives

- Plan, design, and implement the International Travel Agency (ITA) EIGRP.
- Integrate the Local Travel Agency OSPF network with the ITA EIGRP network.
- Implement the design on the lab routers.
- Verify that all configurations are operational and functioning according to the guidelines.

Background

ITA requires its core network set up using EIGRP with the following specifications. It has also recently acquired Local Travel Agency, which was running OSPF. Use the addressing scheme shown in the diagram.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. The switch is a Cisco WS-C2960-24TT-L with the Cisco IOS

image c2960-lanbasek9-mz.122-46.SE.bin. You can use other routers (such as 2801 or 2811), switches (such as 2950), and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- 1 switch (Cisco 2960 with the Cisco IOS Release 12.2(46)SE C2960-LANBASEK9-M image or comparable)
- Serial and Ethernet cables

Requirements

1. The ITA core network is running EIGRP in AS 1.
2. Summarize the loopback interfaces on R2 with the best summary route to the other EIGRP routers.
3. Loopback 192 on R3 represents a connection to the Internet. Originate a default route into EIGRP from R3.
4. The Local Travel Agency router, R4, needs to communicate with the ITA core via OSPF area 0.
5. Redistribute OSPF into EIGRP.
6. Originate a default route into the OSPF process from R3.
7. Test connectivity using a Tcl script on all routers.

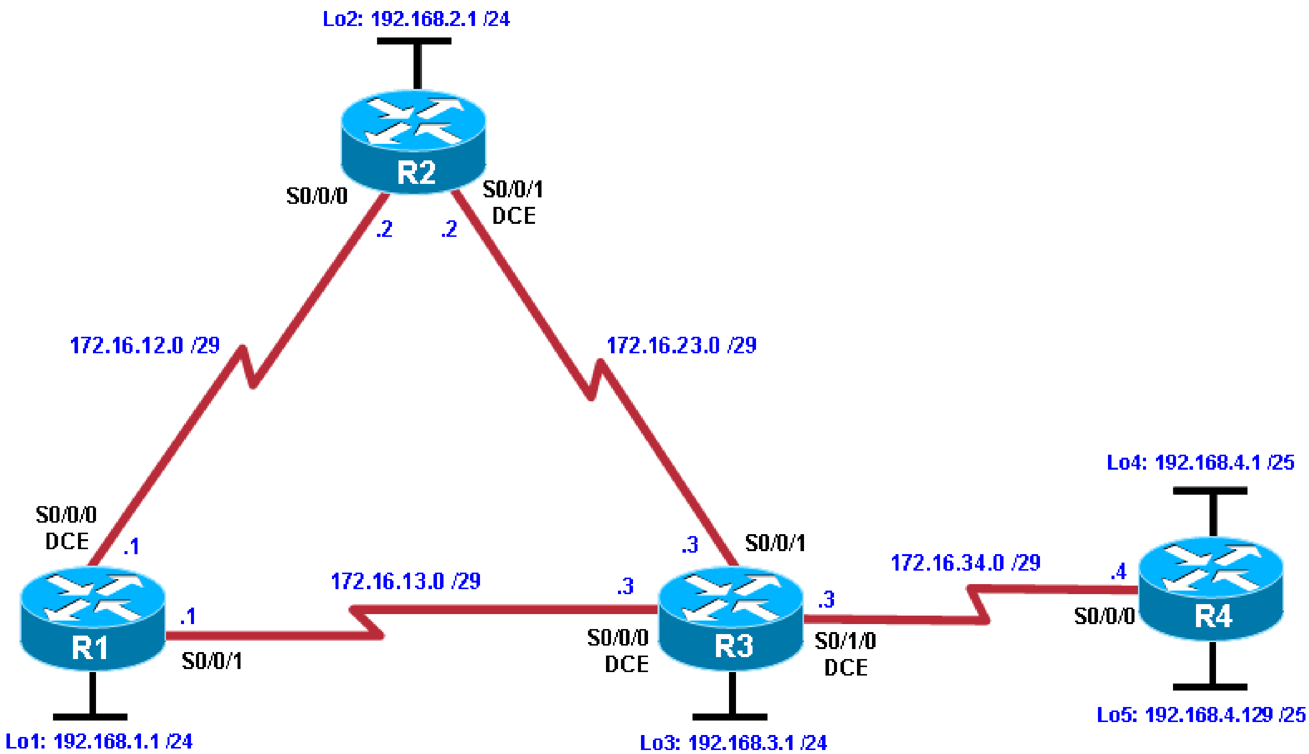
Notes:

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 5 Lab 5-1, Configure and Verify Path Control

Topology



Objectives

- Configure and verify policy-based routing.
- Select the required tools and commands to configure policy-based routing operations.
- Verify the configuration and operation by using the proper **show** and **debug** commands.

Background

You want to experiment with policy-based routing (PBR) to see how it is implemented and to study how it could be of value to your organization. To this end, you have interconnected and configured a test network with four routers. All routers are exchanging routing information using EIGRP.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1, and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router and software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers for the lab.

Cable the network as shown in the topology diagram. Erase the startup configuration, and reload each router to clear previous configurations.

Step 2: Configure router hostname and interface addresses.

a. Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to these and the serial interfaces on R1, R2, R3, and R4. On the serial interfaces connecting R1 to R3 and R3 to R4, specify the bandwidth as 64 Kb/s and set a clock rate on the DCE using the **clock rate 64000** command. On the serial interfaces connecting R1 to R2 and R2 to R3, specify the bandwidth as 128 Kb/s and set a clock rate on the DCE using the **clock rate 128000** command.

You can copy and paste the following configurations into your routers to begin.

Note: Depending on the router model, interfaces might be numbered differently than those listed. You might need to alter them accordingly.

Router R1

```
hostname R1
!
interface Lo1
 description R1 LAN
 ip address 192.168.1.1 255.255.255.0
!
interface Serial0/0/0
 description R1 --> R2
 ip address 172.16.12.1 255.255.255.248
 clock rate 128000
 bandwidth 128
 no shutdown
!
interface Serial0/0/1
 description R1 --> R3
 ip address 172.16.13.1 255.255.255.248
 bandwidth 64
 no shutdown
!
end
```

Router R2

```
hostname R2
!
interface Lo2
 description R2 LAN
 ip address 192.168.2.1 255.255.255.0
!
interface Serial0/0/0
 description R2 --> R1
 ip address 172.16.12.2 255.255.255.248
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description R2 --> R3
 ip address 172.16.23.2 255.255.255.248
 clock rate 128000
```



```
 bandwidth 128
 no shutdown
!
end
```

Router R3

```
hostname R3
!
interface Lo3
 description R3 LAN
 ip address 192.168.3.1 255.255.255.0
!
interface Serial0/0/0
 description R3 --> R1
 ip address 172.16.13.3 255.255.255.248
 clock rate 64000
 bandwidth 64
 no shutdown
!
interface Serial0/0/1
 description R3 --> R2
 ip address 172.16.23.3 255.255.255.248
 bandwidth 128
 no shutdown
!
interface Serial0/1/0
 description R3 --> R4
 ip address 172.16.34.3 255.255.255.248
 clock rate 64000
 bandwidth 64
 no shutdown
!
end
```

Router R4

```
hostname R4
!
interface Lo4
 description R4 LAN A
 ip address 192.168.4.1 255.255.255.128
!
interface Lo5
 description R4 LAN B
 ip address 192.168.4.129 255.255.255.128
!
interface Serial0/0/0
 description R4 --> R3
 ip address 172.16.34.4 255.255.255.248
 bandwidth 64
 no shutdown
!
end
```

b. Verify the configuration with the **show ip interface brief**, **show protocols,** and **show interfaces description** commands. The output from router R3 is shown here as an example.

```
R3# show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        unassigned      YES manual administratively down down

FastEthernet0/1        unassigned      YES unset  administratively down down

Serial0/0/0            172.16.13.3     YES manual up                    up

Serial0/0/1            172.16.23.3     YES manual up                    up

Serial0/1/0            172.16.34.3     YES manual up                    up

Serial0/1/1            unassigned      YES unset  administratively down down

Loopback3              192.168.3.1     YES manual up                    up

R3# show protocols
Global values:
  Internet Protocol routing is enabled
FastEthernet0/0 is administratively down, line protocol is down
FastEthernet0/1 is administratively down, line protocol is down
Serial0/0/0 is up, line protocol is up
  Internet address is 172.16.13.3/29
Serial0/0/1 is up, line protocol is up
  Internet address is 172.16.23.3/29
Serial0/1/0 is up, line protocol is up
  Internet address is 172.16.34.3/29
Serial0/1/1 is administratively down, line protocol is down
Loopback3 is up, line protocol is up
  Internet address is 192.168.3.1/24

R3# show interfaces description
Interface                      Status         Protocol Description
Fa0/0                          admin down     down
Fa0/1                          admin down     down
Se0/0/0                        up             up       R3 --> R1
Se0/0/1                        up             up       R3 --> R2
Se0/1/0                        up             up       R3 --> R4
Se0/1/1                        admin down     down
Lo3                            up             up       R3 LAN
```

Step 3: Configure basic EIGRP.

a. Implement EIGRP AS 1 over the serial and loopback interfaces as you have configured it for the other EIGRP labs.

b. Advertise networks 172.16.12.0/29, 172.16.13.0/29, 172.16.23.0/29, 172.16.34.0/29, 192.168.1.0/24, 192.168.2.0/24, 192.168.3.0/24, and 192.168.4.0/24 from their respective routers.

You can copy and paste the following configurations into your routers.

Router R1

```
router eigrp 1
 network 192.168.1.0
 network 172.16.12.0 0.0.0.7
 network 172.16.13.0 0.0.0.7
 no auto-summary
```


Router R2

```
router eigrp 1
 network 192.168.2.0
 network 172.16.12.0 0.0.0.7
 network 172.16.23.0 0.0.0.7
 no auto-summary
```

Router R3

```
router eigrp 1
 network 192.168.3.0
 network 172.16.13.0 0.0.0.7
 network 172.16.23.0 0.0.0.7
 network 172.16.34.0 0.0.0.7
 no auto-summary
```

Router R4

```
router eigrp 1
 network 192.168.4.0
 network 172.16.34.0 0.0.0.7
 no auto-summary
```

You should see EIGRP neighbor relationship messages being generated.

Step 4: Verify EIGRP connectivity.

a. Verify the configuration by using the **show ip eigrp neighbors** command to check which routers have EIGRP adjacencies.

```
R1# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   172.16.13.3             Se0/0/1           12 00:00:58  127  2280  0  16
0   172.16.12.2             Se0/0/0           13 00:01:20    8  1140  0  17

R2# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   172.16.23.3             Se0/0/1           10 00:01:30   15  1140  0  17
0   172.16.12.1             Se0/0/0           11 00:01:43   14  1140  0  180

R3# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
2   172.16.34.4             Se0/1/0           10 00:02:51   27  2280  0  3
0   172.16.13.1             Se0/0/0           12 00:03:08   45  2280  0  19
1   172.16.23.2             Se0/0/1           12 00:03:13   12  1140  0  16

R4# show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   172.16.34.3             Se0/0/0           13 00:03:33   40  2280  0  15
```


Did you receive the output you expected?

____

b. Run the following Tcl script on all routers to verify full connectivity.

```
R1# tclsh

foreach address {
172.16.12.1
172.16.12.2
172.16.13.1
172.16.13.3
172.16.23.2
172.16.23.3
172.16.34.3
172.16.34.4
192.168.1.1
192.168.2.1
192.168.3.1
192.168.4.1
192.168.4.129
} { ping $address }
```

You should get ICMP echo replies for every address pinged. Make sure to run the Tcl script on each router.

Step 5: Verify the current path.

Before you configure PBR, verify the routing table on R1.

a. On R1, use the **show ip route** command. Notice the next-hop IP address for all networks discovered by EIGRP.

```
R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/29 is subnetted, 4 subnets
D       172.16.34.0 [90/41024000] via 172.16.13.3, 00:05:18, Serial0/0/1
D       172.16.23.0 [90/21024000] via 172.16.12.2, 00:05:18, Serial0/0/0
C       172.16.12.0 is directly connected, Serial0/0/0
C       172.16.13.0 is directly connected, Serial0/0/1
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/41152000] via 172.16.13.3, 00:05:06, Serial0/0/1
D       192.168.4.128 [90/41152000] via 172.16.13.3, 00:05:06, Serial0/0/1
C    192.168.1.0/24 is directly connected, Loopback1
D    192.168.2.0/24 [90/20640000] via 172.16.12.2, 00:05:18, Serial0/0/0
D    192.168.3.0/24 [90/21152000] via 172.16.12.2, 00:05:18, Serial0/0/0
```

b. On R4, use the **traceroute** command to the R1 LAN address and source the ICMP packet from R4 LAN A and LAN B.

Note: You can specify the source as the interface address (for example 192.168.4.1) or the interface designator (for example, Fa0/0).

```
R4# traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 12 msec 12 msec 16 msec
  2 172.16.23.2 20 msec 20 msec 20 msec
  3 172.16.12.1 28 msec 24 msec *

R4# traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 12 msec 12 msec 16 msec
  2 172.16.23.2 20 msec 20 msec 20 msec
  3 172.16.12.1 28 msec 24 msec *
```

Notice that the path taken for the packets sourced from the R4 LANs are going through R3 --> R2 --> R1.

Why are the R4 interfaces not using the R3 --> R1 path?

____

____

____

c. On R3, use the **show ip route** command and note that the preferred route from R3 to R1 LAN 192.168.1.0/24 is via R2 using the R3 exit interface S0/0/1.

```
R3# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/29 is subnetted, 4 subnets
C       172.16.34.0 is directly connected, Serial0/1/0
C       172.16.23.0 is directly connected, Serial0/0/1
D       172.16.12.0 [90/21024000] via 172.16.23.2, 00:15:07, Serial0/0/1
C       172.16.13.0 is directly connected, Serial0/0/0
     192.168.4.0/25 is subnetted, 2 subnets
D       192.168.4.0 [90/40640000] via 172.16.34.4, 00:14:55, Serial0/1/0
D       192.168.4.128 [90/40640000] via 172.16.34.4, 00:14:55, Serial0/1/0
D    192.168.1.0/24 [90/21152000] via 172.16.23.2, 00:15:07, Serial0/0/1
D    192.168.2.0/24 [90/20640000] via 172.16.23.2, 00:15:07, Serial0/0/1
C    192.168.3.0/24 is directly connected, Loopback3
```

d. On R3, use the **show interfaces serial 0/0/0** and **show interfaces s0/0/1** commands.

```
R3# show interfaces s0/0/0
```



```
Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  Description: R3 --> R1
  Internet address is 172.16.13.3/29
  MTU 1500 bytes, BW 64 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  CRC checking enabled
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 48 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     771 packets input, 53728 bytes, 0 no buffer
     Received 489 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     768 packets output, 54404 bytes, 0 underruns
     0 output errors, 0 collisions, 6 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up

R3# show interfaces s0/0/1
Serial0/0/1 is up, line protocol is up
  Hardware is GT96K Serial
  Description: R3 --> R2
  Internet address is 172.16.23.3/29
  MTU 1500 bytes, BW 128 Kbit/sec, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  CRC checking enabled
  Last input 00:00:00, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 1158 kilobits/sec
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     894 packets input, 65653 bytes, 0 no buffer
     Received 488 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     895 packets output, 66785 bytes, 0 underruns
     0 output errors, 0 collisions, 6 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```


Notice that the bandwidth of the serial link between R3 and R1 (S0/0/0) is set to 64 Kb/s, while the bandwidth of the serial link between R3 and R2 (S0/0/1) is set to 128 Kb/s.

e. Confirm that R3 has a valid route to reach R1 from its serial 0/0/0 interface using the **show ip eigrp topology 192.168.1.0** command.

```
R3# show ip eigrp topology 192.168.1.0
IP-EIGRP (AS 1): Topology entry for 192.168.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 21152000
  Routing Descriptor Blocks:
  172.16.23.2 (Serial0/0/1), from 172.16.23.2, Send flag is 0x0
      Composite metric is (21152000/20640000), Route is Internal
      Vector metric:
        Minimum bandwidth is 128 Kbit
        Total delay is 45000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
  172.16.13.1 (Serial0/0/0), from 172.16.13.1, Send flag is 0x0
      Composite metric is (40640000/128256), Route is Internal
      Vector metric:
        Minimum bandwidth is 64 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

As indicated, R4 has two routes to reach 192.168.1.0. However, the metric for the route to R1 (172.16.13.1) is much higher (40640000) than the metric of the route to R2 (21152000), making the route through R2 the successor route.

Step 6: Configure PBR to provide path control.

Now you will deploy source-based IP routing by using PBR. You will change a default IP routing decision based on the EIGRP-acquired routing information for selected IP source-to-destination flows and apply a different next-hop router.

Recall that routers normally forward packets to destination addresses based on information in their routing table. By using PBR, you can implement policies that selectively cause packets to take different paths based on source address, protocol type, or application type. Therefore, PBR overrides the router's normal routing behavior.

Configuring PBR involves configuring a route map with **match** and **set** commands and then applying the route map to the interface.

The steps required to implement path control include the following:

- Choose the path control tool to use. Path control tools manipulate or bypass the IP routing table. For PBR, **route-map** commands are used.
- Implement the traffic-matching configuration, specifying which traffic will be manipulated. The **match** commands are used within route maps.
- Define the action for the matched traffic using **set** commands within route maps.
- Apply the route map to incoming traffic.

As a test, you will configure the following policy on router R3:

- All traffic sourced from R4 LAN A must take the R3 --> R2 --> R1 path.
- All traffic sourced from R4 LAN B must take the R3 --> R1 path.

a. On router R3, create a standard access list called **PBR-ACL** to identify the R4 LAN B network.

```
R3(config)# ip access-list standard PBR-ACL
R3(config-std-nacl)# remark ACL matches R4 LAN B traffic
R3(config-std-nacl)# permit 192.168.4.128 0.0.0.127
R3(config-std-nacl)# exit
```

b. Create a route map called **R3-to-R1** that matches PBR-ACL and sets the next-hop interface to the R1 serial 0/0/1 interface.

```
R3(config)# route-map R3-to-R1 permit
R3(config-route-map)# match ip address PBR-ACL
R3(config-route-map)# set ip next-hop 172.16.13.1
R3(config-route-map)# exit
```

c. Apply the R3-to-R1 route map to the serial interface on R3 that receives the traffic from R4. Use the **ip policy route-map** command on interface S0/1/0.

```
R3(config)# interface s0/1/0
R3(config-if)# ip policy route-map R3-to-R1
R3(config-if)# end
```

d. On R3, display the policy and matches using the **show route-map** command.

```
R3# show route-map
route-map R3-to-R1, permit, sequence 10
  Match clauses:
    ip address (access-lists): PBR-ACL
  Set clauses:
    ip next-hop 172.16.13.1
  Policy routing matches: 0 packets, 0 bytes
```

Note: There are currently no matches because no packets matching the ACL have passed through R3 S0/1/0.

Step 7: Test the policy.

Now you are ready to test the policy configured on R3. Enable the **debug ip policy** command on R3 so that you can observe the policy decision-making in action. To help filter the traffic, first create a standard ACL that identifies all traffic from the R4 LANs.

a. On R3, create a standard ACL which identifies all of the R4 LANs.

```
R3# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)# access-list 1 permit 192.168.4.0 0.0.0.255
R3(config)# exit
```

b. Enable PBR debugging only for traffic that matches the R4 LANs.

```
R3# debug ip policy ?
  <1-199>  Access list
  dynamic  dynamic PBR
  <cr>

R3# debug ip policy 1
Policy routing debugging is on for access list 1
```

c. Test the policy from R4 with the **traceroute** command, using R4 LAN A as the source network.


```
R4# traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 0 msec 0 msec 4 msec
  2 172.16.23.2 0 msec 0 msec 4 msec
  3 172.16.12.1 4 msec 0 msec *
```

Notice the path taken for the packet sourced from R4 LAN A is still going through R3 --> R2 --> R1.

As the traceroute was being executed, router R3 should be generating the following debug output.

```
R3#
*Feb 23 06:59:20.931: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, policy rejected -- normal forwarding
*Feb 23 06:59:29.935: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, policy rejected -- normal forwarding
*Feb 23 06:59:29.939: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, policy rejected -- normal forwarding
*Feb 23 06:59:29.939: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, FIB policy rejected(no match) - normal forwarding
*Feb 23 06:59:38.943: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, FIB policy rejected(no match) - normal forwarding
*Feb 23 06:59:38.947: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, FIB policy rejected(no match) - normal forwarding
*Feb 23 06:59:38.947: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, FIB policy rejected(no match) - normal forwarding
*Feb 23 06:59:47.951: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, FIB policy rejected(no match) - normal forwarding
*Feb 23 06:59:47.955: IP: s=192.168.4.1 (Serial0/1/0), d=192.168.1.1, len
28, FIB policy rejected(no match) - normal forwarding
```

Why is the traceroute traffic not using the R3 --> R1 path as specified in the R3-to-R1 policy?

____

____

d. Test the policy from R4 with the **traceroute** command, using R4 LAN B as the source network.

```
R4# traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 12 msec 12 msec 16 msec
  2 172.16.13.1 28 msec 28 msec *
```

Now the path taken for the packet sourced from R4 LAN B is R3 --> R1, as expected.

The debug output on R3 also confirms that the traffic meets the criteria of the R3-to-R1 policy.

```
R3#
*Feb 23 07:07:46.467: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, policy match
*Feb 23 07:07:46.467: IP: route map R3-to-R1, item 10, permit
*Feb 23 07:07:46.467: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1 (Se
rial0/0/0), len 28, policy routed
*Feb 23 07:07:46.467: IP: Serial0/1/0 to Serial0/0/0 172.16.13.1
*Feb 23 07:07:55.471: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
```



```
n 28, policy match
*Feb 23 07:07:55.471: IP: route map R3-to-R1, item 10, permit
*Feb 23 07:07:55.471: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1 (Se
rial0/0/0), len 28, policy routed
*Feb 23 07:07:55.471: IP: Serial0/1/0 to Serial0/0/0 172.16.13.1
*Feb 23 07:07:55.471: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, policy match
*Feb 23 07:07:55.471: IP: route map R3-to-R1, item 10, permit
*Feb 23 07:07:55.475: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1 (Se
rial0/0/0), len 28, policy routed
*Feb 23 07:07:55.475: IP: Serial0/1/0 to Serial0/0/0 172.16.13.1
*Feb 23 07:07:55.475: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, FIB policy match
*Feb 23 07:07:55.475: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, g=
172.16.13.1, len 28, FIB policy routed
*Feb 23 07:08:04.483: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, FIB policy match
*Feb 23 07:08:04.483: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, g=
172.16.13.1, len 28, FIB policy routed
*Feb 23 07:08:04.491: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, le
n 28, FIB policy match
*Feb 23 07:08:04.491: IP: s=192.168.4.129 (Serial0/1/0), d=192.168.1.1, g=
172.16.13.1, len 28, FIB policy routed
```

e. On R3, display the policy and matches using the **show route-map** command.

```
R3# show route-map
route-map R3-to-R1, permit, sequence 10
  Match clauses:
    ip address (access-lists): PBR-ACL
  Set clauses:
    ip next-hop 172.16.13.1
  Policy routing matches: 12 packets, 384 bytes
```

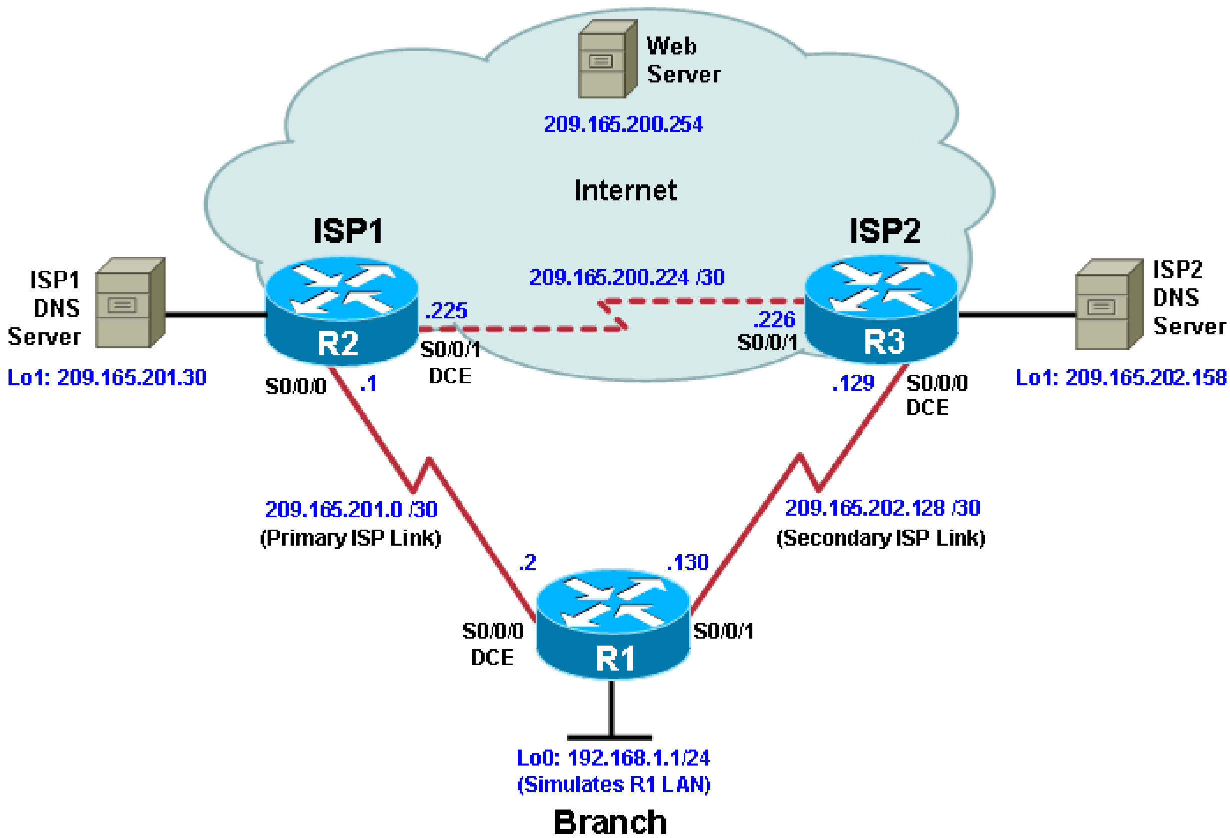
Note: There are now matches to the policy because packets matching the ACL have passed through R3 S0/1/0.

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 5 Lab 5-2, Configure IP SLA Tracking and Path Control

Topology



Objectives

- Configure and verify the IP SLA feature.
- Test the IP SLA tracking feature.
- Verify the configuration and operation using **show** and **debug** commands.

Background

You want to experiment with the Cisco IP Service Level Agreement (SLA) feature to study how it could be of value to your organization.

At times, a link to an ISP could be operational, yet users cannot connect to any other outside Internet resources. The problem might be with the ISP or downstream from them. Although policy-based routing (PBR) can be implemented to alter path control, you will implement the Cisco IOS SLA feature to monitor this behavior and intervene by injecting another default route to a backup ISP.

To test this, you have set up a three-router topology in a lab environment. Router R1 represents a branch office connected to two different ISPs. ISP1 is the preferred connection to the Internet, while ISP2 provides a backup link. ISP1 and ISP2 can also interconnect, and both can reach the web server. To monitor ISP1 for failure, you will configure IP SLA probes to track the reachability to the ISP1 DNS server. If connectivity to the ISP1 server fails, the SLA probes detect the failure and alter the default static route to point to the ISP2 server.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers and configure the router hostname and interface addresses.

a. Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear the previous configurations. Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to them as well as the serial interfaces on R1, ISP1, and ISP2.

You can copy and paste the following configurations into your routers to begin.

Note: Depending on the router model, interfaces might be numbered differently than those listed. You might need to alter them accordingly.

Router R1

```
hostname R1

interface Loopback 0
 description R1 LAN
 ip address 192.168.1.1 255.255.255.0

interface Serial0/0/0
 description R1 --> ISP1
 ip address 209.165.201.2 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description R1 --> ISP2
 ip address 209.165.202.130 255.255.255.252
 bandwidth 128
 no shutdown
```

Router ISP1 (R2)

```
hostname ISP1

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP1 DNS Server
 ip address 209.165.201.30 255.255.255.255
```



```
interface Serial0/0/0
 description ISP1 --> R1
 ip address 209.165.201.1 255.255.255.252
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description ISP1 --> ISP2
 ip address 209.165.200.225 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown
```

Router ISP2 (R3)

```
hostname ISP2

interface Loopback0
 description Simulated Internet Web Server
 ip address 209.165.200.254 255.255.255.255

interface Loopback1
 description ISP2 DNS Server
 ip address 209.165.202.158 255.255.255.255

interface Serial0/0/0
 description ISP2 --> R1
 ip address 209.165.202.129 255.255.255.252
 clock rate 128000
 bandwidth 128
 no shutdown

interface Serial0/0/1
 description ISP2 --> ISP1
 ip address 209.165.200.226 255.255.255.252
 bandwidth 128
 no shutdown
```

b. Verify the configuration by using the **show interfaces description** command. The output from router R1 is shown here as an example.

```
R1# show interfaces description
Interface                      Status         Protocol Description
Fa0/0                          admin down     down
Fa0/1                          admin down     down
Se0/0/0                        up             up       R1 --> ISP1
Se0/0/1                        up             up       R1 --> ISP2
Lo0                            up             up       R1 LAN
```

All three interfaces should be active. Troubleshoot if necessary.

c. The current routing policy in the topology is as follows:

- Router R1 establishes connectivity to the Internet through ISP1 using a default static route.
- ISP1 and ISP2 have dynamic routing enabled between them, advertising their respective public address pools.
- ISP1 and ISP2 both have static routes back to the ISP LAN.

Note: For the purpose of this lab, the ISPs have a static route to an RFC 1918 private network address on the branch router R1. In an actual branch implementation, Network Address Translation (NAT) would be configured for all traffic exiting the branch LAN. Therefore, the static routes on the ISP routers would be pointing to the provided public pool of the branch office. This is covered in Lab 7-1, “Configure Routing Facilities to the Branch Office.”

Implement the routing policies on the respective routers. You can copy and paste the following configurations.

Router R1

```
ip route 0.0.0.0 0.0.0.0 209.165.201.1
```

Router ISP1 (R2)

```
router eigrp 1
 network 209.165.200.224 0.0.0.3
 network 209.165.201.0 0.0.0.31
 no auto-summary

ip route 192.168.1.0 255.255.255.0 209.165.201.2
```

Router ISP2 (R3)

```
router eigrp 1
 network 209.165.200.224 0.0.0.3
 network 209.165.202.128 0.0.0.31
 no auto-summary

ip route 192.168.1.0 255.255.255.0 209.165.202.130
```

EIGRP neighbor relationship messages on ISP1 and ISP2 should be generated. Troubleshoot if necessary.

```
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 209.165.200.225 (Serial0/0/1) is
up: new adjacency
```

Step 2: Verify server reachability.

The Cisco IOS IP SLA feature enables an administrator to monitor network performance between Cisco devices (switches or routers) or from a Cisco device to a remote IP device. IP SLA probes continuously check the reachability of a specific destination, such as a provider edge router interface, the DNS server of the ISP, or any other specific destination, and can conditionally announce a default route only if the connectivity is verified.

a. Before implementing the Cisco IOS SLA feature, you must verify reachability to the Internet servers. From router R1, ping the web server, ISP1 DNS server, and ISP2 DNS server to verify connectivity. You can copy the following Tcl script and paste it intoR1.

```
foreach address {
209.165.200.254
209.165.201.30
209.165.202.158
} {
ping $address source 192.168.1.1
}

R1(tcl)# foreach address {
+>(tcl)# 209.165.200.254
```



```
+>(tcl)# 209.165.201.30
+>(tcl)# 209.165.202.158
+>(tcl)# } {
+>(tcl)# ping $address source 192.168.1.1
+>(tcl)#}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.254, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.201.30, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/14/16 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.202.158, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/24 ms
```

b. Trace the path taken to the web server, ISP1 DNS server, and ISP2 DNS server. You can copy the following Tcl script and paste it into R1.

```
foreach address {
209.165.200.254
209.165.201.30
209.165.202.158
} {
trace $address source 192.168.1.1
}
```

```
R1(tcl)# foreach address {
+>(tcl)# 209.165.200.254
+>(tcl)# 209.165.201.30
+>(tcl)# 209.165.202.158
+>(tcl)# } {
+>(tcl)# trace $address source 192.168.1.1
+>(tcl)# }

Type escape sequence to abort.
Tracing the route to 209.165.200.254

  1 209.165.201.1 20 msec 8 msec *
Type escape sequence to abort.
Tracing the route to 209.165.201.30

  1 209.165.201.1 8 msec 8 msec *
Type escape sequence to abort.
Tracing the route to 209.165.202.158

  1 209.165.201.1 8 msec 8 msec 4 msec
  2 209.165.200.226 8 msec 8 msec *
```

Through which ISP is traffic flowing?

____

____

Step 3: Configure IP SLA probes.

When the reachability tests are successful, you can configure the Cisco IOS IP SLAs probes. Different types of probes can be created, including FTP, HTTP, and jitter probes. In this scenario, you will configure ICMP echo probes.

a. Create an ICMP echo probe on R1 to the primary DNS server on ISP1 using the **ip sla** command.

Note: With Cisco IOS Release 12.4(4)T, 12.2(33)SB, and 12.2(33)SXI, the **ip sla** command has replaced the previous **ip sla monitor** command. In addition, the **icmp-echo** command has replaced the **type echo protocol ipIcmpEcho** command.

```
R1(config)# ip sla 11
R1(config-ip-sla)# icmp-echo 209.165.201.30
R1(config-ip-sla-echo)# frequency 10
R1(config-ip-sla-echo)# exit
R1(config)# ip sla schedule 11 life forever start-time now
```

The operation number of 11 is only locally significant to the router. The **frequency 10** command schedules the connectivity test to repeat every 10 seconds. The probe is scheduled to start now and to run forever.

b. Verify the IP SLAs configuration of operation 11 using the **show ip sla configuration 11** command.

Note: With Cisco IOS Release 12.4(4)T, 12.2(33)SB, and 12.2(33)SXI, the **show ip sla configuration** command has replaced the **show ip sla monitor configuration** command.

```
R1# show ip sla configuration 11
IP SLAs, Infrastructure Engine-II.
Entry number: 11
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.201.30/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:
```


The output lists the details of the configuration of operation 11. The operation is an ICMP echo to 209.165.201.30, with a frequency of 10 seconds, and it has already started (the start time has already passed).

c. Issue the **show ip sla statistics** command to display the number of successes, failures, and results of the latest operations.

Note: With Cisco IOS Release 12.4(4)T, 12.2(33)SB, and 12.2(33)SXI, the **show ip sla statistics** command has replaced the **show ip sla monitor statistics** command.

```
R1# show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 11

Latest operation start time: *21:22:29.707 UTC Fri Apr 2 2010
Latest operation return code: OK
Number of successes: 5
Number of failures: 0
Operation time to live: Forever
```

You can see that operation 11 has already succeeded five times, has had no failures, and the last operation returned an OK result.

d. Although not actually required because IP SLA session 11 alone could provide the desired fault tolerance, create a second probe, 22, to test connectivity to the second DNS server located on router ISP2. You can copy and paste the following commands on R1.

```
ip sla 22
icmp-echo 209.165.202.158
frequency 10
exit
ip sla schedule 22 life forever start-time now
```

e. Verify the new probe using the **show ip sla configuration** and **show ip sla statistics** commands.

```
R1# show ip sla configuration 22
IP SLAs, Infrastructure Engine-II.
Entry number: 22
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.201.158/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
```



```
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:

R1# show ip sla statistics 22
IPSLAs Latest Operation Statistics

IPSLA operation id: 22

Latest operation start time: *21:24:14.215 UTC Fri Apr 2 2010
Latest operation return code: OK
Number of successes: 4
Number of failures: 0
Operation time to live: Forever
```

The output lists the details of the configuration of operation 22. The operation is an ICMP echo to 209.165.202.158, with a frequency of 10 seconds, and it has already started (the start time has already passed). The statistics also prove that operation 22 is active.

Step 4: Configure tracking options.

Although PBR could be used, you will configure a floating static route that appears or disappears depending on the success or failure of the IP SLA.

a. Remove the current default route on R1, and replace it with a floating static route having an administrative distance of 5.

```
R1(config)# no ip route 0.0.0.0 0.0.0.0 209.165.201.1
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 5
R1(config)# exit
```

b. Verify the routing table.

```
R1# show ip route
*Apr  2 20:00:37.367: %SYS-5-CONFIG_I: Configured from console by console
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial0/0/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial0/0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [5/0] via 209.165.201.1
```

Notice that the default static route is now using the route with the administrative distance of 5. The first tracking object is tied to IP SLA object 11.

c. Use the **track 1 ip sla 11 reachability** command to enter the config-track subconfiguration mode.

Note: With Cisco IOS Release 12.4(20)T, 12.2(33)SXI1, and 12.2(33)SRE and Cisco IOS XE Release 2.4, the **track ip sla** command has replaced the **track rtr** command.

```
R1(config)# track 1 ip sla 11 reachability
R1(config-track)#
```

d. Specify the level of sensitivity to changes of tracked objects to 10 seconds of down delay and 1 second of up delay using the **delay down 10 up 1** command. The delay helps to alleviate the effect of flapping objects—objects that are going down and up rapidly. In this situation, if the DNS server fails momentarily and comes back up within 10 seconds, there is no impact.

```
R1(config-track)# delay down 10 up 1
R1(config-track)# exit
R1(config)#
```

e. Configure the floating static route that will be implemented when tracking object 1 is active. To view routing table changes as they happen, first enable the **debug ip routing** command. Next, use the **ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1** command to create a floating static default route via 209.165.201.1 (ISP1). Notice that this command references the tracking object number 1, which in turn references IP SLA operation number 11.

```
R1# debug ip routing
IP routing debugging is on
R1#
*Apr  2 21:26:46.171: RT: NET-RED 0.0.0.0/0

R1# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1
R1(config)#
*Apr  2 21:27:02.851: RT: closer admin distance for 0.0.0.0, flushing 1
routes
*Apr  2 21:27:02.851: RT: NET-RED 0.0.0.0/0
*Apr  2 21:27:02.851: RT: add 0.0.0.0/0 via 209.165.201.1, static metric
[2/0]
*Apr  2 21:27:02.851: RT: NET-RED 0.0.0.0/0
*Apr  2 21:27:02.851: RT: default path is now 0.0.0.0 via 209.165.201.1
*Apr  2 21:27:02.855: RT: new default network 0.0.0.0
*Apr  2 21:27:02.855: RT: NET-RED 0.0.0.0/0
*Apr  2 21:27:07.851: RT: NET-RED 0.0.0.0/0
```

Notice that the default route with an administrative distance of 5 has been immediately flushed because of a route with a better admin distance. It then adds the new default route with the admin distance of 2.

f. Repeat the steps for operation 22, track number 2, and assign the static route an admin distance higher than track 1 and lower than 5. On R1, copy the following configuration, which sets an admin distance of 3.

```
track 2 ip sla 22 reachability
delay down 10 up 1
exit
ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2
```

g. Verify the routing table again.

```
R1# show ip route
Codes: C - connected, S - static, 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
```



```
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial0/0/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial0/0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [2/0] via 209.165.201.1
```

Although a new default route was entered, its administrative distance is not better than 2. Therefore, it does not replace the previously entered default route.

Step 5: Verify IP SLA operation.

In this step you observe and verify the dynamic operations and routing changes when tracked objects fail. The following summarizes the process:

- Disable the DNS loopback interface on ISP1 (R2).
- Observe the output of the **debug** command on R1.
- Verify the static route entries in the routing table and the IP SLA statistics of R1.
- Re-enable the loopback interface on ISP1 (R2) and again observe the operation of the IP SLA tracking feature.

```
ISP1(config)# interface loopback 1
ISP1(config-if)# shutdown
ISP1(config-if)#
*Apr  2 15:53:14.307: %LINK-5-CHANGED: Interface Loopback1, changed state to
administratively down
*Apr  2 15:53:15.307: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Loopback1, changed state to down
```

a. Shortly after the loopback interface is administratively down, observe the debug output being generated on R1.

```
R1#
*Apr  2 21:32:33.323: %TRACKING-5-STATE: 1 ip sla 11 reachability Up->Down
*Apr  2 21:32:33.323: RT: del 0.0.0.0 via 209.165.201.1, static metric [2/0]
*Apr  2 21:32:33.323: RT: delete network route to 0.0.0.0
*Apr  2 21:32:33.323: RT: NET-RED 0.0.0.0/0
*Apr  2 21:32:33.323: RT: NET-RED 0.0.0.0/0
*Apr  2 21:32:33.323: RT: add 0.0.0.0/0 via 209.165.202.129, static metric
[3/0]
*Apr  2 21:32:33.323: RT: NET-RED 0.0.0.0/0
*Apr  2 21:32:33.323: RT: default path is now 0.0.0.0 via 209.165.202.129
*Apr  2 21:32:33.323: RT: new default network 0.0.0.0
*Apr  2 21:32:33.327: RT: NET-RED 0.0.0.0/0
*Apr  2 21:32:46.171: RT: NET-RED 0.0.0.0/0
```

The tracking state of track 1 changes from up to down. This is the object that tracked reachability for IP SLA object 11, with an ICMP echo to the ISP1 DNS server at 209.165.201.30.

R1 then proceeds to delete the default route with the administrative distance of 2 and installs the next highest default route to ISP2 with the administrative distance of 3.

b. Verify the routing table.

```
R1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.202.129 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial0/0/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial0/0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [3/0] via 209.165.202.129
```

The new static route has an administrative distance of 3 and is being forwarded to ISP2 as it should.

c. Verify the IP SLA statistics.

```
R1# show ip sla statistics
IPSLAs Latest Operation Statistics

PSLA operation id: 11
Type of operation: icmp-echo
        Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: *15:36:42.871 UTC Fri Apr 2 2010
Latest operation return code: No connection
Number of successes: 84
Number of failures: 13
Operation time to live: Forever


IPSLA operation id: 22
Type of operation: icmp-echo
        Latest RTT: 8 milliseconds
Latest operation start time: *15:36:46.335 UTC Fri Apr 2 2010
Latest operation return code: OK
Number of successes: 81
Number of failures: 1
Operation time to live: Forever
```

Notice that the latest return code is **No connection** and there have been 12 failures on IP SLA object 11.

d. Initiate a trace to the web server from the internal LAN IP address.

```
R1# trace 209.165.200.254 source 192.168.1.1

Type escape sequence to abort.
Tracing the route to 209.165.200.254

  1 209.165.202.129 8 msec 8 msec *
```

This confirms that traffic is leaving router R1 and being forwarded to the ISP2 router.

e. To examine the routing behavior when connectivity to the ISP1 DNS is restored, re-enable the DNS address on ISP1 (R2) by issuing the **no shutdown** command on the loopback 1 interface on ISP2.

```
ISP1(config-if)# no shutdown
*Apr  2 15:56:24.655: %LINK-3-UPDOWN: Interface Loopback1, changed state to
up
*Apr  2 15:56:25.655: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Loopback1, changed state to up
```

Notice the output of the **debug ip routing** command on R1.

```
R1#
*Apr  2 21:35:34.327: %TRACKING-5-STATE: 1 ip sla 11 reachability Down->Up
*Apr  2 21:35:34.327: RT: closer admin distance for 0.0.0.0, flushing 1
routes
*Apr  2 21:35:34.327: RT: NET-RED 0.0.0.0/0
*Apr  2 21:35:34.327: RT: add 0.0.0.0/0 via 209.165.201.1, static metric
[2/0]
*Apr  2 21:35:34.327: RT: NET-RED 0.0.0.0/0
*Apr  2 21:35:34.327: RT: default path is now 0.0.0.0 via 209.165.201.1
*Apr  2 21:35:34.327: RT: new default network 0.0.0.0
*Apr  2 21:35:34.327: RT: NET-RED 0.0.0.0/0
*Apr  2 21:35:39.327: RT: NET-RED 0.0.0.0/0
*Apr  2 21:35:46.171: RT: NET-RED 0.0.0.0/0
```

Now the IP SLA 11 operation transitions back to an up state and reestablishes the default static route to ISP1 with an administrative distance of 2.

f. Again examine the IP SLA statistics.

```
R1# show ip sla statistics
IPSLAs Latest Operation Statistics

Type of operation: icmp-echo
        Latest RTT: 8 milliseconds
Latest operation start time: *15:40:42.871 UTC Fri Apr 2 2010
Latest operation return code: OK
Number of successes: 88
Number of failures: 35
Operation time to live: Forever


IPSLA operation id: 22
Type of operation: icmp-echo
        Latest RTT: 16 milliseconds
Latest operation start time: *15:40:46.335 UTC Fri Apr 2 2010
Latest operation return code: OK
Number of successes: 105
Number of failures: 1
Operation time to live: Forever
```

The IP SLA 11 operation is active again, as indicated by the OK return code, and the number of successes is incrementing.

g. Verify the routing table.

```
R1# show ip route
Codes: C - connected, S - static, 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
```



```
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

     209.165.201.0/30 is subnetted, 1 subnets
C       209.165.201.0 is directly connected, Serial0/0/0
     209.165.202.0/30 is subnetted, 1 subnets
C       209.165.202.128 is directly connected, Serial0/0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [2/0] via 209.165.201.1
```

The default static through ISP1 with an administrative distance of 2 is reestablished.

There are many possibilities available with object tracking and Cisco IOS IP SLAs. As shown in this lab, a probe can be based on reachability, changing routing operations, and path control based on the ability to reach an object. However, Cisco IOS IP SLAs also allow paths to be changed based on network conditions such as delay, load, and other factors.

Before deploying a Cisco IOS IP SLA solution, the impact of the additional probe traffic being generated should be considered, including how that traffic affects bandwidth utilization, and congestion levels. Tuning the configuration (for example, with the **delay** and **frequency** commands) is critical to mitigate possible issues related to excessive transitions and route changes in the presence of flapping tracked objects.

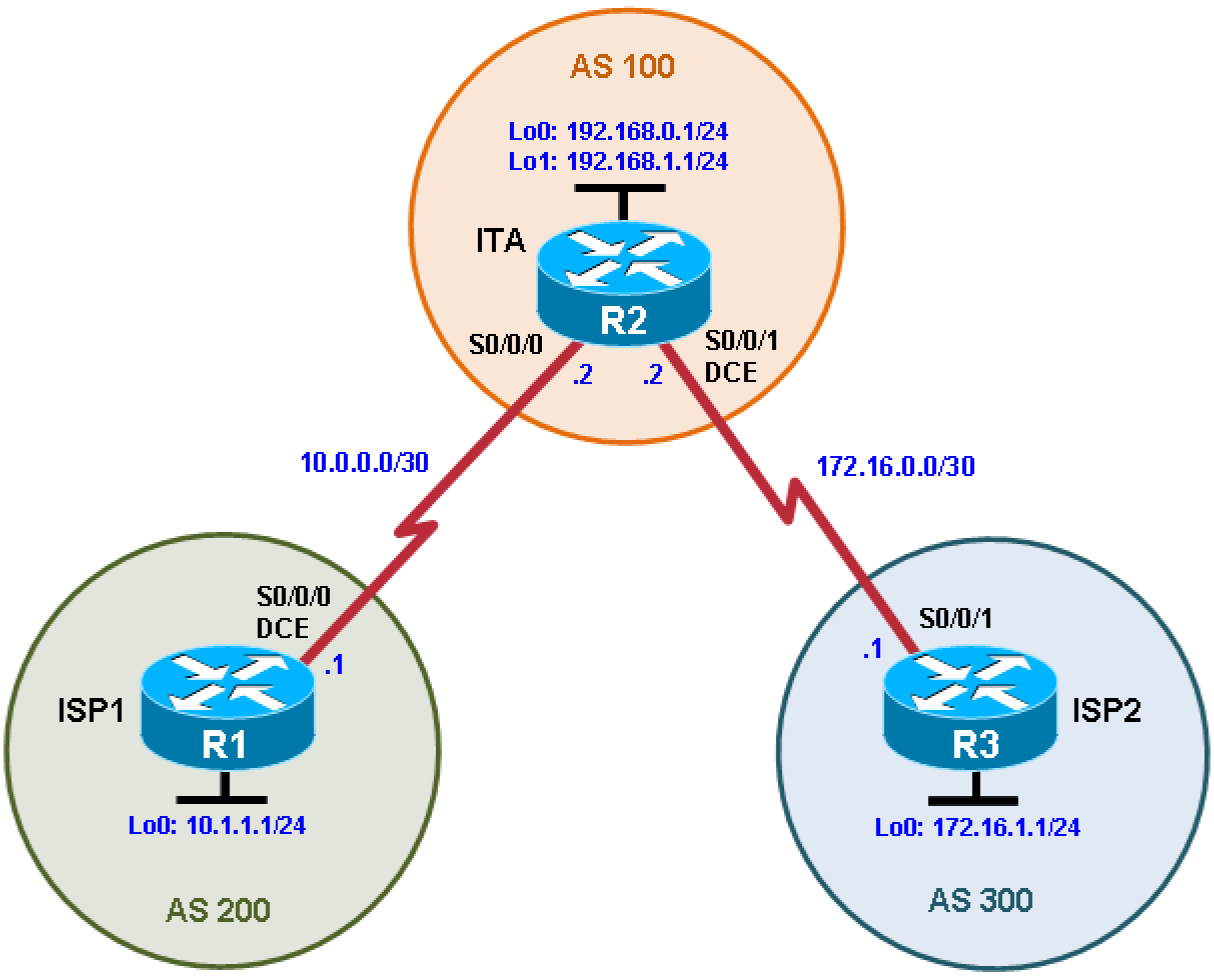
The benefits of running IP SLAs should be carefully evaluated. The IP SLA is an additional task that must be performed by the router's CPU. A large number of intensive SLAs could be a significant burden on the CPU, possibly interfering with other router functions and having detrimental impact on the overall router performance. The CPU load should be monitored after the SLAs are deployed to verify that they do not cause excessive utilization of the router CPU.

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 6 Lab 6-1, Configuring BGP with Default Routing

Topology



Objectives

- Configure BGP to exchange routing information with two ISPs.

Background

The International Travel Agency (ITA) relies extensively on the Internet for sales. For this reason, the ITA has decided to create a multihomed ISP connectivity solution and contracted with two ISPs for Internet connectivity with fault tolerance. Because the ITA is connecting to two different service providers, you must configure BGP, which runs between the ITA boundary router and the two ISP routers.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and

Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers for the lab.

Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations.

Step 2: Configure the hostname and interface addresses.

a. Assign the routers hostnames. Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to these and the serial interfaces on ISP1 (R1), ISP2 (R3), and ITA (R2). The ISP loopbacks simulate real networks that can be reached through the ISP. The two loopbacks for the ITA router simulate the connections between the ITA boundary router and their core routers. Set a clock rate on the DCE serial interfaces.

   You can copy and paste the following configurations into your routers to begin.

   Router R1 (hostname ISP1)

```
hostname ISP1
!
interface Lo0
 description ISP1 Internet Network
 ip address 10.1.1.1 255.255.255.0
!
interface Serial0/0/0
 description ISP1 -> ITA
 ip address 10.0.0.1 255.255.255.252
 clock rate 128000
 no shutdown
!
end
```

   Router R2 (hostname ITA)

```
hostname ITA
!
interface Lo0
 description Core router network link 1
 ip address 192.168.0.1 255.255.255.0
!
interface Lo1
 description Core router network link 2
 ip address 192.168.1.1 255.255.255.0
!
interface Serial0/0/0
 description ITA -> ISP1
 ip address 10.0.0.2 255.255.255.252
 no shutdown

interface Serial0/0/1
```



```
 description ITA -> ISP2
 ip address 172.16.0.2 255.255.255.252
 clock rate 128000
 no shutdown
!
end
```

Router R3 (hostname ISP2)

```
hostname ISP2
!
interface Lo0
 description ISP2 Internet Network
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0/0/1
 description ISP2 -> ITA
 ip address 172.16.0.1 255.255.255.252
 no shutdown
!
end
```

b. Use **ping** to test the connectivity between the directly connected routers. Note that router ISP1 cannot reach router ISP2.

Step 3: Configure BGP on the ISP routers.

On the ISP1 and ISP2 routers, configure BGP to peer with the ITA boundary router and advertise the ISP loopback networks.

```
ISP1(config)# router bgp 200
ISP1(config-router)# neighbor 10.0.0.2 remote-as 100
ISP1(config-router)# network 10.1.1.0 mask 255.255.255.0

ISP2(config)# router bgp 300
ISP2(config-router)# neighbor 172.16.0.2 remote-as 100
ISP2(config-router)# network 172.16.1.0 mask 255.255.255.0
```

Step 4: Configure BGP on the ITA boundary router.

a. Configure the ITA router to run BGP with both Internet providers.

```
ITA(config)# router bgp 100
ITA(config-router)# neighbor 10.0.0.1 remote-as 200
ITA(config-router)# neighbor 172.16.0.1 remote-as 300
ITA(config-router)# network 192.168.0.0
ITA(config-router)# network 192.168.1.0
```

You should see BGP neighbor peering messages on the console similar to the following.

```
*Mar  4 14:07:38.667: %BGP-5-ADJCHANGE: neighbor 10.0.0.2 Up
```

b. To verify the configuration, check the ITA routing table with the **show ip route** command.

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

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
```

```
B       172.16.1.0/24 [20/0] via 172.16.0.1, 00:00:32
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 10.0.0.1, 00:00:31
C       10.0.0.0/30 is directly connected, Serial0/0/0
C    192.168.0.0/24 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Loopback1
```

ITA has routes to the loopback networks at each ISP router.

c. Run the following Tcl script on all routers to verify connectivity If these pings are not successful, troubleshoot.

Note: The WAN subnets connecting ITA (R2) to the ISPs (R1 and R3) are not advertised in BGP, so the ISPs will not be able to ping each other's serial interface address.

```
ITA# tclsh

foreach address {
10.0.0.1
10.0.0.2
10.1.1.1
172.16.0.1
172.16.0.2
172.16.1.1
192.168.0.1
192.168.1.1
} {
ping $address }
```

Step 5: Verify BGP on the routers.

a. To verify the BGP operation on ITA, issue the **show ip bgp** command.

```
ITA# show ip bgp
BGP table version is 5, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      10.0.0.1                 0             0 200 i
*> 172.16.1.0/24    172.16.0.1               0             0 300 i
*> 192.168.0.0      0.0.0.0                  0         32768 i
*> 192.168.1.0      0.0.0.0                  0         32768 i
```

What is the local router ID?

____

Which table version is displayed?

____

An asterisk (*) next to a route indicates that it is valid. An angle bracket (>) indicates that the route has been selected as the best route.

b. To verify the operation of ISP1, issue the **show ip bgp** command.

```
ISP1# show ip bgp
BGP table version is 5, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
```



```
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
*> 172.16.1.0/24    10.0.0.2                               0 100 300 i
*> 192.168.0.0      10.0.0.2                 0             0 100 i
*> 192.168.1.0      10.0.0.2                 0             0 100 i
```

From ISP1, what is the path to network 172.16.1.0/24?

____

c. On the ISP1 router, issue the **shutdown** command on Loopback0. Then on ITA, issue the **show ip bgp** command again.

```
ITA# show ip bgp
BGP table version is 6, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.1.0/24    172.16.0.1               0             0 300 i
*> 192.168.0.0      0.0.0.0                  0         32768 i
*> 192.168.1.0      0.0.0.0                  0         32768 i
```

Which table version is displayed? Why?

____

____

What happened to the route for network 10.1.1.0/24?

____

d. Bring ISP1 router Loopback0 back up by issuing the **no shutdown** command.

e. On ITA, issue the **show ip bgp neighbors** command. The following is a partial sample output of the command showing neighbor 172.16.0.1.

```
BGP neighbor is 172.16.0.1,  remote AS 300, external link
  BGP version 4, remote router ID 172.16.1.1
  BGP state = Established, up for 00:16:00
  Last read 00:00:54, last write 00:00:43, hold time is 180, keepalive
interval
is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    New ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                5          1
    Keepalives:            15         17
```



```
    Route Refresh:           0          0
    Total:                  21         19
  Default minimum time between advertisement runs is 30 seconds

<output omitted>
```

Based on the output of this command, what is the BGP state between this router and ISP2?

____

____

How long has this connection been up?

____

Step 6: Configure route filters.

a. Check the ISP2 routing table using the **show ip route** command. ISP2 should have a route that belongs to ISP1, network 10.1.1.0.

```
ISP2# show ip route
<output omitted>

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
C       172.16.1.0/24 is directly connected, Loopback0
     10.0.0.0/24 is subnetted, 1 subnets
B       10.1.1.0 [20/0] via 172.16.0.2, 00:05:22
B    192.168.0.0/24 [20/0] via 172.16.0.2, 00:17:45
B    192.168.1.0/24 [20/0] via 172.16.0.2, 00:17:45
```

If ITA advertises a route belonging to ISP1, ISP2 installs that route in its table. ISP2 might then attempt to route transit traffic through the ITA. Configure the ITA router so that it advertises only ITA networks 192.168.0.0 and 192.168.1.0 to both providers.

b. On the ITA router, configure the following access list.

```
ITA(config)# access-list 1 permit 192.168.0.0 0.0.1.255
```

c. Apply this access list as a route filter using the **distribute-list** keyword with the BGP **neighbor** statement.

```
ITA(config)# router bgp 100
ITA(config-router)# neighbor 10.0.0.1 distribute-list 1 out
ITA(config-router)# neighbor 172.16.0.1 distribute-list 1 out
```

d. Check the routing table for ISP2 again. The route to 10.1.1.0, ISP1, should still be in the table.

e. Return to ITA and issue the **clear ip bgp *** command. Wait until the routers reach the established state, which might take several seconds, and then recheck the ISP2 routing table. The route to ISP1, network 10.1.1.0, should no longer be in the routing table for ISP2, and the route to ISP2, network 172.16.1.0, should not be in the routing table for ISP1.

```
ITA# clear ip bgp *
ITA#
*Mar  4 14:45:28.091: %BGP-5-ADJCHANGE: neighbor 10.0.0.1 Down User reset
*Mar  4 14:45:28.091: %BGP-5-ADJCHANGE: neighbor 172.16.0.1 Down User reset
*Mar  4 14:45:31.151: %BGP-5-ADJCHANGE: neighbor 172.16.0.1 Up
*Mar  4 14:45:47.095: %BGP-5-ADJCHANGE: neighbor 10.0.0.1 Up
```

Note: The **clear ip bgp *** command is disruptive because it completely resets all BGP adjacencies. This is acceptable in a lab environment but could be problematic in a production network. Instead, if only a change of inbound/outbound routing policies is to be performed, it is sufficient to issue the **clear ip bgp * in** or **clear ip bgp * out** commands**.** These commands perform only a new BGP database synchronization without the disruptive effects of a complete BGP adjacency reset. All current Cisco IOS versions support the route refresh capability that replaces the inbound soft reconfiguration feature that previously had to be configured on a per-neighbor basis.

```
ISP2# show ip route
<output omitted>

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
C       172.16.1.0/24 is directly connected, Loopback0
B    192.168.0.0/24 [20/0] via 172.16.0.2, 00:02:13
B    192.168.1.0/24 [20/0] via 172.16.0.2, 00:02:13

ISP1# show ip route
<output omitted>

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.1.0/24 is directly connected, Loopback0
C       10.0.0.0/30 is directly connected, Serial0/0/0
B    192.168.0.0/24 [20/0] via 10.0.0.2, 00:05:06
B    192.168.1.0/24 [20/0] via 10.0.0.2, 00:05:06
```

Step 7: Configure primary and backup routes using floating static routes.

With bidirectional communication established with each ISP via BGP, configure the primary and backup routes. This can be done with floating static routes or BGP.

a. Issue the **show ip route** command on the ITA router.

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

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
B       172.16.1.0/24 [20/0] via 172.16.0.1, 00:06:58
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 10.0.0.1, 00:06:58
C       10.0.0.0/30 is directly connected, Serial0/0/0
C    192.168.0.0/24 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Loopback1
```

Notice that there is no gateway of last resort defined. This is a problem because ITA is the border router for the corporate network.

b. Configure static routes to reflect the policy that ISP1 is the primary provider and that ISP2 acts as the backup by specifying a lower distance metric for the route to ISP1 (210) as compared to the backup route to ISP2 (distance metric 220).

```
ITA(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.1 210
ITA(config)# ip route 0.0.0.0 0.0.0.0 172.16.0.1 220
```

c. Verify that a default route is defined using the **show ip route** command.

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



```
Gateway of last resort is 10.0.0.1 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
B       172.16.1.0/24 [20/0] via 172.16.0.1, 00:11:41
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 10.0.0.1, 00:11:41
C       10.0.0.0/30 is directly connected, Serial0/0/0
C    192.168.0.0/24 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Loopback1
S*   0.0.0.0/0 [210/0] via 10.0.0.1
```

d. Test this default route by creating an unadvertised loopback on the router for ISP1.

```
ISP1# config t
ISP1(config)# interface loopback 100
ISP1(config-if)# ip address 192.168.100.1 255.255.255.0
```

e. Issue the **show ip route** command to ensure that the newly added 192.168.100.0 /24 network does not appear in the routing table.

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

Gateway of last resort is 10.0.0.1 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
B       172.16.1.0/24 [20/0] via 172.16.0.1, 00:16:24
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 10.0.0.1, 00:01:47
C       10.0.0.0/30 is directly connected, Serial0/0/0
C    192.168.0.0/24 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Loopback1
S*   0.0.0.0/0 [210/0] via 10.0.0.1
```

f. In extended ping mode, ping the ISP1 loopback 1 interface 192.168.100.1 with the source originating from the ITA loopback 1 interface 192.168.1.1.

```
ITA# ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

Note: You can bypass extended ping prompted mode and ping while specifying a source address using one of these abbreviated commands:


```
ITA# ping 192.168.100.1 source 192.168.1.1
```

or

```
ITA# ping 192.168.100.1 source Lo1
```

Note: Testing the default route by creating an unadvertised network on ISP1 and pinging it works only because the default route also points toward ISP1. If the preferred default route pointed toward ISP2, the ping to that unadvertised network on ISP1 would not succeed. If the link to ISP1 failed, the default route to ISP2 would become active, but the pings would be successful only if ISP1 and ISP2 have another working interconnection and appropriate BGP peering between them, which is currently not the case.

Step 8: Configure primary and backup routes using a default network and a static route.

Another method for configuring primary and backup routes is to use the **ip default-network** command instead of a 0.0.0.0/0 route.

a. Remove the floating static routes configured in Step 7.

```
ITA(config)# no ip route 0.0.0.0 0.0.0.0 10.0.0.1 210
ITA(config)# no ip route 0.0.0.0 0.0.0.0 172.16.0.1 220
```

b. The network that was added in Step 7, 192.168.100.0/24, should now be advertised on the ISP1 router. You might need to wait a few moments for BGP to advertise the new network.

```
ISP1(config)# router bgp 200
ISP1(config-router)# network 192.168.100.0
ISP1(config-router)# end
```

c. Make sure that the classful network 192.168.100.0 /24 appears in the ITA routing table.

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

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
B       172.16.1.0/24 [20/0] via 172.16.0.1, 00:30:10
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 10.0.0.1, 00:02:33
C       10.0.0.0/30 is directly connected, Serial0/0/0
C    192.168.0.0/24 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Loopback1
B    192.168.100.0/24 [20/0] via 10.0.0.1, 00:02:33
```

d. On the ITA router, configure the **ip default-network** statement to reestablish a gateway of last resort.

```
ITA(config)# ip default-network 192.168.100.0
```

Note: The behavior of this command is oriented toward legacy classful protocols and should be used only with classful networks.

e. Wait a few moments and then reexamine the routing table on ITA.

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

Gateway of last resort is 10.0.0.1 to network 192.168.100.0

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
```



```
B       172.16.1.0/24 [20/0] via 172.16.0.1, 00:32:55
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 10.0.0.1, 00:05:19
C       10.0.0.0/30 is directly connected, Serial0/0/0
C    192.168.0.0/24 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Loopback1
B*   192.168.100.0/24 [20/0] via 10.0.0.1, 00:05:19
```

This establishes ISP1 as the only default route.

f. Make ISP2 the backup ISP by adding a backup route on ITA to ISP2 serial 0/0/1 interface 172.16.0.1.

```
ITA(config)# ip route 0.0.0.0 0.0.0.0 172.16.0.1 220
```

EBGP learned routes have an administrative distance of 20 and are preferred over any routes with an administrative distance greater than 20, such as the default route defined above with an administrative distance of 220. The default route acts as a backup if the 192.168.100.0 /24 network is unavailable because of a fault or misconfiguration, or during the short period after a **clear ip bgp 10.0.0.1** command is issued.

g. Verify that this newly added route establishes a consistent default route while the BGP conversation between ITA and ISP1 reestablishes. Notice that the routing table includes two candidate default routes (*), only one of which is used because of different administrative distances.

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

Gateway of last resort is 10.0.0.1 to network 192.168.100.0

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
B       172.16.1.0/24 [20/0] via 172.16.0.1, 00:35:42
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 10.0.0.1, 00:08:05
C       10.0.0.0/30 is directly connected, Serial0/0/0
C    192.168.0.0/24 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Loopback1
B*   192.168.100.0/24 [20/0] via 10.0.0.1, 00:08:05
S*   0.0.0.0/0 [220/0] via 172.16.0.1

ITA# clear ip bgp 10.0.0.1
ITA# show ip route
<output omitted>

Gateway of last resort is 172.16.0.1 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
B       172.16.1.0/24 [20/0] via 172.16.0.1, 00:36:46
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 10.0.0.1, 00:00:00
C       10.0.0.0/30 is directly connected, Serial0/0/0
C    192.168.0.0/24 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Loopback1
B    192.168.100.0/24 [20/0] via 10.0.0.1, 00:00:00
S*   0.0.0.0/0 [220/0] via 172.16.0.1

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



```
Gateway of last resort is 10.0.0.1 to network 192.168.100.0

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.0.0/30 is directly connected, Serial0/0/1
B       172.16.1.0/24 [20/0] via 172.16.0.1, 00:38:05
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
B       10.1.1.0/24 [20/0] via 10.0.0.1, 00:01:19
C       10.0.0.0/30 is directly connected, Serial0/0/0
C    192.168.0.0/24 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Loopback1
B*   192.168.100.0/24 [20/0] via 10.0.0.1, 00:01:19
S*   0.0.0.0/0 [220/0] via 172.16.0.1
```

As expected, while the BGP conversation was down between ITA and ISP1, the route to ISP2 was added as the gateway of last resort. However, when BGP reestablished the conversation between ITA and ISP1, the default route of 192.168.100.0 was again set as the gateway of last resort on ITA.

h. Run the following Tcl script on router ITA to verify connectivity.

```
ITA# tclsh

foreach address {
10.0.0.1
10.0.0.2
10.1.1.1
172.16.0.1
172.16.0.2
172.16.1.1
192.168.0.1
192.168.1.1
192.168.100.1
} {
ping $address }
```

Should ISP1 and ISP2 be able to ping all networks in the topology?

____

____

____

____

____

Note: Another option for setting up default routing is to inject a default route via BGP. The following example configures the ISP1 router to inject a default route to itself that can be used by the ITA router:

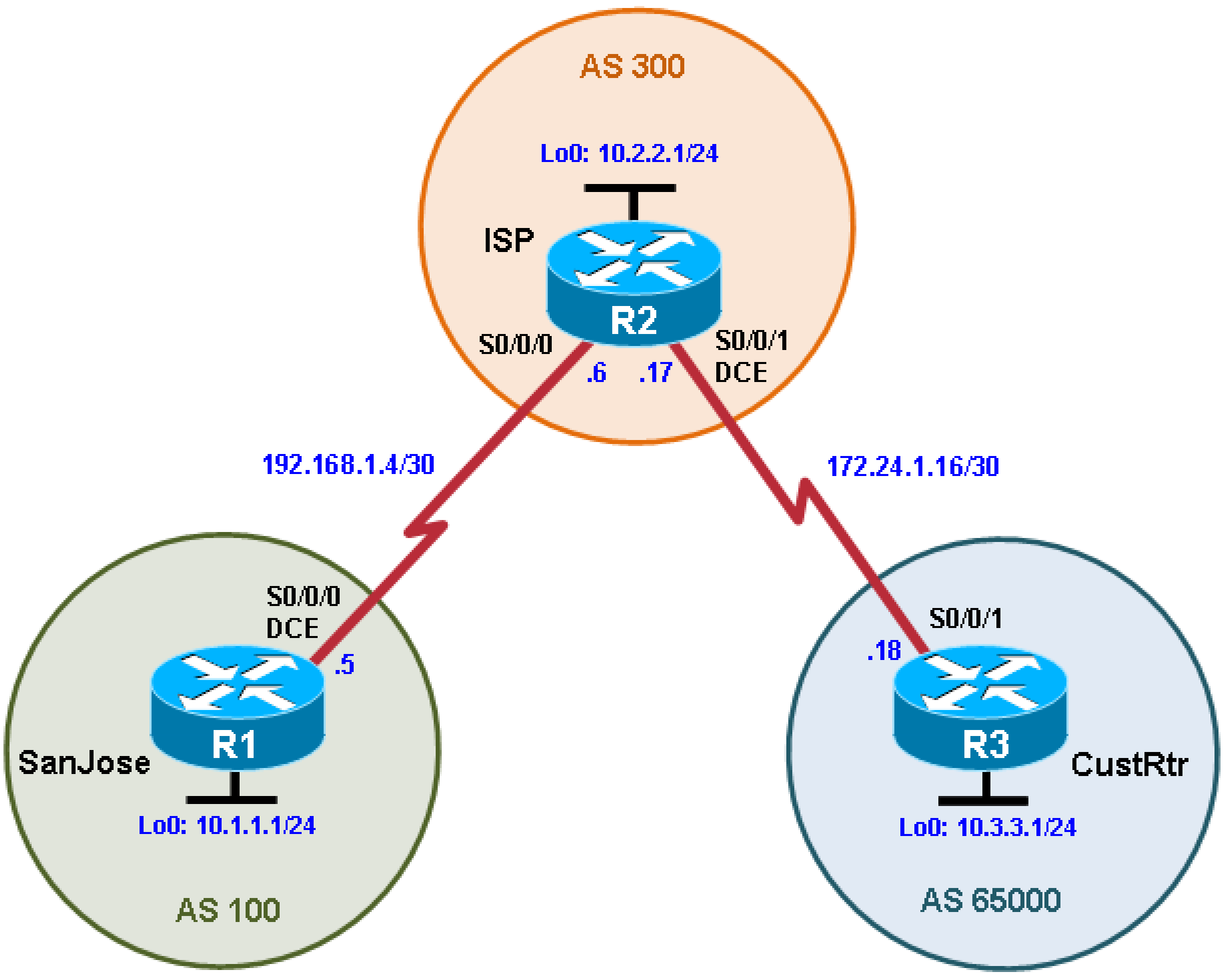
```
ISP1(config)# router bgp 200
ISP1(config-router)# neighbor 10.0.0.2 default-originate
```


Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 6 Lab 6-2, Using the AS_PATH Attribute

Topology



Objectives

- Use BGP commands to prevent private AS numbers from being advertised to the outside world.
- Use the AS_PATH attribute to filter BGP routes based on their source AS numbers.

Background

The International Travel Agency's ISP has been assigned an AS number of 300. This provider uses BGP to exchange routing information with several customer networks. Each customer network is assigned an AS number from the private range, such as AS 65000. Configure the ISP router to remove the private AS numbers from the AS Path information of CustRtr. In addition, the ISP would like to prevent its customer networks from receiving route information from International Travel Agency's AS 100. Use the AS_PATH attribute to implement this policy.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as 2801 or 2811) and Cisco IOS Software versions, if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers for the lab.

Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations.

Step 2: Configure the hostname and interface addresses.

a. You can copy and paste the following configurations into your routers to begin.

Router R1 (hostname SanJose)

```
hostname SanJose
!
interface Loopback0
 ip address 10.1.1.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.5 255.255.255.252
 clock rate 128000
 no shutdown
```

Router R2 (hostname ISP)

```
hostname ISP
!
interface Loopback0
 ip address 10.2.2.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.6 255.255.255.252
 no shutdown
!
interface Serial0/0/1
 ip address 172.24.1.17 255.255.255.252
 clock rate 128000
 no shutdown
```

Router R3 (hostname CustRtr)

```
hostname CustRtr
!
interface Loopback0
 ip address 10.3.3.1 255.255.255.0
!
interface Serial0/0/1
 ip address 172.24.1.18 255.255.255.252
 no shutdown
```


b. Use **ping** to test the connectivity between the directly connected routers.

Note: SanJose will not be able to reach either ISP's loopback (10.2.2.1) or CustRtr's loopback (10.3.3.1), nor will it be able to reach either end of the link joining ISP to CustRtr (172.24.1.17 and 172.24.1.18).

Step 3: Configure BGP.

a. Configure BGP for normal operation. Enter the appropriate BGP commands on each router so that they identify their BGP neighbors and advertise their loopback networks.

```
SanJose(config)# router bgp 100
SanJose(config-router)# neighbor 192.168.1.6 remote-as 300
SanJose(config-router)# network 10.1.1.0 mask 255.255.255.0

ISP(config)# router bgp 300
ISP(config-router)# neighbor 192.168.1.5 remote-as 100
ISP(config-router)# neighbor 172.24.1.18 remote-as 65000
ISP(config-router)# network 10.2.2.0 mask 255.255.255.0

CustRtr(config)# router bgp 65000
CustRtr(config-router)# neighbor 172.24.1.17 remote-as 300
CustRtr(config-router)# network 10.3.3.0 mask 255.255.255.0
```

b. Verify that these routers have established the appropriate neighbor relationships by issuing the **show ip bgp neighbors** command on each router.

```
ISP# show ip bgp neighbors
BGP neighbor is 172.24.1.18,  remote AS 65000, external link
  BGP version 4, remote router ID 10.3.3.1
  BGP state = Established, up for 00:02:05
<output omitted>

BGP neighbor is 192.168.1.5,  remote AS 100, external link
  BGP version 4, remote router ID 10.1.1.1
  BGP state = Established, up for 00:04:19
<output omitted>
```

Step 4: Remove the private AS.

a. Display the SanJose routing table using the **show ip route** command. SanJose should have a route to both 10.2.2.0 and 10.3.3.0. Troubleshoot if necessary.

```
SanJose# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 3 subnets
B       10.3.3.0 [20/0] via 192.168.1.6, 00:01:11
B       10.2.2.0 [20/0] via 192.168.1.6, 00:02:16
C       10.1.1.0 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 1 subnets
```



```
C       192.168.1.4 is directly connected, Serial0/0/0
```

b. Ping the 10.3.3.1 address from SanJose.

Why does this fail?

____

____

c. Ping again, this time as an extended ping, sourcing from the Loopback0 interface address.

```
SanJose# ping
Protocol [ip]:
Target IP address: 10.3.3.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/68 ms
```

Note: You can bypass extended ping mode and specify a source address using one of these commands:

```
SanJose# ping 10.3.3.1 source 10.1.1.1
```

or

```
SanJose# ping 10.3.3.1 source Lo0
```

d. Check the BGP table from SanJose by using the **show ip bgp** command. Note the AS path for the 10.3.3.0 network. The AS 65000 should be listed in the path to 10.3.3.0.

```
SanJose# show ip bgp

BGP table version is 5, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal   Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0         0.0.0.0                  0         32768 i
*> 10.2.2.0         192.168.1.6              0             0 300 i
*> 10.3.3.0         192.168.1.6                            0 300 65000 i
```

Why is this a problem?

____

____

e. Configure ISP to strip the private AS numbers from BGP routes exchanged with SanJose using the following commands.

```
ISP(config)# router bgp 300
```



```
ISP(config-router)# neighbor 192.168.1.5 remove-private-as
```

f. After issuing these commands, use the **clear ip bgp *** command on ISP to reestablish the BGP relationship between the three routers. Wait several seconds and then return to SanJose to check its routing table.

Note: The **clear ip bgp * soft** command can also be used to force each router to resend its BGP table.

Does SanJose still have a route to 10.3.3.0?

____

SanJose should be able to ping 10.3.3.1 using its loopback 0 interface as the source of the ping.

```
SanJose# ping 10.3.3.1 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
```

g. Now check the BGP table on SanJose. The AS_ PATH to the 10.3.3.0 network should be AS 300. It no longer has the private AS in the path.

```
SanJose# show ip bgp

BGP table version is 8, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal   Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0         0.0.0.0                  0         32768 i
*> 10.2.2.0         192.168.1.6              0             0 300 i
*> 10.3.3.0         192.168.1.6                            0 300 i
```

Step 5: Use the AS_PATH attribute to filter routes.

As a final configuration, use the AS_PATH attribute to filter routes based on their origin. In a complex environment, you can use this attribute to enforce routing policy. In this case, the provider router, ISP, must be configured so that it does not propagate routes that originate from AS 100 to the customer router CustRtr.

AS-path access lists are read like regular access lists. The statements are read sequentially, and there is an implicit deny at the end. Rather than matching an address in each statement like a conventional access list, AS path access lists match on something called a regular expression. Regular expressions are a way of matching text patterns and have many uses. In this case, you will be using them in the AS path access list to match text patterns in AS paths.

a. Configure a special kind of access list to match BGP routes with an AS_PATH attribute that both begins and ends with the number 100. Enter the following commands on ISP.

```
ISP(config)# ip as-path access-list 1 deny ^100$
ISP(config)# ip as-path access-list 1 permit .*
```

The first command uses the **^** character to indicate that the AS path must begin with the given number 100. The **$** character indicates that the AS_PATH attribute must also end with 100. Essentially, this statement matches only paths that are sourced from AS 100. Other paths, which might include AS 100 along the way, will not match this list.

In the second statement, the **.** (period) is a wildcard, and the ***** (asterisk) stands for a repetition of the wildcard. Together, **.*** matches any value of the AS_PATH attribute, which in effect permits any update that has not been denied by the previous **access-list** statement.

For more details on configuring regular expressions on Cisco routers, see:

http://www.cisco.com/en/US/docs/ios/12_2/termserv/configuration/guide/tcfaapre_ps1835_TSD_Products_Configuration_Guide_Chapter.html

b. Apply the configured access list using the **neighbor** command with the **filter-list** option.

```
ISP(config)# router bgp 300
ISP(config-router)# neighbor 172.24.1.18 filter-list 1 out
```

The **out** keyword specifies that the list is applied to routing information sent to this neighbor.

c. Use the **clear ip bgp *** command to reset the routing information. Wait several seconds and then check the routing table for ISP. The route to 10.1.1.0 should be in the routing table.

Note: To force the local router to resend its BGP table, a less disruptive option is to use the **clear ip bgp * out** or **clear ip bgp * soft** command (the second command performs both outgoing and incoming route resync).

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

     172.24.0.0/30 is subnetted, 1 subnets
C       172.24.1.16 is directly connected, Serial0/0/1
     10.0.0.0/24 is subnetted, 3 subnets
B       10.3.3.0 [20/0] via 172.24.1.18, 00:07:34
C       10.2.2.0 is directly connected, Loopback0
B       10.1.1.0 [20/0] via 192.168.1.5, 00:10:53
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0/0
```

d. Check the routing table for CustRtr. It should not have a route to 10.1.1.0 in its routing table.

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

     172.24.0.0/30 is subnetted, 1 subnets
C       172.24.1.16 is directly connected, Serial0/0/1
     10.0.0.0/24 is subnetted, 2 subnets
C       10.3.3.0 is directly connected, Loopback0
B       10.2.2.0 [20/0] via 172.24.1.17, 00:11:57
```

e. Return to ISP and verify that the filter is working as intended. Issue the **show ip bgp regexp ^100$** command.

```
ISP# show ip bgp regexp ^100$
BGP table version is 4, local router ID is 10.2.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal   Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.1.0         192.168.1.5              0             0 100 i
```

The output of this command shows all matches for the regular expressions that were used in the access list. The path to 10.1.1.0 matches the access list and is filtered from updates to CustRtr.

f. Run the following Tcl script on all routers to verify whether there is connectivity. All pings from ISP should be successful. SanJose should not be able to ping the CustRtr loopback 10.3.3.1 or the WAN link

172.24.1.16/30. CustRtr should not be able to ping the SanJose loopback 10.1.1.1 or the WAN link 192.168.1.4/30.

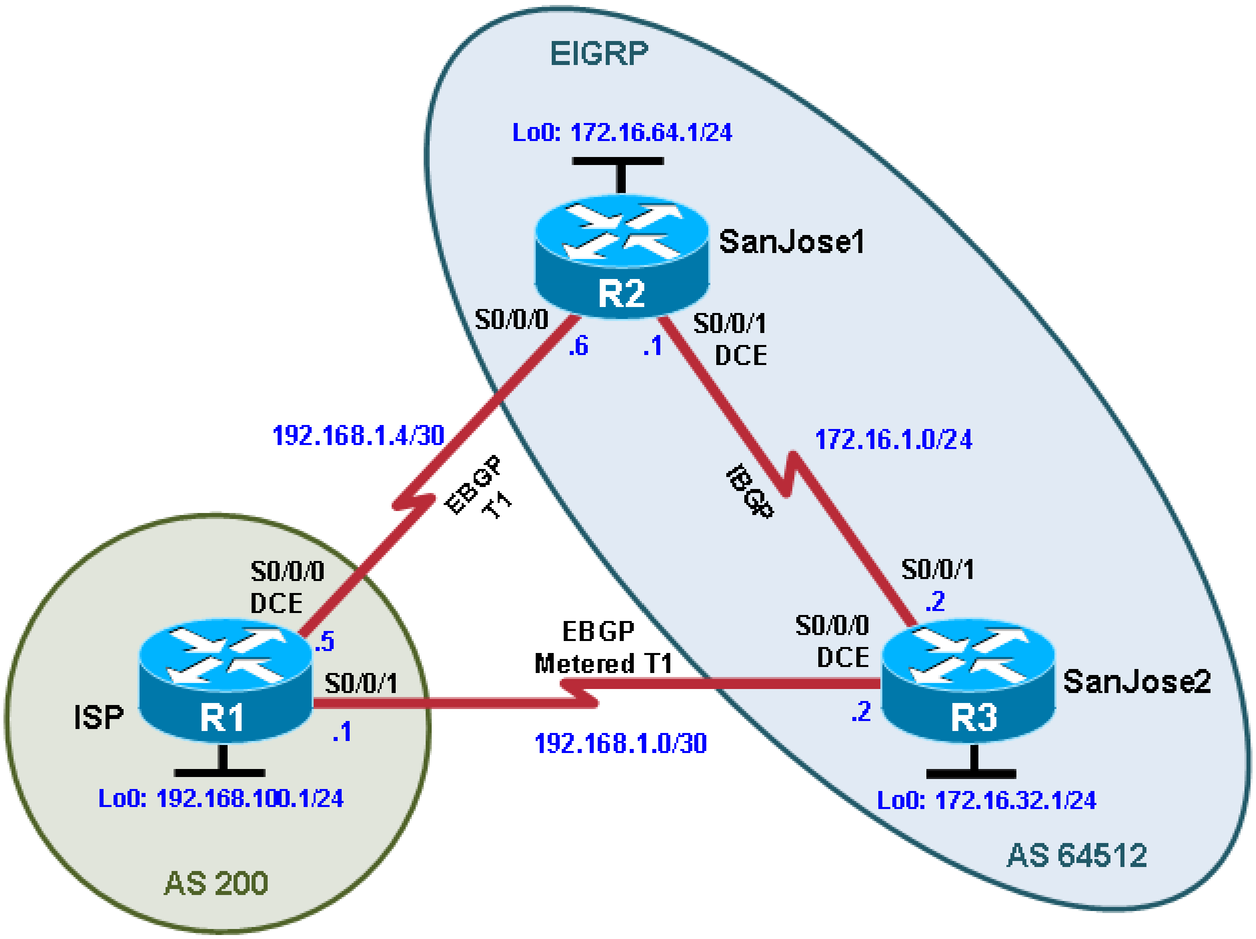
```
ISP# tclsh

foreach address {
10.1.1.1
10.2.2.1
10.3.3.1
192.168.1.5
192.168.1.6
172.24.1.17
172.24.1.18
} {
ping $address }
```


Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 6 Lab 6-3, Configuring IBGP and EBGP Sessions, Local Preference, and MED

Topology



Objectives

- For IBGP peers to correctly exchange routing information, use the **next-hop-self** command with the **Local-Preference** and **MED** attributes.
- Ensure that the flat-rate, unlimited-use T1 link is used for sending and receiving data to and from the AS 200 on ISP and that the metered T1 only be used in the event that the primary T1 link has failed.

Background

The International Travel Agency runs BGP on its SanJose1 and SanJose2 routers externally with the ISP router in AS 200. IBGP is run internally between SanJose1 and SanJose2. Your job is to configure both

EBGP and IBGP for this internetwork to allow for redundancy. The metered T1 should only be used in the event that the primary T1 link has failed. Traffic sent across the metered T1 link offers the same bandwidth of the primary link but at a huge expense. Ensure that this link is not used unnecessarily.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers for the lab.

Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations.

Step 2: Configure the hostname and interface addresses.

a. You can copy and paste the following configurations into your routers to begin.

Router R1 (hostname ISP)

```
hostname ISP
!
interface Loopback0
 ip address 192.168.100.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.5 255.255.255.252
 clock rate 128000
 no shutdown
!
interface Serial0/0/1
 ip address 192.168.1.1 255.255.255.252
 no shutdown
```

Router R2 (hostname SanJose1)

```
hostname SanJose1
!
interface Loopback0
 ip address 172.16.64.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.6 255.255.255.252
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.1.1 255.255.255.0
 clock rate 128000
 no shutdown
```

Router R3 (hostname SanJose2)

```
hostname SanJose2
!
interface Loopback0
 ip address 172.16.32.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.2 255.255.255.252
 clock rate 128000
 no shutdown
!
interface Serial0/0/1
 ip address 172.16.1.2 255.255.255.0
 no shutdown
```

b. Use **ping** to test the connectivity between the directly connected routers. Both SanJose routers should be able to ping each other and their local ISP serial link IP address. The ISP router cannot reach the segment between SanJose1 and SanJose2.

Step 3: Configure EIGRP.

Configure EIGRP between the SanJose1 and SanJose2 routers.

```
SanJose1(config)# router eigrp 64512
SanJose1(config-router)# no auto-summary
SanJose1(config-router)# network 172.16.0.0

SanJose2(config)# router eigrp 64512
SanJose2(config-router)# no auto-summary
SanJose2(config-router)# network 172.16.0.0
```

Step 4: Configure IBGP and verify BGP neighbors.

a. Configure IBGP between the SanJose1 and SanJose2 routers. On the SanJose1 router, enter the following configuration.

```
SanJose1(config)# router bgp 64512
SanJose1(config-router)# neighbor 172.16.32.1 remote-as 64512
SanJose1(config-router)# neighbor 172.16.32.1 update-source lo0
```

If multiple pathways to the BGP neighbor exist, the router can use multiple IP interfaces to communicate with the neighbor. The source IP address therefore depends on the outgoing interface. The **update-source lo0** command instructs the router to use the IP address of the interface Loopback0 as the source IP address for all BGP messages sent to that neighbor.

b. Complete the IBGP configuration on SanJose2 using the following commands.

```
SanJose2(config)# router bgp 64512
SanJose2(config-router)# neighbor 172.16.64.1 remote-as 64512
SanJose2(config-router)# neighbor 172.16.64.1 update-source lo0
```

c. Verify that SanJose1 and SanJose2 become BGP neighbors by issuing the **show ip bgp neighbors** command on SanJose1. View the following partial output. If the BGP state is not established, troubleshoot the connection.

```
SanJose2# show ip bgp neighbors
BGP neighbor is 172.16.64.1,  remote AS 64512, internal link
            BGP version 4, remote router ID 172.16.64.1
            BGP state = Established, up for 00:00:01
```



```
<output omitted>
```

The link between SanJose1 and SanJose2 should be identified as an internal link, as shown in the output.

Step 5: Configure EBGP and verify BGP neighbors.

a. Configure ISP to run EBGP with SanJose1 and SanJose2. Enter the following commands on ISP.

```
ISP(config)# router bgp 200
ISP(config-router)# neighbor 192.168.1.6 remote-as 64512
ISP(config-router)# neighbor 192.168.1.2 remote-as 64512
ISP(config-router)# network 192.168.100.0
```

Because EBGP sessions are almost always established over point-to-point links, there is no reason to use the **update-source** keyword in this configuration. Only one path exists between the peers. If this path goes down, alternative paths are not available.

b. Configure SanJose1 as an EBGP peer to ISP.

```
SanJose1(config)# ip route 172.16.0.0 255.255.0.0 null0
SanJose1(config)# router bgp 64512
SanJose1(config-router)# neighbor 192.168.1.5 remote-as 200
SanJose1(config-router)# network 172.16.0.0
```

c. Use the **show ip bgp neighbors** command to verify that SanJose1 and ISP have reached the established state. Troubleshoot if necessary.

```
SanJose1# show ip bgp neighbors

BGP neighbor is 172.16.32.1,  remote AS 64512, internal link
  BGP version 4, remote router ID 172.16.32.1
  BGP state = Established, up for 00:03:10
<output omitted>

BGP neighbor is 192.168.1.5,  remote AS 200, external link
  BGP version 4, remote router ID 192.168.100.1
  BGP state = Established, up for 00:03:10
<output omitted>
```

You should also see an informational message indicating the establishment of the BGP neighbor relationship.

```
*Mar  8 19:41:14.111: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Up
```

d. Configure SanJose2 as an EBGP peer to ISP.

```
SanJose2(config)# ip route 172.16.0.0 255.255.0.0 null0
SanJose2(config)# router bgp 64512
SanJose2(config-router)# neighbor 192.168.1.1 remote-as 200
SanJose2(config-router)# network 172.16.0.0
```

Step 6: View BGP summary output.

In Step 5, the **show ip bgp neighbors** command was used to verify that SanJose1 and ISP had reached the established state. A useful alternative command is **show ip bgp summary**. The output should be similar to the following.

```
SanJose2# show ip bgp summary

BGP router identifier 172.16.32.1, local AS number 64512
BGP table version is 2, main routing table version 2
1 network entries and 1 paths using 137 bytes of memory
```

```
1 BGP path attribute entries using 60 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 2/1 prefixes, 2/1 paths, scan interval 15 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down
State/PfxRcd
172.16.64.1     4 64512      21      24        2    0    0 00:03:02        2
192.168.1.1     4   200      14      15        2    0    0 00:03:36        1
```

Step 7: Verify which path the traffic takes.

a. Clear the IP BGP conversation with the **clear ip bgp** * command on ISP. Wait for the conversations to reestablish with each SanJose router.

b. Test whether ISP can ping the loopback 0 address of 172.16.64.1 on SanJose1 and the serial link between SanJose1 and SanJose2, 172.16.1.1.

c. Now ping from ISP to the loopback 0 address of 172.16.32.1 on SanJose2 and the serial link between SanJose1 and SanJose2, 172.16.1.2.

   You should see successful pings to each IP address on SanJose2 router. Ping attempts to 172.16.64.1 and 172.16.1.1 should fail. Why does this happen?

   ____

   ____

d. Issue the **show ip bgp** command on ISP to verify BGP routes and metrics.

```
ISP# show ip bgp

BGP table version is 3, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       192.168.1.2              0             0 64512 i
*                   192.168.1.6              0             0 64512 i
*> 192.168.100.0    0.0.0.0                  0         32768 i
```

   Notice that ISP has two valid routes to the 172.16.0.0 network, as indicated by the *. However, the link to SanJose2 has been selected as the best path. Why did the ISP prefer the link to SanJose2 over SanJose1?

   ____

   ____

   ____

   Would changing the bandwidth metric on each link help to correct this issue? Explain.

   ____

   ____

   BGP operates differently than all other protocols. Unlike other routing protocols that use complex algorithms involving factors such as bandwidth, delay, reliability, and load to formulate a metric, BGP is policy-based. BGP determines the best path based on variables, such as AS path, weight, local preference, MED, and so on. If all things are equal, BGP prefers the route leading to the BGP speaker with the lowest BGP router ID. The SanJose2 router with BGP router ID 172.16.32.1 was preferred to the higher BGP router ID of the SanJose1 router (172.16.64.1).

e. At this point, the ISP router should be able to get to each network connected to SanJose1 and SanJose2 from the loopback address 192.168.100.1. Use the extended **ping** command and specify the source address of ISP Lo0 to test.

```
ISP# ping 172.16.1.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/24 ms

ISP# ping 172.16.32.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms

ISP# ping 172.16.1.2 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/15/16 ms
ISP#

ISP# ping 172.16.64.1 source 192.168.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/24 ms
```

You can also use the extended ping dialogue to specify the source address, as shown in this example.

```
ISP# ping
Protocol [ip]:
Target IP address: 172.16.64.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.100.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/48/52 ms
```

Complete reachability has been demonstrated between the ISP router and both SanJose1 and SanJose2.

Step 8: Configure the BGP next-hop-self feature.

SanJose1 is unaware of the link between ISP and SanJose2, and SanJose2 is unaware of the link between ISP and SanJose1. Before ISP can successfully ping all the internal serial interfaces of AS 64512, these serial links should be advertised via BGP on the ISP router. This can also be resolved via EIGRP on each SanJose router. The preferred method is for ISP to advertise these links.

a. Issue the following commands on the ISP router.

```
ISP(config)# router bgp 200
ISP(config-router)# network 192.168.1.0 mask 255.255.255.252
ISP(config-router)# network 192.168.1.4 mask 255.255.255.252
```

b. Issue the **show ip bgp** command to verify that the ISP is correctly injecting its own WAN links into BGP.

```
ISP# show ip bgp
BGP table version is 5, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal Origin codes: i - IGP, e - EGP, ? - incomplete

 Network          Next Hop            Metric LocPrf   Weight Path
*> 172.16.0.0       192.168.1.2            0              0 64512 i
*                   192.168.1.6            0              0 64512 i
*> 192.168.1.0/30   0.0.0.0                0          32768 i
*> 192.168.1.4/30   0.0.0.0                0          32768 i
*> 192.168.100.0    0.0.0.0                0          32768 i
```

c. Verify on SanJose1 and SanJose2 that the opposite WAN link is included in the routing table. The output from SanJose2 is as follows.

```
SanJose2# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.32.0/24 is directly connected, Loopback0
S       172.16.0.0/16 is directly connected, Null0
C       172.16.1.0/24 is directly connected, Serial0/0/1
D       172.16.64.0/24 [90/2297856] via 172.16.1.1, 01:02:10, Serial0/0/1
     192.168.1.0/30 is subnetted, 2 subnets
C       192.168.1.0 is directly connected, Serial0/0/0
B       192.168.1.4 [20/0] via 192.168.1.1, 00:01:13
B    192.168.100.0/24 [20/0] via 192.168.1.1, 00:33:32
```

The next issue to consider is BGP policy routing between autonomous systems. The next-hop attribute of a route in a different AS is set to the IP address of the border router in the next AS toward the destination, and this attribute is not modified by default when advertising this route through IBGP. Therefore, for all IBGP peers, it is either necessary to know the route to that border router (in a different neighboring AS), or our own border router needs to advertise the foreign routes using the next-hop-self feature, overriding the next-hop address with its own IP address. The SanJose2 router is passing a policy to SanJose1 and vice versa. The policy for routing from AS 64512 to AS 200 is to forward packets to the 192.168.1.1 interface. SanJose1 has a similar yet opposite policy: it forwards requests to the 192.168.1.5 interface. If either WAN link fails, it is critical that the opposite router become a valid gateway. This is achieved if the **next-hop-self** command is configured on SanJose1 and SanJose2.

d. View the output before the **next-hop-self** command is issued.

```
SanJose2# show ip bgp
BGP table version is 11, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal  Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       0.0.0.0                  0         32768 i
* i192.168.1.0/30   192.168.1.5              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
* i192.168.1.4/30   192.168.1.5              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
* i192.168.100.0    192.168.1.5              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
```

e. Issue the **next-hop-self** command on SanJose1 and SanJose2.

```
SanJose1(config)# router bgp 64512
SanJose1(config-router)# neighbor 172.16.32.1 next-hop-self

SanJose2(config)# router bgp 64512
SanJose2(config-router)# neighbor 172.16.64.1 next-hop-self
```

f. Reset BGP operation on either router with the **clear ip bgp * soft** command.

g. After the routers have returned to established BGP speakers, issue the **show ip bgp** command to validate that the next hop has also been corrected.

```
SanJose2# show ip bgp
BGP table version is 11, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       0.0.0.0                  0         32768 i
* i192.168.1.0/30   172.16.64.1              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
* i192.168.1.4/30   172.16.64.1              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
* i192.168.100.0    172.16.64.1              0    100      0 200 i
*>                  192.168.1.1              0             0 200 i
```

Step 9: Set BGP local preference.

At this point, everything looks good, with the exception of default routes, the outbound flow of data, and inbound packet flow.

a. Because the local preference value is shared between IBGP neighbors, configure a simple route map that references the local preference value on SanJose1 and SanJose2. This policy adjusts outbound traffic to prefer the link off the SanJose1 router instead of the metered T1 off SanJose2.


```
SanJose1(config)# route-map PRIMARY_T1_IN permit 10
SanJose1(config-route-map)# set local-preference 150
SanJose1(config-route-map)# exit
SanJose1(config)# router bgp 64512
SanJose1(config-router)# neighbor 192.168.1.5 route-map PRIMARY_T1_IN in

SanJose2(config)# route-map SECONDARY_T1_IN permit 10
SanJose2(config-route-map)# set local-preference 125
SanJose1(config-route-map)# exit
SanJose2(config)# router bgp 64512
SanJose2(config-router)# neighbor 192.168.1.1 route-map SECONDARY_T1_IN in
```

b. Use the **clear ip bgp * soft** command after configuring this new policy. When the conversations have been reestablished, issue the **show ip bgp** command on SanJose1 and SanJose2.

```
SanJose1# clear ip bgp * soft
SanJose2# clear ip bgp * soft

SanJose1# show ip bgp

BGP table version is 8, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal  Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i172.16.0.0       172.16.32.1              0    100      0 i
*>                  0.0.0.0                  0         32768 i
*> 192.168.1.0/30   192.168.1.5              0    150      0 200 i
*> 192.168.1.4/30   192.168.1.5              0    150      0 200 i
*> 192.168.100.0    192.168.1.5              0    150      0 200 i

SanJose2# show ip bgp

BGP table version is 11, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal  Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0       0.0.0.0                  0         32768 i
*i                  172.16.64.1              0    100      0 i
*>i192.168.1.0/30   172.16.64.1              0    150      0 200 i
*                   192.168.1.1              0    125      0 200 i
*>i192.168.1.4/30   172.16.64.1              0    150      0 200 i
*                   192.168.1.1              0    125      0 200 i
*>i192.168.100.0    172.16.64.1              0    150      0 200 i
*                   192.168.1.1              0    125      0 200 i
```

This now indicates that routing to the loopback segment for ISP 192.168.100.0 /24 can be reached only through the link common to SanJose1 and ISP.

Step 10: Set BGP MED.

How will traffic return from network 192.168.100.0 /24? Will it be routed through SanJose1 or SanJose2?

____

____

____

The simplest solution is to issue the **show ip bgp** command on the ISP router. What if access was not given to the ISP router? Traffic returning from the Internet should not be passed across the metered T1. Is there a simple way to verify before receiving the monthly bill? How can it be checked instantly?

____

____

____

a. Use an extended **ping** command in this situation. Specify the **record** option and compare your output to the following.

```
SanJose2# ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]: 2
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet has IP options:  Total option bytes= 39, padded length=40
 Record route: <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)

Reply to request 0 (48 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list

Reply to request 1 (48 ms).  Received packet has options
 Total option bytes= 40, padded length=40
```



```
Record route:
  (172.16.1.2)
  (192.168.1.6)
  (192.168.100.1)
  (192.168.1.1)
  (172.16.32.1) <*>
  (0.0.0.0)
  (0.0.0.0)
  (0.0.0.0)
  (0.0.0.0)
End of list
```

If you are unfamiliar with the **record** option, the important thing to note is that each IP address in brackets is an outgoing interface. The output can be interpreted as follows:

1. A ping that is sourced from 172.16.32.1 exits SanJose2 through s0/0/1, 172.16.1.2. It then arrives at the s0/0/1 interface for SanJose1.
2. SanJose1 S0/0/0, 192.168.1.6, routes the packet out to arrive at the S0/0/0 interface of ISP.
3. The target of 192.168.100.1 is reached: 192.168.100.1.
4. The packet is next forwarded out the S0/0/1, 192.168.1.1 interface for ISP and arrives at the S0/0/0 interface for SanJose2.
5. SanJose2 then forwards the packet out the last interface, loopback 0, 172.16.32.1.

Although the unlimited use of the T1 from SanJose1 is preferred here, ISP currently takes the link from SanJose2 for all return traffic.

b. Create a new policy to force the ISP router to return all traffic via SanJose1. Create a second route map utilizing the MED (metric) that is shared between EBGP neighbors.

```
SanJose1(config)#route-map PRIMARY_T1_MED_OUT permit 10
SanJose1(config-route-map)#set Metric 50
SanJose1(config-route-map)#exit
SanJose1(config)#router bgp 64512
SanJose1(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_MED_OUT out

SanJose2(config)#route-map SECONDARY_T1_MED_OUT permit 10
SanJose2(config-route-map)#set Metric 75
SanJose2(config-route-map)#exit
SanJose2(config)#router bgp 64512
SanJose2(config-router)#neighbor 192.168.1.1 route-map SECONDARY_T1_MED_OUT
out
```

c. Use the **clear ip bgp * soft** command after issuing this new policy. Issuing the **show ip bgp** command as follows on SanJose1 or SanJose2 does not indicate anything about this newly defined policy.

```
SanJose1# clear ip bgp * soft
SanJose2# clear ip bgp * soft

SanJose1# show ip bgp
BGP table version is 10, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal  Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i172.16.0.0       172.16.32.1              0    100      0 i
*>                  0.0.0.0                  0         32768 i
*> 192.168.1.0/30   192.168.1.5              0    150      0 200 i
```



```
*> 192.168.1.4/30   192.168.1.5              0    150      0 200 i
*> 192.168.100.0    192.168.1.5              0    150      0 200 i
```

d. Reissue an extended **ping** command with the **record** command.

```
SanJose2# ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]: 2
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet has IP options:  Total option bytes= 39, padded length=40
 Record route: <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
Reply to request 0 (64 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
 End of list
Reply to request 1 (64 ms).  Received packet has options
 Total option bytes= 40, padded length=40
 Record route:
   (172.16.1.2)
   (192.168.1.6)
   (192.168.100.1)
   (192.168.1.5)
   (172.16.1.1)
   (172.16.32.1) <*>
   (0.0.0.0)
   (0.0.0.0)
   (0.0.0.0)
```



```
End of list
```

Does the output look correct? Does the 192.168.1.5 above mean that the ISP now prefers SanJose1 for return traffic?

____

____

There might not be a chance to use Telnet to the ISP router and to issue the **show ip bgp** command. However, the command on the opposite side of the newly configured policy MED is clear, showing that the lower value is considered best. The ISP now prefers the route with the lower MED value to AS 64512. This is just opposite from the **local-preference** command configured earlier.

```
ISP# show ip bgp

BGP table version is 12, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i –
internal   Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*  172.16.0.0       192.168.1.2             75             0 64512 i
*>                  192.168.1.6             50             0 64512 i
*> 192.168.1.0/30   0.0.0.0                  0         32768 i
*> 192.168.1.4/30   0.0.0.0                  0         32768 i
*> 192.168.100.0    0.0.0.0                  0         32768 i
```

Step 11: Establish a default network.

The final step is to establish a default route that uses a policy statement that adjusts to changes in the network. Configure both SanJose1 and SanJose2 to use the 192.168.100.0 /24 network as the default network. The following steps configure the SanJose1 router. Do the same on the SanJose2 router.

a. View the routing table prior to issuing the **ip default-network** statement.

```
SanJose1# show ip route
Gateway of last resort is not set
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.32.0/24 [90/20640000] via 172.16.1.2, 02:43:46, Serial0/1
B       172.16.0.0/16 [200/0] via 172.16.32.1, 00:12:32
C       172.16.1.0/24 is directly connected, Serial0/1
C       172.16.64.0/24 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 2 subnets
B       192.168.1.0 [20/0] via 192.168.1.5, 00:14:05
C       192.168.1.4 is directly connected, Serial0/0
B    192.168.100.0/24 [20/0] via 192.168.1.5, 00:14:05
```

b. Configure the default network.

```
SanJose1(config)#ip default-network 192.168.100.0
```

Note: The above command works well only with remotely-learned classful networks. It should not be used with classless networks. An alternative to using the **ip default-network** command on SanJose1 is issuing the **neighbor X.X.X.X default-originate** configuration on the ISP router.

c. View the routing table after issuing the **ip default-network** statement.

```
SanJose1# show ip route
Gateway of last resort is 192.168.1.5 to network 192.168.100.0
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
D       172.16.32.0/24 [90/20640000] via 172.16.1.2, 02:44:09, Serial0/1
B       172.16.0.0/16 [200/0] via 172.16.32.1, 00:12:55
```



```
C       172.16.1.0/24 is directly connected, Serial0/1
C       172.16.64.0/24 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 2 subnets
B       192.168.1.0 [20/0] via 192.168.1.5, 00:14:28
C       192.168.1.4 is directly connected, Serial0/0
B*   192.168.100.0/24 [20/0] via 192.168.1.5, 00:14:29
```

What would be required to add a future T3 link on SanJose2 and for it to have preference for incoming and outgoing traffic?

____

____

____

A newly added route is as easy as adding another route map for local preference with a value of 175 and a route map referencing a MED (metric) value of 35.

NOTE: By default, the MED is compared only when the route is being received from the same neighboring AS, although advertised by different border routers. The nondefault behavior of comparing the MED regardless of the AS advertising the route can be activated using the **bgp always-compare-med** command, however, the results of this command have to be carefully considered .

NOTE: Because the MED is an optional attribute, it might not be present in BGP updates. RFC 4271 requires that a missing MED is equivalent to having the MED set to 0. However, a missing MED can also be considered to be the worst possible MED, which is activated using the **bgp bestpath med missing-as-worst** command.

d. Run the following Tcl script on all routers to verify full connectivity.

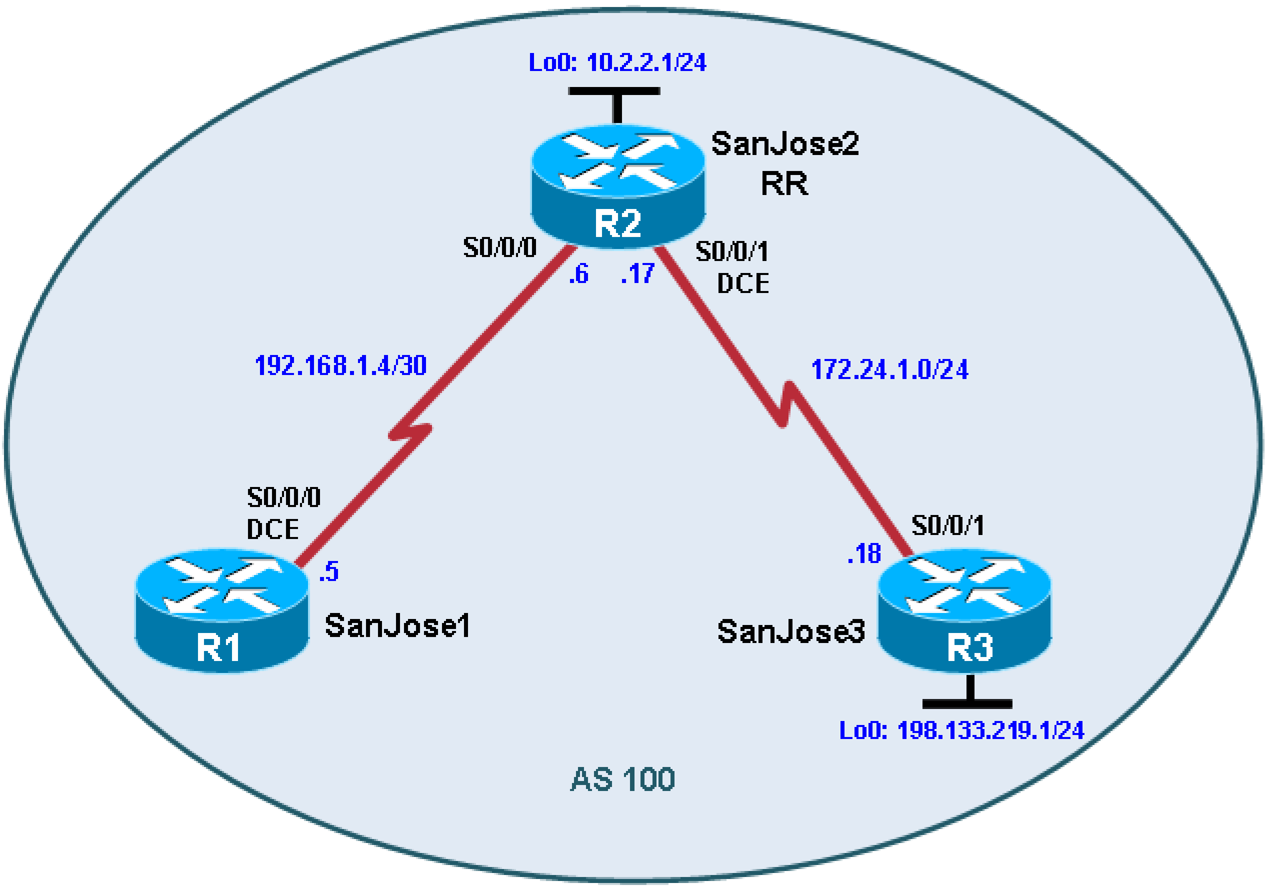
```
ISP# tclsh

foreach address {
192.168.100.1
172.16.64.1
172.16.32.1
192.168.1.1
192.168.1.2
192.168.1.5
192.168.1.6
172.16.1.1
172.16.1.2
} {
ping $address }
```


Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 6 Lab 6-4, BGP Route Reflectors and Route Filters

Topology



Objectives

- Configure IBGP routers to use a route reflector and a simple route filter.

Background

The International Travel Agency maintains a full-mesh IBGP network that has quickly scaled beyond 100 routers. The company wants to implement route reflectors to work around the full-mesh IBGP requirement. Configure a small cluster and observe how BGP operates in this configuration. Use IP prefix filters to control the updates between IBGP peers.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers for the lab.

Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations. Do not configure Loopback 0 on SanJose3 at this time.

Step 2: Configure the hostname and interface addresses.

You can copy and paste the following configurations into your routers to begin.

Router R1 (hostname SanJose1)

```
hostname SanJose1
!
interface Serial0/0/0
 ip address 192.168.1.5 255.255.255.252
 clock rate 128000
 no shutdown
```

Router R2 (hostname SanJose2)

```
hostname SanJose2
!
interface Loopback0
 ip address 10.2.2.1 255.255.255.0
!
interface Serial0/0/0
 ip address 192.168.1.6 255.255.255.252
 no shutdown
!
interface Serial0/0/1
 ip address 172.24.1.17 255.255.255.0
 clock rate 128000
 no shutdown
```

Router R3 (hostname SanJose3)

```
hostname SanJose3
!
interface Serial0/0/1
 ip address 172.24.1.18 255.255.255.0
 no shutdown
```

Note: Do not configure R3 (SanJose3) with loopback 0 at this time. That will be done in a later step.

Step 3: Configure RIPv2.

a. Build and configure the network according to the diagram. Use RIPv2 as the IGP. Do not configure the 198.133.219.0 network under the RIP process.

```
SanJose1(config)# router rip
SanJose1(config-router)# version 2
SanJose1(config-router)# no auto-summary
SanJose1(config-router)# network 192.168.1.0

SanJose2(config)# router rip
SanJose2(config-router)# version 2
```



```
SanJose2(config-router)# no auto-summary
SanJose2(config-router)# network 172.24.0.0
SanJose2(config-router)# network 192.168.1.0
SanJose2(config-router)# network 10.0.0.0

SanJose3(config)# router rip
SanJose3(config-router)# version 2
SanJose3(config-router)# no auto-summary
SanJose3(config-router)# network 172.24.0.0
```

b. Issue the **show ip route** command on the routers to verify that each router has a complete routing table.

```
SanJose1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.24.0.0/24 is subnetted, 1 subnets
R       172.24.1.0 [120/1] via 192.168.1.6, 00:00:21, Serial0/0/0
     10.0.0.0/24 is subnetted, 1 subnets
R       10.2.2.0 [120/1] via 192.168.1.6, 00:00:21, Serial0/0/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0/0
```

c. Run the following Tcl script on all routers to verify connectivity.

```
SanJose1# tclsh

foreach address {
10.2.2.1
192.168.1.5
192.168.1.6
172.24.1.17
172.24.1.18
} {
ping $address }
```

Step 4: Configure IBGP peers and route reflectors.

In this lab, you will configure a route reflector. By default, a router that receives an EBGP route advertises it to its EBGP and IBGP peers. However, if it receives it through IBGP, it does not advertise it to its IBGP peers, as a loop prevention mechanism. To maintain loop prevention, a route reflector adds two optional, nontransitive BGP attributes to each reflected route, the ORIGINATOR_ID and CLUSTER_LIST. It uses these attributes in a similar way to AS_PATH list to prevent routing loops from occurring. See http://tools.ietf.org/html/rfc4456 for more information.

However, because of this behavior, the only way for all IBGP routers to receive a route after it is originated into the AS is to have a full mesh of IBGP peers. This can get complex with a large number of peers. A route reflector allows a topology to get around the IBGP limitation of having to have a full mesh. To do this, a route reflector specifies some of its neighbors as route reflector clients. When a route reflector receives an update from a route reflector client, it can pass it on to its other clients. The route reflector would also pass that client-learned route on to its other non-client peers (both IBGP and EBGP peers). Similarly, a route learned from a non-client peer (again, from either an IBGP or EBGP peer) would be passed on to its client peers. This greatly simplifies configuration because only the route reflector needs to know all the other peers. The clients do not even know that they are clients. To them, it is just a normal IBGP peering relationship. You can even set up multiple route reflectors in a more advanced configuration for redundancy.

a. Configure the IBGP peers for BGP. Later, you will configure SanJose2 as the route reflector. However, first configure it to peer with both of the other routers.

```
SanJose2(config)# router bgp 100
SanJose2(config-router)# neighbor 192.168.1.5 remote-as 100
SanJose2(config-router)# neighbor 172.24.1.18 remote-as 100
```

After SanJose2 is configured, configure the other two routers as route reflector clients. Remember that to set up clients simply, configure peering between the client and the server. IBGP does not need to be configured in a full mesh.

b. Issue the following commands on SanJose1:

```
SanJose1(config)# router bgp 100
SanJose1(config-router)# neighbor 192.168.1.6 remote-as 100
```

c. Issue the following commands on SanJose3:

```
SanJose3(config)# router bgp 100
SanJose3(config-router)# neighbor 172.24.1.17 remote-as 100
```

d. Use the **show ip bgp neighbors** command to verify that SanJose2 has established a peering relationship with both SanJose1 and SanJose3. Troubleshoot as necessary.

```
SanJose2# show ip bgp neighbors
BGP neighbor is 172.24.1.18,  remote AS 100, internal link
  BGP version 4, remote router ID 172.24.1.18
  BGP state = Established, up for 00:02:10

<output omitted>

BGP neighbor is 192.168.1.5,  remote AS 100, internal link
  BGP version 4, remote router ID 192.168.1.5
  BGP state = Established, up for 00:04:15
```

SanJose1 and SanJose3 should not have established a connection. Why?

____

____

SanJose1 and SanJose3 were not configured with the appropriate BGP **neighbor** command. As route reflector clients, SanJose1 and SanJose3 do not need to reach an established state.

Step 5: Inject a network into BGP.

a. To observe the full effect of using a route reflector, configure SanJose3 to inject external routing information into BGP.

```
SanJose3(config)# interface loopback 0
SanJose3(config-if)# ip address 198.133.219.1 255.255.255.0

SanJose3(config-if)# router bgp 100
SanJose3(config-router)# network 198.133.219.0
```


This configuration forces SanJose3 to inject the external route 198.133.219.0 into BGP. Use the **show ip route** command to check if SanJose2 has picked up this route through BGP. SanJose2 should have a route to 198.133.219.0.

```
SanJose2# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.24.0.0/24 is subnetted, 1 subnets
C       172.24.1.0 is directly connected, Serial0/0/1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.2.2.0 is directly connected, Loopback0
B    198.133.219.0/24 [200/0] via 172.24.1.18, 00:01:48
C       10.2.2.0 is directly connected, Loopback0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0/0
```

What is the next hop for this route? Explain.

____

____

b. Verify that you can ping 198.133.219.1 from SanJose2. If not, troubleshoot.

c. Check the routing table of SanJose1. There should not be a route to 198.133.219.0. Why?

____

____

d. Remember that SanJose1 is not configured to peer with SanJose3. To eliminate the need for a full IBGP mesh, SanJose2 must be configured as a route reflector. Issue the following commands on SanJose2:

```
SanJose2(config)# router bgp 100
SanJose2(config-router)# neighbor 192.168.1.5 route-reflector-client
SanJose2(config-router)# neighbor 172.24.1.18 route-reflector-client

*Mar  9 19:02:27.831: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Down RR client
conf
ig change
*Mar  9 19:02:27.931: %BGP-5-ADJCHANGE: neighbor 172.24.1.18 Down RR client
conf
ig change
*Mar  9 19:02:32.387: %BGP-5-ADJCHANGE: neighbor 172.24.1.18 Up
*Mar  9 19:02:37.507: %BGP-5-ADJCHANGE: neighbor 192.168.1.5 Up
```

e. Verify that an IBGP cluster was successfully created by issuing the **show ip protocols** command on SanJose2. The output of this command should indicate that SanJose2 is a route reflector.

```
SanJose2# 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 26 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:
    10.0.0.0
    172.24.0.0
    192.168.1.0
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: (default is 120)

Routing Protocol is "bgp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Route Reflector for address family IPv4 Unicast, 2 clients
  Route Reflector for address family IPv6 Unicast, 2 clients
  Route Reflector for address family IPv4 MDT, 2 clients
  Route Reflector for address family VPNv4 Unicast, 2 clients
  Route Reflector for address family VPNv6 Unicast, 2 clients
  Route Reflector for address family IPv4 Multicast, 2 clients
  Route Reflector for address family IPv6 Multicast, 2 clients
  Route Reflector for address family NSAP Unicast, 2 clients
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Neighbor(s):
    Address          FiltIn FiltOut DistIn DistOut Weight RouteMap
    172.24.1.18
    192.168.1.5
  Maximum path: 1
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.24.1.18          200      00:01:43
  Distance: external 20 internal 200 local 200
```

How many clients does SanJose2 have?

____

f. Issue the **show ip protocols** command on SanJose1. The output of this command does not include information about route reflectors. Remember that SanJose1 is a client and not a route reflector server, so it is unaware of route reflection.

g. Finally, verify that route reflection is working by checking the routing table on SanJose1. SanJose1 will have a route to network 198.133.219.0.

```
SanJose1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
```



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

Gateway of last resort is not set

     172.24.0.0/24 is subnetted, 1 subnets
R       172.24.1.0 [120/1] via 192.168.1.6, 00:00:08, Serial0/0/0
     10.0.0.0/24 is subnetted, 1 subnets
R       10.2.2.0 [120/1] via 192.168.1.6, 00:00:08, Serial0/0/0
B    198.133.219.0/24 [200/0] via 172.24.1.18, 00:01:25
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0/0
```

Is 172.24.1.18 the IP address of the next hop of this route on the SanJose1 table? Explain.

____

____

Notice that SanJose1 is not directly connected to the IP network for the next hop. Why?
Hint: From which router did SanJose1 learn the route?

____

____

h. Ping 198.133.219.1 from SanJose1. This ping should be successful.

   Notice that SanJose1 pings to R3 198.133.219.1 are successful even though the next-hop address is not on a directly-connected network. For example, the next-hop address could be 192.168.1.6 on R2 if it were not for the behavior of IBGP.

Step 6: Inject a summary address into BGP.

a. For the purpose of this lab, configure SanJose3 to inject a summary address into BGP.

```
SanJose3(config)# router bgp 100
SanJose3(config-router)# aggregate-address 198.0.0.0 255.0.0.0
```

   BGP should now send the supernet route 198.0.0.0.0/8 to SanJose2 with the attribute ATOMIC_AGGREGATE set.

   Note: By default, BGP on Cisco routers advertises both aggregate routes and the individual component routes. If only the aggregate route is to be advertised, use the **aggregate-address** *network mask* **summary-only** command.

b. On SanJose2, issue the following command:

```
SanJose2# show ip bgp 198.0.0.0
BGP routing table entry for 198.0.0.0/8, version 8
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
  Advertised to update-groups:
        1
  Local, (aggregated by 100 172.24.1.18), (Received from a RR-client)
    172.24.1.18 from 172.24.1.18 (172.24.1.18)
    Origin IGP, metric 0, localpref 100, valid, internal, atomic-aggregate,
best
```

   According to the output of this command, which address aggregated this route?

____

What indicates that route reflection is involved in this process?

____

Is there an indication that the ATOMIC_AGGREGATE attribute has been set?

____

c. SanJose2 should, in turn, reflect this route to SanJose1. Check both the routing table and BGP table on SanJose1 to be sure. Both the route to 198.133.219.0 and the supernet route 198.0.0.0 should be installed in the SanJose1 routing table and the BGP table.

```
SanJose1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.24.0.0/24 is subnetted, 1 subnets
R       172.24.1.0 [120/1] via 192.168.1.6, 00:00:20, Serial0/0/0
     10.0.0.0/24 is subnetted, 1 subnets
R       10.2.2.0 [120/1] via 192.168.1.6, 00:00:20, Serial0/0/0
B    198.133.219.0/24 [200/0] via 172.24.1.18, 00:08:34
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0/0
B    198.0.0.0/8 [200/0] via 172.24.1.18, 00:04:19
```

The International Travel Agency has decided to filter specific routes to the 198.0.0.0/8 address space. Configure a route filter to prevent SanJose2 from sending the 198.133.219.0/24 route to its other clients, in this case to SanJose1.

d. Issue the following commands on SanJose2:

```
SanJose2(config)# ip prefix-list SUPERNETONLY permit 198.0.0.0/8
SanJose2(config)# router bgp 100
SanJose2(config-router)# neighbor 192.168.1.5 prefix-list SUPERNETONLY out
```

e. Return to SanJose1, issue the **clear ip bgp * soft** command, and verify that the prefix list has done its job by issuing a **show ip bgp** command. Troubleshoot as necessary.

Unlike before, where routes to 198.133.219.0 and 198.0.0.0 were present, now only one route to 198.0.0.0 in the routing and BGP tables should be seen. Troubleshoot as necessary.

```
SanJose1# show ip route
Codes: C - connected, S - static, 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
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route
```



```
Gateway of last resort is not set

     172.24.0.0/24 is subnetted, 1 subnets
R       172.24.1.0 [120/1] via 192.168.1.6, 00:00:20, Serial0/0/0
     10.0.0.0/24 is subnetted, 1 subnets
R       10.2.2.0 [120/1] via 192.168.1.6, 00:00:20, Serial0/0/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.4 is directly connected, Serial0/0/0
B    198.0.0.0/8 [200/0] via 172.24.1.18, 00:04:19
```

f. Run the following Tcl script on all routers to verify full connectivity. All pings should be successful.

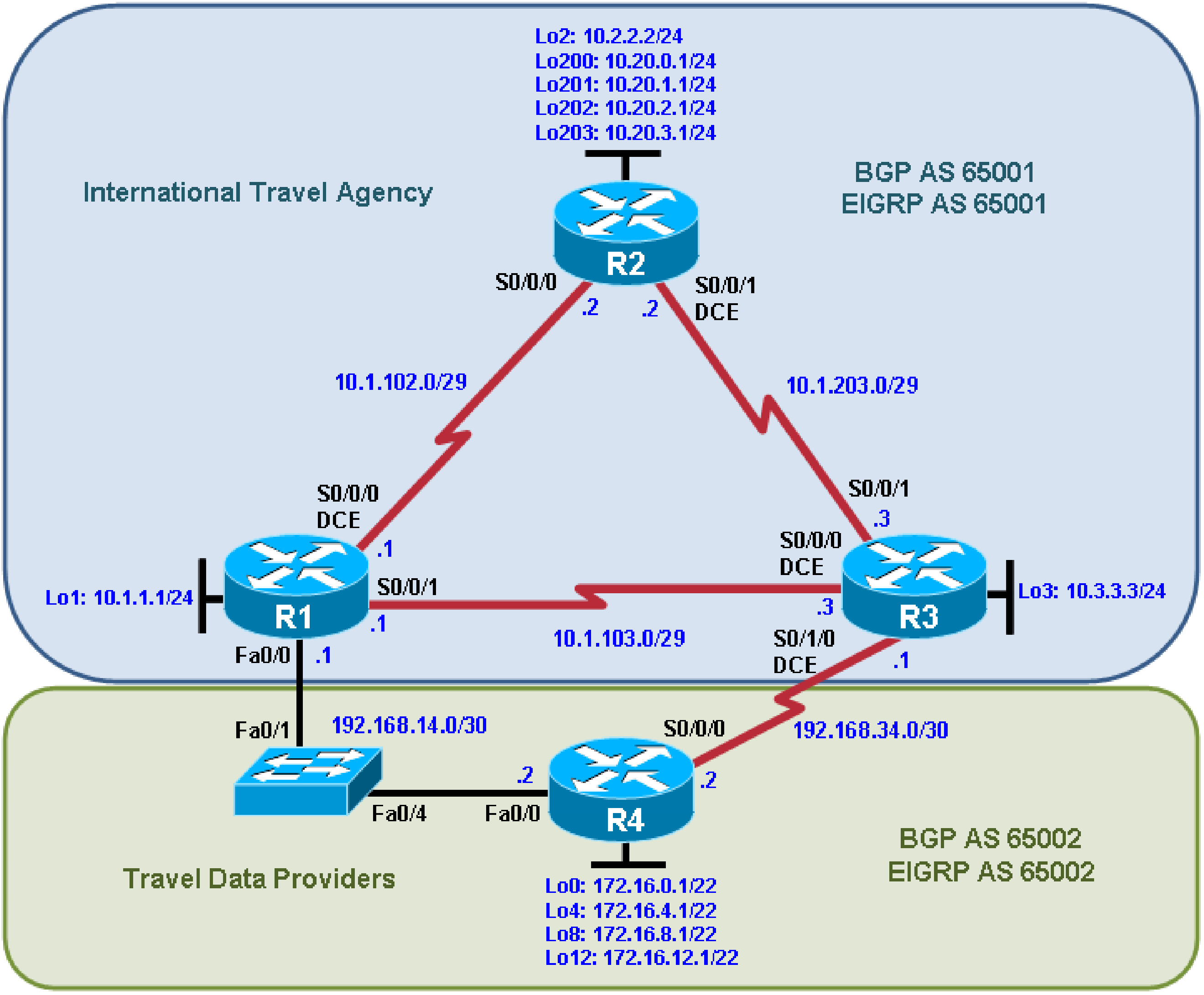
```
SanJose1# tclsh

foreach address {
10.2.2.1
198.133.219.1
192.168.1.5
192.168.1.6
172.24.1.17
172.24.1.18
} {
ping $address }
```

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than try to list all the combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. An example of this is an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 6 Lab 6-5, BGP Case Study

Topology



Objectives

- Plan, design, and implement the International Travel Agency core network.
- Plan, design, and implement the Travel Data Providers network.
- Allow the networks to communicate via BGP.
- Verify that all implementations are operational and functional according to the guidelines.

Background

The International Travel Agency (ITA) needs both its core network and its Travel Data Providers (TDP) network set up. Configure each network to run EIGRP, and use BGP to advertise routes between the two.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. The switch is a Cisco WS-C2960-24TT-L with the Cisco IOS image c2960-lanbasek9-mz.122-46.SE.bin, You can use other routers (such as a 2801 or 2811), switches (such as a 2950), and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- 1 switch (Cisco 2960 with the Cisco IOS Release 12.2(46)SE C2960-LANBASEK9-M image or comparable)
- Serial and Ethernet cables

Requirements

1. Use the addressing scheme shown in the diagram.
2. Configure the ITA network to be in EIGRP AS 65001.
3. Configure the TDP network to be in EIGRP AS 65002.
4. Disable automatic summarization in both EIGRP domains.
5. Configure the ITA network to be in BGP AS 65001, and the TDP network to be in BGP AS 65002.
6. Advertise the 192.168.14.0/30 and 192.168.34.0/30 networks in both EIGRP autonomous systems.
7. Configure the interfaces on the border routers between the two EIGRP autonomous systems, so they do not send EIGRP packets.
8. All routers will be participating in BGP. Configure all routers for a full mesh of IBGP peers in each system.
9. Peer R1 and R2 using loopback addresses, not their directly connected interfaces.
10. Advertise all loopback interfaces into the BGP process, except on R2, where the only loopback advertised should be loopback 2.
11. On R2, create a static summary route for the rest of its loopback interfaces and advertise this static route in BGP.
12. R4 should send a summary route to ITA representing all the R4 loopback interfaces.
13. R4 should prefer the path to ITA networks via the Ethernet link between R1 and R4. Accomplish this by modifying the MED advertised to TDP.
14. Routers in the ITA AS should prefer the path to TDP networks via the Ethernet link between R1 and R4. Accomplish this by modifying the local preference of routes being advertised in from TDP.

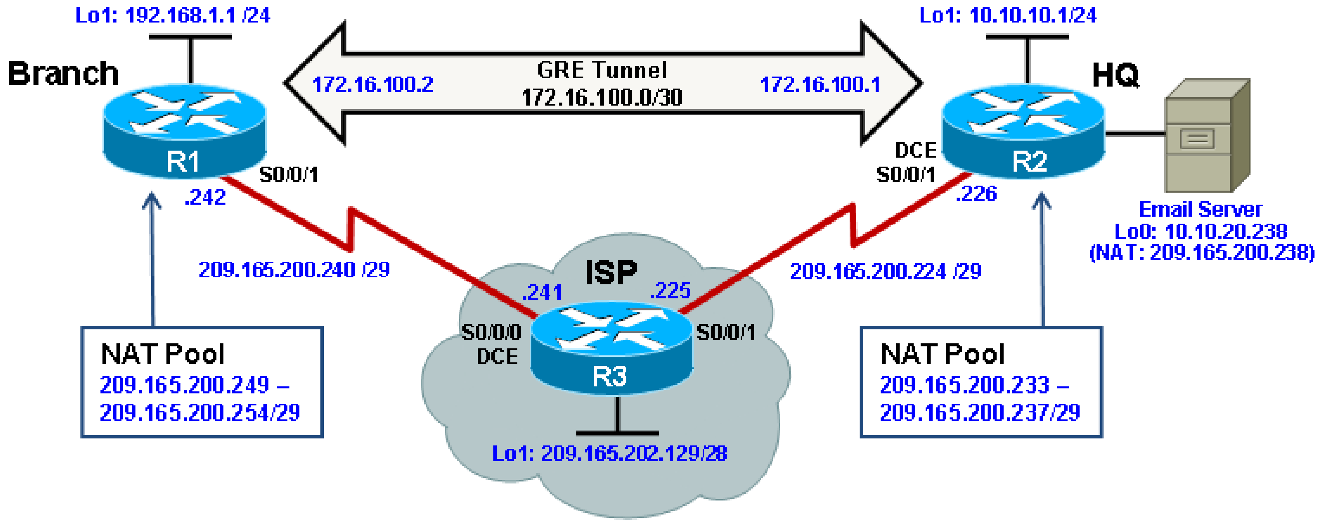
Notes:

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 7 Lab 7-1, Configure Routing Facilities to the Branch Office

Topology



Objectives

- Configure NAT.
- Configure an IPsec VPN.
- Configure a GRE tunnel over IPsec.
- Enable dynamic routing over a GRE tunnel.
- Verify the configuration and operation using **show** and **debug** commands.

Background

Your organization is expanding its operation and wants to connect a branch site. To avoid expensive WAN costs, the decision was made to use the Internet as the WAN link. You suggest using a test network to implement an IPsec VPN to support all traffic going between corporate sites. In addition, you want to configure dynamic routing between sites, by implementing Generic Routing Encapsulation (GRE).

Note: The intent of this lab is to illustrate the impact on routing services and addressing schemes when deploying IPsec VPNs at branch office routers. Although sample configurations are provided, detailed explanations of Network Address Translation (NAT), IPsec VPNs, and GRE are beyond the scope of this course. For more details on these technologies, see the Cisco Networking Academy CCNA Security course or www.cisco.com.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and

Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers and configure the router hostname and interface addresses.

a. Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations. Using the addressing scheme in the diagram, apply the IP addresses to the interfaces on Branch, HQ, and ISP.

You can copy and paste the following configurations into your routers to begin.

Note: Depending on the router model, interfaces might be numbered differently than those listed. You might need to alter the designations accordingly.

Branch (R1)

```
hostname Branch
!
interface Loopback1
 description Branch LAN
 ip address 192.168.1.1 255.255.255.0
!
interface Serial0/0/1
 description Connection to ISP
 ip address 209.165.200.242 255.255.255.248
 bandwidth 64
 no shut
!
end
```

HQ (R2)

```
hostname HQ
!
interface Loopback1
 description Headquarters LAN
 ip address 10.10.10.1 255.255.255.0
!
interface Serial0/0/1
 description Connection to ISP
 ip address 209.165.200.226 255.255.255.248
 clock rate 64000
 bandwidth 64
 no shut
!
end
```

ISP (R3)

```
hostname ISP
!
interface Loopback1
 description Simulating the Internet
 ip address 209.165.202.129 255.255.255.240
```



```
!
interface Serial0/0/0
 description Connection to Branch
 ip address 209.165.200.241 255.255.255.248
 clock rate 64000
 bandwidth 64
 no shut
!
interface Serial0/0/1
 description Connection to HQ
 ip address 209.165.200.225 255.255.255.248
 bandwidth 64
 no shut

ip route 209.165.200.232 255.255.255.248 Serial0/0/1
ip route 209.165.200.248 255.255.255.248 Serial0/0/0
!
end
```

b. Verify your configuration by using the **show ip interface brief** command. The output from the Branch router is shown here as an example.

```
Branch# show ip interface brief
Interface          IP-Address      OK? Method Status                Protocol
FastEthernet0/0    unassigned      YES unset  administratively down down
FastEthernet0/1    unassigned      YES unset  administratively down down
Serial0/0/0        unassigned      YES unset  administratively down down
Serial0/0/1        209.165.200.242 YES manual up                    up
Loopback1          192.168.1.1     YES manual up                    up
```

c. From the Branch LAN interface, use an extended ping to verify connectivity to the directly connected interface of the ISP, the ISPs loopback interface, and the HQ Internet interface. Run the following Tcl script on the Branch router to verify connectivity.

```
Branch# tclsh

foreach address {
209.165.200.241
209.165.202.129
209.165.200.226
} { ping $address}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.241, 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 209.165.202.129, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.226, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Branch(tcl)#
```


Why do the pings to the ISPs loopback and HQ router address fail?

____

____

d. Configure a default static route to ISP on the Branch and HQ routers.

You can copy and paste the following configurations into your routers.

```
Branch(config)# ip route 0.0.0.0 0.0.0.0 209.165.200.241

HQ(config)# ip route 0.0.0.0 0.0.0.0 209.165.200.225
```

e. From the Branch router, run the following Tcl script on the Branch router to verify connectivity.

```
Branch# tclsh

foreach address {
209.165.200.241
209.165.202.129
209.165.200.226
+>} { ping $address}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.241, 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 209.165.202.129, 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 209.165.200.226, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
Branch(tcl)#
```

Are the pings now successful?

____

____

f. Connectivity from the Branch router to external addresses has been established. But could a Branch LAN user successfully reach those external addresses? To verify, initiate pings sourced from the Branch LAN interface to the ISP interface, the ISPs loopback interface, and the HQ Internet interface. Run the following Tcl script on the Branch router to verify connectivity.

```
Branch# tclsh

foreach address {
209.165.200.241
209.165.202.129
209.165.200.226
} { ping $address source 192.168.1.1}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.241, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
.....
```



```
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.202.129, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
.....
Success rate is 0 percent (0/5)
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.226, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
.....
Success rate is 0 percent (0/5)
Branch(tcl)#
```

Note: You can also specify the router interface designator (for example, S0/0/0, Fa0/0, or Lo1) as the source for the extended ping, as follows:

Branch# **ping 209.165.200.226 source Lo1**

Why are the pings unsuccessful?

____

____

The ISP cannot route back to the internal private address of the Branch LAN.

Step 2: Configure NAT on the Branch and HQ routers.

The internal LAN private IP addresses will be translated to global public IP addresses using NAT. The ISP has provided the HQ and Branch sites with the following pools of public addresses:

- HQ: 209.165.200.233 – 209.165.200.238 (209.165.200.232/29)
- Branch: 209.165.200.249 – 209.165.200.254 (209.165.200.248/29)

The HQ site also has an email server that must be accessible to mobile users and Branch office users. Therefore, static NAT must also be configured to use a public address to reach the email server.

a. The NAT pool identifies public IP addresses, while the NAT ACL identifies which inside hosts can use these public IP addresses. For the Branch router, this means that the NAT ACL must identify the 192.168.1.0 LAN, and the NAT pool must identify addresses 209.165.200.248 /29. The LAN interface must be identified as an inside NAT interface, and the Internet interface must be identified as an outside NAT interface.

Note: The NAT ACL must not translate the Branch LAN addresses if they are destined to the corporate HQ LAN. Therefore, the NAT ACL denies the Branch LAN public addresses from being translated when attempting to connect to the HQ LANs. This will be required when the IPsec VPN is configured.

You can copy and paste the following configuration into the Branch router.

Branch Router

```
ip access-list extended BRANCH-NAT-ACL
 remark Do not translate Local LAN to HQ LAN addresses
 deny ip 192.168.1.0 0.0.0.255 10.10.0.0 0.0.255.255
 remark Translate Local LAN to all other Internet destinations
 permit ip 192.168.1.0 0.0.0.255 any
 exit
!
ip nat pool BRANCH-NAT-POOL 209.165.200.249 209.165.200.254 prefix-length 29
!
```



```
ip nat inside source list BRANCH-NAT-ACL pool BRANCH-NAT-POOL
!
interface Loopback 1
 ip nat inside
exit
!
interface Serial0/0/1
 ip nat outside
end
```

b. On the HQ router, the NAT ACL must identify the 10.10.10.0 and the 10.10.20.0 LANs. The NAT pool must identify addresses 209.165.200.232 /29. The LAN interface must be identified as an inside NAT interface, and the Internet interface must be identified as an outside NAT interface.

   The email server with private IP address 10.10.20.238 will be statically assigned the last public IP address from the NAT pool, 209.165.200.238. Interface loopback 0 on HQ simulates this server.

   Note: Again the NAT ACL denies the HQ LAN public addresses from being translated when attempting to connect to the Branch LAN which will be required when the IPsec VPN is configured.

   You can copy and paste the following configuration into the HQ router.

   HQ Router

```
interface Loopback 0
 description HQ email server address
 ip add 10.10.20.238 255.255.255.0
!
ip nat pool HQ-NAT-POOL 209.165.200.233 209.165.200.237 prefix-length 29
ip nat inside source list HQ-NAT-ACL pool HQ-NAT-POOL
ip nat inside source static 10.10.20.238 209.165.200.238
!
ip access-list extended HQ-NAT-ACL
 remark Do not translate HQ LAN to Branch LAN addresses
 deny ip 10.10.0.0 0.0.255.255 192.168.1.0 0.0.0.255
 remark Translate Local LAN to all other Internet destinations
 permit ip 10.10.0.0 0.0.255.255 any
 exit
!
interface Loopback 0
 ip nat inside
!
interface Loopback 1
 ip nat inside
!
interface Serial0/0/1
 ip nat outside
end
```

c. Verify the NAT configuration by using the **show ip nat statistics** and **show ip nat translations** commands.

```
Branch# show ip nat statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Peak translations: 0, occurred 00:018:28 ago
Outside interfaces:
  Serial0/0/1
Inside interfaces:
  Loopback1
Hits: 0  Misses: 0
```



```
CEF Translated packets: 0, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list BRANCH-NAT-ACL pool BRANCH-NAT-POOL refcount 0
 pool BRANCH-NAT-POOL: netmask 255.255.255.248
        start 209.165.200.249 end 209.165.200.254
        type generic, total addresses 6, allocated 0 (0%), misses 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

As shown above, the pool has been configured and the interfaces assigned. The output of the **show ip nat translations** command confirms that there are currently no active NAT translations:

```
Branch# show ip nat translations

Branch#
```

d. Initiate NAT traffic by pinging from the Branch LAN to the ISP interface, ISP's loopback, the HQ Internet interface, and this time also include the HQ public email server address. Run the following Tcl script on the Branch router to verify connectivity.

```
Branch# tclsh

foreach address {
209.165.200.241
209.165.202.129
209.165.200.226
209.165.200.238
} { ping $address source 192.168.1.1}

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.241, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
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 209.165.202.129, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
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 209.165.200.226, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.238, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
Branch(tcl)#
```

All pings should be successful.

e. Verify that NAT is occurring by using the **show ip nat statistics** and **show ip nat translations** commands.


```
Branch# show ip nat statistics
Total active translations: 5 (0 static, 5 dynamic; 4 extended)
Peak translations: 5, occurred 00:00:12 ago
Outside interfaces:
  Serial0/0/1
Inside interfaces:
  Loopback1
Hits: 40  Misses: 0
CEF Translated packets: 20, CEF Punted packets: 0
Expired translations: 0
Dynamic mappings:
-- Inside Source
[Id: 1] access-list BRANCH-NAT-ACL pool BRANCH-NAT-POOL refcount 5
 pool BRANCH-NAT-POOL: netmask 255.255.255.248
        start 209.165.200.249 end 209.165.200.254
        type generic, total addresses 6, allocated 1 (16%), misses 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
Branch#
Branch# show ip nat translations
Pro Inside global      Inside local     Outside local      Outside global
icmp 209.165.200.249:9 192.168.1.1:9    209.165.200.241:9  209.165.200.241:9
icmp 209.165.200.249:10 192.168.1.1:10  209.165.202.129:10 209.165.202.129:10
icmp 209.165.200.249:11 192.168.1.1:11  209.165.200.226:11 209.165.200.226:11
icmp 209.165.200.249:12 192.168.1.1:12  209.165.200.238:12 209.165.200.238:12
--- 209.165.200.249    192.168.1.1      ---                ---
Branch#
```

Notice that translations are occurring. The output lists the details of the NAT translations sourced by the 192.168.1.1 Branch LAN IP address, which was translated to public IP address 209.165.200.249.

f. Now clear the NAT translations, verify connectivity from the Branch LAN to the HQ LAN interface and then display the NAT translations.

```
Branch# clear ip nat translation *
Branch#
Branch# ping 10.10.10.1 source 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
.....
Success rate is 0 percent (0/5)
Branch#
Branch# show ip nat translations

Branch#
```

As expected, Branch LAN traffic going to the HQ LAN is not translated by NAT. The ISP cannot route the pings to the private address on HQ and, therefore, the pings fail.

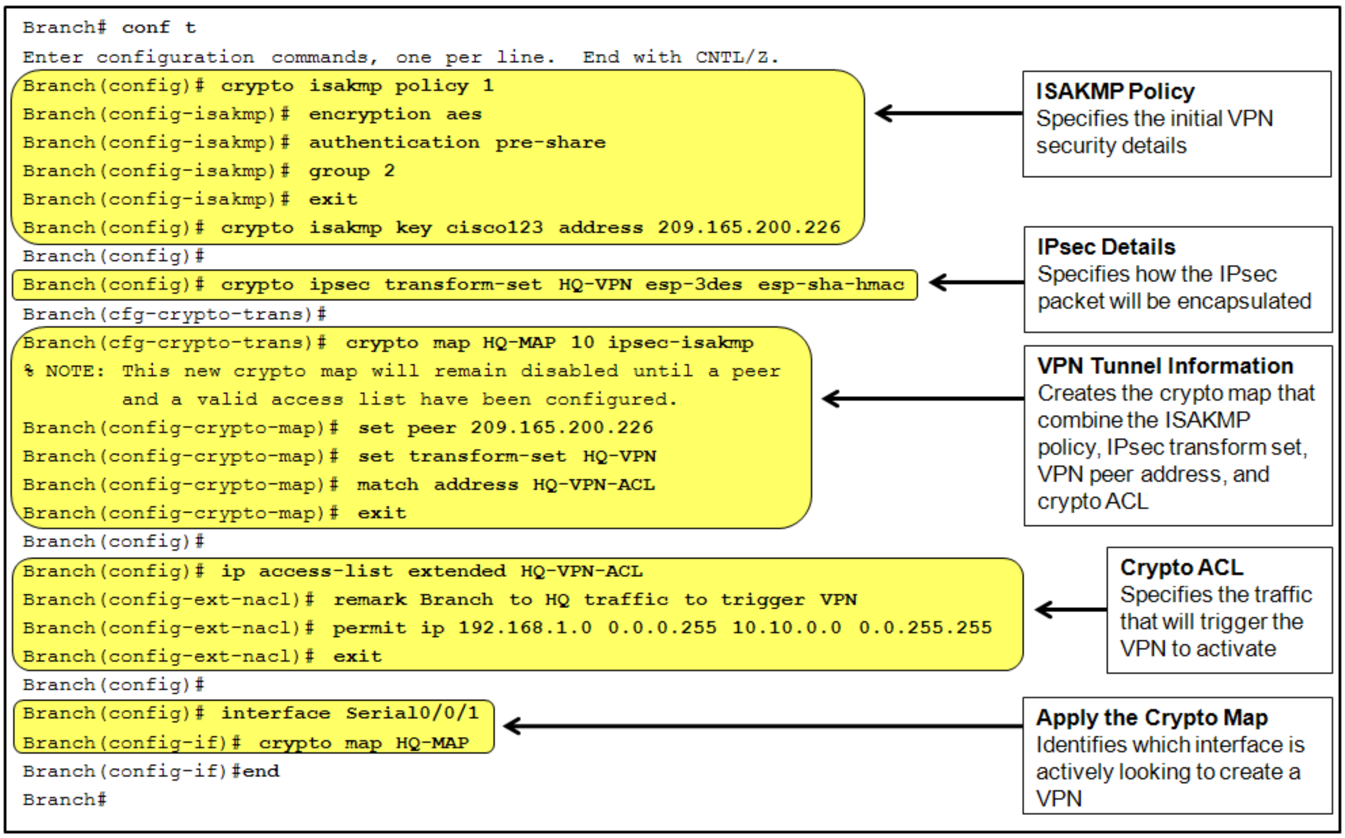
NAT works as expected. Traffic source from the Branch LAN going to Internet destinations is translated while traffic sourced from the Branch LAN to the HQ LAN is not translated. However, this traffic should be protected when traversing the public Internet. To solve this problem, an IPsec VPN will be configured next.

Step 3: Implement an IPsec VPN between the Branch and HQ sites.

An IPsec VPN can secure and protect all unicast IP traffic within it. IPsec cannot forward multicast or broadcast traffic, which means it cannot support interior gateway protocols such as EIGRP and OSPF.

For this lab, assume that the network security team has provided a basic IPsec VPN configuration with which to test your network design. As shown in the following figure, it consists of several configuration components:

- The ISAKMP policy identifies the specifics for the initial key and secure parameters exchange.
- The IPsec details define how the IP packet is encapsulated.
- The VPN tunnel information is identified in a named crypto map which combines the ISAKMP policies, IPsec packet detail, the peer address, and the crypto ACL.
- The crypto ACL identifies traffic that will trigger the tunnel to activate. This component must sometimes be tuned when implemented along with other services such as NAT and GRE.
- The crypto map is then applied to the tunnel interface.



Note: How to configure an IPsec VPN is beyond the scope of this lab. For more information on cryptography, IPsec VPNs, and GRE, see the Cisco Networking Academy CCNA Security courses or www.cisco.com.

a. Copy and paste the following configurations on the routers.

Branch Router

```
crypto isakmp policy 1
 encryption aes
 authentication pre-share
 group 2
crypto isakmp key cisco123 address 209.165.200.226
!
crypto ipsec transform-set HQ-VPN esp-3des esp-sha-hmac
```

```
!
crypto map HQ-MAP 10 ipsec-isakmp
 set peer 209.165.200.226
 set transform-set HQ-VPN
 match address HQ-VPN-ACL
!
ip access-list extended HQ-VPN-ACL
 remark Branch to HQ traffic to trigger VPN
 permit ip 192.168.1.0 0.0.0.255 10.10.0.0 0.0.255.255
!
interface Serial0/0/1
 crypto map HQ-MAP
end
```

HQ Router

```
crypto isakmp policy 1
 encryption aes
 authentication pre-share
 group 2
crypto isakmp key cisco123 address 209.165.200.242
!
crypto ipsec transform-set Branch-VPN esp-3des esp-sha-hmac
!
crypto map Branch-MAP 10 ipsec-isakmp
 set peer 209.165.200.242
 set transform-set Branch-VPN
 match address Branch-VPN-ACL
!
ip access-list extended Branch-VPN-ACL
 remark HQ to Branch traffic to trigger VPN
 permit ip 10.10.0.0 0.0.255.255 192.168.1.0 0.0.0.255
!
interface Serial0/0/1
 crypto map Branch-MAP
 end
```

Notice that the crypto ACLs are referring to the public IP addresses and not the private IP addresses. This is because the crypto map applies to the traffic after the NAT has already taken place. Another alternative approach would be to exempt site-to-site traffic from the NAT translation pool and have the crypto ACLs trigger based on private addresses instead of the public address pool.

b. Use the **show crypto session detail** command on the Branch router to verify the overall configuration of the IPsec VPN.

```
Branch# show crypto session detail

Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Serial0/0/1
Session status: DOWN
Peer: 209.165.200.226 port 500 fvrf: (none) ivrf: (none)
      Desc: (none)
```

```
        Phase1_id: (none)
    IPSEC FLOW: permit ip 192.168.1.0/255.255.255.0 10.10.0.0/255.255.0.0
          Active SAs: 0, origin: crypto map
          Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
          Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0

Branch#
```

The VPN tunnel is currently down because the traffic identified in the IPSEC FLOW has not yet been processed.

c. To test the VPN link, use an extended ping from the Branch LAN interface to the HQ LAN interface.

```
Branch# ping 10.10.10.1 source 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 84/86/88 ms
Branch#
```

This time 80% of the pings were successful. This is typical because the VPN tunnel requires a few seconds to negotiate the security parameters specified in the crypto map.

d. Now display the VPN tunnel details again.

```
Branch# show crypto session detail
Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Serial0/0/1
Uptime: 00:00:10
Session status: UP-ACTIVE
Peer: 209.165.200.226 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: 209.165.200.226
      Desc: (none)
  IKE SA: local 209.165.200.242/500 remote 209.165.200.226/500 Active
          Capabilities:(none) connid:1001 lifetime:23:59:49
  IPSEC FLOW: permit ip 192.168.1.0/255.255.255.0 10.10.0.0/255.255.0.0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 4 drop 0 life (KB/Sec) 4430126/3589
        Outbound: #pkts enc'ed 4 drop 1 life (KB/Sec) 4430126/3589

Branch#
```

The VPN tunnel did become active as indicated by the UP-ACTIVE session status. Also notice that it was the **permit** statement is referring to the private addresses defined in the crypto ACL and that it encrypted and decrypted four packets, with only one packet dropped due to the IPsec negotiation.

e. Before proceeding, manually disable the IPsec VPN tunnel using the **clear crypto isakmp** and **clear crypto sa** commands on the Branch router.


```
Branch# clear crypto isakmp
Branch# clear crypto sa
Branch#
```

You now have encrypted connectivity from the Branch LAN to HQ LAN. the problem with an IPsec VPN is that it does not allow dynamic routing protocols to operate over it. However, GRE can help solve this problem.

Step 4: Implement GRE over IPsec.

A GRE tunnel over IPsec can be implemented between the Branch and HQ sites. The tunnel will protect all corporate LAN traffic. As a bonus, GRE can forward multicast and broadcast traffic, so dynamic routing can also be enabled.

a. Configure the tunnel interfaces on the Branch router and HQ routers with GRE encapsulation. Copy and paste the following configurations on the routers.

Branch Router

```
interface Tunnel0
 ip address 172.16.100.2 255.255.255.252
 tunnel source 209.165.200.242
 tunnel destination 209.165.200.226
```

HQ Router

```
interface Tunnel0
 ip address 172.16.100.1 255.255.255.252
 tunnel source 209.165.200.226
 tunnel destination 209.165.200.242
```

You should notice the state of the tunnel interfaces to change to up on both routers.

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state to up
```

b. Verify that the tunnel interface is up and running using the **show interface tunnel 0** command.

```
Branch# show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.100.2/30
  MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 209.165.200.242, destination 209.165.200.226
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255
  Fast tunneling enabled
  Tunnel transport MTU 1476 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
```



```
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
Branch#
```

The tunnel interface is up. Also notice that the encapsulation and tunnel transport protocol is GRE/IP.

c. Verify connectivity across the tunnel by pinging the tunnel destination on the HQ router. The pings should be successful.

```
Branch# ping 172.16.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 68/70/72 ms
```

d. The pings successfully reach the other side of the tunnel. But is the traffic being encrypted? Display the IPsec VPN specifics.

```
Branch# show crypto session detail
Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Serial0/0/1
Session status: DOWN-NEGOTIATING
Peer: 209.165.200.226 port 500 fvrf: (none) ivrf: (none)
      Desc: (none)
      Phase1_id: (none)
  IKE SA: local 209.165.200.242/500 remote 209.165.200.226/500 Inactive
          Capabilities:(none) connid:1001 lifetime:0
  IPSEC FLOW: permit ip 192.168.1.0/255.255.255.0 10.10.0.0/255.255.0.0
        Active SAs: 0, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0

Branch#
```

The IPsec VPN is down because the tunnel traffic has not been identified in the crypto ACL.

e. To solve this problem, replace the crypto ACL to make GRE traffic interesting on the Branch and HQ routers. Copy and paste the following configurations on the routers.

Branch Router

```
no ip access-list extended HQ-VPN-ACL
ip access-list extended HQ-VPN-ACL
 remark HQ to Branch GRE traffic to trigger VPN
 permit gre host 209.165.200.242 host 209.165.200.226
```

HQ Router

```
no ip access-list extended Branch-VPN-ACL
ip access-list extended Branch-VPN-ACL
 remark Branch to HQ GRE traffic to trigger VPN
 permit gre host 209.165.200.226 host 209.165.200.242
```

f. Test the link again. Notice the pings are 80% successful again.

```
Branch# ping 172.16.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.100.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 96/98/100 ms
```

g. Display the IPsec session details.

```
Branch# show crypto session detail
Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Serial0/0/1
Uptime: 00:00:05
Session status: UP-ACTIVE
Peer: 209.165.200.226 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: 209.165.200.226
      Desc: (none)
  IKE SA: local 209.165.200.242/500 remote 209.165.200.226/500 Active
          Capabilities:(none) connid:1003 lifetime:23:59:54
  IPSEC FLOW: permit 47 host 209.165.200.242 host 209.165.200.226
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 4 drop 0 life (KB/Sec) 4422647/3594
        Outbound: #pkts enc'ed 4 drop 1 life (KB/Sec) 4422647/3594
```

The IPsec tunnel is now up and active. The “permit 47” identifies GRE traffic as interesting. The value 47 refers to the GRE protocol number.

h. Ping from LAN to LAN.

```
Branch# ping 10.10.10.1 source 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
.....
Success rate is 0 percent (0/5)
Branch#
```

The pings are unsuccessful. Does the Branch router have an entry to the 10.10.10.0 network?

i. Verify the routing table.

```
Branch# show ip route

<output omitted>
```



```
Gateway of last resort is 209.165.200.241 to network 0.0.0.0

     172.16.0.0/30 is subnetted, 1 subnets
C       172.16.100.0 is directly connected, Tunnel0
     209.165.200.0/29 is subnetted, 1 subnets
C       209.165.200.240 is directly connected, Serial0/0/1
C    192.168.1.0/24 is directly connected, Loopback1
S*   0.0.0.0/0 [1/0] via 209.165.200.241
Branch#
```

The pings are unsuccessful because there is no specific route to the other LAN. The traffic is finally routed using the default route, which bypasses the GRE tunnel. The Branch router and the HQ router must be configured to share each other's LAN information.

This could be accomplished using static routes pointing to the GRE tunnel destination IP address. Although this is valid option, GRE tunnels also support multicast and broadcast traffic. Therefore, a dynamic routing protocol should be configured.

j. Configure EIGRP, and advertise the LANs and the tunnel segment on the Branch and HQ routers. Copy and paste the following configurations on the routers.

Branch Router

```
router eigrp 1
network 192.168.1.0 0.0.0.255
network 172.16.100.0 0.0.0.3
```

HQ Router

```
router eigrp 1
network 10.10.0.0 0.0.255.255
network 172.16.100.0 0.0.0.3
```

An EIGRP neighbor adjacency message should appear almost immediately.

```
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 172.16.100.2 (Tunnel0) is up: new
adjacency
```

k. Verify the routing table.

```
Branch# show ip route

<output omitted>

Gateway of last resort is 209.165.200.241 to network 0.0.0.0

     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
D       172.16.0.0/16 is a summary, 00:00:22, Null0
C       172.16.100.0/30 is directly connected, Tunnel0
     209.165.200.0/29 is subnetted, 1 subnets
C       209.165.200.240 is directly connected, Serial0/0/1
D    10.0.0.0/8 [90/27008000] via 172.16.100.1, 00:00:10, Tunnel0
C    192.168.1.0/24 is directly connected, Loopback1
S*   0.0.0.0/0 [1/0] via 209.165.200.241
Branch#
```

l. Display the IPsec session detail.

```
Branch# show crypto session detail
Crypto session current status
```



```
Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Serial0/0/1
Uptime: 00:02:36
Session status: UP-ACTIVE
Peer: 209.165.200.226 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: 209.165.200.226
      Desc: (none)
  IKE SA: local 209.165.200.242/500 remote 209.165.200.226/500 Active
          Capabilities:(none) connid:1002 lifetime:23:57:23
  IPSEC FLOW: permit 47 host 209.165.200.242 host 209.165.200.226
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 18 drop 0 life (KB/Sec) 4436519/3443
        Outbound: #pkts enc'ed 21 drop 1 life (KB/Sec) 4436519/3443

Branch#
```

m. Test the LAN-to-LAN connectivity, and display the IPsec session detail.

```
Branch# ping 10.10.10.1 source 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/100/100 ms

Branch# show crypto session detail
Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Serial0/0/1
Uptime: 00:03:15
Session status: UP-ACTIVE
Peer: 209.165.200.226 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: 209.165.200.226
      Desc: (none)
  IKE SA: local 209.165.200.242/500 remote 209.165.200.226/500 Active
          Capabilities:(none) connid:1002 lifetime:23:56:44
  IPSEC FLOW: permit 47 host 209.165.200.242 host 209.165.200.226
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 31 drop 0 life (KB/Sec) 4436517/3404
        Outbound: #pkts enc'ed 34 drop 1 life (KB/Sec) 4436517/3404

Branch#
```

The pings are successful, but notice that the packet counters have increased by more than five ping packets. The reason is because EIGRP is also exchanging hello packets and therefore increasing the counters.

n. Trace the path that the packets take from the Branch LAN to the email server using the inside private address.


```
Branch# trace 10.10.20.238 source 192.168.1.1

Type escape sequence to abort.
Tracing the route to 10.10.20.238

  1 172.16.100.1 68 msec 68 msec *
```

Notice that the packet hops only to the end of the tunnel. It is completely unaware that it actually traversed the public Internet.

o. To prove that you still have Internet access without going through the GRE tunnel, trace the path from the Branch LAN to the email server using the outside static NAT address.

```
Branch# trace 209.165.200.238 source 192.168.1.1

Type escape sequence to abort.
Tracing the route to 209.165.200.238

  1 209.165.200.241 12 msec 12 msec 16 msec
  2 209.165.200.238 28 msec 28 msec *
```

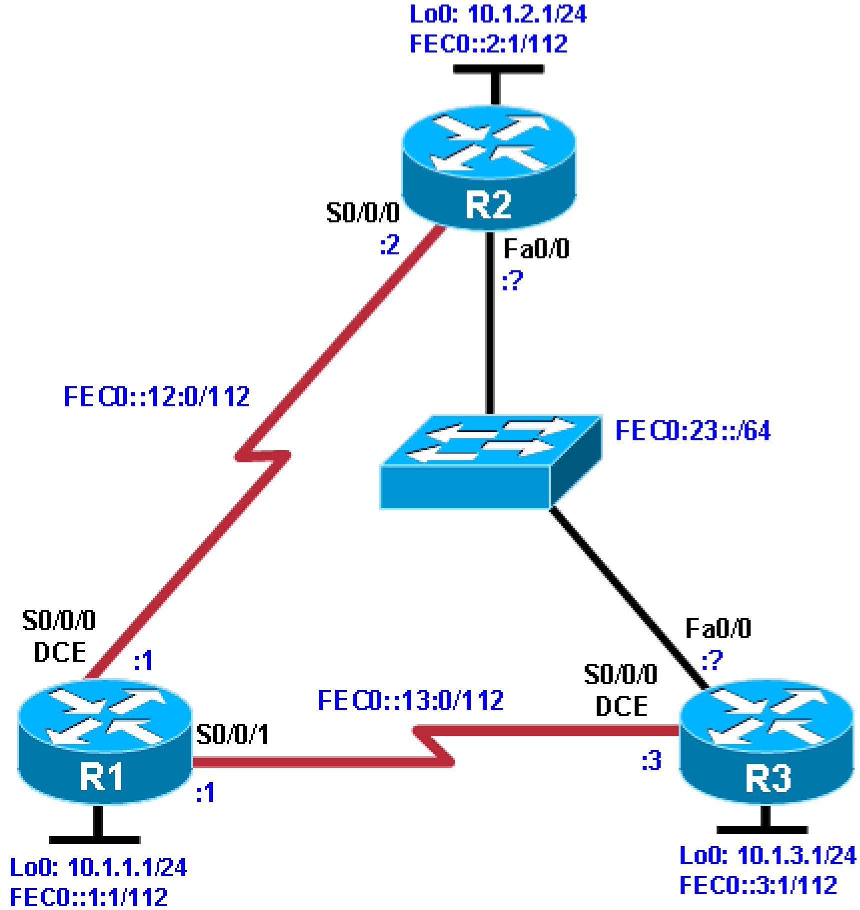
The packet now hops across the ISP router and then to the HQ router. In essence, this proves that Internet-bound traffic will not be encrypted.

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 8 Lab 8-1, Configuring OSPF for IPv6

Topology



Objectives

- Configure a static IPv6 address on an interface.
- Change the default link-local address on an interface.
- Configure an EUI-64 IPv6 address on an interface.
- Enable IPv6 routing and CEF.
- Configure and verify single-area OSPFv3 operation.

Background

In this lab, you configure static IPv6 addresses and EUI-64 IPv6 addresses on interfaces. You then configure OSPFv3 to route between the IPv6 networks.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services Image c1841-advipservicesk9-mz.124-24.T1.bin. The switch is a Cisco WS-C2960-24TT-L with the Cisco IOS

image c2960-lanbasek9-mz.122-46.SE.bin. You can use other routers (such as a 2801 or 2811), switches (such as 2950), and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router or switch model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- 1 switch (Cisco 2960 with the Cisco IOS Release 12.2(46)SE C2960-LANBASEK9-M image or comparable)
- Serial and Ethernet cables

Step 1: Prepare the routers for the lab.

Cable the network as shown in the topology diagram. Erase the startup configuration, and reload each router to clear the previous configurations.

Step 2: Configuring the hostname and loopback interfaces.

Configure the loopback interface on each router with both the IPv4 address and IPv6 address shown in the diagram. The IPv4 address is configured using the **ip address** *address mask* command. The IPv6 address configuration is similar, using the **ipv6 address** *address/mask* command. With IPv6 addresses, you can enter the mask length in bits with a decimal /mask, rather than entering the whole mask out in hexadecimal.

IPv6 addresses consist of eight groups of 16 bits (four hexadecimal characters) separated by colons. You can also enter the IPv6 address in an abbreviated form. For example, you can abbreviate any continuous group of zeros with two colons "::". However, you can only use this abbreviation once per address. Also, leading zeros in each group can be omitted. For example, FEC0:0:0:0:0:0:12:1 /112 can be shortened to FEC0::12:1 /112.

```
Router(config)# hostname R1
R1(config)# interface loopback0
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# ipv6 address FEC0::1:1/112

Router(config)# hostname R2
R2(config)# interface loopback0
R2(config-if)# ip address 10.1.2.1 255.255.255.0
R2(config-if)# ipv6 address FEC0::2:1/112

Router(config)# hostname R3
R3(config)# interface loopback0
R3(config-if)# ip address 10.1.3.1 255.255.255.0
R3(config-if)# ipv6 address FEC0::3:1/112
```

If you accidentally enter the wrong IPv6 address on an interface, make sure you remove it with the **no** version of the command that you entered. Unlike IPv4 addresses, where the **ip address** command overwrites the existing address, multiple IPv6 addresses can exist on an interface. Issuing the **ipv6 address** command multiple times adds more addresses rather than replacing them.

Notice that both an IPv4 and an IPv6 address are on the same interface, and they do not conflict with each other. This is because they are different Layer 3 protocols, and they run independently.

Step 3: Configure static IPv6 addresses.

a. Configure the two serial links with IPv6 addresses. Use the **ipv6 address** *address/mask* command to configure the interfaces with the addresses shown in the diagram. Set the clock rates where appropriate, and bring up the interfaces.

```
R1(config)# interface serial0/0/0
R1(config-if)# ipv6 address FEC0::12:1/112
R1(config-if)# clockrate 64000
R1(config-if)# bandwidth 64
R1(config-if)# no shutdown

R1(config-if)# interface serial0/0/1
R1(config-if)# ipv6 address FEC0::13:1/112
R1(config-if)# bandwidth 64
R1(config-if)# no shutdown

R2(config)# interface serial0/0/0
R2(config-if)# ipv6 address FEC0::12:2/112
R2(config-if)# bandwidth 64
R2(config-if)# no shutdown

R3(config)# interface serial0/0/0
R3(config-if)# ipv6 address FEC0::13:3/112
R3(config-if)# clockrate 64000
R3(config-if)# bandwidth 64
R3(config-if)# no shutdown
```

b. Use **ping** to verify local subnet connectivity.

```
R1# ping FEC0::12:2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::12:2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

R1# ping FEC0::13:3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::13:3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

R2# ping FEC0::12:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::12:1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

R3# ping FEC0::13:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::13:1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```


Step 4: Change the link-local address on an interface.

a. Use the **show ipv6 interface** command to look at IPv6-related properties of the router interfaces. You can also specify a type/number of an interface to see the output of that interface only.

```
R1# show ipv6 interface serial 0/0/0
Serial0/0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::219:6FF:FE23:4380
  No Virtual link-local address(es):
  Global unicast address(es):
    FEC0::12:1, subnet is FEC0::12:0/112
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF12:1
    FF02::1:FF23:4380
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds

R2# show ipv6 interface serial 0/0/0
Serial0/0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::218:B9FF:FE92:28D8
  Global unicast address(es):
    FEC0::12:2, subnet is FEC0::12:0/112
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF12:2
    FF02::1:FF92:28D8
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
```

Notice that in addition to the address that you already configured, there is a link-local address starting with FE80 (your actual address will vary). You can change the addresses on the link between R1 and R2 by putting the link-local address FE80::1 on R1 and FE80::2 on R2. Link-local addresses do not have a subnet mask because they are not routed, hence the term “link-local.”

b. To change the link-local address, use the **ipv6 address** *address* **link-local** command.

```
R1(config)# interface serial0/0/0
R1(config-if)# ipv6 address FE80::1 link-local

R2(config)# interface serial0/0/0
R2(config-if)# ipv6 address FE80::2 link-local
```

c. Verify that you can ping the link-local address on the other side. When pinging link-local addresses, you must specify an outgoing interface because the addresses are not routed and not in the routing table.

Note: When prompted for the output interface with this command, you must use the full interface name without spaces (for example, use serial0/0/0, not s0/0/0).

```
R1# ping FE80::2
Output Interface: Serial0/0/0
```



```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::2, timeout is 2 seconds:
Packet sent with a source address of FE80::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

R2# ping FE80::1
Output Interface: Serial0/0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::1, timeout is 2 seconds:
Packet sent with a source address of FE80::2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```

d. Verify the link-local addresses with the **show ipv6 interface** command.

```
R1# show ipv6 interface serial 0/0/0
Serial0/0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::1
  No Virtual link-local address(es):
  Global unicast address(es):
    FEC0::12:1, subnet is FEC0::12:0/112
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF12:1
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds

R2# show ipv6 interface serial 0/0/0
Serial0/0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::2
  Global unicast address(es):
    FEC0::12:2, subnet is FEC0::12:0/112
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:2
    FF02::1:FF12:2
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
```

Note: Manually modifying a link-local address is seldom needed. It is demonstrated here for the purpose of this lab.

Step 5: Configure EUI-64 addresses.

EUI-64 IPv6 addresses are addresses where the first 64 bits are the network portion of the address and are specified. The second 64 bits are the host portion of the address and are automatically generated by the device. RFC 4921 splits the 48-bit MAC address into two 24-bit segments: the Organizational Unique Identifier (OUI) and the NIC-specific component. The seventh bit of the OUI (bit 7 from the left) is inverted (changed from zero to one or vice versa). The 16-bit hex value 0xFFFE is then inserted between the modified OUI and the NIC-specific component to create the modified EUI-64 address. This can be seen in the highlighted EUI-64 address in Step 5b.

a. Configure IPv6 EUI-64 addresses on an interface using the **ipv6 address** *address/mask* **eui-64** command. Configure the R2 and R3 Fast Ethernet interfaces with the subnet shown in the diagram, and enable the interfaces.

```
R2(config)# interface fastEthernet 0/0
R2(config-if)# ipv6 address FEC0:23::/64 eui-64
R2(config-if)# no shutdown

R3(config)# interface fastEthernet 0/0
R3(config-if)# ipv6 address FEC0:23::/64 eui-64
R3(config-if)# no shutdown
```

b. Get the IPv6 addresses of the interfaces with the **show ipv6 interface** or **show ipv6 interface brief** command, and then ping the other side of the link.

```
R2# show ipv6 interface fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::218:B9FF:FE92:28D8
  No Virtual link-local address(es):
  Global unicast address(es):
    FEC0:23::218:B9FF:FE92:28D8, subnet is FEC0:23::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::1:FF92:28D8
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 17162)

R2# show ipv6 interface brief
FastEthernet0/0            [up/up]
    FE80::218:B9FF:FE92:28D8
    FEC0:23::218:B9FF:FE92:28D8
FastEthernet0/1            [administratively down/down]
Serial0/0/0                [up/up]
    FE80::2
    FEC0::12:2
Serial0/0/1                [administratively down/down]
Serial0/1/0                [administratively down/down]
Serial0/1/1                [administratively down/down]
Loopback0                  [up/up]
    FE80::218:B9FF:FE92:28D8
    FEC0::2:1

R3# show ipv6 interface brief
FastEthernet0/0            [up/up]
    FE80::218:B9FF:FECD:BEF0
    FEC0:23::218:B9FF:FECD:BEF0
FastEthernet0/1            [administratively down/down]
Serial0/0/0                [up/up]
    FE80::218:B9FF:FECD:BEF0
    FEC0::13:3
Serial0/0/1                [administratively down/down]
```



```
Serial0/1/0                [administratively down/down]
Serial0/1/1                [administratively down/down]
Loopback0                  [up/up]
    FE80::218:B9FF:FECD:BEF0
    FEC0::3:1

R2# ping FEC0:23::218:B9FF:FECD:BEF0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:23::218:B9FF:FECD:BEF0, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/4 ms

R3# ping FEC0:23::218:B9FF:FE92:28D8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:23::218:B9FF:FE92:28D8, timeout is 2
seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

Note: Your addresses will be different from the addresses displayed in the example because EUI-64 addresses include the MAC address of the interface, which is unique per interface.

At this point, you should have local subnet connectivity.

Note: Although not configured in this lab, the **ipv6 general-prefix** command can simplify the configuration of IPv6 addresses, including the EUI-64. You can use this command to assign a name to an often-used IPv6 prefix and then use the name to configure interfaces.

Step 6: Enable IPv6 routing and CEF.

The Cisco IOS version used with the routers in this lab has IPv6 routing and IPv6 CEF disabled by default. To enable IPv6 routing, use the **ipv6 unicast-routing** command in global configuration mode. To enable IPv6 CEF, use the **ipv6 cef** command. Enter these commands on all three routers.

```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 cef

R2(config)# ipv6 unicast-routing
R2(config)# ipv6 cef

R3(config)# ipv6 unicast-routing
R3(config)# ipv6 cef
```

Step 7: Configure OSPFv3.

Unlike IPv4 OSPF, where networks are added to the OSPF process using **network** statements under the routing protocol configuration prompt, IPv6 OSPF uses the interface-level command **ipv6 ospf** *process* **area** *area-id* to add an interface to an area.

a. Add all interfaces shown in the diagram into OSPF process 1, area 0. After you add the interfaces to the OSPF process, the OSPF process starts automatically. If the adjacencies do not come up after a reasonable period of time, troubleshoot using the **debug ipv6 ospf adjacency** and **debug ipv6 packet** commands. Make sure that the packets are being sent to their destination and that adjacencies are forming correctly.

```
R1(config)# interface loopback0
```



```
R1(config-if)# ipv6 ospf 1 area 0
R1(config-if)# interface serial0/0/0
R1(config-if)# ipv6 ospf 1 area 0
R1(config-if)# interface serial0/0/1
R1(config-if)# ipv6 ospf 1 area 0

R2(config)# interface loopback0
R2(config-if)# ipv6 ospf 1 area 0
R2(config-if)# interface serial0/0/0
R2(config-if)# ipv6 ospf 1 area 0
R2(config-if)# interface fastEthernet 0/0
R2(config-if)# ipv6 ospf 1 area 0

R3(config)# interface loopback0
R3(config-if)# ipv6 ospf 1 area 0
R3(config-if)# interface serial0/0/0
R3(config-if)# ipv6 ospf 1 area 0
R3(config-if)# interface fastEthernet 0/0
R3(config-if)# ipv6 ospf 1 area 0
```

b. Verify that you have OSPFv3 neighbors with the **show ipv6 ospf neighbor** command.

```
R1# show ipv6 ospf neighbor

Neighbor ID     Pri   State         Dead Time   Interface ID   Interface
10.1.3.1          1   FULL/  -      00:00:39    6              Serial0/0/1
10.1.2.1          1   FULL/  -      00:00:34    6              Serial0/0/0

R2# show ipv6 ospf neighbor

Neighbor ID     Pri   State         Dead Time   Interface ID   Interface
10.1.3.1          1   FULL/DR       00:00:39    4              FastEthernet0/0
10.1.1.1          1   FULL/  -      00:00:32    6              Serial0/0/0

R3# show ipv6 ospf neighbor

Neighbor ID     Pri   State         Dead Time   Interface ID   Interface
10.1.2.1          1   FULL/BDR      00:00:39    4              FastEthernet0/0
10.1.1.1          1   FULL/  -      00:00:39    7              Serial0/0/0
```

The router IDs for each router are created the same way that they are in OSPFv2 or BGP. Without any IPv4 addresses on the router, the OSPFv3 process will not start unless you manually set the router IDs. This is why the loopback interfaces have been configured with both IPv4 and IPv6 addresses.

c. View the routing table on all three routers with the **show ipv6 route** command.

```
R1# show ipv6 route
IPv6 Routing Table - 11 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
L   FE80::/10 [0/0]
     via ::, Null0
C   FEC0::1:0/112 [0/0]
     via ::, Loopback0
L   FEC0::1:1/128 [0/0]
     via ::, Loopback0
```



```
O   FEC0::2:1/128 [110/1562]
     via FE80::2, Serial0/0/0
O   FEC0::3:1/128 [110/1562]
     via FE80::218:B9FF:FECD:BEF0, Serial0/0/1
C   FEC0::12:0/112 [0/0]
     via ::, Serial0/0/0
L   FEC0::12:1/128 [0/0]
     via ::, Serial0/0/0
C   FEC0::13:0/112 [0/0]
     via ::, Serial0/0/1
L   FEC0::13:1/128 [0/0]
     via ::, Serial0/0/1
O   FEC0:23::/64 [110/1563]
     via FE80::2, Serial0/0/0
     via FE80::218:B9FF:FECD:BEF0, Serial0/0/1
L   FF00::/8 [0/0]
     via ::, Null0

R2# show ipv6 route
IPv6 Routing Table - 11 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
L   FE80::/10 [0/0]
     via ::, Null0
O   FEC0::1:1/128 [110/1562]
     via FE80::1, Serial0/0/0
C   FEC0::2:0/112 [0/0]
     via ::, Loopback0
L   FEC0::2:1/128 [0/0]
     via ::, Loopback0
O   FEC0::3:1/128 [110/1]
     via FE80::218:B9FF:FECD:BEF0, FastEthernet0/0
C   FEC0::12:0/112 [0/0]
     via ::, Serial0/0/0
L   FEC0::12:2/128 [0/0]
     via ::, Serial0/0/0
O   FEC0::13:0/112 [110/1563]
     via FE80::218:B9FF:FECD:BEF0, FastEthernet0/0
C   FEC0:23::/64 [0/0]
     via ::, FastEthernet0/0
L   FEC0:23::218:B9FF:FE92:28D8/128 [0/0]
     via ::, FastEthernet0/0
L   FF00::/8 [0/0]
     via ::, Null0

R3# show ipv6 route
IPv6 Routing Table - 11 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
L   FE80::/10 [0/0]
     via ::, Null0
O   FEC0::1:1/128 [110/1562]
```



```
       via FE80::219:6FF:FE23:4380, Serial0/0/0
O   FEC0::2:1/128 [110/1]
     via FE80::218:B9FF:FE92:28D8, FastEthernet0/0
C   FEC0::3:0/112 [0/0]
     via ::, Loopback0
L   FEC0::3:1/128 [0/0]
     via ::, Loopback0
O   FEC0::12:0/112 [110/1563]
     via FE80::218:B9FF:FE92:28D8, FastEthernet0/0
C   FEC0::13:0/112 [0/0]
     via ::, Serial0/0/0
L   FEC0::13:3/128 [0/0]
     via ::, Serial0/0/0
C   FEC0:23::/64 [0/0]
     via ::, FastEthernet0/0
L   FEC0:23::218:B9FF:FECD:BEF0/128 [0/0]
     via ::, FastEthernet0/0
L   FF00::/8 [0/0]
     via ::, Null0
```

d. You can also look at per-interface OSPF behavior with the **show ipv6 ospf interface** command.

```
R1# show ipv6 ospf interface
Serial0/0/1 is up, line protocol is up
  Link Local Address FE80::219:6FF:FE23:4380, Interface ID 7
  Area 0, Process ID 1, Instance ID 0, Router ID 10.1.1.1
  Network Type POINT_TO_POINT, Cost: 1562
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:06
  Index 1/3/3, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 2, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.3.1
  Suppress hello for 0 neighbor(s)
Serial0/0/0 is up, line protocol is up
  Link Local Address FE80::1, Interface ID 6
  Area 0, Process ID 1, Instance ID 0, Router ID 10.1.1.1
  Network Type POINT_TO_POINT, Cost: 1562
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:00
  Index 1/2/2, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 4
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.2.1
  Suppress hello for 0 neighbor(s)
Loopback0 is up, line protocol is up
  Link Local Address FE80::219:6FF:FE23:4380, Interface ID 20
  Area 0, Process ID 1, Instance ID 0, Router ID 10.1.1.1
  Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host

R2# show ipv6 ospf interface
FastEthernet0/0 is up, line protocol is up
```

```
  Link Local Address FE80::218:B9FF:FE92:28D8, Interface ID 4
  Area 0, Process ID 1, Instance ID 0, Router ID 10.1.2.1
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 10.1.3.1, local address FE80::218:B9FF:FECD:BEF0
  Backup Designated router (ID) 10.1.2.1, local address
FE80::218:B9FF:FE92:28D8
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:04
  Index 1/3/3, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 2, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.3.1  (Designated Router)
  Suppress hello for 0 neighbor(s)
Serial0/0/0 is up, line protocol is up
  Link Local Address FE80::2, Interface ID 6
  Area 0, Process ID 1, Instance ID 0, Router ID 10.1.2.1
  Network Type POINT_TO_POINT, Cost: 1562
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:07
  Index 1/2/2, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 4
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.1.1
  Suppress hello for 0 neighbor(s)
Loopback0 is up, line protocol is up
  Link Local Address FE80::218:B9FF:FE92:28D8, Interface ID 17
  Area 0, Process ID 1, Instance ID 0, Router ID 10.1.2.1
  Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host

R3# show ipv6 ospf interface
FastEthernet0/0 is up, line protocol is up
  Link Local Address FE80::218:B9FF:FECD:BEF0, Interface ID 4
  Area 0, Process ID 1, Instance ID 0, Router ID 10.1.3.1
  Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.1.3.1, local address FE80::218:B9FF:FECD:BEF0
  Backup Designated router (ID) 10.1.2.1, local address
FE80::218:B9FF:FE92:28D8
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
  Index 1/3/3, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 4
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.2.1  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
Serial0/0/0 is up, line protocol is up
  Link Local Address FE80::218:B9FF:FECD:BEF0, Interface ID 6
  Area 0, Process ID 1, Instance ID 0, Router ID 10.1.3.1
  Network Type POINT_TO_POINT, Cost: 1562
```



```
    Transmit Delay is 1 sec, State POINT_TO_POINT,
    Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello due in 00:00:07
    Index 1/2/2, flood queue length 0
    Next 0x0(0)/0x0(0)/0x0(0)
    Last flood scan length is 1, maximum is 4
    Last flood scan time is 0 msec, maximum is 0 msec
    Neighbor Count is 1, Adjacent neighbor count is 1
      Adjacent with neighbor 10.1.1.1
    Suppress hello for 0 neighbor(s)
  Loopback0 is up, line protocol is up
    Link Local Address FE80::218:B9FF:FECD:BEF0, Interface ID 17
    Area 0, Process ID 1, Instance ID 0, Router ID 10.1.3.1
    Network Type LOOPBACK, Cost: 1
    Loopback interface is treated as a stub Host
```

e. Run the following Tcl script on all routers to verify full connectivity. If these pings are not successful, troubleshoot. Modify the script to include the correct EUI addresses on the FEC0:23:: /64 subnet. The addresses for the router interfaces used in this lab are shown below.

```
tclsh

foreach address {
FEC0::1:1
FEC0::2:1
FEC0::3:1
FEC0::12:1
FEC0::12:2
FEC0::13:1
FEC0::13:3
FEC0:23::
FEC0:23::
} {
ping $address }

R1#tclsh
R1(tcl)#
R1(tcl)#foreach address {
+>(tcl)#FEC0::1:1
+>(tcl)#FEC0::2:1
+>(tcl)#FEC0::3:1
+>(tcl)#FEC0::12:1
+>(tcl)#FEC0::12:2
+>(tcl)#FEC0::13:1
+>(tcl)#FEC0::13:3
+>(tcl)#FEC0:23::218:B9FF:FE92:28D8
+>(tcl)#FEC0:23::218:B9FF:FECD:BEF0
+>(tcl)#} {
+>(tcl)#ping $address }
```

Challenge: Summarize OSPFv3 Areas

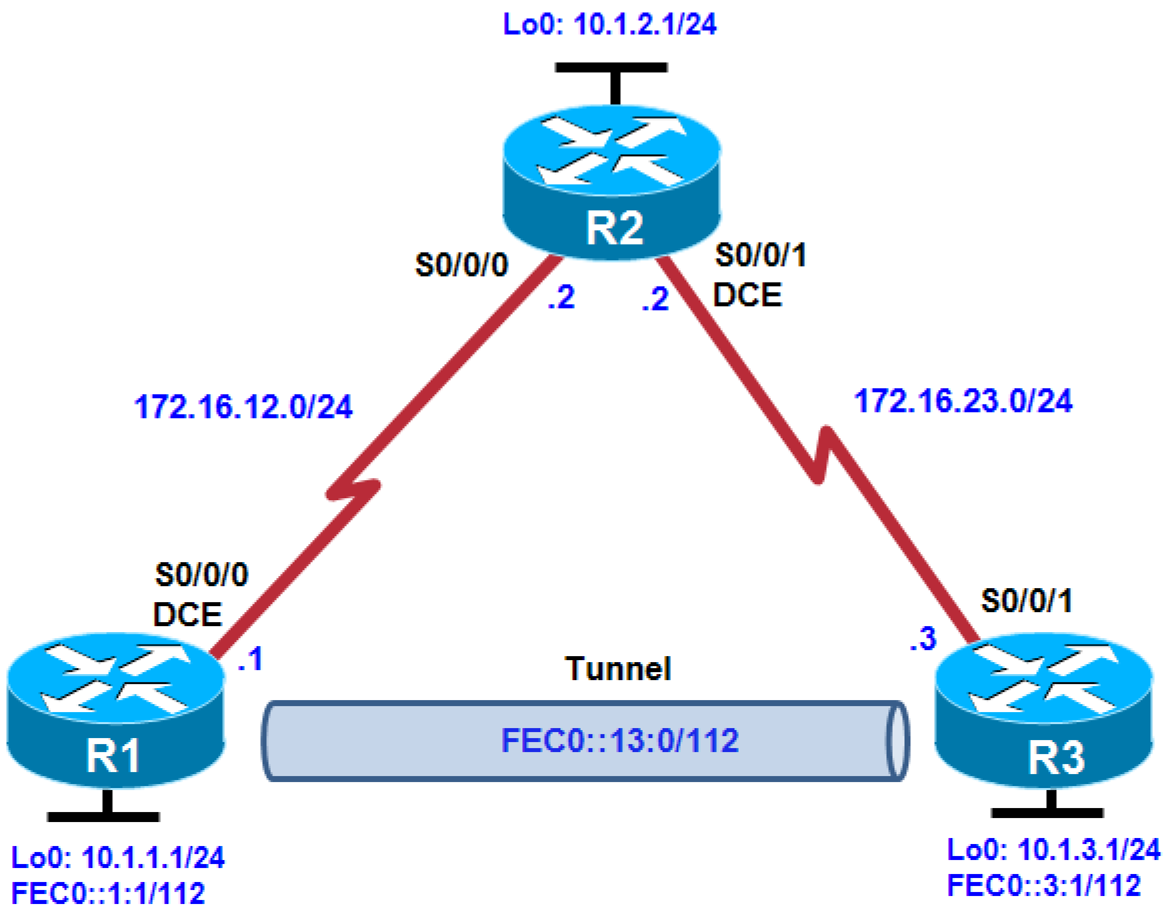
The commands available for OSPFv3 are similar to the commands available for OSPFv2. On R2, add in two loopback interfaces, with the addresses FEC0:500::100:1 /112 and FEC0:500::200:1 /112. Add both of these interfaces to the OSPF process in area 500. Summarize area 500 to FEC0:500:: /64. To enter the OSPF configuration prompt, use the **ipv6 router ospf** *process-id* command. Unlike the IPv4 (config-router) prompt, the IPv6 router prompt is (config-rtr). When in the IPv6 router prompt, use the **area range** command to summarize the address. Use the question mark (?) if you need help.

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 8 Lab 8-2, Using Manual IPv6 Tunnels with EIGRP for IPv6

Topology



Objectives

- Configure EIGRP for IPv4.
- Create a manual IPv6 tunnel.
- Configure EIGRP for IPv6 across the tunnel.

Background

In this lab, you configure EIGRP for full connectivity between all IPv4 subnets. Then you create a manual IPv6 tunnel and run EIGRP for IPv6 over it.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router

model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers for the lab.

Cable the network as shown in the topology diagram. Erase the startup configuration, and reload each router to clear the previous configurations. Configure the hostnames as shown.

Step 2: Configure loopbacks and physical interfaces.

Configure the loopback interfaces with IPv4 addresses and IPv6 addresses, where appropriate. Also configure the serial interfaces with the IPv4 addresses shown in the diagram. Set the clock rates on the appropriate interfaces, and issue the **no shutdown** command on all serial connections. Verify that you have local subnet connectivity with **ping**.

```
R1(config)# interface loopback0
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# ipv6 address FEC0::1:1/112
R1(config-if)# interface serial0/0/0
R1(config-if)# ip address 172.16.12.1 255.255.255.0
R1(config-if)# clockrate 64000
R1(config-if)# bandwidth 64
R1(config-if)# no shutdown

R2(config)# interface loopback0
R2(config-if)# ip address 10.1.2.1 255.255.255.0
R2(config-if)# interface serial0/0/0
R2(config-if)# ip address 172.16.12.2 255.255.255.0
R2(config-if)# bandwidth 64
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)# clockrate 64000
R2(config-if)# bandwidth 64
R2(config-if)# no shutdown

R3(config)# interface loopback0
R3(config-if)# ip address 10.1.3.1 255.255.255.0
R3(config-if)# ipv6 address FEC0::3:1/112
R3(config-if)# interface serial0/0/1
R3(config-if)# ip address 172.16.23.3 255.255.255.0
R3(config-if)# bandwidth 64
R3(config-if)# no shutdown
```

Step 3: Configure EIGRP for IPv4.

Configure EIGRP for AS 1 for the major networks 172.16.0.0 and 10.0.0.0 on all three routers. Make sure that you disable auto-summarization. You should have full IPv4 connectivity after this.

```
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# network 10.0.0.0
R1(config-router)# network 172.16.0.0
```



```
R2(config)# router eigrp 1
R2(config-router)# no auto-summary
R2(config-router)# network 10.0.0.0
R2(config-router)# network 172.16.0.0

R3(config)# router eigrp 1
R3(config-router)# no auto-summary
R3(config-router)# network 10.0.0.0
R3(config-router)# network 172.16.0.0
```

Step 4: Configure a manual IPv6 tunnel.

A tunnel is a logical interface that acts as a logical connection between two endpoints. It is similar to a loopback interface in that there is no corresponding physical interface, but it is different in that there is more than one router involved. An IPv6 manual tunnel is a type of tunnel that has hard-coded source and destination addresses, with an IPv6 address on the tunnel itself.

a. Use the **interface tunnel** *number* command to create a manual tunnel. For simplicity, use tunnel number 0 on both routers. Configure the tunnel mode for a manual tunnel with the **tunnel mode ipv6ip** command. Then configure an IPv6 address with the **ipv6 address** *address/mask* command. Finally, assign source and destination addresses for the tunnel using the **tunnel source** *address* and **tunnel destination** *address* commands. You can also specify the source by interface.

```
R1(config)# interface tunnel0
R1(config-if)# tunnel mode ipv6ip
R1(config-if)# tunnel source serial0/0/0
R1(config-if)# tunnel destination 172.16.23.3
R1(config-if)# ipv6 address FEC0::13:1/112

R3(config)# interface tunnel0
R3(config-if)# tunnel mode ipv6ip
R3(config-if)# tunnel source serial0/0/1
R3(config-if)# tunnel destination 172.16.12.1
R3(config-if)# ipv6 address FEC0::13:3/112
```

b. Verify that you can ping across the tunnel from one side to the other using the tunnel address of the opposite router.

```
R1# ping FEC0::13:3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::13:3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/66/68 ms

R3# ping FEC0::13:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::13:1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/66/68 ms
```

Note: Although not done in this lab, you can configure the tunnel interfaces using only the **ipv6 enable** command, instead of unicast IPv6 addresses. This command configures the tunnel interfaces for IPv6 operation and assigns automatically generated link-local addresses to them. This allows the tunnel to transport IPv6 packets and an IGP can be run over it. The individual endpoints of the tunnel are not globally addressable, but the tunnel does not require global unicast addresses.

Step 5: Configure EIGRP for IPv6 over a tunnel.

a. Enable IPv6 routing with the **ipv6 unicast-routing** command on R1 and R3. Configure EIGRP for IPv6 on those routers to run over the tunnel and advertise the loopback interfaces into IPv6-EIGRP AS 100.

```
R1(config)# ipv6 unicast-routing
R1(config)# interface loopback0
R1(config-if)# ipv6 eigrp 100
R1(config-if)# interface tunnel0
R1(config-if)# ipv6 eigrp 100

R3(config)# ipv6 unicast-routing
R3(config)# interface loopback0
R3(config-if)# ipv6 eigrp 100
R3(config-if)# interface tunnel0
R3(config-if)# ipv6 eigrp 100
```

b. Verify the configuration using the **show ipv6 eigrp neighbor** command.

```
R1# show ipv6 eigrp neighbor
IPv6-EIGRP neighbors for process 100
% EIGRP 100 is in SHUTDOWN
```

c. IPv6 EIGRP routing is shut down by default. To enable IPv6-EIGRP for process 100, use the following commands on R1 and R3.

```
R1(config-if)# ipv6 router eigrp 100
R1(config-rtr)# no shutdown

*Apr 19 17:27:08.639: %DUAL-5-NBRCHANGE: IPv6-EIGRP(0) 100: Neighbor
FE80::AC10:
1703 (Tunnel0) is up: new adjacency
```

d. Verify the configuration using the **show ipv6 eigrp neighbors** command.

```
R1# show ipv6 eigrp neighbors

IPv6-EIGRP neighbors for process 100
H   Address                  Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                             (sec)         (ms)       Cnt Num
0   Link-local address:      Tu0               13 00:01:18  104  5000  0  3
    FE80::AC10:1703
```

Note: The link-local neighbor address is the IPv6 equivalent (AC10:1703) of the R3 serial interface IPv4 address (172.16.23.3).

e. Ping the R3 Lo0 IPv6 address from R1, and ping the R1 Lo0 IPv6 address from R3.

```
R1# ping FEC0::3:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::3:1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/68 ms

R3# ping FEC0::1:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::1:1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/66/68 ms
```


f. Use the following Tcl script on R1 and R3 to verify connectivity for R1 and R3.

```
R1# tclsh

foreach address {
10.1.1.1
10.1.2.1
10.1.3.1
172.16.12.1
172.16.12.2
172.16.23.2
172.16.23.3
FEC0::1:1
FEC0::3:1
FEC0::13:1
FEC0::13:3
} {
ping $address }
```

Were all pings successful? ____

g. Run the Tcl script on R2. Were all IP addresses reachable? Explain.

____

____

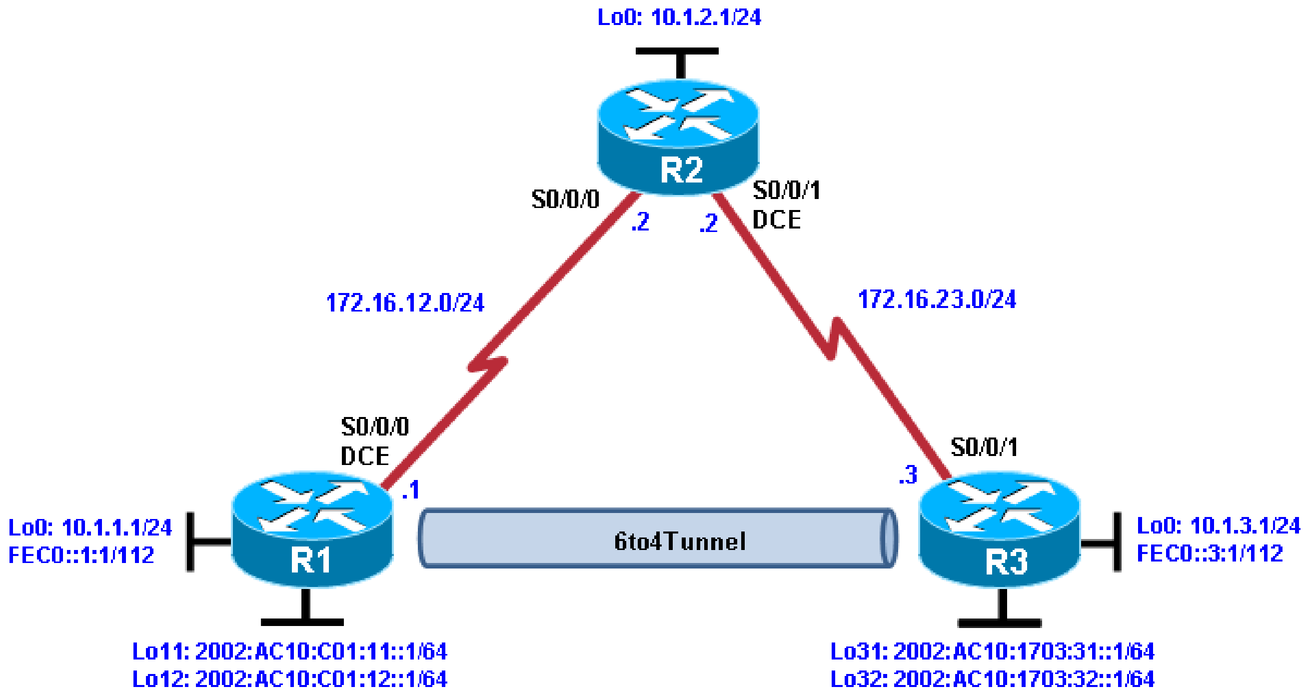
____

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 8 Lab 8-3, Configuring 6to4 Tunnels

Topology



Objectives

- Configure EIGRP for IPv4.
- Create a 6to4 tunnel.
- Configure static IPv6 routes.

Background

In this lab, you configure EIGRP for full connectivity between all IPv4 subnets. You then create a 6to4 tunnel and create static routes over it.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 3 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Step 1: Prepare the routers for the lab.

Clear previous configurations by erasing the startup configuration and reloading the routers. After the routers are reloaded, set up the appropriate hostnames.

Step 2: Configure loopbacks and physical interfaces.

a. Configure the loopback interfaces with IPv4 addresses and IPv6 addresses, where appropriate. Also configure the serial interfaces with the IPv4 addresses shown in the diagram. Set the clock rates on the appropriate interfaces and the bandwidth on all serial connections.

```
R1(config)# interface loopback0
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# ipv6 address FEC0::1:1/112
R1(config-if)# interface serial0/0/0
R1(config-if)# ip address 172.16.12.1 255.255.255.0
R1(config-if)# clockrate 64000
R1(config-if)# bandwidth 64
R1(config-if)# no shutdown

R2(config)# interface loopback0
R2(config-if)# ip address 10.1.2.1 255.255.255.0
R2(config-if)# interface serial0/0/0
R2(config-if)# ip address 172.16.12.2 255.255.255.0
R2(config-if)# bandwidth 64
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)# clockrate 64000
R2(config-if)# bandwidth 64
R2(config-if)# no shutdown

R3(config)# interface loopback0
R3(config-if)# ip address 10.1.3.1 255.255.255.0
R3(config-if)# ipv6 address FEC0::3:1/112
R3(config-if)# interface serial0/0/1
R3(config-if)# ip address 172.16.23.3 255.255.255.0
R3(config-if)# bandwidth 64
R3(config-if)# no shutdown
```

b. Verify that you have local subnet connectivity with **ping**.

Step 3: Configure EIGRP.

Configure EIGRP for AS 1 for the major networks 172.16.0.0 and 10.0.0.0 on all three routers. You should have full IPv4 connectivity.

```
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# network 10.0.0.0
R1(config-router)# network 172.16.0.0

R2(config)# router eigrp 1
R2(config-router)# no auto-summary
R2(config-router)# network 10.0.0.0
R2(config-router)# network 172.16.0.0

R3(config)# router eigrp 1
R3(config-router)# no auto-summary
```



```
R3(config-router)# network 10.0.0.0
R3(config-router)# network 172.16.0.0
```

Step 4: Configure a manual IPv6 6to4 tunnel.

A tunnel is a logical interface that acts as a logical connection between two endpoints. It is similar to a loopback interface in that there is no corresponding physical interface, but it is different in that there is more than one router involved. A 6to4 tunnel uses special IPv6 addresses in the 2002::/16 address space. The first 16 bits are the hexadecimal number 2002, and the next 32 bits are the original source IPv4 address in hexadecimal form. The remaining bits can be specified as shown in Step 4c. A 6to4 tunnel does not require a destination address because it is not a point-to-point link.

In this step, you configure additional 6to4 IPv6 loopback interfaces to represent LANs on R1 and R3, and then configure a 6to4 tunnel to provide IPv6 connectivity between the loopback interfaces.

a. Add the following loopbacks to R1 and R3.

```
R1(config-if)# interface loopback11
R1(config-if)# ipv6 address 2002:AC10:0C01:11::1/64
R1(config-if)# interface loopback12
R1(config-if)# ipv6 address 2002:AC10:0C01:12::1/64

R3(config-if)# interface loopback31
R3(config-if)# ipv6 address 2002:AC10:1703:31::1/64
R3(config-if)# interface loopback32
R3(config-if)# ipv6 address 2002:AC10:1703:32::1/64
```

b. Configure a 6to4 tunnel using the **interface tunnel** *number* command to get to the tunnel interface configuration prompt. For simplicity, use interface number 0.

```
R1(config)# interface tunnel 0
```

c. Set the tunnel mode with the **tunnel mode ipv6ip 6to4** command. Then set up the IPv6 address with the **ipv6 address** *address*/*mask* command. The R1 address is 2002:AC10:0C01:1::1/64, because AC10:0C01 corresponds to 172.16.12.1, with 172 being AC, 16 being 10, 12 being C, and 1 being 1. The 1 after this address is just a more specific subnet, and the 1 at the end is the host address. The R3 address is 2002:AC10:1703:1::3/64. The two addresses are not in the same /64 subnet. After setting the IPv6 addresses, set the source interface for the tunnel with the **tunnel source** *type*/*number* command.

```
R1(config-if)# tunnel mode ipv6ip 6to4
R1(config-if)# ipv6 address 2002:AC10:0C01:1::1/64
R1(config-if)# tunnel source serial0/0/0
```

d. Now that all the tunnel settings are set, enable IPv6 routing with the **ipv6 unicast-routing** command, and set up an IPv6 static route for the whole 2002::/16 network with the global command **ipv6 route** *address*/*mask interface*, with the interface being the tunnel you just created.

```
R1(config-if)# exit
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 route 2002::/16 tunnel0
```

e. Enter the following commands on R3.

```
R3(config)# interface tunnel 0
R3(config-if)# tunnel mode ipv6ip 6to4
R3(config-if)# ipv6 address 2002:AC10:1703:1::3/64
R3(config-if)# tunnel source serial0/0/1
R3(config-if)# exit
R3(config)# ipv6 unicast-routing
R3(config)# ipv6 route 2002::/16 tunnel0
```

f. Verify that you can ping the other side of the tunnel from R1 to R3, and ping the newly created loopback interfaces from each router .

```
R1# ping 2002:AC10:1703:1::3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002:AC10:1703:1::3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/67/68 ms

R1# ping 2002:AC10:1703:31::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002:AC10:1703:31::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/65/68 ms

R1# ping 2002:AC10:1703:32::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002:AC10:1703:32::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/65/68 ms

R3# ping 2002:AC10:C01:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002:AC10:C01:1::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/66/68 ms

R3# ping 2002:AC10:0C01:11::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002:AC10:C01:11::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/67/68 ms

R3# ping 2002:AC10:0C01:12::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2002:AC10:C01:12::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/65/68 ms
```

Step 5: Configure static IPv6 routes.

Just like IPv4, IPv6 can have static routes entered into its routing table. You already created one for the 2002::/16 network in Step 4. Now you will configure a static route on R1 telling it how to get to the R3 loopback 0 address. On R3, you will configure a static route pointing to R1.

a. Static routes with a next-hop IPv6 address are created with the **ipv6 route** *address*/*mask next-hop* command. The next hop for both routers is the IPv6 address of the other end of the tunnel.

```
R1(config)# ipv6 route FEC0::3:0/112 2002:AC10:1703:1::3

R3(config)# ipv6 route FEC0::1:0/112 2002:AC10:C01:1::1
```


b. Verify the IPv6 static routes using the **show ipv6 route** command or by pinging the remote loopback address from each router.

```
R1# show ipv6 route
IPv6 Routing Table - Default - 11 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   2002::/16 [1/0]
     via Tunnel0, directly connected
C   2002:AC10:C01:1::/64 [0/0]
     via Tunnel0, directly connected
L   2002:AC10:C01:1::1/128 [0/0]
     via Tunnel0, receive
C   2002:AC10:C01:11::/64 [0/0]
     via Loopback11, directly connected
L   2002:AC10:C01:11::1/128 [0/0]
     via Loopback11, receive
C   2002:AC10:C01:12::/64 [0/0]
     via Loopback12, directly connected
L   2002:AC10:C01:12::1/128 [0/0]
     via Loopback12, receive
C   FEC0::1:0/112 [0/0]
     via Loopback0, directly connected
L   FEC0::1:1/128 [0/0]
     via Loopback0, receive
S   FEC0::3:0/112 [1/0]
     via 2002:AC10:1703:1::3
L   FF00::/8 [0/0]
     via Null0, receive

R3# show ipv6 route
IPv6 Routing Table - Default - 11 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   2002::/16 [1/0]
     via Tunnel0, directly connected
C   2002:AC10:1703:1::/64 [0/0]
     via Tunnel0, directly connected
L   2002:AC10:1703:1::3/128 [0/0]
     via Tunnel0, receive
C   2002:AC10:1703:31::/64 [0/0]
     via Loopback31, directly connected
L   2002:AC10:1703:31::1/128 [0/0]
     via Loopback31, receive
C   2002:AC10:1703:32::/64 [0/0]
     via Loopback32, directly connected
L   2002:AC10:1703:32::1/128 [0/0]
     via Loopback32, receive
S   FEC0::1:0/112 [1/0]
     via 2002:AC10:C01:1::1
```

```
C   FEC0::3:0/112 [0/0]
     via Loopback0, directly connected
L   FEC0::3:1/128 [0/0]
     via Loopback0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

c. From R1 and R3, ping the loopback 0 IPv6 address of the opposite router.

```
R1# ping FEC0::3:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::3:1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/67/68 ms

R3# ping FEC0::1:1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0::1:1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/66/68 ms
```

d. Use the following Tcl script on R1 and R3 to verify network connectivity.

```
R1# tclsh

foreach address {
10.1.1.1
10.1.2.1
10.1.3.1
172.16.12.1
172.16.12.2
172.16.23.2
172.16.23.3
FEC0::1:1
FEC0::3:1
2002:AC10:C01:1::1
2002:AC10:1703:1::3
2002:AC10:1703:31::1
2002:AC10:1703:32::1
2002:AC10:0C01:11::1
2002:AC10:0C01:12::1
} {
ping $address }
```

Were all pings successful?

____

e. Run the Tcl script on R2. Were all IP addresses reachable? Explain.

____

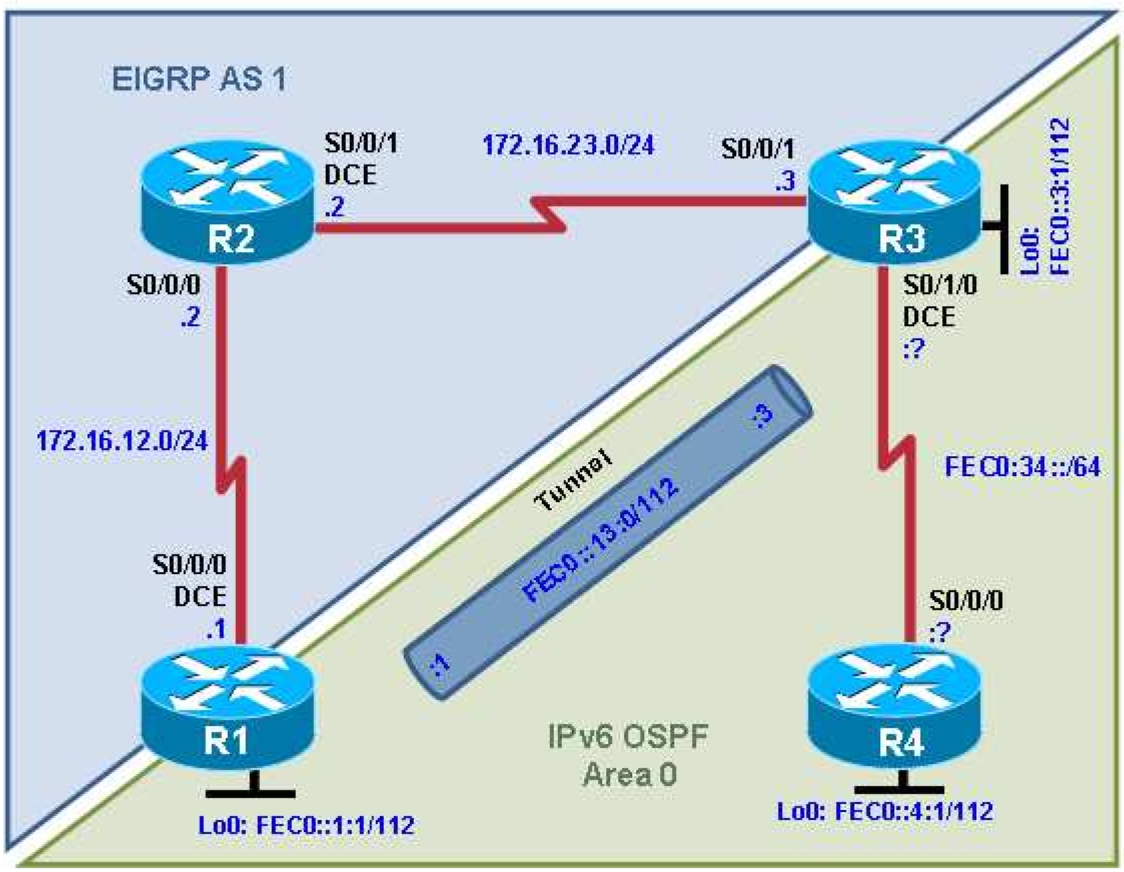
____

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 8 Lab 8-4, IPv6 Challenge Lab

Topology



Objectives

- Implement the topology diagram using the instructions in the Requirements section.
- Change the IPv6 IGP from OSPFv3 to RIPng.

Background

In the first part of this lab (Steps 1 through 7), you configure IPv4 with EIGRP on routers R1, R2 and R3. You also configure IPv6 with OSPFv3 on routers R1, R3 and R4, create an IPv6 tunnel between R1 and R3 and then test network connectivity. In the second part of the lab (Step 8), you replace the OSPFv3 routing protocol with RIPng and re-test connectivity.

Required Resources

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Requirements

1. Configure all interfaces in the topology diagram with the IPv4 or IPv6 addresses shown.
2. Use EUI-64 addresses on the link between R3 and R4.
3. Configure EIGRP AS 1 on R1, R2, and R3 to route all IPv4 networks.
4. Disable EIGRP automatic summarization.
5. Configure a manual IPv6 tunnel between R1 and R3.
6. Include all IPv6 networks in OSPF area 0 on R1, R3, and R4.
7. Manually configure a router ID of 172.16.4.1 on R4 (this address does not need to be reachable).
8. Remove the OSPFv3 configuration commands from R1, R3, and R4 and configure RIPng to run on these routers using RIP1 as the process name.

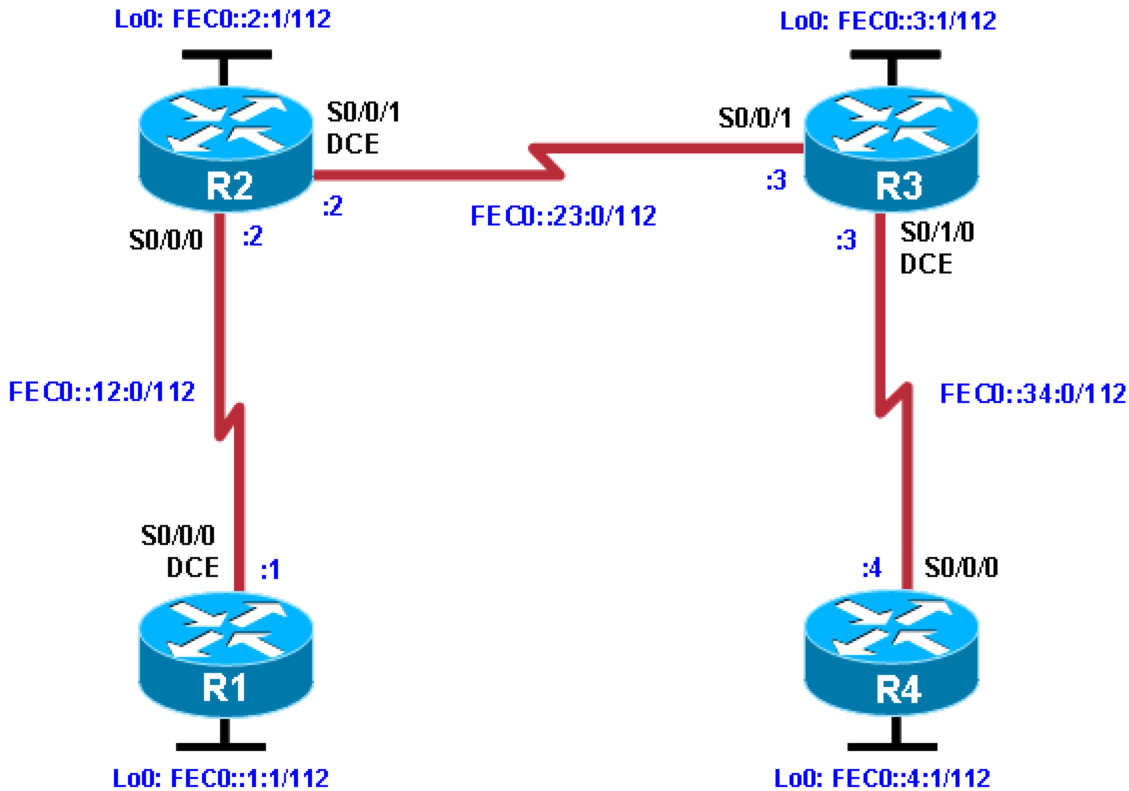
Notes: ____

Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				

Chapter 8 Lab 8-5, IPv6 Troubleshooting Lab

Topology



Objective

- Troubleshoot and correct issues in an IPv6 topology.

Background

In this lab, you troubleshoot existing configurations to get a working topology. Copy and paste the initial configurations from this lab into the routers. Some of these configurations have intentional errors introduced. Your goal is to troubleshoot and correct any problems in the scenario that prevent full IPv6 connectivity. Full IPv6 connectivity means every address in the scenario is reachable from every router. If you do not know where to start, try pinging remote addresses and see which ones are reachable by either manually performing pings or using a Tcl script.

Note: This lab uses Cisco 1841 routers with Cisco IOS Release 12.4(24)T1 and the Advanced IP Services image c1841-advipservicesk9-mz.124-24.T1.bin. You can use other routers (such as a 2801 or 2811) and Cisco IOS Software versions if they have comparable capabilities and features. Depending on the router model and Cisco IOS Software version, the commands available and output produced might vary from what is shown in this lab.

Required Resources

- 4 routers (Cisco 1841 with Cisco IOS Release 12.4(24)T1 Advanced IP Services or comparable)
- Serial and console cables

Requirements

1. Use the IPv6 addressing scheme shown in the diagram.
2. All interfaces must be in OSPFv3 area 0.
3. The router IDs are manually configured with the initial configurations.
4. Do not use static routes, default routes, or routing protocols other than OSPFv3.
5. All IPv6 addresses in the topology must be reachable from all routers.

Notes: ______________________________

Initial Configurations

Router R1

```
hostname R1
!
ipv6 unicast-routing
!
interface Loopback0
 ipv6 address FEC0::1:1/112
!
interface Serial0/0/0
 ipv6 address FEC0::12:1/112
 ipv6 ospf 1 area 0
 clock rate 64000
 bandwidth 64
 no shutdown
!
ipv6 router ospf 1
 router-id 172.16.1.1
end
```

Router R2

```
hostname R2
!
ipv6 unicast-routing
!
interface Loopback0
 ipv6 address FEC0::2:1/112
 ipv6 ospf 1 area 0
!
interface Serial0/0/0
 ipv6 address FEC0::12:2/112
 ipv6 ospf 1 area 0
 bandwidth 64
 no shutdown
!
interface Serial0/0/1
 ipv6 address FEC0::23:2/112
 ipv6 ospf 1 area 0
 clock rate 64000
 bandwidth 64
 no shutdown
!
ipv6 router ospf 1
 router-id 172.16.2.1
end
```

Router R3

```
hostname R3
!
ipv6 unicast-routing
!
interface Loopback0
 ipv6 address FEC0::3:1/112
 ipv6 ospf 1 area 0
!
interface Serial0/0/1
```



```
 ipv6 address FEC0::23:3/112
 ipv6 ospf 1 area 0
 bandwidth 64
 no shutdown
!
interface Serial0/1/0
 ipv6 address FEC0::34:3/112
 ipv6 ospf 1 area 0
 clock rate 64000
 bandwidth 64
 no shutdown
!
ipv6 router ospf 1
 router-id 172.16.3.1
end
```

Router R4

```
hostname R4
!
ipv6 unicast-routing
!
interface Loopback0
 ipv6 address FEC0::4:1/112
 ipv6 ospf 1 area 0
!
interface Serial0/0/0
 ipv6 address FEC0::34:4/112
 ipv6 ospf 100 area 0
 bandwidth 64
 no shutdown
!
ipv6 router ospf 1
 router-id 172.16.4.1
end
```


Router Interface Summary Table

Router Interface Summary				
Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1700	Fast Ethernet 0 (FA0)	Fast Ethernet 1 (FA1)	Serial 0 (S0)	Serial 1 (S1)
1800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2600	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0 (S0/0)	Serial 0/1 (S0/1)
2800	Fast Ethernet 0/0 (FA0/0)	Fast Ethernet 0/1 (FA0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. Rather than list all combinations of configurations for each router class, this table includes identifiers for the possible combinations of Ethernet and serial interfaces in the device. The table does not include any other type of interface, even though a specific router might contain one. For example, for an ISDN BRI interface, the string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.				