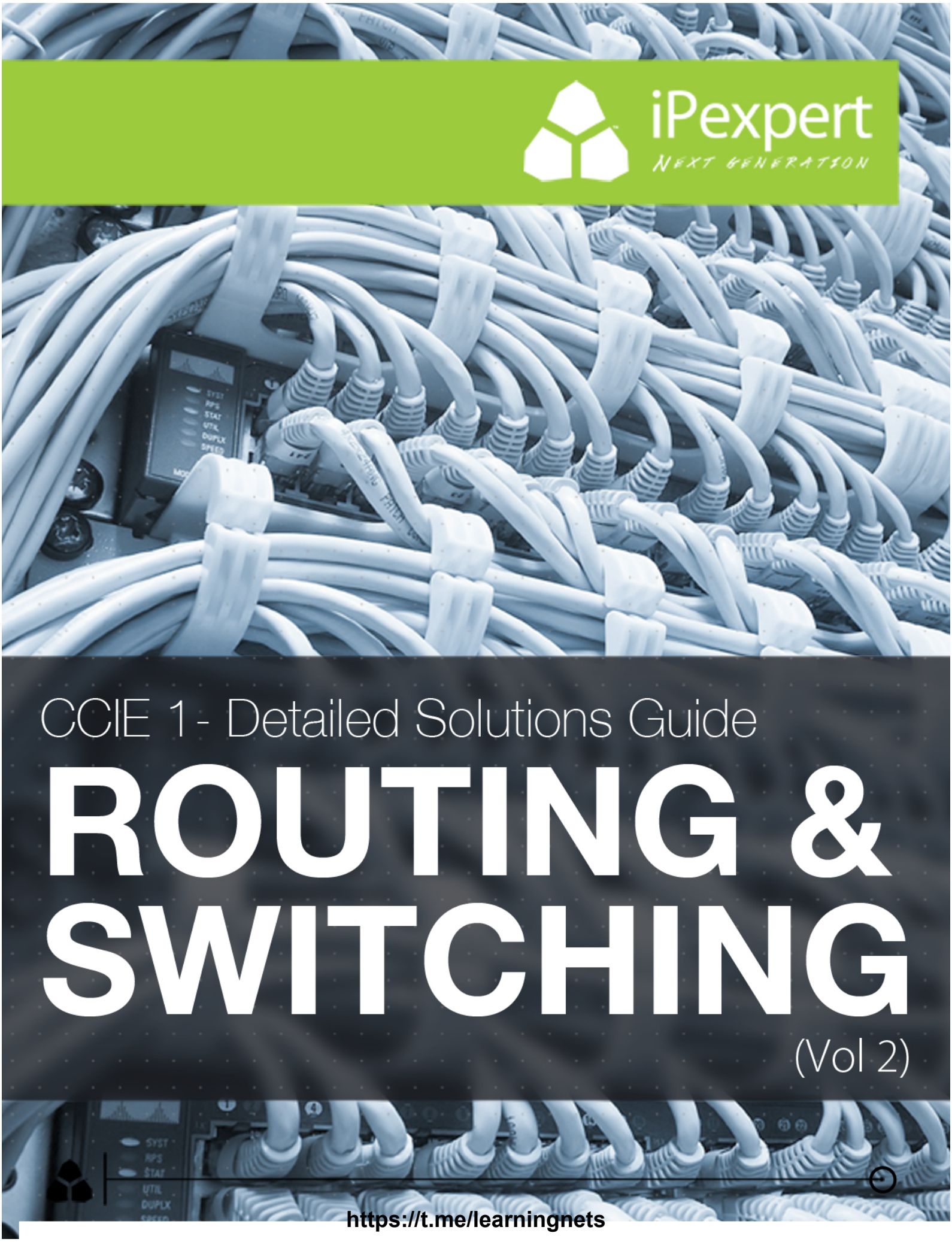

iPexpert's Cisco CCIE R&S (v5) 8-Hour Mock Lab DSG (Vol. 2)

Congratulations! You now possess one of the ULTIMATE CCIE Lab preparation resources available! This resource was produced by senior engineers, technical instructors, and authors boasting decades of internetworking experience. Although there is no way to guarantee a 100% success rate on the CCIE Lab exam, we feel confident that your chances of passing the Lab will improve dramatically after completing this industry-recognized Workbook!

iPexpert is proud to lead the industry with multiple support options at your disposal free of charge. Our online communities have attracted a membership of your peers from around the world! At blog.ipexpert.com you can keep up to date with everything iPexpert does. At community.ipexpert.com, you may collaborate with your CCIE and CCIE-candidate peers.

Feedback

Do you have a suggestion or other feedback regarding this book or other IPexpert products? At iPexpert, we look to you – our valued clients – for the real world, frontline evaluation that we believe is necessary to improve continually. Please send an email with your thoughts to feedback@ipexpert.com or call 1.866.225.8064 (international callers dial +1.810.326.1444).

In addition, when you pass the CCIE Lab exam, we want to hear about it! Email your CCIE number to success@ipexpert.com and let us know how iPexpert helped you succeed. We would like to send you a gift of thanks and congratulations.

Additional CCIE Preparation Material

iPexpert is committed to developing the most effective Cisco CCIE Collaboration, Data Center, Routing & Switching, Security, and Wireless certification preparation tools available. Our team of certified networking professionals develops the most up-to-date and comprehensive materials for networking certification, including self-paced workbooks, online Cisco hardware

rental, classroom training, online (distance learning) instructor-led training, audio products, and video training materials.

We employ an experienced and accomplished team of experts to create, maintain, and constantly update our products. At iPexpert, we are focused on making your CCIE Lab preparation more effective.

A message from the Author

The scenarios covered in this workbook were developed by CCIEs to help you prepare for the Cisco CCIE Lab. It is strongly recommended that you use other reading materials in addition to this workbook.

Training is not this CCIE workbook's objective. The intent of these labs is to test your knowledge and abilities in implementing Cisco Enterprise Security Solutions.

Time management is very important. If you get stuck on a lab scenario be sure to write it down. Formulate a Checklist for skipped sections and then return to those sections once you have gone through the entire lab. Be sure to revisit the questions that you do not understand.

For more information on the CCIE lab, please visit http://www.cisco.com/go/ccie.

Helpful Hints

- Keep It Simple; try to avoid any extra work (example: adding descriptions).
- Always reference everything from the Documentation Website: http://www.cisco.com/web/psa/products/index.html
- Save your router configurations often (wr is the quickest command).

Troubleshooting Lab 1 Detailed Solutions

Incident 1 TFTP Between R20 and R17

Troubleshooting

First, start out by performing the command specified in the incident to verify the issue. Also, make sure the R20 and R17 have basic IP connectivity.

R20

```
R20#copy tftp://172.16.10.2/startup-config null:
Accessing tftp://172.16.10.2/startup-config...
%Error opening tftp://172.16.10.2/startup-config (Timed out)


R20#ping 172.16.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
```

At this point, the issue in the incident has been confirmed and connectivity between R20 and R17 has also been confirmed. This would lead to a few conclusions. First, TFTP may not be enabled on R17. Second, there may be filtering that blocks TFTP somewhere in the path between R17 and R20. Third, the location of the file may be different or there may be a TFTP setting on R17 that is not correctly setup. Let's start by looking at the tftp config on R17.

R17

```
R17#sh run | sec tftp
tftp-server nvram:startup-config 99
```

Notice there is an access-list 99 on the tftp statement. Let's check it out.

R17

```
R17#sh access-list 99

Standard IP access list 99

  10 permit 172.16.70.18
```

Access-list 99 is only permitting 172.16.70.18, which is not even an IP on this network. Modify the config to allow R20 tftp access from IP 172.16.10.18 and test tftp connectivity from R20 again.

R17

```
R17(config)#ip access-list standard 99

R17(config-std-nacl)#no 10

R17(config-std-nacl)#10 permit 172.16.10.18
```

R20

```
R20#copy tftp://172.16.10.2/startup-config null:
Accessing tftp://172.16.10.2/startup-config...
%Error opening tftp://172.16.10.2/startup-config (Timed out)
```

Connectivity still does not work. There may be other filtering applied within the network. Let's start on R20 and work our way up to R17.

R20

```
R20#sh run int e0/1

Current configuration : 70 bytes

!

interface Ethernet0/1

 ip address 172.16.10.18 255.255.255.252
```

SW8

```
SW8#sh run int vlan 188

Current configuration : 83 bytes

!

interface Vlan188

 bandwidth 10000

 ip address 172.16.10.10 255.255.255.252
```



```
end
```

R18

```
R18#sh run int e0/1
Current configuration : 69 bytes
!
interface Ethernet0/1
 ip address 172.16.10.9 255.255.255.252
end

R18#sh run int e1/0
Current configuration : 95 bytes
!
interface Ethernet1/0
 ip address 172.16.10.1 255.255.255.252
 ip access-group TFTP out
```

There is an access-list on the link between R18 and R17. Let's take a look at the ACL.

R18

```
R18#sh access-list TFTP
Extended IP access list TFTP
    10 permit udp host 172.16.10.18 host 172.16.70.2
    20 deny udp any any (42 matches)
    30 permit ip any any (5 matches)
```

Just like on the TFTP access-list on R17, there is a typo in the access-list here on R18. We need to change line 10 to allow access to 172.16.10.2, not 172.16.70.2 and test connectivity once more.

R18

```
R18(config)#ip access-list extended TFTP
```

```
R18(config-ext-nacl)#no 10
R18(config-ext-nacl)#10 permit udp host 172.16.10.18 host 172.16.10.2
```

Troubleshooting Verification

R20

```
R20#copy tftp://172.16.10.2/startup-config null:
Accessing tftp://172.16.10.2/startup-config...
Loading startup-config from 172.16.10.2 (via Ethernet0/1): !
[OK - 2705 bytes]


2705 bytes copied in 0.027 secs (100185 bytes/sec)
```

Configuration Changes Summary

R17

```
R17(config)#ip access-list standard 99
R17(config-std-nacl)#no 10
R17(config-std-nacl)#10 permit 172.16.10.18
```

R18

```
R18(config)#ip access-list extended TFTP
R18(config-ext-nacl)#no 10
R18(config-ext-nacl)#10 permit udp host 172.16.10.18 host 172.16.10.2
```


Incident 2 Acquisition Site Connectivity

Troubleshooting

The incident states that there were changes in the ISP cloud, which should signal to us that we should look their first. If this connectivity was working before the changes, we need to figure out what was changed. First, verify that VLAN 311 on SW3 can not ping any address at HQ on the 172.16.10.x subnets.

SW3

```
SW3#sh ip route 172.16.10.8
% Subnet not in table


SW3#ping 172.16.10.9 source vlan 311
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.9, timeout is 2 seconds:
Packet sent with a source address of 10.10.99.1
.....
Success rate is 0 percent (0/5)
```

SW3 does not have a route to 172.16.10.9. Let's take a look at SW3's next hop, R2, and see if it has a route to any of the HQ subnets.

R2

```
R2# sh ip route 172.16.10.8
% Subnet not in table
```

R2 is not learning the HQ routes through BGP. Again, since the incident specifies that changes were made in the ISP cloud, let's take a look at the BGP configurations there.

ISP1

```
ISP1#sh run | sec bgp
router bgp 31
 bgp log-neighbor-changes
 redistribute connected
 neighbor 13.13.13.5 remote-as 32
```

```
 neighbor 13.13.13.26 remote-as 333
 neighbor 13.13.13.26 route-map FILTER-ROUTES out
```

ISP2

```
ISP2#sh run | sec bgp
router bgp 32
 bgp log-neighbor-changes
 redistribute connected
 neighbor 13.13.13.2 remote-as 33
 neighbor 13.13.13.2 prefix-list FILTER_PROVIDER in
 neighbor 13.13.13.6 remote-as 31
 neighbor 13.13.13.6 prefix-list FILTER_PROVIDER in
 neighbor 13.13.13.10 remote-as 1110
 neighbor 13.13.13.10 route-map FILTER-ROUTES out
```

ISP3

```
ISP3#sh run | sec bgp
router bgp 33
 bgp log-neighbor-changes
 redistribute connected
 neighbor 13.13.13.1 remote-as 32
 neighbor 13.13.13.22 remote-as 1111
 neighbor 13.13.13.22 route-map FILTER-ROUTES out
```

Notice the route-maps attached to the neighbor statements. Let's take a look at the route-map configurations.

ISP1

```
ISP1#sh route-map FILTER-ROUTES
route-map FILTER-ROUTES, permit, sequence 10
  Match clauses:
    ip address prefix-lists: FILTER-PROVIDER
```



```
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes


ISP1#show ip prefix-list FILTER-PROVIDER
ip prefix-list FILTER-PROVIDER: 4 entries
   seq 5 deny 16.16.16.0/24 le 32
   seq 10 deny 14.14.14.0/24 le 32
   seq 15 deny 13.13.13.0/24 le 32
   seq 20 permit 0.0.0.0/0 le 32
```

ISP2

```
ISP2#sh route-map FILTER-ROUTES
route-map FILTER-ROUTES, permit, sequence 10
  Match clauses:
    ip address prefix-lists: FILTER-PROVIDER
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes


ISP2#show ip prefix-list FILTER-PROVIDER
ip prefix-list FILTER-PROVIDER: 5 entries
   seq 1 deny 172.16.10.8/29 le 32
   seq 5 deny 16.16.16.0/24 le 32
   seq 10 deny 14.14.14.0/24 le 32
   seq 15 deny 13.13.13.0/24 le 32
   seq 20 permit 0.0.0.0/0 le 32
```

ISP3

```
ISP3#sh route-map FILTER-ROUTES
route-map FILTER-ROUTES, permit, sequence 10
  Match clauses:
    ip address prefix-lists: FILTER-PROVIDER
```

```
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes


ISP3#show ip prefix-list FILTER-PROVIDER
ip prefix-list FILTER-PROVIDER: 5 entries
   seq 1 deny 10.10.99.0/24 le 32
   seq 5 deny 16.16.16.0/24 le 32
   seq 10 deny 14.14.14.0/24 le 32
   seq 15 deny 13.13.13.0/24 le 32
   seq 20 permit 0.0.0.0/0 le 32
```

Notice the highlighted entries in the prefix-list. There is one HQ subnet route being blocked inbound and outbound to ISP1 and ISP3 from ISP2. Also, the Aquisition routes are being blocked inbound to R18 on ISP3. Let's remove these entries from the prefix-lists.

ISP2

```
ISP2(config)#no ip prefix-list FILTER-PROVIDER seq 1 deny
172.16.10.8/29 le 32
```

ISP3

```
ISP3(config)#no ip prefix-list FILTER-PROVIDER seq 1 deny 10.10.99.0/24
le 32
```

Observe the results. The routers are now learning the routes for end to end connectivity between HQ and the Acquisition site.

SW3

```
SW3#sh ip route 172.16.10.0
Routing entry for 172.16.10.0/30
  Known via "eigrp 100", distance 170, metric 25602816
  Tag 32, type external
  Redistributing via eigrp 200, eigrp 100
  Advertised by eigrp 200 metric 100 10 1 1 1
  Last update from 172.16.30.25 on Vlan23, 00:00:37 ago
  Routing Descriptor Blocks:
```



```
  * 172.16.30.25, from 172.16.30.25, 00:00:37 ago, via Vlan23
      Route metric is 25602816, traffic share count is 1
      Total delay is 110 microseconds, minimum bandwidth is 100 Kbit
      Reliability 1/255, minimum MTU 1 bytes
      Loading 1/255, Hops 1
      Route tag 32
```

```
SW3#ping 172.16.10.9 source vlan 311
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.9, timeout is 2 seconds:
Packet sent with a source address of 10.10.99.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 22/25/28 ms
```

R2

```
R2#show ip route 172.16.10.8
Routing entry for 172.16.10.8/30
  Known via "bgp 1110", distance 20, metric 0
  Tag 32, type external
  Redistributing via eigrp 100
  Advertised by eigrp 100 metric 100 10 1 1 1
  Last update from 13.13.13.9 00:02:26 ago
  Routing Descriptor Blocks:
  * 13.13.13.9, from 13.13.13.9, 00:02:26 ago
      Route metric is 0, traffic share count is 1
      AS Hops 3
      Route tag 32
      MPLS label: none
```

R18

```
R18#show ip route 10.10.99.0
```

```
Routing entry for 10.10.99.0/30
  Known via "bgp 1111", distance 20, metric 0
  Tag 33, type external
  Redistributing via eigrp 200
  Advertised by eigrp 200 metric 100 10 1 1 1
  Last update from 13.13.13.21 00:02:48 ago
  Routing Descriptor Blocks:
  * 13.13.13.21, from 13.13.13.21, 00:02:48 ago
      Route metric is 0, traffic share count is 1
      AS Hops 3
      Route tag 33
      MPLS label: none
```

Troubleshooting Verification

SW3 is now able to ping subnets at HQ. Let's verify this by running a TCL script on SW3.

SW3

```
SW3#tclsh
SW3(tcl)#foreach address {
+>172.16.10.1
+>172.16.10.2
+>172.16.10.5
+>172.16.10.6
+>172.16.10.9
+>172.16.10.10
+>172.16.10.13
+>172.16.10.14
+>172.16.10.17
+>172.16.10.18
+>} { ping $address source vlan 311
```



```
+>}
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.1, timeout is 2 seconds:
Packet sent with a source address of 10.10.99.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 25/25/26 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
Packet sent with a source address of 10.10.99.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 25/25/27 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.5, timeout is 2 seconds:
Packet sent with a source address of 10.10.99.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 23/25/26 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.6, timeout is 2 seconds:
Packet sent with a source address of 10.10.99.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 25/25/26 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.9, timeout is 2 seconds:
Packet sent with a source address of 10.10.99.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 25/25/26 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.10, timeout is 2 seconds:
Packet sent with a source address of 10.10.99.1
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 25/25/26 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.10.13, timeout is 2 seconds:

Packet sent with a source address of 10.10.99.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 24/25/26 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.10.14, timeout is 2 seconds:

Packet sent with a source address of 10.10.99.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 25/25/26 ms
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.10.17, timeout is 2 seconds:

Packet sent with a source address of 10.10.99.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 23/24/26 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.10.18, timeout is 2 seconds:

Packet sent with a source address of 10.10.99.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 25/25/26 ms
```

Configuration Changes Summary

ISP2

```
ISP2(config)#no ip prefix-list FILTER-PROVIDER seq 1 deny
172.16.10.8/29 le 32
```

ISP3

```
ISP3(config)#no ip prefix-list FILTER-PROVIDER seq 1 deny 10.10.99.0/24
le 32
```


Incident 3 BB2 Connectivity

Troubleshooting

First, verify that SW2 cannot ping 192.168.100.222 on BB2.

SW2

```
SW2#ping 172.16.40.9

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.40.9, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/5 ms


SW2#ping 192.168.100.222

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.100.222, timeout is 2
seconds:

.....

Success rate is 0 percent (0/5)
```

With the above ping commands, we have established that SW2 has reachability to the directly connected interface of BB2, but not the loopback interface behind it. Looking at the incident details and the diagram, the domain is using RIPv2 as the routing protocol. Since we cannot access BB2, lets take a look at the RIP config of SW2.

SW2

```
SW2#sh run | sec rip

router rip

 version 1

 network 172.16.0.0

 offset-list 1 in 15

 no auto-summary
```

There are few things to notice in this config. First, RIP is using version 1. Also notice that the there is an offset-list inbound that sets the metric to 15. Remember that RIP uses hop count as

its metric and has a maximum hop count of 15 before it expires. This offset-list is immediately setting the route to the maximum metric. We need to change this to something less than 15. Since there is no rule or guideline specifying what the distance should be in this case, we will change it to 1.

SW2

```
SW2(config)#router rip
SW2(config-router)#version 2
SW2(config-router)#offset-list 1 in 1


SW2(config)access-list 1 permit 192.168.100.0 0.0.0.255
```

Lets now verify that you are learning routes from BB2 and verify full reachability to 192.168.100.222 from all devices in the RIPv2 routing domain.

SW2

```
SW2#sh ip route
R        172.16.112.0/24 [120/2] via 172.16.40.9, 00:00:20, Vlan222
      192.168.100.0/32 is subnetted, 1 subnets
R        192.168.100.222 [120/2] via 172.16.40.9, 00:00:20, Vlan22
```

Troubleshooting Verification

SW2

```
SW2#ping 192.168.100.222
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.222,timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

R6

```
R6#ping 192.168.100.222
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.222,timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

As you can see from the ping results above, 192.168.100.222 no has full reachability within the RIP routing domain.

Configuration Changes Summary

SW2

```
SW2(config)#router rip
SW2(config-router)#version 2
```



```
SW2(config-router)#offeset-list 1 in 1
```

Incident 4 IP Precedence Marking

Troubleshooting

Let's first look at the output from the specified command in the incident to determine where to focus our efforts. Also remember that this incident is dependent on Incident 3.

SW2

```
SW2#sh interface vlan 620 precedence
Vlan620
 Output
       Precedence 6:  1385 packets, 116082 bytes
```

We can see from the config that IPP 6 is being indicated, but IPP 5 is not. Also notice that the direction of the marking is Output, not Input. Let's look at the configuration of VLAN 620 on SW2

SW2

```
SW2#interface Vlan620
 ip address 172.16.40.6 255.255.255.252
 ip accounting precedence output
```

The output from this command explains why SW2 is not showing the input IPP accounting. Since the packets are being marked coming from the Denver Local Office, the accounting on VLAN 620 should be set for input. Add ip accounting for IPP for input on VLAN 620.

SW2

```
SW2(config)#int vlan 620
SW2(config-if)#ip accounting precedence input
```

Confirm you are accounting for IPP inbound on VLAN 620

SW2

```
SW2#sh interface vlan 620 precedence
Vlan620
  Input
    Precedence 6:  8 packets, 884 bytes
```



```
  Output

    Precedence 6:  1516 packets, 127080 bytes
```

SW2 is now accounting for marked IPP packets inbound on VLAN 620, however, they are coming in marked as IPP 6, not 5. Let's find where the marking is being performed and verify that the packets are being marked correctly. We will start with R6.

R6

```
R6#sh run int e0/0.620
interface Ethernet0/0.620
 encapsulation dot1Q 620
 ip address 172.16.40.5 255.255.255.252
 service-policy input IPP
end

R6#sh policy-map IPP
  Policy Map IPP
    Class DENVERtoBB2_IPP
      set ip precedence 4

R6#show class-map DENVERtoBB2_IPP
 Class Map match-all DENVERtoBB2_IPP (id 1)
   Match access-group name DENVERtoBB2

R6#show access-list DENVERtoBB2
Extended IP access list DENVERtoBB2
    10 permit ip 172.16.50.0 0.0.0.255 host 192.168.100.222
```

On R6, the policy-map is set to mark packets destined to BB2 as IPP4, not IPP5. Also, the service-policy is applied in the wrong direction under int e0/0.620. Let's change the service-policy to output and set the IPP to 5 under the policy-map and observe the results.

R6

```
R6(config)#policy-map IPP

R6(config-pmap)#class DENVERtoBB2_IPP

R6(config-pmap-c)#no set ip precedence 4

R6(config-pmap-c)#set ip precedence 5



R6(config)#int e0/0.620

R6(config-subif)#no service-policy input IPP

R6(config-subif)#service-policy output IPP
```

Troubleshooting Verification

SW2

```
SW2#sh interface vlan 620 precedence
Vlan620
  Input
    Precedence 0:  5 packets, 570 bytes
    Precedence 5:  5 packets, 570 bytes
    Precedence 6:  106 packets, 19716 bytes
  Output
    Precedence 0:  5 packets, 570 bytes
    Precedence 5:  5 packets, 570 bytes
    Precedence 6:  182 packets, 20512 bytes
```

Configuration Changes Summary

SW2

```
SW2(config)#int vlan 620

SW2(config-if)#ip accounting precedence input
```

R6

```
R6(config)#policy-map IPP

R6(config-pmap)#class DENVERtoBB2_IPP

R6(config-pmap-c)#no set ip precedence 4

R6(config-pmap-c)#set ip precedence 5
```



```
R6(config)#int e0/0.620
R6(config-subif)#no service-policy input IPP
R6(config-subif)#service-policy output IPP
```

Incident 6 Voice Users at Regional Office 2

Troubleshooting

Let's first look at the output from the specified command in the incident to determine where to focus our efforts. Let's also make sure we have same subnet reachability between R16 and R17. Remember that this incident is dependent on Incident 3.

R17

```
R17#ping 172.16.16.16

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.16.16, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)


R17#ping 172.16.70.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.70.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Taking a look at the logs on both devices uncovers our first issue, found on R16. We take a peek at the config for OSPF for each device now as well.

R16

```
R16#sh log

%OSPF-4-ERRRCV: Received invalid packet: mismatched area ID from
backbone area from 172.16.70.1, Ethernet0/0

%OSPF-4-ERRRCV: Received invalid packet: mismatched area ID from
backbone area from 172.16.70.1, Ethernet0/0


R16#sh run | sec ospf

router ospf 1

 network 10.10.0.0 0.0.255.255 area 0
```



```
 network 172.16.70.2 0.0.0.0 area 1
 network 172.16.0.0 0.0.255.255 area 0
```

R17

```
R17#sh run | sec ospf
router ospf 1
 redistribute eigrp 200 subnets
 network 10.10.0.0 0.0.255.255 area 0
 network 172.16.70.1 0.0.0.0 area 0
 network 172.16.0.0 0.0.255.255 area 0
```

There is a mis-configuration of OSPF. Either R16 or R17 is configured with the wrong Area. Let's fix this issue by changing the network command on R16. The diagram shows that this connection should be in OSPF Area 0. Since there is already a network statement for 172.16.0.0/16 for Area 0, we just need to remove the network statement for area 1.

R16

```
R16(config)#router ospf 1
R16(config-router)#no network 172.16.70.2 0.0.0.0 area 1
%OSPF-6-AREACHG: 172.16.70.2/32 changed from area 1 to area 0
```

On R16, you will notice that the adjacency builds but then tears back down after the dead timer expires.

R16

```
R16(config-router)#
%OSPF-5-ADJCHG: Process 1, Nbr 172.16.17.17 on Ethernet0/0 from LOADING
to FULL, Loading Done
R16(config-router)#
%OSPF-5-ADJCHG: Process 1, Nbr 172.16.17.17 on Ethernet0/0 from FULL to
DOWN, Neighbor Down: Dead timer expired
```

The output above is indicative of a timer mismatch or an OSPF network-type mismatch, so let's look at the interface configurations to verify that the OSPF network types match.

R16

```
R16#sh run int e0/0
Current configuration : 86 bytes
!
interface Ethernet0/0
 ip dhcp client client-id ascii R16dhcp
 ip address dhcp
```

R17

```
R17#sh run int e0/0
Current configuration : 101 bytes
!
interface Ethernet0/0
 ip address 172.16.70.1 255.255.255.252
 ip ospf network point-to-point
```

On R17, the OSPF network type is set to point-to-point. R16 is using the default network type for Ethernet, which is broadcast. Point-to-Point does not elect a DR/BDR while Broadcast does. We need these settings to match on each side. When presented with a choice, it is always a good idea to add commands rather than to remove them. So let's set the network type on E0/0 of R16 to point-to-point.

R16

```
R16(config)#int e0/0
R16(config-if)#ip ospf network point-to-point
%OSPF-5-ADJCHG: Process 1, Nbr 172.16.17.17 on Ethernet0/0 from LOADING
to FULL, Loading Done
```

Troubleshooting Verification

R17

```
R17#ping 172.16.16.16
```



```
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.16.16, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Configuration Changes Summary

R16

```
R16(config)#router ospf 1

R16(config-router)#no network 172.16.70.2 0.0.0.0 area 1


R16(config)#int e0/0

R16(config-if)#ip ospf network point-to-point
```

Incident 6 Call Center 2 lost Connectivity to Network

Troubleshooting

The first thing to notice is that the serial connection between R6 and R3 is up/down. You can glean this information by performing a "show ip int brief" command on either device.

R3

```
R3#sh ip int bri
Interface              IP-Address      OK? Method Status                Protocol
Serial4/2              172.16.60.29    YES manual up                    down
```

By looking at the configurations on the serial interfaces of R6 and R3, you can see that the link is using PPP encapsulation with CHAP authentication. You can also notice that the "ppp chap refuse" command is configured, which blocks all chap requests from the adjacent link.

R6

```
R6#sh run int s4/0
Current configuration : 175 bytes
!
interface Serial4/0
 ip address 172.16.60.30 255.255.255.252
 encapsulation ppp
 ppp authentication chap
 ppp chap hostname R3
 ppp chap refuse
 serial restart-delay 0
```

R3

```
R3# sh run int s4/2
Current configuration : 136 bytes
!
interface Serial4/2
 ip address 172.16.60.29 255.255.255.252
```



```
 encapsulation ppp
 ppp authentication chap
 serial restart-delay 0
```

The first step to resolving this incident would be removing the ppp chap refuse command from R6 and observe the results:

R6

```
R6(config)#interface s4/0
R6(config-if)#no ppp chap refuse


R6#sh ip int bri
Interface          IP-Address      OK? Method Status           Protocol
Serial4/0          172.16.60.30    YES manual up                   up
```

Notice however, that even though the link shows up, pinging across the link still does not work.

R6

```
R6#ping 172.16.60.29
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.60.29, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Remember that CHAP was being used for authentication. Verify the username and password for the connection on each Router.

R3

```
R3#sh run | sec username
username R6 password 0 IPExpertPPP
```

R6

```
R6#sh run | sec username
username R3 password 0 1PExpertPPP
```

Notice that the passwords do not match. You can fix this issue by changing the password to match on both sides. Also, the CHAP username is set to R3 on R6. By default, the hostname is used for the CHAP username. This can be done on either router. Since the username needs to be removed from R6 as well, we will apply the change to R6 to match R3.

R6

```
R6(config)#username R3 password IPExpertPPP
R6(config)#interface S4/0
R6(config-if)#no ppp chap hostname R3
```

Troubleshooting Verification

Verify that connectivity is up between the serial interfaces and that the OSPF neighbor relationship forms. Also, ping from an interface inside of Call Center 2 to a device inside of Call Center 3 to verify connectivity outside of the local domain to satisfy the requirements of the Incident.

R6

```
R6#ping 172.16.60.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.60.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/17/18 ms


R6#
R6#sh ip ospf nei
Neighbor ID     Pri   State           Dead Time   Address
Interface
172.16.3.3        0   FULL/  -        00:00:33    172.16.60.29
Serial4/0


R6#ping 192.168.100.1 source e0/0.620
Type escape sequence to abort.
```



```
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.40.5
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/8/9 ms
```

Configuration Changes Summary

R6

```
R6(config)#interface s4/0
R6(config-if)#no ppp chap refuse
R6(config-if)#no ppp chap hostname R3

R6(config)#username R3 password IPExpertPPP
```

Incident 7 Call Center 1 Connectivity

Troubleshooting

First thing to notice is the incident refers to the BGP diagram. This gives us a good starting place to start troubleshooting. Lets observe the results of the required successful pings first and then take a look at the BGP config of both R24 and R25.

R24

```
R24#ping 192.168.100.25 source loo 100
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.25, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.24
.....
Success rate is 0 percent (0/5)


R24#sh run | sec bgp
router bgp 2424
 bgp log-neighbor-changes
 redistribute connected
 neighbor 1.1.1.22 remote-as 666
 neighbor 16.16.16.5 remote-as 66
```

R25

```
R25#ping 192.168.100.24 source loo 100
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.24, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.25
.....
Success rate is 0 percent (0/5)


R25#sh run | sec bgp
```



```
router bgp 2525
 bgp log-neighbor-changes
 redistribute connected route-map CONNECTED
 neighbor 1.1.1.24 remote-as 666
 neighbor 16.16.16.9 remote-as 66


R25#show route-map CONNECTED
route-map CONNECTED, permit, sequence 10
  Match clauses:
    ip address (access-lists): 40
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes


R25#sh access-list 40
Standard IP access list 40
    10 permit 192.168.100.24
```

We have now confirmed that the loopbacks cannot ping each other. The only difference between the configs is the route-map attached to the redistribute connected command on R25. As you can see from the config, it is incorrectly configured with the IP address of R24's loopback address, instead of it's own. Lets change the ACL to reflect R25's loopback instead of R24's.

R25

```
R25(config)#no access-list 40
R25(config)#access-list 40 permit host 192.168.100.25
```

Troubleshooting Verification

R24

```
R24#ping 192.168.100.25 source loopback 100
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.168.100.25, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.24
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/16/17 ms
```

R25

```
R25#ping 192.168.100.25 source loopback 100
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.25, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.25
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/5 ms
```

Configuration Changes Summary

R25

```
R25(config)#no access-list 40
R25(config)#access-list 40 permit host 192.168.100.25
```


Incident 8 HQ Route Preference

Troubleshooting

Start by verifying which path R20 takes to reach R17. According the incident, it is currently is taking the path through SW7, which is not optimal. Also, the incident clearly states that all links between R18, SW7, and SW8 should have a bandwidth of 10Mbps. Then, we need to verify each link specified is set to 10Mbps, since it is specifically called out in the incident. Also, since this routing domain is EIGRP, bandwidth is the key factor in deciding which path to use.

R20

```
R20#ping 172.16.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/5 ms


R20#trace 172.16.10.2
Type escape sequence to abort.
Tracing the route to 172.16.10.2
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.10.17 0 msec 0 msec 1 msec
  2 172.16.10.14 0 msec 0 msec 0 msec
  3 172.16.10.5 1 msec 0 msec 0 msec
  4 172.16.10.5 !A  *  !A
```

SW8

```
SW8#sh ip route 172.16.10.2
Routing entry for 172.16.10.0/30
  Known via "eigrp 200", distance 90, metric 282112, type internal
  Redistributing via eigrp 200
  Last update from 172.16.10.14 on Vlan78, 00:04:39 ago
```

```
  Routing Descriptor Blocks:
  * 172.16.10.14, from 172.16.10.14, 00:04:39 ago, via Vlan78
      Route metric is 282112, traffic share count is 1
      Total delay is 1020 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
```

R17

```
R17#ping 172.16.10.18
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.10.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms


R17#traceroute 172.16.10.18
Type escape sequence to abort.
Tracing the route to 172.16.10.18
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.10.1 0 msec 1 msec 0 msec
  2 172.16.10.6 0 msec 0 msec 0 msec
  3 172.16.10.13 0 msec 1 msec 0 msec
  4 172.16.10.18 0 msec *  0 msec
```

There is definite connectivity from R20 back to R16 and we have confirmed that the path indeed uses SW7 as the transport. It's time to verify the bandwidth statements on each link so that they are all 10MB as indicated in the incident.

R18

```
R18#sh run int e0/0
interface Ethernet0/0
 bandwidth 100000
 ip address 172.16.10.5 255.255.255.252
```



```
R18#sh run int e0/1
interface Ethernet0/1
 ip address 172.16.10.9 255.255.255.252
```

SW7

```
SW7#sh run int vlan 78
interface Vlan78
 bandwidth 100000
 ip address 172.16.10.14 255.255.255.252

SW7#sh run int vlan 187
interface Vlan187
 bandwidth 100000
 ip address 172.16.10.6 255.255.255.252
```

SW8

```
SW8#sh run int vlan 78
interface Vlan78
 bandwidth 100000
 ip address 172.16.10.13 255.255.255.252

SW8#sh run int vlan 188
interface Vlan188
 bandwidth 10
 ip address 172.16.10.10 255.255.255.252
```

All interfaces listed above should be set to 10Mb. Observe the results after configuring each of these interfaces for 10Mb bandwidth.

R18

```
R18#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
R18(config)#int e0/0
R18(config-if)#bandwidth 10000
R18(config-if)#int e0/1
R18(config-if)#bandwidth 10000
```

SW7

```
SW7# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SW7(config)#int vlan 78
SW7(config-if)#bandwidth 10000
SW7(config-if)#int vlan 187
SW7(config-if)#bandwidth 10000
```

Let's observe the results:

SW8

```
SW8#sh ip eigrp topology 172.16.10.0 255.255.255.252
EIGRP-IPv4 Topology Entry for AS(200)/ID(172.16.108.108) for
172.16.10.0/30
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is
281856
  Descriptor Blocks:
  172.16.10.9 (Vlan188), from 172.16.10.9, Send flag is 0x0
      Composite metric is (281856/281600), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 1010 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
  172.16.10.14 (Vlan78), from 172.16.10.14, Send flag is 0x0
```



```
        Composite metric is (282112/281856), route is Internal
        Vector metric:
          Minimum bandwidth is 10000 Kbit
          Total delay is 1020 microseconds
          Reliability is 255/255
          Load is 1/255
          Minimum MTU is 1500
          Hop count is 2


SW8#sh ip route 172.16.10.2
Routing entry for 172.16.10.0/30
  Known via "eigrp 200", distance 90, metric 281856, type internal
  Redistributing via eigrp 200
  Last update from 172.16.10.9 on Vlan188, 00:02:57 ago
  Routing Descriptor Blocks:
  * 172.16.10.9, from 172.16.10.9, 00:02:57 ago, via Vlan188
      Route metric is 281856, traffic share count is 1
      Total delay is 1010 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

R18

```
R18#sh ip eigrp topology 172.16.10.16 255.255.255.252
EIGRP-IPv4 Topology Entry for AS(200)/ID(172.16.18.18) for
172.16.10.16/30
  State is Passive, Query origin flag is 1, 2 Successor(s), FD is 51712
  Descriptor Blocks:
  172.16.10.10 (Ethernet0/1), from 172.16.10.10, Send flag is 0x0
      Composite metric is (281856/2816), route is Internal
      Vector metric:
```

```
            Minimum bandwidth is 10000 Kbit
            Total delay is 1010 microseconds
            Reliability is 255/255
            Load is 1/255
            Minimum MTU is 1500
            Hop count is 1
      172.16.10.6 (Ethernet0/0), from 172.16.10.6, Send flag is 0x0
          Composite metric is (282112/256512), route is Internal
          Vector metric:
            Minimum bandwidth is 10000 Kbit
            Total delay is 1020 microseconds
            Reliability is 255/255
            Load is 1/255
            Minimum MTU is 1500
            Hop count is 2


R18#sh ip route 172.16.10.18
Routing entry for 172.16.10.16/30
  Known via "eigrp 200", distance 90, metric 281856, type internal
  Redistributing via eigrp 200, bgp 1111
  Advertised by bgp 1111
  Last update from 172.16.10.6 on Ethernet0/0, 00:05:13 ago
  Routing Descriptor Blocks:
  * 172.16.10.10, from 172.16.10.10, 00:05:13 ago, via Ethernet0/1
      Route metric is 281856, traffic share count is 1
      Total delay is 1010 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
    172.16.10.6, from 172.16.10.6, 00:05:13 ago, via Ethernet0/0
```



```
        Route metric is 282112, traffic share count is 1
        Total delay is 1020 microseconds, minimum bandwidth is 10000 Kbit
        Reliability 255/255, minimum MTU 1500 bytes
        Loading 1/255, Hops 2
```

Now the path from R20 to R17 is taking the correct path, but the return path through R18 is load balancing across both of its paths to SW8. EIGRP performs equal cost load balancing by default, but as you can see from the output of the "show eigrp topology", the paths do not have the same metric. This means that R18 is performing Un-equal cost load balancing by using variance. Instead of removing the variance command under EIGRP on R18, let's change it to variance of 1 and observe the results. Since variance is a multiplier for the EIGRP metric, changing it to 1 has no affect (anything times 1 is its original value).

R18

```
R18#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R18(config)#router eigrp 200
R18(config-router)#variance 1

R18#sh ip route 172.16.10.18
Routing entry for 172.16.10.16/30
  Known via "eigrp 200", distance 90, metric 281856, type internal
  Redistributing via eigrp 200, bgp 1111
  Advertised by bgp 1111
  Last update from 172.16.10.10 on Ethernet0/1, 00:01:02 ago
  Routing Descriptor Blocks:
  * 172.16.10.10, from 172.16.10.10, 00:01:02 ago, via Ethernet0/1
      Route metric is 281856, traffic share count is 1
      Total delay is 1010 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

```
R18#sh ip eigrp topology 172.16.10.16 255.255.255.252

EIGRP-IPv4 Topology Entry for AS(200)/ID(172.16.18.18) for
172.16.10.16/30

  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 51712

  Descriptor Blocks:

  172.16.10.10 (Ethernet0/1), from 172.16.10.10, Send flag is 0x0

      Composite metric is (281856/2816), route is Internal

      Vector metric:

        Minimum bandwidth is 10000 Kbit

        Total delay is 1010 microseconds

        Reliability is 255/255

        Load is 1/255

        Minimum MTU is 1500

        Hop count is 1

  172.16.10.6 (Ethernet0/0), from 172.16.10.6, Send flag is 0x0

      Composite metric is (282112/256512), route is Internal

      Vector metric:

        Minimum bandwidth is 10000 Kbit

        Total delay is 1020 microseconds

        Reliability is 255/255

        Load is 1/255

        Minimum MTU is 1500

        Hop count is 2
```

Now both R18 and SW8 are both showing the correct path in the routing table, neither are load balancing, and 2 paths exist for each device in the EIGRP topology table. This fulfills the requirements of the incident.

Troubleshooting Verification

To verify we fulfilled all of the requirements, we take a look at the routing tables and EIGRP topology tables of SW8 and R18. The task requires that traffic sourced from R20 destined for Regional Office 2, take the direct path between SW8 and R18. It also states that un-equal cost

load balancing should not be used. Then, we will do a traceroute from R20 to R17, and then from R17 to R20 to verify the packets are taking the correct paths.

SW8

```
SW8#sh ip route 172.16.10.2
Routing entry for 172.16.10.0/30
  Known via "eigrp 200", distance 90, metric 281856, type internal
  Redistributing via eigrp 200
  Last update from 172.16.10.9 on Vlan188, 00:02:57 ago
  Routing Descriptor Blocks:
  * 172.16.10.9, from 172.16.10.9, 00:02:57 ago, via Vlan188
      Route metric is 281856, traffic share count is 1
      Total delay is 1010 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1


SW8#sh ip eigrp topology 172.16.10.0 255.255.255.252
EIGRP-IPv4 Topology Entry for AS(200)/ID(172.16.108.108) for
172.16.10.0/30
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is
281856
  Descriptor Blocks:
  172.16.10.9 (Vlan188), from 172.16.10.9, Send flag is 0x0
      Composite metric is (281856/281600), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 1010 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

```
  172.16.10.14 (Vlan78), from 172.16.10.14, Send flag is 0x0
      Composite metric is (282112/281856), route is Internal
      Vector metric:
        Minimum bandwidth is 10000 Kbit
        Total delay is 1020 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 2
```

R18

```
R18#show ip route 172.16.10.16
Routing entry for 172.16.10.16/30
  Known via "eigrp 200", distance 90, metric 281856, type internal
  Redistributing via eigrp 200, bgp 1111
  Advertised by bgp 1111
  Last update from 172.16.10.10 on Ethernet0/1, 00:10:05 ago
  Routing Descriptor Blocks:
  * 172.16.10.10, from 172.16.10.10, 00:10:05 ago, via Ethernet0/1
      Route metric is 281856, traffic share count is 1
      Total delay is 1010 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1


R18#sh ip eigrp topology 172.16.10.16 255.255.255.252
EIGRP-IPv4 Topology Entry for AS(200)/ID(172.16.18.18) for
172.16.10.16/30
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 51712
  Descriptor Blocks:
  172.16.10.10 (Ethernet0/1), from 172.16.10.10, Send flag is 0x0
```



```
        Composite metric is (281856/2816), route is Internal
        Vector metric:
          Minimum bandwidth is 10000 Kbit
          Total delay is 1010 microseconds
          Reliability is 255/255
          Load is 1/255
          Minimum MTU is 1500
          Hop count is 1
    172.16.10.6 (Ethernet0/0), from 172.16.10.6, Send flag is 0x0
        Composite metric is (282112/256512), route is Internal
        Vector metric:
          Minimum bandwidth is 10000 Kbit
          Total delay is 1020 microseconds
          Reliability is 255/255
          Load is 1/255
          Minimum MTU is 1500
          Hop count is 2
```

R20

```
R20#trace 172.16.10.2
Type escape sequence to abort.
Tracing the route to 172.16.10.2
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.10.17 0 msec 0 msec 0 msec
  2 172.16.10.9 1 msec 0 msec 0 msec
  3 172.16.10.9 !A  *  !A
```

R17

```
R17#traceroute 172.16.10.18
Type escape sequence to abort.
Tracing the route to 172.16.10.18
```

```
VRF info: (vrf in name/id, vrf out name/id)
  1 172.16.10.1 1 msec 0 msec 0 msec
2 172.16.10.10 0 msec 0 msec 0 msec
3 172.16.10.18 1 msec *  0 msec
```

Configuration Change Summary

R18

```
R18(config)#router eigrp 200
R18(config-router)#variance 1
R18(config)#int e0/0
R18(config-if)#bandwidth 10000
R18(config-if)#int e0/1
R18(config-if)#bandwidth 10000
```

SW7

```
SW7# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
SW7(config)#int vlan 78
SW7(config-if)#bandwidth 10000
SW7(config-if)#int vlan 187
SW7(config-if)#bandwidth 10000
```


Incident 9 Multicast at Denver Local Office

Troubleshooting

The first step here is to verify the fault outlined in the Incident. The incident states that Multicast applications are not working properly and that R9 cannot ping all members of the 226.8.8.8 multicast group. The multicast group members are R7, R8, SW4.

R9

```
R9#ping 226.8.8.8
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 226.8.8.8, timeout is 2 seconds:


Reply to request 0 from 172.16.50.2, 11 ms
```

R9 only receives a response from SW4. Lets start at the beginning and work our way to R7. First we need to verify that Multicast PIM is turned on the link between SW4 and R7. Then, we will need to verify that R7 has joined the IGMP group.

SW4

```
SW4#sh run int vlan 740
!
interface Vlan740
 ip address 172.16.50.9 255.255.255.252
 ip pim sparse-dense-mode
```

R7

```
R7#sh run int e0/1.4
!
interface Ethernet0/1.4
 encapsulation dot1Q 740
 ip address 172.16.50.10 255.255.255.252
 ip pim sparse-dense-mode
 ip igmp join-group 226.8.8.8
```

```
R7#sh ip igmp membership
 Channel/Group          Reporter        Uptime    Exp.  Flags  Interface
 *,226.8.8.8            172.16.50.10    00:05:08 02:10 2LA    Et0/1.4
 *,224.0.1.40           172.16.50.10    00:05:08 02:09 2LA    Et0/1.4

R7#sh run | i igmp
ip igmp immediate-leave group-list 1
```

PIM is enabled between SW4 and R7 and R7 has the configuration to join the 226.8.8.8 group. However, notice that there is an igmp filter applied which calls upon access-list 1. There is not an access-list 1 configured indicating an implicit "deny all". We need to create an access-list 1 and add the 226.8.8.8 igmp group as a permit statement.

R7

```
R7(config)#access-list 1 permit host 226.8.8.8
```

R9

```
R9#ping 226.8.8.8
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 226.8.8.8, timeout is 2 seconds:

Reply to request 0 from 172.16.50.2, 11 ms
Reply to request 0 from 172.16.50.10, 37 ms
```

We are now getting responses from SW4 and R7 but still not R8. Notice that R8 has 2 paths to get to SW4 and R9. Keep in mind that multicast uses a built in "reverse path forwarding" check to verify the path back to the multicast receiver matches the routes in the routing table. Let's verify the multicast route to R9 as well as the unicast route to R9 and make sure they match.

R8

```
R8#show ip route 172.16.50.1
Routing entry for 172.16.50.0/30
  Known via "eigrp 101", distance 90, metric 281856, type internal
```



```
  Redistributing via eigrp 101
  Last update from 172.16.50.5 on Ethernet0/0.4, 00:30:10 ago
  Routing Descriptor Blocks:
  * 172.16.50.5, from 172.16.50.5, 00:30:10 ago, via Ethernet0/0.4
      Route metric is 281856, traffic share count is 1
      Total delay is 1010 microseconds, minimum bandwidth is 10000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1

R8#show ip mroute
(*, 226.8.8.8), 20:53:54/00:02:13, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/0.8, Forward/Sparse-Dense, 00:31:03/stopped
    Ethernet0/0.4, Forward/Sparse-Dense, 20:53:54/stopped

(*, 224.0.1.40), 20:53:53/00:02:19, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/0.8, Forward/Sparse-Dense, 00:31:03/stopped
    Ethernet0/0.4, Forward/Sparse-Dense, 20:53:53/stopped

R8#sh run int e0/0.4
!
interface Ethernet0/0.4
 encapsulation dot1Q 840
 ip address 172.16.50.6 255.255.255.252
 ip pim sparse-dense-mode
 ip igmp join-group 226.8.8.8
```

```
end


R8#sh run int e0/0.8
!
interface Ethernet0/0.8
 encapsulation dot1Q 78
 ip address 172.16.50.14 255.255.255.252
 ip pim sparse-dense-mode
end


R8#sh run | sec mroute
ip mroute 0.0.0.0 0.0.0.0 172.16.50.5
```

R8 is preferring the directly connected link to SW4, however, on PIM is not enabled on SW4 for this link. Also notice that there is a static multicast route on R8 that forces all multicast traffic across this same link. Lets change the route to prefer the link through R7, instead of SW4 since PIM is enabled on this path all the way back to R9.

R8

```
R8(config)#no ip mroute 0.0.0.0 0.0.0.0 172.16.50.5
R8(config)#ip mroute 0.0.0.0 0.0.0.0 172.16.50.13
```

Troubleshooting Verification

R9

```
R9#ping 226.8.8.8
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 226.8.8.8, timeout is 2 seconds:

Reply to request 0 from 172.16.50.2, 10 ms
Reply to request 0 from 172.16.50.6, 30 ms
Reply to request 0 from 172.16.50.10, 20 ms
```

Configuration Changes Summary

R8


```
R8(config)#no ip mroute 0.0.0.0 0.0.0.0 172.16.50.5
R8(config)#ip mroute 0.0.0.0 0.0.0.0 172.16.50.13
```

Incident 10 Issues at Remote Offices

Troubleshooting

Since the incident did not specify what type of issue the Remote Offices are having, we will need to take a structured approach to this issue. Since there is an ISP in the middle of the topology and the entry point into the Regional Office 1 is R5, let's start by looking at the config for R5.

R5

```
R5#sh run
!
version 15.4
no service timestamps debug uptime
no service timestamps log uptime
no service password-encryption
!
hostname R5
!
boot-start-marker
boot-end-marker
!
no logging monitor
!
no aaa new-model
no ip domain lookup
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
redundancy
!
```



```
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key IPXpsk123 address 14.14.14.0
!
!
crypto ipsec transform-set IPXTransformSet esp-aes esp-sha-hmac
 mode tunnel
!
crypto ipsec profile IPXprofile
 set transform-set IPXTransformSet
!
interface Loopback0
 ip address 172.16.5.5 255.255.255.0
!
interface Tunnel5
 ip address 192.168.5.5 255.255.255.0
 no ip redirects
 no ip split-horizon eigrp 100
 ip nhrp map multicast dynamic
 ip nhrp network-id 5
 tunnel source 14.14.14.5
 tunnel mode gre multipoint
 tunnel protection ipsec profile IPXprofile
!
interface Ethernet0/0
 ip address 172.16.30.2 255.255.255.252
!
interface Serial2/0
 ip address 14.14.14.5 255.255.255.252
 serial restart-delay 0
```

```
!
!
router eigrp 100
 network 172.16.30.2 0.0.0.0
 network 192.168.5.5 0.0.0.0
!
router bgp 1110
 bgp log-neighbor-changes
 neighbor 14.14.14.6 remote-as 45
!
```

We can glean some good information from this config. First, it looks like there is a DMVPN setup using IPSec. Second, R5 is connected to the provider via BGP. Third, EIGRP is configured for AS 100. This is important to note since the Remote Offices are also apart of EIGRP 100. At this point, you can assume that the Remote Offices are connected to R5 via the DMVPN Tunnel 5 interface. On R5, let's see if we can gain any information from the logs.

R5

```
R5#sh log

%BGP_SESSION-5-ADJCHANGE: neighbor 14.14.14.6 IPv4 Unicast topology
base removed from session  BGP Notification received

%BGP-3-NOTIFICATION: received from neighbor 14.14.14.6 active 2/2 (peer
in wrong AS) 2 bytes 0456

%BGP-5-NBR_RESET: Neighbor 14.14.14.6 active reset (BGP Notification
received)

%BGP-5-ADJCHANGE: neighbor 14.14.14.6 active Down BGP Notification
received

%BGP_SESSION-5-ADJCHANGE: neighbor 14.14.14.6 IPv4 Unicast topology
base removed from session  BGP Notification received

%BGP-3-NOTIFICATION: received from neighbor 14.14.14.6 active 2/2 (peer
in wrong AS) 2 bytes 0456
```

Now we can see that the BGP is not in the correct AS. Looking at the BGP diagram, we can see that ISP4 is in BGP AS 44. It is currently set for remote-as 45. Lets change this and observe the results.

R5


```
R5(config)#router bgp 1110

R5(config-router)#no nei 14.14.14.6 remote-as 45

R5(config-router)#nei 14.14.14.6 remote-as 44

%BGP-5-NBR_RESET: Neighbor 14.14.14.6 active reset (Remote AS changed)

%BGP-5-ADJCHANGE: neighbor 14.14.14.6 active Down BGP Notification sent

%BGP_SESSION-5-ADJCHANGE: neighbor 14.14.14.6 IPv4 Unicast topology
base removed from session  BGP Notification sent

R5(config-router)#

%BGP-5-ADJCHANGE: neighbor 14.14.14.6 Up

R5(config-router)#

%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.168.5.23 (Tunnel5) is
up: new adjacency

%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.168.5.22 (Tunnel5) is
up: new adjacency
```

Nice! You can see that BGP came up as well as the EIGRP neighbor relationships on the DMVPN interface with R22 and R23! It's not all good news though. The EIGRP relationship with R21 did not come up. Let's take a look at R21.

R21

```
R21#sh dmvpn
Interface: Tunnel5, IPv4 NHRP Details
Type:Spoke, NHRP Peers:1,

 # Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb
 ----- --------------- --------------- ----- -------- -----
     1 14.14.14.5          192.168.5.5   IKE 10:24:51
```

You can see that the DMVPN is not up on R21. The state should show UP, not IKE. This is indicative of a mis-configuration with IPSEC. We need to compare the IPSec/ISAKMP configuration of R5 with that of R21.

R5

```
R5#sh run | sec crypto
crypto isakmp policy 10
 authentication pre-share
crypto isakmp key IPXpsk123 address 14.14.14.0
crypto ipsec transform-set IPXTransformSet esp-aes esp-sha-hmac
 mode tunnel
crypto ipsec profile IPXprofile
 set transform-set IPXTransformSet
```

R21

```
R21# sh run | sec crypto
crypto isakmp policy 10
 authentication pre-share
crypto ipsec transform-set IPXTransformSet esp-aes esp-sha-hmac
 mode tunnel
crypto ipsec profile IPXprofile
 set transform-set IPXTransformSet
```

Did you notice that R21 does not have a pre-shared key configured? The next step is to configure the pre-shared key on R21.

R21

```
R21(config)#crypto isakmp key IPXpsk123 address 14.14.14.0

R21(config-if)#do sh dmvpn

# Ent  Peer NBMA Addr Peer Tunnel Add State  UpDn Tm Attrb

 ----- --------------- --------------- ----- -------- -----

     1 14.14.14.5          192.168.5.5    UP 00:01:05     S


R21(config-if)#do sh ip eigrp nei

EIGRP-IPv4 Neighbors for AS(100)


R21(config-if)#do sh run | sec eigrp

 no ip split-horizon eigrp 100

router eigrp 100

 network 192.168.21.1 0.0.0.0
```

The DMVPN connection is now up, however, the EIGRP relationship is still not forming. There is not a network statement for the Tunnel interface.

R21

```
R21(config)#router eigrp 100
R21(config-router)#network 192.168.5.21 0.0.0.0
R21(config-router)#
%DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 192.168.5.5 (Tunnel5) is
up: new adjacency
```


Troubleshooting Verification

Since the task asks for full reachability within the EIGRP 100 topology, the following ping tests from each router within the AS is now working. In the example, the pings are sourced from SW1.

SW1

```
SW1#ping 192.168.21.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.21.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 17/17/18 ms


SW1#ping 192.168.22.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.22.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 14/16/18 ms


SW1#ping 192.168.23.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.23.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/17/18 ms
```

Configuration Changes Summary

R21

```
R21(config)#router eigrp 100
R21(config-router)#network 192.168.5.21 0.0.0.0


R21(config)#crypto isakmp key IPXpsk123 address 14.14.14.0
```

R5

```
R5(config)#router bgp 1110
R5(config-router)#no nei 14.14.14.6 remote-as 45
R5(config-router)#nei 14.14.14.6 remote-as 44
```


Diagnostic Lab 1 – Detailed Solutions Guide

Task #1 EIGRP

Answer

Using the information provided, select the most logical cause of the issue from the list below:

- ○ The network statements are not correctly configured on R2
- ○ The network summary from R1 is configured with the wrong subnet mask.
- ○ R1 has an ACL that filters and drops EIGRP packets.
- ○ The EIGRP hold-time on R2 is set smaller than the hello-interval on R1, causing the neighbors to flap continuously.
- ○ R2 has an ACL that filters and drops EIGRP packets.
- ○ R1 int e0/0 has the wrong subnet mask.
- ○ The EIGRP hold-interval on R2 is set smaller than the hello-interval on R2, causing the neighbors to flap continuously.
- ○ The variance is incorrectly set on R2.
- ○ The network summary on R1 is configured with the wrong subnet mask.
- ○ R2 int e0/0 has an incorrectly configured subnet mask.
- ○ The EIGRP hello-interval on R2 is set smaller than the hold-interval on R1, causing the neighbors to flap continuously.
- ○ The variance is incorrectly set on R1.

Explanation

Looking at the output provided by the client, we see that the EIGRP adjacency is bouncing between R1 and R2. We also notice in the configs that the EIGRP hello-interval has been manipulated.

Let's discuss EIGRP hello/hold/dead timers.

- EIGRP Hold Interval – This is the rate at which EIGRP sends hello packets to it's neighbor. The normal default interval is 5 seconds for Ethernet and 60 seconds for NBMA networks.

- EIGRP Hold-Down Timer – This is the amount of time a router will wait for a hello from it's neighbor before considering the the neighbor dead.

Let's take a look at what's going on in this incident.

R1

```
interface Ethernet0/0
 ip address 10.10.1.1 255.255.255.0
 ip summary-address eigrp 2001 192.168.0.0 255.255.254.0
 ip hello-interval eigrp 2001 120
```

R2

```
interface Ethernet0/1.1
 encapsulation dot1Q 12
 ip address 10.10.1.2 255.255.255.0
 ip hold-time eigrp 2001 60
```

R1 is sending out a hello every 120 and R2's hold-timer is set to 60 seconds. If after 60 seconds R2 does not recive a hello, it tears down the EIGRP neighbor. This is why the link is bouncing.

Task #2 BGP

Answers

Put the following BGP route selection criteria in order from the top down.

1 - Prefer the path with the highest WEIGHT

2 - Prefer the path with the highest LOCAL_PREF

3 - Prefer the path that was locally originated via a network or aggregate BGP subcommand or through redistribution from an IGP.

4 - Prefer the path with the shortest AS_PATH.

5 - Prefer the path with the lowest origin type.

6 - Prefer the path with the lowest MED.

7 - Prefer eBGP over iBGP paths

8 - Prefer the path with the lowest IGP metric to the BGP next hop.

9 - When both paths are external, prefer the path that was received first (the oldest one).

10 - Prefer the route that comes from the BGP router with the lowest router ID

11 - Prefer the path with the minimum cluster list length.

12 - Prefer the path that comes from the lowest neighbor address

Explanation

For this explanation, lets reference the cisco document “BGP Best Path Selection Algorithm” found at http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html.

1. Weight – Cisco defined attribute and is assigned locally. Weight does not get passed through router updates.
2. LOCAL_PREF – This is advertised to the AS and indicates which path has the highest preference. Think of the customer telling the provider router which path they should prefer to reach the customer network.
3. Network/Aggregate – This specifies if the path was advertised using a network command or if it is an aggregate (summary-address) which reduces overhead across the network.

4. AS-Path – BGP compares the first 3 criteria, and if there is a tie, the path with the shortest AS-PATH value wins.
5. Origin Type – We all know that IGP's are lower preference than BGP. This tells BGP to prefer BGP routes over IGP routes.
6. MED – Multi-Exit-Discriminator, this is commonly refered to as the external metric of BGP. Lowest MED wins.
7. eBGP vs iBGP – BGP best path selection prefers eBGP routes over iBGP.
8. Lowest IGP Metric – the path with the lowest IGP metric will be chosen as the best path.
9. First Received – If multiple paths exist, and if those multiple paths are external routes, prefer the route that was learned first.
10. Lowest Router-ID – Prefer the route with the lowest router ID.
11. Minimum Cluster-List – Yet another attribute to compare. If the router-id somehow is the same, then prefer the path with the minimum cluster list length.
12. Lowest Neighbor Address – Lastly, prefer the path with the lowest neighbor address.

Task #3 (3 points):

Answers

Device with Issue:

- ○ R1
- ○ R2
- ○ R3
- ○ R4
- ○ ISP1

Area of Issue:

- ○ Change OSPF network type
- ○ Change OSPF priority to 1
- ○ Change to Area 1
- ○ Change OSPF cost to 100
- ○ Change OSPF priority to 0

Explanation

We can see from the debug output of R1 that R4 has been elected the DR for the OSPF network. Since R1 is the DMVPN hub, it should be elected DR and no other router should be DR or BDR. Even though the adjacency forms, no routes will be advertised to R2 or R3 since they are not connected to a DR or BDR. If we make R1 the DR, it will advertise the routes properly to all devices in the OSPF domain. However, keep in mind that by just changing R1's priority, it does not guarantee that R4 will not become DR or BDR. It would be safer and more efficient to change R4's OSPF priority to 0, telling it to never become DR/BDR.

Configuration Lab 1 – Detailed Solution Guide

Layer 2 Technologies San Jose Switch-to-Switch Links

Configuration

Create the trunk links between SW1, SW2, SW3, and SW4. Remember to only allow the VLAN's required across the trunk. Also, use Dot1Q encapsulation.

SW1

```
SW1(config)#int range e3/0 - 2 , e4/0 - 1 , e5/0 - 1

SW1(config-if-range)#switchport

SW1(config-if-range)#switchport trunk encapsulation dot1q

SW1(config-if-range)#switchport mode trunk

SW1(config-if-range)#switchport trunk allowed vlan
45,93,101,103,104,105,910
```

SW2

```
SW2(config)#int range e3/0 - 2 , e4/0 - 1 , e5/0 - 1

SW2(config-if-range)#switchport

SW2(config-if-range)#switchport trunk encapsulation dot1q

SW2(config-if-range)#switchport mode trunk

SW2(config-if-range)#switchport trunk allowed vlan
45,93,101,103,104,105,910
```

SW3

```
SW3(config)#int range e3/0 -1 , e4/0 - 1 , e5/0 - 1

SW3(config-if-range)#switchport

SW3(config-if-range)#switchport trunk encap dot1q
```



```
SW3(config-if-range)#switchport mode trunk

SW3(config-if-range)#switchport trunk allowed vlan
45,93,101,103,104,105,910
```

SW4

```
SW4(config)#int range e3/0 -1 , e4/0 - 1 , e5/0 - 1

SW4(config-if-range)#switchport

SW4(config-if-range)#switchport trunk encap dot1q

SW4(config-if-range)#switchport mode trunk

SW4(config-if-range)#switchport trunk allowed vlan
45,93,101,103,104,105,910
```

Create the Ether-channel between SW1 and SW2. The task asks you to use a Cisco proprietary protocol so make sure to use PagP.

SW1

```
SW1(config-if-range)#interface port-channel 1

SW1(config-if)#switchport

SW1(config-if)#switchport trunk encap dot1q

SW1(config-if)#switchport mode trunk

SW1(config-if)#switchport trunk allowed vlan 45,93,101,103,104,105,910


SW1(config-if)#int range e3/0 - 2

SW1(config-if-range)#channel-group 1 mode desirable
```

SW2

```
SW2(config-if-range)#interface port-channel 1

SW2(config-if)#switchport

SW2(config-if)#switchport trunk encap dot1q

SW2(config-if)#switchport mode trunk

SW2(config-if)#switchport trunk allowed vlan 45,93,101,103,104,105,910
```

```
SW2(config-if)#int range e3/0 - 2
SW2(config-if-range)#channel-group 1 mode desirable
```

Configuration Verification

Since there no VLANs configured yet, let's verify that the port channel came up and the appropriate interfaces are in the port-channel. Also verify that it is using PAgP.

SW1

```
SW1#sh etherchannel
                Channel-group listing:
                ----------------------
Group: 1
----------
Group state = L2
Ports: 3   Maxports = 4
Port-channels: 1 Max Port-channels = 1
Protocol:   PAgP
Minimum Links: 0
```

```
SW1#show interface port-channel 1
Port-channel1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is aabb.cc00.6523 (bia aabb.cc00.6523)
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed, media type is unknown
```



```
  input flow-control is off, output flow-control is unsupported
  Members in this channel: Et3/0 Et3/1 Et3/2
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops:
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 1000 bits/sec, 1 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     12920 packets input, 1834868 bytes, 0 no buffer
     Received 9837 broadcasts (0 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     29488 packets output, 2971809 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

SW2

```
SW2#sh etherchannel
                Channel-group listing:
                ----------------------
Group: 1
----------
Group state = L2
```

```
Ports: 3   Maxports = 4
Port-channels: 1 Max Port-channels = 1
Protocol:   PAgP
Minimum Links: 0
```

```
SW2#sh int port-channel 1
Port-channel1 is up, line protocol is up (connected)
  Hardware is Ethernet, address is aabb.cc00.6623 (bia aabb.cc00.6623)
  MTU 1500 bytes, BW 10000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed, media type is unknown
  input flow-control is off, output flow-control is unsupported
  Members in this channel: Et3/0 Et3/1 Et3/2
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops:
0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
     28152 packets input, 2870559 bytes, 0 no buffer
     Received 25080 broadcasts (0 multicasts)
     0 runts, 0 giants, 0 throttles
```



```
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 input packets with dribble condition detected
12964 packets output, 1835537 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 unknown protocol drops
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out
```

Layer 2 Technologies California HQ VLANs

Configuration

Configure SW2, SW3, and SW4 as VTP clients and SW1 as the VTP master using the credentials supplied in the section. Configure all required VLANs on SW1 according to the diagram.

SW2, SW3, SW4

```
(config)#vtp domain CCIE
(config)#vtp password iPexpert!
(config)#vtp mode client
```

SW1

```
SW1(config)#vtp domain CCIE
SW1(config)#vtp password iPexpert!
SW1(config)#vtp mode server
SW1(config)#vlan 45
SW1(config-vlan)#vlan 93
SW1(config-vlan)#vlan 104
SW1(config-vlan)#vlan 101
SW1(config-vlan)#vlan 103
SW1(config-vlan)#vlan 105
```

```
SW1(config-vlan)#vlan 910
```

Configuration Verfication

We need to verify that VTP is working and that SW2 is the VTP master. We also need to verify that all of the VTP client switches are learning the VLAN's that were created on SW1.

SW1

```
SW1#show vtp status

VTP Version capable             : 1 to 3

VTP version running             : 1

VTP Domain Name                 : CCIE

VTP Pruning Mode                : Disabled

VTP Traps Generation            : Disabled

Device ID                       : aabb.cc00.6500

Local updater ID is 131.10.45.17 on interface Vl101 (lowest numbered
VLAN interface found)


Feature VLAN:

--------------

VTP Operating Mode              : Server

Maximum VLANs supported locally   : 1005

Number of existing VLANs        : 14

Configuration Revision          : 8
```

SW2

```
SW2#show vtp status

VTP Version capable             : 1 to 3

VTP version running             : 1

VTP Domain Name                 : CCIE

VTP Pruning Mode                : Disabled

VTP Traps Generation            : Disabled
```



```
Device ID                          : aabb.cc00.6600


Feature VLAN:

--------------

VTP Operating Mode                 : Client

Maximum VLANs supported locally    : 1005

Number of existing VLANs           : 14

Configuration Revision             : 8
```

SW3

```
SW3#show vtp status

VTP Version capable                : 1 to 3

VTP version running                : 1

VTP Domain Name                    : CCIE

VTP Pruning Mode                   : Disabled

VTP Traps Generation               : Disabled

Device ID                          : aabb.cc00.6700


Feature VLAN:

--------------

VTP Operating Mode                 : Client

Maximum VLANs supported locally    : 1005

Number of existing VLANs           : 14

Configuration Revision             : 8
```

SW4

```
SW4#show vtp status

VTP Version capable                : 1 to 3

VTP version running                : 1

VTP Domain Name                    : CCIE
```

```
VTP Pruning Mode                 : Disabled

VTP Traps Generation             : Disabled

Device ID                        : aabb.cc00.6800

Configuration last modified by 2.2.2.2 at 10-5-14 10:49:03


Feature VLAN:

--------------

VTP Operating Mode               : Client

Maximum VLANs supported locally  : 1005

Number of existing VLANs         : 14

Configuration Revision           : 8
```


Layer 2 Technologies California HQ Router Links

Configuration

Using the layer 2 diagram, IPv4 diagram, and the table in the task, configure each Router link interfaces as access ports and assign them to the correct VLAN.

SW1

```
SW1(config-if)# switchport trunk encapsulation dot1q
SW1(config-if)# switchport trunk allowed vlan 103,105
SW1(config-if)# switchport mode trunk
SW1(config-if)#int e0/3
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 93
SW1(config-if)#int e1/2
SW1(config-if)#switchport
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 104
SW1(config-if)#int e1/1
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 105


SW1(config)#int vlan 910
SW1(config-if)#ip address 131.10.45.14 255.255.255.252
SW1(config-if)#no shut
SW1(config-if)#int vlan 101
SW1(config-if)#ip address 131.10.45.17 255.255.255.252
SW1(config-if)#no shut
SW1(config-if)#int vlan 104
SW1(config-if)#ip address 131.10.45.5 255.255.255.252
```

```
SW1(config-if)#no shut
```

SW2

```
SW2(config)#int e0/1
SW2(config-if)#switchport mode access
SW2(config-if)#switchport access vlan 101
SW2(config-if)#int e0/3
SW2(config-if)#switchport mode access
SW2(config-if)#switchport access vlan 103
SW2(config-if)#int e2/1
SW2(config-if)#switchport mode access
SW2(config-if)#switchport access vlan 910
```

SW3

```
SW3(config)#int e1/0
SW3(config-if)#switchport mode access
SW3(config-if)#switchport access vlan 45
SW3(config-if)#int e1/1
SW3(config-if)#switchport mode access
SW3(config-if)#switchport access vlan 45
```

SW4

```
SW4(config)#int e2/1
SW4(config-if)#switchport mode access
SW4(config-if)#switchport access vlan 93
```

Configuration Verification

We now need to take a look at the vlan assignments. We also need to verify that only the correct VLANs are going across the trunk links since we are now done with configuring VLANs and uplinks.

SW1

```
SW1#show interface trunk

Port        Mode             Encapsulation  Status        Native vlan
Et4/0       on               802.1q         trunking      1
Et4/1       on               802.1q         trunking      1
Et5/0       on               802.1q         trunking      1
Et5/1       on               802.1q         trunking      1
Po1         on               802.1q         trunking      1

Port        Vlans allowed on trunk
Et4/0       45,93,101,103,104,105,910
Et4/1       45,93,101,103,104,105,910
Et5/0       45,93,101,103,104,105,910
Et5/1       45,93,101,103,104,105,910
Po1         45,93,101,103,104,105,910

Port        Vlans allowed and active in management domain
Et4/0       45,93,101,103,104,105,910
Et4/1       45,93,101,103,104,105,910
Et5/0       45,93,101,103,104,105,910
Et5/1       45,93,101,103,104,105,910
Po1         45,93,101,103,104,105,910

Port        Vlans in spanning tree forwarding state and not pruned
Et4/0       45,93,101,103,104,105,910
Et4/1       45,93,101,103,104,105,910
Et5/0       45,93,101,103,104,105,910
```

```
Et5/1       45,93,101,103,104,105,910
Po1         45,93,101,103,104,105,910
```

SW2

```
SW2#show interface trunk
Port        Mode             Encapsulation  Status        Native vlan
Et4/0       on               802.1q         trunking      1
Et4/1       on               802.1q         trunking      1
Et5/0       on               802.1q         trunking      1
Et5/1       on               802.1q         trunking      1
Po1         on               802.1q         trunking      1

Port        Vlans allowed on trunk
Et4/0       45,93,101,103,104,105,910
Et4/1       45,93,101,103,104,105,910
Et5/0       45,93,101,103,104,105,910
Et5/1       45,93,101,103,104,105,910
Po1         45,93,101,103,104,105,910

Port        Vlans allowed and active in management domain
Et4/0       45,93,101,103,104,105,910
Et4/1       45,93,101,103,104,105,910
Et5/0       45,93,101,103,104,105,910
Et5/1       45,93,101,103,104,105,910
Po1         45,93,101,103,104,105,910

Port        Vlans in spanning tree forwarding state and not pruned
Et4/0       45,93,101,103,104,105,910
Et4/1       45,93,101,103,104,105,910
```



```
Et5/0       45,93,101,103,104,105,910
Et5/1       45,93,101,103,104,105,910
Po1         45,93,101,103,104,105,910
```

SW3

```
SW3#show interface trunk
Port        Mode             Encapsulation  Status        Native vlan
Et3/0       on               802.1q         trunking      1
Et3/1       on               802.1q         trunking      1
Et4/0       on               802.1q         trunking      1
Et4/1       on               802.1q         trunking      1
Et5/0       on               802.1q         trunking      1
Et5/1       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Et3/0       45,93,101,103,104,105,910
Et3/1       45,93,101,103,104,105,910
Et4/0       45,93,101,103,104,105,910
Et4/1       45,93,101,103,104,105,910
Et5/0       45,93,101,103,104,105,910
Et5/1       45,93,101,103,104,105,910

Port        Vlans allowed and active in management domain
Et3/0       45,93,101,103,104,105,910
Et3/1       45,93,101,103,104,105,910
Et4/0       45,93,101,103,104,105,910
Et4/1       45,93,101,103,104,105,910
Et5/0       45,93,101,103,104,105,910
Et5/1       45,93,101,103,104,105,910
```

```
Port        Vlans in spanning tree forwarding state and not pruned
Et3/0       45,93,101,103,104,105,910
Et3/1       45,93,101,103,104,105,910
Et4/0       104,910
Et4/1       none
Et5/0       45,93,101,103,105
Et5/1       none
```

SW4

```
SW4#show interface trunk
Port        Mode             Encapsulation  Status        Native vlan
Et3/0       on               802.1q         trunking      1
Et3/1       on               802.1q         trunking      1
Et4/0       on               802.1q         trunking      1
Et4/1       on               802.1q         trunking      1
Et5/0       on               802.1q         trunking      1
Et5/1       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Et3/0       45,93,101,103,104,105,910
Et3/1       45,93,101,103,104,105,910
Et4/0       45,93,101,103,104,105,910
Et4/1       45,93,101,103,104,105,910
Et5/0       45,93,101,103,104,105,910
Et5/1       45,93,101,103,104,105,910

Port        Vlans allowed and active in management domain
Et3/0       45,93,101,103,104,105,910
```



```
Et3/1       45,93,101,103,104,105,910
Et4/0       45,93,101,103,104,105,910
Et4/1       45,93,101,103,104,105,910
Et5/0       45,93,101,103,104,105,910
Et5/1       45,93,101,103,104,105,910

Port        Vlans in spanning tree forwarding state and not pruned
Et3/0       none
Et3/1       none
Et4/0       45,93,101,103,105
Et4/1       none
Et5/0       104,910
Et5/1       none
```

Layer 2 Technologies California HQ Spanning-Tree

Configuration

The task calls for us to use 802.1S, which is MST. We need to configure MST spanning-tree, assign all even VLANs to MST Instance 10, and assign all odd VLANs to MST Instance 11.

SW1

```
SW1(config)#spanning-tree mst configuration
SW1(config-mst)#instance 10 vlan 104,910
SW1(config-mst)#instance 11 vlan 45,93,101,103,105
SW1(config-mst)#exit
SW1(config)#spanning-tree mode mst
```

SW2

```
SW2(config)#spanning-tree mst configuration
SW2(config-mst)#instance 10 vlan 104,910
SW2(config-mst)#instance 11 vlan 45,93,101,103,105
SW2(config-mst)#exit
SW2(config)#spanning-tree mode mst
```

SW3

```
SW3(config)#spanning-tree mst configuration
SW3(config-mst)#instance 10 vlan 104,910
SW3(config-mst)#instance 11 vlan 45,93,101,103,105
SW3(config-mst)#spanning-tree mode mst
```

SW4

```
SW4(config-if)#spanning-tree mst configuration
SW4(config-mst)#instance 10 vlan 104,910
SW4(config-mst)#instance 11 vlan 45,93,101,103,105
SW4(config-mst)#spanning-tree mode mst
```


Now we need to make SW1 the primary root for instance 11 and secondary root for instance 11. Then, lets make SW2 the primary root for instance 10 and secondary root for instance 11. We also need to make sure that these root assignments stay the way they are, even if a more preferred switch is plugged into the network. We can do this by using the "root primary" and "root secondary" commands instead of manipulating priority.

SW1

```
SW1(config)#spanning-tree mst 10 root secondary
SW1(config)#spanning-tree mst 11 root primary
```

SW2

```
SW2(config)#spanning-tree mst 10 root primary
SW2(config)#spanning-tree mst 11 root secondary
```

Configuration Verification

Let's take a look at the spanning-tree tables to verify root assignments.

SW1

```
SW1#show spanning-tree
MST10
  Spanning tree enabled protocol mstp
  Root ID    Priority    24586
             Address     aabb.cc00.6600
             Cost        2000000
             Port        65 (Port-channel1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    28682  (priority 28672 sys-id-ext 10)
             Address     aabb.cc00.6500
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Et1/2               Desg FWD 2000000   128.7    Shr
Et4/0               Desg FWD 2000000   128.17   Shr
Et4/1               Desg FWD 2000000   128.18   Shr
Et5/0               Desg FWD 2000000   128.21   Shr
Et5/1               Desg FWD 2000000   128.22   Shr
Po1                 Root FWD 2000000   128.65   Shr

MST11
  Spanning tree enabled protocol mstp
  Root ID    Priority    24587
             Address     aabb.cc00.6500
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24587  (priority 24576 sys-id-ext 11)
             Address     aabb.cc00.6500
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Et0/1               Desg FWD 2000000   128.2    Shr
Et0/3               Desg FWD 2000000   128.4    Shr
Et1/1               Desg FWD 2000000   128.6    Shr
Et4/0               Desg FWD 2000000   128.17   Shr
Et4/1               Desg FWD 2000000   128.18   Shr
```



```
Et5/0               Desg FWD 2000000   128.21   Shr
Et5/1               Desg FWD 2000000   128.22   Shr
Po1                 Desg FWD 2000000   128.65   Shr
```

SW2

```
SW2#show spanning-tree
MST10
  Spanning tree enabled protocol mstp
  Root ID    Priority    24586
             Address     aabb.cc00.6600
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24586  (priority 24576 sys-id-ext 10)
             Address     aabb.cc00.6600
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Et0/3               Desg FWD 2000000   128.4    Shr
Et2/1               Desg FWD 2000000   128.10   Shr
Et4/0               Desg FWD 2000000   128.17   Shr
Et4/1               Desg FWD 2000000   128.18   Shr
Et5/0               Desg FWD 2000000   128.21   Shr
Et5/1               Desg FWD 2000000   128.22   Shr
Po1                 Desg FWD 2000000   128.65   Shr
```

```
MST11
  Spanning tree enabled protocol mstp
  Root ID    Priority    24587
             Address     aabb.cc00.6500
             Cost        2000000
             Port        65 (Port-channel1)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    28683  (priority 28672 sys-id-ext 11)
             Address     aabb.cc00.6600
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------------
Et0/1               Desg FWD 2000000   128.2    Shr
Et4/0               Desg FWD 2000000   128.17   Shr
Et4/1               Desg FWD 2000000   128.18   Shr
Et5/0               Desg FWD 2000000   128.21   Shr
Et5/1               Desg FWD 2000000   128.22   Shr
Po1                 Root FWD 2000000   128.65   Shr
```


Layer 2 Technologies Texas VLANs

Configuration

The tasks asks us to configure a VTP mode that doesn't modify the VLAN database, but passes VTP information through. On both SW5 and SW6, we need to configure VTP in transparent mode and in version 2.

SW5 and SW6

```
(config)#vtp mode transparent
(config)#vtp version 2
(config-vlan)#vlan 105
(config-vlan)#vlan 53
(config-vlan)#vlan 120
(config-vlan)#vlan 123
(config-vlan)#vlan 121
(config-vlan)#vlan 134
(config-vlan)#vlan 114
```

Configuration Verification

Let's take a look at the VTP status of both SW5 and SW6 to verify both are version 2 and running in transparent mode.

SW5

```
SW5#show vtp status
VTP Version capable             : 1 to 3
VTP version running             : 2
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : aabb.cc00.6900
Configuration last modified by 172.16.105.105 at 0-0-00 00:00:00
```

```
Feature VLAN:

--------------

VTP Operating Mode                : Transparent

Maximum VLANs supported locally   : 1005

Number of existing VLANs          : 12

Configuration Revision            : 0
```

SW6

```
SW6#show vtp status

VTP Version capable             : 1 to 3

VTP version running             : 2

VTP Domain Name                 :

VTP Pruning Mode                : Disabled

VTP Traps Generation            : Disabled

Device ID                       : aabb.cc00.6a00

Configuration last modified by 172.16.105.105 at 0-0-00 00:00:00


Feature VLAN:

--------------

VTP Operating Mode                : Transparent

Maximum VLANs supported locally   : 1005

Number of existing VLANs          : 12

Configuration Revision            : 0
```


Layer 2 Technologies Texas Router Links

Configuration

Now it is time to configure the switchports that are connected to the Routers in Texas. Use the table provided, along with the Layer 2 and IPv4 diagrams to assign the correct VLAN to the switchports needed.

SW5

```
SW5(config)#int e0/0
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 105
SW5(config-if)#int e0/1
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 121
SW5(config-if)#int e0/2
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 123
SW5(config-if)#int e1/0
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 123
SW5(config-if)#int e0/3
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 134
SW5(config-if)#int e1/1
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 134
SW5(config-if)#int e1/3
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 105
```

SW6

```
SW5(config)#int e0/0
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 120
SW5(config-if)#int e0/1
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 114
SW5(config-if)#int e0/2
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 120
SW5(config-if)#int e0/3
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 121
SW5(config-if)#int e1/0
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 53
SW5(config-if)#int e1/2
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 114
SW5(config-if)#int e1/3
SW5(config-if)#switchport mode access
SW5(config-if)#switchport access vlan 53
```

Configuration Verification

Let's verify VLAN assignments.

SW5

```
SW5#show int status
Port      Name               Status       Vlan       Duplex  Speed Type
```



```
Et0/0                     connected    105          auto   auto unknown
Et0/1                     connected    121          auto   auto unknown
Et0/2                     connected    123          auto   auto unknown
Et0/3                     connected    134          auto   auto unknown
Et1/0                     connected    123          auto   auto unknown
Et1/1                     connected    134          auto   auto unknown
Et1/2                     connected    1            auto   auto unknown
Et1/3                     connected    105          auto   auto unknown
Et2/0                     connected    trunk        auto   auto unknown
Et2/1                     connected    trunk        auto   auto unknown
Et2/2                     connected    trunk        auto   auto unknown
Et2/3                     connected    trunk        auto   auto unknown
Et3/0                     connected    1            auto   auto unknown
Et3/1                     connected    1            auto   auto unknown
Et3/2                     connected    1            auto   auto unknown
Et3/3                     connected    1            auto   auto unknown
Et4/0                     connected    1            auto   auto unknown
Et4/1                     connected    1            auto   auto unknown
Et4/2                     connected    1            auto   auto unknown
Et4/3                     connected    1            auto   auto unknown
Et5/0                     connected    1            auto   auto unknown
Et5/1                     connected    1            auto   auto unknown
Et5/2                     connected    1            auto   auto unknown
Et5/3                     connected    1            auto   auto unknown
Po1                       connected    trunk        auto   auto
```

SW6

```
SW6#show interface status
Port      Name               Status       Vlan       Duplex  Speed Type
```

```
Et0/0                     connected    120             auto   auto unknown
Et0/1                     connected    114             auto   auto unknown
Et0/2                     connected    120             auto   auto unknown
Et0/3                     connected    121             auto   auto unknown
Et1/0                     connected    53              auto   auto unknown
Et1/1                     connected    1               auto   auto unknown
Et1/2                     connected    114             auto   auto unknown
Et1/3                     connected    53              auto   auto unknown
Et2/0                     connected    trunk           auto   auto unknown
Et2/1                     connected    trunk           auto   auto unknown
Et2/2                     connected    trunk           auto   auto unknown
Et2/3                     connected    trunk           auto   auto unknown
Et3/0                     connected    1               auto   auto unknown
Et3/1                     connected    1               auto   auto unknown
Et3/2                     connected    1               auto   auto unknown
Et3/3                     connected    1               auto   auto unknown
Et4/0                     connected    1               auto   auto unknown
Et4/1                     connected    1               auto   auto unknown
Et4/2                     connected    1               auto   auto unknown
Et4/3                     connected    1               auto   auto unknown
Et5/0                     connected    1               auto   auto unknown
Et5/1                     connected    1               auto   auto unknown
Et5/2                     connected    1               auto   auto unknown
Et5/3                     connected    1               auto   auto unknown
Po1                       connected    trunk           auto   auto
```


Layer 2 Technologies Texas Switch to Switch Links

Configuration

The first step to this task is to configure a port channel between the uplinks of SW5 and SW6. This needs to be done using LaCP as it states to use a non-proprietary protocol. Also, we need to only allow the needed VLANs across the trunks.

SW5 and SW6

```
(config-if)#int port-channel 1
(config-if)#switchport
(config-if)#switchport trunk encapsulation dot1q
(config-if)#switchport mode trunk
(config-if)#switchport trunk allowed vlan 53,105,120,123,121,134,114


(config-if)#int range e2/0 - 3
(config-if-range)#switchport trunk encap dot1q
(config-if-range)#switchport mode trunk
(config-if-range)#switchport trunk allowed vlan
53,105,120,123,121,134,114
(config-if-range)#channel-group 1 mode active
```

Configuration Verification

Let's show the trunk interfaces to see which VLANs are allowed across the trunks. It will also show if Port-Channel 1 is being used as a trunk.

SW5

```
SW5#show interface trunk

Port        Mode             Encapsulation  Status        Native vlan
Po1         on               802.1q         trunking      1
```

```
Port        Vlans allowed on trunk
Po1         53,105,114,120-121,123,134

Port        Vlans allowed and active in management domain
Po1         53,105,114,120-121,123,134

Port        Vlans in spanning tree forwarding state and not pruned
Po1         53,105,114,120-121,123,134
```

SW6

```
SW6#show interface trunk

Port        Mode             Encapsulation  Status        Native vlan
Po1         on               802.1q         trunking      1

Port        Vlans allowed on trunk
Po1         53,105,114,120-121,123,134

Port        Vlans allowed and active in management domain
Po1         53,105,114,120-121,123,134

Port        Vlans in spanning tree forwarding state and not pruned
Po1         53,105,114,120-121,123,134
```


Layer 2 Technologies Configuration Verification

Connectivity Verification

The task asks us to ping all directly connected links to verify Layer 2 connectivity.

SW1

```
SW1#ping 131.10.45.13
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.13, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

SW1#ping 131.10.45.18
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.18, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

SW1#ping 131.10.45.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.6, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

R1

```
R1#ping 131.10.45.17
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.17, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

```
R1#ping 131.10.45.34
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.34, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

R1#ping 131.10.45.38
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.38, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

R3

```
R3#ping 131.10.45.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

R3#ping 131.10.45.37
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.37, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

R4

```
R4#ping 131.10.45.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.5, timeout is 2 seconds:
```



```
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms


R4#ping 131.10.45.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.30, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

R5

```
R5#ping 131.10.45.29
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.29, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms


R5#ping 131.10.45.33
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.33, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

R9

```
R9#ping 131.10.45.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms


R9#ping 131.10.45.14
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.14, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

R10

```
R10#ping 172.16.11.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R10#ping 172.16.11.14
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.14, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

R11

```
R11#ping 172.16.11.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R11#ping 172.16.11.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```


R12

```
R12#ping 172.16.11.13
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.13, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R12#ping 172.16.11.18
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R12#ping 172.16.11.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

R13

```
R13#ping 172.16.11.26
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.26, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R13#ping 172.16.11.17
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.17, timeout is 2 seconds:
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms


R13#ping 172.16.11.21
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.21, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

R14

```
R14#ping 172.16.11.25
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.25, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms


R14#ping 172.16.11.29
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.29, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

R15

```
R15#ping 172.16.11.22
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.22, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms


R15#ping 172.16.11.1
```



```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

R16

```
R16#ping 172.16.18.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

R16#ping 172.16.18.6
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.6, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

R16#ping 172.16.18.22
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.22, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

R17

```
R17#ping 172.16.18.21
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.21, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
R17#ping 172.16.18.17
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.17, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

R17#ping 172.16.18.25
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.25, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

R18

```
R18#ping 172.16.18.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R18#ping 172.16.18.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.10, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

R18#ping 172.16.18.14
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.14, timeout is 2 seconds:
```



```
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

R19

```
R19#ping 172.16.18.9
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.9, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R19#ping 172.16.18.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R19#ping 172.16.18.26
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.26, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R19#ping 172.16.18.30
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.30, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

R20

```
R20#ping 172.16.18.29
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.29, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R20#ping 172.16.18.13
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.13, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R20#ping 172.16.18.18
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.18, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```


IP Routing OSPF at California

Configuration

We need to first acknowledge all of the requirements for this task. First, let's consider the basic OSPF configuration first, then we will work through the advanced configuration steps. This configuration will be in Area 0. Serial Links will not be added to OSPF. To restrict the OSPF process to only the interfaces specified, we should use very specific network commands under the OSPF process. Also, even though the task does not specifically call for us to do so, loopback 0 interfaces should be advertised with their original masks by using the OSPF network type point-to-point. This will be a requirement when we get to the MPLS section of the lab.

SW1

```
SW1(config)#router ospf 1
SW1(config-router)#network 131.10.45.14 0.0.0.0 area 0
SW1(config-router)#network 131.10.45.17 0.0.0.0 area 0
SW1(config-router)#network 131.10.45.5 0.0.0.0 area 0
SW1(config-router)#network 172.16.101.101 0.0.0.0 area 0

SW1(config-router)#interface loopback 0
SW1(config-if)#ip ospf network point-to-point
```

R1

```
R1(config)#router ospf 1
R1(config-router)#network 131.10.45.33 0.0.0.0 area 0
R1(config-router)#network 131.10.45.37 0.0.0.0 area 0
R1(config-router)#network 131.10.45.18 0.0.0.0 area 0
R1(config-router)#network 172.16.1.1 0.0.0.0 area 0

R1(config-router)#int loopback 0
R1(config-if)#ip ospf network point-to-point
```

R3

```
R3(config)#router ospf 1
R3(config-router)#network 131.10.45.38 0.0.0.0 area 0
R3(config-router)#network 131.10.45.2 0.0.0.0 area 0
R3(config-router)#network 172.16.3.3 0.0.0.0 area 0

R3(config-router)#int loopback 0
R3(config-if)#ip ospf network point-to-point
```

R4

```
R4(config)#router ospf 1
R4(config-router)#network 131.10.45.29 0.0.0.0 area 0
R4(config-router)#network 131.10.45.6 0.0.0.0 area 0
R4(config-router)#network 172.16.4.4 0.0.0.0 area 0

R4(config-router)#int loopback 0
R4(config-if)#ip ospf network point-to-point
```

R5

```
R5(config)#router ospf 1
R5(config-router)#network 131.10.45.33 0.0.0.0 area 0
R5(config-router)#network 131.10.45.30 0.0.0.0 area 0
R5(config-router)#network 172.16.5.5 0.0.0.0 area 0

R5(config-router)#int loopback 0
R5(config-if)#ip ospf network point-to-point
```

R9

```
R9(config)#router ospf 1
R9(config-router)#network 131.10.45.1 0.0.0.0 area 0
R9(config-router)#network 131.10.45.13 0.0.0.0 area 0
```



```
R9(config-router)#network 172.16.9.9 0.0.0.0 area 0


R9(config-router)#int loopback 0
R9(config-if)#ip ospf network point-to-point
```

Configuration Verification

Let's perform a connectivity test of all subnets at California HQ. We can do this with a TCL script. Below, there is an example from R9, but you will need to do this on all devices at California HQ.

R9

```
R9#tclsh
R9(tcl)#foreach address {
+>(tcl)#131.10.45.1
+>(tcl)#131.10.45.2
+>(tcl)#131.10.45.5
+>(tcl)#131.10.45.6
+>(tcl)#131.10.45.13
+>(tcl)#131.10.45.14
+>(tcl)#131.10.45.17
+>(tcl)#131.10.45.18
+>(tcl)#131.10.45.29
+>(tcl)#131.10.45.30
+>(tcl)#131.10.45.33
+>(tcl)#131.10.45.34
+>(tcl)#131.10.45.37
+>(tcl)#131.10.45.38
+>(tcl)#172.16.1.1
+>(tcl)#172.16.3.3
+>(tcl)#172.16.4.4
```

```
+>(tcl)#172.16.5.5
+>(tcl)#172.16.9.9
+>(tcl)#172.16.101.101
+>(tcl)#172.16.102.102
+>(tcl)#} { ping $address
+>(tcl)#}
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/5 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 131.10.45.2, 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 131.10.45.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Results Truncated….

IP Routing EIGRP AS 23456

Configuration

The first thing to note here is that EIGRP wide metrics are required. The only configuration that is able to use EIGRP wide metrics is EIGRP named config. This is a key observation as it changes the mindset of our configuration. Second, notice that EIGRP is using MD5 authentication. Also, as with our earlier OSPF configuration, we need to be very specific in our network statements.

R10

```
R10(config)#key chain EIGRP
R10(config-keychain)#key 1
R10(config-keychain-key)#key-string CCIERock$

R10(config)#router eigrp NAMED
R10(config-router-af)#af-interface default
R10(config-router-af-interface)#authentication mode md5
R10(config-router-af-interface)#authentication key-chain EIGRP
R10(config-router-af-interface)#exit
R10(config-router)#address-family ipv4 autonomous-system 23456
R10(config-router-af)#network 172.16.11.1 0.0.0.0
R10(config-router-af)#network 172.16.11.13 0.0.0.0
R10(config-router-af)#network 172.16.10.10 0.0.0.0
```

R11

```
R11(config-router-af)#key chain EIGRP
R11(config-keychain)#key 1
R11(config-keychain-key)#key-string CCIERock$

R11(config)#router eigrp NAMED
R11(config-router)#address-family ipv4 autonomous-system 23456
```

```
R11(config-router-af)#af-interface default
R11(config-router-af-interface)#authentication mode md5
R11(config-router-af-interface)#authentication key-chain EIGRP
R11(config-router-af-interface)#exit
R11(config-router-af)#network 172.16.11.29 0.0.0.0
R11(config-router-af)#network 172.16.11.6 0.0.0.0
R11(config-router-af)#network 172.16.111.111 0.0.0.0
```

R12

```
R12(config)#key chain EIGRP
R12(config-keychain)#key 1
R12(config-keychain-key)#key-string CCIERock$

R12(config)#router eigrp NAMED
R12(config-router)#address-family ipv4 autonomous-system 23456
R12(config-router-af)#af-interface default
R12(config-router-af-interface)#authentication mode md5
R12(config-router-af-interface)#authentication key-chain EIGRP
R12(config-router-af-interface)#exit
R12(config-router-af)#network 172.16.11.14 0.0.0.0
R12(config-router-af)#network 172.16.11.17 0.0.0.0
R12(config-router-af)#network 172.16.11.5 0.0.0.0
R12(config-router-af)#network 172.16.12.12 0.0.0.0
```

R13

```
R13(config)#key chain EIGRP
R13(config-keychain)#key 1
R13(config-keychain-key)#key-string CCIERock$

R13(config)#router eigrp NAMED
```



```
R13(config-router)#address-family ipv4 autonomous-system 23456
R13(config-router-af)#af-interface default
R13(config-router-af-interface)#authentication mode md5
R13(config-router-af-interface)#authentication key-chain EIGRP
R13(config-router-af-interface)#exit
R13(config-router-af)#network 172.16.11.22 0.0.0.0
R13(config-router-af)#network 172.16.11.25 0.0.0.0
R13(config-router-af)#network 172.16.11.18 0.0.0.0
R13(config-router-af)#network 172.16.13.13 0.0.0.0
```

R14

```
R14(config)#key chain EIGRP
R14(config-keychain)#key 1
R14(config-keychain-key)#key-string CCIERock$

R14(config)#router eigrp NAMED
R14(config-router)#address-family ipv4 autonomous-system 23456
R14(config-router-af)#af-interface default
R14(config-router-af-interface)#authentication mode md5
R14(config-router-af-interface)#authentication key-chain EIGRP
R14(config-router-af-interface)#exit
R14(config-router-af)#network 172.16.11.26 0.0.0.0
R14(config-router-af)#network 172.16.11.26 0.0.0.0
R14(config-router-af)#network 172.16.11.30 0.0.0.0
R14(config-router-af)#network 172.16.14.14 0.0.0.0
```

R15

```
R15(config)#key chain EIGRP
R15(config-keychain)#key 1
R15(config-keychain-key)#key-string CCIERock$
```

```
R15(config)#router eigrp NAMED
R15(config-router)#address-family ipv4 autonomous-system 23456
R15(config-router-af)#af-interface default
R15(config-router-af-interface)#authentication mode md5
R15(config-router-af-interface)#authentication key-chain EIGRP
R15(config-router-af-interface)#exit
R15(config-router-af)#network 172.16.11.2 0.0.0.0
R15(config-router-af)#network 172.16.11.2 0.0.0.0
R15(config-router-af)#network 172.16.11.21 0.0.0.0
R15(config-router-af)#network 172.16.15.15 0.0.0.0
```

Configuration Verification

By this time, you should have seen log messages showing that your neighbor relationships are up. Let's verify that they are using MD5 encryption. We can do this by debugging EIGRP packets.

R12

```
R12#debug eigrp packets
EIGRP Packet debugging is on
EIGRP: received packet with MD5 authentication, key id = 1
EIGRP: Received HELLO on Et0/0 - paklen 60 nbr 172.16.11.18
EIGRP: received packet with MD5 authentication, key id = 1
EIGRP: Received HELLO on Et1/0 - paklen 60 nbr 172.16.11.6
EIGRP: Sending HELLO on Et1/0 - paklen 60
EIGRP: received packet with MD5 authentication, key id = 1
EIGRP: Received HELLO on Et0/1 - paklen 60 nbr 172.16.11.13
```

Now, let's verify full reachability with another TCL script. The example shows the test from R14, but you will need to verify from all devices.

R14


```
R14#tclsh
R14(tcl)#foreach address {
+>172.16.11.1
+>172.16.11.2
+>172.16.11.5
+>172.16.11.6
+>172.16.11.13
+>172.16.11.14
+>172.16.11.17
+>172.16.11.18
+>172.16.11.21
+>172.16.11.22
+>172.16.11.25
+>172.16.11.26
+>172.16.11.29
+>172.16.11.30
+>172.16.10.10
+>172.16.111.111
+>172.16.12.12
+>172.16.13.13
+>172.16.14.14
+>172.16.15.15
+>} { ping $address
+>}
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

```
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.11.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/6 ms
```

Results Truncated…

IP Routing EIGRP AS 34567

Configuration

As with AS 23456, this autonomous system also requires EIGRP named configuration. Looking at the task, there are not any special configurations.

R16

```
R16(config)#router eigrp NAMED
R16(config-router)#address-family ipv4 autonomous-system 34567
R16(config-router-af)#network 172.16.18.2 0.0.0.0
R16(config-router-af)#network 172.16.18.5 0.0.0.0
R16(config-router-af)#network 172.16.18.21 0.0.0.0
R16(config-router-af)#network 172.16.16.16 0.0.0.0
```

R17

```
R17(config)#router eigrp NAMED
R17(config-router)#address-family ipv4 autonomous-system 34567
R17(config-router-af)#network 172.16.18.18 0.0.0.0
R17(config-router-af)#network 172.16.18.26 0.0.0.0
R17(config-router-af)#network 172.16.18.22 0.0.0.0
R17(config-router-af)#network 172.16.17.17 0.0.0.0
```

R18

```
R18(config)#router eigrp named
R18(config-router)#address-family ipv4 autonomous-system 34567
R18(config-router-af)#network 172.16.18.1 0.0.0.0
R18(config-router-af)#network 172.16.18.9 0.0.0.0
R18(config-router-af)#network 172.16.18.13 0.0.0.0
R18(config-router-af)#network 172.16.188.188 0.0.0.0
```

R19

```
R19(config)#router eigrp NAMED
R19(config-router)#address-family ipv4 autonomous-system 34567
R19(config-router-af)#network 172.16.18.6 0.0.0.0
R19(config-router-af)#network 172.16.18.25 0.0.0.0
R19(config-router-af)#network 172.16.18.29 0.0.0.0
R19(config-router-af)#network 172.16.19.19 0.0.0.0
```

R20

```
R20(config)#router eigrp NAMED
R20(config-router)#address-family ipv4 autonomous-system 34567
R20(config-router-af)#network 172.16.18.14 0.0.0.0
R20(config-router-af)#network 172.16.18.17 0.0.0.0
R20(config-router-af)#network 172.16.18.30 0.0.0.0
R20(config-router-af)#network 172.16.20.20 0.0.0.0
```

Configuration Verification

Let's take a look at the routing table of R20 and verify it is learning all of the routes it should be.

R20

```
R20#sh ip route

Gateway of last resort is not set

      6.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        6.6.6.0/24 is directly connected, Serial2/2
L        6.6.6.20/32 is directly connected, Serial2/2
      172.16.0.0/16 is variably subnetted, 16 subnets, 3 masks
D        172.16.16.0/24
           [90/1536640] via 172.16.18.29, 00:04:14, Ethernet0/0.219
           [90/1536640] via 172.16.18.18, 00:04:14, Ethernet0/0.217
```



```
            [90/1536640] via 172.16.18.13, 00:04:14, Ethernet0/1

D        172.16.17.0/24

            [90/1024640] via 172.16.18.18, 00:04:14, Ethernet0/0.217

D        172.16.18.0/30 [90/1536000] via 172.16.18.13, 00:04:14,
Ethernet0/1

D        172.16.18.4/30

            [90/1536000] via 172.16.18.29, 00:04:14, Ethernet0/0.219

D        172.16.18.8/30

            [90/1536000] via 172.16.18.29, 00:04:14, Ethernet0/0.219

            [90/1536000] via 172.16.18.13, 00:04:14, Ethernet0/1

C        172.16.18.12/30 is directly connected, Ethernet0/1

L        172.16.18.14/32 is directly connected, Ethernet0/1

C        172.16.18.16/30 is directly connected, Ethernet0/0.217

L        172.16.18.17/32 is directly connected, Ethernet0/0.217

D        172.16.18.20/30

            [90/1536000] via 172.16.18.18, 00:04:14, Ethernet0/0.217

D        172.16.18.24/30

            [90/1536000] via 172.16.18.29, 00:04:14, Ethernet0/0.219

            [90/1536000] via 172.16.18.18, 00:04:14, Ethernet0/0.217

C        172.16.18.28/30 is directly connected, Ethernet0/0.219

L        172.16.18.30/32 is directly connected, Ethernet0/0.219

D        172.16.19.0/24

            [90/1024640] via 172.16.18.29, 00:04:05, Ethernet0/0.219

C        172.16.20.0/24 is directly connected, Loopback0

L        172.16.20.20/32 is directly connected, Loopback0
```

Now, let’s do the final verification and test connectivity to all addresses using a TCL script. The example is from R17, but you will need to perform this test from all devices In Germany.

R17

```
R17#tclsh
R17(tcl)#foreach address {
+>172.16.18.1
+>172.16.18.2
+>172.16.18.5
+>172.16.18.6
+>172.16.18.9
+>172.16.18.10
+>172.16.18.13
+>172.16.18.14
+>172.16.18.17
+>172.16.18.18
+>172.16.18.21
+>172.16.18.22
+>172.16.18.25
+>172.16.18.26
+>172.16.18.29
+>172.16.18.30
+>172.16.188.188
+>172.16.16.16
+>172.16.17.17
+>172.16.19.19
+>172.16.20.20
+>} { ping $address
+>}
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.18.1, timeout is 2 seconds:
!!!!!
```



```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.18.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Results Truncated…

IP Routing BGP AS 65611

Configuration

This task is little tricky. The basic BGP configuration is simple, however, you are learning routes from two different VRF VPNs. Unless you read the entire lab as recommended, you would not know this, which is a gotcha every time. We do not want any routes from the DR VRF to leak into the RMT VRF. This would happen if we mutually redistributed EIGRP and BGP without any filtering. The task specifically calls for us not to advertise any BGP learned routes back into BGP. This is relevant due to the VRF configuration. Let's create some filters and apply them to the BGP neighbors.

R11

```
R11(config)#access-list 1 deny 172.16.22.0 0.0.0.255
R11(config)#access-list 1 deny host 172.16.2.2
R11(config)#access-list 1 deny 133.33.2.0 0.0.0.255
R11(config)#access-list 1 permit any
R11(config)#route-map EIGRP->BGP->RMT
R11(config-route-map)#match ip address 1

R11(config)#access-list 2 permit 172.16.11.0 0.0.0.255
R11(config)#access-list 2 permit host 172.16.10.10
R11(config)#access-list 2 permit host 172.16.111.111
R11(config)#access-list 2 permit host 172.16.12.12
R11(config)#access-list 2 permit host 172.16.13.13
R11(config)#access-list 2 permit host 172.16.14.14
R11(config)#access-list 2 permit host 172.16.15.15
R11(config)#access-list 2 deny any
R11(config)#route-map EIGRP->BGP->DR
R11(config-route-map)#match ip address 2
R11(config-router)#router bgp 65611
R11(config-router)#bgp log-neighbor-changes
```



```
R11(config-router)#redistribute eigrp 23456
R11(config-router)#neighbor 133.33.11.1 remote-as 333
R11(config-router)#neighbor 133.33.14.1 remote-as 222
R11(config-router)#nei 133.33.14.1 route-map EIGRP->BGP->DR out
R11(config-router)#nei 133.33.11.1 route-map EIGRP->BGP->RMT out


R11(config)#router eigrp NAMED
R11(config-router)#address-family ipv4 autonomous-system 23456
R11(config-router-af-topology)#redistribute bgp 65611 metric 100 10 10
1 1
```

In the above configuration, we are only allowing 172.16.11.0/24 routes to be advertised to the DR VRF along with the loopback addresses we advertised into EIGRP. Then, we are denying the routes from the DR site from being advertised to the RMT VRF. Is there a better/more efficient way to configure access-list 2?

Configuration Verification

We need to make sure our peerings are up and that we are sending prefixes. We will not be learning prefixes yet as no other BGP peerings have been established up to this point. We also need to verify that the filtering is working properly.

R11

```
R11#sh ip bgp summary
BGP router identifier 172.16.111.111, local AS number 65611
BGP table version is 14, main routing table version 14
13 network entries using 1820 bytes of memory
13 path entries using 1040 bytes of memory
6/6 BGP path/bestpath attribute entries using 864 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 using 3724 total bytes of memory
BGP activity 13/0 prefixes, 13/0 paths, scan interval 60 secs
```

```
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ
Up/Down  State/PfxRcd

133.33.11.1     4          333      31      36       14    0    0
00:24:22        0

133.33.14.1     4          222      26      31       14    0    0

00:20:44        0
```

Now let's take a look at the prefixes that are being advertised to verify our filtering.

R11

```
R11#show bgp ipv4 unicast neigh 133.33.11.1 advertised-routes


     Network          Next Hop            Metric LocPrf Weight Path
 *>  172.16.10.0/24   172.16.11.5        1536640         32768 ?
 *>  172.16.11.0/30   172.16.11.5        2048000         32768 ?
 *>  172.16.11.4/30   0.0.0.0                  0         32768 ?
 *>  172.16.11.12/30  172.16.11.5        1536000         32768 ?
 *>  172.16.11.16/30  172.16.11.5        1536000         32768 ?
 *>  172.16.11.20/30  172.16.11.5        2048000         32768 ?
 *>  172.16.11.24/30  172.16.11.30       1536000         32768 ?
 *>  172.16.11.28/30  0.0.0.0                  0         32768 ?
 *>  172.16.12.0/24   172.16.11.5        1024640         32768 ?
 *>  172.16.13.0/24   172.16.11.5        1536640         32768 ?
 *>  172.16.14.0/24   172.16.11.30       1024640         32768 ?
 *>  172.16.15.0/24   172.16.11.5        2048640         32768 ?
 *>  172.16.111.0/24  0.0.0.0                  0         32768 ?


R11#show bgp ipv4 unicast neigh 133.33.14.1 advertised-routes


     Network          Next Hop            Metric LocPrf Weight Path
```



```
 *>  172.16.11.0/30   172.16.11.5        2048000         32768 ?

 *>  172.16.11.4/30   0.0.0.0                   0         32768 ?

 *>  172.16.11.12/30  172.16.11.5        1536000         32768 ?

 *>  172.16.11.16/30  172.16.11.5        1536000         32768 ?

 *>  172.16.11.20/30  172.16.11.5        2048000         32768 ?

 *>  172.16.11.24/30  172.16.11.30       1536000         32768 ?

 *>  172.16.11.28/30  0.0.0.0                   0         32768 ?


Total number of prefixes 7
```

IP Routing BGP AS 65618

Configuration

Configure BGP as outlined in the task. Also, mutually redistribute EIGRP 34567 and BGP 65618

R18

```
R18(config)#router bgp 65618
R18(config-router)#neighbor 133.33.18.1 remote-as 333
R18(config-router)#redistribute eigrp 34567
R18(config-router)#router eigrp NAMED
R18(config-router)#address-family ipv4 autonomous-system 34567


R18(config)#router eigrp named
R18(config-router)#address-family ipv4 unicast autonomous-system 34567
R18(config-router-af)#topology base
R18(config-router-af-topology)#redistribute bgp 65618 metric 100 10 1 1
1
```

Configuration Verification

We should now see routes coming from the Texas network. Let's take a look at the routing table of R18.

R18

```
R18#show ip route
Gateway of last resort is not set

      133.33.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        133.33.18.0/24 is directly connected, Serial2/0
L        133.33.18.18/32 is directly connected, Serial2/0
      172.16.0.0/16 is variably subnetted, 30 subnets, 3 masks
B        172.16.10.0/24 [20/0] via 133.33.18.1, 00:05:24
B        172.16.11.0/30 [20/0] via 133.33.18.1, 00:05:24
B        172.16.11.4/30 [20/0] via 133.33.18.1, 00:05:24
B        172.16.11.12/30 [20/0] via 133.33.18.1, 00:05:24
B        172.16.11.16/30 [20/0] via 133.33.18.1, 00:05:24
B        172.16.11.20/30 [20/0] via 133.33.18.1, 00:05:24
B        172.16.11.24/30 [20/0] via 133.33.18.1, 00:05:24
B        172.16.11.28/30 [20/0] via 133.33.18.1, 00:05:24
B        172.16.12.0/24 [20/0] via 133.33.18.1, 00:05:24
B        172.16.13.0/24 [20/0] via 133.33.18.1, 00:05:24
B        172.16.14.0/24 [20/0] via 133.33.18.1, 00:05:24
B        172.16.15.0/24 [20/0] via 133.33.18.1, 00:05:24
D        172.16.16.0/24
           [90/1024640] via 172.16.18.2, 01:36:58, Ethernet0/0.816
D        172.16.17.0/24 [90/1536640] via 172.16.18.14, 01:36:50,
Ethernet0/1
                        [90/1536640] via 172.16.18.10, 01:36:50,
Ethernet0/0.819
```



```
                        [90/1536640] via 172.16.18.2, 01:36:50,
Ethernet0/0.816

C        172.16.18.0/30 is directly connected, Ethernet0/0.816

L        172.16.18.1/32 is directly connected, Ethernet0/0.816

D        172.16.18.4/30

           [90/1536000] via 172.16.18.10, 01:36:43, Ethernet0/0.819

           [90/1536000] via 172.16.18.2, 01:36:43, Ethernet0/0.816

C        172.16.18.8/30 is directly connected, Ethernet0/0.819

L        172.16.18.9/32 is directly connected, Ethernet0/0.819

C        172.16.18.12/30 is directly connected, Ethernet0/1

L        172.16.18.13/32 is directly connected, Ethernet0/1

D        172.16.18.16/30 [90/1536000] via 172.16.18.14, 01:36:50,
Ethernet0/1

D        172.16.18.20/30

           [90/1536000] via 172.16.18.2, 01:36:58, Ethernet0/0.816

D        172.16.18.24/30

           [90/1536000] via 172.16.18.10, 01:36:50, Ethernet0/0.819

D        172.16.18.28/30 [90/1536000] via 172.16.18.14, 01:36:43,
Ethernet0/1

                         [90/1536000] via 172.16.18.10, 01:36:43,
Ethernet0/0.819

D        172.16.19.0/24

           [90/1024640] via 172.16.18.10, 01:36:34, Ethernet0/0.819

D        172.16.20.0/24 [90/1024640] via 172.16.18.14, 01:36:43,
Ethernet0/1

B        172.16.111.0/24 [20/0] via 133.33.18.1, 00:05:24

C        172.16.188.0/24 is directly connected, Loopback0

L        172.16.188.188/32 is directly connected, Loopback0
```

Now, let's do a connectivity test between Texas and Germany. We can use a TCL script for this. The example is performed from R10.

R10

```
R10#tclsh
R10(tcl)#foreach address {
+>172.16.11.1
+>172.16.11.2
+>172.16.11.5
+>172.16.11.6
+>172.16.11.13
+>172.16.11.14
+>172.16.11.17
+>172.16.11.18
+>172.16.11.21
+>172.16.11.22
+>172.16.11.25
+>172.16.11.26
+>172.16.11.29
+>172.16.11.30
+>172.16.10.10
+>172.16.111.111
+>172.16.12.12
+>172.16.13.13
+>172.16.14.14
+>172.16.15.15
+>172.16.18.1
+>172.16.18.2
+>172.16.18.5
+>172.16.18.6
+>172.16.18.9
```



```
+>172.16.18.10
+>172.16.18.13
+>172.16.18.14
+>172.16.18.17
+>172.16.18.18
+>172.16.18.21
+>172.16.18.22
+>172.16.18.25
+>172.16.18.26
+>172.16.18.29
+>172.16.18.30
+>172.16.188.188
+>172.16.16.16
+>172.16.17.17
+>172.16.19.19
+>172.16.20.20
+>} { ping $address
+>}
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/5 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Results Truncated.

IP Routing BGP AS 65602

Configuration

Configure BGP as outlined in the task. We need to redistribute connected interfaces but only Loopback 22. We can do this with a route-map as follows. The filtering called out in the last item of the task was done under the BGP AS 65611 section.

R2

```
R2(config)#router bgp 65602
R2(config-router)#neighbor 133.33.2.1 remote-as 222
R2(config-router)#redistribute connected route-map LO22


R2(config)#route-map LO22
R2(config-route-map)#match interface Loopback 22
```

Configuration Verification

Let's verify that R2 is only learning the routes from Texas, and nowhere else by looking at the route table of R2.

R2

```
R2#sh ip route
Gateway of last resort is not set


      133.33.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        133.33.2.0/24 is directly connected, Serial3/3
L        133.33.2.22/32 is directly connected, Serial3/3
      172.16.0.0/16 is variably subnetted, 11 subnets, 3 masks
C        172.16.2.0/24 is directly connected, Loopback0
L        172.16.2.2/32 is directly connected, Loopback0
B        172.16.11.0/30 [20/0] via 133.33.2.1, 00:14:17
B        172.16.11.4/30 [20/0] via 133.33.2.1, 00:14:17
```



```
B        172.16.11.12/30 [20/0] via 133.33.2.1, 00:14:17
B        172.16.11.16/30 [20/0] via 133.33.2.1, 00:14:17
B        172.16.11.20/30 [20/0] via 133.33.2.1, 00:14:17
B        172.16.11.24/30 [20/0] via 133.33.2.1, 00:14:17
B        172.16.11.28/30 [20/0] via 133.33.2.1, 00:14:17
B        172.16.10.10/32 [20/0] via 133.33.2.1, 00:14:17
B        172.16.111.0/24 [20/0] via 133.33.2.1, 00:14:17
B        172.16.12.0/24 [20/0] via 133.33.2.1, 00:14:17
B        172.16.13.0/24 [20/0] via 133.33.2.1, 00:14:17
B        172.16.14.0/24 [20/0] via 133.33.2.1, 00:14:17
B        172.16.15.0/24 [20/0] via 133.33.2.1, 00:14:17
C        172.16.22.0/24 is directly connected, Loopback22
L        172.16.22.22/32 is directly connected, Loopback22
```

Notice, there are no 172.16.18.X routes in the table. One last check, just to be sure, lets run a TCL script to verify we can ping all devices in Texas from R2.

R2

```
R2#tclsh
R2(tcl)#foreach address {
+>(tcl)#172.16.11.1
+>(tcl)#172.16.11.2
+>(tcl)#172.16.11.5
+>(tcl)#172.16.11.6
+>(tcl)#172.16.11.13
+>(tcl)#172.16.11.14
+>(tcl)#172.16.11.17
+>(tcl)#172.16.11.18
+>(tcl)#172.16.11.21
+>(tcl)#172.16.11.22
```

```
+>(tcl)#172.16.11.25
+>(tcl)#172.16.11.26
+>(tcl)#172.16.11.29
+>(tcl)#172.16.11.30
+>(tcl)#172.16.10.10
+>(tcl)#172.16.111.111
+>(tcl)#172.16.12.12
+>(tcl)#172.16.13.13
+>(tcl)#172.16.14.14
+>(tcl)#172.16.15.15
+>(tcl)#} { ping $address source loopback22
+>(tcl)#}
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.22.22
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 14/16/18 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.2, timeout is 2 seconds:
Packet sent with a source address of 172.16.22.22
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 15/16/18 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.5, timeout is 2 seconds:
Packet sent with a source address of 172.16.22.22
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 14/16/21 ms
```

Results truncated…

IP Routing BGP AS 65621, 65222, 62623

Configuration

We need to configure R21, R22, and R23 with BGP according to the task. No redistribution is required.

R21

```
R21(config)#router bgp 65621
R21(config-router)#neighbor 133.33.21.1 remote-as 1010
```

R22

```
R22(config)#router bgp 65622
R22(config-router)#neighbor 133.33.22.1 remote-as 1010
```

R23

```
R23(config)#router bgp 65623
R23(config-router)#neighbor 133.33.23.1 remote-as 1010
```

Configuration Verification

We should now be learning routes on R21, R22, and R23 from BGP. But only the routes that are directly connected to the Local Service Provider.

R21

```
R21#sh ip route

Gateway of last resort is not set

      133.10.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        133.10.21.0/24 is directly connected, Loopback2121
L        133.10.21.21/32 is directly connected, Loopback2121
      133.33.0.0/16 is variably subnetted, 5 subnets, 2 masks
B        133.33.5.0/24 [20/0] via 133.33.21.1, 00:01:16
```

```
C        133.33.21.0/24 is directly connected, Serial2/0
L        133.33.21.21/32 is directly connected, Serial2/0
B        133.33.22.0/24 [20/0] via 133.33.21.1, 00:01:16
B        133.33.23.0/24 [20/0] via 133.33.21.1, 00:01:16
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.21.0/24 is directly connected, Loopback0
L        172.16.21.21/32 is directly connected, Loopback0
```


IP Routing Multicast

Configuration

Let's first enable multicast sparse-dense-mode on R1, R3, R5, and SW2. We also need to add the specified interfaces to the IGMP group 224.5.5.5.

R1

```
R1(config)#ip multicast-routing
R1(config)#ip pim rp-address 131.10.45.2
R1(config)#int e0/0
R1(config-if)#ip pim sparse-dense-mode
R1(config-if)#ip igmp join-group 224.5.5.5
R1(config)#int e0/1.103
R1(config-if)#ip pim sparse-dense-mode
R1(config)#int e0/1.105
R1(config-if)#ip pim sparse-dense-mode
```

R3

```
R3(config)#ip multicast-routing
R3(config)#ip pim rp-address 131.10.45.2
R3(config)#interface e0/0
R3(config-if)#ip pim sparse-dense-mode
R3(config-if)#ip igmp join-group 224.5.5.5
```

R5

```
R5(config)#ip multicast-routing
R5(config)#ip pim rp-address 131.10.45.2
R5(config)#interface e0/0
R5(config-if)#ip pim sparse-dense-mode
R5(config-if)#ip igmp join-group 224.5.5.5
```

SW1

```
SW1(config)#ip multicast-routing
SW1(config)#ip pim rp-address 131.10.45.2
SW1(config)#interface vlan 910
SW1(config-if)#ip pim sparse-dense-mode
SW1(config-if)#int vlan 101
SW1(config-if)#ip pim sparse-dense-mode
```

Configuration Verification

- From SW1, ping the multicast group and verify you get responses from R1, R3, and R5.

SW1

```
SW1# ping 224.5.5.5
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.5.5.5, timeout is 2 seconds:

Reply to request 0 from 131.10.45.18, 10 ms
Reply to request 0 from 131.10.45.2, 22 ms
Reply to request 0 from 131.10.45.34, 22 ms
```

Results Truncated…

IP Routing IPv6

Configuration

The first step for this task is to configure the addressing. We will then move on to configure the routing processes.

SW1

```
SW1(config)#ipv6 unicast-routing
SW1(config)#int vlan 101
SW1(config-if)#ipv6 address 2001:101::10/64
SW1(config-if)#int vlan 104
SW1(config-if)# ipv6 address 2001:104::10/64
SW1(config-if)#int vlan 910
SW1(config-if)#ipv6 address 2001:209::10/64
```

R1

```
R1(config)#ipv6 unicast-routing
R1(config)#int e0/0
R1(config-if)#ipv6 address 2001:101::1/64
R1(config-if)#int e0/1.103
R1(config-if)#ipv6 address 2001:103::1/64
R1(config-if)#int e0/1.105
R1(config-if)#ipv6 address 2001:105::1/64
```

R3

```
R3(config)#ipv6 unicast-routing
R3(config-if)#int e0/0
R3(config-if)#ipv6 address 2001:93::3/64
R3(config)#int e0/1
R3(config-if)#ipv6 address 2001:103::3/64
```

R4

```
R4(config)#ipv6 unicast-routing
R4(config)#int e0/0
R4(config-if)#ipv6 address 2001:104::4/64
R4(config-if)#int e0/1
R4(config-if)#ipv6 address 2001:45::4/64
```

R5

```
R5(config)#ipv6 unicast-routing
R5(config)#int e0/0
R5(config-if)#ipv6 address 2001:105::5/64
R5(config-if)#int e0/1
R5(config-if)#ipv6 address 2001:45::5/64
```

R9

```
R9(config)#ipv6 unicast-routing
R9(config)#int e0/0
R9(config-if)#ipv6 address 2001:209::9/64
R9(config-if)#int e0/1
R9(config-if)#ipv6 address 2001:93::9/64
```

Second Step to this task is to configure EIGRP AS66. With IPv6, this is easily doable by enabling the global routing process and then issuing 1 command under the interface.

R1

```
R1(config)#ipv6 router eigrp 66
R1(config)#int e0/1.103
R1(config-if)#ipv6 eigrp 66
R1(config)#int e0/1.105
R1(config-if)#ipv6 eigrp 66
```

R3


```
R3(config)#ipv6 router eigrp 66
R3(config)#int e0/1
R3(config-if)#ipv6 eigrp 66
```

R5

```
R5(config)#ipv6 router eigrp 66
R5(config)#int e0/0
R5(config-if)#ipv6 eigrp 66
```

Next step is to configure OSPFv3 Area0 and Area 2. The process is very similar to the EIGRP process, however, you do not need to enable it globally. Once OSPFv3 is enabled on an interface, the router enables the process globally.

R1

```
R1(config)#int e0/0
R1(config-if)#ipv6 ospf 1 area 0
```

R3

```
R3(config)#int e0/0
R3(config-if)#ipv6 ospf 1 area 0
```

R4

```
R4(config)#int e0/0
R4(config-if)#ipv6 ospf 1 area 2
R4(config-if)#int e0/1
R4(config-if)#ipv6 ospf 1 area 2
```

R9

```
R9(config)#int e0/0
R9(config-if)#ipv6 ospf 1 area 0
R9(config-if)#int e0/1
R9(config-if)#ipv6 ospf 1 area 0
```

SW1

```
SW1(config-if)#int vlan 101
SW1(config-if)#ipv6 ospf 1 area 0
SW1(config-if)#int vlan 104
SW1(config-if)#ipv6 ospf 1 area 2
SW1(config-if)#int vlan 910
SW1(config-if)#ipv6 ospf 1 area 0
```

Finally, the task asks us to mutually redistribute EIGRP and OSPFv3 on R1. With IPv6, you need to specify if you want to include connected subnets with your redistribution. Also, like with IPv4 EIGRP, you have to assign a metric when redistributing any routing protocol into EIGRP.

R1

```
R1(config)#ipv6 router ospf 1
R1(config-rtr)#redistribute eigrp 66 include-connected
R1(config-rtr)#exit
R1(config)#ipv6 router eigrp 66
R1(config-rtr)#redistribute ospf 1 include-connected metric 10 10 1 1 1
```

Configuration Verification

We can verify the requirements of this task by first looking at the routing table of R9. It should hold OSPF inter-area routes, intra-area routes, and external routes that were redistributed from EIGRP.

R9

```
R9#sh ipv6 route
OI  2001:45::/64 [110/21]
     via FE80::A8BB:CCFF:FE80:6500, Ethernet0/0
C   2001:93::/64 [0/0]
     via Ethernet0/1, directly connected
L   2001:93::9/128 [0/0]
     via Ethernet0/1, receive
O   2001:101::/64 [110/11]
```



```
     via FE80::A8BB:CCFF:FE80:6500, Ethernet0/0
OI  2001:104::/64 [110/11]
     via FE80::A8BB:CCFF:FE80:6500, Ethernet0/0
OE2 2001:201::/64 [110/20]
     via FE80::A8BB:CCFF:FE80:6500, Ethernet0/0
OE2 2001:205::/64 [110/20]
     via FE80::A8BB:CCFF:FE80:6500, Ethernet0/0
C   2001:209::/64 [0/0]
     via Ethernet0/0, directly connected
L   2001:209::9/128 [0/0]
     via Ethernet0/0, receive
OE2 2001:320::/64 [110/20]
     via FE80::A8BB:CCFF:FE80:6500, Ethernet0/0
L   FF00::/8 [0/0]
     via Null0, receive
```

We do indeed see each of these routes. Now, let's verify full connectivity of the IPv6 topology using a… wait, you guessed it, a TCL script!

R9

```
R9#tclsh
R9(tcl)#foreach address {
+>(tcl)#2001:101::1
+>(tcl)#2001:101::10
+>(tcl)#2001:93::9
+>(tcl)#2001:93::3
+>(tcl)#2001:45::4
+>(tcl)#2001:45::5
+>(tcl)#2001:105::1
+>(tcl)#2001:105::5
```

```
+>(tcl)#2001:209::9
+>(tcl)#2001:209::10
+>(tcl)#2001:320::3
+>(tcl)#2001:104::4
+>(tcl)#2001:104::10
+>(tcl)#2001:103::1
+>(tcl)#2001:103::3
+>(tcl)#} { ping $address
+>(tcl)#}
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:101::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 2001:201::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms.
```

Results Truncated…

IPv4 VPN MPLS VPN

Configuration

There is a lot going on in this task. Not to mention, it requires the configuration of the next task as well to work properly. Let's take it step by step. First, let's enable MPLS throughout the MPLS Core network.

R1

```
R3(config)#mpls ip
R3(config)#int e0/1.103
R5(config-if)#mpls ip
R3(config)#int e0/1.105
R5(config-if)#mpls ip
```

R3

```
R3(config)#mpls ip
R3(config)#int e0/1
R5(config-if)#mpls ip
```

R3

```
R3(config)#mpls ip
R3(config)#int e0/0
R5(config-if)#mpls ip
```

Next, let's configure the VRF's that are to be used for the MPLS VPN's on each router. The task does not specifically call this out, but this is a requirement to get the MPLS VPN's to work correctly. The VRF's are listed in the BGP diagram.

R1, R3, R5

```
(config)#ip vrf DR
(config-vrf)#rd 2222:6
(config-vrf)#route-target both 2222:6
(config-vrf)#ip vrf RMT
```

```
(config-vrf)#rd 2222:4
(config-vrf)#route-target both 2222:4
```

Configuration Verification

At this point, the only thing we can verify is the LDP relationships. We can glean the status of LDP by looking at R1 which should have a peering with all the MPLS routers. We expect 2 LDP neighbors.

R1

```
R1#sh mpls ldp neighbor
    Peer LDP Ident: 172.16.3.3:0; Local LDP Ident 172.16.1.1:0
        TCP connection: 172.16.3.3.36089 - 172.16.1.1.646
        State: Oper; Msgs sent/rcvd: 422/417; Downstream
        Up time: 05:49:37
        LDP discovery sources:
          Targeted Hello 172.16.1.1 -> 172.16.3.3, passive
          Ethernet0/1.103, Src IP addr: 131.10.45.38
        Addresses bound to peer LDP Ident:
          172.16.3.3      131.10.45.2     131.10.45.21    131.10.45.38
    Peer LDP Ident: 172.16.5.5:0; Local LDP Ident 172.16.1.1:0
        TCP connection: 172.16.5.5.20319 - 172.16.1.1.646
        State: Oper; Msgs sent/rcvd: 383/388; Downstream
        Up time: 05:24:53
        LDP discovery sources:
          Ethernet0/1.105, Src IP addr: 131.10.45.34
          Targeted Hello 172.16.1.1 -> 172.16.5.5, passive
        Addresses bound to peer LDP Ident:
          172.16.5.5      131.10.45.34    131.10.45.30
```

We will do the full verification of MPLS VRF VPN's after the next section.

IPv4 VPN BGP California HQ

Configuration

The first thing we should do for this task is enable MPLS on all links and configure basic BGP between the peers inside California HQ. Remember that we need to use the Loopback 0 address on the routers to form the peers. eBGP requires neighbors to be directly connected. One way around this is to use the eBGP multi-hop feature. Also remember that BGP IPv4 should be disabled by default. This task goes hand in hand with the above task.

R1

```
R1(config-if)#router bgp 65601
R1(config-router)#no bgp default ipv4-unicast
R1(config-router)#neigh 172.16.3.3 remote-as 65603
R1(config-router)#neigh 172.16.3.3 update-source loopback 0
R1(config-router)#neigh 172.16.3.3 ebgp-multihop
R1(config-router)#neigh 172.16.5.5 remote-as 65605
R1(config-router)#neigh 172.16.5.5 ebgp-multihop
R1(config-router)#neigh 172.16.5.5 update-source loopback 0

R1(config-router-af)#address-family vpnv4
R1(config-router-af)#nei 172.16.1.3 activate
R1(config-router-af)#nei 172.16.1.3 send-community ext
R1(config-router-af)#nei 172.16.1.5 activate
R1(config-router-af)#nei 172.16.1.5 send-community ext
```

R3 and R5

```
(config)#router bgp 6560X (this will be different for each Router)
(config-router)#no bgp default ipv4-unicast
(config-router)#neighbor 172.16.1.1 remote-as 65601
(config-router)#neighbor 172.16.1.1 update-source loopback 0
(config-router)#neighbor 172.16.1.1 ebgp-multihop
```

```
(config-router-af)#address-family vpnv4
(config-router-af)#nei 172.16.1.1 activate
(config-router-af)#nei 172.16.1.1 send-community ext
```

Let's go ahead and peer R3 with AS111. We can use the directly connected serial link instead of the Loopback 0 address. We also need to configure the serial interface of R3 in VRF RMT.

R3

```
R3(config)#int s2/2
R3(config-int)#ip vrf forwarding RMT
R3(config-int)#ip address 133.33.3.3 255.255.255.0

R3(config)#router bgp 65603
R3(config-router)#address-family ipv4 vrf RMT
R3(config-router-af)#neigh 133.33.3.1 remote-as 111
```

Last part of this task is to enable the peer relationship between R5 and AS1010. We also need to configure the serial interface of R5 in VRF RMT

R5

```
R5(config)#int s2/0
R5(config-int)#ip vrf forwarding RMT
R5(config-int)#ip address 133.33.5.5 255.255.255.0

R5(config-vrf)#router bgp 65605
R5(config-router)#address-family ipv4 vrf RMT
R5(config-router-af)#neigh 133.33.5.1 remote-as 1010
```

Configuration Verification

Let's take a look at the routing table of R1 on VRF RMT. Since this is a P router, this will tell us whether or not the MPLS VPN configuration is working. We should expect to see routes from Texas, and Germany. We should also see the routes for the Remote Offices Serial interfaces. All of these routes should be learned from R1.

R9

```
R9#sh ip route vrf RMT
Gateway of last resort is not set

      133.33.0.0/24 is subnetted, 4 subnets
B        133.33.5.0 [20/0] via 172.16.1.1, 00:01:11
B        133.33.21.0 [20/0] via 172.16.1.1, 00:01:11
B        133.33.22.0 [20/0] via 172.16.1.1, 00:01:11
B        133.33.23.0 [20/0] via 172.16.1.1, 00:01:11
      172.16.0.0/16 is variably subnetted, 26 subnets, 2 masks
B        172.16.10.0/24 [20/0] via 172.16.1.1, 00:01:11
B        172.16.11.0/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.11.4/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.11.12/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.11.16/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.11.20/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.11.24/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.11.28/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.12.0/24 [20/0] via 172.16.1.1, 00:01:11
B        172.16.13.0/24 [20/0] via 172.16.1.1, 00:01:11
B        172.16.14.0/24 [20/0] via 172.16.1.1, 00:01:11
B        172.16.15.0/24 [20/0] via 172.16.1.1, 00:01:11
B        172.16.16.0/24 [20/0] via 172.16.1.1, 00:01:11
B        172.16.17.0/24 [20/0] via 172.16.1.1, 00:01:11
B        172.16.18.0/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.18.4/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.18.8/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.18.12/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.18.16/30 [20/0] via 172.16.1.1, 00:01:11
```

```
B        172.16.18.20/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.18.24/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.18.28/30 [20/0] via 172.16.1.1, 00:01:11
B        172.16.19.0/24 [20/0] via 172.16.1.1, 00:01:11
B        172.16.20.0/24 [20/0] via 172.16.1.1, 00:01:11
B        172.16.111.0/24 [20/0] via 172.16.1.1, 00:01:11
B        172.16.188.0/24 [20/0] via 172.16.1.1, 00:01:11
```


IPv4 VPN DMVPN

Configuration

Again, there is a lot going here. The first step is to configure R5 as the DMVPN hub. Then, we need to configure R21, R22, and R23 as DMVPN spokes. Then, we need to enable EIGRP using named mode between the 4 routers.

Let's start by configuring R5 as the DMVPN hub. We need to consider that the Tunnel interface is going to be assigned to VRF RMT. This will affect how we configure the interface. Notice the command highlighted below.

R5

```
R5(config-vrf)#interface Tunnel555
R5(config-if)#ip vrf forwarding RMT
R5(config-if)#ip address 5.5.5.5 255.255.255.0
R5(config-if)#no ip redirects
R5(config-if)#ip nhrp map multicast dynamic
R5(config-if)#ip nhrp network-id 5
R5(config-if)#tunnel source 133.33.5.5
R5(config-if)#tunnel mode gre multipoint
R5(config-if)#tunnel vrf RMT
```

Now let's configure the DMVPN spokes.

R21

```
R21(config)#interface Tunnel555
R21(config-if)# ip address 5.5.5.21 255.255.255.0
R21(config-if)# no ip redirects
R21(config-if)# ip nhrp map 5.5.5.5 133.33.5.5
R21(config-if)# ip nhrp map multicast 133.33.5.5
R21(config-if)# ip nhrp network-id 5
R21(config-if)# ip nhrp nhs 5.5.5.5
```

```
R21(config-if)# tunnel source 133.33.21.21
R21(config-if)# tunnel mode gre multipoint
R21(config-if)#no shut
```

R22

```
R22(config)#interface Tunnel555
R22(config-if)# ip address 5.5.5.22 255.255.255.0
R22(config-if)# no ip redirects
R22(config-if)# ip nhrp map multicast dynamic
R22(config-if)# ip nhrp map 5.5.5.5 133.33.5.5
R22(config-if)# ip nhrp map multicast 133.33.5.5
R22(config-if)# ip nhrp network-id 1
R22(config-if)# ip nhrp nhs 5.5.5.5
R22(config-if)# tunnel source Serial2/0
R22(config-if)# tunnel mode gre multipoint
R22(config-if)#no shut
```

R23

```
R23(config)#interface Tunnel555
R23(config-if)# ip address 5.5.5.23 255.255.255.0
R23(config-if)# no ip redirects
R23(config-if)# ip nhrp map multicast dynamic
R23(config-if)# ip nhrp map 5.5.5.5 133.33.5.5
R23(config-if)# ip nhrp map multicast 133.33.5.5
R23(config-if)# ip nhrp network-id 1
R23(config-if)# ip nhrp nhs 5.5.5.5
R23(config-if)# tunnel source Serial2/0
R23(config-if)# tunnel mode gre multipoint
R23(config-if)#no shut
```


The next requirement is to set the MTU on each tunnel interface to allow for VPN overhead. The default MTU is 1500 and the typical VPN takes less than 8 bits, so let's set it to 1492.

R5, R21, R22, R23

```
(config)#int tunnel 555
(config-if)#ip mtu 1492
```

Now, let's setup EIGRP named mode according to the task requirements.

R5

```
R5(config)#router eigrp NAMED
R5(config-router)#address-family ipv4 vrf RMT autonomous-system 555
R5(config-router-af)#network 5.5.5.0 0.0.0.255
```

R21, R22, R23

```
(config)#router eigrp NAMED
(config-router)#address-family ipv4 autonomous-system 555
(config-router-af)#network 5.5.5.0 0.0.0.255
(config-router-af)#network 133.10.0.0 0.0.255.255
```

Oh, almost forgot! Since R5 has multiple neighbors connected to the same interface, we need to disable Split-Horizon.

R5

```
R5(config)#router eigrp NAMED
R5(config-router)#address-family ipv4 unicast vrf RMT autonomous-system
555
R5(config-router-af)#af-interface tunnel 555
R5(config-router-af-interface)#no split-horizon
```

Last step is to perform mutual redistribution between EIGRP and BGP on R5 within VRF RMT.

R5

```
R5(config)#router bgp 65605
R5(config-router)#address-family ipv4 vrf RMT
```

```
R5(config-router-af)#redistribute eigrp 555


R5(config-router-af)#router eigrp NAMED
R5(config-router)#address-family ipv4 unicast vrf RMT autonomous-system
555
R5(config-router-af)#topology base
R5(config-router-af-topology)#redistribute bgp 65605 metric 10 1 1 1 1
```

Configuration Verification

OK, this is our final reachability test. And of course, we are going to use our favorite testing mechanism, a TCL script to accomplish this. The below test is ran from R21. You need to run this same script from R22 and R23. Let's also run this TCL script from all devices in Texas and Germany. Just change the source address in the ping at the bottom of the script.

R21

```
R21#tclsh
R21(tcl)#foreach address {
+>172.16.11.1
+>172.16.11.2
+>172.16.11.5
+>172.16.11.6
+>172.16.11.13
+>172.16.11.14
+>172.16.11.17
+>172.16.11.18
+>172.16.11.29
+>172.16.11.30
+>172.16.11.33
+>172.16.11.34
+>172.16.11.37
```



```
+>172.16.11.38
+>172.16.10.10
+>172.16.111.111
+>172.16.12.12
+>172.16.13.13
+>172.16.14.14
+>172.16.15.15
+>172.16.18.1
+>172.16.18.2
+>172.16.18.5
+>172.16.18.6
+>172.16.18.9
+>172.16.18.10
+>172.16.18.13
+>172.16.18.14
+>172.16.18.17
+>172.16.18.18
+>172.16.18.21
+>172.16.18.22
+>172.16.18.25
+>172.16.18.26
+>172.16.18.29
+>172.16.18.30
+>172.16.188.188
+>172.16.16.16
+>172.16.17.17
+>172.16.19.19
+>172.16.20.20
+>131.10.21.21
```

```
+>131.10.22.22
+>131.10.23.23
+>172.16.10.10
+>172.16.111.111
+>131.10.12.12
+>131.10.13.13
+>131.10.14.14
+>131.10.15.15
+>172.16.16.16
+>172.16.17.17
+>172.16.18.18
+>172.16.19.19
+>172.16.20.20
+>131.10.21.21
+>131.10.22.22
+>131.10.23.23
+>5.5.5.5
+>5.5.5.21
+>5.5.5.22
+>5.5.5.23
+>} { ping $address source loopback2121
+>}
```


IP Security Access Control

Configuration

We can glean a few key things from the task requirements. First, we will need to enable HTTP server on R16 and force it to use port 8000. Second, we will need to setup an access-list and apply it to the HTTP configuration. Third, we will need to setup an ACL on E0/0 to block this HTTP traffic. Let's start by enabling HTTP on port 8000.

R16

```
R16(config)#ip http server
R16(config)#ip http port 8000
```

Now, let's configure the access-list and apply it to the HTTP configuration.

R16

```
R16(config)#access-list 80 permit 172.16.18.14
R16(config)#access-list 80 permit 172.16.18.13
R16(config)#ip http access-class 80
```

The next requirement asks us to only allow this traffic through interface E0/1 and it should be blocked on all other interfaces.

R16

```
R16(config)#ip access-list extended HTTP
R16(config-ext-nacl)#permit tcp 172.16.18.0 0.0.0.252 host 172.16.18.21
eq 8000
R16(config-ext-nacl)# deny   tcp any any eq 8000
R16(config-ext-nacl)# permit ip any any
R16(config)#interface Ethernet0/0.816
R16(config-subif)#ip access-group HTTP in
R16(config-subif)#interface Ethernet0/0.619
R16(config-subif)#ip access-group HTTP in
```

Telnet from the e0/1 interface of R20 on port 8000 on R16 to verify connectivity.

Configuration Verification

According to the task we need to telnet from e0/1 interface of R20 on port 8000 to R16. Let's observe.

R20

```
R20#telnet 172.16.18.21 8000 /source-interface e0/1
Trying 172.16.18.21, 8000 ... Open
```


IP Security Unicast Source Verification

Configuration

The task is asking us to configure unicast source verification on R12's connection to R13. The key words in the task are: "Do not use any prefix-based filtering techniques".

R12

```
R12(config)#int e0/0
R12(config-if)#ip verify unicast source reachable-via any
```

Configuration Verification

There is no verification for this task.

IP Services Quality of Service

Configuration

Right up front, we can see that the QoS policy needs to be applied outbound on the serial link of R18. Packets sourced from 172.16.18.x needs to be marked with DSCP 2 and given 30% of the bandwidth in the priority queue.

R18

```
R18(config)#class-map match-all DSCP2
R18(config-cmap)# match access-group 2

R18(config-cmap)#policy-map DSCP2-POLICY
R18(config-pmap)#class DSCP2
R18(config-pmap-c)#priority percent 30

R18(config)#access-list 2 permit 172.16.18.0 0.0.0.255

R18(config)#int s2/0
R18(config-if)# service-policy output DSCP2-POLICY
```

Configuration Verification

We can source a ping from R18's E0/1 interface destined for 172.16.11.1 and look at the policy-map to verify the packets are going through the policy. This will also verify that those packets are also going into the priority queue and with what bandwidth.

R18

```
R18#show policy-map interface
 Serial2/0

  Service-policy output: DSCP2-POLICY
```



```
          queue stats for all priority classes:
            Queueing
            queue limit 64 packets
            (queue depth/total drops/no-buffer drops) 0/0/0
            (pkts output/bytes output) 5/520

          Class-map: DSCP2 (match-all)
            5 packets, 520 bytes
            5 minute offered rate 0000 bps, drop rate 0000 bps
            Match: access-group 2
            Priority: 30% (463 kbps), burst bytes 11550, b/w exceed drops: 0
```

IP Services Address Administration

Configuration

Configure a DHCP server on SW1 using the parameters specified in the task.

SW1

```
SW1(config)#ip dhcp excluded-address 2.2.2.1 2.2.2.99
SW1(config)#ip dhcp excluded-address 2.2.2.101 2.2.2.255
SW1(config)#!
SW1(config)#ip dhcp pool DHCP2
SW1(dhcp-config)# network 2.2.2.0 255.255.255.0
SW1(dhcp-config)# dns-server 8.8.8.8
SW1(dhcp-config)# default-router 2.2.2.2
```

Configuration Verification

R2

```
R2#
%DHCP-6-ADDRESS_ASSIGN: Interface Ethernet0/0 assigned DHCP address
2.2.2.100, mask 255.255.255.0, hostname R2


R2#ping 2.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```


You have completed this Mock Lab

If you need assistance with any of this book's content, please visit our Member Community at http://community.ipexpert.com.