Advanced
CCIE SERVICE PROVIDER
v3.0

www.MicronicsTraining.com

Narbik Kocharians
CCIE #12410
R&S, Security, SP

Paul Negron
CCIE #14856
SP

VOL-II

Table of Content:

Advanced
CCIE SERVICE PROVIDER
v3.0

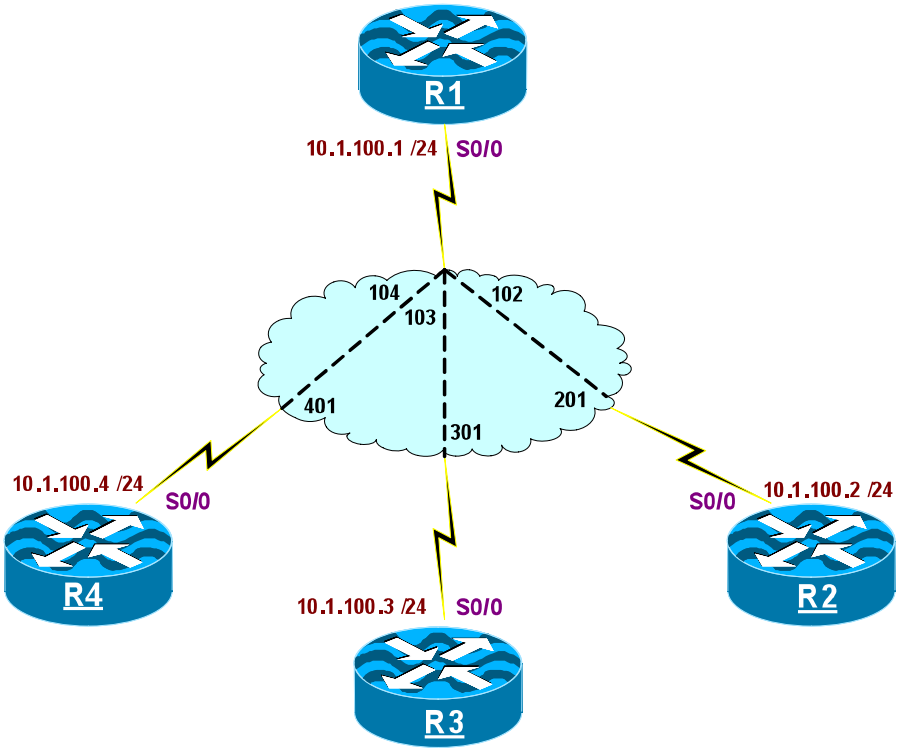
www.MicronicsTraining.com

Narbik Kocharians
CCIE #12410
R&S, Security, SP

Paul Negron
CCIE #14856
SP

Frame-relay

Lab 1 – Hub-n-Spoke using Frame-Relay Map Statements



IP addressing and DLCI information chart:

Routers	IP address	Local DLCI	Connecting to:
R1's Frame-Relay interface S0/0	10.1.100.1 /24	102 103 104	R2 R3 R4
R2's Frame-Relay interface S0/0	**10.1.100.2 /24**	**201**	**R1**
R3's Frame-Relay interface S0/0	10.1.100.3 /24	301	R1
R4's Frame-Relay interface S0/0	**10.1.100.4 /24**	**401**	**R1**

Task 1

Configure a frame-relay hub and spoke using frame-relay map statements. Use the IP addressing in the above chart.
Disable inverse-arp such that the routers do not generate inverse-arp request packets, and ensure that only the assigned DLCIs are used and mapped. These mappings should be as follows:

- **On R1:**: DLCIs 102, 103, and 104 should be mapped to R2, R3, and R4 respectively.
- **On R2:, R3 and R4**: DLCIs 201, 301, and 401 should be used on R2, R3, and R4 respectively for their mapping to R1 (the hub).

In the future, the EIGRP routing protocol will be configured on these routers. Ensure that the routers can handle the multicast traffic generated by the EIGRP routing protocol. **Do not** configure any sub-interface(s) to accomplish this task.

On R1

```
R1(config)#int s0/0
R1(config-if)#ip address 10.1.100.1 255.255.255.0
R1(config-if)#encapsulation frame
R1(config-if)#frame-relay map ip 10.1.100.2 102  broadcast
R1(config-if)#frame-relay map ip 10.1.100.3 103  broadcast
R1(config-if)#frame-relay map ip 10.1.100.4 104  broadcast
R1(config-if)#NO frame-relay inverse-arp
R1(config-if)#NO shut
```

To verify the configuration:

On R1

```
R1#Show frame-relay map

Serial0/0 (up): ip 10.1.100.2 dlci 102(0x66,0x1860), static,
              broadcast,
              CISCO, status defined, inactive
Serial0/0 (up): ip 10.1.100.3 dlci 103(0x67,0x1870), static,
              broadcast,
              CISCO, status defined, inactive
Serial0/0 (up): ip 10.1.100.4 dlci 104(0x68,0x1880), static,
              broadcast,
              CISCO, status defined, inactive
```

Note: You may see DLCIs 105 and 106 mapped to the 0.0.0.0 IP address. These dynamic mappings may not affect Unicast traffic, but they will affect Multicast and/or Broadcast traffic. Therefore, they should be removed from the mapping table. The "Clear frame-relay inarp" command will not have any effect on these entries, whereas, saving the configuration and then reloading the routers will clear the "0.0.0.0" mappings. Another way to clear the "0.0.0.0" mapping is to remove the encapsulation and reconfigure the encapsulation back again, but once the encapsulation is removed, the frame-relay commands configured under the interface are also removed.

The output of the above show command shows that the DLCIs are all in "inactive" status. This means that the problem is on the other side of the VC, in this case, the other end of these VCs are not configured yet. Once they are configured correctly, the status should transition to active state.

Let's configure the spoke routers:

On R2

```
R2(config)#int s0/0
R2(config-if)#ip address 10.1.100.2 255.255.255.0
R2(config-if)#encapsulation frame
R2(config-if)#frame-relay map ip 10.1.100.1 201 broadcast
R2(config-if)#No frame-relay inverse-arp
R2(config-if)#No shut
```

To verify the configuration:

On R2

Let us start with layer one and see if we have a serial cable connected to the Frame-Relay switch. If so, which end of the cable is connected to our router, DTE or DCE?

The output of the following show command shows that the DTE end of the cable is connected to our local router. The "clocks detected" tells us that we are receiving clocking from a DCE device. This should always be the first step in troubleshooting frame-relay. If the output of the following command showed that we have the DCE end of the cable connected to our router, then the local router has to provide clocking. This means that the "clockrate" command must be configured or else the VC will not transition into UP/UP state.

```
R2#Show controller s0/0 | inc clocks

DTE V.35 TX and RX clocks detected.
```

In the next step, we should see if the local router is exchanging LMIs with the frame-relay switch.

Note: Keepalive LMIs are exchanged every 10 seconds, which means that if the frame-relay switch is configured correctly and the LMI types are configured correctly (they match on both ends), then you

should see the number of status enquires sent and received increment every 10 seconds.

```
R2#Show frame-relay lmi | inc Num

   Num Status Enq. Sent 68          Num Status msgs Rcvd 69
   Num Update Status Rcvd 0         Num Status Timeouts 0

R2#Show frame-relay lmi | inc Num

   Num Status Enq. Sent 69          Num Status msgs Rcvd 70
   Num Update Status Rcvd 0         Num Status Timeouts 0
```

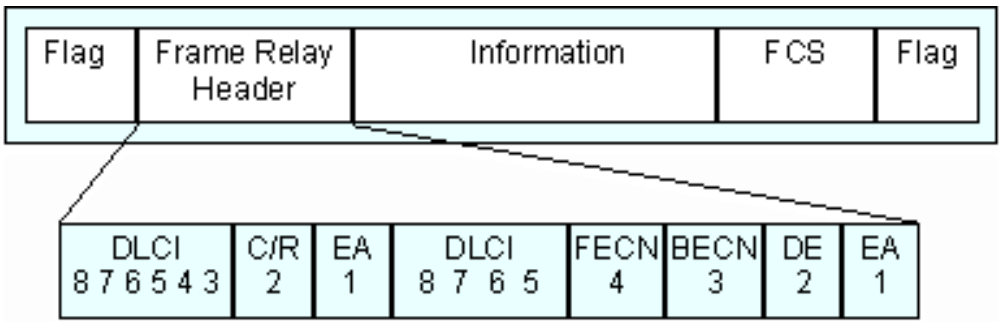
Next the frame-relay maps are checked:

```
R2#Show frame-relay map 201

Serial0/0 (up): ip 10.1.100.1 dlci 201(0xC9,0x3090), static,
                broadcast,
                CISCO, status defined, active
```

Note: The output of the above show command reveals that the remote IP address of 10.1.100.1 is mapped to the local DLCI of 201. Make sure you see the correct IP address.

In the parenthesis, DLCI 201, is presented in hexadecimal and Q922 format. If the hexadecimal value of 0xC9 is converted to decimal, the result is 201, which is the local DLCI number.

The second hexadecimal value of 0x3090, indicates how the DLCI is split into two sections within the Frame-Relay header; a DLCI is a 10 bit digit and the first 6 bits (the most significant 6 bits) are in the first byte. The last 4 bits of the DLCI is found in the beginning of the second byte of the Frame-Relay frame, as follows:



Frame Relay header structure

Notice how the 10 bits are divided? 6 bits are in the first byte and the remaining 4 bits are in the second byte.
If the hex value of 0x3090 is converted to decimal, you will once again see a DLCI value of 201. As follows:

Convert 0x3090 to binary:

3	0	9	0
0011	0 0 0 0	1001	0000

Take the most significant 6 bits, in this case: 001100

Take the most significant 4 bits of the second byte, in this case: 1001

Note: The most significant 6 bits of the first byte and the most significant 4 bits of the second byte are concatenated into a 10 bit value, as follows:

0011001001

If the above binary number is converted to decimal (1 + 8 + 64 + 128), you should get 201.

In the final step, end to end reachability is tested:

```
R2#Ping 10.1.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

Let's configure R3:

On R3

```
R3(config)#int s0/0
R3(config-if)#ip address 10.1.100.3 255.255.255.0
R3(config-if)#encapsulation frame
R3(config-if)#frame-relay map ip 10.1.100.1 301 broadcast
R3(config-if)#NO frame-relay inverse-arp
R3(config-if)#NO shut
```

To verify the configuration:

On R3

```
R3#Ping 10.1.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.1, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms

R3#Show frame map

Serial0/0 (up): ip 10.1.100.1 dlci 301(0x12D,0x48D0), static,
              broadcast,
              CISCO, status defined, active
```

Let's configure R4:

On R4

```
R4(config)#int s0/0
R4(config)#ip address 10.1.100.4 255.255.255.0
R4(config)#encapsulation frame
R4(config)#frame-relay map ip 10.1.100.1 401 broadcast
R4(config)#NO frame-relay inverse-arp
R4(config)#NO shut
```

To verify the configuration:

On R4

```
R4#Ping 10.1.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/50/52 ms

R4#Show frame-relay map

Serial0/0 (up): ip 10.1.100.1 dlci 401(0x191,0x6410), static,
              broadcast,
              CISCO, status defined, active
```

Task 2

Ensure that every router can ping every IP address connected to the cloud. When configuring this task, ensure that the hub router does **not** receive redundant routing traffic.

Note: Every IP address connected to the cloud also includes the local router's IP address. Let's test the existing situation:

On R1

```
R1#Ping 10.1.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

The ping is not successful. Let's enable the "debug frame-relay packet" and try the ping again:

```
R1#Debug frame-relay packet
Frame Relay packet debugging is on

R1#Ping 10.1.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.1, timeout is 2 seconds:

Serial0/0:Encaps failed--no map entry link 7(IP).
Serial0/0:Encaps failed--no map entry link 7(IP).
Serial0/0:Encaps failed--no map entry link 7(IP).
Serial0/0:Encaps failed--no map entry link 7(IP).
Serial0/0:Encaps failed--no map entry link 7(IP).
Success rate is 0 percent (0/5)
```

Let's disable the debug:

On R1

```
R1#U all
```

The output of the above debug states that there is no mapping and encapsulation failed. Because of this, frame-relay could be configured in two different ways: multipoint and point-to-point.
There is only one way to configure frame-relay in a point-to-point manner, and that's through a point-to-point sub-interface configuration. Whereas a multipoint can be configured in two ways:

- **Perform the entire configuration directly under the main interface.**
- **Configure a sub-interface in a multipoint manner.**

Since the entire configuration was performed without the use of sub-interfaces, this is a multipoint interface. In a multipoint frame-relay configuration, two conditions must be met before an IP address

is reachable:

A. **The destination IP address must be in the routing table with a valid next hop.**

B. **There must be a frame-relay mapping for that destination.**

In this case, the destination IP address is in the routing table, but the frame-relay mapping is missing.

When configuring the frame-relay mapping, you can use any active DLCI:

On R1::

```
R1(config)#interface s0/0
R1(config-if)#frame-relay map ip 10.1.100.1 102
```

Note: Since the local router will not be sending multicast or broadcast traffic to itself, there is no need to add the "broadcast" keyword for this configuration.

To verify the configuration:

On R1

```
R1#Ping 10.1.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/101/108 ms
```

Let's test R2's reachability, we already know that it needs a frame-relay map or else it will not be able to ping its own IP address, let's configure one and test:

On R2

```
R2(config)#int s0/0
R2(config-if)#frame-relay map ip 10.1.100.2 201
```

To test the configuration:

On R2

```
R2#Ping 10.1.100.2

Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.1.100.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 96/100/108 ms
```

Let's see if R2 can ping the other spokes:

On R2

```
R2#Ping 10.1.100.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R2#Ping 10.1.100.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.4, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Do we have frame-relay mappings for these destinations? Let's check:

On R2

```
R2#Show frame-relay map

Serial0/0 (up): ip 10.1.100.2 dlci 201(0xC9,0x3090), static,
              CISCO, status defined, active
Serial0/0 (up): ip 10.1.100.1 dlci 201(0xC9,0x3090), static,
              broadcast,
              CISCO, status defined, active
```

Note: There are two frame-relay mappings, one for 10.1.100.2 and the second one is for 10.1.100.1 IP addresses. Let's add two more frame-relay mappings, one for 10.1.100.3 and the second one for 10.1.100.4:

On R2

```
R2(config)#int s0/0
R2(config-if)#frame-relay map ip 10.1.100.3 201
R2(config-if)#frame-relay map ip 10.1.100.4 201
```

There are two points that you need to remember:

a. **The destination IP address must be in the routing table with a valid next hop.**
b. **There must be a frame-relay mapping for that destination.**

To test the configuration:

On R2

```
R2#Ping 10.1.100.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Let's turn on the "debug frame-relay packet" and ping again and see the result:

On R2

```
R2#Deb frame pack
Frame Relay packet debugging is on

R2#Ping 10.1.100.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.3, timeout is 2 seconds:

Serial0/0(o): dlci 201(0x3091), pkt type 0x800(IP), datagramsize 104.
Serial0/0(o): dlci 201(0x3091), pkt type 0x800(IP), datagramsize 104.
Serial0/0(o): dlci 201(0x3091), pkt type 0x800(IP), datagramsize 104.
Serial0/0(o): dlci 201(0x3091), pkt type 0x800(IP), datagramsize 104.
Serial0/0(o): dlci 201(0x3091), pkt type 0x800(IP), datagramsize 104.

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

It seems like the local router (R2) is sending the packets out, let's enable the same debugging on R3 and see the result:

On R2:

```
R2#Ping 10.1.100.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.3, timeout is 2 seconds:
.....
```

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

On R3

```
Serial0/0(i): dlci 301(0x48D1), pkt type 0x800, datagramsize 104
Serial0/0:Encaps failed--no map entry link 7(IP)
```

It looks like R3 is missing a frame-relay map back to R2. Let's configure a frame-relay map on R3 for R2 and test again:

On R3:

```
R3(config)#int s0/0
R3(config-if)#frame-relay map ip 10.1.100.2 301
```

To verify the configuration:

On R2

```
R2#Ping 10.1.100.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/100/100 ms
```

Let's do the same On R4:.

On R4

```
R4(config)#int s0/0
R4(config-if)#frame-relay map ip 10.1.100.2 401
```

To verify the configuration:

On R2

```
R2#Ping 10.1.100.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 96/100/108 ms
```

When configuring the frame-relay mapping from one spoke to another spoke, the "broadcast"

keyword should not be used, if this keyword is used, the hub router will receive redundant routing traffic. This can be verified by running RIPv2 and performing a "debug ip rip" command on the hub router.

Task 3

Configure the routers such that the LMI status inquiries are sent every 5 seconds and full status LMI requests are sent every 3 cycles instead of 6.

By default, frame-relay routers generate LMI status inquiries every 10 seconds, and a full status inquiry every 6th cycle (every 60 seconds). The interval for status inquiries can be changed using the "keepalive" command, whereas, the "frame-relay lmi-n391dte" command can be used to change the interval for the complete status inquiries.

Note: The output of the following debug command reveals the status inquiries and full status inquiries:

On R1:

```
R1#Debug frame lmi

Serial0/0(out): StEnq, myseq 125, yourseen 124, DTE up
datagramstart = 0x3F401ED4, datagramsize = 14
FR encap = 0x00010308
00 75 95 01 01 01 03 02 7D 7C

Serial0/0(in): Status, myseq 125, pak size 14
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 125, myseq 125

Serial0/0(out): StEnq, myseq 126, yourseen 125, DTE up
datagramstart = 0x3F6B0294, datagramsize = 14
FR encap = 0x00010308
407: 00 75 95 01 01 01 03 02 7E 7D

Serial0/0(in): Status, myseq 126, pak size 14
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 126, myseq 126

Serial0/0(out): StEnq, myseq 127, yourseen 126, DTE up
datagramstart = 0x3F400C14, datagramsize = 14
FR encap = 0x00010308
00 75 95 01 01 01 03 02 7F 7E
```

```
Serial0/0(in): Status, myseq 127, pak size 14
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 127, myseq 127

Serial0/0(out): StEnq, myseq 128, yourseen 127, DTE up
datagramstart = 0x3F6AF394, datagramsize = 14
FR encap = 0x00010308
00 75 95 01 01 01 03 02 80 7F
Serial0/0(in): Status, myseq 128, pak size 14
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 128, myseq 128

Serial0/0(out): StEnq, myseq 129, yourseen 128, DTE up
datagramstart = 0x3F644ED4, datagramsize = 14

FR encap = 0x00010308
00 75 95 01 01 01 03 02 81 80
Serial0/0(in): Status, myseq 129, pak size 14
RT IE 1, length 1, type 1
KA IE 3, length 2, yourseq 129, myseq 129

Serial0/0(out): StEnq, myseq 130, yourseen 129, DTE up
datagramstart = 0x3F6B03D4, datagramsize = 14
FR encap = 0x00010308
00 75 95 01 01 00 03 02 82 81

Serial0/0(in): Status, myseq 130, pak size 59
RT IE 1, length 1, type 0
KA IE 3, length 2, yourseq 130, myseq 130

PVC IE 0x7 , length 0x3 , dlci 102, status 0x2
PVC IE 0x7 , length 0x3 , dlci 103, status 0x2
PVC IE 0x7 , length 0x3 , dlci 104, status 0x2
PVC IE 0x7 , length 0x3 , dlci 105, status 0x0
PVC IE 0x7 , length 0x3 , dlci 106, status 0x0
```

Note: The status inquiries are sent every 10 seconds, these messages are "type 1s", whereas, the complete status inquiries are generated by the local router every 6th cycle, these message are "type 0" messages, and when the frame-relay switch receives these messages it responds with all the DLCIs that are configured for that given router.

To change these timers:

On all routers:

```
Rx(config)#interface s0/0
Rx(config-if)#keepalive 5
Rx(config-if)#frame-gelay lmi-n391dte 3
```

To test the configuration:

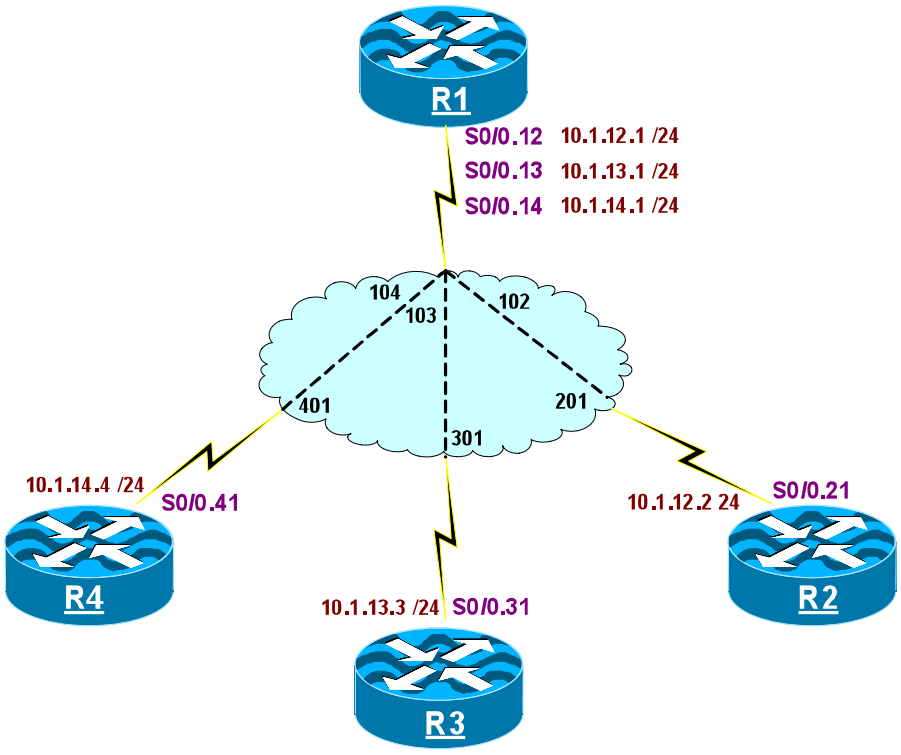
```
Rx#Debug frame lmi

*Nov 24 20:13:52.411: Serial0/0(out): StEnq, myseq 221, yourseen 220, DTE up
*Nov 24 20:13:52.411: datagramstart = 0x3F6AEFD4, datagramsize = 14
*Nov 24 20:13:52.411: FR encap = 0x00010308
*Nov 24 20:13:52.411: 00 75 95 01 01 01 03 02 DD DC

*Nov 24 20:13:52.415: Serial0/0(in): Status, myseq 221, pak size 14
*Nov 24 20:13:52.415: RT IE 1, length 1, type 1
*Nov 24 20:13:52.415: KA IE 3, length 2, yourseq 221, myseq 221

*Nov 24 20:13:57.411: Serial0/0(out): StEnq, myseq 222, yourseen 221, DTE up
*Nov 24 20:13:57.411: datagramstart = 0x3F400D54, datagramsize = 14
*Nov 24 20:13:57.411: FR encap = 0x00010308
*Nov 24 20:13:57.411: 00 75 95 01 01 01 03 02 DE DD

*Nov 24 20:13:57.415: Serial0/0(in): Status, myseq 222, pak size 14
*Nov 24 20:13:57.415: RT IE 1, length 1, type 1
*Nov 24 20:13:57.415: KA IE 3, length 2, yourseq 222, myseq 222

*Nov 24 20:14:02.411: Serial0/0(out): StEnq, myseq 223, yourseen 222, DTE up
*Nov 24 20:14:02.411: datagramstart = 0x3F6AF394, datagramsize = 14
*Nov 24 20:14:02.411: FR encap = 0x00010308
*Nov 24 20:14:02.411: 00 75 95 01 01 00 03 02 DF DE

*Nov 24 20:14:02.423: Serial0/0(in): Status, myseq 223, pak size 59
*Nov 24 20:14:02.423: RT IE 1, length 1, type 0
*Nov 24 20:14:02.423: KA IE 3, length 2, yourseq 223, myseq 223
*Nov 24 20:14:02.423: PVC IE 0x7 , length 0x3 , dlci 102, status 0x2
*Nov 24 20:14:02.423: PVC IE 0x7 , length 0x3 , dlci 103, status 0x2
*Nov 24 20:14:02.423: PVC IE 0x7 , length 0x3 , dlci 104, status 0x2
*Nov 24 20:14:02.423: PVC IE 0x7 , length 0x3 , dlci 105, status 0x0
*Nov 24 20:14:02.423: PVC IE 0x7 , length 0x3 , dlci 106, status 0x0
```

Note: Initially the router and the frame-relay switch exchange two "type 1" inquiries. The third message that the local router generates is a "type 0" message which tells the switch to respond with all the DLCIs.

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 2 – Hub-n-Spoke Using Frame-relay Point-to-Point Configuration



IP addressing and DLCI information Chart:

Routers	IP address	Local DLCI	Connecting to:
R1's Frame-Relay interface	10.1.12.1 /24 10.1.13.1 /24 10.1.14.1 /24	102 103 104	R2 R3 R4
R2's Frame-Relay interface	**10.1.12.2 /24**	**201**	**R1**
R3's Frame-Relay interface	10.1.13.3 /24	301	R1
R4's Frame-Relay interface	**10.1.14.4 /24**	**401**	**R1**

Task 1

Configure the routers in a hub and spoke manner using the IP addressing in the above chart.

These routers should be configured with point-to-point sub-interface/s, and ensure that only the assigned DLCIs are used, these DLCIs should be as follows:

- **On R1::** 102, 103 and 104, should be used for connections to R2, R3, and R4 respectively.
- **On R2: R3 and R4:** DLCIs 201, 301, and 401 should be used on R2, R3 and R4 respectively for their connection to R1 (the hub).

These routers should be able to ping every IP address **within their IP address space**.

On R1:

```
R1(config)#interface s0/0
R1(config-if)#encap frame
R1(config-if)#No shut

R1(config)#interface s0/0.12 point-to-point
R1(config-subif)#ip address 10.1.12.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 102

R1(config-subif)#interface s0/0.13 point-to-point
R1(config-subif)#ip address 10.1.13.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 103

R1(config-subif)#interface s0/0.14 point-to-point
R1(config-subif)#ip address 10.1.14.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 104
```

To verify the configuration:

On R1:

```
R1#Show frame map

Serial0/0.12 (down): point-to-point dlci, dlci 102(0x66,0x1860), broadcast
          status defined, inactive
Serial0/0.13 (down): point-to-point dlci, dlci 103(0x67,0x1870), broadcast
          status defined, inactive
```

```
Serial0/0.14 (down): point-to-point dlci, dlci 104(0x68,0x1880), broadcast
          status defined, inactive
```

Note: when frame-relay is configured in a point-to-point manner it's important to understand the following two behaviors:

A. **There is no need to disable inverse-arp, because inverse-arp is disabled when frame-relay is configured in a point-to-point manner.**

B. **No need for frame-relay mapping/s, because there can only be another router on the other end of the PVC, therefore, all IP addresses (this includes the local router's IP address) are reachable as long as the destination IP address is in the routing table with a valid next hop IP address.**

On R2:

```
R2(config)#int s0/0
R2(config-if)#encap frame
R2(config-if)#No shut

R2(config)#int s0/0.21 point-to-point
R2(config-subif)#ip address 10.1.12.2 255.255.255.0
R2(config-subif)#frame-relay interface-dlci 201
```

To verify and test the configuration:

On R2:

```
R2#Show frame map

Serial0/0.21 (up): point-to-point dlci, dlci 201(0xC9,0x3090), broadcast
          status defined, active

R2#Ping 10.1.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms

R2#Ping 10.1.12.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/114/120 ms
```

Let's configure R3:

On R3:

```
R3(config)#int s0/0
R3(config-if)#encap frame
R3(config-if)#No shut

R3(config)#int s0/0.31 point-to-point
R3(config-subif)#ip address 10.1.13.3 255.255.255.0
R3(config-subif)#frame-relay interface-dlci 301
```

To verify and test the configuration:

On R3:

```
R3#Show frame map

Serial0/0.31 (up): point-to-point dlci, dlci 301(0x12D,0x48D0), broadcast
          status defined, active

R3#Ping 10.1.13.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms

R3#Ping 10.1.13.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/114/120 ms
```

Let's configure R4:

On R4:

```
R4(config)#int s0/0
R4(config-if)#encap frame
R4(config-if)#No shut

R4(config)#int s0/0.41 point-to-point
R4(config-subif)#ip address 10.1.14.4 255.255.255.0
```

```
R4(config-subif)#frame-relay interface-dlci 401
```

To verify and test the configuration:

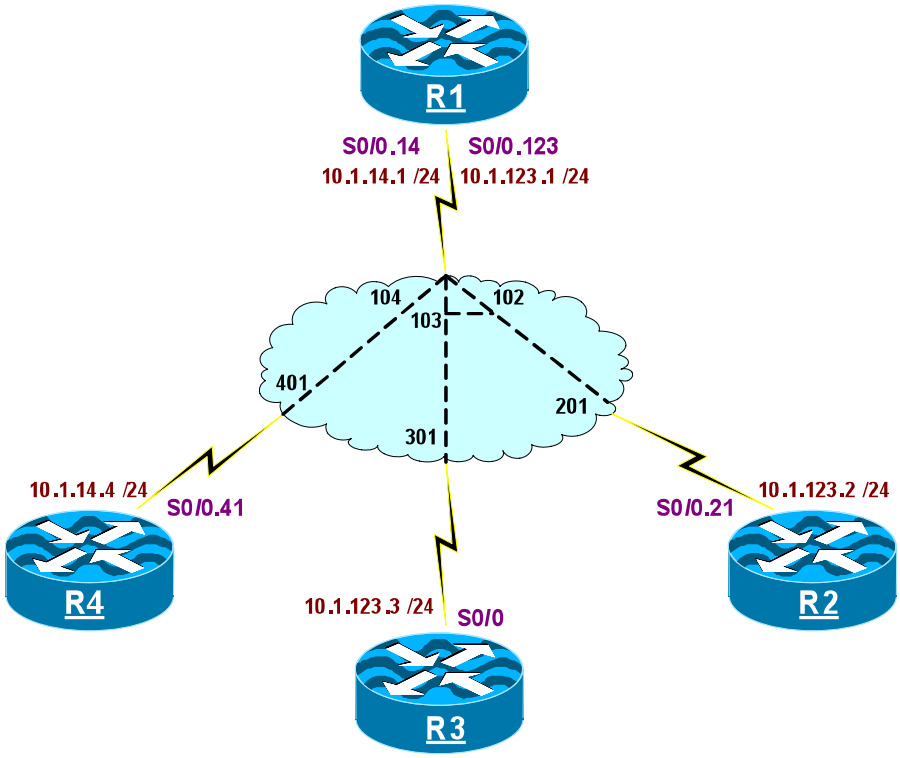
On R4:

```
R4#Show frame map

Serial0/0.41 (up): point-to-point dlci, dlci 401(0x191,0x6410), broadcast
          status defined, active

R4#Ping 10.1.14.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.14.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms

R4#Ping 10.1.14.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.14.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/114/120 ms
```

Task 2

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 3 – Mixture of Point-to-point and Multipoint Frame-Relay



IP addressing and DLCI information Chart:

Routers	IP address	Local DLCI	Connecting to:
R1's Frame-Relay interface	10.1.123.1 /24 10.1.123.1 /24 10.1.14.1 /24	102 103 104	R2 R3 R4
R2's Frame-Relay interface	10.1.123.2 /24	201	R1
R3's Frame-Relay interface	10.1.123.3 /24	301	R1
R4's Frame-Relay interface	10.1.14.4 /24	401	R1

Task 1

Configure frame-relay on the routers as follows:

R1: This router should be configured in a point-to-point manner for its connection to R4 and in a multipoint manner for its connection to R2 and R3. Use the IP addressing and DLCI information in the above chart.

R2: This router should be configured in a point-to-point manner for its connection to R1. Use the IP addressing and DLCI information in the above chart.

R3: This router should be configured using its main interface for its connection to R1. Use the IP addressing and DLCI information in the above chart.

R4: This router must be configured in a point-to-point manner for its connection to R1. Use the IP addressing and DLCI information in the above chart.

Disable inverse-arp where appropriate. These routers should be able to ping every IP address within **their IP address space**.

On R1:

```
R1(config)#int s0/0
R1(config-if)#encap frame
R1(config-if)#No shut

R1(config-subif)#int S0/0.123 multipoint
R1(config-subif)#ip address 10.1.123.1 255.255.255.0
R1(config-subif)#frame-relay map ip 10.1.123.2 102
R1(config-subif)#frame-relay map ip 10.1.123.3 103
R1(config-subif)#frame-Relay map ip 10.1.123.1 102

R1(config)#int s0/0.14 point-to-point
R1(config-subif)#ip address 10.1.14.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 104
```

On R2:

```
R2(config)#int s0/0
R2(config-if)#encap frame
R2(config-if)#No shut

R2(config)#int s0/0.21 point-to-point
R2(config-subif)#ip address 10.1.123.2 255.255.255.0
R2(config-subif)#frame-relay interface-dlci 201
```

Note there is no need to disable inverse-arp, because it is disabled when a sub-interface is configured.

To test and verify the configuration:

On R2:

```
R2#Show frame-relay map

Serial0/0.21 (up): point-to-point dlci, dlci 201(0xC9,0x3090), broadcast
          status defined, active

R2#Ping 10.1.123.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.123.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
```

On R3:

```
R3(config)#int s0/0
R3(config-if)#encap frame
R3(config-if)#ip address 10.1.123.3 255.255.255.0
R3(config-if)#frame-relay map ip 10.1.123.1 301
R3(config-if)#frame-relay map ip 10.1.123.2 301
R3(config-if)#Frame-relay map ip 10.1.123.3 301
R3(config-if)#No frame-relay inverse-arp
R3(config-if)#No shut
```

Note inverse-arp should be Disabled because the configuration is done directly under the main Interface.

To verify and test the configuration:

On R3:

```
R3#Show frame map

Serial0/0 (up): ip 10.1.123.1 dlci 301(0x12D,0x48D0), static,
             CISCO, status defined, active
Serial0/0 (up): ip 10.1.123.2 dlci 301(0x12D,0x48D0), static,
             CISCO, status defined, active
Serial0/0 (up): ip 10.1.123.3 dlci 301(0x12D,0x48D0), static,
             CISCO, status defined, active

R3#Ping 10.1.123.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.123.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/113/116 ms

R3#Ping 10.1.123.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.123.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/114/120 ms

R3#Ping 10.1.123.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.123.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

On R4:

```
R4(config)#int s0/0
R4(config-if)#encap frame
R4(config-if)#No shut

R4(config)#int s0/0.41 point-to-point
R4(config-subif)#ip address 10.1.14.4 255.255.255.0
R4(config-subif)#frame-relay interface-dlci 401
```

To verify and test the configuration:

On R4:

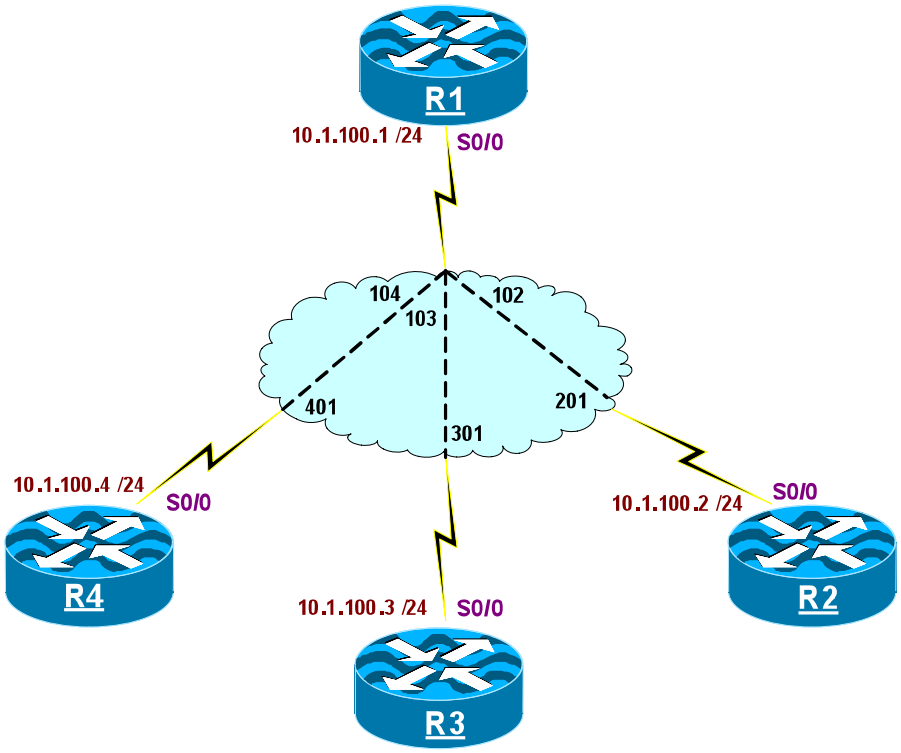
```
R4#Show frame map

Serial0/0.41 (up): point-to-point dlci, dlci 401(0x191,0x6410), broadcast
          status defined, active

R4#Ping 10.1.14.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.14.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms

R4#Ping 10.1.14.4
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.14.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/114/120 ms
```

Task 2

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 4 – Multipoint Frame-Relay With Out Frame-Relay Mapping



IP addressing and DLCI information chart:

Routers	IP address	Local DLCI	Connecting to:
R1's Frame-Relay interface	10.1.100.1 /24	102 103 104	R2 R3 R4
R2's Frame-Relay interface	**10.1.100.2 /24**	**201**	**R1**
R3's Frame-Relay interface	10.1.100.3 /24	301	R1
R4's Frame-Relay interface	**10.1.100.4 /24**	**401**	**R1**

Task 1

Configure the routers in a hub and spoke manner, with R1 as the hub and R2, R3 and R4 as the spokes.
Ensure that these routers have full reachability to each other without configuring the "frame-relay map" command.
Do not use PBR to accomplish this task.

In the following solution PPP is configured on the DLCIs, when PPP is configured, a host route is injected into the routing table. This host route provides NLRI to the next hop IP address.

On R1:

```
R1(config)#interface S0/0
R1(config-if)#encap frame-relay
R1(config-if)#frame-relay interface-dlci 102 ppp virtual-template1
R1(config-if)#frame-relay interface-dlci 103 ppp virtual-template1
R1(config-if)#frame-relay interface-dlci 104 ppp virtual-template1
R1(config-if)#NO shut

R1(config)#interface virtual-template 1
R1(config-if)#ip address 10.1.100.1 255.255.255.0
```

To verify the configuration:

On R1:

```
R1#Show frame-relay map
R1#
```

Note: There are no frame-relay maps.

On R2:

```
R2(config)#interface s0/0
R2(config-if)#encap frame-relay
R2(config-if)#frame-relay interface-dlci 201 ppp virtual-template 2
R2(config-if)#No shut

R2(config)#interface virtual-template 2
R2(config-if)#ip address 10.1.100.2 255.255.255.0
```

To verify the configuration:

On R2:

```
R2#Show frame-relay map
R2#
```

On R3:

```
R3(config)#interface s0/0
R3(config-if)#encap frame-relay
R3(config-if)#frame-relay interface-dlci 301 ppp virtual-template 3
R3(config-if)#No shut

R3(config)#interface virtual-template 3
R3(config-if)#ip address 10.1.100.3 255.255.255.0
```

To verify the configuration:

On R3:

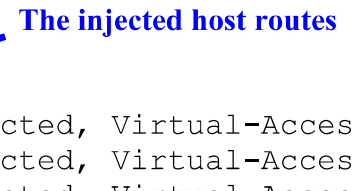
```
R2#Show frame-relay map
R2#
```

On R4:

```
R4(config)#interface s0/0
R4(config-if)#encap frame-relay
R4(config-if)#frame-helay interface-dlci 401 ppp virtual-template 4
R4(config-if)#No shut

R4(config)#interface virtual-template 4
R4(config-if)#ip address 10.1.100.4 255.255.255.0
```

To verify and test the configuration:

On R1:




```
R1#Show ip route | inc /32

C       10.1.100.4/32 is directly connected, Virtual-Access4
C       10.1.100.3/32 is directly connected, Virtual-Access3
C       10.1.100.2/32 is directly connected, Virtual-Access2

R1#Ping 10.1.100.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

R1#Ping 10.1.100.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

R1#Ping 10.1.100.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
```

On R2:

```
R2#Show ip route | inc /32

C       10.1.100.1/32 is directly connected, Virtual-Access2

R2#Ping 10.1.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

R2#Ping 10.1.100.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/115/116 ms

R2#Ping 10.1.100.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/113/116 ms
```

On R3:

```
R3#Show ip route | inc /32

C       10.1.100.1/32 is directly connected, Virtual-Access2

R3#Ping 10.1.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

R3#Ping 10.1.100.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/115/116 ms

R3#Ping 10.1.100.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/115/116 ms
```

On R4:

```
R4#Show ip route

C       10.1.100.1/32 is directly connected, Virtual-Access2

R4#Ping 10.1.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

R4#Ping 10.1.100.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/114/116 ms
```

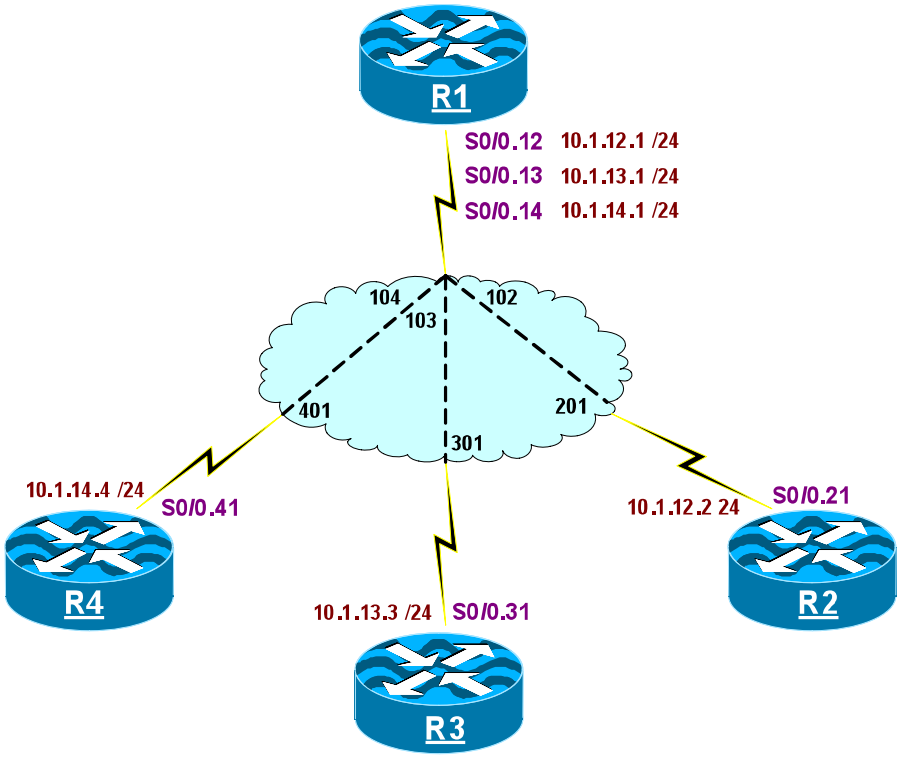
```
R4#Ping 10.1.100.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/114/116 ms
```

How do these routers communicate?
When running PPP (point-to-point protocol) a host route is injected by IPCP; if the routing table of a router is checked, you will see that next-hop is reachable via the local router's virtual-template interface. Since the VC are configured as P2P (because of PPP), any packet the local router puts on the virtual-template is received by one and only one router on the other side of the DLCI.

Task 2

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 5 – Frame-Relay and Authentication



IP addressing and DLCI information chart:

Routers	IP address	Local DLCI	Connecting to:
R1's Frame-Relay interface	10.1.12.1 /24 10.1.13.1 /24 10.1.14.1 /24	102 103 104	R2 R3 R4
R2's Frame-Relay interface	**10.1.12.2 /24**	**201**	**R1**
R3's Frame-Relay interface	10.1.13.3 /24	301	R1
R4's Frame-Relay interface	**10.1.14.4 /24**	**401**	**R1**

Task 1

Configure the routers in a hub and spoke manner using the IP addressing in the above chart.

These routers should be configured in a point-to-point manner as follows:

- **On R1::** DLCIs 102, 103, and 104 should be used for its connection to R2, R3, and R4 respectively.
- **On R2:, R3 and R4:** DLCIs 201, 301, and 401 should be used on R2, R3, and R4 respectively for their point-to-point frame-relay connection to R1 (the hub).

On R1:

```
R1(config)#interface s0/0
R1(config-if)#encap frame
R1(config-if)#No shut

R1(config)#interface s0/0.12 point-to-point
R1(config-subif)#ip address 10.1.12.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 102
R1(config-subif)#exit

R1(config)#interface s0/0.13 point-to-point
R1(config-subif)# ip address 10.1.13.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 103

R1(config)#interface s0/0.14 point-to-point
R1(config-subif)#ip address 10.1.14.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 104
```

To verify the configuration:

On R1:

```
R1#Show frame map

Serial0/0.12 (down): point-to-point dlci, dlci 102(0x66,0x1860), broadcast
          status defined, inactive
Serial0/0.13 (down): point-to-point dlci, dlci 103(0x67,0x1870), broadcast
          status defined, inactive
Serial0/0.14 (down): point-to-point dlci, dlci 104(0x68,0x1880), broadcast
          status defined, inactive
```

On R2:

```
R2(config)#int s0/0
R2(config-if)#encap frame
R2(config-if)#No shut

R2(config)#int s0/0.21 point-to-point
R2(config-subif)#ip address 10.1.12.2 255.255.255.0
R2(config-subif)#frame-relay interface-dlci 201
```

To verify and test the configuration:

On R2:

```
R2#Show frame map

Serial0/0.21 (up): point-to-point dlci, dlci 201(0xC9,0x3090), broadcast
          status defined, active

R2#Ping 10.1.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

On R3:

```
R3(config)#int s0/0
R3(config-if)#encap frame
R3(config-if)#No shut

R3(config)#int s0/0.31 point-to-point
R3(config-subif)#ip address 10.1.13.3 255.255.255.0
R3(config-subif)#frame-relay interface-dlci 301
```

To verify and test the configuration:

On R3:

```
R3#Show frame map

Serial0/0.31 (up): point-to-point dlci, dlci 301(0x12D,0x48D0), broadcast
          status defined, active
```

```
R3#Ping 10.1.13.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

On R4:

```
R4(config)#int s0/0
R4(config-if)#encap frame
R4(config-if)#No shut

R4(config)#int s0/0.41 point-to-point
R4(config-subif)#ip address 10.1.14.4 255.255.255.0
R4(config-subif)#frame-relay interface-dlci 401
```

To verify and test the configuration:

On R4:

```
R4#Show frame map

Serial0/0.41 (up): point-to-point dlci, dlci 401(0x191,0x6410), broadcast
          status defined, active

R4#Ping 10.1.14.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.14.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

Task 2

Configure authentication on the routers as follows:

A. **For R1 and R2's connection:**
   R1 should send a challenge when it is called by R2.
   R2 should **not** authenticate when it is called.
   The password for this authentication should be "**cisco12**".
   This authentication should be successful even if the host name of the router is changed.

B. For R1 and R3's connection:

R1 should **not** authenticate when it is called.
R3 should use PAP authentication when it is called by R1.
The password for this authentication should be "**cisco13**".
The host name of the router should be used for this authentication.

C. For R1 and R4's connection:

R1 should send a challenge when it is called by R4.
R4 should use PAP authentication when it is called by R1.
The password for CHAP authentication should be "**cisco**", whereas, the password for PAP should be set to "**ciscoPAP**" and the hostname should be configured to be "**R1-PAP**".

For R1 and R2's connection:

On R1:

```
R1(config)#username r2 password 0 cisco12

R1(config)#int s0/0.12
R1(config-if)#No ip addr
R1(config-if)#frame-relay interface-dlci 102 ppp virtual-template 12

R1(config)#int s0/0.13
R1(config-subif)#No ip address
R1(config-subif)#frame-relay interface-dlci 103 ppp virtual-template 13

R1(config)#int s0/0.14
R1(config-subif)#No ip address
R1(config-subif)#frame-relay interface-dlci 104 ppp virtual-template 14

R1(config)#int virtual-template12
R1(config-if)#ip address 10.1.12.1 255.255.255.0
R1(config-if)#ppp authentication chap callin
R1(config-if)#ppp chap hostname R1
```

On R2:

```
R2(config)#username r1 password 0 cisco12

R2(config)#int s0/0.21
R2(config-subif)#no ip addr
R2(config-subif)#frame-relay interface-dlci 201 ppp virtual-template 21
```

```
R2(config)#int virtual-template21
R2(config-if)#ip address 10.1.12.2 255.255.255.0
R2(config-if)#ppp chap hostname R2
```

To test and verify the configuration:

On R2:

```
R2#Debug ppp authentication

R2(config)#int s0/0
R2(config-if)#shut
R2(config-if)#No shut

R2#Ping 10.1.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
.!!
Vi1 PPP: No authorization without authentication
Vi1 CHAP: I CHALLENGE id 2 len 23 from "R1"
Vi1 CHAP: Using hostname from interface CHAP
Vi1 CHAP: Using password from AAA
Vi1 CHAP: O RESPONSE id 2 len 23 from "R2"
Vi1 CHAP: I SUCCESS id 2 len 4!!

Success rate is 80 percent (4/5), round-trip min/avg/max = 48/50/52 ms
```

The output of the above debug command shows the "Challenge" packet coming inbound, "Response" packet going outbound, and the "Success" coming inbound.

For R1 and R3's connection:

On R1:

```
R1(config)#pnt virtual-template13
R1(config-if)#ip address 10.1.13.1 255.255.255.0
R1(config-if)#ppp pap sent-username r1 password 0 cisco13
```

On R3:

```
R3(config)#username r1 password 0 cisco13

R3(config)#int s0/0.31
```

```
R3(config-subif)#No ip address
R3(config-subif)#frame-relay interface-dlci 301 ppp virtual-template 31

R3(config)#int virtual-template31
R3(config-if)#ip address 10.1.13.3 255.255.255.0
R3(config-if)#ppp authentication pap callin
```

To test and verify the configuration:

On R3:

```
R3#Debug ppp authentication

R3(config)#int s0/0
R3(config-if)#shut
R3(config-if)#No shut

R3#Ping 10.1.13.1

Vi2 PPP: Authorization required
Vi2 PAP: I AUTH-REQ id 3 len 15 from "R1"
Vi2 PAP: Authenticating peer R1
Vi2 PPP: Sent PAP LOGIN Request
Vi2 PPP: Received LOGIN Response PASS
Sent LCP AUTHOR Request
Vi2 PPP: Sent IPCP AUTHOR Request
Vi2 LCP: Received AAA AUTHOR Response PASS
Vi2 IPCP: Received AAA AUTHOR Response PASS
Vi2 PAP: O AUTH-ACK id 3 len 5
Vi2 PPP: Sent IPCP AUTHOR Request

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
```

For R1 and R4's connection:

On R1:

```
R1(config)#username r4 password cisco

R1(config)#int Virtual-Template14
R1(config)#ip address 10.1.14.1 255.255.255.0
R1(config)#ppp authentication chap callin
```

```
R1(config)# ppp pap sent-username r1-pap password 0 ciscoPAP
```

On R4:

```
R4(config)#username r1-pap password ciscoPAP
R4(config)#username r1 password cisco

R4(config)#int s0/0.41
R4(config-subif)#No ip address
R4(config-subif)#frame-relay interface-dlci 401 ppp virtual-template 41

R4(config)#int Virtual-Template41
R4(config-if)#ip address 10.1.14.4 255.255.255.0
R4(config-if)#ppp authentication pap callin
```

To test and verify the configuration:

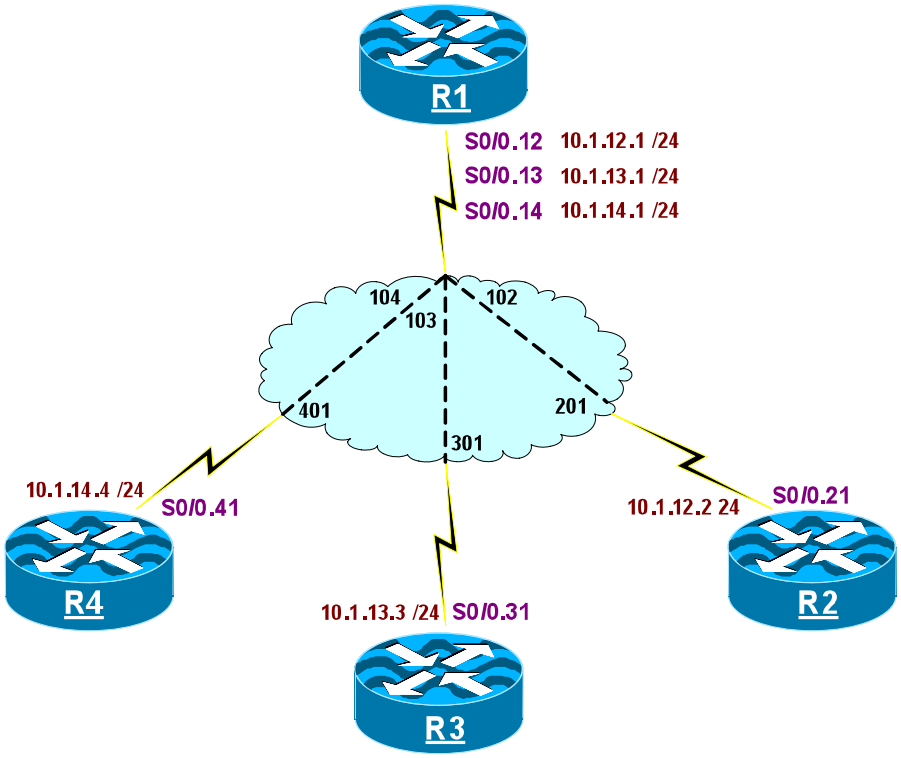
On R4:

```
R4#Debug ppp authentication

R4#(config)#int s0/0
R4#(config-if)#shut
R4#(config-if)#No shut

R4#Ping 10.1.14.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.14.1, timeout is 2 seconds:
...
Vi1 PAP: I AUTH-REQ id 6 len 20 from "R1-PAP"
Vi1 PAP: Authenticating peer R1-PAP
Vi1 PPP: Sent PAP LOGIN Request
Vi1 PPP: Received LOGIN Response PASS
Vi1 CHAP: I CHALLENGE id 6 len 23 from "R1"
Vi1 CHAP: O RESPONSE id 6 len 23 from "R4"
Vi1 LCP: Received AAA AUTHOR Response PASS.!

Success rate is 20 percent (1/5), round-trip min/avg/max = 56/56/56 ms

Vi1 IPCP: Received AAA AUTHOR Response PASS

Vi1 PAP: O AUTH-ACK id 6 len 5
Vi1 CHAP: I SUCCESS id 6 len 4
Vi1 PPP: Sent IPCP AUTHOR Request
```

Task 3

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 6 – Frame-Relay End-to-End Keepalive



IP addressing and DLCI information chart:

Routers	IP address	Local DLCI	Connecting to:
R1's Frame-Relay interface	10.1.12.1 /24 10.1.13.1 /24 10.1.14.1 /24	102 103 104	R2 R3 R4
R2's Frame-Relay interface	**10.1.12.2 /24**	**201**	**R1**
R3's Frame-Relay interface	10.1.13.3 /24	301	R1
R4's Frame-Relay interface	**10.1.14.4 /24**	**401**	**R1**

Task 1

Configure the routers in a hub and spoke manner using the IP addressing in the above chart.

These routers should be configured in a point-to-point manner as follows:

- **On R1::** DLCIs 102, 103, and 104 should be used for its connection to R2, R3, and R4 respectively.
- **On R2:, R3 and R4:** DLCIs 201, 301, and 401 should be used on R2, R3, and R4 respectively for their point-to-point frame-relay connection to R1 (the hub).

On R1:

```
R1(config)#interface s0/0
R1(config-if)#encap frame
R1(config-if)#No shut

R1(config)#interface s0/0.12 point-to-point
R1(config-subif)# ip address 10.1.12.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 102

R1(config)#interface s0/0.13 point-to-point
R1(config-subif)#ip address 10.1.13.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 103

R1(config)#interface s0/0.14 point-to-point
R1(config-subif)#ip address 10.1.14.1 255.255.255.0
R1(config-subif)#frame-relay interface-dlci 104
```

To verify the configuration:

On R1:

```
R1#Show frame map

Serial0/0.12 (down): point-to-point dlci, dlci 102(0x66,0x1860), broadcast
          status defined, inactive
Serial0/0.13 (down): point-to-point dlci, dlci 103(0x67,0x1870), broadcast
          status defined, inactive
Serial0/0.14 (down): point-to-point dlci, dlci 104(0x68,0x1880), broadcast
          status defined, inactive
```

On R2:

```
R2config-subif)# int s0/0
R2config-if)#encap frame
R2config-if)#No shut

R2config)#int s0/0.21 point-to-point
R2config-subif)#ip address 10.1.12.2 255.255.255.0
R2config-subif)#frame-relay interface-dlci 201
```

To verify and test the configuration:

On R2:

```
R2#Show frame map

Serial0/0.21 (up): point-to-point dlci, dlci 201(0xC9,0x3090), broadcast
          status defined, active

R2#Ping 10.1.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

Note: A "Show Frame-Relay Map" on R1 will show that DLCI 102 is active. Whereas, the other DLCIs are inactive. As the spokes are configured, the DLCIs transition into active state.

On R1:

```
R1#Show frame map

Serial0/0.12 (up): point-to-point dlci, dlci 102(0x66,0x1860), broadcast
          status defined, active
Serial0/0.13 (down): point-to-point dlci, dlci 103(0x67,0x1870), broadcast
          status defined, inactive
Serial0/0.14 (down): point-to-point dlci, dlci 104(0x68,0x1880), broadcast
          status defined, inactive
```

On R3:

```
R3config-subif)#int s0/0
R3config-if)#encap frame
R3config-if)#No shut

R3config-subif)#int s0/0.31 point-to-point
R3config-subif)#ip address 10.1.13.3 255.255.255.0
```

```
R3config-subif)#frame-relay interface-dlci 301
```

To verify and test the configuration:

On R3:

```
R3#Show frame map

Serial0/0.31 (up): point-to-point dlci, dlci 301(0x12D,0x48D0), broadcast
          status defined, active

R3#Ping 10.1.13.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

On R4:

```
R4config-subif)#int s0/0
R4config-if)#encap frame
R4config-if)#No shut

R4config-subif)#int s0/0.41 point-to-point
R4config-subif)#ip address 10.1.14.4 255.255.255.0
R4config-subif)#frame-relay interface-dlci 401
```

To verify and test the configuration:

On R4:

```
R4#Show frame map

Serial0/0.41 (up): point-to-point dlci, dlci 401(0x191,0x6410), broadcast
          status defined, active

R4#Ping 10.1.14.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.14.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

Task 2

Configure frame-relay end-to-end keepalives on R1 and R2, these routers should be configured in bidirectional mode using the default values.

Routers depend on the LMIs to maintain the status of an active connection, since the intermediate switches in the cloud may not support NNI LMIs; FREEK can be used to provide the local router with the status of the remote end. FREEK accomplishes this by providing an end to end keepalive, this

Keepalive runs on the data DLCI (16 – 997) and not the LMI DLCI (cisco LMI uses DLCI 1023, and Q933a and ANSI uses DLCI 0).

FREEK maintains two internal keepalives:

- **The first one is used to send out keepalive requests and to handle responses to the requests; this is considered the send side.**
- **The second one is to handle and reply to the requests; this is considered the receive side.**

At the send side when the timer expires, the send side transmits a keepalive and waits for a reply.
When the send side receives the reply before the timer expires a frame-relay, keepalive is recorded. If the timer expires and no keepalives are received, an error event is recorded.
If a sufficient number of error events are observed, the PVC will transition to a down state. The number of events necessary to change the status from up to down is known as event window.
Some of the parameters and values can be changed as follows:

Frame-Relay End-to-End Keepalive [send | receive] error-threshold

This command configures the number of frame-relay end-to-end keepalive errors that must occur in the event window before the interface goes down. Default is 2, and the maximum number is 32.

Frame-Relay End-to-End Keepalive [send | receive] success-events
This command configures the number of frame-relay end-to-end keepalive successes that must occur before the interface comes up. Default is 2, and the maximum number is 32.

Frame-Relay End-to-End Keepalive [send | receive] timer
This command configures end to end keepalive timers; this can be configured for send or receive side

Frame-Relay End-to-End Keepalive event-window
This command tells the IOS to keep track of x number of most recent events.

On R1:

```
R1(config)#map-class frame-relay TST12
R1(config-map-class)#frame-relay end-to-end keepalive mode bidirectional
```

```
R1(config)#int serial0/0.12 point-to-point
R1(config-subif)#frame-relay interface-dlci 102
R1(config-fr-dlci)#class TST12
```

On R2:

```
R2(config)#map-class frame-relay TST21
R2(config-map-class)#frame-relay end-to-end keepalive mode bidirectional

R2(config)#interface serial0/0.21
R2(config-subif)#frame interface-dlci 201
R2(config-fr-dlci)#class TST21
```

To verify the configuration:

On R2:

```
R2#Show frame-relay end-to-end keepalive interface s0/0.21

End-to-end Keepalive Statistics for Interface Serial0/0.21 (Frame Relay DTE)

DLCI = 201, DLCI USAGE = LOCAL, VC STATUS = ACTIVE (EEK UP)

SEND SIDE STATISTICS

Send Sequence Number: 1,          Receive Sequence Number: 2
Configured Event Window: 3,       Configured Error Threshold: 2
Total Observed Events: 4,         Total Observed Errors: 0
Monitored Events: 3,              Monitored Errors: 0
Successive Successes: 3,          End-to-end VC Status: UP

RECEIVE SIDE STATISTICS

Send Sequence Number: 1,          Receive Sequence Number: 255
Configured Event Window: 3,       Configured Error Threshold: 2
Total Observed Events: 3,         Total Observed Errors: 0
Monitored Events: 2,              Monitored Errors: 0
Successive Successes: 2,          End-to-end VC Status: UP
```

To test the configuration:

On R2:

```
R2#Debug frame-relay end-to-end keepalive event
```

```
*Aug 16 10:36:47.355: EEK SUCCESS (request, Serial0/0.21 DLCI 201)
*Aug 16 10:36:49.779: EEK SUCCESS (reply, Serial0/0.21 DLCI 201)
*Aug 16 10:36:57.275: EEK SUCCESS (request, Serial0/0.21 DLCI 201)
*Aug 16 10:36:59.687: EEK SUCCESS (reply, Serial0/0.21 DLCI 201)
*Aug 16 10:37:07.263: EEK SUCCESS (request, Serial0/0.21 DLCI 201)
*Aug 16 10:37:09.603: EEK SUCCESS (reply, Serial0/0.21 DLCI 201)
*Aug 16 10:37:17.187: EEK SUCCESS (request, Serial0/0.21 DLCI 201)
*Aug 16 10:37:19.543: EEK SUCCESS (reply, Serial0/0.21 DLCI 201)
```

Note: The output of the above debug command shows the events that are recorded. You can see that every 10 seconds the local router sends a keepalive (request). It receives a reply within 10 seconds therefore, it records a success event.

Let's turn off the debug on R2:

```
R2#U all
```

To test and verify the configuration:

In order to test the configuration, the "debug frame-relay end-to-end keepalive events" is turned on and the S0/0.21 sub-interface on R2 is shut down and the output of the debug command is analyzed:

On R1:

```
R2#Debug frame-relay end-to-end keepalive event
```

On R2:

```
R2(config)#int s0/0.21
R2(config-subif)#shut
```

You should see the following output on R2's console:

On R1:

The following reply was successful:

```
*Aug 16 15:44:02.827: EEK SUCCESS (reply, Serial0/0.12 DLCI 102)
```

The following request timed out, because R2 did not respond at all, this recorded the first error:

```
*Aug 16 15:44:09.843: EEK receiver timeout (Serial0/0.12 DLCI 102)

*Aug 16 15:44:22.739: EEK sender timeout (Serial0/0.12 DLCI 102)
```

The following request timed out, because R2 did not respond again, this recorded the second error:

```
*Aug 16 15:44:24.843: EEK receiver timeout (Serial0/0.12 DLCI 102)
```

The VC is forced down because the error threshold is exceeded:

```
*Aug 16 15:44:24.843: EEK force VC DOWN (Serial0/0.12 DLCI 102)
```

EEK brought down the S0/0.12 sub-interface:

```
*Aug 16 15:44:24.843: %FR_EEK-5-FAILED: Interface Serial0/0.12 - DLCI 102
```

To see that the S0/0.12 sub-interface is down:

```
R1#Show ip int br | inc Interface|Serial0/0.12
Interface                  IP-Address      OK? Method Status                Protocol
Serial0/0.12               10.1.12.1       YES manual down                  down
```

The main interface (S0/0) is still sending the keepalives every 10 seconds:

```
*Aug 16 15:44:32.651: EEK sender timeout (Serial0/0.12 DLCI 102)
*Aug 16 15:44:39.843: EEK receiver timeout (Serial0/0.12 DLCI 102)
*Aug 16 15:44:42.583: EEK sender timeout (Serial0/0.12 DLCI 102)

R1#Show frame end keep inter s0/0.12

End-to-end Keepalive Statistics for Interface Serial0/0.12 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, VC STATUS = ACTIVE (EEK DOWN)

SEND SIDE STATISTICS

Send Sequence Number: 166,      Receive Sequence Number: 88
Configured Event Window: 3,     Configured Error Threshold: 2
Total Observed Events: 192,     Total Observed Errors: 102
Monitored Events: 3,            Monitored Errors: 3
Successive Successes: 0,        end-to-end VC Status: DOWN

RECEIVE SIDE STATISTICS

Send Sequence Number: 88,       Receive Sequence Number: 87
Configured Event Window: 3,     Configured Error Threshold: 2
Total Observed Events: 158,     Total Observed Errors: 68
Monitored Events: 3,            Monitored Errors: 3
Successive Successes: 0,        end-to-end VC Status: DOWN
```

```
Failures Since Started: 2,      Last Failure: 00:12:52
```

By default, the "error threshold" is set to 2, therefore, when R1 did not receive two replies within three events, its sub-interface S0/0.12 transitioned into down/down state. However, the main interface (S0/0), is still in up/up state.

```
R1#Show ip int br | inc Interface|Serial0/0_

Interface                  IP-Address      OK? Method Status                Protocol
Serial0/0                  unassigned      YES NVRAM  up                    up
```

To test the success events:

In order to test the success events, the "debug frame-relay end-to-end keepalive events" is turned on and the S0/0.21 sub-interface on R2 is brought up and the output of the debug command is analyzed:

On R1:

```
R2#Debug frame-relay end-to-end keepalive event
```

On R2:

```
R2(config)#interface s0/0.21
R2(config-subif)#No shut
```

On R1:

The output of the debug command is modified, the following three lines show that the replies are not successful:

```
*Aug 16 16:00:47.827: EEK sender timeout (Serial0/0.12 DLCI 102)
*Aug 16 16:00:57.739: EEK sender timeout (Serial0/0.12 DLCI 102)
*Aug 16 16:01:07.659: EEK sender timeout (Serial0/0.12 DLCI 102)
```

The S0/0.21 sub-interface is brought back up, therefore, the local router receives two successful replies in a row:

```
*Aug 16 16:01:07.667: EEK SUCCESS (reply, Serial0/0.12 DLCI 102)
*Aug 16 16:01:17.571: EEK SUCCESS (reply, Serial0/0.12 DLCI 102)
```

The VC is forced to UP/UP state:

```
*Aug 16 16:01:19.759: EEK force VC UP (Serial0/0.12 DLCI 102)
```

End-to-End keepalives continue successfully:

```
*Aug 16 16:01:27.523: EEK SUCCESS (reply, Serial0/0.12 DLCI 102)
*Aug 16 16:01:29.683: EEK SUCCESS (request, Serial0/0.12 DLCI 102)
```

On R1:

```
R1#Sh frame end keep inter S0/0.12

End-to-end Keepalive Statistics for Interface Serial0/0.12 (Frame Relay DTE)

DLCI = 102, DLCI USAGE = LOCAL, VC STATUS = ACTIVE (EEK UP)

SEND SIDE STATISTICS

Send Sequence Number: 11,        Receive Sequence Number: 165
Configured Event Window: 3,      Configured Error Threshold: 2
Total Observed Events: 293,      Total Observed Errors: 126
Monitored Events: 3,             Monitored Errors: 0
Successive Successes: 3,         end-to-end VC Status: UP

RECEIVE SIDE STATISTICS

Send Sequence Number: 165,       Receive Sequence Number: 164
Configured Event Window: 3,      Configured Error Threshold: 2
Total Observed Events: 250,      Total Observed Errors: 83
Monitored Events: 3,             Monitored Errors: 0
Successive Successes: 3,         end-to-end VC Status: UP

Failures Since Started: 2,       Last Failure: 00:29:28
```

Note: after three success events in a row, the sub-interface is transitioned into up state.

Task 3

Configure frame-relay end-to-end keepalives for the VC that connects R1 to R3. R1 should be configured in request mode, whereas R3 should be configured in reply mode using the default values.

On R1:

```
R1(config)#map-class frame-relay TST13
```

```
R1(config-map-class)#frame-relay end-to-end keepalive mode request

R1(config)#interface serial0/0.13
R1(config-subif)#frame-relay interface-dlci 103
R1(config-fr-dlci)#class TST13
```

On R3:

```
R3(config)#map-class frame-relay TST31
R3(config-map-class)#frame-relay end-to-end keepalive mode reply

R3(config)#interface serial0/0.31
R3(config-subif)#frame-relay interface-dlci 301
R3(config-fr-dlci)#class TST31
```

To verify the configuration:

On R3:

```
R3#Show frame end-to-end keepalive Interface s0/0.31

End-to-end Keepalive Statistics for Interface Serial0/0.31 (Frame Relay DTE)

DLCI = 301, DLCI USAGE = LOCAL, VC STATUS = ACTIVE (EEK UP)

RECEIVE SIDE STATISTICS

Send Sequence Number: 3,          Receive Sequence Number: 2
Configured Event Window: 3,       Configured Error Threshold: 2
Total Observed Events: 5,         Total Observed Errors: 0
Monitored Events: 3,              Monitored Errors: 0
Successive Successes: 3,          end-to-end VC Status: UP
```

To test the configuration:

On R3:

```
R3#Debug frame-relay end-to-end keep events
```

On R1:

```
R1(config)#int s0/0.13
R1(config-subif)#shut
```

On R3:

The following four requests are successful:

```
*Aug  9 06:10:13.607: EEK SUCCESS (request, Serial0/0.31 DLCI 301)
*Aug  9 06:10:23.599: EEK SUCCESS (request, Serial0/0.31 DLCI 301)
*Aug  9 06:10:33.515: EEK SUCCESS (request, Serial0/0.31 DLCI 301)
*Aug  9 06:10:43.479: EEK SUCCESS (request, Serial0/0.31 DLCI 301)
```

The following shows that the local router recorded two error events:

```
*Aug  9 06:10:58.479: EEK receiver timeout (Serial0/0.31 DLCI 301)
*Aug  9 06:11:13.479: EEK receiver timeout (Serial0/0.31 DLCI 301)
```

The VC is forced down and the sub-interface is transitioned into DOWN/DOWN state:

```
*Aug  9 06:11:13.479: EEK force VC DOWN (Serial0/0.31 DLCI 301)
*Aug  9 06:11:13.479: %FR_EEK-5-FAILED: Interface Serial0/0.31 - DLCI 301
```

On R3:

```
R3#Show frame end-to-end keepalive interface s0/0.31

End-to-end Keepalive Statistics for Interface Serial0/0.31 (Frame Relay DTE)

DLCI = 301, DLCI USAGE = LOCAL, VC STATUS = ACTIVE (EEK DOWN)

RECEIVE SIDE STATISTICS

Send Sequence Number: 16,        Receive Sequence Number: 15
Configured Event Window: 3,      Configured Error Threshold: 2
Total Observed Events: 37,       Total Observed Errors: 19
Monitored Events: 3,             Monitored Errors: 3
Successive Successes: 0,         end-to-end VC Status: DOWN
Failures Since Started: 1,       Last Failure: 00:04:16
```

On R1:

```
R1(config)#int s0/0.13
R1(config-subif)No shut
```

On R3:

```
R3#Show frame end-to-end keepalive interface s0/0.31
```

```
End-to-end Keepalive Statistics for Interface Serial0/0.31 (Frame Relay DTE)

DLCI = 301, DLCI USAGE = LOCAL, VC STATUS = ACTIVE (EEK UP)

RECEIVE SIDE STATISTICS

Send Sequence Number: 18,        Receive Sequence Number: 17
Configured Event Window: 3,      Configured Error Threshold: 2
Total Observed Events: 43,       Total Observed Errors: 23
Monitored Events: 0,             Monitored Errors: 0
Successive Successes: 0,         end-to-end VC Status: UP
Failures Since Started: 1,       Last Failure: 00:05:42
```

Note: the sub-interface S0/0.31 on R3 transitioned into up/up state.

Task 4

Configure frame-relay end-to-end keepalives for the VC that connects R1 to R4. These two routers should be configured in **bidirectional** mode using the following policy:

If these routers have three errors within 5 events, the sub-interface should transition into down/down state, and if they have four success events in a row, the sub-interface should transition into up/up state. Ensure that the keepalives are exchanged every 20 seconds.

On R1:

```
R1(config)#map-class frame-relay TST14
R1(config-map-class)#frame-relay end-to-end keep mode bidirectional


R1(config-map-class)#frame-relay end-to-end keep event-window recv 5
R1(config-map-class)#frame-relay end-to-end keep event-window send 5

R1(config-map-class)#frame-relay end-to-end keep error-threshold recv 3
R1(config-map-class)#frame-relay end-to-end keep error-threshold send 3

R1(config-map-class)#frame-relay end-to-end keep success-events recv 4
R1(config-map-class)#frame-relay end-to-end keep success-events send 4

R1(config-map-class)#frame-relay end-to-end keepalive timer recv 20
R1(config-map-class)#frame-relay end-to-end keepalive timer send 20
```

```
R1(config)#int serial0/0.14
R1(config-subif)#frame-relay interface-dlci 104
R1(config-subif)#class TST14
```

On R4:

```
R4(config)#map-class frame-relay TST41
R4(config-map-class)#frame-relay end-to-end keep mode bidirectional

R4(config-map-class)#frame-relay end-to-end keep event-window recv 5
R4(config-map-class)#frame-relay end-to-end keep event-window send 5

R4(config-map-class)#frame-relay end-to-end keep error-threshold recv 3
R4(config-map-class)#frame-relay end-to-end keep error-threshold send 3

R4(config-map-class)#frame-relay end-to-end keep success-events recv 4
R4(config-map-class)#frame-relay end-to-end keep success-events send 4

R4(config-map-class)#frame-relay end-to-end keepalive timer recv 20
R4(config-map-class)#frame-relay end-to-end keepalive timer send 20

R4(config)#int serial0/0.41
R4(config-subif)#frame-relay interface-dlci 401
R4(config-fr-dlci)#class TST41
```

To verify the configuration:

On R1:

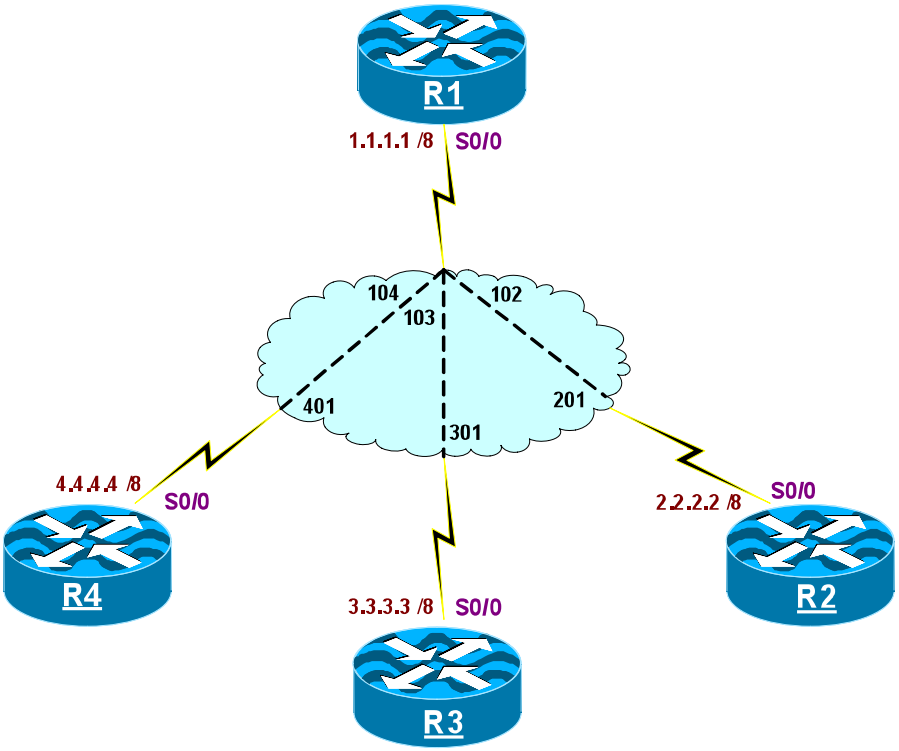
```
R4#Show frame-relay end-to-end keepalive interface s0/0.14

End-to-end Keepalive Statistics for Interface Serial0/0.41 (Frame Relay DTE)

DLCI = 401, DLCI USAGE = LOCAL, VC STATUS = ACTIVE (EEK UP)

SEND SIDE STATISTICS

Send Sequence Number: 255,      Receive Sequence Number: 255
Configured Event Window: 5,     Configured Error Threshold: 3
Total Observed Events: 1,       Total Observed Errors: 0
Monitored Events: 0,            Monitored Errors: 0
Successive Successes: 0,        End-to-end VC Status: UP

RECEIVE SIDE STATISTICS
```

```
Send Sequence Number: 0,          Receive Sequence Number: 0
Configured Event Window: 5,       Configured Error Threshold: 3
Total Observed Events: 0,         Total Observed Errors: 0
Monitored Events: 0,              Monitored Errors: 0
Successive Successes: 0,          End-to-end VC Status: UP
```

Task 5

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 7 – Tricky Frame-relay configuration



IP addressing and DLCI information Chart:

Routers	IP address	Local DLCI	Connecting to:
R1's Loopback 0 interface	1.1.1.1 /8		
R1's Frame-Relay interface	Ip unnumbered Lo0	102	R2
	Ip unnumbered Lo0	103	R3
	Ip unnumbered Lo0	104	R4
R2's Loopback 0 interface	**2.2.2.2 /8**		
R2's Frame-Relay interface	**Ip unnumbered Lo0**	**201**	**R1**
R3's Loopback 0 interface	3.3.3.3 /8		
R3's Frame-Relay interface	Ip unnumbered Lo0	301	R1
R4's Loopback 0 interface	**4.4.4.4 /8**		
R4's Frame-Relay interface	**Ip unnumbered Lo0**	**401**	**R1**

Task 1

Configure the routers in a hub and spoke manner using the IP addressing in the above chart.

The hub router (R1): This router should use DLCIs 102, 103, and 104 for its connection to R2, R3, and R4 respectively. This router should be configured in a multipoint manner.

The spokes, R2, R3 and R4: DLCIs 201, 301, and 401 should be used by R2, R3, and R4 respectively for their frame-relay connection to R1 (the hub).

Ensure that these routers have full reachability to every loopback interface this should include their own. You should **not** use "**frame-relay map**", and/or static/dynamic routing to accomplish this task.

None of the routers should be configured with sub-interface/s.

On R1:

```
R1(config)#int s0/0
R1(config-if)#encap frame-relay
R1(config-if)#frame-relay interface-dlci 102 ppp virtual-template 1
R1(config-if)#Frame-relay interface-dlci 103 ppp virtual-template 1
R1(config-if)#frame-relay interface-dlci 104 ppp virtual-template 1
R1(config-if)#No shu

R1(config)#int virtual-template 1
R1(config-if)#ip unnumbered lo0

R1(config)#int lo0
R1(config-if)#ip address 1.1.1.1 255.0.0.0
```

On R2:

```
R2(config)#int s0/0
R2(config-if)#encap frame-relay
R2(config-if)#frame-relay interface-dlci 201 ppp virtual-template 2
R2(config-if)#No shu

R2(config)#int virtual-template 2
R2(config-if)#ip unnumbered lo0

R2(config)#int lo0
```

```
R2(config-if)#ip address 2.2.2.2 255.0.0.0
```

On R3:

```
R3(config)#int s0/0
R3(config-if)#encap frame-relay
R3(config-if)#frame-relay interface-dlci 301 ppp virtual-template 3
R3(config-if)#No shu

R3(config)#int virtual-template 3
R2)config-if)#ip unnumbered lo0

R3(config)#int lo0
R3(config-if)#ip address 3.3.3.3 255.0.0.0
```

On R4:

```
R4(config)#int s0/0
R4(config-if)#encap frame-relay
R4(config-if)#frame-relay interface-dlci 401 ppp virtual-template 4
R4(config-if)#No shu

R4(config)#int virtual-template 4
R4(config-if)#ip unnumbered lo0

R4(config)#int lo0
R4(config-if)#ip address 4.4.4.4 255.0.0.0
```

To verify and test connectivity between the hub and its attached spokes:

On R1:

```
R1#Show ip route | b Gateway

Gateway of last resort is not set

C    1.0.0.0/8 is directly connected, Loopback0
     2.0.0.0/32 is subnetted, 1 subnets
C       2.2.2.2 is directly connected, Virtual-Access1
     3.0.0.0/32 is subnetted, 1 subnets
C       3.3.3.3 is directly connected, Virtual-Access2
     4.0.0.0/32 is subnetted, 1 subnets
C       4.4.4.4 is directly connected, Virtual-Access3
```

Note: when PPP is configured, in the last step of PPP connection, IPCP creates a host route for the

router's interface that is connected to your local router. This behavior can be disabled using the "no peer neighbor-route" command. Because of this behavior, in PPP, R1 should have connectivity to every spoke router, as follows:

On R1:

```
R1#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#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 = 56/58/60 ms

R1#Ping 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

R1#Ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms
```

On R2:

```
R2#Show ip route | b Gateway

Gateway of last resort is not set

     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Virtual-Access1
C    2.0.0.0/8 is directly connected, Loopback0
```

Note: R2 has reachability to R1 and its own interface but not to any of the spokes

```
R2#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

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

R2#Ping 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R2#Ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

On R3:

```
R3#Show ip route | b Gate

Gateway of last resort is not set

     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Virtual-Access1
C    3.0.0.0/8 is directly connected, Loopback0
```

Note: R3 has reachability to R1 and its own interface but not to any of the spokes

```
R3#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

R3#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 0 percent (0/5)

R3#Ping 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

R3#Ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

On R4:

```
R2#Show ip route | b Gate

Gateway of last resort is not set

     1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Virtual-Access1
C    4.0.0.0/8 is directly connected, Loopback0
```

Note: R4 has reachability to R1 and its own interface but not to any of the spokes

```
R4#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

R4#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 0 percent (0/5)

R4#Ping 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R4#Ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms
```

PBR is configured to provide reachability between the spokes, as follows:

On R2:, R3 and R4

```
Rx(config)#ip local policy route-map TST

Rx(config-route-map)#route-map TST permit 10
Rx(config-route-map)#set ip next-hop 1.1.1.1

Rx(config-route-map)#route-map TST permit 90
```

To test the configuration:

On R2:

```
R2#Debug ip policy

R2#Ping 3.3.3.3 source 2.2.2.2 repeat 1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 2.2.2.2
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 116/117/120 ms

IP: s=2.2.2.2 (local), d=3.3.3.3, len 100, policy match
IP: route map TST, item 10, permit
```

```
IP: s=2.2.2.2 (local), d=3.3.3.3 (Virtual-Access1), len 100, policy
routed
IP: local to Virtual-Access1 1.1.1.1
```

Let's turn off the debug and try reachability to the other spokes

On R2:

```
R2#U all

R2#Ping 4.4.4.4 source 2.2.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 2.2.2.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/100/104 ms
```

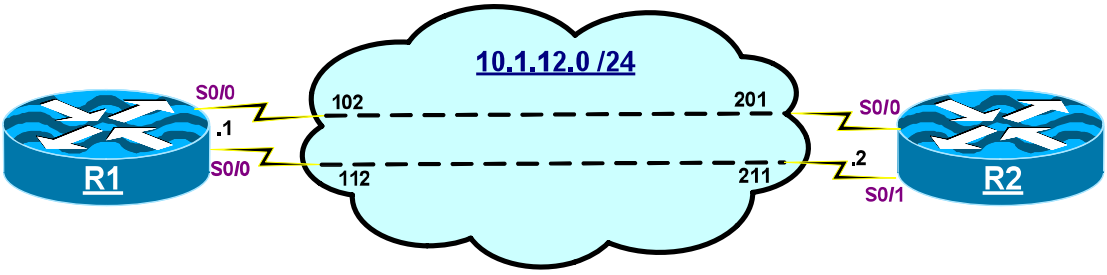
On R3:

```
R3#Ping 4.4.4.4 source 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 3.3.3.3
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 100/100/104 ms
```

Task 2

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 8 – Frame-Relay Multilinking



Task 1

Configure the frame-relay connections between R1 and R2 in a point-to-point manner using the DLCIs and interfaces in the diagram. Configure R1 and R2 using 10.1.12.1 /24 and 10.1.12.2 /24 IP addresses respectively. Ensure that these two VCs appear as one link, a maximum number of two links and a minimum number of one link is required to ensure proper operation. These links should have authentication capability.

Note: the task does not specifically ask for PPP Multilink to be configured. Since the task asks each router to have a single IP address and it states that the links should appear as one with authentication capability, that should be enough to indicate the PPP Multilink configuration.

Most of the time there is only a single connection between two routers, but there are situations where you may need to have multiple layer one connections between the two routers. One reason could be to increase the size of the pipe between the two routers. The point of Multilink PPP is to take multiple PPP links and "bond" them together to act as a single PPP link. These PPP links that are being bonded could be an ISDN BRI circuit, T1 circuits, or other types of PPP circuits as long as they are from the same provider.

On R1:

The following command creates a logical multilink group, in the following configuration, the multilink group is assigned a value of 12, but the range is 1 – 2.14 Billion.

```
R1(config)#int multilink 12
```

An IP address is assigned to this logical interface, as follows:

```
R1(config-if)#ip addr 10.1.12.1 255.255.255.0
```

The "PPP Multilink links maximum 2" command states that there should be a maximum of 2 links and the second command states that there should be a minimum of 1 link.

```
R1(config-if)#ppp multilink links maximum 2
R1(config-if)#ppp multilink links minimum 1
```

To verify the configuration:

On R1:

```
R1#Show run int multilink 12 | b interface

interface Multilink12
 ip address 10.1.12.1 255.255.255.0
 ppp multilink
 ppp multilink links maximum 2
 ppp multilink links minimum 1
 ppp multilink group 12
```

The "PPP multilink" command enables the interface to support MLP (multilink point-to-point protocol) and the "PPP multilink group 12" command identifies the multilink group that will later be assigned to two or more interfaces that will restrict them to joining only the designated multilink-group.

The following command creates a virtual-template interface and assigns the multilink group 12 to this logical interface.

```
R1(config)#inter virtual-template 12
R1(config-if)#ppp multilink group 12
```

The virtual-template 12 is assigned to the DLCIs and the frame-relay traffic-shaping is enabled:

```
R1(config-if)#int s0/0
R1(config-if)#encap frame
R1(config-if)#frame-relay traffic-shaping

R1(config-if)#int s0/0.12 multipoint
R1(config-subif)#frame-relay interface-dlci 102 ppp virtual-template 12
R1(config-subif)#frame-relay interface-dlci 112 ppp virtual-template 12

R1(config)#int s0/0
R1(config-if)#No shut
```

On R2:

```
R2(config)#int multilink 21
R2(config-if)#ip addr 10.1.12.2 255.255.255.0
R2(config-if)#ppp multilink links maximum 2
R2(config-if)#ppp multilink links minimum 1

R2(config)#int virtual-template 21
R2(config-if)#ppp multilink group 21

R2(config)#int s0/0
R2(config-if)#encap frame-relay
R2(config-if)#frame-relay traffic-shaping
R2(config-if)#frame-relay interface-dlci 201 ppp virtual-template 21
R2(config-if)#No shut

R2(config-if)#int s0/1
R2(config-if)#encap frame-relay
R2(config-if)#frame-relay traffic-shaping
R2(config-if)#frame-relay interface-dlci 211 ppp virtual-template 21
R2(config-if)#No shut
```

Note: on R2 the virtual-template is assigned to two different physical interfaces, and frame-relay is configured directly under the physical interfaces. This is done intentionally to show the different implementations of this configuration.

To verify the configuration:

On R1:

Note: the multilink 12 logical interface is now up, this is because both routers/end points are configured with PPP multilink.

```
R1#Show ppp multilink

Multilink12
  Bundle name: R2
  Remote Endpoint Discriminator: [1] R2
  Local Endpoint Discriminator: [1] R1
  Bundle up for 00:09:30, total bandwidth 56, load 1/255
  Receive buffer limit 24000 bytes, frag timeout 3428 ms
    0/0 fragments/bytes in reassembly list
    0 lost fragments, 0 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x22 received sequence, 0x22 sent sequence
  Member links: 2 active, 1 inactive (max 2, min 1)
    Vi3, since 00:09:29
```

```
    Vi2, since 00:07:05
    Vt12 (inactive)
No inactive multilink interfaces
```

To verify the configuration:

On R2:

```
R2#Show ppp multilink

Multilink21
  Bundle name: R1
  Remote Endpoint Discriminator: [1] R1
  Local Endpoint Discriminator: [1] R2
  Bundle up for 00:11:00, total bandwidth 56, load 1/255
  Receive buffer limit 24000 bytes, frag timeout 3428 ms
    0/0 fragments/bytes in reassembly list
    1 lost fragments, 1 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x24 received sequence, 0x24 sent sequence
  Member links: 2 active, 1 inactive (max 2, min 1)
    Vi3, since 00:11:00
    Vi2, since 00:08:35
    Vt21 (inactive)
No inactive multilink interfaces
```

To test the configuration:

On R1:

```
R1#Ping 10.1.12.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms

R1#Show ip route | b Gateway

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.12.2/32 is directly connected, Multilink12
C       10.1.12.0/24 is directly connected, Multilink12
```

Note: the host route is installed because of PPP implementation.

To test and verify the configuration:

On R2:

```
R2(config)#int s0/0
R2(config-if)#shut

R2#Show ppp multilink

Multilink21
  Bundle name: R1
  Remote Endpoint Discriminator: [1] R1
  Local Endpoint Discriminator: [1] R2
  Bundle up for 00:15:14, total bandwidth 28, load 1/255
  Receive buffer limit 12000 bytes, frag timeout 3428 ms
    0/0 fragments/bytes in reassembly list
    1 lost fragments, 1 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x36 received sequence, 0x36 sent sequence
  Member links: 1 active, 2 inactive (max 2, min 1)
    Vi3, since 00:15:14
    Vi2 (inactive)
    Vt21 (inactive)
No inactive multilink interfaces
```

Note: ONLY one link is active

To test the configuration:

On R2:

```
R2#Ping 10.1.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/51/52 ms

R2(config)#int s0/0
R2(config-if)#No shut

R2#Show ppp multilink

Multilink21
```

```
  Bundle name: R1
  Remote Endpoint Discriminator: [1] R1
  Local Endpoint Discriminator: [1] R2
  Bundle up for 00:20:49, total bandwidth 56, load 1/255
  Receive buffer limit 24000 bytes, frag timeout 3428 ms
    0/0 fragments/bytes in reassembly list
    1 lost fragments, 1 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x36 received sequence, 0x36 sent sequence
  Member links: 2 active, 1 inactive (max 2, min 1)
    Vi3, since 00:20:49
    Vi2, since 00:00:20
    Vt21 (inactive)
No inactive multilink interfaces
```

Task 2

Configure CHAP authentication between the two routers. Use "**Cisco**" as the password.

On R1:

```
R1(config)#username r2 password Cisco

R1(config)#int virtual-template 12
R1(config-if)#ppp authentication chap
```

On R2:

```
R2(config)#username r1 password Cisco

R2(config)#int virtual-template 21
R2(config-if)#ppp authentication chap
```

Note: the authentication is configured under the virtual-template interface.

To verify the configuration:

On R1:

```
R2#Show ppp multilink

Multilink12
```

```
  Bundle name: R2
  Remote Username: R2
  Remote Endpoint Discriminator: [1] R2
  Local Username: R1
  Local Endpoint Discriminator: [1] R1
  Bundle up for 00:00:47, total bandwidth 56, load 1/255
  Receive buffer limit 24000 bytes, frag timeout 3428 ms
    0/0 fragments/bytes in reassembly list
    0 lost fragments, 0 reordered
    0/0 discarded fragments/bytes, 0 lost received
    0x6 received sequence, 0x10 sent sequence
  Member links: 2 active, 1 inactive (max 2, min 1)
    Vi3, since 00:00:47
    Vi2, since 00:00:44
    Vt12 (inactive)
No inactive multilink interfaces
```

Note: this line is added and it indicates that authentication is configured.

To test the configuration:

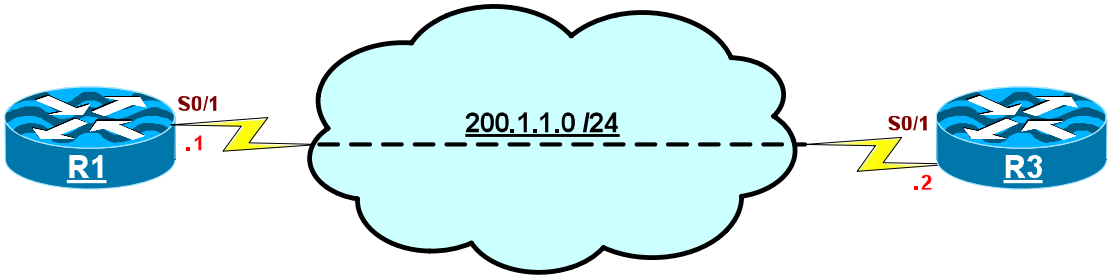
On R2:

```
R2#Ping 10.1.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/55/56 ms
```

Task 3

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 9 – Back-to-Back Frame-Relay Connection



IP addressing:

Router	Interface / IP address	DLCI assignment
R1	S0/1 = 200.1.1.1 /24	113
R3	**S0/1 = 200.1.1.3 /24**	**113**

Task 1

Configure frame-relay between R1 and R3, you should use the IP address, interface and the DLCIs provided in the IP addressing table above.

In this scenario, we do not have a frame-relay switch connecting the routers; these routers are connected back-to-back using a DTE ←→ DCE serial cable. The router that is connected to the DCE side should provide the clocking using the "clock rate" interface configuration command. The DCE side can be determined using the "show controller s 0/1" command as follows:

```
R1#Sh controller s 0/1 | inc clock

DCE V.35, clock rate 64000
```

In this case, since the frame-relay switch does not exist, the LMIs should be disabled using the "no keepalive" interface configuration command, and the frame-relay mapping should be done statically. When configuring the frame-relay mapping, the DLCIs should be identical on both ends.

On R1:

```
R1(config)#interface serial0/1
R1(config-if)#ip address 200.1.1.1 255.255.255.0
R1(config-if)#encapsulation frame-relay
R1(config-if)#NO keepalive
R1(config-if)#clock rate 64000
R1(config-if)#frame-relay map ip 200.1.1.3 113
R1(config-if)#NO shut
```

On R3:

```
R3(config)#interface serial0/1
R3(config-if)#ip address 200.1.1.3 255.255.255.0
R3(config-if)#encapsulation frame-relay
R3(config-if)#NO keepalive
R3(config-if)#frame-relay map ip 200.1.1.1 113
```

To verify & test the configuration:

On R1:

```
R1#Ping 200.1.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.1.1.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

R1#Show frame-relay lmi
R1#
```

Note: there are no LMIs, because they are disabled.

```
R1#Show frame-relay pic

PVC Statistics for interface Serial0/1 (Frame Relay DTE)

                Active      Inactive       Deleted        Static
  Local            1             0             0             0
  Switched         0             0             0             0
  Unused           0             0             0             0

DLCI = 113, DLCI USAGE = LOCAL, PVC STATUS = STATIC, INTERFACE = Serial0/1
```

```
  input pkts 5              output pkts 10           in bytes 520
  out bytes 1040            dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                  out bytes dropped 0
  in FECN pkts 0            in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0           in DE pkts 0             out DE pkts 0
  out bcast pkts 0          out bcast bytes 0
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:03:53, last time pvc status changed 00:02:39

R1#Show frame-relay map

Serial0/1 (up): ip 200.1.1.3 dlci 113(0x71,0x1c10), static,
                      CISCO
```

Task 2

Configure the routers such that R1 uses DLCI 103 to send and DLCI 301 to receive packets. R3 should use DLCI 301 to send and DLCI 103 to receive packets. You should configure interface S0/1 to accomplish this task.

In this task, we are asked to configure these routers to use different DLCIs, 103 connecting R1 to R3 and 301 connecting R3 to R1.

On R1:

```
R1(config)#interface serial0/1
R1(config-if)#ip address 200.1.1.1 255.255.255.0
R1(config-if)#encapsulation frame-relay
R1(config-if)#NO keepalive
R1(config-if)#clock rate 64000
```

The following command removes the frame-relay mapping that was configured in the previous task and adds the new mapping:

```
R1(config-if)#NO frame-relay map ip 200.1.1.3 113
R1(config-if)#frame-relay map ip 200.1.1.3 103
```

On R3:

```
R3(config)#interface serial0/1
R3(config-if)#ip address 200.1.1.3 255.255.255.0
```

```
R3(config-if)#encapsulation frame-relay
R3(config-if)#NO keepalive
R3(config-if)#NO frame-relay map ip 200.1.1.1 113
R3(config-if)#frame-relay map ip 200.1.1.1 301
```

To verify and test the configuration:

On Both Routers:

```
#Debug frame-relay packet
```

On R1:

```
R1#Ping 200.1.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.1.1.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

You should see the following debug output on R1 and R3:

On R1:

```
Serial0/1(o): dlci 103(0x1871), pkt type 0x800(IP), datagramsize 104.
Serial0/1(o): dlci 103(0x1871), pkt type 0x800(IP), datagramsize 104.
Serial0/1(o): dlci 103(0x1871), pkt type 0x800(IP), datagramsize 104.
Serial0/1(o): dlci 103(0x1871), pkt type 0x800(IP), datagramsize 104.
Serial0/1(o): dlci 103(0x1871), pkt type 0x800(IP), datagramsize 104.
```

On R3:

```
Serial0/1: FR invalid/unexpected pak received on DLCI 103
Serial0/1: FR invalid/unexpected pak received on DLCI 103
Serial0/1: FR invalid/unexpected pak received on DLCI 103
Serial0/1: FR invalid/unexpected pak received on DLCI 103
Serial0/1: FR invalid/unexpected pak received on DLCI 103
```

Note: The output of the debug messages on R3 reveals the reason that the ping was not successful. It is telling us that it received 5 invalid and unexpected packets on DLCI 103. The reason the local router (R3) sees R1's DLCI is because they are directly connected.
To fix this problem, R3 can be configured to receive data on DLCI 103 and send on DLCI 301, as follows:

On R3:

```
R3(config)#int s0/1
R3(config-if)#frame-relay interface-dlci 103
```

To verify and test the configuration:

On R1:

```
R1#Ping 200.1.1.3 repeat 4
```

On R3:

```
Serial0/1(i): dlci 103(0x1871), pkt type 0x800, datagramsize 104
Serial0/1(o): dlci 301(0x48D1), pkt type 0x800(IP), datagramsize 104

Serial0/1(i): dlci 103(0x1871), pkt type 0x800, datagramsize 104
Serial0/1(o): dlci 301(0x48D1), pkt type 0x800(IP), datagramsize 104

Serial0/1(i): dlci 103(0x1871), pkt type 0x800, datagramsize 104
Serial0/1(o): dlci 301(0x48D1), pkt type 0x800(IP), datagramsize 104

Serial0/1(i): dlci 103(0x1871), pkt type 0x800, datagramsize 104
Serial0/1(o): dlci 301(0x48D1), pkt type 0x800(IP), datagramsize 104
```

Note: the incoming traffic uses DLCI 103, whereas, the outgoing traffic uses DLCI 301. Let's try to ping R1 and see why the pings are unsuccessful:

To test the configuration:

On R3:

```
R3#Ping 200.1.1.1 repeat 4
```

On R1:

```
Serial0/1: FR invalid/unexpected pak received on DLCI 301
Serial0/1: FR invalid/unexpected pak received on DLCI 301
Serial0/1: FR invalid/unexpected pak received on DLCI 301
Serial0/1: FR invalid/unexpected pak received on DLCI 301
```

Note: we are experiencing the same problem on R3, the traffic comes in on DLCI 301 and the local router is not aware of this DLCI. To fix this problem:

```
R1(config)#int s0/1
R1(config-if)#frame-relay interface-dlci 301
```

To verify and test the configuration:

On R3:

```
R3#Ping 200.1.1.1 repeat 4

Type escape sequence to abort.
Sending 4, 100-byte ICMP Echos to 200.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (4/4), round-trip min/avg/max = 28/29/32 ms
```

On R1:

```
Serial0/1(i): dlci 301(0x48D1), pkt type 0x800, datagramsize 104
Serial0/1(o): dlci 103(0x1871), pkt type 0x800(IP), datagramsize 104

Serial0/1(i): dlci 301(0x48D1), pkt type 0x800, datagramsize 104
Serial0/1(o): dlci 103(0x1871), pkt type 0x800(IP), datagramsize 104

Serial0/1(i): dlci 301(0x48D1), pkt type 0x800, datagramsize 104
Serial0/1(o): dlci 103(0x1871), pkt type 0x800(IP), datagramsize 104

Serial0/1(i): dlci 301(0x48D1), pkt type 0x800, datagramsize 104
Serial0/1(o): dlci 103(0x1871), pkt type 0x800(IP), datagramsize 104

R1#Show frame map

Serial0/1 (up): ip 200.1.1.3 dlci 103(0x67,0x1870), static,
                      CISCO
```

On R3:

```
R3#Show frame map

Serial0/1 (up): ip 200.1.1.1 dlci 301(0x12D,0x48D0), static,
                      CISCO
```

Task 3

Re-configure R1 as a frame-relay switch and a router connecting to R3, whereas, R3 should be configured as a router connecting to R1 using S0/1 interface. R1 should use DLCI 103 for its connection to R3 and R3 should use DLCI 301 for its connection to R1. You should **not** disable LMIs to accomplish this task.

On R1:

```
R1(config)#frame switching

R1(config)#int s0/1
R1(config-if)#ip addr 200.1.1.1 255.255.255.0
R1(config-if)#encap frame-relay
R1(config-if)#clock rate 64000

R1(config-if)#frame map ip 200.1.1.3 103
R1(config-if)#frame interface-dlci 301
R1(config-if)#frame-relay intf-type dce
```

On R3:

```
R3(config-if)#int s0/1
R3(config-if)#ip addr 200.1.1.3 255.255.255.0
R3(config-if)#encap frame-relay
R3(config-if)#frame map ip 200.1.1.1 301
```

To verify and test the configuration:

On R1:

```
R1#Show frame lmi | b Num

  Num Status Enq. Rcvd 11                 Num Status msgs Sent 11
  Num Update Status Sent 0                Num St Enq. Timeouts 0
```

On R3:

```
R3#Show frame-relay lmi | b Num

  Num Status Enq. Sent 18                 Num Status msgs Rcvd 19
  Num Update Status Rcvd 0                Num Status Timeouts 0
  Last Full Status Req 00:00:00           Last Full Status Rcvd 00:00:00


R3#Show frame-relay map

Serial0/1 (up): ip 200.1.1.1 dlci 301(0x12D,0x48D0), static,
              CISCO, status defined, active

R3#Ping 200.1.1.1
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/33 ms
```

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Advanced
CCIE SERVICE PROVIDER
v3.0

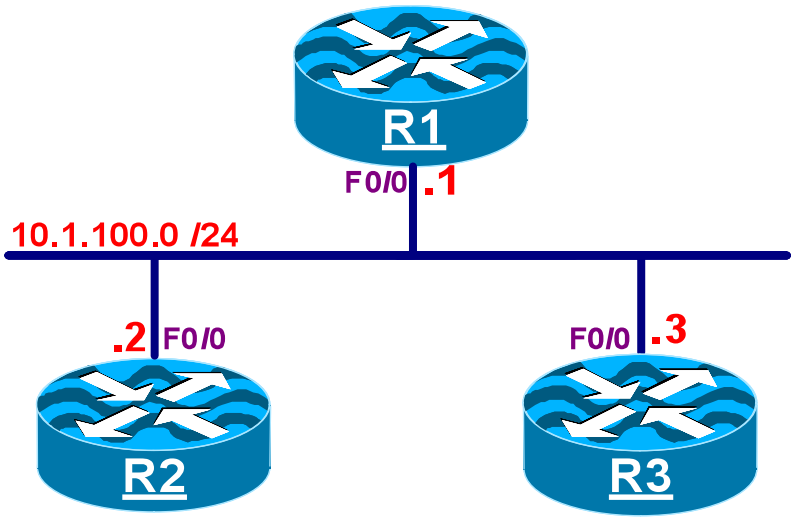
www.MicronicsTraining.com

Narbik Kocharians
CCIE #12410
R&S, Security, SP

Paul Negron
CCIE #14856
SP

PPPoE

Lab 1
Configuring PPPoE



Task 1

Configure PPP on the Ethernet link connecting R1 to R2. R1 should be the server and R2 should be configured as the client. You should configure 0000.1111.1111 and 0000.2222.2222 as the MAC address of R1 and R2's F0/0 respectively.

On R1

```
R1(config)#int virtual-template 1
R1(config-if)#ip addr 10.1.100.1 255.255.255.0

R1(config)#bba-group pppoe TST
R1(config-bba-group)#virtual-template 1

R1(config-bba-group)#int f0/0
R1(config-if)#pppoe enable group TST
R1(config-if)#mac-address 0000.1111.1111
```

On R2

```
R2(config)#Int dialer 2
R2(config-if)#ip address 10.1.100.2 255.255.255.0
R2(config-if)#encapsulation ppp
R2(config-if)#dialer pool 21
```

```
R2(config-if)#int f0/0
R2(config-if)#pppoe-client dial-pool-number 21
R2(config-if)#mac-address 0000.2222.2222
```

By default the "pppoe enable" command is inserted by the IOS:

```
R2#Show run int f0/0 | B inter

interface FastEthernet0/0
 mac-address 0000.2222.2222
 no ip address
 duplex auto
 speed auto
 pppoe enable
 pppoe-client dial-pool-number 21
```

To verify the configuration:

On R2

```
R2#Show ip route | Inc /32

C       10.1.100.1/32 is directly connected, Dialer2
```

On R1

```
R1#Show ip route | Inc /32

C       10.1.100.2/32 is directly connected, Virtual-Access2.1
```

To test the configuration:

```
R1#Ping 10.1.100.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#Show pppoe session

     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID   PPPoE   RemMAC           Port                 VT  VA          State
            SID   LocMAC                                    VA-st
      2       2   0000.2222.2222   Fa0/0                 1  Vi2.1       PTA
                  0000.1111.1111                             UP
```

DO NOT ERASE THE CONFIGURATION OF THESE ROUTERS.

Task 2

Configure R1 to assign an IP address to R2; R1 should assign an IP address to R2 from the following range:

10.1.100.2 – 10.1.100.254 /24

On R1

```
R1(config)#IP local pool TST 10.1.100.2 10.1.100.254

R1(config)#int virtual-template 1
R1(config-if)#peer default ip address pool TST
```

On R2

Before configuring this part of the task, "Debug PPP Negotiated" is configured on R2:

```
R2#Deb ppp negotiation

R2(config)#int dialer 2
R2(config-if)#ip address negotiated
R2(config-if)#shu

R2(config-if)#NO Shut
```

You should see the following debug output:

```
Vi2 IPCP: State is Open
Di2 IPCP: Install negotiated IP interface address 10.1.100.2
Di2 IPCP: Install route to 10.1.100.1
Vi2 IPCP: Add link info for cef entry 10.1.100.1
```

To verify the configuration:

On R2

```
R2#Show ip int brief  dialer 2

Interface              IP-Address      OK? Method Status                Protocol
Dialer2                10.1.100.2      YES IPCP   up                    up
```

DO NOT ERASE THE CONFIGURATION OF THESE ROUTERS.

Task 3

Configure authentication between R1 and R2. These routers should use the strongest authentication method using "cisco" as the password and their hostname as the "username" for this authentication.

On R1

```
R1(config)#username R2 password cisco

R1(config)#int virtual-template 1
R1(config-if)#ppp authentication chap
R1(config-if)#ppp chap hostname R1
```

On R2

```
R2(config)#username R1 password cisco

R2(config)#int dialer 2
R2(config-if)#ppp authentication chap
R2(config-if)#ppp chap hostname R2
```

To verify the configuration

On R1

```
R1#Clear pppoe all
```

On R2

```
R2#Show ip int brie dialer 2

Interface              IP-Address      OK? Method Status                Protocol
Dialer2                10.1.100.3      YES IPCP   up                    up
```

DO NOT ERASE THE CONFIGURATION OF THESE ROUTERS.

Task 4

Add R3 to the same Ethernet segment and configure this router as an PPPoE clients; this router should use a MAC address of 0000.3333.3333 and acquire an IP address from R1. DO NOT change R1's configuration to accomplish this task.

On R3

```
R3(config)#username R1 password cisco

R3(config)#int dialer 3
R3(config-if)#ip address negotiated
R3(config-if)#encapsulation ppp
R3(config-if)#ppp authentication chap
R3(config-if)#ppp chap hostname R3
R3(config-if)#dialer pool 31

R3(config)#int f0/0
R3(config-if)#pppoe-client dial-pool-number 31
R3(config-if)#mac-address 0000.3333.3333
R3(config-if)#NO shut
```

To verify the configuration:

On R3

```
R3#Show ip int brief dialer 3

Interface              IP-Address      OK? Method Status                Protocol
Dialer3                10.1.100.4      YES IPCP   up                    up
```

On R1

```
R1#Ping 10.1.100.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

R1#Ping 10.1.100.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

```
R1#Show pppoe session

     2 sessions in LOCALLY_TERMINATED (PTA) State
     2 sessions total

Uniq ID   PPPoE   RemMAC           Port                      VT   VA          State
            SID   LocMAC                                       VA-st
      8       6   0000.2222.2222   Fa0/0                      1   Vi2.1       PTA
                  0000.1111.1111                                  UP
     12      10   0000.3333.3333   Fa0/0                      1   Vi2.2       PTA
                  0000.1111.1111                                  UP
```

DO NOT ERASE THE CONFIGURATION OF THESE ROUTERS.

Task 5

Configure R1 to limit the number of sessions to 1.

The task is NOT asking for the following Mac addresses to be configured, this is done to determine with which PPPoE client the local router (R1) has established a session:

On R1

```
R1(config)#bba-group pppoe global
R1(config-bba-group)#sessions max limit 1
```

To verify the configuration:

On R1

```
R1#Clear pppoe all

R1(config)#int f0/0
R1(config-if)#shu

R1(config-if)#NO shu

R1#Show pppoe session


     1 session  in LOCALLY_TERMINATED (PTA) State
     1 session  total

Uniq ID   PPPoE   RemMAC           Port                      VT   VA          State
            SID   LocMAC                                       VA-st
```

```
13      11  0000.2222.2222  Fa0/0                    1  Vi2.1       PTA
            0000.1111.1111                              UP
```

Task 6

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Advanced
CCIE SERVICE PROVIDER
v3.0

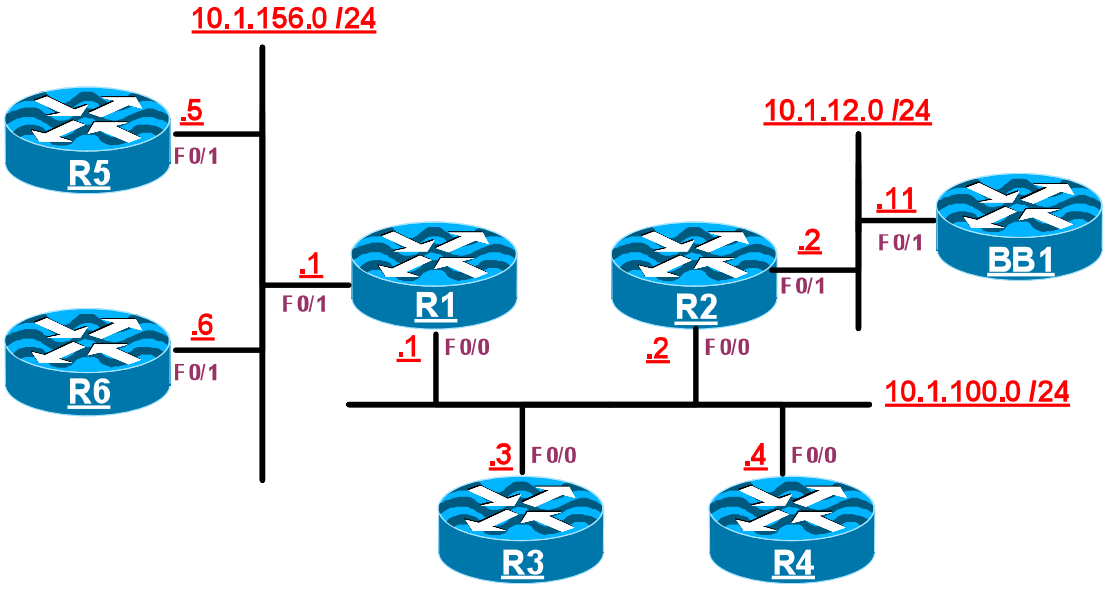
www.MicronicsTraining.com

Narbik Kocharians
CCIE #12410
R&S, Security, SP

Paul Negron
CCIE #14856
SP

Multicasting

Lab 1 – Configuring IGMP



Lab Setup:

- Configure the F0/0 interface of R1, R2, R3 and R4 in VLAN 100.
- Configure the F0/1 interface of R1, R5 and R6 in VLAN 200.
- Configure the F0/1 interface of R2 and BB1 in VLAN 300.
- Configure the IP addressing based on the diagram.

Task 1

Enable Multicast routing on R1 and R2 and configure their F0/0 and F0/1 interfaces in PIM Dense mode.

On Both Routers

IP multicast routing is disabled by default and needs to be enabled using the following command.

```
Rx(config)#ip multicast-routing
```

The following commands enable PIM Dense mode for the requested interfaces; you MUST have the "IP multicast-routing" configured for the "IP Pim dense-mode" command to work, if the "IP Multicast-routing" is NOT configured, you will receive a warning console message stating the following:

WARNING: "ip multicast-routing" is not configured,
IP Multicast packets will not be forwarded

To enable PIM Dense mode:

```
Rx(config)#int f0/0
Rx(config-if)#ip pim dense-mode

Rx(config-if)#int f0/1
Rx(config-if)#ip pim dense-mode
```

To verify the configuration:

On R1

```
R1#Show ip pim interface

Address          Interface                Ver/   Nbr    Query  DR     DR
                                          Mode   Count  Intvl  Prior
10.1.100.1       FastEthernet0/0          v2/D   1      30     1      10.1.100.2
10.1.156.1       FastEthernet0/1          v2/D   0      30     1      10.1.156.1
```

On R2

```
R2#Show ip pim interface

Address          Interface                Ver/   Nbr    Query  DR     DR
                                          Mode   Count  Intvl  Prior
10.1.100.2       FastEthernet0/0          v2/D   1      30     1      10.1.100.2
```

```
10.1.12.2         FastEthernet0/1           v2/D   0       30      1       10.1.12.2
```

NOTE: R2 is the DR on Both segments (F0/0 and F0/1).

```
R2#Show ip mroute | b out

Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:01:25/00:02:48, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 00:01:25/00:00:00
```

On R1

```
R1#Show ip mroute | b out

Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:02:50/00:02:37, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 00:02:50/00:00:00
```

When PIM is enabled on a given interface, IGMP is also enabled. NOTE: The (*,224.0.1.40) entry is always created when PIM is enabled on a Cisco router, each router will start generating V2 General Queries on their F0/0 interface for group 224.0.1.40.

To verify the configuration:

On R1

```
R1#Show ip igmp interface f0/0

FastEthernet0/0 is up, line protocol is up
  Internet address is 10.1.100.1/24
  IGMP is enabled on interface ------------------------------Line 1
  Current IGMP host version is 2 ----------------------------Line 2
  Current IGMP router version is 2
  IGMP query interval is 60 seconds -------------------------Line 3
  IGMP querier timeout is 120 seconds -----------------------Line 4
```

```
IGMP max query response time is 10 seconds ----------------Line 5
Last member query count is 2
Last member query response interval is 1000 ms
Inbound IGMP access group is not set
IGMP activity: 1 joins, 0 leaves
Multicast routing is enabled on interface
Multicast TTL threshold is 0
Multicast designated router (DR) is 10.1.100.2 -----------Line 6
IGMP querying router is 10.1.100.1 (this system) ---------Line 7
Multicast groups joined by this system (number of users):
    224.0.1.40(1)
```

Line 1 and 2:

By default when PIM is configured for an interface, IGMPv2 is also enabled; this can be changed using the "ip igmp version" command.

Line 3 and 4:

The query interval, this command configures the frequency at which the IGMP querier sends IGMP-Host-Query messages through this interface; these messages are sent by the querier in order to discover which Multicast groups have members on the interface. The default value is 60 seconds; this value can be changed using the "ip igmp query-interval" interface configuration command.

By default, if the querier misses two Queries in a row, the other router/s on the subnet will trigger an election to elect a new querier; therefore, when the IP IGMP Query-interval is changed, the IOS will automatically change the IGMP querier timeout value to twice the query interval. However, the querier timeout can be changed using the "ip igmp querier-timeout" interface command.

Line 5:

By default the IGMP max query response time is set to 10 seconds; in igmp version 2, this counter is advertised in igmp queries to the hosts, informing them of the maximum time within which they must respond to a general query, this improves the burstiness of the responses. This default value can be changed using the interface configuration "IP igmp query-max-response-time" command.

Line 6 and 7:

Note the querier is responsible for forwarding the multicast flow. IGMPv1 did NOT have a querier election, therefore, it was the decision of the Multicast routing protocol to elect a DR for this purpose. In IGMPv2, a formal querying router election process was specified within the IGMPv2 protocol. In IGMPv2 each router on a multi-access network will initially assume that it is the querier and begins by sending queries, each router connected to that multi-access network will see the queries from the other IGMPv2 router and will examine the source IP address of the query messages. All IGMPv2 routers will then elect the router with the lowest source IP address as the IGMP querier. If the elected router fails to send query messages within a specified time limit, the routers on that multi-access network will re-initiate the query election once again.
The DR concept will be covered in later labs.

Task 2

Disable IP Routing on R3 and R4 and configure their F0/0 interface to join 224.1.1.1 multicast group.

On R3 and R4

```
Rx(config)#NO ip routing
```

There are two commands that are somewhat identical yet different, these are the "IP igmp static-group" and "IP igmp join-group" commands.

The "ip igmp static-group" command configures a static group membership entry on an interface, which allows the router to join the multicast group. This configuration of the "IP igmp static-group" command would cause the upstream router/s to maintain the multicast routing table information for that group, which would ensure that all the paths to that multicast group are active. Remember that this command does NOT process the ping packets, it just joins the group and floods the Multicast flow.

The "IP igmp join-group" command allows the router to process and respond to ping commands. This can be a useful administrative and debugging tool. This command can configure the router to emulate a client connected to a last hop router, whereas, the ping command performed on a router can configure the router to act as a server.

On R3 and R4

```
Rx(config)#int f0/0
Rx(config-if)#ip igmp join-group 224.1.1.1
```

To verify the configuration:

On R1

```
R1#Show ip igmp groups

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.1.1.1        FastEthernet0/0          00:02:18  00:02:32  10.1.100.3
224.0.1.40       FastEthernet0/0          01:06:03  00:02:27  10.1.100.2
```

NOTE: R3 (10.1.100.3) was the last host to report being a member of the 224.1.1.1 Multicast group. The "Uptime" column specifies how long the multicast group has been known, the "Expire" column specifies how long until the entry expires, the entry starts with 2 minutes and 59 seconds and it counts down all the way to 2:00 minutes and then, back to 2:59, but remember that every 60 seconds the querier sends IGMP-host-query messages and if there is an active group member, it

will reply within 10 seconds, therefore, this counter should NOT go below 2 minutes unless there are no active group members.

Task 3

Disable IP Routing on R5 and R6 and configure their F0/1 interface to join 224.56.56.56 multicast group. R1's F0/1 interface should be configured for PIM Dense mode.

On R5 and R6

```
Rx(config)#NO ip routing

Rx(config)#int f0/1
Rx(config-if)#ip igmp join-group 224.56.56.56
```

To verify the configuration:

On R1

```
R1#Show ip igmp groups

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.56.56.56     FastEthernet0/1          00:02:38  00:02:05  10.1.156.6
224.1.1.1        FastEthernet0/0          00:21:41  00:02:06  10.1.100.3
224.0.1.40       FastEthernet0/0          01:25:26  00:02:11  10.1.100.1
```

On R5

```
R5#Sho ip igmp inter f0/1 | b multicast groups

  Multicast groups joined by this system (number of users):
      224.56.56.56(1)

R5#Sh ip igmp groups

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.56.56.56     FastEthernet0/1          00:06:10  stopped   10.1.156.5
```

NOTE: Since the router's interface is configured with an IGMP join-group command the "Expire" column shows as "Stopped".

Task 4

Disable IP Routing on BB1 and configure its F0/1 interface to join 224.2.2.2 multicast group.

On BB1

```
BB1(config)#NO ip routing

BB1(config)#int f0/1
BB1(config-if)#ip igmp join-group 224.2.2.2
```

To verify the configuration:

On BB1

```
BB1#Show ip igmp interface F0/1 | b Multicast groups
```

Multicast groups joined by this system (number of users):
224.2.2.2(1)

On R2

```
R2#Show ip igmp groups | exc 224.0.1.40

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.2.2.2        FastEthernet0/1          00:00:42  00:02:17  10.1.12.10
224.1.1.1        FastEthernet0/0          00:31:42  00:02:07  10.1.100.3
```

Task 5

Configure R1 to restrict hosts connected to its F0/1 (VLAN 200) from joining 224.5.5.5 and 224.6.6.6 multicast groups.

Note the following standard access-list denies the two groups and allows the others:

On R1

```
R1(config)#ip access-list standard TST
R1(config-std-nacl)#deny 224.5.5.5
R1(config-std-nacl)#deny 224.6.6.6
```

```
R1(config-std-nacl)#permit any
```

The Access-list is applied to the F0/1 interface of R1

```
R1(config)#int f0/1
R1(config-if)#ip igmp access-group TST
```

The following is turned on for verification purpose:

```
R1#Debug ip igmp
```

To verify and test the configuration

On R5

```
R5(config)#int f0/1
R5(config-if)#ip igmp join-group 224.5.5.5
```

On R6

```
R6(config)#int f0/1
R6(config-if)#ip igmp join-group 224.6.6.6
```

On R1

Note you should see the following messages in the output of the debug:

```
Received v2 Report on FastEthernet0/1 from 10.1.156.5 for 224.5.5.5
Group 224.5.5.5 access denied on FastEthernet0/1

Received v2 Report on FastEthernet0/1 from 10.1.156.6 for 224.6.6.6
Group 224.6.6.6 access denied on FastEthernet0/1
```

```
R1#Show ip igmp groups | exc 224.0.1.40

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.56.56.56     FastEthernet0/1          00:27:54  00:02:57  10.1.156.6
224.1.1.1        FastEthernet0/0          00:46:56  00:02:51  10.1.100.4

R1#Show ip igmp interface f0/1

FastEthernet0/1 is up, line protocol is up
  Internet address is 10.1.156.1/24
  IGMP is enabled on interface
```

```
  Current IGMP host version is 2
  Current IGMP router version is 2
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query count is 2
  Last member query response interval is 1000 ms
  Inbound IGMP access group is TST
  IGMP activity: 1 joins, 0 leaves
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 10.1.156.1 (this system)
  IGMP querying router is 10.1.156.1 (this system)
  No multicast groups joined by this system
```

On R1

```
R1#U all
```

Task 6

Since there is ONLY a single host connected to the F0/1 interface of R2, R2 should be configured such that it stops forwarding multicast traffic for all groups immediately upon receipt of an IGMPv2 group leave message.

On R2

```
R2(config)#int f0/1
R2(config-if)#ip igmp immediate-leave group-list 1

R2(config)#access-list 1 permit 224.0.0.0 15.255.255.255
```

Note if the "IP Igmp immediate-leave group-list 1" is configured in the global configuration mode, it will apply to all interfaces, in this case it is applied to F0/1 interface, therefore, it effects hosts that are connected to F0/1 interface of R1. Note the access-list matches all class D addresses, therefore, affecting all Multicast groups.

Task 7

Configure R1 such that before it stops forwarding the Multicast traffic, it sends three Igmp query messages at 500 ms intervals after receiving an Igmp group specific leave message.

To see the default value:

On R1

```
R1#Show ip igmp interface f0/0

FastEthernet0/0 is up, line protocol is up
  Internet address is 10.1.100.1/24
  IGMP is enabled on interface
  Current IGMP host version is 2
  Current IGMP router version is 2
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query count is 2
  Last member query response interval is 1000 ms
  Inbound IGMP access group is not set
  IGMP activity: 5 joins, 0 leaves
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 10.1.156.1
  IGMP querying router is 10.1.12.2
  Multicast groups joined by this system (number of users):
      224.0.1.40(1)
```

Note the default Last Member Query Count is set to two, the following command sets the LMQC to three:

On R1

```
R1(config)#int f0/0
R1(config-if)#ip igmp last-member-query-count 3
```

To verify the configuration:

On R1

```
R1#Show ip igmp interface f0/0 | inc last member query count
```

```
 Last member query count is 3
```

To set the interval for these messages:

On R1

```
R1(config)#int f0/0
R1(config-if)#ip igmp last-member-query-interval 500
```

To verify the configuration:

On R1

```
R1#Show ip igmp interface f0/0 | inc last member query response

  Last member query response interval is 500 ms
```

Task 8

Configure R2 such that the number of mroute states created because of host membership reports to 3. This policy should ONLY affect R2's F0/1 interface.

The following solution can be applied in Global configuration mode or interface configuration mode, when it's configured in Global config mode, it's applied to the entire router and it's referred to as "Global IGMP State Limiter" which means that it effects the router globally. However, if it is configured in the interface config mode, the effect is for the hosts connected to that interface.

On R2

```
R2(config)#int f0/1
R2(config-if)#ip igmp limit 3
```

To verify the configuration:

On R2

```
R2#Show ip igmp inter f0/1

FastEthernet0/1 is up, line protocol is up
  Internet address is 10.1.12.2/24
  IGMP is enabled on interface
```

```
  Current IGMP host version is 2
  Current IGMP router version is 2
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query count is 2
  Last member query response interval is 1000 ms
  Inbound IGMP access group is not set
  IGMP activity: 10 joins, 5 leaves
  Interface IGMP State Limit : 1 active out of 3 max
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 10.1.156.1
  IGMP querying router is 10.1.12.2 (this system)
  No multicast groups joined by this system
```

NOTE: the highlighted section states that there is 1 active group out of a maximum of 5 groups, the output of the following Show command reveals that there are 2 groups:

```
R2#Show ip igmp groups f0/1

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.2.2.2        FastEthernet0/1          00:01:26  00:02:36  10.1.12.10
```

To test the configuration:

On BB1

In testing the solution, BB1 is configured to join additional groups and the mroute state is verified on R2, once the limit is reached, R2 denies creating additional mroute states:

```
BB1(config)#int f0/1
BB1(config-if)#ip igmp join-group 224.22.22.22
```

On R2

```
R2#Sh ip igmp group f0/1

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.22.22.22     FastEthernet0/1          00:00:09  00:02:50  10.1.12.10
224.2.2.2        FastEthernet0/1          00:02:48  00:02:08  10.1.12.10
```

NOTE: An mroute state is created for group 224.22.22.22:

```
R2#Show ip mroute | inc 224.22.22.22

 (*, 224.22.22.22), 00:02:15/00:02:02, RP 0.0.0.0, flags: DC
```

On BB1

Note the following is the third group:

```
BB1(config)#int f0/1
BB1(config-if)#ip igmp join-group 224.222.222.222
```

On R2

```
R2#Sh ip igmp group f0/1

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.22.22.22     FastEthernet0/1          00:01:57  00:02:03  10.1.12.10
224.2.2.2        FastEthernet0/1          00:04:51  00:02:00  10.1.12.10
224.222.222.222  FastEthernet0/1          00:00:16  00:02:43  10.1.12.10

R2#Sh ip mroute | inc 224.222.222.222

 (*, 224.222.222.222), 00:00:58/00:02:01, RP 0.0.0.0, flags: DC
```

Note R2 has created an mroute state and it has reached its maximum allowable mroute states. To see the effect of the "IP Igmp limit" command, the "Debug ip igmp" is configured on R2, as follows:

On R2

```
R2#Debug ip igmp
```

On BB1

Note BB1 is configured with the forth group:

```
BB1(config)#int f0/1
BB1(config-if)#ip igmp join-group 224.220.220.220
```

On R2

You should receive the following debug output:

```
Received v2 Report on FastEthernet0/1 from 10.1.12.10 for 224.220.220.220
Received Group record for group 224.220.220.220, mode 2 from 10.1.12.1
```

```
Group 224.220.220.220 access denied on FastEthernet0/1
```

```
R2#Show ip igmp groups f0/1
```

NOTE: There are seven groups, which is the maximum allowed number:

```
IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.22.22.22     FastEthernet0/1          00:03:46  00:02:12  10.1.12.10
224.2.2.2        FastEthernet0/1          00:08:50  00:02:07  10.1.12.10
224.222.222.222  FastEthernet0/1          00:02:51  00:02:12  10.1.12.10
```

```
R2#Sh ip igmp inter f0/1

FastEthernet0/1 is up, line protocol is up
  Internet address is 10.1.12.2/24
  IGMP is enabled on interface
  Current IGMP host version is 2
  Current IGMP router version is 2
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query count is 2
  Last member query response interval is 1000 ms
  Inbound IGMP access group is not set
  IGMP activity: 12 joins, 5 leaves
  Interface IGMP State Limit : 3 active out of 3 max
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 10.1.156.1
  IGMP querying router is 10.1.12.2 (this system)
  No multicast groups joined by this system
```

On R2

R2#**U all**

Task 9

Configure R1 and R2 such that if the existing querier is down for longer than 90 seconds a new querier is elected for 10.1.100.0 /24 network.

The output of the following show command displays the default value:

```
R1#Sh ip igmp inter f0/0
```

```
FastEthernet0/0 is up, line protocol is up
  Internet address is 10.1.100.1/24
  IGMP is enabled on interface
  Current IGMP host version is 2
  Current IGMP router version is 2
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query count is 3
  Last member query response interval is 1000 ms
  Inbound IGMP access group is not set
  IGMP activity: 8 joins, 0 leaves
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 10.1.156.1
  IGMP querying router is 10.1.12.2
  Multicast groups joined by this system (number of users):
      224.0.1.40(1)
```

To configure this task:

On R1 and R2

NOTE: By default, if the querier is down, the other routers on the same subnet will wait twice the query interval specified by the "Ip igmp query-interval" before the election is reinitiated, the following command changes the wait time to 90 seconds:

```
Rx(config)#int f0/0
Rx(config-if)#ip igmp querier-timeout 90
```

To verify the configuration:

On R1

```
R1#Show ip igmp inter f0/0 | inc igmp que

  IGMP query interval is 60 seconds
  IGMP querier timeout is 90 seconds
  IGMP querying router is 10.1.12.2
```

On R2

```
R2#Show ip igmp inter f0/0 | inc igmp que

IGMP query interval is 60 seconds
```

```
  IGMP querier timeout is 90 seconds
  IGMP querying router is 10.1.12.2
```

To verify and test the configuration:

On R1 and R2

```
R1(config)#int f0/1
R1(config-if)#shut
```

To see the existing querier:

On R2

```
R2#Sh ip igmp inter f0/0 | inc igmp que

  IGMP query interval is 60 seconds
  IGMP querier timeout is 90 seconds
  IGMP querying router is 10.1.100.1
```

NOTE: the existing querier is R1 (10.1.100.1), let's shut down the F0/0 interface of R1 and see the change in 90 seconds:

On R1

```
R1(config-if)#int f0/0
R1(config-if)#shut
```

NOTE: The following show command is entered after 60 seconds, and as you can see, R1 is still the querier:

```
R2#Sh ip igmp inter f0/0 | inc igmp que

  IGMP query interval is 60 seconds
  IGMP querier timeout is 90 seconds
  IGMP querying router is 10.1.100.1
```

The following show command is entered after 91 seconds:

Note the querier changed after 90 seconds

```
R2#Sh ip igmp inter f0/0 | inc igmp que

  IGMP query interval is 60 seconds
  IGMP querier timeout is 90 seconds
  IGMP querying router is 10.1.100.2 (this system)
```

To reconfigure R1 as the querier:

On R1

```
R1(config)#int f0/0
R1(config-if)#NO shut
```

To verify the configuration:

On R2

Once the F0/0 interface of R1 is UP, R1 will take over the querier responsibilities:

```
R2#Show ip igmp inter f0/0 | inc igmp que

  IGMP query interval is 60 seconds
  IGMP querier timeout is 90 seconds
  IGMP querying router is 10.1.100.1
```

Task 10

R1 and R2 should be configured to advertise the period during which the responder can respond to an Igmp query message before these routers delete the group to its maximum allowable value.

This is controlled through the "Ip Igmp query-max-response-time" interface configuration command.

On R1 and R2

```
R2#Show ip igmp inter f0/0 | inc igmp max

  IGMP max query response time is 10 seconds
```

Note 10 seconds is the default, the following command changes this value; the range is 1 – 25 seconds:

On R1 and R2

```
Rx(config)#int f0/0
Rx(config-if)#ip igmp query-max-response-time 25
```

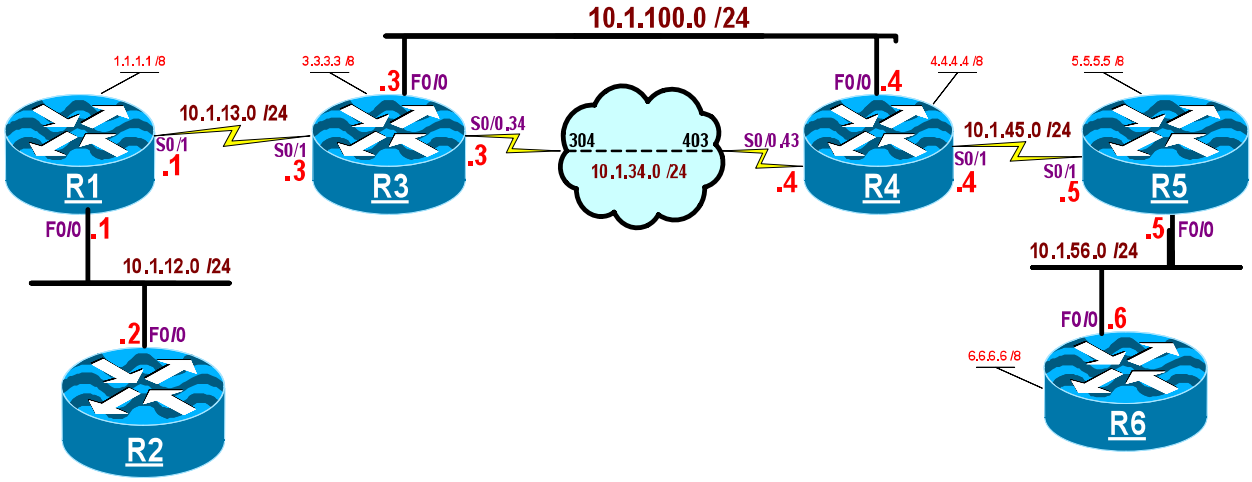
To verify the configura max

```
Rx#Show ip igmp inter f0/0 | inc igmp max
  IGMP max query response time is 25 seconds
```

Task 11

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 2 – Dense mode



Lab Setup:

- Configure the F0/0 interface of R1 and R2 in VLAN 100, the F0/0 interface of R3 and R4 in VLAN 200 and the F0/0 interface of R5 and R6 in VLAN 300.

- The frame-Relay connection between R3 and R4 should be configured in a Point-to-point manner.
- The connection between R1 and R3, and R4 and R5 should be configured in HDLC.
- Use the IP addressing chart below for IP address assignment.

IP addressing Chart:

Router	Interface / IP addressing
R1	F0/0 = 10.1.12.1 /24 S0/1 = 10.1.13.1 /24 Lo0 = 1.1.1.1 /8
R2	**F0/0 = 10.1.12.2/24** **Lo0 = 2.2.2.2 /8**
R3	S0/1 = 10.1.13.3 /24 S0/0.34 = 10.1.34.3 /24 F0/0 = 10.1.100.3 /24 Lo0 = 3.3.3.3 /8
R4	**S0/0.43 = 10.1.34.4 /24** **F0/0 = 10.1.100.4 /24** **S0/1 = 10.1.45.4 /24** **Lo0 = 4.4.4.4 /8**
R5	S0/1 = 10.1.45.5 /24 F0/0 = 10.1.56.5 /24 Lo0 = 5.5.5.5 /8
R6	**F0/0 = 10.1.56.6 /24** **Lo0 = 6.6.6.6 /8**

Task 1

Enable OSPF area 0 on all interfaces.

On All Routers

```
Rx(config)#router ospf 1
Rx(config-router)#Netw 0.0.0.0 0.0.0.0 area 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf | inc O

O       3.3.3.3 [110/65] via 10.1.13.3, 00:01:01, Serial0/1
O       4.4.4.4 [110/66] via 10.1.13.3, 00:00:22, Serial0/1
O       5.5.5.5 [110/130] via 10.1.13.3, 00:00:22, Serial0/1
O       6.6.6.6 [110/131] via 10.1.13.3, 00:00:12, Serial0/1
O       10.1.45.0 [110/129] via 10.1.13.3, 00:00:22, Serial0/1
O       10.1.34.0 [110/128] via 10.1.13.3, 00:01:01, Serial0/1
```

```
O       10.1.56.0 [110/130] via 10.1.13.3, 00:00:22, Serial0/1
O       10.1.100.0 [110/65] via 10.1.13.3, 00:00:22, Serial0/1
```

Task 2

Configure PIM on the interfaces according to the following table:

Router	Interface / PIM
R1	F0/0 = PIM Dense mode S0/1 = PIM Dense mode Lo0 = PIM Dense mode
R3	**S0/1 = PIM Dense mode** **S0/0.34 = PIM Dense mode** **Lo0: PIM Dense mode**
R4	S0/0.43 = PIM Dense mode S0/1 = PIM Dense mode Lo0 = PIM Dense mode
R5	**F0/0 = PIM Dense mode** **S0/1 = PIM Dense mode** **Lo0: PIM Dense mode**
R6	F0/0 = PIM Dense mode Lo0 = PIM Dense mode

On R1:

By default, Multicast routing is disabled on Cisco routers, and it must be enabled using the following command:

```
R1(config)#ip multicast-routing

R1(config)#int lo0
R1(config-if)#ip pim dense-mode

R1(config-if)#int s0/1
R1(config-if)#ip pim dense-mode

R1(config-if)#int f0/0
R1(config-if)#ip pim dense-mode
```

Note if "IP Multicast-routing" is NOT configured, the "ip pim dense-mode" will NOT have any effect, and the following message will be received:

WARNING: "ip multicast-routing" is not configured, IP Multicast packets will not be forwarded

When "IP multicast-routing" is configured and PIM is enabled on interface/s, IGMP version 2 is automatically enabled on the interfaces that are PIM enabled:

```
R1#Sh ip igmp interface | inc line proto

Loopback0 is up, line protocol is up
Serial0/1 is up, line protocol is up
FastEthernet0/0 is up, line protocol is up

R1#Sh ip igmp interface | inc current

  Current IGMP host version is 2
  Current IGMP router version is 2
  Current IGMP host version is 2
  Current IGMP router version is 2
  Current IGMP host version is 2
  Current IGMP router version is 2
```

The reason we have 6 entries is that there are 3 interfaces.

On R3

```
R3(config)#ip multicast-routing

R3(config-if)#int s0/1
R3(config-subif)#ip pim dense-mode

R3(config-if)#int s0/0.34
R3(config-subif)#ip pim dense-mode

R3(config-if)#int lo0
R3(config-subif)#ip pim dense-mode
```

To verify the configuration:

On R3

```
R3#Sh ip igmp inter | inc line

Loopback0 is up, line protocol is up
Serial0/1 is up, line protocol is up
Serial0/0.34 is up, line protocol is up
```

On R1 and R3

```
R3#Debug ip mpacket
```

Note when "Ip multicast-routing" is enabled and the interface facing R1 is configured with "Ip pim dense-mode", R3 and R1 will establish a neighbor adjacency and the following debug output is produced:

%PIM-5-NBRCHG: neighbor 10.1.13.1 UP on interface Serial0/1

Once the two routers establish a neighbor adjacency, they will maintain the adjacency by exchanging hello messages, this can be verified in the output of the "Debug ip pim hello", as follows:

On R1 and R3

```
Rx#Debug ip pim hello
```

On R1

Note R1 has sent hellos out of its F0/0, S0/1 and Lo0 interfaces, but received a hello through its Lo0 and S0/1 interface, whereas, R3 is sending hellos out of its Lo0, S0/1 and S0/0.34, but since R4 is NOT configured with dense-mode yet, it ONLY received hellos back through its Lo0 and S0/1 interface

```
PIM(0): Send periodic v2 Hello on FastEthernet0/0
PIM(0): Send periodic v2 Hello on Serial0/1
PIM(0): Send periodic v2 Hello on Loopback0
PIM(0): Received v2 hello on Loopback0 from 1.1.1.1
PIM(0): Received v2 hello on Serial0/1 from 10.1.13.3
```

On R3

```
PIM(0): Send periodic v2 Hello on Loopback0
PIM(0): Send periodic v2 Hello on Serial0/1
PIM(0): Send periodic v2 Hello on Serial0/0.34
PIM(0): Received v2 hello on Loopback0 from 3.3.3.3
PIM(0): Received v2 hello on Serial0/1 from 10.1.13.1
```

On R3

```
R3#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.13.1         Serial0/1                00:01:27/00:01:16 v2    1 / S
```

Note this neighbor adjacency was established using the hello messages, the hello messages by default are generated every 30 seconds and are sent to 224.0.0.13 destination with a TTL of 1. The above output reveals the following:

Neighbor Address: The IP address of the neighbor discovered

Interface: The local interface where the hello messages were received

Uptime/Expires: This counter shows how long the neighbor has been discovered, the "Expires" timer specifies the hold time value, which is 3.5 times the hello interval (one minute and 45 seconds). After entering the above Show command, if "up arrow" and the "Enter" is pressed in rapid succession, you will see that at one point the timer starts at 1 minute and 45 seconds and it decrements to 1 minute and 15 seconds, in which case a "Hello" is received and the timer is refreshed back to 1 minute and 45 seconds.

Version: This specifies the neighbor's PIM version.

DR Priority/Mode: The default DR priority is 1, on multi-access networks DRs are elected in PIM-SM; this is the router that sends the register and join messages to the RP. In PIM-DM, this role has no meaning, unless IGMP version 1 is used, IGMP Version 1 has no concept in electing a Querier for multi-access networks and therefore, it relies on the Multicast routing protocols. In IGMPv2, it is the Querier that forwards the Multicast stream in a Multi-access networks.
The DR is the router with the highest priority, if the routers have identical priority, the highest IP address is elected as a tiebreaker.

On R4

```
R4(config)#ip multicast-routing

R4(config)#int lo0
R4(config-if)#ip pim dense-mode

R4(config)#int s0/0.43
R4(config-subif)#ip pim dense-mode

R4(config)#int s0/1
R4(config-if)#ip pim dense-mode
```

To verify the configuration:

On R4

```
R4#Sh ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
```

```
10.1.34.3         Serial0/0.43              00:02:27/00:01:44 v2    1 / S
```

On R5

```
R5(config)#ip multicast-routing

R5(config)#int lo0
R5(config-if)#ip pim dense-mode

R5(config-if)#int f0/0
R5(config-if)#ip pim dense-mode

R5(config-if)#int s0/1
R5(config-if)#ip pim dense-mode
```

To verify the configuration:

On R5

```
R5#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.45.4         Serial0/1                00:00:15/00:01:29 v2    1 / S
```

On R6

```
R6(config)#ip multicast-routing

R6(config)#int lo0
R6(config-if)#ip pim dense-mode

R6(config)#int f0/0
R6(config-subif)#ip pim dense-mode
```

To verify the configuration:

On R6

```
R6#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.5         FastEthernet0/0          00:00:18/00:01:26 v2    1 / S
```

On R5

```
R5#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.6         FastEthernet0/0          00:00:59/00:01:44 v2    1 / DR S
10.1.45.4         Serial0/1                00:02:40/00:01:32 v2    1 / S
```

Note since the priority of all routers in Vlan 300 is identical, the router with numerically highest IP address is elected as the DR, in this case R6 with an IP address of 10.1.56.6.

The letter "S" under the Mode column states that the router is state refresh capable.
Since it is NOT desirable to keep on sending the Multicast flow every three minutes, there is a "state-refresh" option, that refreshes the Prune state by introducing a periodic control message that is flooded down the (S,G) tree. When this is received by a router on the RPF interface, the state-refresh message causes the existing Prune state to be refreshed.

These messages are sent every 60 seconds using 224.0.0.13 with a TTL of 1.
Remember that it's the first hop router that sends the state-refresh messages, and if one of the last hop routers has a directly connected host that likes to receive the Multicast flow by sending a join (Host membership report) message, it'll cause the last hop router to send a join message to its upstream router indicating its desire to receive the Multicast flow for a given (S,G), which means that the state will change from Prune to Forward.

```
R5#Sh ip pim interface f0/0 detail | inc pim state

    PIM State-Refresh processing: enabled
    PIM State-Refresh origination: disabled
```

This option can be enabled using the following command; DO NOT enable this option:

```
Rx(config-if)#Ip pim state-refresh origination-interval <4-100 sec>
```

Task 3

Configure Lo0 interface of all routers except R2 to reply to ICMP echo messages for group 224.1.1.1. In this lab, R2 is acting as a source and it should be the router to initiate the Pings.

There are two commands that are somewhat identical yet different, these are the "IP igmp static-group" and "IP igmp join-group" commands.

The "ip igmp static-group" command configures a static group membership entry on an interface, which allows the router to join a given Multicast group. This command would cause the upstream

router/s to maintain the multicast routing table information for that group, which would ensure that all the paths to that multicast group are active. Remember that this command does NOT process the packets; it just joins the group and floods.

The "IP igmp join-group" command allows the router to process and respond to ping commands. Pinging a given group causes all multicast routers to respond. This can be a useful administrative and debugging tool. This command can configure the router to emulate a client connected to a last hop router, whereas, the ping command performed on a router can configure the router to act as a server.

On All Routers except R2

```
Rx(config)#int lo0
Rx(config-if)#Ip igmp join-group 224.1.1.1
```

To verify the configuration:

On R1

```
R1#Show ip igmp groups

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.1.1.1        Loopback0                00:02:07  00:02:33  1.1.1.1
224.0.1.40       FastEthernet0/0          00:25:24  00:02:28  10.1.12.1
```

To verify the configuration

On R2

```
R2#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.12.1, 4 ms
Reply to request 0 from 10.1.13.3, 32 ms
```

R2 ONLY received a reply back from 10.1.12.1 (R1) and 10.1.13.3 (R3) but NOT the other routers. The reason R2 received a reply from R1 and R3 ONLY and NOT the other routers, is because of an

RPF failure. To troubleshoot RPF failures, the best command is ping; Note the reply stopped at R3, therefore, it is safe to assume that the problem is with the next hop router. The following show commands can help to determine where the RPF check failed: Since the entries will timeout of the Mroute table, R2 should be configured to send 100,000 pings:

On R2

```
R2#Ping 224.1.1.1 rep 100000
```

On R1

```
R1#Show ip pim interface count | b address

Address          Interface                FS  Mpackets In/Out
10.1.12.1        FastEthernet0/0          *   8/0
1.1.1.1          Loopback0                *   0/0
10.1.13.1        Serial0/1                *   0/8
```

NOTE: R1 received 8 Mpackets on the F0/0, and it sent 8 Mpackets out of S0/1 toward R3.

```
R1#Sh ip mroute | s 224.1.1.1

 (*, 224.1.1.1), 00:04:15/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1, Forward/Dense, 00:04:15/00:00:00
    Loopback0, Forward/Dense, 00:04:15/00:00:00
(10.1.12.2, 224.1.1.1), 00:02:38/00:02:59, flags: LT
  Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback0, Forward/Dense, 00:02:38/00:00:00
    Serial0/1, Forward/Dense, 00:02:38/00:00:00
```

Note R1 is receiving Mcast traffic through its F0/0 interface and sending it out of Lo0 and S0/1 interfaces.

On R3

```
R3#Show ip pim interface count | b address

Address          Interface                FS  Mpackets In/Out
10.1.13.3        Serial0/1                *   151/0
10.1.34.3        Serial0/0.34             *   0/151
3.3.3.3          Loopback0                *   0/0
```

NOTE: R3 is receiving Mcast traffic from S0/1 interface (coming from R1) and forwards the Mcast traffic out of its S0/0.34 subinterface.

```
R3#Show ip mroute | s 224.1.1.1
```

```
(*, 224.1.1.1), 00:08:47/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback0, Forward/Dense, 00:08:47/00:00:00
    Serial0/0.34, Forward/Dense, 00:08:47/00:00:00
    Serial0/1, Forward/Dense, 00:08:47/00:00:00
(10.1.12.2, 224.1.1.1) , 00:07:09/00:02:58, flags: LT
  Incoming interface: Serial0/1, RPF nbr 10.1.13.1
  Outgoing interface list:
    Serial0/0.34, Forward/Dense, 00:07:09/00:00:00
    Loopback0, Forward/Dense, 00:07:09/00:00:00
```

Note R3 is receiving Mcast traffic from its S0/1 (coming from R1) interface and forwards the traffic out of its Lo0 and S0/0.34 interfaces.

On R4

```
R4#Show ip pim interface count | b address

Address          Interface                FS  Mpackets In/Out
10.1.34.4        Serial0/0.43             *   250/0
10.1.45.4        Serial0/1                *   0/0
4.4.4.4          Loopback0                *   0/0
```

NOTE: R4 is receiving Mcast traffic from its S0/0.43 (Coming from R3) interface, but it is NOT forwarding the traffic to anyone.

```
R4#Show ip mroute | s 224.1.1.1

(*, 224.1.1.1), 00:18:25/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1, Forward/Dense, 00:18:25/00:00:00
    Serial0/0.43, Forward/Dense, 00:18:25/00:00:00
    Loopback0, Forward/Dense, 00:18:25/00:00:00
(10.1.12.2, 224.1.1.1), 00:01:47/00:01:12, flags: L
  Incoming interface: Null, RPF nbr 10.1.100.3
  Outgoing interface list:
    Loopback0, Forward/Dense, 00:01:47/00:00:00
    Serial0/0.43, Forward/Dense, 00:01:47/00:00:00
    Serial0/1, Forward/Dense, 00:01:47/00:00:00
```

Note the output of the mroute table reveals that the RPF interface is 10.1.100.3.

```
R4#Traceroute 10.1.12.2
```

```
Type escape sequence to abort.
Tracing the route to 10.1.12.2
  1 10.1.100.3 4 msec 0 msec 0 msec
  2 10.1.13.1 12 msec 16 msec 12 msec
  3 10.1.12.2 12 msec *  12 msec
```

Note the route back is through 10.1.100.3, the F0/0 interface; BUT this is NOT the interface through which Mcast traffic was received.

```
R4#Show ip rpf 10.1.12.2
```

RPF information for? (10.1.12.2) failed, no route exists

```
R4#MTrace 10.1.12.2

Type escape sequence to abort.
Mtrace from 10.1.12.2 to 10.1.100.4 via RPF
From source (?) to destination (?)
Querying full reverse path...
 0  10.1.100.4
-1  10.1.100.4 None No route
```

There are two types of RPFs, Unicast and Multicast. Unicast RPFs are disabled by default and it can be enabled, once it's enabled, if the router receives traffic on an RPF enabled interface, it checks the source IP address of the traffic and consults its routing table. If the interface through which the traffic was received is the closest interface back to the source, the RPF check pass and the traffic is forwarded.

In Multicasting the RPF checks are enabled by default and they can NOT be disabled, these checks are performed hop by hop to ensure a loop free connection. When a router receives a given Multicast flow through an interface, it checks the source IP address against the "Static Mroute table" first, if there is an entry and it points to the same interface, RPF checks are successful and the Mcast traffic is forwarded. If an entry for the source is NOT found, then, and ONLY then, the IP routing table is consulted, and once again, the source IP address of the Mcast flow should be the closest interface back to the source, if it is not, the local router will discard the traffic.

In this case, 10.1.100.3 is NOT an RPF interface, but it is the closest interface to the source (10.1.12.2) but the traffic is received from S0/0.43, so we should insert an entry in the static Mroute table so the IP routing table is NOT checked.

On R4

```
R4(config)#ip mroute 10.1.12.2 255.255.255.255 10.1.34.3
```

To verify the configuration:

On R4

```
R4#Show ip mroute | b outgoing

Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:24:34/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback0, Forward/Dense, 00:24:34/00:00:00
    Serial0/1, Forward/Dense, 00:24:34/00:00:00
    Serial0/0.43, Forward/Dense, 00:24:34/00:00:00

(10.1.12.2, 224.1.1.1), 00:02:00/00:02:59, flags: LT
  Incoming interface: Serial0/0.43, RPF nbr 10.1.34.3, Mroute
  Outgoing interface list:
    Serial0/1, Forward/Dense, 00:02:00/00:00:00
    Loopback0, Forward/Dense, 00:02:00/00:00:00

(*, 224.0.1.40), 00:37:23/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1, Forward/Dense, 00:36:55/00:00:00
    Serial0/0.43, Forward/Dense, 00:37:23/00:00:00

R4#Show ip rpf 10.1.12.2

RPF information for ? (10.1.12.2)
  RPF interface: Serial0/0.43
  RPF neighbor: ? (10.1.34.3)
  RPF route/mask: 10.1.12.2/32
  RPF type: static
  RPF recursion count: 0
  Doing distance-preferred lookups across tables

R4#MTrace 10.1.12.2

Type escape sequence to abort.
Mtrace from 10.1.12.2 to 10.1.100.4 via RPF
From source (?) to destination (?)
Querying full reverse path...
 0  10.1.100.4
-1  10.1.100.4 PIM/Static  [10.1.12.2/32]
-2  10.1.34.3 PIM  [10.1.12.0/24]
```

The Mroute means that a static mroute was configured.

```
-3  10.1.13.1 PIM  [10.1.12.0/24]
-4  10.1.12.2

R4#Show ip pim interface count | b Address

Address          Interface                FS  Mpackets In/Out
10.1.34.4        Serial0/0.43             *   576/0
10.1.45.4        Serial0/1                *   0/117
4.4.4.4          Loopback0                *   0/0
```

To test the configuration:

On R2

You should see that every router is now responding:

```
Reply to request 0 from 10.1.12.1, 1 ms
Reply to request 0 from 10.1.56.6, 100 ms
Reply to request 0 from 10.1.45.5, 88 ms
Reply to request 0 from 10.1.34.4, 56 ms
Reply to request 0 from 10.1.13.3, 28 ms
```

Task 4

Remove the "**IP Pim join-group 224.1.1.1**" from Lo0 interface of R6.

On R6

```
R6(config)#int lo0
R6(config-if)#NO ip igmp join-group 224.1.1.1
```

To verify the configuration:

On R6

```
R6#Show ip igmp groups

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.0.1.40       FastEthernet0/0          00:43:02  00:02:38  10.1.56.6
```

Note group 224.1.1.1 is NO LONGER there.

On R6

```
R6#Show ip mroute | s 224.1.1.1

 (*, 224.1.1.1), 00:31:52/stopped, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Dense, 00:31:52/00:00:00
(10.1.12.2, 224.1.1.1), 00:07:29/00:00:40, flags: PT
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.56.5
  Outgoing interface list: Null
```

Note the "P" flag is set which means that the local router sent a Prune back toward the source. Once the upstream router receives the Prune message, both routers will maintain a prune state for 3 minutes, after which the traffic is sent to R6 again.

The following show commands reveals this fact:

```
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)

(10.1.12.2, 224.1.1.1), 00:01:15/00:01:44, flags: PT

R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)

(10.1.12.2, 224.1.1.1), 00:01:29/00:01:30, flags: PT

R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)

(10.1.12.2, 224.1.1.1), 00:01:32/00:01:27, flags: PT

R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)

(10.1.12.2, 224.1.1.1), 00:01:34/00:01:25, flags: PT

R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)

(10.1.12.2, 224.1.1.1), 00:02:06/00:00:53, flags: PT

R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)

(10.1.12.2, 224.1.1.1), 00:02:33/00:00:26, flags: PT

R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
```

```
(10.1.12.2, 224.1.1.1), 00:02:45/00:00:14, flags: PT

R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)

(10.1.12.2, 224.1.1.1), 00:02:55/00:00:04, flags: PT

R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)

(10.1.12.2, 224.1.1.1), 00:02:59/00:00:00, flags: PT

R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)

(10.1.12.2, 224.1.1.1), 00:00:00/00:02:59, flags: PT
```

Task 5

Configure R6's F0/0 interface to keep the pruned state mroute entries alive as long as needed without going through flooding periods. R6 should send a refresh message back toward R5 every 60 seconds to maintain the prune state.

On R1

```
R1(config)#int f0/0
R1(config-if)#ip pim state-refresh origination-interval 60
```

To verify the configuration:

On R1

```
R1#Show ip pim interface f0/0 detail | inc state-refresh

    PIM State-Refresh processing: enabled
    PIM State-Refresh origination: enabled, interval: 60 seconds
```

To verify the configuration:

On R6

```
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
(10.1.12.2, 224.1.1.1), 00:01:44/00:02:52, flags: PT
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
(10.1.12.2, 224.1.1.1), 00:01:48/00:02:49, flags: PT
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
(10.1.12.2, 224.1.1.1), 00:01:54/00:02:42, flags: PT
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
(10.1.12.2, 224.1.1.1), 00:02:03/00:02:33, flags: PT
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
(10.1.12.2, 224.1.1.1), 00:02:10/00:02:26, flags: PT
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
(10.1.12.2, 224.1.1.1), 00:02:17/00:02:19, flags: PT
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
(10.1.12.2, 224.1.1.1), 00:02:34/00:02:02, flags: PT
R6#Show ip mroute | inc (10.1.12.2, 224.1.1.1)
(10.1.12.2, 224.1.1.1), 00:02:39/00:02:57, flags: PT
```

PIM-DM keeps a timer on its pruned interface, and when the timer expires, the prune status is removed and Mcast traffic starts to flow again, until a new prune message is received from downstream routers.

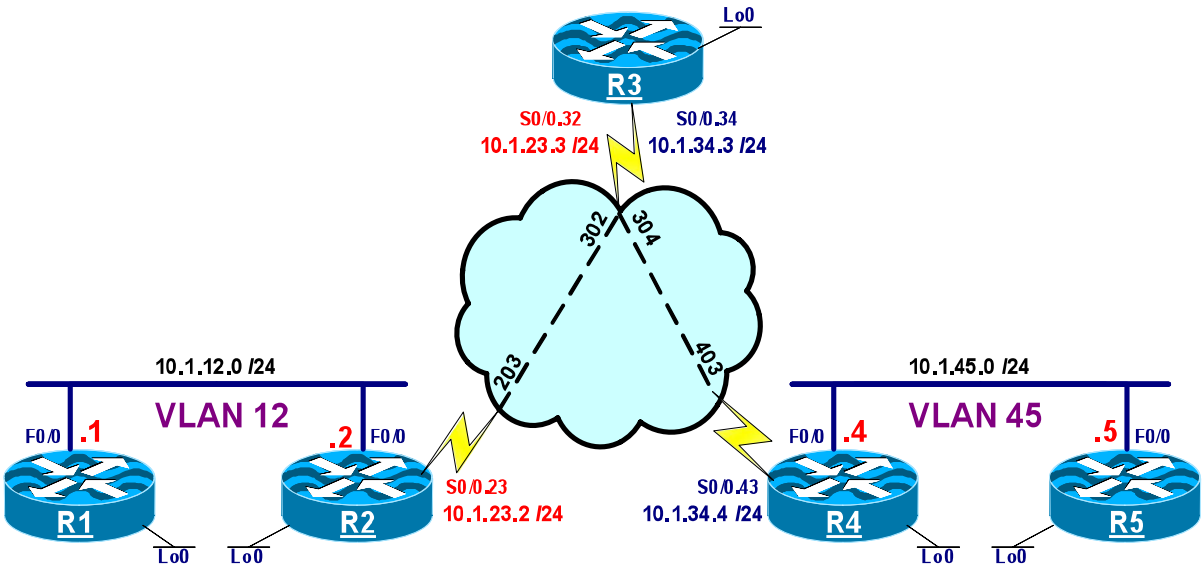
The "ip pim state-refresh" interface command sends a Mcast control packet down its pruned interface. If a downstream router sees the control Mcast packet, and it is not interested in that (*,G) traffic, the downstream router sends a new prune message. This will reset the timer on the upstream router's pruned entry.

The interval is how often R1 sends the Mcast control packet downstream on R5. Note the "State-Refresh" processing is enabled by default on all routers.

Task 6

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 3 – Static RP Configuration



Lab Setup

- Configure the rack according to the diagram
- Configure OSPF area 0 on all interfaces.
- Use the IP addressing chart below for IP addressing scheme

IP addressing Chart:

Router	Interface / IP addressing
R1	F0/0 = 10.1.12.1/24 Lo0 = 1.1.1.1 /32
R2	**F0/0 = 10.1.12.2/24** **S0/0.23 = 10.1.23.2/24** **Lo0 = 1.1.1.2/32**
R3	S0/0.32 = 10.1.23.3/24 S0/0.34 = 10.1.34.3/24 Lo0 = 1.1.1.3/32
R4	**F0/0 = 10.1.45.4/24** **S0/0.43 = 10.1.34.4 /24** **Lo0 = 1.1.1.4/32**
R5	F0/0 = 10.1.45.5/24 Lo0 = 1.1.1.5/32

Task 1

Configure PIM on the interfaces according to the following table:

Router	Interface / PIM
R1	F0/0 = **PIM Sparse mode** Lo0 = **PIM Sparse mode**
R2	F0/0 = **PIM Sparse mode** S0/0.23 = **PIM Sparse mode** Lo0 = **PIM Sparse mode**
R3	S0/0.34 = **PIM Sparse mode** S0/0.32 = **PIM Sparse mode** Lo0 = **PIM Sparse mode**
R4	F0/0 = **PIM Sparse mode** S0/0.43 = **PIM Sparse mode** Lo0 = **PIM Sparse mode**
R5	F0/0 = **PIM Sparse mode** Lo0 = **PIM Sparse mode**

The first step is to enable Multicast routing and configure PIM Sparse mode on the specified interfaces:

On R1

```
R1(config)#ip multicast-routing
```

```
R1(config)#int lo0
R1(config-if)#ip pim sparse-mode

R1(config-if)#int f0/0
R1(config-if)#ip pim sparse-mode
```

On R2

```
R2(config)#ip multicast-routing

R2(config)#int lo0
R2(config-if)#ip pim sparse-mode

R2(config-if)#int f0/0
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.23
R2(config-subif)#ip pim sparse-mode
```

On R3

```
R3(config)#ip multicast-routing

R3(config)#int lo0
R3(config-if)#ip pim sparse-mode

R3(config-if)#int s0/0.32
R3(config-subif)#ip pim sparse-mode

R3(config-subif)#int s0/0.34
R3(config-subif)#ip pim sparse-mode
```

On R4

```
R4(config)#ip multicast-routing

R4(config)#int lo0
R4(config-if)#ip pim sparse-mode

R4(config-if)#int f0/0
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.43
R4(config-if)#ip pim sparse-mode
```

On R5

```
R5(config)#ip multicast-routing

R5(config)#int lo0
R5(config-if)#ip pim sparse-mode

R5(config-if)#int f0/0
R5(config-subif)#ip pim sparse-mode
```

To verify the configuration:

On R5

Verification is performed using the "show ip pim neighbor" commands, which enables us to verify that the routers actually see other routers as PIM neighbors.

```
R5#Show ip pim neighbor

PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.45.4         FastEthernet0/0          00:02:05/00:01:36 v2    1 / S
```

On R4

```
R4#Show ip pim neighbor

PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.34.3         Serial0/0.43             00:10:12/00:01:21 v2    1 / S
10.1.45.5         FastEthernet0/0          00:01:47/00:01:24 v2    1 / DR S
```

Note R5 is the DR because it has the highest IP address on 10.1.45.0 /24 segment.

Task 2

R2 should be configured as the RP for the private group addresses, whereas, R3 should be configured as the RP for all groups. You should use static configuration to accomplish this task.

Static RP configuration, even though the most popular and simple solution, it is extremely easy to make configuration typos, because the exact same configuration needs to be replicated accurately on all routers including the RP.

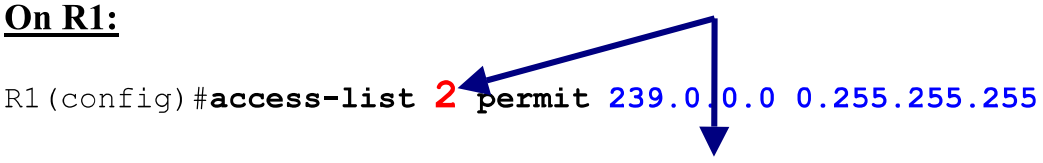
Configuration tasks:

- **R2's Lo0 is configured as the RP for the private group addresses.**
- **R3's Lo0 interface is configured as the RP for all other groups.**

Step 1:

For R2's requirement, an access-list is configured to identify the private multicast group address range and the static RP command is configured to reference the access-list. This configuration MUST be done on all routers including the RP. Note, when an access-list is NOT referenced, the IP address in the configuration is configured to be the RP for all groups (224.0.0.0 15.255.255.255) as follows:

On R1:



```
R1(config)#access-list 2 permit 239.0.0.0 0.255.255.255

R1(config)#ip pim rp-address 1.1.1.2 2
R1(config)#ip pim rp-address 1.1.1.3
```

To verify the configuration:

On R1

```
R1#Sh ip pim rp map

PIM Group-to-RP Mappings

Acl: 2, Static
  RP: 1.1.1.2 (?)
Group(s): 224.0.0.0/4, Static
  RP: 1.1.1.3 (?)
```

Note since an access-list was NOT referenced, the router becomes the RP for ALL Mcast groups

Note the output of the above show command reveals that the lo0 of R2 is the RP for the groups that are referenced in ACL 2, whereas, the lo0 of R3 is the RP for the group range of 224.0.0.0 /4, meaning all groups.

IMPORTANT **********
In the above configuration the private group addresses are overlapping, the following simple rule determines which router is the RP for a given group:

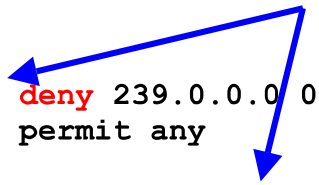
- **If two RPs have overlapping scope of groups, the RP with the higher source IP address wins.**

In our case, R3 is the RP for 224.0.0.0/4, and R2 is the RP for 239.0.0.0/8, so let's examine the following two groups, note the first group is from Global scope, whereas, the second group is from Admin scope (Private groups) and determine to which RP they are mapped to:

- **232.1.2.3 – Only R3's scope covers this group, therefore, R3 is the RP.**
- **239.1.1.1 – Both R3's and R2's scope cover this group, but since R3's Lo0 is numerically higher than R2, therefore, R3 wins!**

NOTE: based on the existing configuration as long as R3 is up, it will be chosen as the RP for all groups and R2 will NOT be used at all. The correct solution is to configure another access-list for R3, deny group "239.0.0.0/8", and allow all other groups.

On R1



```
R1(config)#access-list 3 deny 239.0.0.0 0.255.255.255
R1(config)#access-list 3 permit any

R1(config)#ip pim rp-address 1.1.1.3 3
```

To verify the configuration:

On R1

```
R1#Sh ip pim rp map
```

PIM Group-to-RP Mappings

```
Acl: 2, Static
     RP: 1.1.1.2 (?)
Acl: 3, Static
     RP: 1.1.1.3 (?)
```

Note R2's Lo0 is the RP for the groups defined in ACL #2, and R3's Lo0 is the RP for groups defined in ACL #3.
Remember that the configuration should be performed on all routers including the RP.

On R2:

```
R2(config)#access-list 2 permit 239.0.0.0 0.255.255.255
R2(config)#ip pim rp-address 1.1.1.2 2

R2(config)#access-list 3 deny 239.0.0.0 0.255.255.255
R2(config)#access-list 3 permit any
```

```
R2(config)#ip pim rp-address 1.1.1.3 3
```

On R3:

```
R3(config)#access-list 2 permit 239.0.0.0 0.255.255.255
R3(config)#ip pim rp-address 1.1.1.2 2

R3(config)#access-list 3 deny 239.0.0.0 0.255.255.255
R3(config)#access-list 3 permit any
R3(config)#ip pim rp-address 1.1.1.3 3
```

On R4:

```
R4(config)#access-list 2 permit 239.0.0.0 0.255.255.255
R4(config)#ip pim rp-address 1.1.1.2 2

R4(config)#access-list 3 deny 239.0.0.0 0.255.255.255
R4(config)#access-list 3 permit any
R4(config)#ip pim rp-address 1.1.1.3 3
```

On R5:

```
R5(config)#access-list 2 permit 239.0.0.0 0.255.255.255
R5(config)#ip pim rp-address 1.1.1.2 2

R5(config)#access-list 3 deny 239.0.0.0 0.255.255.255
R5(config)#access-list 3 permit any
R5(config)#ip pim rp-address 1.1.1.3 3
```

To verify the configuration:

On R5

```
R5#Sh ip pim rp mapping

PIM Group-to-RP Mappings

Acl: 2, Static
    RP: 1.1.1.2 (?)
Acl: 3, Static
    RP: 1.1.1.3 (?)
```

Task 3

Configure R1 to be the DR for VLAN 12 and R4 to be the DR for VLAN 45.

Just like OSPF routing protocol, on multi-access networks there is a DR election. The rule for electing a DR is as follows:

- **The router with the highest DR priority is chosen as the DR. The default priority is 1.**
- **If the priority of the routers is the same, numerically highest IP address is chosen as the DR.**

To verify the existing DR:

On R1

```
R1#Sh ip pim neigh | inc dr

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
Neighbor          Interface                Uptime/Expires    Ver   DR
10.1.12.2         FastEthernet0/0          00:29:22/00:01:23 v2    1 / DR S
```

Note the output of the above show command reveals the existing DR, in this case, it is R2 because it has the highest IP address.

On R1

```
R1(config)#int f0/0
R1(config-if)#ip pim dr-priority 2
```

You should see the following console message:

%PIM-5-DRCHG: DR change from neighbor 10.1.12.2 to 10.1.12.1 on interface FastEthernet0/0

The above console message states that the DR has been changed from 10.1.12.2 to 10.1.12.1

To Verify the configuration:

On R2

```
R2#Show ip pim neighbor | inc dr
```

Note R1 is the DR with a priority of 2.

```
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                          Prio/Mode
10.1.12.1         FastEthernet0/0          00:34:26/00:01:42 v2    2 / DR S
10.1.23.3         Serial0/0.23             00:20:35/00:01:19 v2    1 / S
```

To change the DR in VLAN 45:

On R4

```
R4(config)#int f0/0
R4(config-if)#ip pim dr-priority 2
```

To verify the configuration:

On R5

```
R5#Sh ip pim neighbor | inc dr

Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
Neighbor          Interface                Uptime/Expires    Ver    DR
10.1.45.4         FastEthernet0/0          00:27:53/00:01:26 v2     2 / DR S
```

Task 4

- Configure R5's F0/0 interface to join group 225.5.5.5
- All routers MUST be able to ping this group.

On R5

```
R5(config)#int f0/0
R5(config-if)#ip igmp join-group 225.5.5.5
```

To verify the configuration:

On R5

```
R5#Show ip mroute | s 225.5.5.5

(*, 225.5.5.5), 00:00:08/00:02:51, RP 1.1.1.3, flags: SJPL
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.45.4
  Outgoing interface list: Null
```

On R4

```
R4#Show ip mroute | s 225.5.5.5
```

```
(*, 225.5.5.5), 00:02:00/00:02:26, RP 1.1.1.3, flags: SJC
  Incoming interface: Serial0/0.43, RPF nbr 10.1.34.3
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:02:00/00:02:26
```

On R3

```
R3#Show ip mroute | s 225.5.5.5

(*, 225.5.5.5), 00:02:47/00:02:41, RP 1.1.1.3, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.34, Forward/Sparse, 00:02:47/00:02:41
```

On R2 or R1

```
Rx#Show ip mroute | s 225.5.5.5
Rx#
```

Note every PIM enabled router will create the parent entry in their mroute table; this is created as the join messages are sent from R5 to the RP, R2 and R1 will NOT have any entry for this group. The RP for this group is 1.1.1.3 (R3), remember that R3 is the RP for all scope (Local, Global scopes) except the Admin scope.

To verify the configuration:

On R1

```
R1#Ping 225.5.5.5

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 225.5.5.5, timeout is 2 seconds:
Reply to request 0 from 10.1.45.5, 124 ms
Reply to request 0 from 10.1.45.5, 140 ms
```

To verify the configuration:

On R1

```
R1#Sh ip mroute | s 225.5.5.5

(*, 225.5.5.5), 00:00:17/stopped, RP 1.1.1.3, flags: SPF
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.12.2
  Outgoing interface list: Null
(1.1.1.1, 225.5.5.5), 00:00:17/00:03:20, flags: FT
```

```
  Incoming interface: Loopback0, RPF nbr 0.0.0.0, Registering
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:00:17/00:03:12
(10.1.12.1, 225.5.5.5), 00:00:17/00:02:50, flags: PFT
  Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0, Registering
  Outgoing interface list: Null
```

On R2

```
R2#Show ip mroute | s 225.5.5.5

(*, 225.5.5.5), 00:01:58/stopped, RP 1.1.1.3, flags: SP
  Incoming interface: Serial0/0.23, RPF nbr 10.1.23.3
  Outgoing interface list: Null
(1.1.1.1, 225.5.5.5), 00:01:58/00:01:31, flags:
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.12.1
  Outgoing interface list:
    Serial0/0.23, Forward/Sparse, 00:01:58/00:03:29
(10.1.12.1, 225.5.5.5), 00:01:58/00:01:38, flags: T
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.12.1
  Outgoing interface list:
    Serial0/0.23, Forward/Sparse, 00:01:58/00:03:29
```

On R3

```
R3#Show ip mroute | s 225.5.5.5

(*, 225.5.5.5), 00:08:38/stopped, RP 1.1.1.3, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.34, Forward/Sparse, 00:08:38/00:02:44
(1.1.1.1, 225.5.5.5), 00:00:24/00:03:20, flags: T
  Incoming interface: Serial0/0.32, RPF nbr 10.1.23.2
  Outgoing interface list:
    Serial0/0.34, Forward/Sparse, 00:00:24/00:03:05
(10.1.12.1, 225.5.5.5), 00:00:24/00:03:20, flags: T
  Incoming interface: Serial0/0.32, RPF nbr 10.1.23.2
  Outgoing interface list:
    Serial0/0.34, Forward/Sparse, 00:00:24/00:03:05
```

On R4

```
R4#Show ip mroute | s 225.5.5.5

(*, 225.5.5.5), 00:09:00/stopped, RP 1.1.1.3, flags: SJC
```

```
  Incoming interface: Serial0/0.43, RPF nbr 10.1.34.3
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:09:00/00:02:24
(1.1.1.1, 225.5.5.5), 00:00:46/00:02:52, flags: JT
  Incoming interface: Serial0/0.43, RPF nbr 10.1.34.3
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:00:46/00:02:24
(10.1.12.1, 225.5.5.5), 00:00:46/00:02:52, flags: JT
  Incoming interface: Serial0/0.43, RPF nbr 10.1.34.3
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:00:46/00:02:24
```

On R5

```
R5#Sh ip mroute | s 225.5.5.5

(*, 225.5.5.5), 00:09:18/stopped, RP 1.1.1.3, flags: SJPL
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.45.4
  Outgoing interface list: Null
(1.1.1.1, 225.5.5.5), 00:01:05/00:02:58, flags: PLT
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.45.4
  Outgoing interface list: Null
(10.1.12.1, 225.5.5.5), 00:01:05/00:02:58, flags: PLT
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.45.4
  Outgoing interface list: Null
```

On R5

```
R5#Show ip igmp groups

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
225.5.5.5        FastEthernet0/0          00:13:47  00:02:37  10.1.45.5
224.0.1.40       FastEthernet0/0          00:28:21  00:02:39  10.1.45.4
224.0.1.40       Loopback0                00:28:26  00:02:48  1.1.1.5
```

The output of the above show command reveals the active groups on the local router's interface.

Task 5

From time to time, a bogus router on VLAN 45 appears with the IP address of 10.1.45.100; configure R4 and R5 to deny any PIM relationship with that router.

The "ip pim neighbor-filter" command is usually used for IP IGMP helper function for stub Multicast routing, but in this scenario it is used to filter such bogus router.

On R4

```
R4(config)#access-list 1 deny 10.1.45.100
R4(config)#access-list 1 permit any

R4(config)#int f0/0
R4(config-subif)#ip pim neighbor-filter 1
```

On R5

```
R5(config)#access-list 1 deny 10.1.45.100
R5(config)#access-list 1 permit any

R5(config)#int f0/0
R5(config-subif)#ip pim neighbor-filter 1
```

To verify the configuration:

On R5

```
R5#Sh access-list 1

Standard IP access list 1
    10 deny   10.1.45.100
    20 permit any log (4 matches)
```

To test the configuration:

R6 is added to VLAN 45 and its F0/0 interface is configured with "IP PIM Sparse-mode" as follows:

On R6

```
R6(config)#ip multicast-routing

R6(config)#int f0/0
R6(config-if)#ip addr 10.1.45.100 255.255.255.0
R6(config-if)#ip pim sparse-mode
R6(config-if)#No shut
```

On SW1

```
SW1(config)#int f0/6
SW1(config-if)#swi mode acc
SW1(config-if)#swi acc v 45
SW1(config-if)#spanning portf
SW1(config-if)#No shut
```

On R4

```
R4#Show ip pim neighbor

PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.45.5         FastEthernet0/0          02:37:43/00:01:33 v2    1 / S
10.1.34.3         Serial0/0.43             02:38:35/00:01:42 v2    1 / S
```

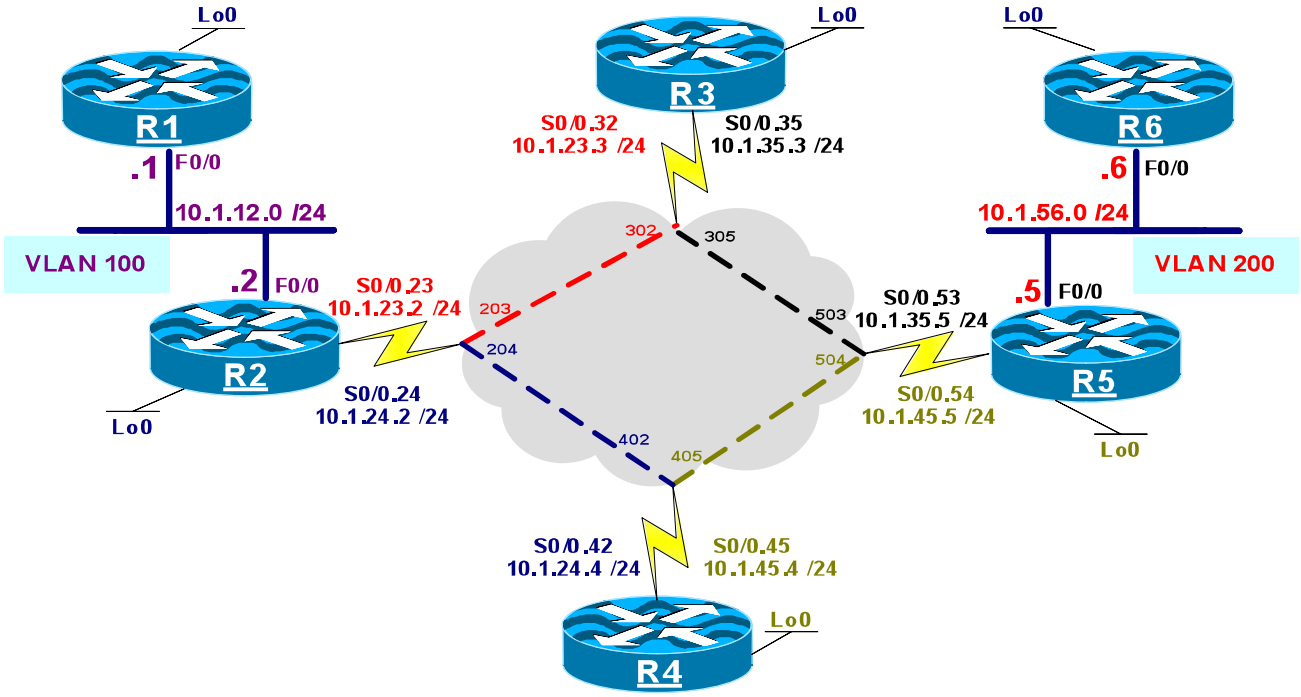
On R5

```
R5#Show ip pim neighbor

PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.45.4         FastEthernet0/0          02:38:24/00:01:41 v2    2 / DR S

R5#Show ip access-list 1

Standard IP access list 1
    10 deny   10.1.45.100 (5 matches)
    20 permit any log (17 matches)
```

Task 6

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 4 – Auto-RP



Lab Setup

- Configure the F0/0 interface of R1 and R2 in VLAN 100, and F0/0 interface of R4 and R5 in VLAN 200.
- All frame-Relay connections must be configured in a point-to-point manner.
- Use the IP addressing chart below for IP addressing scheme

IP addressing Chart:

Router	Interface / IP addressing
R1	F0/0 = 10.1.12.1 /24 Lo0= 1.1.1.1 /32
R2	**F0/0 = 10.1.12.2 /24** **S0/0.23 = 10.1.23.2/24** **S0/0.24 = 10.1.24.2/24** **Lo0= 1.1.1.2 /32**
R3	S0/0.32 = 10.1.23.3/24 S0/0.35 = 10.1.35.3/24 Lo0= 1.1.1.3 /32
R4	**S0/0.42 = 10.1.24.4/24** **S0/0.45 = 10.1.45.4 /24** **Lo0= 1.1.1.4 /32**
R5	S0/0.53 = 10.1.35.5/24 S0/0.54 = 10.1.45.5/24 F0/0 = 10.1.56.5/24 Lo0= 1.1.1.5 /32
R6	**F0/0 = 10.1.56.6/24** **Lo0= 1.1.1.6/32**

Task 1

Configure OSPF area 0 on all interfaces.

On All Routers

```
Rx(config)#router ospf 1
Rx(config-router)#netw 0.0.0.0 0.0.0.0 are 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf | inc O

O       1.1.1.3 [110/66] via 10.1.12.2, 00:00:10, FastEthernet0/0
O       1.1.1.2 [110/2] via 10.1.12.2, 00:00:10, FastEthernet0/0
O       1.1.1.5 [110/130] via 10.1.12.2, 00:00:10, FastEthernet0/0
O       1.1.1.4 [110/66] via 10.1.12.2, 00:00:10, FastEthernet0/0
O       1.1.1.6 [110/131] via 10.1.12.2, 00:00:00, FastEthernet0/0
O       10.1.24.0 [110/65] via 10.1.12.2, 00:00:10, FastEthernet0/0
```

```
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:10, FastEthernet0/0
O       10.1.45.0 [110/129] via 10.1.12.2, 00:00:10, FastEthernet0/0
O       10.1.35.0 [110/129] via 10.1.12.2, 00:00:10, FastEthernet0/0
O       10.1.56.0 [110/130] via 10.1.12.2, 00:00:10, FastEthernet0/0
```

Task 2

Configure PIM on the interfaces according to the following table:

Router	Interface / PIM
R1	F0/0 = PIM Sparse-Dense Lo0 = PIM Sparse-Dense mode
R2	**F0/0 = PIM Sparse-Dense mode** **S0/0.23 = PIM Sparse-Dense mode** **S0/0.24 = PIM Sparse-Dense mode** **Lo0 = PIM Sparse-Dense mode**
R3	S0/0.32 = PIM Sparse-Dense mode S0/0.35 = PIM Sparse-Dense mode Lo0 = PIM Sparse-Dense mode
R4	**S0/0.42 = PIM Sparse-Dense mode** **S0/0.45 = PIM Sparse-Dense mode** **Lo0= PIM Sparse-Dense mode**
R5	F0/0 = PIM Sparse-Dense mode S0/0.53 = PIM Sparse-Dense mode S0/0.54 = PIM Sparse-Dense mode Lo0 = PIM Sparse-Dense mode
R6	**F0/0 = PIM Sparse-Dense mode** **Lo0 = PIM Sparse-Dense mode**

On R1

```
R1(config)#ip multicast-routing

R1(config)#int lo 0
R1(config-if)#ip pim sparse-dense-mode

R1(config-if)#int f0/0
R1(config-if)#ip pim sparse-dense-mode
```

On R2

```
R2(config)#ip multicast-routing

R2(config)#int lo0
R2(config-if)#ip pim sparse-dense-mode

R2(config-if)#int f0/0
R2(config-subif)#ip pim sparse-dense-mode

R2(config-subif)#int s0/0.23
R2(config-subif)#ip pim sparse-dense-mode

R2(config-subif)#int s0/0.24
R2(config-subif)#ip pim sparse-dense-mode
```

On R3

```
R3(config)#ip multicast-routing

R3(config)#int lo0
R3(config-if)#ip pim sparse-dense-mode

R3(config-if)#int s0/0.32
R3(config-subif)#ip pim sparse-dense-mode

R3(config-subif)#int s0/0.35
R3(config-subif)#ip pim sparse-dense-mode
```

On R4

```
R4(config)#ip multicast-routing

R4(config)#int lo0
R4(config-if)#ip pim sparse-dense-mode

R4(config-if)#int s0/0.42
R4(config-if)#ip pim sparse-dense-mode

R4(config-if)#int s0/0.45
R4(config-if)#ip pim sparse-dense-mode
```

On R5

```
R5(config)#ip multicast-routing

R5(config)#int lo0
```

```
R5(config-if)#ip pim sparse-dense-mode

R5(config-if)#int f0/0
R5(config-subif)#ip pim sparse-dense-mode

R5(config-subif)#int s0/0.53
R5(config-subif)#ip pim sparse-dense-mode

R5(config-subif)#int s0/0.54
R5(config-subif)#ip pim sparse-dense-mode
```

On R6

```
R6(config)#ip multicast-routing

R6(config)#int lo0
R6(config-if)#ip pim sparse-dense-mode

R6(config-if)#int f0/0
R6(config-if)#ip pim sparse-dense-mode
```

To verify the configuration:

On R1

```
R1#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.2         FastEthernet0/0          00:17:54/00:01:34 v2    1 / DR S
```

On R2

```
R2#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.1         FastEthernet0/0          00:18:59/00:01:27 v2    1 / S
10.1.23.3         Serial0/0.23             00:17:45/00:01:41 v2    1 / S
10.1.24.4         Serial0/0.24             00:04:33/00:01:38 v2    1 / S
```

On R3

```
R3#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
```

```
Address                                                    Prio/Mode
10.1.23.2         Serial0/0.32              00:17:45/00:01:38 v2    1 / S
10.1.35.5         Serial0/0.35              00:03:41/00:01:30 v2    1 / S
```

On R4

```
R4#Show ip pim neighbor | b interface

Neighbor          Interface                 Uptime/Expires    Ver   DR
Address                                                             Prio/Mode
10.1.24.2         Serial0/0.42              00:04:33/00:01:36 v2    1 / S
10.1.45.5         Serial0/0.45              00:03:35/00:01:36 v2    1 / S
```

On R5

```
R5#Show ip pim neighbor | b interface

Neighbor          Interface                 Uptime/Expires    Ver   DR
Address                                                             Prio/Mode
10.1.56.6         FastEthernet0/0           00:03:23/00:01:18 v2    1 / DR S
10.1.35.3         Serial0/0.53              00:03:41/00:01:30 v2    1 / S
10.1.45.4         Serial0/0.54              00:03:35/00:01:36 v2    1 / S
```

On R6

```
R6#Show ip pim neighbor

Neighbor          Interface                 Uptime/Expires    Ver   DR
Address                                                             Prio/Mode
10.1.56.5         FastEthernet0/0           00:03:23/00:01:18 v2    1 / S
```

Task 3

Configure R4 as the Primary RP for 238.0.0.0/8 and R3 as the backup RP. You should use Auto-RP to distribute the RP mappings information through R5.

In Auto-RP, two roles must be defined, candidate-RP/s and the Mapping-agent:

1. **Candidate-RP/s – The routers that are configured to be the RP will announce themselves as the RP for all or specific group/s, they accomplish this by sending RP-announce messages. The destination address of these announcements is to 224.0.1.39.**

2. **All routers will hear these announcements but ONLY the router configured as the mapping agent will process them.**

3. **Mapping-Agent – The router that is configured to be the Mapping-Agent will process the RP announce messages and decide the RP to group mapping, if there are more than one RP announcements for a given group, the Mapping-Agent will elect the RP with the highest source IP address as the RP for that group. The MA will send RP-discover messages to announce the RP to group mapping.**

Why Sparse-dense-mode?

Since 224.0.1.39 and 224.0.1.40 can ONLY work in Dense-mode, configuring Sparse-mode will NOT work. Therefore, Sparse-dense-mode is required for Auto-RP configuration.

Why do these groups have to operate in dense mode?

In Auto-RP, the multicast packets are forwarded on the shared tree which means that the routers need to be aware of the RP. The router that's listening for 224.0.1.40 should notify its RP that it needs to join group 224.0.1.40 so it can receive and process the RP-Discover messages, but the big question is how would that router know where the RP is if it has not received the RP-discover messages for that specific group? Other words, to join the group, they need to know the RP, but to know the RP they need to join the group.

Therefore, the "ip pim sparse-dense-mode" was created to fix this problem for groups 224.0.1.39 and 224.0.1.40. Remember that in Sparse-dense-mode configuration if the RP is known for the group, the interface uses Sparse mode, however, if the RP is NOT known, then, it will use dense mode.

In this task both R3 and R4 should advertise themselves, as the RP for the specific group, and R5 will be configured to collect the RP announcements and decide which router will be the primary and which one will be the Backup RP.

On R3

```
R3(config)#access-list 1 permit 238.0.0.0 0.255.255.255

R3(config)#ip pim send-rp-announce loopback 0 scope 2 group-list 1
```

NOTE: the "group-list" keyword limits the RP announcements only for the 238.0.0.0/8 group. The scope defines the TTL of these packets, since it was NOT mentioned in the requirements; a scope of 2 was configured so ONLY the Mapping Agent within this topology can receive the announcements.

In order for a router to use its loopback interface as the source IP address of RP-Announce messages, the "ip pim sparse-dense-mode" must be configured on that loopback interface. If it's not

configured, you will receive the following message:

Must first configure PIM mode on the interface: Loopback0

Note if "Debug ip pim auto-rp" was entered before configuring the Candidate RP, the following output will result every 60 seconds, this interval can be changed using the "interval" keyword:

```
Auto-RP(0): Build RP-Announce for 1.1.1.4, PIMv2/v1, ttl 2, ht 181
Auto-RP(0):  Build announce entry for (238.0.0.0/8)
Auto-RP(0): Send RP-Announce packet on Serial0/0.42
Auto-RP(0): Send RP-Announce packet on Serial0/0.45
Auto-RP: Send RP-Announce packet on Loopback0
```

On R4

```
R4(config)#access-list 1 permit 238.0.0.0 0.255.255.255

R4(config)#ip pim send-rp-announce loopback 0 scope 2 group-list 1
```

On R5

```
R5(config)#ip pim send-rp-discovery loopback 0 scope 4
```

If the "Debug ip pim auto-rp" was configured on this router (R5) before the above command was entered, you would see the following output:

```
Auto-RP(0): Build RP-Discovery packet
```

Note the output of the debug command reveals that R5 (The MA), received an RP-announce-message from 1.1.1.3 claiming to be the RP for all the groups within 238.0.0.0/8 range:

```
Auto-RP(0): Received RP-announce, from 1.1.1.3, RP_cnt 1, ht 181
Auto-RP(0): Added with (238.0.0.0/8, RP:1.1.1.3), PIMv2 v1
```

Next, R5 is building a Discovery packet for group to RP mapping of 238.0.0.0/8 ←→ 1.1.1.3 (R3) and sends an announcement out of its S0/0.53, S0/0.54, F0/0 and Lo0:

```
Auto-RP(0): Build RP-Discovery packet
Auto-RP:  Build mapping (238.0.0.0/8, RP:1.1.1.3), PIMv2 v1,
Auto-RP(0): Send RP-discovery packet on Serial0/0.53 (1 RP entries)
Auto-RP(0): Send RP-discovery packet on Serial0/0.54 (1 RP entries)
Auto-RP(0): Send RP-discovery packet on FastEthernet0/0 (1 RP entries)
Auto-RP: Send RP-discovery packet on Loopback0 (1 RP entries)
```

R5 receives another RP-announcement, but this time it is from R4 (1.1.1.4), claiming to be the RP for the same groups that R3 was announcing. Because R4 has a higher Source IP address, R5 the mapping agent elects R4 as the RP for groups within 238.0.0.0 /8 and once again sending RP-Discovery messages out of its S0/0.53, S0/0.54, F0/0 and Lo0:

```
Auto-RP(0): Received RP-announce, from 1.1.1.4, RP_cnt 1, ht 181
```

```
Auto-RP(0): Added with (238.0.0.0/8, RP:1.1.1.4), PIMv2 v1
Auto-RP(0): Build RP-Discovery packet
Auto-RP:  Build mapping (238.0.0.0/8, RP:1.1.1.4), PIMv2 v1,
Auto-RP(0): Send RP-discovery packet on Serial0/0.53 (1 RP entries)
Auto-RP(0): Send RP-discovery packet on Serial0/0.54 (1 RP entries)
Auto-RP(0): Send RP-discovery packet on FastEthernet0/0 (1 RP entries)
Auto-RP: Send RP-discovery packet on Loopback0 (1 RP entries)
```

To verify the configuration:

On R1

```
R1#Show ip pim rp mapping
```

PIM Group-to-RP Mappings

```
Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 00:22:04, expires: 00:02:45
```

The output of the above show command reveals the following:

- **R4 (1.1.1.4) is the RP for group 238.0.0.0 /8.**
- **The "?" indicates a failure in the name resolution.**
- **The "Info source" identifies the MA, in this case 1.1.1.5**
- **This was elected by Auto-RP process**
- **The "Uptime" section: it shows the Uptime of existing RP for this Multicast group**
- **The "Expires" section: This indicates the expiration time for RP; this timer is refreshed when the local router receives the RP-Discovery messages, which occur every 60 seconds by default.**

To verify the name resolution, the second bullet item from the above explanation:

On R1

```
R1(config)#ip host r4 1.1.1.4
R1(config)#ip host r5 1.1.1.5

R1#Show ip pim rp mapping
```

Successful Name resolution

PIM Group-to-RP Mappings

```
Group(s) 238.0.0.0/8
  RP 1.1.1.4 (R4), v2v1
    Info source: 1.1.1.5 (R5), elected via Auto-RP
```

```
         Uptime: 00:03:34, expires: 00:02:24
```

On R2

```
R2#Sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 01:03:32, expires: 00:02:57
```

On R3

```
R3#Show ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)

Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 01:05:34, expires: 00:02:52
```

On R4

```
R4#Show ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)

Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 01:09:18, expires: 00:02:08
```

On R5

```
R5#Show ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP-mapping agent (Loopback0)

Group(s) 238.0.0.0/8
```

```
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
    Uptime: 01:12:09, expires: 00:02:55
RP 1.1.1.3 (?), v2v1
 Info source: 1.1.1.3 (?), via Auto-RP
    Uptime: 01:10:50, expires: 00:02:12
```

Note the MA knows both RPs but it elected R4, because it has a higher Source IP address.

Note R5 (The MA) is aware of both RPs, but it elected R4 (1.1.1.4) because it has a higher IP address. After making this decision, it advertises R4 to the rest of the routers in RP-Discovery messages.

To test the Backup RP:

On R4

```
R4(config)#int lo0
R4(config-if)#shut
```

Note the "expires counter" is counting down, since R4 is down R5 (The MA) will NOT receive the RP-announce messages, therefore, the counter expires and the entry is removed, as follows:

```
R5#Show ip pim rp mapping | s 1.1.1.4

  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:16:07, expires: 00:01:50

R5#Show ip pim rp mapping | s 1.1.1.4

RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:16:29, expires: 00:01:28

R5#Show ip pim rp mapping | s 1.1.1.4

RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:16:58, expires: 00:00:59

R5#Show ip pim rp mapping | s 1.1.1.4

RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:17:21, expires: 00:00:36
```

```
R5#Show ip pim rp mapping | s1.1.1.4

  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:17:50, expires: 00:00:07

R5#Show ip pim rp mapping | s 1.1.1.4

  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:17:54, expires: 00:00:02

R5#Show ip pim rp mapping | s 1.1.1.4
R5#

R5#Show ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP-mapping agent (Loopback0)

Group(s) 238.0.0.0/8
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), elected via Auto-RP
         Uptime: 00:17:28, expires: 00:02:27
```

Finally the entry has expired and removed, therefore, R3 (1.1.1.3) is the RP for groups in 238.0.0.0 /8 range. If the lo0 interface of R4 is brought back up, it should once again resume its role as the primary RP:

On R4

```
R4(config)#int lo0
R4(config-if)#No shut
```

On R5

```
R5#Show ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP-mapping agent (Loopback0)

Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:30, expires: 00:02:25
  RP 1.1.1.3 (?), v2v1
```

```
    Info source: 1.1.1.3 (?), via Auto-RP
         Uptime: 00:24:12, expires: 00:02:43
```

Task 4

Configure R5 such that R6 does NOT receive the RP announcements.

To see the RP mapping on R6 before the configuration:

On R6

```
R6#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 00:05:59, expires: 00:02:59
```

The output of the above show command reveals that R6 receives the RP mapping information from R5 (The MA). The following configures filtering of these announcements:

First an Access-list is configured to deny 224.0.1.40 and permit anything else:

```
R5(config)#access-list 1 deny host 224.0.1.40
R5(config)#access-list 1 permit any
```

Once the Access-list is configured, it should be applied to the F0/0 interface facing R6, in this configuration the "IP Multicast Boundary" command is used, this command sets a boundary for administratively scoped multicast address/es.

```
R5(config)#int f0/0
R5(config-if)#ip multicast boundary 1
```

To verify the configuration:

On R6

Note R5 will no longer send the RP-Discovery messages to R6, and as a result of that, the RP mapping will expire, as follows:

```
R6#Sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 00:02:19, expires: 00:02:37

R6#Sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 00:02:59, expires: 00:01:57

R6#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 00:04:28, expires: 00:00:28

R6#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 00:04:54, expires: 00:00:03

R6#Show ip pim rp mapping

PIM Group-to-RP Mappings
R6#
```

Note once the timers expire, R6 no longer has any RP mappings.

Task 5

Configure R3 to be the RP for 224.1.1.1 ONLY.

On R3

```
R3(config)#NO access-list 1

R3(config)#access-list 1 permit host 224.1.1.1

R3#Sh ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)

Group(s) 224.1.1.1/32
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 00:15:47, expires: 00:02:03

Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 00:26:09, expires: 00:02:05
```

On R1

```
R1#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.1.1.1/32
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 00:19:42, expires: 00:02:10
Group(s) 238.0.0.0/8
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.5 (?), elected via Auto-RP
         Uptime: 00:30:04, expires: 00:02:09
```

Task 6

Configure the Lo0 interface of R1 to join group 224.1.1.1, this router should be

configured such that it responds to pings generated by **all** routers in this topology.

On R1

```
R1(config)#int lo0
R1(config-if)#ip igmp join-group 224.1.1.1
```

To test the configuration:

On R2

```
R2#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 4 ms
Reply to request 0 from 10.1.12.1, 196 ms
Reply to request 0 from 10.1.12.1, 4 ms
```

On R3

```
R3#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 60 ms
Reply to request 0 from 10.1.12.1, 212 ms
Reply to request 0 from 10.1.12.1, 164 ms
```

On R4

```
R4#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 128 ms
Reply to request 0 from 10.1.12.1, 192 ms
Reply to request 0 from 10.1.12.1, 144 ms
```

On R5

```
R5#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:
```

```
Reply to request 0 from 10.1.12.1, 124 ms
Reply to request 0 from 10.1.12.1, 220 ms
Reply to request 0 from 10.1.12.1, 140 ms
```

On R6

```
R6#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:
.
```

Since Auto-RP will not work on R6, as RP-Discovery messages are blocked by R5, static RP mapping is implemented as the solution to this task.

On R6

```
R6(config)#access-list 1 permit host 224.1.1.1
R6(config)#ip pim rp-address 1.1.1.3 1

R6#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 124 ms
Reply to request 0 from 10.1.12.1, 152 ms
```

Task 7

Configure the Lo0 interface of R1 to join group 224.10.10.10, this router should be configured such that it responds to pings generated by **all** routers in this topology.

On R1

```
R1(config-if)#int lo0
R1(config-if)#ip igmp join-group 224.10.10.10
```

To verify the configuration:

On R1

```
R1#Sh ip igmp groups

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
224.10.10.10     Loopback0                00:00:51  00:02:18  1.1.1.1
224.1.1.1        Loopback0                00:14:42  00:02:24  1.1.1.1
224.0.1.40       FastEthernet0/0          21:47:24  00:02:18  10.1.12.2
224.0.1.40       Loopback0                21:47:47  00:02:24  1.1.1.1
```

To test the configuration:

On R2

```
R2#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 1 ms
Reply to request 0 from 10.1.12.1, 1 ms
```

On R3

```
R3#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 205 ms
Reply to request 0 from 10.1.12.1, 233 ms
```

On R4

```
R4#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 76 ms
Reply to request 0 from 10.1.12.1, 104 ms
```

On R5

```
R5#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 116 ms
```

```
Reply to request 0 from 10.1.12.1, 240 ms
Reply to request 0 from 10.1.12.1, 212 ms
```

On R6

```
R6#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 148 ms
Reply to request 0 from 10.1.12.1, 164 ms

R6#Show ip mroute | s 224.10.10.10

(*, 224.10.10.10), 00:08:37/00:00:45, RP 0.0.0.0, flags: D

  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse-Dense, 00:08:37/00:00:00
```

(The output of the above show command is modified to show the important section)

Note the reason R6 was able to reach 224.10.10.10 is because the interface of this router is configured as sparse-dense-mode, and by default, when an interface is configured with the "ip pim sparse-dense-mode", the router will use PIM SM for groups, which it has an RP mappings for, and PIM DM for groups, which it does not.

Task 8

Configure the routers to disable the use of PIM-DM for groups with no RP.

On All Routers:

```
Rx(config)#NO ip pim dm-fallback
```

To test the configuration:

On R2

```
R2#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
```

.

On R3

```
R3#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
.
```

On R4

```
R4#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
.
```

On R5

```
R5#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
.
```

On R6

```
R6#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
.
```

To test the configuration further

On All Routers

```
Rx(config)#ip pim dm-fallback
```

On R2

```
R2#Ping 224.10.10.10
```

```
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 8 ms
Reply to request 0 from 10.1.12.1, 8 ms
```

On R3

```
R3#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 285 ms
Reply to request 0 from 10.1.12.1, 393 ms
```

On R4

```
R4#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 96 ms
Reply to request 0 from 10.1.12.1, 144 ms
```

On R5

```
R5#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 184 ms
Reply to request 0 from 10.1.12.1, 244 ms
Reply to request 0 from 10.1.12.1, 216 ms
```

On R6

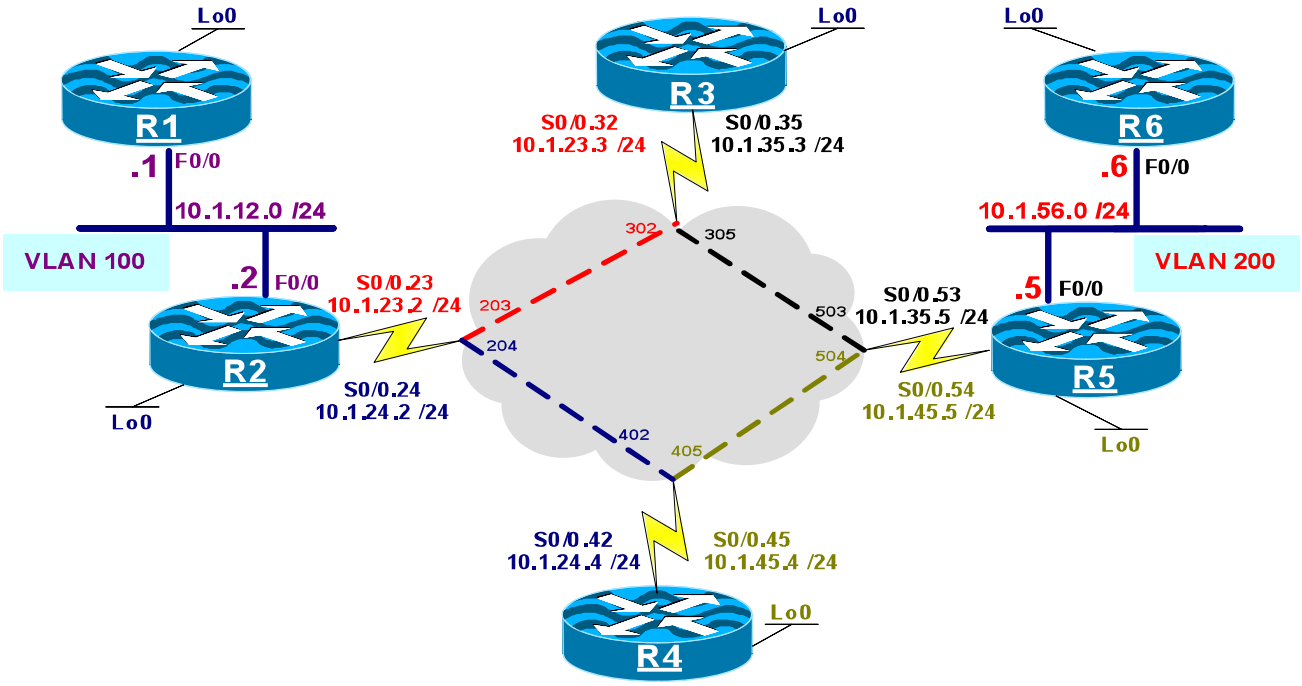
```
R6#Ping 224.10.10.10

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.10.10.10, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 221 ms
Reply to request 0 from 10.1.12.1, 281 ms
```

Note the routers can reach 224.10.10.10 once again using PIM Dense mode.

Task 9

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 5 – Auto-RP Filtering & Listener



Lab Setup:

- Configure the F0/0 interface of R1 and R2 in VLAN 100, and F0/0 interface of R4 and R5 in VLAN 200.
- All frame-Relay connections must be configured in a point-to-point manner.
- Use the IP addressing chart below for IP addressing scheme

IP addressing Chart:

Router	Interface / IP addressing
R1	F0/0 = 10.1.12.1 /24 Lo0= 1.1.1.1 /32
R2	**F0/0 = 10.1.12.2 /24** **S0/0.23 = 10.1.23.2/24** **S0/0.24 = 10.1.24.2/24** **Lo0= 1.1.1.2 /32**
R3	S0/0.32 = 10.1.23.3/24 S0/0.35 = 10.1.35.3/24 Lo0= 1.1.1.3 /32
R4	**S0/0.42 = 10.1.24.4/24** **S0/0.45 = 10.1.45.4 /24** **Lo0= 1.1.1.4 /32**
R5	S0/0.53 = 10.1.35.5/24 S0/0.54 = 10.1.45.5/24 F0/0 = 10.1.56.5/24 Lo0= 1.1.1.5 /32
R6	**F0/0 = 10.1.56.6/24** **Lo0= 1.1.1.6/32**

Task 1

Configure OSPF area 0 on all interfaces.

On All Routers

```
Rx(config)#router ospf 1
Rx(config-router)#netw 0.0.0.0 0.0.0.0 are 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf

     1.0.0.0/32 is subnetted, 6 subnets
O       1.1.1.3 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.2 [110/2] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.5 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.4 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.6 [110/131] via 10.1.12.2, 00:00:19, FastEthernet0/0
```

```
     10.0.0.0/24 is subnetted, 6 subnets
O       10.1.24.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.45.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.35.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.56.0 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
```

Task 2

Configure PIM on the interfaces according to the following table:

Router	Interface / PIM
R1	F0/0 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R2	**F0/0 = PIM Sparse-mode** **S0/0.23 = PIM Sparse-mode** **S0/0.24 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**
R3	S0/0.32 = PIM Sparse-mode S0/0.35 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R4	**S0/0.42 = PIM Sparse-mode** **S0/0.45 = PIM Sparse-mode** **Lo0= PIM Sparse-mode**
R5	F0/0 = PIM Sparse-mode S0/0.53 = PIM Sparse-mode S0/0.54 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R6	**F0/0 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**

On R1

```
R1(config)#ip multicast-routing

R1(config)#int lo 0
R1(config-if)#ip pim sparse-mode

R1(config-if)#int f0/0
R1(config-if)#ip pim sparse-mode
```

On R2

```
R2(config)#ip multicast-routing

R2(config)#int lo0
R2(config-if)#ip pim sparse-mode

R2(config-if)#int f0/0
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.23
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.24
R2(config-subif)#ip pim sparse-mode
```

On R3

```
R3(config)#ip multicast-routing

R3(config)#int lo0
R3(config-if)#ip pim sparse-mode

R3(config-if)#int s0/0.32
R3(config-subif)#ip pim sparse-mode

R3(config-subif)#int s0/0.35
R3(config-subif)#ip pim sparse-mode
```

On R4

```
R4(config)#ip multicast-routing

R4(config)#int lo0
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.42
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.45
R4(config-if)#ip pim sparse-mode
```

On R5

```
R5(config)#ip multicast-routing
```

```
R5(config)#int lo0
R5(config-if)#ip pim sparse-mode

R5(config-if)#int f0/0
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.53
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.54
R5(config-subif)#ip pim sparse-mode
```

On R6

```
R6(config)#ip multicast-routing

R6(config)#int lo0
R6(config-if)#ip pim sparse-mode

R6(config-if)#int F0/0
R6(config-if)#ip pim sparse-mode
```

To verify the configuration:

On R1

```
R1#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.2         FastEthernet0/0          00:00:19/00:01:25 v2    1 / DR S
```

On R2

```
R2#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.1         FastEthernet0/0          00:00:58/00:01:15 v2    1 / S
10.1.23.3         Serial0/0.23             00:03:24/00:01:17 v2    1 / S
10.1.24.4         Serial0/0.24             00:03:03/00:01:37 v2    1 / S
```

On R3

```
R3#Show ip pim neighbor | b interface
```

```
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.23.2         Serial0/0.32             00:04:33/00:01:36 v2    1 / S
10.1.35.5         Serial0/0.35             00:03:49/00:01:22 v2    1 / S
```

On R4

```
R4#Show ip pim neighbor | b inter

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.24.2         Serial0/0.42             00:05:01/00:01:38 v2    1 / S
10.1.45.5         Serial0/0.45             00:04:29/00:01:40 v2    1 / S
```

On R5

```
R5#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.6         FastEthernet0/0          00:05:23/00:01:15 v2    1 / DR S
10.1.35.3         Serial0/0.53             00:05:39/00:01:29 v2    1 / S
10.1.45.4         Serial0/0.54             00:05:31/00:01:37 v2    1 / S
```

On R6

```
R6#Show ip pim neighbor

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.5         FastEthernet0/0          00:00:44/00:01:29 v2    1 / S
```

Task 3

Configure R1 to be the RP and advertise RP to group mappings, the timer for both roles should be set to 5 seconds; you should use Auto-RP to accomplish this task. R1 should be configured such that R6 does NOT receive RP-Discovery messages. DO NOT change the PIM mode on any interface to accomplish this task.

On R1

```
R1(config)#ip pim send-rp-announce lo0 scope 5 interval 5
R1(config)#ip pim send-rp-discovery lo0 scope 5 interval 5
```

To verify the configuration:

On R1

```
R1#Sh ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)

Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:00:12, expires: 00:00:13
```

On R2

```
R2#Show ip pim rp mapping

PIM Group-to-RP Mappings
Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:01:34, expires: 00:00:11
```

On R3

```
R3#Show ip pim rp mapping
PIM Group-to-RP Mappings
R3#
```

On R4

```
R4#Show ip pim rp mapping
PIM Group-to-RP Mappings
R4#
```

On R5

```
R5#Show ip pim rp mapping
PIM Group-to-RP Mappings
R5#
```

On R6

```
R6#Show ip pim rp mapping
PIM Group-to-RP Mappings
R6#
```

NOTE: None of the routers have the RP mapping except R2, R2 gets the RP mapping because it is directly connected to R1, but because it's configured as "Sparse-mode" it will NOT forward that information, therefore, None of the other routers will receive the RP mappings.

To verify the information:

On R2

```
R2#Show ip pim inter count

State: * - Fast Switched, D - Distributed Fast Switched
       H - Hardware Switching Enabled
Address          Interface                FS  Mpackets In/Out
1.1.1.2          Loopback0                *   0/0
10.1.12.2        FastEthernet0/0          *   100/0
10.1.23.2        Serial0/0.23             *   0/0
10.1.24.2        Serial0/0.24             *   0/0
```

As you can see in the output of the above command the local router (R2) receives 100 packets through its F0/0 interface BUT it does NOT send any M-packets OUT.

There are several solutions to this problem, one way to fix this problem is to use "spare-dense-mode" configuration, which means that the router will use SM for every G that has an RP and DM for any G whose RP is unknown. Since the groups 224.0.1.39 and 224.0.1.40 are unknown, the routers will use DM to receive the RP mappings.

In this task we are not allowed to change the PIM mode of the interface, therefore, one way to fix this problem is to configure the "ip pim autorp listener" command, which allows the two group addresses used by RP and the MA (224.0.1.39 and 224.0.1.40) to work in DM and SM for the rest of the groups.

To configure the task:

On All Routers

```
Rx(config)#ip pim autorp listener
```

To verify the configuration:

On R2

```
R2#Show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 02:46:07, expires: 00:00:14
```

On R3

```
R3#Show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:02:53, expires: 00:00:11
```

On R4

```
R4#Show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:03:40, expires: 00:00:11
```

On R5

```
R5#Show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:04:24, expires: 00:00:11
```

On R6

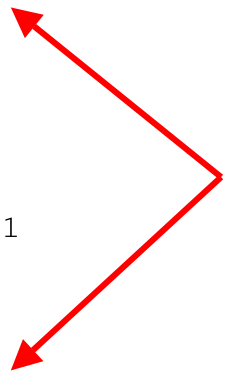
```
R6#Show ip pim rp mapping
PIM Group-to-RP Mappings
```

```
Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:09:34, expires: 00:00:12
```

NOTE: All the routers have the RP Mappings, to see the dense mode operation of the two groups (224.0.1.39, and 224.0.1.40):

On R2

```
R2#Sh ip mroute | s 224.0.1.39|224.0.1.40

(*, 224.0.1.39), 00:15:57/stopped, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.24, Forward/Sparse, 00:15:57/00:00:00
    Serial0/0.23, Forward/Sparse, 00:15:57/00:00:00
    FastEthernet0/0, Forward/Sparse, 00:15:57/00:00:00
(1.1.1.1, 224.0.1.39), 00:00:04/00:02:55, flags: T
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.12.1

  Outgoing interface list:
    Serial0/0.23, Forward/Sparse, 00:00:04/00:00:00
    Serial0/0.24, Forward/Sparse, 00:00:04/00:00:00
(*, 224.0.1.40), 00:15:57/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.24, Forward/Sparse, 00:15:57/00:00:00
    Serial0/0.23, Forward/Sparse, 00:15:57/00:00:00
    FastEthernet0/0, Forward/Sparse, 00:15:57/00:00:00
    Loopback0, Forward/Sparse, 00:15:57/00:00:00

(The rest of the output is omitted)
```



********** Informational ONLY **********

Because R6 is directly connected to R5, we did not have to configure R6 with the "IP PIM autorp listener" command; R6 would receive the RP mapping, but understand that R6 will NOT forward it to any of its future downstream routers.

The following examines the mroute table of R5:

On R5

```
R5#Show ip mroute | S 224.0.1.40
```

```
(*, 224.0.1.40), 00:28:02/stopped, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:28:02/00:00:00
    Serial0/0.54, Forward/Sparse, 00:28:02/00:00:00
    Serial0/0.53, Forward/Sparse, 00:28:02/00:00:00
    Loopback0, Forward/Sparse, 00:28:02/00:00:00
(1.1.1.1, 224.0.1.40), 00:28:00/00:02:57, flags: LT
  Incoming interface: Serial0/0.54, RPF nbr 10.1.45.4
  Outgoing interface list:
    Loopback0, Forward/Sparse, 00:28:00/00:00:00
    Serial0/0.53, Forward/Sparse, 00:28:00/00:00:00
    FastEthernet0/0, Forward/Sparse, 00:28:00/00:00:00
```

R5 is forwarding RP mappings to R6 because in DM (remember that with the *ip pim autorp listener* command, the group 224.0.1.40 is operating in DM) a router sends Mcast traffic on all of its PIM enabled interfaces as long it has NOT received a prune message.

****

Next we must configure R1 such that R6 does NOT receive the RP mapping, or the RP discovery messages do NOT reach this router. The "scope" keyword defines the TTL of the RP-Discovery messages; therefore, if it is set to 4, it should NOT reach R6.

On R1

Note by setting the scope of these messages to 4, R6 will never receive the rp-discovery messages.

```
R1(config)#ip pim send-rp-discovery lo0 scope 4 interval 5
```

To verify the configuration:

On R6

```
R6#Sh ip pim rp mapping
PIM Group-to-RP Mappings
R6#
```

Note R6 no longer has the RP mapping.

Task 4

Configure R2 to send Auto-RP announce messages announcing itself as the RP for all groups, this router should send its announce messages every 10 seconds, with a TTL of 5; the source IP address of these announce messages should be based on its loopback 0 interface; ensure that R2 is **never** elected as the RP for any of the groups.

First part of this task is to configure R2 as the RP for all groups:

On R2

```
R2(config)#ip pim send-rp-announce lo0 scope 5 interval 10
```

To verify the configuration:

On R1

```
R1#Sh ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)

Group(s) 224.0.0.0/4
  RP 1.1.1.2 (?), v2v1
    Info source: 1.1.1.2 (?), elected via Auto-RP
         Uptime: 00:02:16, expires: 00:01:41
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), via Auto-RP
         Uptime: 00:15:13, expires: 00:00:14
```

It should be no surprise that R2 is elected as the RP, since it has the higher source IP address.

To resolve this task, filtering can be applied on the MA, since in Auto-RP it's the mapping agent that decides which router is elected as the primary RP; the MA can be configured to filter R2 all together as follows:

On R1

```
R1(config)#access-list 1 permit host 1.1.1.2
R1(config)#access-list 2 deny any

R1(config)#ip pim rp-announce-filter rp-list 1 group-list 2
```

RP-List – this keyword references an access-list, in this access-list the IP address(es) that are permitted are the IP address(es) that the filtering is performed on. The IP address(es) that are implicitly or explicitly denied will NOT be subject to this filtering.

Therefore, RPs that is matched by the RP-List (Allowed by permit statement in the ACL) will have their Mcast groups filtered by group-list. In this case, the access-list referenced by the Group-List denies "any", which means that The RP that is permitted in the ACL 1 is denied to be the RP for the groups referenced in ACL 2, in this case all groups.

To verify the configuration:

On R1

```
R1#Show ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)

Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:41:55, expires: 00:00:10
```

On R2

```
R2#Sh ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:21:31, expires: 00:00:11
```

On R3

```
R3#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:22:01, expires: 00:00:11
```

On R4

```
R4#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:22:36, expires: 00:00:11
```

On R5

```
R5#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:23:21, expires: 00:00:11
```

On R6

```
R6#Show ip pim rp mapping

PIM Group-to-RP Mappings
R6#
```

Note R6 does NOT have the RP mappings, because of the configured "Scope" keyword on R1.

Task 5

Configure R3 and R4 as RPs for groups 224.1.1.1 to 224.1.1.10; R1 should be re-configured to be the Mapping Agent ONLY.

On R3 and R4

```
R3(config)#ip pim send-rp-announce lo0 scope 5 group-list 10 interval 5

R3(config)#access-list 10 permit host 224.1.1.1
R3(config)#access-list 10 permit host 224.1.1.2
```

```
R3(config)#access-list 10 permit host 224.1.1.3
R3(config)#access-list 10 permit host 224.1.1.4
R3(config)#access-list 10 permit host 224.1.1.5
R3(config)#access-list 10 permit host 224.1.1.6
R3(config)#access-list 10 permit host 224.1.1.7
R3(config)#access-list 10 permit host 224.1.1.8
R3(config)#access-list 10 permit host 224.1.1.9
R3(config)#access-list 10 permit host 224.1.1.10
```

To verify the configuration:

On R1

```
R1#Show ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback0)

Group(s) 224.0.0.0/4
  RP 1.1.1.1 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:37:48, expires: 00:00:12

Group(s) 224.1.1.1/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:26, expires: 00:00:14
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), via Auto-RP
         Uptime: 00:04:00, expires: 00:00:15

Group(s) 224.1.1.2/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:26, expires: 00:00:14
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), via Auto-RP
         Uptime: 00:03:55, expires: 00:00:15

Group(s) 224.1.1.3/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:26, expires: 00:00:14
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), via Auto-RP
```

```
         Uptime: 00:03:55, expires: 00:00:15

Group(s) 224.1.1.4/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:27, expires: 00:00:13
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), via Auto-RP
         Uptime: 00:03:52, expires: 00:00:13

Group(s) 224.1.1.5/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:27, expires: 00:00:12
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), via Auto-RP
         Uptime: 00:03:52, expires: 00:00:13

Group(s) 224.1.1.6/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:27, expires: 00:00:13
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), via Auto-RP
         Uptime: 00:03:52, expires: 00:00:13

Group(s) 224.1.1.7/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:27, expires: 00:00:12
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), via Auto-RP
         Uptime: 00:03:54, expires: 00:00:11

Group(s) 224.1.1.8/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:29, expires: 00:00:11
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), via Auto-RP
         Uptime: 00:03:54, expires: 00:00:11
Group(s) 224.1.1.9/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:29, expires: 00:00:11
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), via Auto-RP
```

```
        Uptime: 00:03:49, expires: 00:00:11

Group(s) 224.1.1.10/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.4 (?), elected via Auto-RP
         Uptime: 00:00:29, expires: 00:00:10
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.3 (?), via Auto-RP
         Uptime: 00:03:51, expires: 00:00:14
```

Note even though both RPs are seen by the Mapping Agent, the Mapping Agent chooses R4 because it has a higher IP address.

Task 6

Configure the Mapping Agent such that it announces R3 as the RP for the odd numbered groups (224.1.1.1, 224.1.1.3, 224.1.1.5….) and R4 for the even numbered groups (224.1.1.2, 224.1.1.4, 224.1.1.6……) within the previously specified range.

On R1

```
R1(config)#ip access-list standard R3
R1(config-std-nacl)#permit host 1.1.1.3

R1(config)#ip access-list standard R4
R1(config-std-nacl)#permit host 1.1.1.4

R1(config)#ip access-list standard Even-G
R1(config-std-nacl)#permit host 224.1.1.2

R1(config-std-nacl)#permit host 224.1.1.4
R1(config-std-nacl)#permit host 224.1.1.6
R1(config-std-nacl)#permit host 224.1.1.8
R1(config-std-nacl)#permit host 224.1.1.10

R1(config)#IP access-list standard Odd-G
R1(config-std-nacl)#permit host 224.1.1.1
R1(config-std-nacl)#permit host 224.1.1.3
R1(config-std-nacl)#permit host 224.1.1.5
R1(config-std-nacl)#permit host 224.1.1.7
R1(config-std-nacl)#permit host 224.1.1.9
```

NOTE: The following access-lists will also work:

```
R1(config)#ip access-list standard Even-G
R1(config-std-nacl)#permit 224.1.1.2 0.0.0.254

R1(config)#ip access-list standard Odd-G
R1(config-std-nacl)#permit 224.1.1.1 0.0.0.254
```

```
R1(config)#ip pim rp-announce-filter rp-list R3 group-list Odd-G
R1(config)#ip pim rp-announce-filter rp-list R4 group-list Even-G
```

To verify the configuration:

On R2

```
R2#Show ip pim rp mapping

PIM Group-to-RP Mappings
This system is an RP (Auto-RP)

Group(s) 224.1.1.1/32
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:00:42, expires: 00:00:14
Group(s) 224.1.1.2/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:18:10, expires: 00:00:14
Group(s) 224.1.1.3/32
  RP 1.1.1.3 (?), v2v1

    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:00:42, expires: 00:00:14
Group(s) 224.1.1.4/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:18:10, expires: 00:00:14
Group(s) 224.1.1.5/32
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:00:42, expires: 00:00:14
Group(s) 224.1.1.6/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
```

```
         Uptime: 00:18:11, expires: 00:00:12
Group(s) 224.1.1.7/32
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:00:42, expires: 00:00:12
Group(s) 224.1.1.8/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:18:11, expires: 00:00:13
Group(s) 224.1.1.9/32
  RP 1.1.1.3 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:00:43, expires: 00:00:12
Group(s) 224.1.1.10/32
  RP 1.1.1.4 (?), v2v1
    Info source: 1.1.1.1 (?), elected via Auto-RP
         Uptime: 00:18:11, expires: 00:00:12
```

Task 7

Configure the lo0 interface of R4 to join 224.1.1.1 and ensure every router but R6 can ping this address.

On R4

```
R4(config)#int lo0
R4(config-if)#ip igmp join-group 224.1.1.1
```

To test the configuration:

On R1

```
R1#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.24.4, 116 ms
Reply to request 0 from 10.1.24.4, 140 ms
```

On R2

```
R2#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.24.4, 180 ms
Reply to request 0 from 10.1.45.4, 364 ms
Reply to request 0 from 10.1.45.4, 336 ms
Reply to request 0 from 10.1.45.4, 316 ms
```

On R3

```
R3#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.45.4, 204 ms
Reply to request 0 from 10.1.45.4, 420 ms
Reply to request 0 from 10.1.45.4, 360 ms
```

On R4

```
R4#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 1.1.1.4, 4 ms
```

On R5

```
R5#Ping 224.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.45.4, 112 ms
Reply to request 0 from 10.1.45.4, 464 ms
Reply to request 0 from 10.1.45.4, 252 ms
```

On R6

```
R6#Ping 224.1.1.1
```

```
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.45.4, 60 ms
```

Task 8

Configure R2 to do SPT switch only if traffic for the 224.1.1.1 group reaches 20Kb/sec.

By default, IOS will do SPT switch ASAP the first packet arrives for the *,G entry. In order to change the default value to 20 Kbps ONLY for 224.1.1.1 group, an access-list is configured to identify the group, and then, the access-list is applied to the "ip pim spt-threshold" command, as follows:

On R2

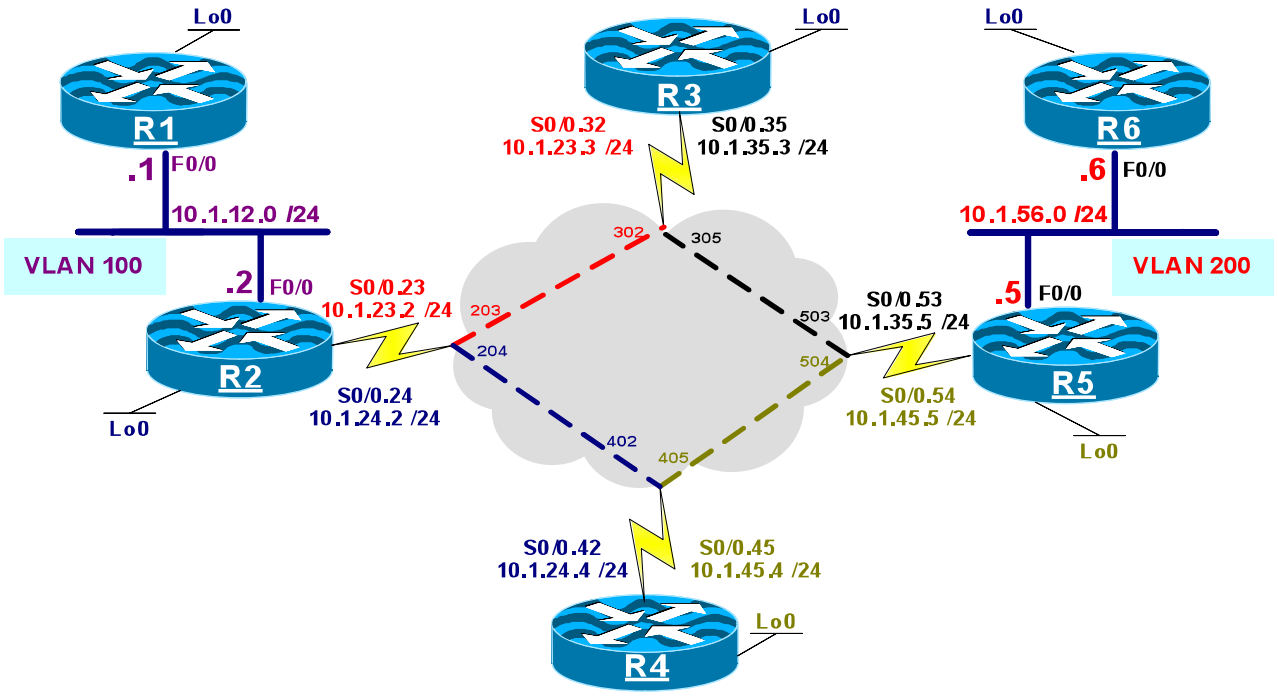
```
R2(config)#access-list 1 permit host 224.1.1.1

R2(config)#ip pim spt-threshold 20 group-list 1
```

Without the "group-list" keyword, IOS will wait for any (*,G) traffic threshold to reach 20Kb/sec before performing the switchover. The "group-list" command instructs the router to perform a switchover ONLY for the permitted group.

Task 9

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 6 – Configuring BSR



Lab Setup

- Configure the F0/0 interface of R1 and R2 in VLAN 100, and F0/0 interface of R4 and R5 in VLAN 200.
- All frame-Relay connections must be configured in a point-to-point manner.
- Use the IP addressing chart below for IP addressing scheme

IP addressing Chart:

Router	Interface / IP addressing
R1	F0/0 = 10.1.12.1 /24 Lo0= 1.1.1.1 /32
R2	**F0/0 = 10.1.12.2 /24** **S0/0.23 = 10.1.23.2/24** **S0/0.24 = 10.1.24.2/24** **Lo0= 1.1.1.2 /32**
R3	S0/0.32 = 10.1.23.3/24 S0/0.35 = 10.1.35.3/24 Lo0= 1.1.1.3 /32
R4	**S0/0.42 = 10.1.24.4/24** **S0/0.45 = 10.1.45.4 /24** **Lo0= 1.1.1.4 /32**
R5	S0/0.53 = 10.1.35.5/24 S0/0.54 = 10.1.45.5/24 F0/0 = 10.1.56.5/24 Lo0= 1.1.1.5 /32
R6	**F0/0 = 10.1.56.6/24** **Lo0= 1.1.1.6/32**

Task 1

Configure OSPF area 0 on all interfaces.

On All Routers

```
Rx(config)#router ospf 1
Rx(config-router)#netw 0.0.0.0 0.0.0.0 are 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf

     1.0.0.0/32 is subnetted, 6 subnets
O       1.1.1.3 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.2 [110/2] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.5 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.4 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.6 [110/131] via 10.1.12.2, 00:00:19, FastEthernet0/0
```

```
     10.0.0.0/24 is subnetted, 6 subnets
O       10.1.24.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.45.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.35.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.56.0 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
```

Task 2

Configure PIM on the interfaces according to the following table:

Router	Interface / PIM
R1	F0/0 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R2	**F0/0 = PIM Sparse-mode** **S0/0.23 = PIM Sparse-mode** **S0/0.24 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**
R3	S0/0.32 = PIM Sparse-mode S0/0.35 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R4	**S0/0.42 = PIM Sparse-mode** **S0/0.45 = PIM Sparse-mode** **Lo0= PIM Sparse-mode**
R5	F0/0 = PIM Sparse-mode S0/0.53 = PIM Sparse-mode S0/0.54 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R6	**F0/0 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**

On R1

```
R1(config)#ip multicast-routing

R1(config)#int lo 0
R1(config-if)#ip pim sparse-mode

R1(config-if)#int f0/0

R1(config-if)#ip pim sparse-mode
```

On R2

```
R2(config)#ip multicast-routing

R2(config)#int lo0
R2(config-if)#ip pim sparse-mode

R2(config-if)#int f0/0
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.23
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.24
R2(config-subif)#ip pim sparse-mode
```

On R3

```
R3(config)#ip multicast-routing

R3(config)#int lo0
R3(config-if)#ip pim sparse-mode

R3(config-if)#int s0/0.32
R3(config-subif)#ip pim sparse-mode

R3(config-subif)#int s0/0.35
R3(config-subif)#ip pim sparse-mode
```

On R4

```
R4(config)#ip multicast-routing

R4(config)#int lo0
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.42
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.45
R4(config-if)#ip pim sparse-mode
```

On R5

```
R5(config)#ip multicast-routing
```

```
R5(config)#int lo0
R5(config-if)#ip pim sparse-mode

R5(config-if)#int f0/0
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.53
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.54
R5(config-subif)#ip pim sparse-mode
```

On R6

```
R6(config)#ip multicast-routing

R6(config)#int lo0
R6(config-if)#ip pim sparse-mode

R6(config-if)#int f0/0
R6(config-if)#ip pim sparse-mode
```

To verify the configuration:

On R1

```
R1#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.2         FastEthernet0/0          00:01:31/00:01:42 v2    1 / DR S
```

On R2

```
R2#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.1         FastEthernet0/0          00:03:19/00:01:21 v2    1 / S
10.1.23.3         Serial0/0.23             00:02:59/00:01:42 v2    1 / S
10.1.24.4         Serial0/0.24             00:00:09/00:01:35 v2    1 / S
```

On R3

```
R3#Show ip pim neighbor | b interface
```

```
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.23.2         Serial0/0.32             00:01:10/00:01:32 v2    1 / S
10.1.35.5         Serial0/0.35             00:00:33/00:01:40 v2    1 / S
```

On R4

```
R4#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.45.5         Serial0/0.45             00:04:05/00:01:36 v2    1 / S
10.1.24.2         Serial0/0.42             00:01:58/00:01:15 v2    1 / S
```

On R5

```
R5#Show ip pim neighbor

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.6         FastEthernet0/0          00:00:19/00:01:25 v2    1 / DR S
10.1.35.3         Serial0/0.53             00:00:33/00:01:41 v2    1 / S
10.1.45.4         Serial0/0.54             00:00:27/00:01:17 v2    1 / S
```

On R6

```
R6#Show ip pim neighbor

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.5         FastEthernet0/0          00:00:19/00:01:25 v2    1 / S
```

Task 3

Configure R3 and R4 as the RP for all groups, R3 should be the primary and R4 should be the backup RP, these routers should use their Lo0 IP address as the source of their traffic. You should NOT change the PIM mode or change the IP addressing of the interface(s) to accomplish this task.

The following identifies the possible choices:

- **Auto-RP**
- **BSR**

- **Anycast RP**

Auto-RP cannot be used because R4 has a higher IP address and therefore, it will always be chosen as the primary RP.

Anycast RP will not work because an RP will always choose itself since the IP address of Lo0 is directly connected and the router will always prefer the directly connected over any route received by an IGP. Therefore, BSR is the ONLY choice.

The question should be where to place the BSR routers. The answer is any two routers. Why two? Well, the task asks for a redundant mapping.

In the following solution RP, candidates are also configured as the BSRs.

A "Debug ip pim bsr" is configured to see the messages generated by the RP and BSR.

On R3

```
R3#Debug ip pim bsr

R3(config)#ip pim rp-candidate loopback 0
```

You should see the following debug output on your console:

```
PIM-BSR(0): Build v2 Candidate-RP advertisement for 1.1.1.3 priority 0,
holdtime 150
PIM-BSR(0):  Candidate RP's group prefix 224.0.0.0/4
PIM-BSR(0): no bootstrap router address
```

Note the advertisements are generated from the IP address of the loopback 0 interface, the priority is set to zero and a holdtime is set to 150, the holdtime is 2.5 times the interval in which these messages are generated, by default these messages are generated every 60 seconds. Since a BSR is NOT configured, the last line of the debug states that the local router does not see a BSR. Let's configure the BSR:

```
R3(config)#ip pim bsr-candidate lo0

PIM-BSR(0): Build v2 Candidate-RP advertisement for 1.1.1.3 priority 0,
holdtime 150
PIM-BSR(0):  Candidate RP's group prefix 224.0.0.0/4
PIM-BSR(0): Send Candidate RP Advertisement to 1.1.1.3
PIM-BSR(0):  RP 1.1.1.3, 1 Group Prefixes, Priority 0, Holdtime 150
PIM-BSR(0): RP-set for 224.0.0.0/4
PIM-BSR(0):   RP(1) 1.1.1.3, holdtime 150 sec priority 0
PIM-BSR(0): Bootstrap message for 1.1.1.3 originated
```

Note now that C-RP knows the IP address of the BSR, it sends a Unicast message to the BSR router to identify itself as a C-RP.

On R4

```
R4(config)#ip pim rp-candidate lo0 priority 200
R4(config)#ip pim bsr-candidate lo0
```

In the above configuration, R4 is configured as C-RP with a priority of 200, since a lower priority value has more preference; R3 is elected as the RP.

Contrary to RP election, where lower priority has more preference, in BSR election, the higher priority has more preference and the higher IP address is used as the tiebreaker.

To verify the configuration:

On R3

```
R3#U all

R3#Show ip pim rp mapping

PIM Group-to-RP Mappings
This system is a candidate RP (v2)

Group(s) 224.0.0.0/4
  RP 1.1.1.3 (?), v2
    Info source: 1.1.1.4 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 00:04:18, expires: 00:02:16
  RP 1.1.1.4 (?), v2
    Info source: 1.1.1.4 (?), via bootstrap, priority 200, holdtime 150
         Uptime: 00:01:18, expires: 00:02:16
```

R4 and R3 are both configured as C-BSRs, the priority of these routers are set to zero, therefore, the router with a higher source IP address (1.1.1.4 versus 1.1.1.3) is elected as the BSR. Furthermore, R4 is also elected as the C-RP, since it has a lower priority than R3.

Contrary to Auto-RP, in BSR, the C-BSR collects all RP announcements and creates an RP-SET; the RP-SET is then advertised to all DRs.

Once the DRs receive the RP-SET, they will elect an RP from the list for a given group. Luckily, all DR routers use the same calculations to elect an RP, therefore, the result of the election will be consistent across all routers.

To verify the configuration:

On R1

```
R1#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.3 (?), v2
    Info source: 1.1.1.4 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 00:38:20, expires: 00:02:21
  RP 1.1.1.4 (?), v2
    Info source: 1.1.1.4 (?), via bootstrap, priority 200, holdtime 150
         Uptime: 00:29:37, expires: 00:02:19
```

On R2

```
R2#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.3 (?), v2
    Info source: 1.1.1.4 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 00:39:08, expires: 00:01:32
  RP 1.1.1.4 (?), v2
    Info source: 1.1.1.4 (?), via bootstrap, priority 200, holdtime 150
         Uptime: 00:30:26, expires: 00:01:35
```

On R5

```
R5#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.3 (?), v2
    Info source: 1.1.1.4 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 00:40:02, expires: 00:01:39
  RP 1.1.1.4 (?), v2
    Info source: 1.1.1.4 (?), via bootstrap, priority 200, holdtime 150
         Uptime: 00:31:19, expires: 00:01:39
```

On R6

```
R6#Show ip pim rp mapping
```

```
PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 1.1.1.3 (?), v2
    Info source: 1.1.1.4 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 00:40:19, expires: 00:02:22
  RP 1.1.1.4 (?), v2
    Info source: 1.1.1.4 (?), via bootstrap, priority 200, holdtime 150
         Uptime: 00:31:37, expires: 00:02:21
```

To further verify the configuration:

On R1

```
R1#Show ip pim bsr-router

PIMv2 Bootstrap information
  BSR address: 1.1.1.4 (?)
  Uptime:      00:37:01, BSR Priority: 0, Hash mask length: 0
  Expires:     00:01:12
```

On R2

```
R2#Show ip pim bsr-router

PIMv2 Bootstrap information
  BSR address: 1.1.1.4 (?)
  Uptime:      00:39:50, BSR Priority: 0, Hash mask length: 0
  Expires:     00:01:23
```

On R3

```
R3#Show ip pim bsr-router

PIMv2 Bootstrap information
  BSR address: 1.1.1.4 (?)
  Uptime:      00:14:57, BSR Priority: 0, Hash mask length: 0
  Expires:     00:01:12
This system is a candidate BSR
  Candidate BSR address: 1.1.1.3, priority: 0, hash mask length: 0
  Candidate RP: 1.1.1.3(Loopback0)
    Holdtime 150 seconds
    Advertisement interval 60 seconds
    Next advertisement in 00:00:19
```

On R4

```
R4#Show ip pim bsr-router

PIMv2 Bootstrap information
This system is the Bootstrap Router (BSR)
  BSR address: 1.1.1.4 (?)
  Uptime:      00:17:17, BSR Priority: 0, Hash mask length: 0
  Next bootstrap message in 00:00:43
  Candidate RP: 1.1.1.4(Loopback0)
    Holdtime 150 seconds
    Advertisement interval 60 seconds
    Next advertisement in 00:00:35
    Candidate RP priority : 200
```

On R5

```
R5#Show ip pim bsr-router

PIMv2 Bootstrap information
  BSR address: 1.1.1.4 (?)
  Uptime:      00:48:04, BSR Priority: 0, Hash mask length: 0
  Expires:     00:01:10
```

On R6

```
R6#Show ip pim bsr-router

PIMv2 Bootstrap information
  BSR address: 1.1.1.4 (?)
  Uptime:      00:48:40, BSR Priority: 0, Hash mask length: 0
  Expires:     00:01:34
```

Task 4

Configure the Lo0 interface of R1 to join 239.1.1.1, ensure that R3, R4 and R5 can successfully ping this group.

On R1

```
R1(config)#int lo0
R1(config-if)#ip igmp join 239.1.1.1
```

On R3

```
R3#Ping 239.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.12.1, 56 ms
Reply to request 0 from 10.1.12.1, 188 ms
Reply to request 0 from 10.1.12.1, 104 ms
```

On R4

```
R4#Ping 239.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.12.1, 72 ms
Reply to request 0 from 10.1.12.1, 157 ms
Reply to request 0 from 10.1.12.1, 145 ms
```

On R5

```
R5#Ping 239.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.12.1, 133 ms
Reply to request 0 from 10.1.12.1, 257 ms
Reply to request 0 from 10.1.12.1, 173 ms
Reply to request 0 from 10.1.12.1, 145 ms
```

On R5

```
R5#Show ip mroute |s 239.1.1.1

(*, 239.1.1.1), 00:04:31/stopped, RP 1.1.1.3, flags: SPF
  Incoming interface: Serial0/0.53, RPF nbr 10.1.35.3
  Outgoing interface list: Null
     (The rest of the output is omitted)
```

As expected the RP for the group is R3 because it has a lower RP priority.

Task 5

Configure the RPs to rate limit the number of register messages to 10/sec.

The "IP pim register-rate-limit" global configuration command sets a limit on the maximum number of PIM SM register messages sent per second for each (S,G) entry. This command limits the load on DR and RP and drops the register messages that exceed the threshold.

On R3

```
R3(config)#ip pim register-rate-limit 10
```

On R4

```
R5(config)#ip pim register-rate-limit 10
```

Task 6

Configure R2 to filter pings, which are locally originated from R6's Lo0. Do not use an access-list to accomplish this task.

By default, IOS will send locally generated Mcast packets with a TTL of 255. The "ip multicast ttl-threshold" interface configuration command can be used to filter Mcast traffic, the TTL value controls whether Mcast packets are forwarded OUT of an interface. ONLY Mcast packets with a TTL greater than the interface TTL threshold value are forwarded on the interface, by default, this value is set to zero, which means all Mcast packets are forwarded.

Before the "ip multicast ttl-threshold" command is configured, the "debug ip icmp" is configured on R1 and a ping is generated from R6, as follows:

On R1

```
R1#Debug ip icmp
```

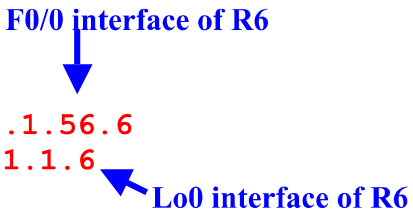
On R6

```
R6#Ping 239.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.1.1, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 124 ms
```

```
Reply to request 0 from 10.1.12.1, 140 ms
```

On R1

The following is the output of the debug command:




```
ICMP: echo reply sent, src 10.1.12.1, dst 10.1.56.6
ICMP: echo reply sent, src 10.1.12.1, dst 1.1.1.6
```

To Configure the task:

On R2

```
R2(config)#int f0/0
R2(config-subif)#ip multicast ttl-threshold 252
```

To verify and test the configuration:

On R6

```
R6#Ping 239.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.1.1, timeout is 2 seconds:
Reply to request 0 from 10.1.12.1, 113 ms
```

On R1

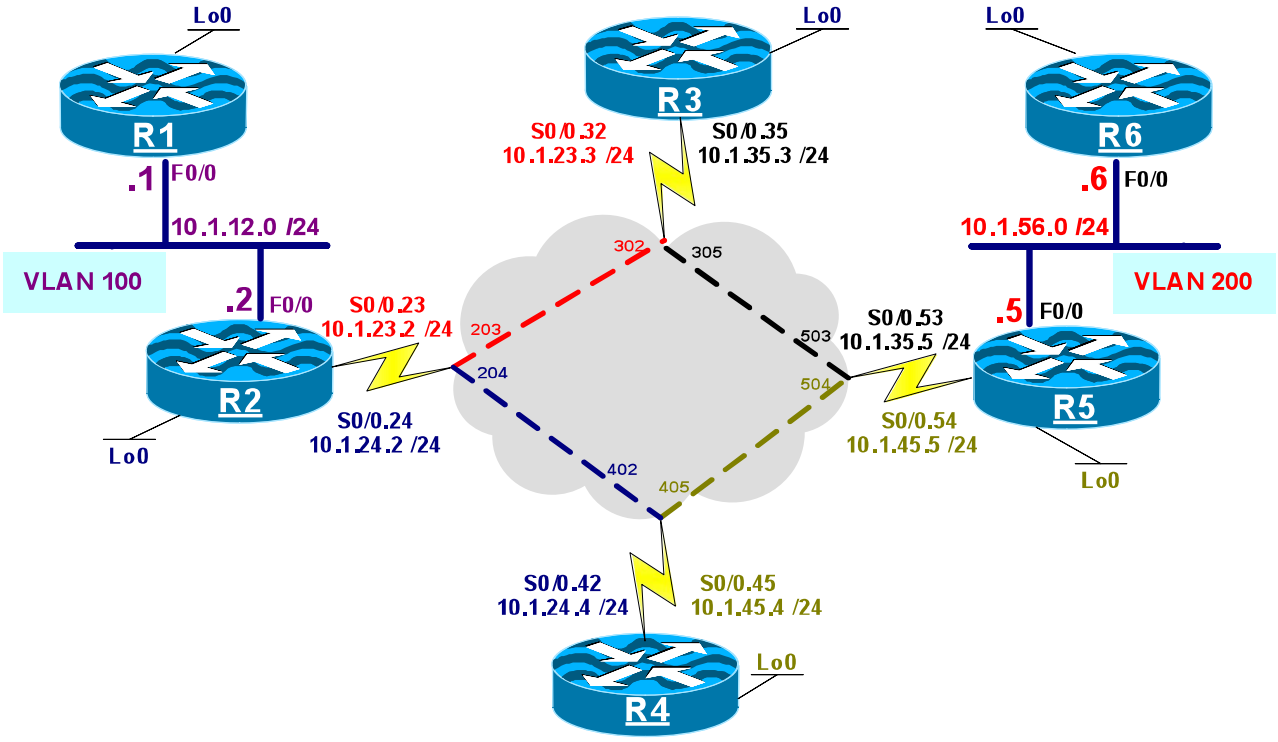
```
ICMP: echo reply sent, src 10.1.12.1, dst 10.1.56.6
```

Note R1 did not receive an ICMP packet from Lo0, therefore, it ONLY responded to the packet that was generated by the F0/0 interface of R6.

Task 7

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 7 – Configuring MSDP



Lab Setup

- Configure the F0/0 interface of R1 and R2 in VLAN 100, and F0/0 interface of R4 and R5 in VLAN 200.
- All frame-Relay connections must be configured in a point-to-point manner.
- Use the IP addressing chart below for IP addressing scheme

IP addressing Chart:

Router	Interface / IP addressing
R1	F0/0 = 10.1.12.1 /24 Lo0 = 1.1.1.1 /32
R2	**F0/0 = 10.1.12.2 /24** **S0/0.23 = 10.1.23.2/24** **S0/0.24 = 10.1.24.2/24** **Lo0 = 1.1.1.2 /32**
R3	S0/0.32 = 10.1.23.3/24 S0/0.35 = 10.1.35.3/24 Lo0= 1.1.1.3 /32
R4	**S0/0.42 = 10.1.24.4/24** **S0/0.45 = 10.1.45.4 /24** **Lo0 = 1.1.1.4 /32**
R5	S0/0.53 = 10.1.35.5/24 S0/0.54 = 10.1.45.5/24 F0/0 = 10.1.56.5/24 Lo0 = 1.1.1.5 /32
R6	**F0/0 = 10.1.56.6/24** **Lo0 = 1.1.1.6/32**

Task 1

Configure OSPF area 0 on all interfaces.

On All Routers

```
Rx(config)#router ospf 1
Rx(config-router)#netw 0.0.0.0 0.0.0.0 are 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf

     1.0.0.0/32 is subnetted, 6 subnets
O       1.1.1.3 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.2 [110/2] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.5 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.4 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.6 [110/131] via 10.1.12.2, 00:00:19, FastEthernet0/0
     10.0.0.0/24 is subnetted, 6 subnets
```

```
O       10.1.24.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.45.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.35.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.56.0 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
```

Task 2

Configure PIM on the interfaces according to the following table:

Router	Interface / PIM
R1	F0/0 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R2	**F0/0 = PIM Sparse-mode** **S0/0.23 = PIM Sparse-mode** **S0/0.24 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**
R3	S0/0.32 = PIM Sparse-mode S0/0.35 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R4	**S0/0.42 = PIM Sparse-mode** **S0/0.45 = PIM Sparse-mode** **Lo0= PIM Sparse-mode**
R5	F0/0 = PIM Sparse-mode S0/0.53 = PIM Sparse-mode S0/0.54 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R6	**F0/0 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**

On R1

```
R1(config)#ip multicast-routing

R1(config)#int lo 0
R1(config-if)#ip pim sparse-mode

R1(config-if)#int f0/0
R1(config-if)#ip pim sparse-mode
```

On R2

```
R2(config)#ip multicast-routing

R2(config)#int lo0
R2(config-if)#ip pim sparse-mode

R2(config-if)#int f0/0
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.23
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.24
R2(config-subif)#ip pim sparse-mode
```

On R3

```
R3(config)#ip multicast-routing

R3(config)#int lo0
R3(config-if)#ip pim sparse-mode

R3(config-if)#int s0/0.32
R3(config-subif)#ip pim sparse-mode

R3(config-subif)#int s0/0.35
R3(config-subif)#ip pim sparse-mode
```

On R4

```
R4(config)#ip multicast-routing

R4(config)#int lo0
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.42
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.45
R4(config-if)#ip pim sparse-mode
```

On R5

```
R5(config)#ip multicast-routing
```

```
R5(config)#int lo0
R5(config-if)#ip pim sparse-mode

R5(config-if)#int f0/0
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.53
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.54
R5(config-subif)#ip pim sparse-mode
```

On R6

```
R6(config)#ip multicast-routing

R6(config)#int lo0
R6(config-if)#ip pim sparse-mode

R6(config-if)#int f0/0
R6(config-if)#ip pim sparse-mode
```

To verify the configuration:

On R1

```
R1#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.2         FastEthernet0/0          00:02:01/00:01:41 v2    1 / DR S
```

On R2

```
R2#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.1         FastEthernet0/0          00:02:01/00:01:41 v2    1 / S
10.1.23.3         Serial0/0.23             00:01:13/00:01:30 v2    1 / S
10.1.24.4         Serial0/0.24             00:00:58/00:01:15 v2    1 / S
```

On R3

```
R3#Show ip pim neighbor | b interface
```

```
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.23.2         Serial0/0.32             00:01:13/00:01:31 v2    1 / S
10.1.35.5         Serial0/0.35             00:00:41/00:01:32 v2    1 / S
```

On R4

```
R4#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.24.2         Serial0/0.42             00:00:58/00:01:15 v2    1 / S
10.1.45.5         Serial0/0.45             00:00:36/00:01:37 v2    1 / S
```

On R5

```
R5#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.6         FastEthernet0/0          00:00:33/00:01:40 v2    1 / DR S
10.1.35.3         Serial0/0.53             00:05:42/00:01:27 v2    1 / S
10.1.45.4         Serial0/0.54             00:05:37/00:01:30 v2    1 / S
```

On R6

```
R6#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.5         FastEthernet0/0          00:01:52/00:01:20 v2    1 / S
```

Task 3

Configure static RP mappings for R1, R3 and R6 according to the following table:

Routers	RP	Interface
R1, R2	R2	Lo0
R3	R3	Lo0
R5,R6	R5	Lo0

On R1 and R2

```
Rx(config)#ip pim rp-address 1.1.1.2
```

To verify the configuration:

On R1

```
R1#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 1.1.1.2 (?)
```

On R2

```
R2#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 1.1.1.2 (?)
```

On R3

```
R3(config)#ip pim rp-address 1.1.1.3
```

To verify the configuration:

On R3

```
R3#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 1.1.1.3 (?)
```

On R5 and R6

```
Rx(config)#ip pim rp-address 1.1.1.5
```

To verify the configuration:

On R5

```
R5#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 1.1.1.5 (?)
```

On R6

```
R6#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 1.1.1.5 (?)
```

Task 4

Configure the Loopback 0 interface of R1, R3 and R6 to join 239.1.3.6 Mcast group.

On R1

```
R1(config)#int lo0
R1(config-if)#ip igmp join-group 239.1.3.6
```

To verify the configuration:

On R1

```
R1#Show ip igmp groups | exc 224.0.1.40

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
239.1.3.6        Loopback0                00:00:59  00:02:51  1.1.1.1
```

On R3

```
R3(config)#int lo0
R3(config-if)#ip igmp join-group 239.1.3.6
```

To verify the configuration:

On R3

```
R3#Show ip igmp groups | exc 224.0.1.40

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
239.1.3.6        Loopback0                00:02:11  00:02:51  1.1.1.3
```

On R6

```
R6(config)#int lo0
R6(config-if)#ip igmp join-group 239.1.3.6
```

To verify the configuration:

On R6

```
R6#Show ip igmp groups | exc 224.0.1.40

IGMP Connected Group Membership
Group Address    Interface                Uptime    Expires   Last Reporter
239.1.3.6        Loopback0                00:03:46  00:02:40  1.1.1.6
```

Task 5

Configure the network such that every router except R4 has the ability to ping group 239.1.3.6 and receive replies from R1, R3 and R6.

On R1

```
R1#Ping 239.1.3.6

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.3.6, timeout is 2 seconds:

Reply to request 0 from 1.1.1.1, 4 ms
```

R1 received a reply from itself ONLY; it did not receive a reply from R3 or R6.

The following reveals the mroute table of R2, this is important, since R2 is R1's RP:

On R2

```
R2#Show ip mroute | s 239.1.3.6

(*, 239.1.3.6), 00:05:59/00:03:25, RP 1.1.1.2, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:05:59/00:03:25
(1.1.1.1, 239.1.3.6), 00:00:52/00:02:07, flags: P
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.12.1
  Outgoing interface list: Null
(10.1.12.1, 239.1.3.6), 00:00:52/00:02:09, flags: PT
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.12.1
  Outgoing interface list: Null
```

As you can see the only outgoing interface for that group is R2's F0/0, which is the interface facing R1. Since R2 is NOT the RP for R3 and R6, the join (*, 239.1.3.6) never reached these routers.

The solution is MSDP.
MSDP is a mechanism to connect multiple PIM-SM domains. MSDP enables multiple sources for a given group to be known to all RPs in the same or different domains.
MSDP peering is configured between RPs; RPS run MSDP over TCP port 639 in order to synchronize their knowledge of sources amongst themselves.

How does MSDP work?
When the source is initiated, the first hop router encapsulates the Mcast data in register messages and Unicasts the flow to the RP; RP de-encapsulates the register messages and forwards it down toward the last hop router. If MSDP is configured on the RP, the packet is also re-encapsulated in Source Active (SA) messages, which are immediately forwarded to all MSDP peers.

The SA messages identify the source, the group address that the source is sending to and the originator-id, the originator-id is an optional and it is included only if it's configured.

To configure an MSDP peer:

On R2

```
R2(config)#ip msdp peer 1.1.1.5 connect-source lo0
```

On R5

```
R5(config)#ip msdp peer 1.1.1.2 connect-source lo0
```

To verify the configuration:

On R2

```
R2#Sh ip msdp peer

MSDP Peer 1.1.1.5 (?), AS ?
  Connection status:
    State: Up, Resets: 0, Connection source: Loopback0 (1.1.1.2)
    Uptime(Downtime): 00:00:12, Messages sent/received: 1/0
    Output messages discarded: 0
    Connection and counters cleared 00:01:12 ago
  SA Filtering:
    Input (S,G) filter: none, route-map: none
    Input RP filter: none, route-map: none
    Output (S,G) filter: none, route-map: none
    Output RP filter: none, route-map: none
  SA-Requests:
    Input filter: none
  Peer ttl threshold: 0
  SAs learned from this peer: 0
  Input queue size: 0, Output queue size: 0
  MD5 signature protection on MSDP TCP connection: not enabled
```

(The rest of the output is omitted)

To verify the configuration:

On R1

```
R1#Ping 239.1.3.6

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.3.6, timeout is 2 seconds:

Reply to request 0 from 1.1.1.1, 1 ms
Reply to request 0 from 10.1.56.6, 136 ms
Reply to request 0 from 10.1.56.6, 112 ms
```

The output of the above ping command reveals that R1 gained reachability to R6 but NOT to R3. The following ping tests reachability of R6:

On R6

```
R6#Ping 239.1.3.6

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.3.6, timeout is 2 seconds:

Reply to request 0 from 1.1.1.6, 4 ms
```

```
Reply to request 0 from 10.1.12.1, 132 ms
Reply to request 0 from 10.1.12.1, 116 ms
```

R6 has reachability only to itself and R1, but not to R3. In order to provide reachability to R3, an MSDP session is configured between R3 and R5, as follows:

On R5

```
R5(config)#ip msdp peer 1.1.1.3 connect-source lo0
```

On R3

```
R3(config)#ip msdp peer 1.1.1.5 connect-source lo0
```

You should see the following console message stating that the peer session to R5 (1.1.1.5) is UP.

%MSDP-5-PEER_UPDOWN: Session to peer 1.1.1.5 going up

To verify the configuration:

On R3

```
R3#Show ip msdp count

SA State per Peer Counters, <Peer>: <# SA learned>
    1.1.1.5: 4

SA State per ASN Counters, <asn>: <# sources>/<# groups>
    Total entries: 4
    ?: 4/1
```

Note R3 has an MSDP peer with R5 but it has NOT received any SA messages from R5, or maybe it had it but they timed out, therefore, it has NO SA messages in its cache. To generate some SA messages, a ping is generated from R1 for group 239.1.3.6.

Note if you have SA messages cached, you should clear them using the "Clear ip msdp sa-cache" before proceeding. You may have to ping three or four times to receive the following result.

On R1

```
R1#Ping 239.1.3.6

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.3.6, timeout is 2 seconds:
```

```
Reply to request 0 from 1.1.1.1, 4 ms
Reply to request 0 from 10.1.56.6, 216 ms
Reply to request 0 from 10.1.23.3, 204 ms
Reply to request 0 from 10.1.56.6, 180 ms
Reply to request 0 from 10.1.23.3, 168 ms
```

On R3

```
R3#Show ip msdp count

SA State per Peer Counters, <Peer>: <# SA learned>
    1.1.1.5: 2

SA State per ASN Counters, <asn>: <# sources>/<# groups>
    Total entries: 2
    ?: 2/1
```

Note since these routers are NOT configured in a BGP AS, the ASN is displayed as unknown (?), but the "2/1" reveals that there are two sources but a single group.

```
R3#Show ip msdp peer 1.1.1.5

MSDP Peer 1.1.1.5 (?), AS ?
  Connection status:
    State: Up, Resets: 0, Connection source: Loopback0 (1.1.1.3)
    Uptime(Downtime): 00:47:30, Messages sent/received: 52/74
```

(The rest of the output is omitted)

Note the output of the above command reveals the following information:

- **MSDP peer is 1.1.1.5, since name resolution failed the "?" is displayed, and since these routers are NOT in a BGP AS, the AS number is also displayed as unknown "?".**
- **To replace the "?" with the actual name of the peer, the following is configured:**

On R3

```
R3(config)#ip host R5 1.1.1.5
```

To verify the configuration:

On R3

```
R3#Show ip msdp peer 1.1.1.5 | inc msdp peer

MSDP Peer 1.1.1.5 (R5), AS ?
```

The following output displays the (S,G) state leaned from R5 (Its MSDP peer).

```
R3#Show ip msdp sa-cache

MSDP Source-Active Cache - 4 entries
(1.1.1.1, 239.1.3.6), RP 1.1.1.2, AS ?,00:05:30/00:05:09, Peer 1.1.1.5
(1.1.1.6, 239.1.3.6), RP 1.1.1.5, AS ?,00:05:31/00:02:17, Peer 1.1.1.5
(10.1.12.1, 239.1.3.6), RP 1.1.1.2, AS ?,00:05:30/00:05:09, Peer 1.1.1.5
(10.1.56.6, 239.1.3.6), RP 1.1.1.5, AS ?,00:05:31/00:02:17, Peer 1.1.1.5
```

To see a summary:

```
R3#Show ip msdp summary

MSDP Peer Status Summary
Peer Address     AS    State    Uptime/  Reset SA    Peer Name
                                Downtime Count Count
1.1.1.5          ?     Up       00:04:57 0     4     ?
```

To test reachability:

```
R3#Deb ip icmp

R3#Ping 239.1.3.6

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.3.6, timeout is 2 seconds:

ICMP: echo reply sent, src 1.1.1.3, dst 1.1.1.3
ICMP: echo reply rcvd, src 1.1.1.3, dst 1.1.1.3
ICMP: echo reply rcvd, src 10.1.12.1, dst 1.1.1.3
ICMP: echo reply rcvd, src 10.1.56.6, dst 1.1.1.3
ICMP: echo reply rcvd, src 10.1.12.1, dst 10.1.23.3
ICMP: echo reply rcvd, src 10.1.56.6, dst 10.1.35.3
Reply to request 0 from 1.1.1.3, 1 ms
Reply to request 0 from 10.1.56.6, 108 ms
Reply to request 0 from 10.1.12.1, 92 ms
Reply to request 0 from 10.1.56.6, 64 ms
Reply to request 0 from 10.1.12.1, 52 ms
```

Note R3 has reachability to R1, R3 and R6.

On R6

```
R6#Ping 239.1.3.6
```

```
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.3.6, timeout is 2 seconds:

Reply to request 0 from 1.1.1.6, 4 ms
Reply to request 0 from 10.1.12.1, 140 ms
Reply to request 0 from 10.1.12.1, 108 ms
Reply to request 0 from 10.1.35.3, 96 ms
Reply to request 0 from 10.1.35.3, 68 ms
```

Note R6 has reachability to R1, R3 and R6.

On R1

```
R1#Ping 239.1.3.6

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.3.6, timeout is 2 seconds:

Reply to request 0 from 1.1.1.1, 1 ms
Reply to request 0 from 10.1.56.6, 200 ms
Reply to request 0 from 10.1.23.3, 188 ms
Reply to request 0 from 10.1.56.6, 164 ms
Reply to request 0 from 10.1.23.3, 148 ms
```

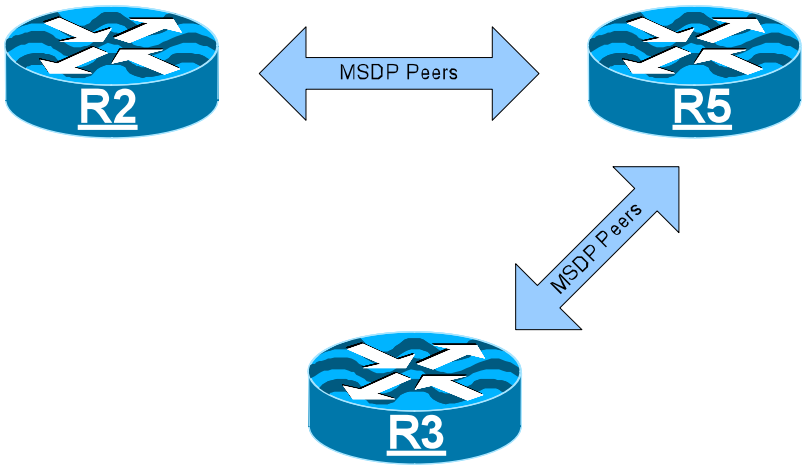
Note R1 has reachability to R1, R3 and R6.

Task 6

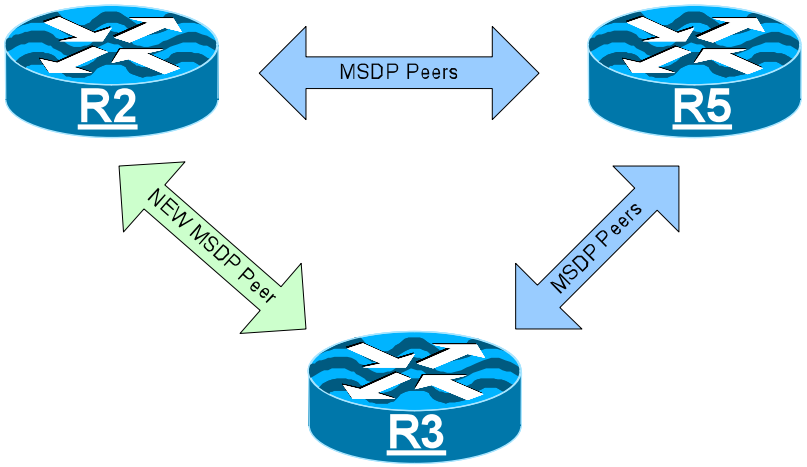
Make sure the MSDP-SA distribution paths are resilient.

MSDP messages are flooded across MSDP peers; that is why R3 has SA messages originated from R2 even though there is no direct MSDP peering between R3 and R2, R3 receives these messages from R5. What happens if R5 goes down? Obviously, R2's SA messages will not reach R3; therefore, based on the existing configuration the resiliency is NOT there.

The following shows the existing peering:



To provide full resiliency, a full mesh peering must be established, as follows:



Therefore, another MSDP peer session must be configured between R2 and R3, this will provide a full mesh peering which in turn provides redundancy and resiliency:

On R2

```
R2(config)#ip msdp peer 1.1.1.3 connect-source lo0
```

On R3

```
R3(config)#ip msdp peer 1.1.1.2 connect-source lo0
```

To verify the configuration:

On R3

Wait for the following console message before entering the following show command:

%MSDP-5-PEER_UPDOWN: Session to peer 1.1.1.2 going up

```
R3#Show ip msdp summary

MSDP Peer Status Summary
Peer Address     AS    State    Uptime/  Reset SA    Peer Name
                                Downtime Count Count
1.1.1.5          ?     Up       00:15:03 0     6     R5
1.1.1.2          ?     Up       00:00:07 0     0     ?
```

Task 7

Configure R3 to limit the number of SA's it receives from R5 to 40.

The "ip msdp sa-limit" command can be used to limit the number of SAs received from a given peer.

On R3

```
R3(config)#ip msdp sa-limit 1.1.1.5 40
```

To verify the configuration:

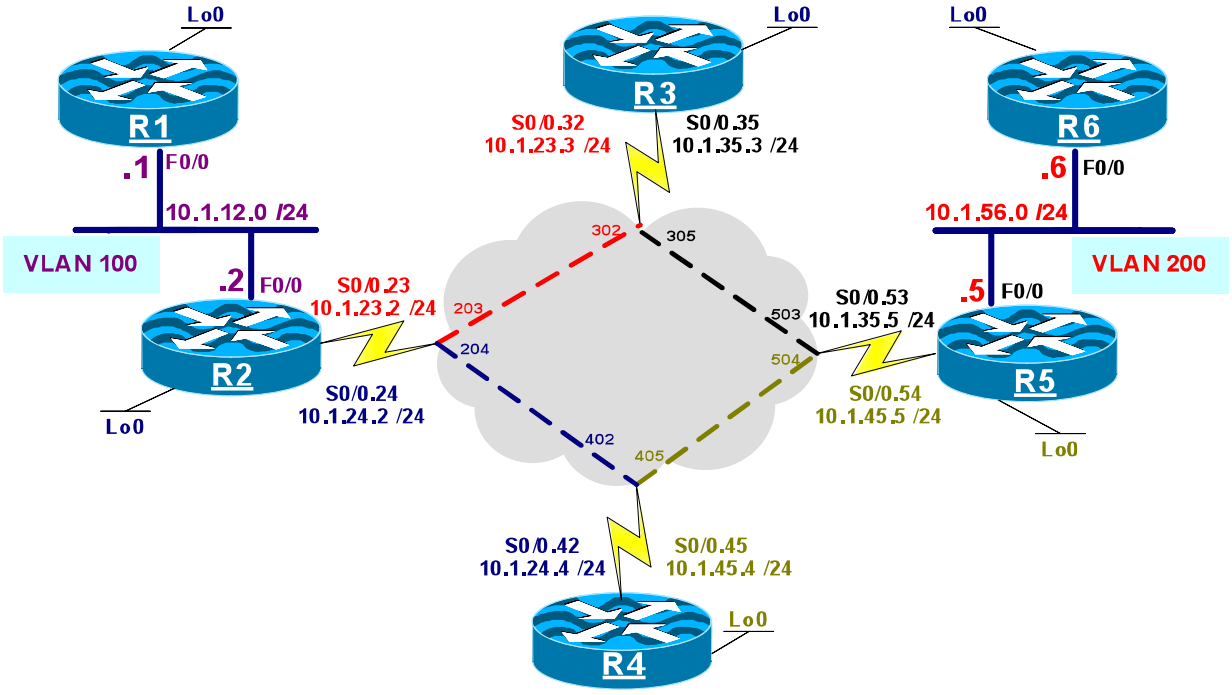
On R3

```
R3#Sh ip msdp peer 1.1.1.5 | inc sas limit

SAs learned from this peer: 0, SAs limit: 40
```

Task 8

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 8 – Anycast IP



These routers are running Version 12.4 (3b) IOS.

Lab Setup

- Configure the F0/0 interface of R1 and R2 in VLAN 100, and F0/0 interface of R4 and R5 in VLAN 200.
- All frame-Relay connections must be configured in a point-to-point manner.
- Use the IP addressing chart below for IP addressing scheme

IP addressing Chart:

Router	Interface / IP addressing
R1	F0/0 = 10.1.12.1 /24 Lo0 = 1.1.1.1 /32
R2	**F0/0 = 10.1.12.2 /24** **S0/0.23 = 10.1.23.2/24** **S0/0.24 = 10.1.24.2/24** **Lo0 = 1.1.1.2 /32**
R3	S0/0.32 = 10.1.23.3/24 S0/0.35 = 10.1.35.3/24 Lo0= 1.1.1.3 /32
R4	**S0/0.42 = 10.1.24.4/24** **S0/0.45 = 10.1.45.4 /24** **Lo0 = 1.1.1.4 /32**
R5	S0/0.53 = 10.1.35.5/24 S0/0.54 = 10.1.45.5/24 F0/0 = 10.1.56.5/24 Lo0 = 1.1.1.5 /32
R6	**F0/0 = 10.1.56.6/24** **Lo0 = 1.1.1.6/32**

Task 1

Configure OSPF area 0 on all interfaces.

On All Routers

```
Rx(config)#router ospf 1
Rx(config-router)#netw 0.0.0.0 0.0.0.0 are 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf

     1.0.0.0/32 is subnetted, 6 subnets
O       1.1.1.3 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.2 [110/2] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.5 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.4 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.6 [110/131] via 10.1.12.2, 00:00:19, FastEthernet0/0
```

```
     10.0.0.0/24 is subnetted, 6 subnets
O       10.1.24.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.45.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.35.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.56.0 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
```

Task 2

Configure PIM on the interfaces according to the following table:

Router	Interface / PIM
R1	F0/0 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R2	**F0/0 = PIM Sparse-mode** **S0/0.23 = PIM Sparse-mode** **S0/0.24 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**
R3	S0/0.32 = PIM Sparse-mode S0/0.35 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R4	**S0/0.42 = PIM Sparse-mode** **S0/0.45 = PIM Sparse-mode** **Lo0= PIM Sparse-mode**
R5	F0/0 = PIM Sparse-mode S0/0.53 = PIM Sparse-mode S0/0.54 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R6	**F0/0 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**

On R1

```
R1(config)#ip multicast-routing

R1(config)#int lo 0
R1(config-if)#ip pim sparse-mode

R1(config-if)#int f0/0

R1(config-if)#ip pim sparse-mode
```

On R2

```
R2(config)#ip multicast-routing

R2(config)#int lo0
R2(config-if)#ip pim sparse-mode

R2(config-if)#int f0/0
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.23
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.24
R2(config-subif)#ip pim sparse-mode
```

On R3

```
R3(config)#ip multicast-routing

R3(config)#int lo0
R3(config-if)#ip pim sparse-mode

R3(config-if)#int s0/0.32
R3(config-subif)#ip pim sparse-mode

R3(config-subif)#int s0/0.35
R3(config-subif)#ip pim sparse-mode
```

On R4

```
R4(config)#ip multicast-routing

R4(config)#int lo0
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.42
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.45
R4(config-if)#ip pim sparse-mode
```

On R5

```
R5(config)#ip multicast-routing
```

```
R5(config)#int lo0
R5(config-if)#ip pim sparse-mode

R5(config-if)#int f0/0
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.53
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.54
R5(config-subif)#ip pim sparse-mode
```

On R6

```
R6(config)#ip multicast-routing

R6(config)#int lo0
R6(config-if)#ip pim sparse-mode

R6(config-if)#int f0/0
R6(config-if)#ip pim sparse-mode
```

To verify the configuration:

On R1

```
R1#Show ip pim neigh | b inter

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.2         FastEthernet0/0          01:00:02/00:01:17 v2    1 / DR S
```

On R2

```
R2#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.1         FastEthernet0/0          01:01:24/00:01:20 v2    1 / S
10.1.23.3         Serial0/0.23             01:00:35/00:01:41 v2    1 / S
10.1.24.4         Serial0/0.24             01:00:21/00:01:31 v2    1 / S
```

On R3

```
R3#Show ip pim neighbor | b interface
```

```
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.23.2         Serial0/0.32             01:00:35/00:01:42 v2    1 / S
10.1.35.5         Serial0/0.35             01:00:04/00:01:40 v2    1 / S
```

On R4

```
R4#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.24.2         Serial0/0.42             01:00:21/00:01:28 v2    1 / S
10.1.45.5         Serial0/0.45             00:59:59/00:01:19 v2    1 / S
```

On R5

```
R5#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.6         FastEthernet0/0          00:54:54/00:01:32 v2    1 / DR S
10.1.35.3         Serial0/0.53             01:00:04/00:01:43 v2    1 / S
10.1.45.4         Serial0/0.54             00:59:59/00:01:16 v2    1 / S
```

On R6

```
R6#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.5         FastEthernet0/0          00:54:54/00:01:21 v2    1 / S
```

Task 3

Configure the following Loopback interface on R2 and R5 and advertise it in OSPF area 0:
Lo1: 200.1.1.1 /32

On R2

```
R2(config)#int lo1
R2(config-if)#ip addr 200.1.1.1 255.255.255.255
```

On R5

```
R5(config)#interface Lo1
R5(config-if)#ip address 200.1.1.1 255.255.255.255
```

Note because the way OSPF's network was configured, all directly connected interfaces will be advertised in OSPF area 0; therefore, there is no need to configure a Network command in OSPF for this prefix.

Task 4

Configure a static RP mapping pointing to 200.1.1.1 /32 on all routers.

On R1

```
R1(config)#ip pim rp-address 200.1.1.1
```

On R2

```
R2(config)#ip pim rp-address 200.1.1.1
```

On R3

```
R3(config)#ip pim rp-address 200.1.1.1
```

On R4

```
R4(config)#ip pim rp-address 200.1.1.1
```

On R5

```
R5(config)#ip pim rp-address 200.1.1.1
```

On R6

```
R6(config)#ip pim rp-address 200.1.1.1
```

To verify the configuration:

On R6

```
R6#Sh ip pim rp mapping

PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 200.1.1.1 (?)
```

On R1

```
R1#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s): 224.0.0.0/4, Static
    RP: 200.1.1.1 (?)
```

Task 5

Configure R1's Lo0 to join the 239.1.1.1. Configure the network to allow R6 to ping this Mcast group address.

On R1

```
R1(config)#int lo0
R1(config-if)#ip igmp join-group 239.1.1.1
```

On R6

```
R6#Ping 239.1.1.1
```

Note the ping is NOT successful.

```
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.1.1, timeout is 2 seconds:
.
```

Remember that we have set two RP's with the same IP address, since R5 is closer to R6 than R2, R6 will use R5 as its RP whereas, R1 will use R2 as its RP.

To verify the configuration:

On R2

```
R2#Show ip mroute | s 239.1.1.1

(*, 239.1.1.1), 00:16:24/00:02:49, RP 200.1.1.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:16:24/00:02:49
```

MSDP should be used to synchronize the knowledge of the two RPs, as follows:

On R2

```
R2(config)#ip msdp peer 1.1.1.5 connect-source lo0
```

On R5

```
R5(config)#ip msdp peer 1.1.1.2 connect-source lo0
```

To test and verify the configuration

On R5

```
R5#Show ip msdp peer | inc peer|state

MSDP Peer 1.1.1.2 (?), AS ?
    State: Up, Resets: 0, Connection source: Loopback0 (1.1.1.5)
  Peer ttl threshold: 0
```

The State between the routers should be UP, if they are in "Listen", wait few seconds and try the show command again:

On R6

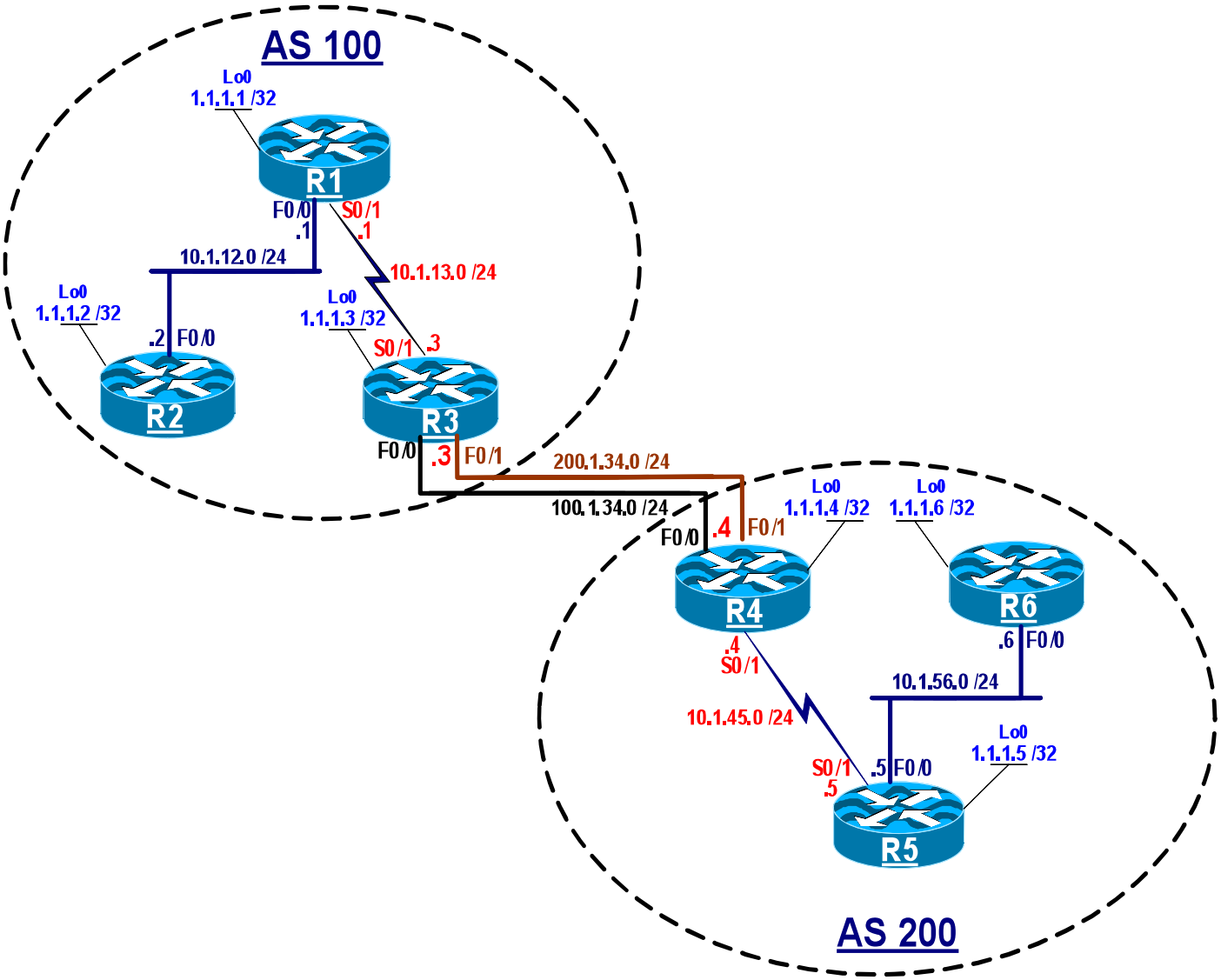
```
R6#Ping 239.1.1.1

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 239.1.1.1, timeout is 2 seconds:

Reply to request 0 from 10.1.12.1, 116 ms
Reply to request 0 from 10.1.12.1, 128 ms
```

Note Pings are successful; the pings will be successful even if R5's Lo1 interface down.

Task 6

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 9 – MSDP/MP-BGP



Surinderjeet Singhh

Lab Setup:

- Configure OSPF Area 0 for all the interfaces in AS 100.
- Configure OSPF Area 0 for all the interfaces in AS 200.
- DO NOT run OSPF on the F0/0 and F0/1 interface connecting R3 to R4
- Use IP addressing in the table above

IP Address:

Router	Interface	IP address
R1	Lo0 F0/0 S0/1	1.1.1.1 /32 10.1.12.1 /24 10.1.13.1 /24
R2	Lo0 F0/0 F0/1	1.1.1.2 /32 10.1.12.2 /24 192.168.1.2 /24
R3	Lo0 S0/1 F0/0 F0/1	1.1.1.3/ 32 10.1.13.3 /24 100.1.34.3 /24 200.1.34.3 /24
R4	Lo0 S0/1 F0/0 F0/1	1.1.1.4 /32 10.1.45.4 /24 100.1.34.4 /24 200.1.34.4 /24
R5	Lo0 S0/1 F0/0	1.1.1.5 /32 10.1.45.5 /24 10.1.56.5/ 24
R6	Lo0 F0/0	1.1.1.6 /32 10.1.56.6/ 24

Task 1

Configure OSPF based on the following policy:

- Configure OSPF Area 0 for all the interfaces in AS 100.
- Configure OSPF Area 0 for all the interfaces in AS 200.
- DO NOT run OSPF on the F0/0 and F0/1 interface connecting R3 to R4

On R1

```
R1(config)#router ospf 1
R1(config-router)#netw 0.0.0.0 0.0.0.0 area 0
```

On R2

```
R2(config)#router ospf 1
R2(config-router)#netw 0.0.0.0 0.0.0.0 area 0
```

On R3

```
R3(config)#router ospf 1
R3(config-router)#netw 1.1.1.3 0.0.0.0 area 0
R3(config-router)#netw 10.1.13.3 0.0.0.0 area 0
```

On R4

```
R4(config)#router ospf 1
R4(config-router)#netw 1.1.1.4 0.0.0.0 area 0
R4(config-router)#netw 10.1.45.4 0.0.0.0 area 0
```

On R5

```
R5(config)#router ospf 1
R5(config-router)#netw 0.0.0.0 0.0.0.0 area 0
```

On R6

```
R6(config)#router ospf 1
R6(config-router)#netw 0.0.0.0 0.0.0.0 area 0
```

On All Routers

```
Rx(config)#int lo0
Rx(config-if)#ip ospf netw point-to-point
```

To verify the configuration:

On R1

```
R1#Show ip route ospf | inc O

O       1.1.1.3 [110/65] via 10.1.13.3, 00:02:36, Serial0/1
O       1.1.1.2 [110/2] via 10.1.12.2, 00:02:26, FastEthernet0/0
O    192.168.1.0/24 [110/2] via 10.1.12.2, 00:02:26, FastEthernet0/0
```

On R6

```
R6#Show ip route ospf | inc O

O       1.1.1.5 [110/2] via 10.1.56.5, 00:00:54, FastEthernet0/0
O       1.1.1.4 [110/66] via 10.1.56.5, 00:00:54, FastEthernet0/0
O       10.1.45.0 [110/65] via 10.1.56.5, 00:00:54, FastEthernet0/0
```

Task 2

Configure Multicast on the interfaces noted in the table below:

Router	Interface / IP addressing
R1	F0/0 = **Pim sparse mode** S0/1 = **Pim sparse mode** Lo0 = **Pim sparse mode**
R2	F0/0 = **Pim sparse mode** Lo0 = **Pim sparse mode**
R3	S0/1 = **Pim sparse mode** F0/0 = **Pim sparse mode** F0/1 = **Pim sparse mode** Lo0 = **Pim sparse mode**
R4	S0/1 = **Pim sparse mode** F0/0 = **Pim sparse mode** F0/1 = **Pim sparse mode** Lo0 = **Pim sparse mode**
R5	S0/1 = **Pim sparse mode** F0/0 = **Pim sparse mode** Lo0 = **Pim sparse mode**
R6	F0/0 = **Pim sparse mode** Lo0 = **Pim sparse mode**

On R1

```
R1(config)#ip multicast-routing

R1(config)#interface lo0
R1(config-if)#ip pim sparse-mode

R1(config)#interface f0/0
R1(config-if)#ip pim sparse-mode

R1(config)#interface s0/1
```

```
R1(config-if)#ip pim sparse-mode
```

On R2

```
R2(config)#ip multicast-routing

R2(config)#interface f0/0
R2(config-if)#ip pim sparse-mode

R2(config)#interface f0/1
R2(config-if)#ip pim sparse-mode

R2(config)#interface lo0
R2(config-if)#ip pim sparse-mode
```

On R3

```
R3(config)#ip multicast-routing

R3(config)#interface f0/0
R3(config-if)#ip pim sparse-mode

R3(config)#interface f0/1
R3(config-if)#ip pim sparse-mode

R3(config)#interface lo0
R3(config-if)#ip pim sparse-mode

R3(config)#interface s0/1
R3(config-if)#ip pim sparse-mode
```

On R4

```
R4(config)#ip multicast-routing

R4(config)#interface s0/1
R4(config-if)#ip pim sparse-mode

R4(config)#interface f0/0
R4(config-if)#ip pim sparse-mode

R4(config)#interface f0/1
R4(config-if)#ip pim sparse-mode

R4(config)#interface lo0
```

```
R4(config-if)#ip pim sparse-mode
```

On R5

```
R5(config)#ip multicast-routing

R5(config)#interface s0/1
R5(config-if)#ip pim sparse-mode

R5(config)#interface f0/0
R5(config-if)#ip pim sparse-mode

R5(config)#interface lo0
R5(config-if)#ip pim sparse-mode
```

On R6

```
R6(config)#ip multicast-routing

R6(config)#interface f0/0
R6(config-if)#ip pim sparse-mode

R6(config)#interface lo0
R6(config-if)#ip pim sparse-mode
```

Verify the Configuration:

On R6

```
R6#Show ip pim neighbor | b neighbor

PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.5         FastEthernet0/0          00:02:32/00:01:40 v2    1 / S
```

On R5

```
R5#Show ip pim neighbor | b neighbor

PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
```

```
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.45.4         Serial0/1                00:02:58/00:01:43 v2    1 / S
10.1.56.6         FastEthernet0/0          00:02:32/00:01:39 v2    1 / DR S
```

On R4

```
R4#Show ip pim neighbor | b neighbor

PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
100.1.34.3        FastEthernet0/0          00:03:24/00:01:19 v2    1 / S
200.1.34.3        FastEthernet0/1          00:03:21/00:01:20 v2    1 / S
10.1.45.5         Serial0/1                00:02:58/00:01:42 v2    1 / S
```

On R3

```
R3#Show ip pim neighbor | b neighbor
PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.13.1         Serial0/1                00:04:06/00:01:33 v2    1 / S
100.1.34.4        FastEthernet0/0          00:03:24/00:01:17 v2    1 / DR S
200.1.34.4        FastEthernet0/1          00:03:21/00:01:19 v2    1 / DR S
```

On R2

```
R2#Show ip pim neighbor | b neighbor

PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.1         FastEthernet0/0          00:04:56/00:01:43 v2    1 / S
```

On R1

```
R1#Show ip pim neighbor | b neighbor

PIM Neighbor Table
Mode: B - Bidir Capable, DR - Designated Router, N - Default DR Priority,
      S - State Refresh Capable
```

```
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.2         FastEthernet0/0          00:04:56/00:01:43 v2    1 / DR S
10.1.13.3         Serial0/1                00:04:06/00:01:34 v2    1 / S
```

Task 3

Configure an ipv4-unicast MP-BGP session between R3 and R4 using their F0/0 interfaces. Disable ipv4-unicast routing in BGP. Advertise the RP addresses (The Lo0 interfaces of R1 and R5) in such a way that they are reachable from each domain. DO NOT configure static routing to accomplish this task. R3 and R4 should be configured to inject a default route using OSPF.

On R3

```
R3(config)#router bgp 100
R3(config-router)#No au
R3(config-router)#No bgp default ipv4-unicast
R3(config-router)#neighbor 100.1.34.4 remote-as 200

R3(config-router)#address-family ipv4 unicast
R3(config-router-af)#neighbor 100.1.34.4 activate
R3(config-router-af)#network 1.1.1.1 mask 255.255.255.255
R3(config-router-af)#No au

R3(config)#router ospf 1
R3(config-router)#default-information originate always
```

To verify the configuration:

On R1

```
R1#Sh ip route ospf | inc O

O       1.1.1.3 [110/65] via 10.1.13.3, 00:16:12, Serial0/1
O       1.1.1.2 [110/2] via 10.1.12.2, 00:16:12, FastEthernet0/0
O    192.168.1.0/24 [110/2] via 10.1.12.2, 00:16:12, FastEthernet0/0
O*E2 0.0.0.0/0 [110/1] via 10.1.13.3, 00:03:10, Serial0/1
```

On R4

```
R4(config)#router bgp 200
```

```
R4(config-router)#No au
R4(config-router)#No bgp default ipv4-unicast
R4(config-router)#neighbor 100.1.34.3 remote-as 100

R4(config-router)#address-family ipv4 unicast
R4(config-router-af)#neighbor 100.1.34.3 activate
R4(config-router-af)#network 1.1.1.5 mask 255.255.255.255
R4(config-router-af)#No au
```

The loopbacks of R2 and R5 are advertised under unicast to ensure that the msdp peering will be successful.

```
R4(config)#router ospf 1
R4(config-router)#default-information originate always
```

To verify the configuration:

On R6

```
R6#Sh ip route ospf | inc O

O       1.1.1.5 [110/2] via 10.1.56.5, 00:19:04, FastEthernet0/0
O       1.1.1.4 [110/66] via 10.1.56.5, 00:19:04, FastEthernet0/0
O       10.1.45.0 [110/65] via 10.1.56.5, 00:19:04, FastEthernet0/0
O*E2 0.0.0.0/0 [110/1] via 10.1.56.5, 00:00:47, FastEthernet0/0
```

On R3

```
R3#Show ip bgp summary | b nei

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
100.1.34.4      4   200       5       5        3    0    0 00:01:08        1

R3#Show ip bgp | b network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.1/32       10.1.13.1                66         32768 i
*> 1.1.1.5/32       100.1.34.4               65             0 200 i
```

On R4

```
R4#Show ip bgp summary | b neighbor

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
100.1.34.3      4   100       7       7        3    0    0 00:03:29        1
```

```
R4#Show ip bgp | b network

   Network          Next Hop            Metric LocPrf Weight Path
*> 1.1.1.1/32       100.1.34.3               66             0 100 i
*> 1.1.1.5/32       10.1.45.5                65         32768 i
```

On R1

```
R1#Ping 1.1.1.5 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.5, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
```

Task 4

Establish RP connectivity within each domain with R1 being the RP for AS100 and R5 as the RP for AS200. The RP's must establish multicast peering sessions with each other. No statics are allowed.

On ALL routers in AS 100

```
Rx(config)#ip pim rp-address 1.1.1.1
```

On ALL routers in AS 200

```
Rx(config)#ip pim rp-address 1.1.1.5
```

On R5

```
R5(config)#ip msdp peer 1.1.1.1 connect-source lo0
```

On R1

```
R1(config)#ip msdp peer 1.1.1.5 connect-source lo0
```

You should see the following console message:

```
%MSDP-5-PEER_UPDOWN: Session to peer 1.1.1.5 going up
```

To verify the configuration:

On R1

```
R1#Show ip msdp peer | s connection status

  Connection status:
    State: Up, Resets: 0, Connection source: Loopback0 (1.1.1.1)
    Uptime(Downtime): 00:02:05, Messages sent/received: 2/2
    Output messages discarded: 0
    Connection and counters cleared 00:02:05 ago

R1#Show ip msdp summary

MSDP Peer Status Summary
Peer Address     AS    State    Uptime/  Reset SA    Peer Name
                                Downtime Count Count
1.1.1.5          ?     Up       00:03:24 0     0     ?
```

NOTE: The peering has been established; the questions marks will be described later in this lab.

To test the configuration:

On R2

```
R2(config)#int f0/1
R2(config-if)#ip igmp join-group 224.1.2.3
```

The above command triggers a static join on the F0/1 interface of R2.

On R2

```
R1#Show ip mroute | s 224.1.2.3

(*, 224.1.2.3), 00:00:56/00:03:13, RP 1.1.1.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:00:16/00:03:13
```

The multicast join has triggered a multicast tree to be built from R2 to R1 (the RP).

On R4

```
R4(config)#router bgp 200
```

```
R4(config-router)#address-family ipv4 unicast
R4(config-router-af)#network 1.1.1.6 mask 255.255.255.255
```

The advertisement of the 1.1.1.6 prefix is needed to correctly verify the source of the multicast stream.

On R6

```
R6#Ping 224.1.2.3 source lo0 repeat 1000

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.2.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.6

Reply to request 0 from 10.1.12.2, 100 ms
```

While R6 is pinging the Mcast group, do the following show commands:

On R5

```
R5#Show ip mroute | s 224.1.2.3

(*, 224.1.2.3), 00:05:08/stopped, RP 1.1.1.5, flags: SP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null
(1.1.1.6, 224.1.2.3), 00:00:04/00:02:55, flags: A
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.56.6
  Outgoing interface list:
    Serial0/1, Forward/Sparse, 00:00:04/00:03:25
```

(The rest of the output is omitted)

R5 is the RP for this AS and uses the S,G state to route the packet via the Multicast Tree. The flags express the interest of the router to own the Shortest Path to the Receiver.

On R4

```
R4#Show ip mroute | s 224.1.2.3

(*, 224.1.2.3), 00:07:30/stopped, RP 1.1.1.5, flags: SP
  Incoming interface: Serial0/1, RPF nbr 10.1.45.5
  Outgoing interface list: Null
(1.1.1.6, 224.1.2.3), 00:02:26/00:01:03, flags:
  Incoming interface: Serial0/1, RPF nbr 10.1.45.5
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:02:26/00:03:02
```

R4 is routing off of the (S,G) state.

On R3

```
R3#Show ip mroute | s 224.1.2.3

(*, 224.1.2.3), 00:09:59/stopped, RP 1.1.1.1, flags: SP
  Incoming interface: Serial0/1, RPF nbr 10.1.13.1
  Outgoing interface list: Null
(1.1.1.6, 224.1.2.3), 00:00:06/00:03:23, flags: T
  Incoming interface: FastEthernet0/0, RPF nbr 100.1.34.4
  Outgoing interface list:
    Serial0/1, Forward/Sparse, 00:00:06/00:03:23
```

(The rest of the output is omitted)

NOTE: R3 receives the Mcast flow from its F0/0 interface and forwards it out of its S0/1 interface toward R1.

On R1

```
R1#Show ip mroute | s 224.1.2.3

(*, 224.1.2.3), 01:43:05/00:02:41, RP 1.1.1.1, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 01:42:24/00:02:41
(1.1.1.6, 224.1.2.3), 00:01:57/00:03:22, flags: MT
  Incoming interface: Serial0/1, RPF nbr 10.1.13.3
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:01:57/00:02:41
(10.1.56.6, 224.1.2.3), 00:01:57/00:01:02, flags: M
  Incoming interface: Serial0/1, RPF nbr 10.1.13.3
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:01:57/00:02:41

R1#Show ip mroute summary | b out

Outgoing interface flags: H - Hardware switched, A - Assert winner
 Timers: Uptime/Expires
 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.2.3), 01:46:16/00:03:26, RP 1.1.1.1, OIF count: 1, flags: S
  (10.1.56.6, 224.1.2.3), 00:01:38/00:01:21, OIF count: 1, flags: M
  (1.1.1.6, 224.1.2.3), 00:05:09/00:03:21, OIF count: 1, flags: MT
(*, 224.0.1.40), 02:37:56/00:03:28, RP 1.1.1.1, OIF count: 3, flags: SJCL
```

The "M" shows that this state was caused by the MSDP relationship. The 10.1.56.6 source is only present due to the way the test was set up as R6 being the source and generator of the traffic. If another router connected to R6 were generating traffic, the additional state would not be present.

On R2

```
R2#Sh ip mroute | s 224.1.2.3

(*, 224.1.2.3), 01:48:07/stopped, RP 1.1.1.1, flags: SJCL
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.12.1
  Outgoing interface list:
    FastEthernet0/1, Forward/Sparse, 01:47:27/00:02:04
(10.1.56.6, 224.1.2.3), 00:01:00/00:02:00, flags: LJT
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.12.1
  Outgoing interface list:
    FastEthernet0/1, Forward/Sparse, 00:01:00/00:02:04
(1.1.1.6, 224.1.2.3), 00:07:00/00:02:59, flags: LJT
  Incoming interface: FastEthernet0/0, RPF nbr 10.1.12.1
  Outgoing interface list:
    FastEthernet0/1, Forward/Sparse, 00:07:00/00:02:04
```

The incoming interface allows the router to have a valid RPF towards the RP. The actual forwarding of the Multicast traffic is performed by the S,G state OIL.

To Verify the RPF check:

On R3

```
R3#Show ip rpf 1.1.1.6

RPF information for ? (1.1.1.6)

  RPF interface: FastEthernet0/0
  RPF neighbor: ? (100.1.34.4)
  RPF route/mask: 1.1.1.6/32
  RPF type: unicast (bgp 100)
  RPF recursion count: 1
  Doing distance-preferred lookups across tables
```

The RPF check is successful across the F0/0 interface and is verified to be a unicast bgp type.

```
R3#Show ip mroute | s 224.1.2.3

(*, 224.1.2.3), 00:12:13/stopped, RP 1.1.1.1, flags: SP
```

```
  Incoming interface: Serial0/1, RPF nbr 10.1.13.1
  Outgoing interface list: Null
(10.1.56.6, 224.1.2.3), 00:00:22/00:03:07, flags:
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1, Forward/Sparse, 00:00:22/00:03:07
(1.1.1.6, 224.1.2.3), 00:12:13/00:03:26, flags: T
  Incoming interface: FastEthernet0/0, RPF nbr 100.1.34.4
  Outgoing interface list:
    Serial0/1, Forward/Sparse, 00:12:13/00:03:06
```

The traffic flows from R4 toward R2, as multicast should in this case. The incoming interface is F0/0…for now.

Task 5

Reconfigure the msdp peering to not use the "connect-source" option. BGP must remain configured on R3 and R4 only. The routers must be configured such that Multicast traffic uses the F0/1 interface and the Unicast traffic uses the F0/0 interface. There should be no "?" status in the "show msdp peer" or "show ip rpf" command output on R3. Configure the RPs to be 1.1.1.3 and 1.1.1.4 for AS 100 and AS 200 respectively.

Reconfigure Multicast:

On R1

```
R1(config)#NO ip pim rp-address 1.1.1.1
R1(config)#ip pim rp-address 1.1.1.3

R2(config)#NO ip msdp peer 1.1.1.5
```

On R2

```
R2(config)#NO ip pim rp-address 1.1.1.1
R2(config)#ip pim rp-address 1.1.1.3
```

On R3

```
R3(config)#NO ip pim rp-address 1.1.1.1
R3(config)#ip pim rp-address 1.1.1.3
```

```
R3(config)#ip msdp peer 200.1.34.4 remote-as 200

R3(config)#router bgp 100
R3(config-router)#neighbor 200.1.34.4 remote-as 200

R3(config-router)#address-family ipv4 multicast
R3(config-router-af)#neighbor 200.1.34.4 activate
```

On R4

```
R4(config)#NO ip pim rp-address 1.1.1.5
R4(config)#ip pim rp-address 1.1.1.4

R4(config)#ip msdp peer 200.1.34.3 remote-as 100

R4(config)#router bgp 200
R4(config-router)#neighbor 200.1.34.3 remote-as 100

R4(config-router-af)#address-family ipv4 multicast
R4(config-router-af)#neighbor 200.1.34.3 activate
```

On R5

```
R5(config)#NO ip pim rp-address 1.1.1.5
R5(config)#ip pim rp-address 1.1.1.4

R5(config)#NO ip msdp peer 1.1.1.1
```

The msdp peering is removed and configured on the PE routers (R3 and R4). This allows us to use the "remote-as option". The peering ID should match the current BGP peering to work correctly. All of the rp-address needs to change to reflect the msdp peering source.

On R6

```
R6(config)#No ip pim rp-address 1.1.1.5
R6(config)#ip pim rp-address 1.1.1.4

R6#Ping 224.1.2.3 source lo0

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.2.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.6

Reply to request 0 from 10.1.12.1, 80 ms
```

Verify the configuration:

On R4

```
R4#Sh ip bgp ipv4 multicast summary

BGP router identifier 1.1.1.4, local AS number 200
BGP table version is 1, main routing table version 1

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
200.1.34.3      4   100      22      22        1    0    0 00:05:07        0
```

On R3

```
R3#Sh ip msdp peer

MSDP Peer 200.1.34.4 (?), AS 200 (configured AS)
  Connection status:
    State: Up, Resets: 0, Connection source: none configured
    Uptime(Downtime): 00:09:23, Messages sent/received: 10/11
    Output messages discarded: 0
    Connection and counters cleared 00:14:53 ago
```

The question mark is still present in the msdp peer output and the "show ip rpf" command. The AS100 was replaced along with the configured AS output due to the AS being used instead of the " connect source option".

```
R3#Sh ip rpf 1.1.1.6

RPF information for ? (1.1.1.6)
  RPF interface: FastEthernet0/0
  RPF neighbor: ? (100.1.34.4)
  RPF route/mask: 1.1.1.6/32
  RPF type: unicast (bgp 100)
  RPF recursion count: 1
  Doing distance-preferred lookups across tables
```

Even though the Multicast AF has been activated, the RPF check is still using the unicast path.

On R4

```
R4(config)#router bgp 200
R4(config-router)#address-family ipv4 multicast
R4(config-router-af)#Netw 1.1.1.6 mask 255.255.255.255
```

The source needs to be added to BGP Multicast AF to aid with the RPF check.

On R3

```
R3#Sh ip rpf 1.1.1.6

RPF information for ? (1.1.1.6)
  RPF interface: FastEthernet0/1
  RPF neighbor: ? (200.1.34.4)
  RPF route/mask: 1.1.1.6/32
  RPF type: mbgp
  RPF recursion count: 0
  Doing distance-preferred lookups across tables

R3#Sh ip mroute | s 224.1.2.3

(*, 224.1.2.3), 01:20:31/stopped, RP 1.1.1.3, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1, Forward/Sparse, 00:54:24/00:03:07
(10.1.56.6, 224.1.2.3), 00:00:08/00:03:21, flags:
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/1, Forward/Sparse, 00:00:08/00:03:21
(1.1.1.6, 224.1.2.3), 00:00:08/00:02:51, flags: M
  Incoming interface: FastEthernet0/1, RPF nbr 200.1.34.4, Mbgp
  Outgoing interface list:
    Serial0/1, Forward/Sparse, 00:00:08/00:03:21
```

The RPF has passed on the MP-BGP Multicast path. BGP can actually find a source through BGP updates instead of the standard unicast RPF check. The interface has changed to be the path of the new peering session.

Dealing with names:

```
R3(config)#ip host r4 200.1.34.4
R3(config)#ip host r6 1.1.1.6
```

We simply add the host commands to replace the question mark output. R4 for the peer and R6 for the source on the "show ip rpf" command.

```
R3#Sh ip msdp peer

MSDP Peer 200.1.34.4 (R4), AS 200 (configured AS)
  Connection status:
    State: Up, Resets: 0, Connection source: none configured
```

```
    Uptime(Downtime): 00:19:06, Messages sent/received: 20/24
    Output messages discarded: 0
    Connection and counters cleared 00:24:36 ago
```

(The rest of the output is omitted)

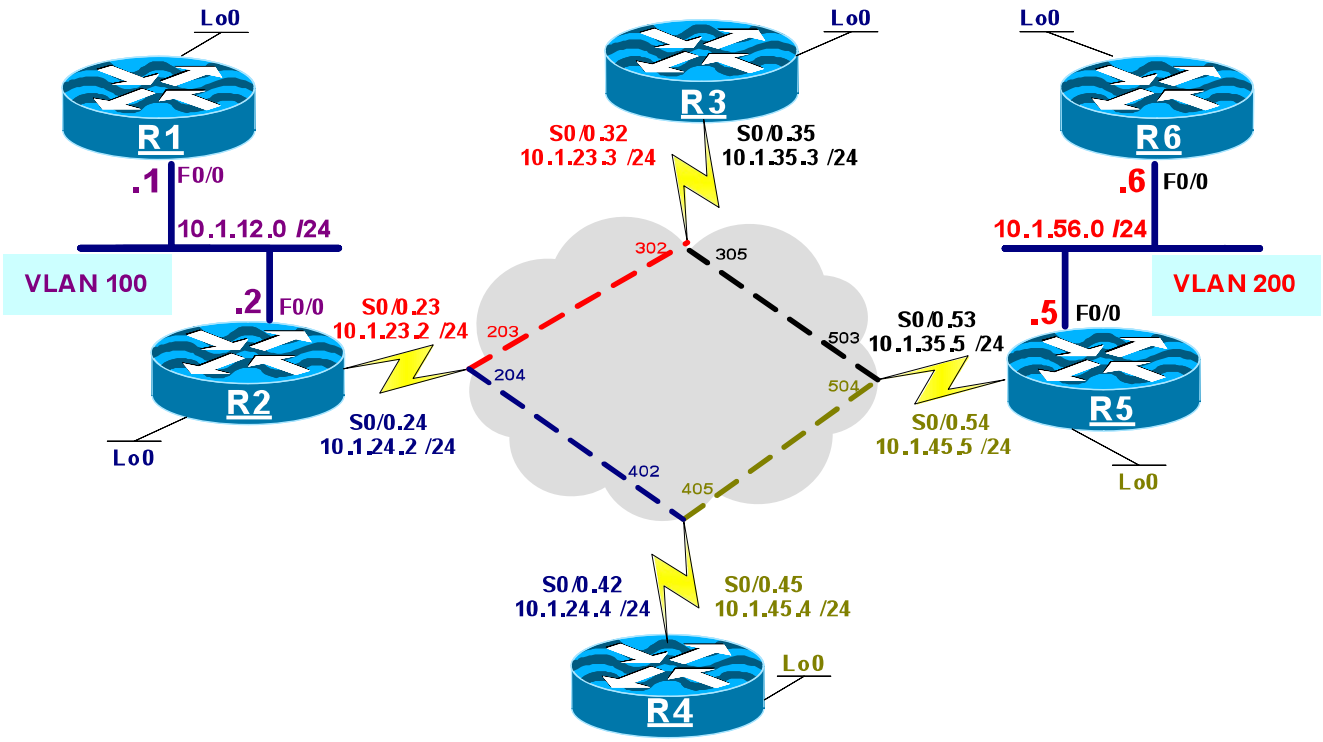
```
R3#Sh ip rpf 1.1.1.6

RPF information for R6 (1.1.1.6)
  RPF interface: FastEthernet0/1
  RPF neighbor: R4 (200.1.34.4)
  RPF route/mask: 1.1.1.6/32
  RPF type: mbgp
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
```

Changes are made and verified.

Task 6

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 10 – Configuring SSM



Lab Setup

- Configure the F0/0 interface of R1 and R2 in VLAN 100, and F0/0 interface of R4 and R5 in VLAN 200.
- All frame-Relay connections must be configured in a point-to-point manner.
- Use the IP addressing chart below for IP addressing scheme

IP addressing Chart:

Router	Interface / IP addressing
R1	F0/0 = 10.1.12.1 /24 Lo0= 1.1.1.1 /32
R2	**F0/0 = 10.1.12.2 /24** **S0/0.23 = 10.1.23.2/24** **S0/0.24 = 10.1.24.2/24** **Lo0= 1.1.1.2 /32**
R3	S0/0.32 = 10.1.23.3/24 S0/0.35 = 10.1.35.3/24 Lo0= 1.1.1.3 /32
R4	**S0/0.42 = 10.1.24.4/24** **S0/0.45 = 10.1.45.4 /24** **Lo0= 1.1.1.4 /32**
R5	S0/0.53 = 10.1.35.5/24 S0/0.54 = 10.1.45.5/24 F0/0 = 10.1.56.5/24 Lo0= 1.1.1.5 /32
R6	**F0/0 = 10.1.56.6/24** **Lo0= 1.1.1.6/32**

Task 1

Configure OSPF area 0 on all interfaces.

On All Routers

```
Rx(config)#router ospf 1
Rx(config-router)#netw 0.0.0.0 0.0.0.0 are 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf

     1.0.0.0/32 is subnetted, 6 subnets
O       1.1.1.3 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.2 [110/2] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.5 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.4 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.6 [110/131] via 10.1.12.2, 00:00:19, FastEthernet0/0
     10.0.0.0/24 is subnetted, 6 subnets
```

```
O       10.1.24.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.45.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.35.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.56.0 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
```

Task 2

Configure PIM on the interfaces according to the following table:

Router	Interface / PIM
R1	F0/0 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R2	**F0/0 = PIM Sparse-mode** **S0/0.23 = PIM Sparse-mode** **S0/0.24 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**
R3	S0/0.32 = PIM Sparse-mode S0/0.35 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R4	**S0/0.42 = PIM Sparse-mode** **S0/0.45 = PIM Sparse-mode** **Lo0= PIM Sparse-mode**
R5	F0/0 = PIM Sparse-mode S0/0.53 = PIM Sparse-mode S0/0.54 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R6	**F0/0 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**

On R1

```
R1(config)#ip multicast-routing

R1(config)#int lo 0
R1(config-if)#ip pim sparse-mode

R1(config-if)#int f0/0
R1(config-if)#ip pim sparse-mode
```

On R2

```
R2(config)#ip multicast-routing

R2(config)#int lo0
R2(config-if)#ip pim sparse-mode

R2(config-if)#int f0/0
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.23
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.24
R2(config-subif)#ip pim sparse-mode
```

On R3

```
R3(config)#ip multicast-routing

R3(config)#int lo0
R3(config-if)#ip pim sparse-mode

R3(config-if)#int s0/0.32
R3(config-subif)#ip pim sparse-mode

R3(config-subif)#int s0/0.35
R3(config-subif)#ip pim sparse-mode
```

On R4

```
R4(config)#ip multicast-routing

R4(config)#int lo0
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.42
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.45
R4(config-if)#ip pim sparse-mode
```

On R5

```
R5(config)#ip multicast-routing
```

```
R5(config)#int lo0
R5(config-if)#ip pim sparse-mode

R5(config-if)#int f0/0
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.53
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.54
R5(config-subif)#ip pim sparse-mode
```

On R6

```
R6(config)#ip multicast-routing

R6(config)#int lo0
R6(config-if)#ip pim sparse-mode

R6(config-if)#int f0/0
R6(config-if)#ip pim sparse-mode
```

To verify the configuration:

On R1

```
R1#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.2         FastEthernet0/0          01:00:02/00:01:17 v2    1 / DR S
```

On R2

```
R2#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.1         FastEthernet0/0          01:01:24/00:01:20 v2    1 / S
10.1.23.3         Serial0/0.23             01:00:35/00:01:41 v2    1 / S
10.1.24.4         Serial0/0.24             01:00:21/00:01:31 v2    1 / S
```

On R3

```
R3#Show ip pim neighbor | b interface
```

```
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.23.2         Serial0/0.32             01:00:35/00:01:42 v2    1 / S
10.1.35.5         Serial0/0.35             01:00:04/00:01:40 v2    1 / S
```

On R4

```
R4#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.24.2         Serial0/0.42             01:00:21/00:01:28 v2    1 / S
10.1.45.5         Serial0/0.45             00:59:59/00:01:19 v2    1 / S
```

On R5

```
R5#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.6         FastEthernet0/0          00:54:54/00:01:32 v2    1 / DR S
10.1.35.3         Serial0/0.53             01:00:04/00:01:43 v2    1 / S
10.1.45.4         Serial0/0.54             00:59:59/00:01:16 v2    1 / S
```

On R6

```
R6#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.5         FastEthernet0/0          00:54:54/00:01:21 v2    1 / S
```

Task 3

Configure all routers except R1 and R6, to use SSM for the standard SSM scope allocated by IANA, you should use an access-list to accomplish this task.

The default SSM scope is 232.0.0.0/8, but it can be changed by specifying a different range, this is accomplished by configuring an Access-list and referencing the access-list by using the "range" keyword.

Note: In SSM, only the router, which is closest to the receiving host/s (The last hop router), needs to have SSM enabled, in this task the range is defined on R2 – R5.

On R2

```
R2(config)#access-list 1 permit 232.0.0.0 0.255.255.255
R2(config)#ip pim ssm range 1
```

On R3

```
R3(config)#access-list 1 permit 232.0.0.0 0.255.255.255
R3(config)#ip pim ssm range 1
```

On R4

```
R4(config)#access-list 1 permit 232.0.0.0 0.255.255.255
R4(config)#ip pim ssm range 1
```

On R5

```
R5(config)#access-list 1 permit 232.0.0.0 0.255.255.255
R5(config)#ip pim ssm range 1
```

Task 4

Configure R1 to join the [10.1.56.6, 232.6.6.6 (S,G)]. You should use ping to verify and test the configuration. R1 and R6 should be configured as hosts having a default route pointing to their directly connected neighbor.

In the following configuration IGMP version 3 is enabled and the "ip igmp join" command is configured to support SSM using the "Source" keyword.

On R1

```
R1(config)#int f0/0
R1(config-if)#ip igmp join-group 232.6.6.6 source 10.1.56.6
R1(config-if)#ip igmp version 3
```

Next, R2 is configured to listen to IGMPv3 on its F0/0 interface facing R1 (Vlan 100).

On R2

```
R2(config)#int f0/0
```

```
R2(config-subif)#ip igmp version 3
```

In the following step R1 and R6 are configured as hosts with a gateway pointing to their directly connected router, this is done to test the configuration, IN THE REAL CCIE LAB YOU SHOULD NOT DO THIS:

On R1:

```
R1(config)#NO ip multicast-routing
R1(config)#NO ip routing
R1(config)#ip default-gateway 10.1.12.2

R1(config)#int lo0
R1(config-if)#NO ip pim sparse-mode

R1(config)#int f0/0
R1(config-if)#NO ip pim sparse-mode
```

On R6

```
R6(config)#NO ip multicast-routing
R6(config)#NO ip routing
R6(config)#ip default-gateway 10.1.56.5
!
R6(config)#int lo0
R6(config-if)#NO ip pim sparse-mode

R6(config)#int f0/0
R6(config-if)#NO ip pim sparse-mode
```

To verify the configuration:

On R2

```
R2#Sh ip igmp groups 232.6.6.6 detail

Flags: L - Local, U - User, SG - Static Group, VG - Virtual Group,
       SS - Static Source, VS - Virtual Source,
       Ac - Group accounted towards access control limit

Interface:      FastEthernet0/0
Group:          232.6.6.6
Flags:          SSM
Uptime:         00:01:34
Group mode:     INCLUDE
Last reporter:  10.1.12.1
```

```
Group source list: (C - Cisco Src Report, U - URD, R - Remote, S - Static,
                    V - Virtual, M - SSM Mapping, L - Local,
                    Ac - Channel accounted towards access control limit)
  Source Address   Uptime     v3 Exp    CSR Exp    Fwd  Flags
  10.1.56.6        00:01:34   00:02:26  stopped    Yes  R
```

To verify the configuration:

On R6

```
R6#Ping 232.6.6.6

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 232.6.6.6, timeout is 2 seconds:

Reply to request 0 from 10.1.12.1, 20 ms

R6#Ping
Protocol [ip]: → Press Enter
Target IP address: 232.6.6.6
Repeat count [1]: → Press Enter
Datagram size [100]: → Press Enter
Timeout in seconds [2]: → Press Enter
Extended commands [n]: y
Interface [All]: loopback 0
Time to live [255]: → Press Enter
Source address: 1.1.1.6
Type of service [0]: → Press Enter
Set DF bit in IP header? [no]: → Press Enter
Validate reply data? [no]: → Press Enter
Data pattern [0xABCD]: → Press Enter
Loose, Strict, Record, Timestamp, Verbose[none]: → Press Enter
Sweep range of sizes [n]: → Press Enter
Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 232.6.6.6, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.6

.
```

NOTE the ping is NOT successful.

On R5

The ping that was generated from the loopback 0 interface of R6 was NOT successful because R1 ONLY responds to the ICMP messages coming from 10.1.56.6, which is the F0/0 interface of R6 and NO OTHER IP ADDRESS.

On R5

```
R5#Show ip mroute | s 232.6.6.6

 (10.1.56.6, 232.6.6.6), 00:33:03/00:02:59, flags: sT
  Incoming interface: FastEthernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.54, Forward/Sparse, 00:33:03/00:02:59
```

Note the (*, G) entry.

On R2

```
R2#Show ip mroute | s 232.6.6.6

(10.1.56.6, 232.6.6.6), 00:35:38/00:02:09, flags: sTI
  Incoming interface: Serial0/0.24, RPF nbr 10.1.24.4
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:35:38/00:02:09
```

The output looks the same except the "I" flag, which indicates that R2 has a listener for the S, G on one of its local interfaces.

Task 5

Configure R2 to allow minimal leave latencies when a host leaves a multicast group in VLAN 100.

The "ip igmp explicit-tracking" interface command causes the router to track all reporters and NOT only the last one, this provides the ability to leave an (S, G) as soon as the last host leaves that given (S, G) without sending a query and waiting for it to timeout.

On R2

```
R2(config)#int f0/0
R2(config-subif)#ip igmp explicit-tracking
```

To verify the configuration:

On R2

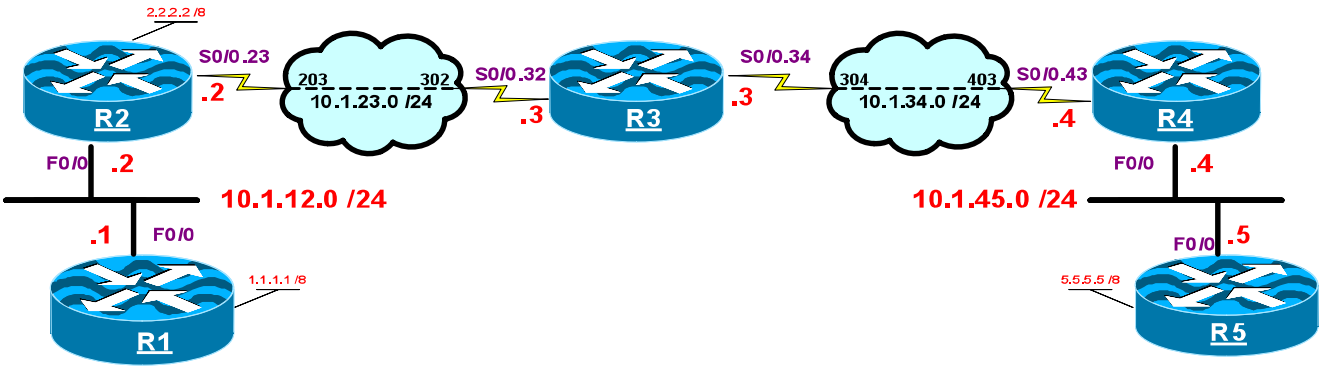
```
R2#Show ip igmp membership tracked

Flags: A  - aggregate, T - tracked
       L  - Local, S - static, V - virtual, R - Reported through v3
       I - v3lite, U - Urd, M - SSM (S,G) channel
       1,2,3 - The version of IGMP the group is in
Channel/Group-Flags:
       / - Filtering entry (Exclude mode (S,G), Include mode (*,G))
Reporter:
       <mac-or-ip-address> - last reporter if group is not explicitly tracked
       <n>/<m>      - <n> reporter in include mode, <m> reporter in exclude
 Channel/Group                 Reporter        Uptime   Exp.  Flags  Interface
 10.1.56.6,232.6.6.6           10.1.12.1       00:00:28 02:31 RT     Fa0/0
```

Notice the "T" flag indicates that the entry is tracked.

Task 6

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 11 – Helper-Map



Lab Setup:

- Configure the F0/0 interface of R1, R2 in VLAN 100.
- Configure the F0/0 interface of R4, R5 in VLAN 200.
- Configure all frame-Relay connections in point-to-point manner.
- Configure the IP addressing based on the following chart.

IP addressing:

Router	Interface / IP address	DLCI assignment
R1	F0/0 = 10.1.12.1 /24 Loopback0 = 1.1.1.1 /8	
R2	**F0/0 = 10.1.12.2 /24** **S0/0.23 = 10.1.23.2 /24** **Loopback0 = 2.2.2.2 /8**	**203**
R3	S0/0.32 = 10.1.23.3 /24 S0/0.34 = 10.1.34.3 /24	302 304
R4	**S0/0.43 = 10.1.34.4 /24** **F0/0 = 10.1.45.4 /24**	**403** **405**
R5	**F0/0 = 10.1.45.5 /24** **Loopback0 = 5.5.5.5 /8**	

Task 1

Configure OSPF Area 0 on **all frame-Relay links** and the loopback0 interface of R2.

On R2

```
R2(config)#router ospf 1
R2(config-router)#netw 10.1.23.2 0.0.0.0 area 0
R2(config-router)#netw 2.2.2.2 0.0.0.0 area 0
```

On R3

```
R3(config)#router ospf 1
R3(config-router)#netw 0.0.0.0 0.0.0.0 area 0
```

On R4

```
R4(config)#router ospf 1
R4(config-router)#netw 10.1.34.4 0.0.0.0 area 0
```

To verify the configuration:

On R2

```
R2#Show ip route ospf | inc O

O       10.1.34.0 [110/128] via 10.1.23.3, 00:00:54, Serial0/0.23
```

On R4

```
R4#Show ip route ospf | inc O

O       2.2.2.2 [110/129] via 10.1.34.3, 00:01:37, Serial0/0.43
O       10.1.23.0 [110/128] via 10.1.34.3, 00:01:37, Serial0/0.43
```

Task 2

Configure **RIPv2 on the F0/0 interface of R1 and R5 and loopback 0** interfaces of these routers. Disable the auto-summarization on both of these routers.

On R1

```
R1(config)#router rip
R1(config-router)#No au
R1(config-router)#ver 2
R1(config-router)#netw 10.0.0.0
R1(config-router)#netw 1.0.0.0
```

On R5

```
R5(config)#router rip
R5(config-router)#No au
R5(config-router)#ver 2
R5(config-router)#netw 10.0.0.0
R5(config-router)#netw 5.0.0.0
```

Task 3

Configure Multicasting on the appropriate routers such that R5 receives all the **RIPv2** updates from R1.
R2 should be configured as the RP and BSR router; this router should use its loopback interface as the source of all its BSR messages. You MUST use 224.1.1.1 to accomplish this task.

This task calls for the "helper-map" configuration, as follows:

In the first step of this configuration, R1 is configured to send RIPv2 updates to a broadcast destination so R2 can map them to a Multicast address of 224.1.1.1.

```
R1(config)#int f0/0
R1(config-if)#ip rip v2-broadcast
```

Step two:

In this step IP, multicast routing should be enabled on all transit routers (R2, R3 and R4).

On R2, R3 and R4

```
Rx(config)#ip multicast-routing
```

Step Three:

Pim sparse mode should be enabled on R2's F0/0, S0/0.23 and its loopback 0 interface, on R3's S0/0.32 and S0/0.34, on R4's S0/0.43 interface:

On R2

```
R2(config)#int f0/0
R2(config-if)#ip pim sparse-mode

R2(config)#int s0/0.23
R2(config-subif)#ip pim sparse-mode

R2(config)#int lo0
R2(config-if)#ip pim sparse-mode
```

On R3

```
R3(config)#int s0/0.32
R3(config-subif)#ip pim sparse-mode

R3(config)#int s0/0.34
R3(config-subif)#ip pim sparse-mode
```

On R4

```
R4(config)#int s0/0.43
R4(config-subif)#ip pim sparse-mode
```

To verify the configuration:

On R4

```
R4#Show ip pim neighbor | b inter

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.34.3         Serial0/0.43             00:00:47/00:01:26 v2    1 / S
```

On R3

```
R3#Show ip pim neighbor | b inter

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.23.2         Serial0/0.32             00:01:50/00:01:22 v2    1 / S
10.1.34.4         Serial0/0.34             00:01:31/00:01:41 v2    1 / S
```

On R2

```
R2#Show ip pim neighbor

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.23.3         Serial0/0.23             00:03:02/00:01:40 v2    1 / S
```

Step four:

BSR multicast routing is configured on R2; in this step R2 is configured as the RP and the BSR using the loopback0 interface as the source of its packets:

On R2

```
R2(config)#ip pim rp-candidate lo0
R2(config)#ip pim bsr-candidate lo0
```

To verify the configuration:

It may take upto 150 seconds before the routers receive the BSR messages:

On R3

```
R3#Show ip pim rp mapping

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 2.2.2.2 (?), v2
    Info source: 2.2.2.2 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 00:00:10, expires: 00:02:24
```

On R4

```
R4#Show ip pim rp map

PIM Group-to-RP Mappings

Group(s) 224.0.0.0/4
  RP 2.2.2.2 (?), v2
    Info source: 2.2.2.2 (?), via bootstrap, priority 0, holdtime 150
         Uptime: 00:01:24, expires: 00:02:09
```

Step five:

In the following step, an access-list is configured to identify the RIP traffic being sent from 10.1.12.1

(R1's F0/0 interface) to a broadcast IP address destined for RIP.

R2(config)#access-list 100 permit udp host 10.1.12.1 eq rip host 255.255.255.255 eq rip

The following command specifies the forwarding of broadcast messages destined to UDP 520 (RIP):

```
R2(config)#ip forward-protocol udp rip
```

The following command is configured to converts the broadcast traffic arriving at the F0/0 interface of the first hop router (The router closest to the source) destined for UDP port 520 (RIP) to a multicast group destination address of 224.1.1.1:

```
R2(config)#int f0/0
R2(config-if)#ip multicast helper-map broadcast 224.1.1.1 100 ttl 3
```

Note the TTL keyword specifies the TTL value of multicast packets generated by the helper-map from incoming broadcast packets. The range is 1 – 50 hops, this keyword is required because RIP is using a TTL=2, which means R4 will not forward packets coming from R2.

On R4

On the last hop router, the traffic is once again identified using access-list 100 and forwarding of broadcast messages destined to UDP 520 (RIP) is specified:

```
R4(config)#access-list 100 permit udp host 10.1.12.1 any eq rip
R4(config)#ip forward-protocol udp rip
```

In the following command, the last hop router (Router closest to the group members) is configured to convert the multicast traffic arriving at S0/0.43 interface back to broadcast and send it out of its F0/0 interface. You should "Clear ip route *" on R1 few times so it generates RIP traffic immediately.

```
R4(config)#int s0/0.43
R4(config-subif)#ip multicast helper-map 224.1.1.1 10.1.45.255 100
```

Since the traffic will be sent to the F0/0 interface of R4 as a directed broadcast, directed broadcast is enabled:

```
R4(config)#int f0/0
R4(config-if)#ip directed-broadcast
```

Step Six:

Since the RIPv2 update will be coming from another subnet (10.1.12.0 /24) the validation of the source IP address by RIPv2 will fail (this is called the sanity check) therefore, RIP routes will NOT be processed by R5, in this step the validation of source IP address of the Updates are disabled:

On R5

```
R5(config)#router rip
R5(config-router)#NO validate-update-source
```

To verify the configuration:

On R5

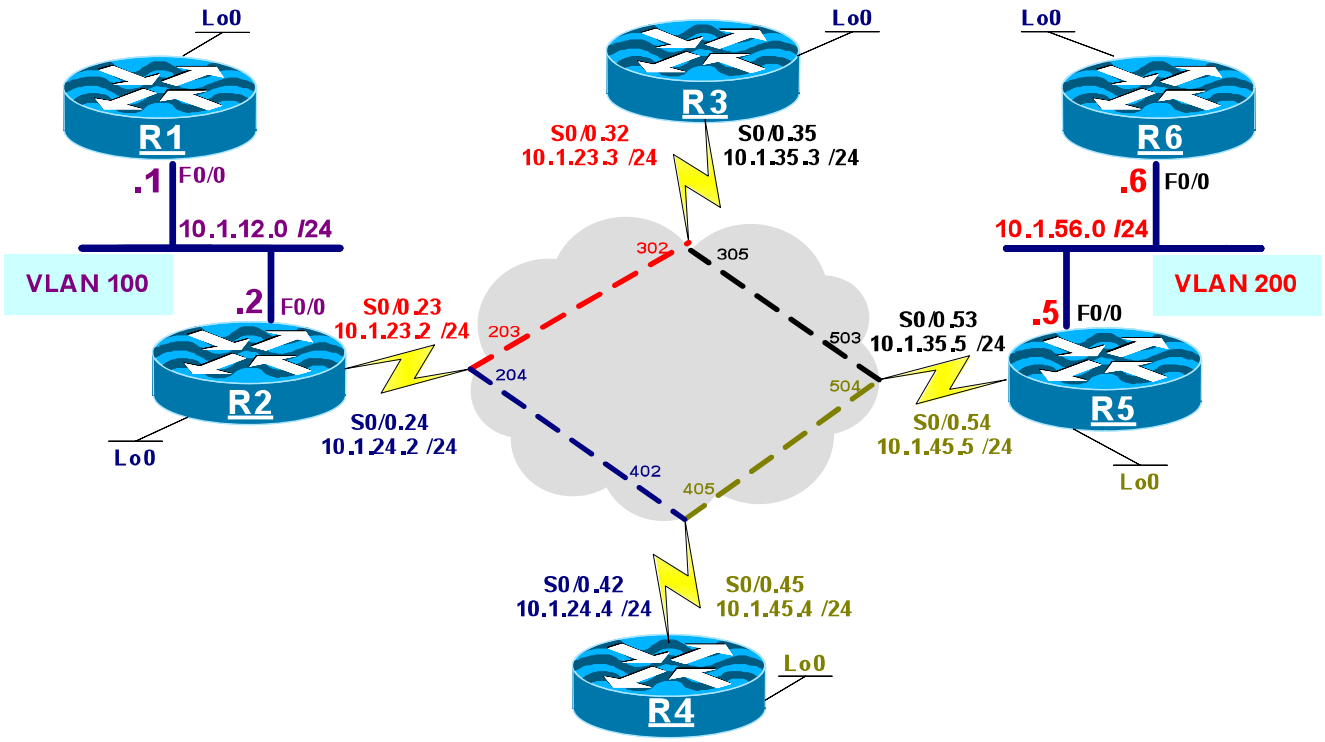
```
R5#Sh ip route rip

R    1.0.0.0/8 [120/1] via 10.1.12.1, 00:00:05
```

Note R5 receives the RIPv2 route/s advertised by R1.

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

LAB 12 - Bidirectional PIM



Lab Setup

- Configure the F0/0 interface of R1 and R2 in VLAN 100, and F0/0 interface of R4 and R5 in VLAN 200.
- All frame-Relay connections must be configured in a point-to-point manner.
- Use the IP addressing chart below for IP addressing scheme

IP addressing Chart:

Router	Interface / IP addressing
R1	F0/0 = 10.1.12.1 /24 Lo0= 1.1.1.1 /32
R2	**F0/0 = 10.1.12.2 /24** **S0/0.23 = 10.1.23.2/24** **S0/0.24 = 10.1.24.2/24** **Lo0= 1.1.1.2 /32**
R3	S0/0.32 = 10.1.23.3/24 S0/0.35 = 10.1.35.3/24 Lo0= 1.1.1.3 /32
R4	**S0/0.42 = 10.1.24.4/24** **S0/0.45 = 10.1.45.4 /24** **Lo0= 1.1.1.4 /32**
R5	S0/0.53 = 10.1.35.5/24 S0/0.54 = 10.1.45.5/24 F0/0 = 10.1.56.5/24 Lo0= 1.1.1.5 /32
R6	**F0/0 = 10.1.56.6/24** **Lo0= 1.1.1.6/32**

Task 1

Configure OSPF area 0 on all interfaces.

On All Routers

```
Rx(config)#router ospf 1
Rx(config-router)#netw 0.0.0.0 0.0.0.0 are 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf

     1.0.0.0/32 is subnetted, 6 subnets
O       1.1.1.3 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.2 [110/2] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.5 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.4 [110/66] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       1.1.1.6 [110/131] via 10.1.12.2, 00:00:19, FastEthernet0/0
     10.0.0.0/24 is subnetted, 6 subnets
```

```
O       10.1.24.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.45.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.35.0 [110/129] via 10.1.12.2, 00:00:19, FastEthernet0/0
O       10.1.56.0 [110/130] via 10.1.12.2, 00:00:19, FastEthernet0/0
```

Task 2

Configure PIM on the interfaces according to the following table:

Router	Interface / PIM
R1	F0/0 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R2	**F0/0 = PIM Sparse-mode** **S0/0.23 = PIM Sparse-mode** **S0/0.24 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**
R3	S0/0.32 = PIM Sparse-mode S0/0.35 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R4	**S0/0.42 = PIM Sparse-mode** **S0/0.45 = PIM Sparse-mode** **Lo0= PIM Sparse-mode**
R5	F0/0 = PIM Sparse-mode S0/0.53 = PIM Sparse-mode S0/0.54 = PIM Sparse-mode Lo0 = PIM Sparse-mode
R6	**F0/0 = PIM Sparse-mode** **Lo0 = PIM Sparse-mode**

On R1

```
R1(config)#ip multicast-routing

R1(config)#int lo 0
R1(config-if)#ip pim sparse-mode

R1(config-if)#int f0/0
R1(config-if)#ip pim sparse-mode
```

On R2

```
R2(config)#ip multicast-routing

R2(config)#int lo0
R2(config-if)#ip pim sparse-mode

R2(config-if)#int f0/0
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.23
R2(config-subif)#ip pim sparse-mode

R2(config-subif)#int s0/0.24
R2(config-subif)#ip pim sparse-mode
```

On R3

```
R3(config)#ip multicast-routing

R3(config)#int lo0
R3(config-if)#ip pim sparse-mode

R3(config-if)#int s0/0.32
R3(config-subif)#ip pim sparse-mode

R3(config-subif)#int s0/0.35
R3(config-subif)#ip pim sparse-mode
```

On R4

```
R4(config)#ip multicast-routing

R4(config)#int lo0
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.42
R4(config-if)#ip pim sparse-mode

R4(config-if)#int s0/0.45
R4(config-if)#ip pim sparse-mode
```

On R5

```
R5(config)#ip multicast-routing
```

```
R5(config)#int lo0
R5(config-if)#ip pim sparse-mode

R5(config-if)#int f0/0
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.53
R5(config-subif)#ip pim sparse-mode

R5(config-subif)#int s0/0.54
R5(config-subif)#ip pim sparse-mode
```

On R6

```
R6(config)#ip multicast-routing

R6(config)#int lo0
R6(config-if)#ip pim sparse-mode

R6(config-if)#int f0/0
R6(config-if)#ip pim sparse-mode
```

To verify the configuration:

On R1

```
R1#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.2         FastEthernet0/0          01:00:02/00:01:17 v2    1 / DR S
```

On R2

```
R2#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.12.1         FastEthernet0/0          01:01:24/00:01:20 v2    1 / S
10.1.23.3         Serial0/0.23             01:00:35/00:01:41 v2    1 / S
10.1.24.4         Serial0/0.24             01:00:21/00:01:31 v2    1 / S
```

On R3

```
R3#Show ip pim neighbor | b interface
```

```
Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.23.2         Serial0/0.32             01:00:35/00:01:42 v2    1 / S
10.1.35.5         Serial0/0.35             01:00:04/00:01:40 v2    1 / S
```

On R4

```
R4#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.24.2         Serial0/0.42             01:00:21/00:01:28 v2    1 / S
10.1.45.5         Serial0/0.45             00:59:59/00:01:19 v2    1 / S
```

On R5

```
R5#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.6         FastEthernet0/0          00:54:54/00:01:32 v2    1 / DR S
10.1.35.3         Serial0/0.53             01:00:04/00:01:43 v2    1 / S
10.1.45.4         Serial0/0.54             00:59:59/00:01:16 v2    1 / S
```

On R6

```
R6#Show ip pim neighbor | b interface

Neighbor          Interface                Uptime/Expires    Ver   DR
Address                                                            Prio/Mode
10.1.56.5         FastEthernet0/0          00:54:54/00:01:21 v2    1 / S
```

Task 3

Configure the routers to support multicast forwarding for groups in two directions. Use the loopback 0 interface on R3 as the RP address.

ON ALL Routers

```
RX(config)#ip pim bidir-enable   (This must be enabled first!)
RX(config)#ip pim rp-address 1.1.1.3 bidir
```

On R3

```
R3#Sh ip pim int s0/0.35 df

* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Serial0/0.35             1.1.1.3          *10.1.35.3       0          00:02:31
```

The Designated Forwarder is a Multicast router that is able to forward (*,G) state in 2 different directions for the same group address. In this case, there are 2 routers, R3 and R4 that are in the same path for the 1.1.1.1 or the 1.1.1.6 sources used in the lab. Only one of the routers is allowed to forward for any one group. The DF winner is determined by the IGP cost on a link-by-link basis.

```
R3#Sh ip pim int s0/0.35 df 1.1.1.3

Designated Forwarder election for Serial0/0.35, 10.1.35.3, RP 1.1.1.3
  State                          DF
  Offer count is                 0
  Current DF ip address          10.1.35.3
  DF winner up time              00:07:12
  Last winner metric preference  0
  Last winner metric             0
  Next winner will be sent in    30188 ms
```

R3 is the DF winner on the interface since it houses the RP-Address. Unlike normal SM operation, the RP does not have to be in the Data Path at all. The RP address must be reachable by all of participating multicast routers. The last winner metric is the cost to the RP. In this case, the cost is "0" since this R3 is the RP. R5 and R2 will show a cost of "0" since they are directly connected to the RP.

```
R3#Sh ip pim int s0/0.32 df

* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Serial0/0.32             1.1.1.3          *10.1.23.3       0          00:06:06

R3#Sh ip pim int s0/0.32 df 1.1.1.3

Designated Forwarder election for Serial0/0.32, 10.1.23.3, RP 1.1.1.3
  State                          DF
  Offer count is                 0
  Current DF ip address          10.1.23.3
  DF winner up time              00:08:12
  Last winner metric preference  0
  Last winner metric             0
  Next winner will be sent in    48364 ms
```

On R4

```
R4#Sh ip pim int s0/0.42 df

* implies this system is the DF
Interface                RP               DF Winner        Metric     Uptime
Serial0/0.42             1.1.1.3          10.1.24.2        65         00:07:50
```

Since R4 does not have the RP address, a true selection will take place that considers the cost to the RP. In this case, R4 elects R2 on one interface and R5 on the other interface as the DF Winner.

```
R4#Sh ip pim int s0/0.42 df 1.1.1.3

Designated Forwarder election for Serial0/0.42, 10.1.24.4, RP 1.1.1.3
  State                          Non-DF
  Offer count is                 0
  Current DF ip address          10.1.24.2
  DF winner up time              00:08:07
  Last winner metric preference  110
  Last winner metric             65
```

On R6

```
R6(config)#int lo0
R6(config-if)#ip igmp join-group 224.1.2.3
```

On R5

```
R5#Sh ip mroute bidirectional | s 224.1.2.3

(*, 224.1.2.3), 00:02:08/00:03:20, RP 1.1.1.3, flags: B
  Bidir-Upstream: Serial0/0.53, RPF nbr 10.1.35.3
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:02:08/00:03:20
    Serial0/0.53, Bidir-Upstream/Sparse, 00:02:08/00:00:00
```

After joining the RP, we can see that R5 forwards multicast traffic out of the F0/0 interface and adds the group entry, which forwards the traffic towards the RP. Bidirectional PIM is the opposite of SSM, as it does not create ANY (S,G) state. In SSM, there is no (*,G) state created. Also not the "B" flag denoting the Bidirectional state.

```
R5#Sh ip mroute bidir | s224.0.1.40

(*, 224.0.1.40), 00:13:58/00:02:58, RP 1.1.1.3, flags: BCL
  Bidir-Upstream: Serial0/0.53, RPF nbr 10.1.35.3
  Outgoing interface list:
```

```
    Serial0/0.54, Forward/Sparse, 00:11:01/00:02:48
    FastEthernet0/0, Forward/Sparse, 00:11:08/00:02:58
    Serial0/0.53, Bidir-Upstream/Sparse, 00:11:27/00:00:00
```

The bidirectional process will also apply to the Auto-RP discovery group as well. This will be filtered out later.

On R3

```
R3#Sh ip mroute bidirectional | s 224.1.2.3

(*, 224.1.2.3), 00:02:27/00:03:00, RP 1.1.1.3, flags: B
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.35, Forward/Sparse, 00:02:27/00:03:00
```

There are no other receivers looking for this group. Therefore, there is no need to add any more interfaces to the OIL.

On R1

```
R1#Ping 224.1.2.3

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.2.3, timeout is 2 seconds:

Reply to request 0 from 10.1.56.6, 68 ms
Reply to request 0 from 10.1.56.6, 104 ms
```

On R2

```
R2#Sh ip mroute bidirectional | s 224.1.2.3

(*, 224.1.2.3), 00:01:36/00:01:24, RP 1.1.1.3, flags: BP
  Bidir-Upstream: Serial0/0.23, RPF nbr 10.1.23.3
  Outgoing interface list:
    Serial0/0.23, Bidir-Upstream/Sparse, 00:01:36/00:00:00
```

The ping helps to create state toward the RP but the entry will not be created for bidirectional use until a join is initiated. The "P" flag is set to note that there is no other router interested in this traffic.

On R1

```
R1(config)#int lo0
R1(config-if)#ip igmp join-group 224.1.2.3
```

On R3

```
R3#Sh ip mroute bidirectional | s 224.1.2.3

(*, 224.1.2.3), 00:03:59/00:03:27, RP 1.1.1.3, flags: B
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.32, Forward/Sparse, 00:00:18/00:03:11
    Serial0/0.35, Forward/Sparse, 00:03:59/00:03:27

R6#Ping 224.1.2.3

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.2.3, timeout is 2 seconds:

Reply to request 0 from 10.1.56.6, 1 ms
Reply to request 0 from 10.1.12.1, 76 ms
Reply to request 0 from 10.1.12.1, 56 ms
Reply to request 0 from 1.1.1.6, 1 ms
```

All pings are successful!

Task4

Allow the R4 to take over as the primary path for forwarding group information.

On R3

```
R3#Sh ip mroute summary | b interface

Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.2.3), 00:01:03/00:02:32, RP 1.1.1.3, OIF count: 2, flags: B
(*, 224.0.1.40), 00:01:03/00:02:32, RP 1.1.1.3, OIF count: 3, flags: BCL

R3(config)#int s0/0.35
R3(config-subif)#shut
```

The shutdown of either subinterface will be fine. If we were to shut down the physical interface, the RP would no longer be reachable by the other routers.

On R4

A debug of R4 would have displayed the following after the R3 failure:

*Jun 24 01:46:14.895: PIM(0): Building Periodic Join/Prune message for 224.1.2.3
***Jun 24 01:46:15.547: PIM(0): Insert (1.1.1.3,224.1.2.3) prune in nbr 10.1.45.5's queue**
*Jun 24 01:46:15.547: PIM(0): Building Triggered Join/Prune message for 224.0.1.40
Jun 24 01:46:15.547: PIM(0): Insert (,224.0.1.40) join in nbr 10.1.24.2's queue
*Jun 24 01:46:15.547: PIM(0): Building Join/Prune packet for nbr 10.1.24.2
*Jun 24 01:46:15.547: PIM(0): Adding v2 (1.1.1.3/32, 224.0.1.40), WC-bit, RPT-bit, S-bit Join
*Jun 24 01:46:15.547: PIM(0): Send v2 join/prune to 10.1.24.2 (Serial0/0.42)
*Jun 24 01:46:15.547: PIM(0): Building Join/Prune packet for nbr 10.1.45.5
*Jun 24 01:46:15.547: PIM(0): Adding v2 (1.1.1.3/32, 224.1.2.3), WC-bit, RPT-bit, S-bit Prune
***Jun 24 01:46:15.547: PIM(0): Send v2 join/prune to 10.1.45.5 (Serial0/0.45)**
***Jun 24 01:46:15.811: PIM(0): Received v2 DF on Serial0/0.45 from 10.1.45.5**
***Jun 24 01:46:15.819: PIM(0): Received v2 DF on Serial0/0.45 from 10.1.45.5**
***Jun 24 01:46:15.911: PIM(0): Received v2 DF on Serial0/0.45 from 10.1.45.5**
***Jun 24 01:46:15.919: PIM(0): Received v2 DF on Serial0/0.45 from 10.1.45.5**
***Jun 24 01:46:15.923: PIM(0): Received v2 DF on Serial0/0.45 from 10.1.45.5**
*Jun 24 01:46:16.319: PIM(0): Received v2 Join/Prune on Serial0/0.45 from 10.1.45.5, to us
Jun 24 01:46:16.319: PIM(0): Join-list: (, 224.0.1.40), RPT-bit set, WC-bit set, S-bit set
Jun 24 01:46:16.323: PIM(0): Add Serial0/0.45/10.1.45.5 to (, 224.0.1.40), Forward state, by PIM *G Join
***Jun 24 01:46:16.323: PIM(0): Join-list: (*, 224.1.2.3), RPT-bit set, WC-bit set, S-bit set**
*Jun 24 01:46:16.323: PIM(0): Add Serial0/0.45/10.1.
R4#45.5 to (*, 224.1.2.3), Forward state, by PIM *G Join
*Jun 24 01:46:16.323: PIM(0): Building Triggered Join/Prune message for 224.1.2.3
Jun 24 01:46:16.323: PIM(0): Insert (,224.1.2.3) join in nbr 10.1.24.2's queue
*Jun 24 01:46:16.323: PIM(0): Building Join/Prune packet for nbr 10.1.24.2
*Jun 24 01:46:16.323: PIM(0): Adding v2 (1.1.1.3/32, 224.1.2.3), WC-bit, RPT-bit, S-bit Join
*Jun 24 01:46:16.323: PIM(0): Send v2 join/prune to 10.1.24.2 (Serial0/0.42)

```
R4#Sh ip mroute bi | s 224.1.2.3

(*, 224.1.2.3), 00:03:26/00:03:12, RP 1.1.1.3, flags: B
  Bidir-Upstream: Serial0/0.42, RPF nbr 10.1.24.2

  Outgoing interface list:
    Serial0/0.45, Forward/Sparse, 00:01:26/00:03:12
    Serial0/0.42, Bidir-Upstream/Sparse, 00:01:27/00:00:00
```

R4 has successfully become the primary path.

```
R4#Sh ip pim int s0/0.45 df

* implies this system is the DF
```

```
Interface                  RP                DF Winner        Metric     Uptime
Serial0/0.45               1.1.1.3           *10.1.45.4       129        00:01:42

R4#Sh ip pim int s0/0.42 df

* implies this system is the DF
Interface                  RP                DF Winner        Metric     Uptime
Serial0/0.42               1.1.1.3           10.1.24.2        65         00:02:08
```

The metric is only used to determine who the DF winner will be on the link between the routers. R4 will now win as the DF forwarder between itself an R5 as the S0/0.35 connection failed. The metric has now doubled for R4 and does represent the shortest path the RP R2 still remains as the DF winner.

On R1

```
R1#Ping 224.1.2.3

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.2.3, timeout is 2 seconds:

Reply to request 0 from 1.1.1.1, 1 ms
Reply to request 0 from 10.1.56.6, 84 ms
Reply to request 0 from 10.1.56.6, 84 ms
```

The DF election process is revertive as the following configuration reveals:

```
R4#Sh ip mroute bidirectional

(*, 224.1.2.3), 00:02:04/00:03:22, RP 1.1.1.3, flags: B
  Bidir-Upstream: Serial0/0.42, RPF nbr 10.1.24.2
  Outgoing interface list:
    Serial0/0.45, Forward/Sparse, 00:02:04/00:03:22
    Serial0/0.42, Bidir-Upstream/Sparse, 00:02:04/00:00:00
```

On R3

```
R3(config)#int s0/0.35
R3(config-subif)#no shut
```

On R4

```
R4#Sh ip mroute bibirectional | s 224.1.2.3

(*, 224.1.2.3), 00:02:40/00:02:47, RP 1.1.1.3, flags: BP
  Bidir-Upstream: Serial0/0.45, RPF nbr 10.1.45.5
```

```
  Outgoing interface list:
    Serial0/0.45, Bidir-Upstream/Sparse, 00:00:15/00:02:54
```

The process is apparently able to revert to what it believes to be the primary. The higher RP ID would normally be chosen if we did not use the L0 of R3.

Task 5

Configure the appropriate router(s) so a bidirectional state is NOT created for unnecessary groups.

On R3

```
R3#Sh ip mroute bidirectional

(*, 224.1.2.3), 00:09:52/00:03:07, RP 1.1.1.3, flags: B
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.35, Forward/Sparse, 00:00:22/00:03:07
    Serial0/0.32, Forward/Sparse, 00:06:11/00:02:40

(*, 224.0.1.40), 00:22:21/00:03:07, RP 1.1.1.3, flags: BCL
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.35, Forward/Sparse, 00:00:22/00:03:07
    Serial0/0.32, Forward/Sparse, 00:18:49/00:02:57
    Loopback0, Forward/Sparse, 00:19:15/00:02:24
```

The 224.0.1.40 group should not be used for bidirectional forwarding.

On ALL Routers

```
RX(config)#access-list 1 deny 224.0.1.40
RX(config)#access-list 1 permit 224.0.0.0 15.255.255.255

RX(config)#ip pim rp-address 1.1.1.3 1  bidir
```

The access-list will allow the routers to allow Bidirectional forwarding for ALL groups except the 224.0.1.40 group.

On R3

```
R3#Sh ip mroute  | b interface

 Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.2.3), 00:02:00/00:03:27, RP 1.1.1.3, flags: B
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.32, Forward/Sparse, 00:01:31/00:02:57
    Serial0/0.35, Forward/Sparse, 00:02:00/00:03:27

(*, 224.0.1.40), 00:01:10/00:02:06, RP 0.0.0.0, flags: DCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Loopback0, Forward/Sparse, 00:01:10/00:02:06
```

A simple show still shows that the Auto-RP announce message is present but does not show any bidirectional interfaces.

```
R3#Sh ip mroute bidirectional

(*, 224.1.2.3), 00:00:59/00:03:29, RP 1.1.1.3, flags: B
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.32, Forward/Sparse, 00:00:30/00:02:59
    Serial0/0.35, Forward/Sparse, 00:00:59/00:03:29
```

Task 6

Add an additional group of 224.2.2.2 on R1 and R6 to be joined. Use R4 for the 224.1.2.3 and R3 for the 224.2.2.2 group.

On ALL Routers

```
Rx(config)#NO ip pim rp-address 1.1.1.3 1 bidir
Rx(config)#NO access-list 1

Rx(config)#ip pim rp-address 1.1.1.3 1 bidir

Rx(config)#access-list 1 deny 224.0.1.40
Rx(config)#access-list 1 permit 224.2.2.2
```

```
Rx(config)#ip pim rp-address 1.1.1.4 2 bidir

Rx(config)#access-list 2 deny 224.0.1.40
Rx(config)#access-list 2 permit 224.1.2.3
```

On R1 and R6

```
Rx(config)#interface lo0
Rx(config-if)#ip igmp join-group 224.2.2.2
```

On R5

```
R5#Sh ip mroute bi

(*, 224.2.2.2), 00:00:03/00:03:26, RP 1.1.1.3, flags: B
  Bidir-Upstream: Serial0/0.53, RPF nbr 10.1.35.3
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:00:03/00:03:26
    Serial0/0.53, Bidir-Upstream/Sparse, 00:00:03/00:00:00

(*, 224.1.2.3), 00:01:23/00:03:06, RP 1.1.1.4, flags: B
  Bidir-Upstream: Serial0/0.54, RPF nbr 10.1.45.4
  Outgoing interface list:
    FastEthernet0/0, Forward/Sparse, 00:01:23/00:03:06
    Serial0/0.54, Bidir-Upstream/Sparse, 00:01:23/00:00:00
```

The groups diverge paths on R5 due to the access-list.

On R4

```
R4#Sh ip mroute bidirectional

(*, 224.1.2.3), 00:01:47/00:02:46, RP 1.1.1.4, flags: B
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:

    Serial0/0.45, Forward/Sparse, 00:01:43/00:02:46
    Serial0/0.42, Forward/Sparse, 00:01:47/00:02:36
```

On R3

```
R3#Sh ip mroute bidirectional

(*, 224.2.2.2), 00:01:33/00:02:54, RP 1.1.1.3, flags: B
```

```
  Bidir-Upstream: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0/0.35, Forward/Sparse, 00:00:43/00:02:46
    Serial0/0.32, Forward/Sparse, 00:01:33/00:02:54
```

On R6

```
R6#MTrace 1.1.1.1 224.1.2.3

Type escape sequence to abort.
Mtrace from 1.1.1.1 to 10.1.56.6 via group 224.1.2.3
From source (?) to destination (?)
Querying full reverse path...
 0  10.1.56.6
-1  10.1.56.6 PIM  [1.1.1.1/32]
-2  10.1.56.5 PIM  [1.1.1.1/32]
-3  10.1.45.4 PIM Reached RP/Core [1.1.1.1/32]

R6#MTrace 1.1.1.1 224.2.2.2

Type escape sequence to abort.
Mtrace from 1.1.1.1 to 10.1.56.6 via group 224.2.2.2
From source (?) to destination (?)
Querying full reverse path...
 0  10.1.56.6
-1  10.1.56.6 PIM  [1.1.1.1/32]
-2  10.1.56.5 PIM  [1.1.1.1/32]
-3  10.1.35.3 PIM Reached RP/Core [1.1.1.1/32]
```

The mtrace from R6 verify's that the paths are separate and correct.

```
R1#Ping 224.1.2.3

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.1.2.3, timeout is 2 seconds:

Reply to request 0 from 1.1.1.1, 1 ms
Reply to request 0 from 10.1.56.6, 112 ms
Reply to request 0 from 10.1.56.6, 60 ms

R1#Ping 224.2.2.2

Type escape sequence to abort.
Sending 1, 100-byte ICMP Echos to 224.2.2.2, timeout is 2 seconds:

Reply to request 0 from 1.1.1.1, 1 ms
```

```
Reply to request 0 from 10.1.56.6, 76 ms
Reply to request 0 from 10.1.56.6, 32 ms
```

Task 7

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Advanced
CCIE SERVICE PROVIDER
v3.0

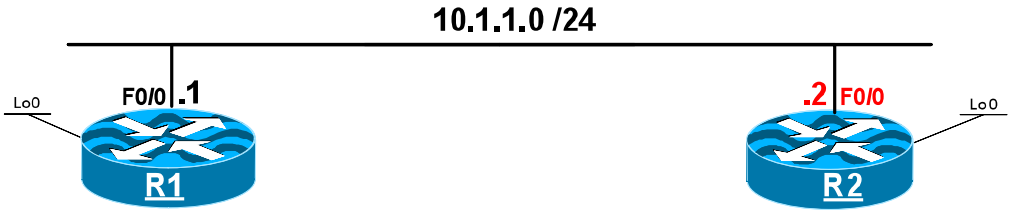
www.MicronicsTraining.com

Narbik Kocharians
CCIE #12410
R&S, Security, SP

Paul Negron
CCIE #14856
SP

Security

Lab 1 – Basic Router Security I



Lab Setup

- Ensure that R1 and R2's F0/0 interface is configured in VLAN 100.
- Assign the IP addressing using the chart below.

Router	Interface	IP address
R1	Lo0 F0/0	1.1.1.1 /8 10.1.1.1 /24
R2	**Lo0** **F0/0**	**2.2.2.2 /8** **10.1.1.2 /24**

Task 1

Ensure that R1 is configured with the following policy:

- Configure this router such that the user needs to enter a password of "Cisco07" before the user is allowed access to the privilege mode.
- The password **must** be at least seven characters in length
- Use the strongest hashing method based on MD5
- No recovery of the passwords are allowed on these routers

On R1

```
R1(config)#security passwords min-length 7
R1(config)#No service password-recovery

R1(config)#enable secret Cisco

% Password too short - must be at least 7 characters. Password configuration failed

Note above message is saying that the length of the password must be 7 characters.

R1(config)#enable secret Cisco07

Note the "Security passwords min-length 7" command effects all the passwords on this router.
```

Task 2

Ensure that all the passwords on both routers are unreadable if a "Show run" command is issued

On Both Routers

```
(config)#Service password-encryption

Note a "NO service password-encryption" command will not decrypt the passwords, but all the passwords configured after this command is entered will be displayed in the running configuration as unencrypted.
```

Task 3

Ensure that if the console port of any of these routers stops functioning, the administrator has another way to connect to the routers locally. DO NOT configure Telnet or modems to accomplish this task. The password for this task should be "Cisc@??07"

On Both Routers

```
(config)#line aux 0
(config-line)#login
(config-line)#password Cisc@??07
```

Note in order to configure a (?) in the password you must enter the "Esc and then Q" before typing each "?", in this task, the "Esc and then Q" needs to be done twice one for each "?", but to enter the password when a login attempt is made, you MUST NOT use the "Esc and then Q" for entering the "?".

Task 4

On R1, configure the number of allowable unsuccessful login attempts to 3 within one minute, if this policy is violated, the router should generate a syslog message.

On R1

```
R1(config)#security authentication failure rate 3 log
```

When the number of failed login attempts reaches the configured threshold, the router will send a "TOOMANY_AUTHFAILS" event message to the configured SYSLOG server. The router will also start a 15-second delay timer, once this delay timer expires, the user can try to login again.

Task 5

Configure both Routers to terminate an unattended console connection after 4 minutes and 30 seconds

On Both Routers

```
(config)#line con 0
(config-line)#exec-timeout 4 30
```

Note that a command "exec-timeout 0 0" should be used to disable console timeout at all.

Task 6

Create the following users on R2 using the chart below and ensure that ONLY the assigned commands are available to these users via the console:

User name	Password	Available commands:
U2	Cisco2	Show interface F0/0, Show ip int brief, ping and Traceroute.
U3	Cisco3	All the User level commands, The user Should be able to assign an IP addresses and Shut and no shut the interfaces.
Admin	Cisco	All levels and commands.

On R2

To configure the privilege level for User U2:

```
R2(config)#privilege exec level 2 show interface f0/0
R2(config)#privilege exec level 2 show ip int brie
R2(config)#privilege exec level 2 ping
R2(config)#privilege exec level 2 traceroute

R2(config)#username u2 privilege 2 password cisco2
```

The following reveals that user U2, has access to all privilege level 1 and 2 commands:

```
R2#Sh privilege
```

Current privilege level is 2

```
R2#Sh ip route | inc 10.1.1.0

C       10.1.1.0 is directly connected, FastEthernet0/0
R2#
```

To configure the privilege level for User U3:

```
R2(config)#privilege exec level 3 configure terminal
R2(config)#privilege configure level 3 interface
R2(config)#privilege interface level 3 shutdown
R2(config)#privilege interface level 3 ip address

R2(config)#username u3 privilege 3 password cisco3
```

To configure the privilege level for User Admin:

```
R2(config)#username admin privilege 15 password cisco
```

Lastly force the users to login using the local user account database:

```
R2(config)#line con 0
R2(config-line)#login local
```

Task 7

Configure a message of the day banner on R1 using the following policy:

The banner should state the router name, line and it should state that you are connected to www.MicronicsTraining.com

On R1

```
R1(config)#Banner motd #you are connected to $(hostname) on line $(line)
on domain $(domain)#

R1(config)#ip domain-name WWW.MicronicsTraining.com
```

To verify the configuration:

On R1

```
R1#logout
```

You should see the following message on the console:

"You are connected to R1 on line 0 on domain www.MicronicsTraining.com"

Task 8

Router R1 should be configured such that when user U2 logs in to the router the following menu is displayed. The following policy must be configured for the menu: The screen must be cleared prior in displaying the menu, the menu should allow the user to type the desired number and then press the enter key to select that option, option 3 should logout the user whereas, option 4 should exit the menu and into privilege exec mode (enable mode)

This is the menu for CCIE candidates

1 Display all the interfaces and their assigned IP addresses
2 Display the configuration of F0/0 interface
3 Logout
4 Exit out of the menu

Please Choose an option and press Enter :

On R1

```
R1(config)#menu u2 title # this is the menu for ccie candidates #

R1(config)#menu u2 text 1 display the interfaces and their assigned ip addresses
R1(config)#menu u2 command 1 show ip int brie
R1(config)#menu u2 options 1 pause
```

If the pause is not configured, the user will not see the output.

```
R1(config)#menu u2 text 2 display the configuration of f0/0 interface
R1(config)#menu u2 command 2 show run int f0/0

R1(config)#menu u2 options 2 pause

R1(config)#menu u2 text 3 logout
R1(config)#menu u2 command 3 logout

R1(config)#menu u2 text 4 exit out of the menu
R1(config)#menu u2 command 4 menu-exit

R1(config)#menu u2 clear-screen
R1(config)#menu u2 line-mode
R1(config)#menu u2 prompt # please choose an option and press enter :  #

R1(config)#username u2 privilege 15 password user2009
R1(config)#username u2 autocommand menu u2
R1(config)#username admin password Cisco2009
```

Always create another user so you will not lock yourself out of the con port, once tested you can always remove that username.

```
R1(config)#line con 0
R1(config-line)#login local
```

Task 9

Configure R2 using the following policy:

- Disable the service that when a wrong command is entered the router performs a broadcast looking for a DNS server to resolve the bad command to an IP address. This search should be restricted to 10.1.1.5 IP address.

- Disable the service that can be used to find out which users are logged into a network device. Although this information is considered not sensitive by many administrators, the information can be used by a hacker for reconnaissance attack.

- Disable the service that instructs an end node to use another and more efficient path to a particular destination.

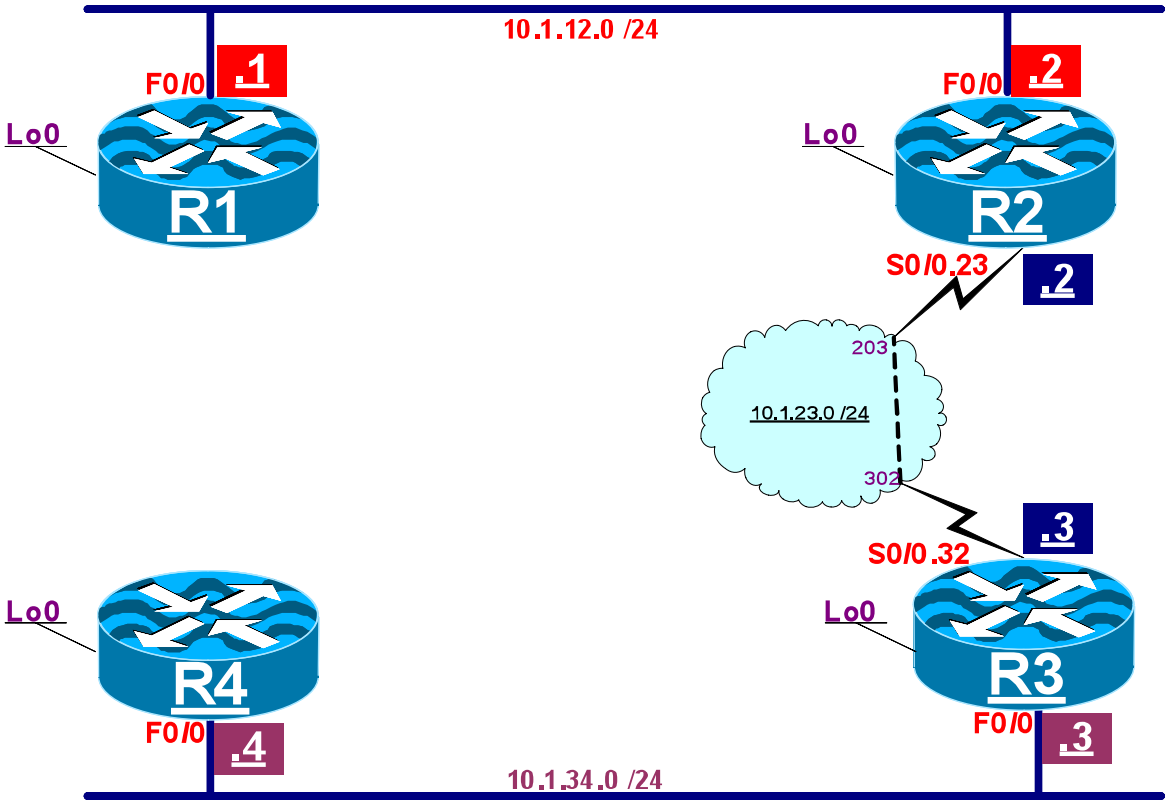
On R2

```
R2(config)#ip name-server 10.1.1.5
R2(config)#No ip finger
R2(config)#No ip icmp redirect
```

Task 10

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 2 – Standard Named Access List



Lab Setup:

- R1 and R2's F0/0 interface should be configured in VLAN 100.
- R2 and R3's S0/0 interface should be configured in a frame-Relay point-to-point manner.
- R3 and R4's F0/0 interface should be configured in VLAN 200.
- Configure Telnet on all routers using "TST" as the username and password.
- Run RIPv2 on the routers and advertise their directly connected networks.

IP Addressing:

Router	Interface	IP address
R1	Lo0 F0/0	1.1.1.1 /24 10.1.12.1 /24
R2	**Lo0** **F0/0** **S0/0.23**	**2.2.2.2 /24** **10.1.12.2 /24** **10.1.23.2 /24**
R3	Lo0 F0/0 S0/0.32	3.3.3.3 /24 10.1.34.3 /24 10.1.23.3 /24
R4	**Lo0** **F0/0**	**4.4.4.4 /24** **10.1.34.4 /24**

Task 1

Configure a standard numbered access-list on R1 to block host 4.4.4.4 from accessing any of its interfaces.

On R1

```
R1(config)#access-list 1 deny 4.4.4.4
R1(config)#access-list 1 permit any

R1(config)#int f0/0
R1(config-if)#ip access-group 1 in
```

To test and verify the configuration:

On R1

```
R1#Debug ip packet
```

On R4

```
R4#Ping 1.1.1.1 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 4.4.4.4
U.U.U
Success rate is 0 percent (0/5)
```

You should see the following debug output on R1:

```
IP: s=4.4.4.4 (FastEthernet0/0), d=1.1.1.1, len 100, access denied
IP: tableid=0, s=10.1.12.1 (local), d=4.4.4.4 (FastEthernet0/0), routed via FIB
IP: s=10.1.12.1 (local), d=4.4.4.4 (FastEthernet0/0), len 56, sending
```

Note the ping fails.

On R1

```
R1#Undebug all
```

Task 2

Replace the previous task using a standard named access-list.

On R1

```
R1(config)#NO access-list 1

R1(config)#ip access-list standard TEST
R1(config-std-nacl)#deny host 4.4.4.4
R1(config-std-nacl)#permit any

R1(config)#int f0/0
R1(config-if)#ip access-group TEST in
```

Task 3

Remove the access-list from the previous step before proceeding to the next Lab.

On R1

```
R1(config)#NO ip access-list standard TEST

R1(config)#int f0/0
R1(config-if)#NO ip access-group TEST in
```

To verify the configuration:

On R4

```
R4#Ping 1.1.1.1 sou lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 4.4.4.4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/61 ms
```

DO NOT erase the startup config or reload the routers.

Lab 3 – Controlling Telnet access and Configuring SSH

Based on the previous Lab's IP addressing, topology and Lab setup

Task 1

Configure an access-list on R2 such that only 1.1.1.1 /24 is allowed to telnet in, other packets should NOT be subject to inspection by this access-list but they should be denied telnet access.

On R2

Note the last line in the following access-list is NOT needed to accomplish this task, but it can be used as a tool to troubleshoot problems, or verify tests:

```
R2(config)#access-list 1 permit host 1.1.1.1
R2(config)#access-list 1 deny any log

R2(config)#line vty 0 ?
R2(config-line)#access-class 1 in
```

The reason a question mark was used in the "line vty 0 ?" command is that different routers with different IOSes have different number of Telnet ports. Most switches have 15 to 181 VTYs by default.

On R1

```
R1#Telnet 10.1.12.2

Trying 10.1.12.2 ...
% Connection refused by remote host

R1#Telnet 10.1.12.2 /source-interface lo0

Trying 10.1.12.2 ... Open
User Access Verification

Password:
```

Note the first time R1 Telnet, it used 10.1.12.1 as the source IP address and the Telnet was denied by R2. The second Telnet was successful, because the source IP address of the Telnet was changed to 1.1.1.1.

To verify the configuration on R2

```
R2#Sh access-list 1

Standard IP access list 1
    10 permit 1.1.1.1 (2 matches)
    20 deny   any log (1 match)
```

On R3

```
R3#Telnet 10.1.23.2

Trying 10.1.23.2 ...

% Connection refused by remote host

R3#Telnet 10.1.23.2 /source-interface lo0

Trying 10.1.23.2 ...

% Connection refused by remote host
```

Note R3 is NOT successful.

On R4

```
R4#Telnet 10.1.23.2

Trying 10.1.23.2 ...

% Connection refused by remote host

R4#Telnet 2.2.2.2

Trying 2.2.2.2 ...

% Connection refused by remote host

R4#Telnet 2.2.2.2 /source-interface Lo0

Trying 2.2.2.2 ...

% Connection refused by remote host
```

Note R4 is NOT successful.

Task 2

Configure R2 such that when host 1.1.1.1 /24 Telnets to any of its interfaces, from the telnet session, this host can only telnet to R4 and no other router. This access-list should not be applied to any of the routers interface/s.

Before configuring the task, you should test to ensure that the telnet works properly.

On R1

```
R1#Telnet 10.1.12.2 /source-interface lo0

Trying 10.1.12.2 ... Open

User Access Verification

Username:
Password:
```

Enter "TST" for the username and password to Login.

Note you are Telnetted to R2:

```
R2>

R2>Telnet 10.1.23.3

Trying 10.1.23.3 ... Open

User Access Verification

Password:
```

Note R1 can successfully Telnet into R3 and R4 through R2

```
R2>Telnet 10.1.34.4

Trying 10.1.34.4 ... Open

User Access Verification

Password:
```

Enter the following commands to configure the task:

```
R2(config)#line vty 0 181
R2(config-line)#access-class 2 out
```

```
R2(config)#access-list 2 permit host 4.4.4.4
```

On R1

Note R1 has successfully Telnetted to R2

```
R1#Telnet 10.1.12.2 /source-interface lo0

Trying 10.1.12.2 ... Open

User Access Verification

Password:
```

Note R1 can NOT Telnet to R3

```
R2>Telnet 3.3.3.3

Trying 3.3.3.3 ...

% Connections to that host not permitted from this terminal
```

Note R1 can ONLY Telnet to R4

```
R2>Telnet 4.4.4.4
Trying 4.4.4.4 ... Open

User Access Verification

Username:
Password:

R4>quit

[Connection to 4.4.4.4 closed by foreign host]

R2>quit

[Connection to 10.1.12.2 closed by foreign host]
```

Note that this solution works only for hosts that have already telnetted to R2 and are trying to establish a Telnet connections originating from R2 VTYs.

It does not work for regular telnet originated from R2 (i.e. where user is connected via console and issues a telnet command).

Task 3

Configure SSH on R4 using the following policy:

- **Domain name: MicronicsTraining.com**
- **Key: 512 bit**
- **Authentication: The authentication should be performed based on the local database.**
- **Username: User1**
- **Password: Cisco**
- **Authentication Ports:**
  Local authentication should be configured on VTY.
  No authentication should be done on the AUX or the CON lines.

- **You should only allow SSH connection to the VTY lines.**

SSH (Secure Shell) is a protocol that enables an SSH client to make a secure and encrypted connection to Cisco Devices.

On R4

The following command configures a host domain for this router:

```
R4(config)#ip domain name MicronicsTraining.com
```

The following command generates an RSA key pair for your router, which automatically enables SSH; the last line of this message states that. Remember if you need to delete the RSA key pair, use the "Crypto key zeroize rsa" command, once the key pair is deleted, SSH is automatically disabled.

```
R4(config)#crypto key generate rsa usage-keys
```

Once the above command is entered, the following message should appear on the console:

The name for the keys will be R4.MicronicsTraining.com Choose the size of the key modulus in the range of 360 to 2048 for your Signature Keys. Choosing a key modulus greater than 512 may take a few minutes.

Press Enter to accept the 512

How many bits in the modulus [512]:
Choose the size of the key modulus in the range of 360 to 2048 for your Encryption Keys. Choosing a key modulus greater than 512 may take a few minutes.

Press Enter to accept the 512

How many bits in the modulus [512]:
% Generating 512 bit RSA keys...[OK]
% Generating 512 bit RSA keys...[OK]

Note that in order to use SSHv2 a key of at least 768 bits must be generated.

%SSH-5-ENABLED: SSH 1.5 has been enabled

The following command enables the AAA services:

```
R4(config)#aaa new-model
```

To create the requested username and password:

```
R4(config)#username user1 password Cisco
```

The following command is configured so it can be applied to all VTY ports.

```
R4(config)#aaa authentication login LOCAL-AUTH local
```

The following commands will apply the "LOCAL-AUTH" policy to the vty ports:

```
R4(config)#line vty 0 181
R4(config-line)#login authentication LOCAL-AUTH
R4(config-line)#transport input ssh
```

Note that there is no need to create "no authentication" policy for AUX and CON. This is because a named authentication list is used and it is only applied to the VTY ports.

To test the configuration:

On R3

```
R3#ssh -l user1 -c 3des 10.1.34.4
```

The following is asking you for the password :

Password :  **You should use "Cisco" and NOT "TST"; "TST" will NOT work**

```
R4>exit
```

[Connection to 10.1.34.4 closed by foreign host]

R3#

Task 4

Remove the configuration commands from the previous four steps before proceeding to the next lab.

Lab 4 – Extended Access List
IP and ICMP

Based on lab 2's IP addressing, topology and Lab setup

Task 1

Deny communication between hosts 1.1.1.1 and 4.4.4.4. This policy must be configured on R2. If these hosts attempt to communicate, their packets should not reach their destination.

On R2

```
R2(config)#access-list 100 deny ip host 4.4.4.4 host 1.1.1.1
R2(config)#access-list 100 permit ip any any

R2(config)#access-list 101 deny ip host 1.1.1.1 host 4.4.4.4
R2(config)#access-list 101 permit ip any any

R2(config)#int s0/0.23
R2(config-subif)#ip access-group 100 in

R2(config)#int f0/0
R2(config-subif)#ip access-group 101 in
```

To Test the configuration:

On R1

```
R1#Ping 2.2.2.2 source 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R1#Ping 3.3.3.3 source 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
```

```
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms

R1#Ping 4.4.4.4 source 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
U.U.U
Success rate is 0 percent (0/5)

R1#Sh ip route rip | inc r

R       2.2.2.0 [120/1] via 10.1.12.2, 00:00:04, FastEthernet0/0
R       3.3.3.0 [120/2] via 10.1.12.2, 00:00:04, FastEthernet0/0
R       4.4.4.0 [120/3] via 10.1.12.2, 00:00:04, FastEthernet0/0
R       10.1.23.0 [120/1] via 10.1.12.2, 00:00:04, FastEthernet0/0
R       10.1.34.0 [120/2] via 10.1.12.2, 00:00:04, FastEthernet0/0
```

Note even though network 4.4.4.0/24 is in R1's routing table, R1 cannot ping 4.4.4.4; because of the Inbound access-list 101 configured on R2's F0/0 interface, host 4.4.4.4 will not be able to communicate with 1.1.1.1.

On R4

```
R4#Ping 3.3.3.3 source 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 4.4.4.4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R4#Ping 2.2.2.2 source 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 4.4.4.4
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

R4#Ping 1.1.1.1 source 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 4.4.4.4
```

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

R4#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
```

Note R4 can ping any IP address except 1.1.1.1 IP address using its Lo0 as the source, this is because of the Inbound access-list configured on R2's S0/0.23 interface.

Task 2

Configure R1 based on the following policy:

- R1 should successfully be able to ping R2 and receive the replies, but R2 should not be able to ping R1.

On R1

Note in the following access-list the "ICMP ECHO" messages are denied sourcing from R2's directly connected IP addresses, whereas, everything else is permitted:

```
R1(config)#access-list 100 deny icmp host 10.1.12.2 any echo
R1(config)#access-list 100 deny icmp host 2.2.2.2 any echo
R1(config)#access-list 100 deny icmp host 10.1.23.2 any echo
R1(config)#access-list 100 permit ip any any

R1(config)#int f0/0
R1(config-if)#ip access-group 100 in
```

To Test the configuration:

On R1

```
R1#Ping 10.1.12.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.2, timeout is 2 seconds:
```

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

R1#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 = 1/2/4 ms

R1#Ping 10.1.23.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Note R1 can successfully ping every IP address on R2. The following reveals that R2 can NOT ping R1 because ICMP Echo messages are denied:

On R2

```
R2#Ping 10.1.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

R2#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

Note R2 cannot ping any IP address configured on R1.

On R3

```
R3#Ping 10.1.12.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
```

Note R3 can successfully ping any IP address configured on R1

On R4

```
R4#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms
```

Note R4 can also successfully ping any IP address configured on R1

Task 3

R3 should be configured such that if R2 pings a host that is not reachable, R3 does NOT send "***ICMP host unreachable***" messages back to R2

Before configuring this task, you should test to see the host unreachable messages. In order to test you must configure the following on R2:

On R2

```
R2(config)#ip route 0.0.0.0 0.0.0.0 10.1.23.3
```

The above static route is needed on R2 so R2 uses R3 for any destination/s that it is not aware of.

```
R2#Debug ip icmp
```

ICMP packet debugging is on

```
R2#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
U.U.U
ICMP: dst (10.1.23.2) host unreachable rcv from 10.1.23.3.U
ICMP: dst (10.1.23.2) host unreachable rcv from 10.1.23.3.U
Success rate is 0 percent (0/5)
```

ICMP: dst (10.1.23.2) **host unreachable rcv from 10.1.23.3**

Note the default route was needed for testing this scenario. Once 5.5.5.5 is pinged, the local router (R2) will perform a route table lookup and the closest match to that destination is the default gateway, which is pointing to R3. Since R3 is not aware of this IP address, it sends "icmp host unreachable" messages back to the source (R2).

To configure this task:

On R3

```
R3(config)#int s0/0.32
R3(config-subif)#No ip unreachables
```

To test the configuration:

On R2

```
R2#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

NOTE ONLY "." Are coming back and NOT "U" also you should NOT see unreachables in the output Of the debug

Note R2 is no longer receiving the ICMP unreachables.

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 5 – Extended Access List
OSPF & Eigrp

Based on Lab 2's IP addressing and Topology

Lab Setup:

- Configure the routers according to LAB 2's IP addressing and topology

Task 1

Configure the routers using the following policies:

- Configure OSPF Area 0 and EIGRP 100 on all routers and advertise every interface in both routing protocols.
- Ensure that "**no auto-summary**" command is used when configuring Eigrp 100.
- In OSPF routing protocol, the loopback interfaces should be advertised with their correct mask.
- Remove RIPv2 configuration from all four routers

On All Routers:

```
Rx(config)#int lo0
Rx(config-if)#ip ospf net point-to-point

Rx(config)#NO router rip

Rx(config)#router ospf 1
Rx(config-router)#netw 0.0.0.0 0.0.0.0 are 0

Rx(config)#router eigrp 100
Rx(config-router)#NO au
Rx(config-router)#netw 0.0.0.0 0.0.0.0
```

To verify the configuration:

On R1

```
R1#Show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time     Address         Interface
2.2.2.2         1     FULL/DR         00:00:31      10.1.12.2       FastEthernet0/0

R5#Show ip eigrp neighbors

IP-EIGRP neighbors for process 100
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   10.1.12.2               Fa0/0             14 00:00:09    1    200  0  39
```

Task 2

Configure an access-list on R1 to block Eigrp routing protocol and allow the rest of the IP protocol stack, if this configuration is performed successfully router R1 should only have OSPF routes in its routing table.

Before you configure this task, you should display the existing routing table:

On R1

```
R1#Sh ip route | B Gateway

Gateway of last resort is not set

     1.0.0.0/24 is subnetted, 1 subnets
C       1.1.1.0 is directly connected, Loopback0
     2.0.0.0/24 is subnetted, 1 subnets
D       2.2.2.0 [90/156160] via 10.1.12.2, 00:06:01, FastEthernet0/0
     3.0.0.0/24 is subnetted, 1 subnets
D       3.3.3.0 [90/2300416] via 10.1.12.2, 00:06:01, FastEthernet0/0
     4.0.0.0/24 is subnetted, 1 subnets
D       4.4.4.0 [90/2302976] via 10.1.12.2, 00:06:01, FastEthernet0/0
     10.0.0.0/24 is subnetted, 3 subnets
C       10.1.12.0 is directly connected, FastEthernet0/0
D       10.1.23.0 [90/2172416] via 10.1.12.2, 00:06:01, FastEthernet0/0
D       10.1.34.0 [90/2174976] via 10.1.12.2, 00:06:01, FastEthernet0/0
```

Note you only see the Eigrp advertised routes because Eigrp has a lower administrative distance, 90 versus OSPF, which has an AD of 110.

Enter the following commands to configure this task:

On R1

```
R1(config)#access-list 100 deny eigrp any any
R1(config)#access-list 100 permit ip any any

R1(config)#int f0/0
R1(config-if)#ip access-group 100 in
```

To verify the configuration:

Note you should receive the following message within 15 seconds:

%DUAL-5-NBRCHANGE: IP-EIGRP(0) 100: Neighbor 10.1.12.2 (FastEthernet0/0) is down: holding time expired

```
R1#Show ip route | b gateway

Gateway of last resort is not set

     1.0.0.0/24 is subnetted, 1 subnets
C       1.1.1.0 is directly connected, Loopback0
     2.0.0.0/24 is subnetted, 1 subnets
O       2.2.2.0 [110/2] via 10.1.12.2, 00:01:13, FastEthernet0/0
     3.0.0.0/24 is subnetted, 1 subnets
O       3.3.3.0 [110/66] via 10.1.12.2, 00:01:13, FastEthernet0/0
     4.0.0.0/24 is subnetted, 1 subnets
O       4.4.4.0 [110/67] via 10.1.12.2, 00:01:13, FastEthernet0/0
     10.0.0.0/24 is subnetted, 3 subnets
C       10.1.12.0 is directly connected, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:01:13, FastEthernet0/0
O       10.1.34.0 [110/66] via 10.1.12.2, 00:01:13, FastEthernet0/0
```

Note when the access-list is configured, the Eigrp neighbor adjacency between R1 and R2 fails, and OSPF is allowed in the routing table.

Task 3

Remove the configuration command from the previous step (Task 1).

On R1

```
R1(config)#NO access-list 100

R1(config)#int f0/0
R1(config-if)#NO ip access-group 100 in
```

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 6 – Using MQC as a filtering tool

Based on Lab 2's IP addressing and Topology

Lab Setup:

- Configure the routers according to LAB 2's IP addressing and topology

Task 1

Configure OSPF Area 0 on all directly connected interfaces on all four routers; ensure that loopback 0 interface of these routers are advertised with their correct mask. You should remove RIPv2 from all four routers.

On All Routers:

```
Rx(config)#NO router rip

Rx(config)#int loopback 0
Rx(config-if)#ip ospf network point-to-point

Rx(config-if)#router ospf 1
Rx(config-router)#network 0.0.0.0 0.0.0.0 area 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf | inc O

O       2.2.2.0 [110/2] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       3.3.3.0 [110/66] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       4.4.4.0 [110/67] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       10.1.34.0 [110/66] via 10.1.12.2, 00:00:05, FastEthernet0/0
```

Task 2

Configure R1 to perform classification and marking. This router should mark all egress Telnet traffic with IP precedence of 1.

On R1

```
R1(config)#access-list 100 permit tcp any any eq 23

R1(config)#class-map TELNET
R1(config-cmap)#match access-group 100

R1(config)#policy-map TST
R1(config-pmap)#class TELNET
R1(config-pmap-c)#set ip precedence 1

R1(config-pmap-c)#int f0/0
R1(config-if)#service-policy output TST
```

To verify the configuration:

On R1

```
R1#Sh class-map TELNET

 Class Map match-all TELNET (id 2)
   Match access-group  100

R1#Sh policy-map TST

  Policy Map TST
    Class TELNET
      set ip precedence 1
```

To test the configuration:

On R1

To generate Telnet traffic:

```
R1#Telnet 10.1.34.4

Trying 10.1.34.4 ... Open
```

```
User Access Verification

Username:
Password:
```

To verify the marking of the traffic:

```
R1#Sh policy-map interface f0/0

 FastEthernet0/0

  Service-policy output: TST

    Class-map: TELNET (match-all)
      20 packets, 1218 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 100
      QoS Set
        precedence 1
          Packets marked 10

    Class-map: class-default (match-any)
      25 packets, 9618 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Task 3

Configure R3 to block all packets with IP Precedence 1 marking coming from S0/0.32 interface. You must use an MQC to accomplish this task.

On R3

```
R3(config)#class-map ip-prec
R3(config-cmap)#match ip precedence 1

R3(config)#policy-map TEST
R3(config-pmap)#class ip-prec
R3(config-pmap-c)#drop

R3(config-pmap)#int s0/0.32
R3(config-subif)#service-policy input TEST
```

To test the configuration:

```
R1#Telnet 10.1.34.4

Trying 10.1.34.4 ...

% Connection timed out; remote host not responding
```

Note the Telnet session failed.

```
R3#Show policy-map interface

 Serial0/0.32

  Service-policy input: TEST

    Class-map: IP-Prec (match-all)
      4 packets, 192 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip precedence 1
      drop

    Class-map: class-default (match-any)
      6 packets, 504 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 7 – Extended Access List With Established

Based on Lab 2s IP addressing and Topology

Lab Setup:

- Configure the routers according to LAB 2's IP addressing and topology

Task 1

Configure OSPF Area 0 on all directly connected interfaces on all four routers; ensure that loopback 0 interface of these routers are advertised with their correct mask. You should remove RIPv2 from all four routers.

On All Routers:

```
Rx(config)#NO router rip

Rx(config)#int loopback 0
Rx(config-if)#ip ospf network point-to-point

Rx(config-if)#router ospf 1
Rx(config-router)#network 0.0.0.0 0.0.0.0 area 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf | inc O

O       2.2.2.0 [110/2] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       3.3.3.0 [110/66] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       4.4.4.0 [110/67] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       10.1.34.0 [110/66] via 10.1.12.2, 00:00:05, FastEthernet0/0
```

Task 2

R1, R3 and R4 are offering Telnet and HTTP services. R1 and R2 are the routers in your company; your company's policy is as follows:

R2 is the border router that connects R1 to the other routers. R2 should be configured with an inbound access-list such that it ONLY allows traffic that was initiated locally and by R1 to be returned. Ensure that the appropriate traffic is allowed. No other traffic should be allowed in.

On R2

```
R2(config)#access-list 100 permit ospf any any
R2(config)#access-list 100 permit tcp any any established

R2(config)#int s0/0.23
R2(config-subif)#ip access-group 100 in
```

To test the configuration:

On R4

```
R4#Telnet 1.1.1.1

Trying 1.1.1.1 ...
% Destination unreachable; gateway or host down
```

Note the telnet originated by R4 was NOT successful

On R1

```
R1#Telnet 4.4.4.4

Trying 4.4.4.4 ... Open

User Access Verification

Username: TST
Password:

R4>
```

Note the responses from R4 gets in ONLY if the traffic was originated by R1.

On R2

```
R2#Sh access-list

Extended IP access list 100
    10 permit ospf any any (30 matches)
    20 permit tcp any any established (15 matches)
```

Remember the "Established" keyword only works on TCP based application.

Task 3

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 8 – Dynamic Access List

Based on Lab 2s IP addressing and Topology

Lab Setup:

- Configure the routers according to LAB 2's IP addressing and topology

Task 1

Configure OSPF Area 0 on all directly connected interfaces on all four routers; ensure that loopback 0 interface of these routers are advertised with their correct mask. You should remove RIPv2 from all four routers.

On All Routers:

```
Rx(config)#NO router rip

Rx(config)#int loopback 0
Rx(config-if)#ip ospf network point-to-point

Rx(config-if)#router ospf 1
Rx(config-router)#network 0.0.0.0 0.0.0.0 area 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf | inc O

O       2.2.2.0 [110/2] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       3.3.3.0 [110/66] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       4.4.4.0 [110/67] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       10.1.34.0 [110/66] via 10.1.12.2, 00:00:05, FastEthernet0/0
```

Task 2

Create the following users on R2:

First user name: U3, password: U3
Second user name: U4, password U4

On R2

```
R2(config)#username u3 password u3
R2(config)#username u4 password u4
```

Task 3

R1 and R2 belong to the same company; R2 is the border router connecting their company to the other routers belonging to another company. Create a dynamic access-list on R2 using the following policy:

- Only the authenticated users are allowed to have access to R1.
- Allow R3 and/or R4 to telnet into R2's S0/0.23 interface to get authenticated (Using the usernames created in the previous step) before they can access any of the services offered by R1.
- This policy should NOT affect OSPF.
- R1 or R2 should be able to have access to R3 and R4, without authentication.

On R2

The following access-list is required so R3 and R4 can telnet and gets authenticated. This second statement is required to allow OSPF through.

The third statement allows the return traffic in the network that was initiated by any of the users within the company (R1 and/or R2 in this case).

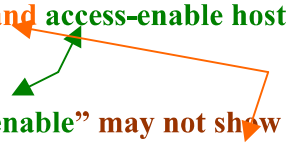
The forth statement in the following access-list tells the router to create a dynamic access-list called TEST.

This named access-list will be created when U3 and/or U4 telnet to this router and get authenticated.

```
R2(config)#access-list 100 permit tcp any host 10.1.23.2 eq 23
R2(config)#access-list 100 permit ospf any any
R2(config)#access-list 100 permit tcp any any established
R2(config)#access-list 100 dynamic TEST permit ip any any

R2(config)#int s0/0.23
R2(config-subif)#ip access-group 100 in
```

Lastly, the telnet ports must be configured for the dynamic access-list:

```
R2(config-subif)#line vty 0 181
R2(config-line)#autocommand access-enable host
R2(config-line)#login local
```



Note the keyword "access-enable" may not show when a question mark is entered, because this is a hidden command. The "autocommand" command links the dynamic access-list to the telnet authentication. It creates an entry in the dynamic access-list using the source IP address of the host. If the "autocommand" is NOT used, the dynamic entry will not be created.
The second line specifies that authentication should be done using the local user account database.

To Test the configuration:

On R4

```
R4#Show ip route ospf | inc O
```

Note the routing table contains all the networks.

```
O       1.1.1.0 [110/67] via 10.1.34.3, 00:28:49, FastEthernet0/0
O       2.2.2.0 [110/66] via 10.1.34.3, 00:28:49, FastEthernet0/0
O       3.3.3.0 [110/2] via 10.1.34.3, 00:28:49, FastEthernet0/0
O       10.1.12.0 [110/66] via 10.1.34.3, 00:28:49, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.34.3, 00:28:49, FastEthernet0/0

R4#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

The ping failed because R4 is not an authenticated user.
In the following steps, R4 will telnet to R2 in order to be authenticated. Note R4's telnet session is closed right after the authentication.

```
R4#Telnet 10.1.23.2

Trying 10.1.23.2 ... Open

User Access Verification

Username: U3
```

Entering the username U3

```
Password:
```

Entering the password "U3" to get authenticated

```
[Connection to 10.1.23.2 closed by foreign host]
```

Note once the user is authenticated, the telnet session is closed by 10.1.23.2.

To test connectivity:

On R4

```
R4#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

R4#Telnet 1.1.1.1

Trying 1.1.1.1 ... Open

User Access Verification

Password:
```

Note both Telnet and ping commands are successful after a successful authentication. One problem with dynamic access-lists is the fact that ONLY one dynamic statement is allowed.

On R2

```
R2#Show access-list

Extended IP access list 100

10 permit tcp any host 10.1.23.2 eq telnet (60 matches)
    20 permit ospf any any (77 matches)
    30 permit tcp any any established (10 matches)
```

```
  40 Dynamic TEST permit ip any any
       permit ip host 10.1.34.4 any (6 matches)
```

Task 4

Clear the dynamically created access-list entry from R2, do not remove and reenter the access-list to accomplish this task. Configure an idle timeout on R2 for the dynamic access-list such that if an authenticated user is idle for 2 minutes the entry is removed.

Note a password was Not entered, the Telnet session expired, but the dynamic access-list on R2 is still allowing IP to any source going to any destination.

```
R4#Telnet 1.1.1.1

Trying 1.1.1.1 ... Open

User Access Verification

Password:
```

[Connection to 1.1.1.1 closed by foreign host]

Check the dynamically created access-list:

On R2

```
R2#Show access-list

Extended IP access list 100
    10 permit tcp any host 10.1.23.2 eq telnet (60 matches)
    20 permit ospf any any (98 matches)
    30 permit tcp any any established (13 matches)
    40 Dynamic TEST permit ip any any
       permit ip host 10.1.34.4 any (6 matches)
```

To clear the dynamically created access-list:

On R2

```
R2#Clear ip access-template 100 TEST host 10.1.34.4 any
```

To verify the configuration:

On R2

```
R2#Show ip access-list

Extended IP access list 100
    10 permit tcp any host 10.1.23.2 eq telnet (60 matches)
    20 permit ospf any any (125 matches)
    30 permit tcp any any established (13 matches)
    40 Dynamic TEST permit ip any any
```

Note the dynamic entry is purged. Enter the following commands to setup the requested timeout value:

```
R2(config)#line vty 0 181
R2(config-line)#autocommand access-enable host timeout 2
```

The timeout here defines the idle timeout and it is in minutes.

To test the configuration:

On R4

```
R4#Telnet 10.1.23.2

Trying 10.1.23.2 ... Open

User Access Verification

Username: U4
Password:
```

[Connection to 10.1.23.2 closed by foreign host]

On R2

You may have to save the configuration and reload for the timeout to take effect.

```
R2#Sh access-list | b dynamic

     40 Dynamic TEST permit ip any any
       permit ip host 10.1.34.4 any (1 match) (time left 107)

R2#Sh access-list | b dynamic

   40 Dynamic TEST permit ip any any
      permit ip host 10.1.34.4 any (1 match) (time left 97)
```

```
R2#Sh access-list | b dynamic

   40 Dynamic TEST permit ip any any
       permit ip host 10.1.34.4 any (1 match) (time left 30)

R2#Sh access-list | b dynamic

   40 Dynamic TEST permit ip any any
       permit ip host 10.1.34.4 any (1 match) (time left 10)

R2#Sh access-list | b dynamic

   40 Dynamic TEST permit ip any any
       permit ip host 10.1.34.4 any (1 match) (time left 1)

R2#Sh access-list | b dynamic

   40 Dynamic TEST permit ip any any
       permit ip host 10.1.34.4 any (1 match) (time left 0)

R2#Sh access-list | b dynamic

   40 Dynamic TEST permit ip any any
       permit ip host 10.1.34.4 any (1 match)
```

Note the (time left) counter starts counting down from 120 seconds and it reached zero, and the entry is removed.

Task 5

Reconfigure this access-list such that the maximum time limit for each entry regardless of activity within this dynamic access-list is set to 3 minutes.

On R2

This command removes the access-list.

```
R2(config)#NO access-list 100
```

The following command removes the "autocommand".

```
R2(config)#line vty 0 181
R2(config-line)#NO autocommand access-enable host timeout 2
```

```
R2(config-line)#autocommand access-enable host

R2(config)#access-list 100 permit tcp any host 10.1.23.2 eq 23
R2(config)#access-list 100 permit ospf any any
R2(config)#access-list 100 permit tcp any any established
R2(config)#access-list 100 dynamic TEST timeout 3 permit ip any any
```

This timeout is the absolute or time to live timeout, which defines the amount of time in minutes a dynamically created access-list, can exist.

Since the access-list is already applied to the S0/0.23 subinterface of R2, there is no need to re-apply the access-list.

Task 6

After configuring the dynamic access-list, you realized that from time to time the administrator needs to telnet into R2 for trouble shooting and management purposes. Reconfigure R2 such that the administrators can telnet into this router successfully to perform their day-to-day management tasks.

On R2

```
R2(config)#line vty 0 181
R2(config-line)#NO autocommand access-enable host
```

In the following configuration, all VTY ports are configured except one:

```
R2(config)#line vty 0 180
R2(config-line)#autocommand access-enable host
R2(config-line)#login local
```

Note only one session is reserved for administrative purposes, more ports can be used for this purpose. The "rotary 5" command allows Telnet access to port 3005 instead of 23, this port number should be allowed in the access-list and therefore, it

must be added to the existing access-list.

```
R2(config)#line vty 181
R2(config-line)#login local
R2(config-line)#rotary 5
```

Adding the access-list statement for port 3005:

```
R2(config)#access-list 100 permit tcp any any eq 3005
```

Remember that "autocommand" can also be used for a specific user configuration by configuring "username... autocommand access-enable host" therefore; another solution in configuring this task is to move the autocommand to the other users and remove it from the vty line.
See that telnet to R1 loopback fails without authentication against R2.

Note that telnet to the port 3005 is successful and it allows administrator to issue commands on R2.

```
R4#Telnet 1.1.1.1
```

Note Telnet is NOT successful:

```
Trying 1.1.1.1 ...
% Destination unreachable; gateway or host down

R4#Telnet 10.1.23.2
```

Note R4 is authenticated successfully:

```
Trying 10.1.23.2 ... Open

User Access Verification

Username: U3
Password:
[Connection to 10.1.23.2 closed by foreign host]

R4#
```

Note after successful authentication R4's prompt is Back to privileged mode.

```
R4#Telnet 10.1.23.2 3005

Trying 10.1.23.2, 3005 ... Open

User Access Verification

Username: U4
Password:
```

Note the prompt

```
R2>
```

Task 7

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 9 – Reflexive Access-lists

Based on Lab 2s IP addressing and Topology

Lab Setup:

- Configure the routers according to LAB 2's IP addressing and topology

Task 1

Configure OSPF Area 0 on all directly connected interfaces on all four routers; ensure that loopback 0 interface of these routers are advertised with their correct mask. You should remove RIPv2 from all four routers.

On All Routers:

```
Rx(config)#NO router rip

Rx(config)#int loopback 0
Rx(config-if)#ip ospf network point-to-point

Rx(config-if)#router ospf 1
Rx(config-router)#network 0.0.0.0 0.0.0.0 area 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf | inc O

O       2.2.2.0 [110/2] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       3.3.3.0 [110/66] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       4.4.4.0 [110/67] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       10.1.34.0 [110/66] via 10.1.12.2, 00:00:05, FastEthernet0/0
```

Task 2

R1 and R2 belong to company-A. R3 and R4 belong to company-B. R2 is the border router that connects these companies to each other; R2 should be configured such that it allows the return traffic for the following protocols:

- R2 should allow the return HTTP, TFTP and Telnet traffic that is originated locally or by R1.
- R2 should allow the OSPF traffic into the network.

On R2

```
R2(config)#ip access-list extended outbound
R2(config-ext-nacl)#permit tcp any any eq 80 reflect TEST
R2(config-ext-nacl)#permit tcp any any eq 23 reflect TEST
R2(config-ext-nacl)#permit udp any any eq 69 reflect TEST
R2(config-ext-nacl)#permit ospf any any

R2(config)#ip access-list extended inbound
R2(config-ext-nacl)#permit ospf any any
R2(config-ext-nacl)#evaluate TEST

R2(config)#int s0/0.23
R2(config-subif)#ip access-group inbound in
R2(config-subif)#ip access-group outbound out
```

To test the configuration:

On R4

```
R4#Sh ip route ospf | inc O

O       1.1.1.0 [110/67] via 10.1.34.3, 00:02:46, FastEthernet0/0
O       2.2.2.0 [110/66] via 10.1.34.3, 00:02:46, FastEthernet0/0
O       3.3.3.0 [110/2] via 10.1.34.3, 00:02:46, FastEthernet0/0
O       10.1.12.0 [110/66] via 10.1.34.3, 00:02:46, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.34.3, 00:02:46, FastEthernet0/0
```

Note R4 has prefix 1.1.1.0 /24 in its routing table, because OSPF is allowed through but it will NOT be able to communicate with that prefix.

```
R4#Ping 1.1.1.1

Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

R4#Telnet 1.1.1.1

Trying 1.1.1.1 ...
% Destination unreachable; gateway or host down
```

Note even though network 1.1.1.0 /24 is in R4's routing table no traffic is allowed in that network.

To verify and test the configuration:

On R1

```
R1#Telnet 4.4.4.4

Trying 4.4.4.4 ... Open

User Access Verification

Username: TST
Password:
```

Note R1 has established a telnet session with R4. To see the dynamically added lines in the ACL that allows this traffic to return:

On R2

```
R2#Show access-list

Reflexive IP access list TEST
     permit tcp host 4.4.4.4 eq telnet host 10.1.12.1 eq 40460 (66 matches) (time left 257)
Extended IP access list inbound
    10 permit ospf any any (57 matches)
    20 evaluate TEST
Extended IP access list outbound
    10 permit tcp any any eq www reflect TEST
    20 permit tcp any any eq telnet reflect TEST (35 matches)
    30 permit udp any any eq tftp reflect TEST
    40 permit ospf any any
```

Note an access-list is created dynamically called TEST. This access-list was created as a result of R1s telnet to R4's 4.4.4.4 IP address using port 40460 as the source and 23 as the destination port, therefore, this is created so the return traffic can be permitted back in.

Task 3

Reconfigure the RACL on R2 using the following parameters:

- R2 should allow the return HTTP traffic that is originated locally or by R1. The temporary access-list should be set with an idle timeout of 120 seconds.
- R2 should allow the return Telnet traffic that is originated locally or by R1. The temporary access-list should be set with an idle timeout of 60 seconds.
- R2 should allow the return TFTP traffic that is originated locally or by R1. The temporary access-list should be set with an idle timeout of 30 seconds.
- R2 should allow the return ICMP and DNS traffic that is originated locally or by R1. The temporary access-list should be set with an idle timeout of 10 seconds.
- R2 should allow the OSPF traffic into the network.

On R2

```
R2(config)#NO ip access-list extended outbound

R2(config)#ip access-list extended outbound
R2(config-ext-nacl)#permit ospf any any
R2(config-ext-nacl)#permit tcp any any eq 80 reflect TEST timeout 120
R2(config-ext-nacl)#permit tcp any any eq 23 reflect TEST timeout 60
R2(config-ext-nacl)#permit udp any any eq 69 reflect TEST timeout 30
R2(config-ext-nacl)#permit icmp any any reflect TEST timeout 10
R2(config-ext-nacl)#permit udp any any eq 53 reflect TEST timeout 10
```

To test the configuration:

On R1

To generate some ICMP traffic:

```
R1#Ping 4.4.4.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 4.4.4.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms
```

On R2

To see the dynamically created access-list:

```
R2#Sh access-list

Reflexive IP access list TEST
     permit icmp host 4.4.4.4 host 10.1.12.1  (19 matches) (time left 1)
Extended IP access list inbound
    10 permit ospf any any (85 matches)
    20 evaluate TEST
Extended IP access list outbound
    10 permit ospf any any
    20 permit tcp any any eq www reflect TEST
    30 permit tcp any any eq telnet reflect TEST
    40 permit udp any any eq tftp reflect TEST
    50 permit icmp any any reflect TEST (11 matches)
    60 permit udp any any eq domain reflect TEST
```

On R1

To generate Telnet traffic:

```
R1#Telnet 4.4.4.4

Trying 4.4.4.4 ... Open

User Access Verification

Password:
```

On R2

To see the dynamically added access-list:

```
R2#Sh access-list

Reflexive IP access list TEST
    permit tcp host 4.4.4.4 eq telnet host 10.1.12.1 eq 41142 (33 matches) (time left 54)
Extended IP access list inbound
   10 permit ospf any any (102 matches)
   20 evaluate TEST
Extended IP access list outbound
   10 permit ospf any any
   20 permit tcp any any eq www reflect TEST
   30 permit tcp any any eq telnet reflect TEST (19 matches)
   40 permit udp any any eq tftp reflect TEST
```

```
50 permit icmp any any reflect TEST (11 matches)
60 permit udp any any eq domain reflect TEST
```

Task 4

Reconfigure the timeout parameter of the extended "outbound" access-list such that all the dynamically created entries have a time to live of 120 seconds, DO NOT use the timeout keyword in the access-list to accomplish this task.

On R2

```
R2(config)#NO ip access-list extended outbound

R2(config)#ip access-list extended outbound
R2(config-ext-nacl)#permit ospf any any
R2(config-ext-nacl)#permit tcp any any eq 80 reflect TEST
R2(config-ext-nacl)#permit tcp any any eq 23 reflect TEST
R2(config-ext-nacl)#permit udp any any eq 69 reflect TEST
R2(config-ext-nacl)#permit icmp any any reflect TEST
R2(config-ext-nacl)#permit udp any any eq 53 reflect TEST

R2(config)#ip reflexive-list timeout 120
```

The above command specifies the length of time that reflexive access list entries will continue to exist when no packets in the session are detected, by default the timeout value is set to 300 seconds.
Note if you have configured a timeout for each RACL entry and you have configured the global timeout command, the more specific ones that are configured for each entry will take precedence over the global command.

Task 5

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 10 – Access-list and Time-range

Based on Lab 2' IP addressing and Topology

Lab Setup:

- Configure the routers according to LAB 2's IP addressing and topology

Task 1

Configure OSPF Area 0 on all directly connected interfaces on all four routers; ensure that loopback 0 interface of these routers are advertised with their correct mask. You should remove RIPv2 from all four routers.

On All Routers:

```
Rx(config)#NO router rip

Rx(config)#int loopback 0
Rx(config-if)#ip ospf network point-to-point

Rx(config-if)#router ospf 1
Rx(config-router)#network 0.0.0.0 0.0.0.0 area 0
```

To verify the configuration:

On R1

```
R1#Show ip route ospf | inc O

O       2.2.2.0 [110/2] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       3.3.3.0 [110/66] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       4.4.4.0 [110/67] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       10.1.23.0 [110/65] via 10.1.12.2, 00:00:05, FastEthernet0/0
O       10.1.34.0 [110/66] via 10.1.12.2, 00:00:05, FastEthernet0/0
```

Task 2

Configure R1 to allow its internal users to have the ability to browse the Internet during the weekdays ONLY. R4 should be configured such that its internal users can only browse the Internet in weekends. The access-list should be applied outbound on their F0/0 interface, since this is the interface that connects these routers to the Internet.

On R1

```
R1(config)#time-range WEEKDAYS
R1(config-time-range)#periodic weekdays 00:00 to 23:59

R1(config)#access-list 100 permit tcp any any eq 80 time-range WEEKDAYS

R1(config)#int f0/0
R1(config-if)#ip access-group 100 out
```

On R4

```
R4(config)#time-range WEEKENDS
R4(config-time-range)#periodic weekend 00:00 to 23:59

R4(config)#access-list 100 permit tcp any any eq 80 time-range WEEKENDS

R4(config)#int f0/0
R4(config-if)#ip access-group 100 out
```

The first step in configuring this policy is to configure the time-range and define the allowed time range. The second step is to configure the access-list referencing the time-range and lastly applying the access-list using the "access-group" command to the interface.

Task 3

Configure R3 to allow its internal users to browse the Internet using the following policy:

- This should ONLY be allowed Weekdays (M – F) between the hours of 2:00 PM and 6:30 PM.
- Because of unusual workload and special projects, this should NOT be allowed starting July 20th through Nov 26th 9:00 AM to 5:00 PM during the weekdays.

- The access-list should be applied outbound on their F0/0 interface, since this is the interface that connects to the Internet.

On R3

```
R3(config)#time-range ALLOW
R3(config-time-range)#periodic weekdays 14:00 to 18:30

R3(config)#time-range DENIED
R3(config-time-range)#absolut start 00:00 20 July 2007 end 23:59 26 November 2008
R3(config-time-range)#periodic weekdays 9:00 to 17:00

R3(config)#access-list 100 deny tcp any any eq 80 time-range DENIED
R3(config)#access-list 100 permit tcp any any eq 80 time-range ALLOW

R3(config)#int f0/0
R3(config-if)#ip access-group 100 out
```

Task 4

Configure R2 using the following policy:

- Outgoing Telnet traffic should only be denied between the hours of 11:00 AM and 2:00 PM, Monday to Friday.
- Outbound HTTP calls should be denied Monday to Friday, between the hours of 9:00 Am and 2:00 Pm starting Feb 19th 2006 to April 24th 2006.
- Any other traffic should be denied. Ensure that the access-list is applied outbound on their F0/0 interface, since this is the interface that connects to the Internet.

On R2

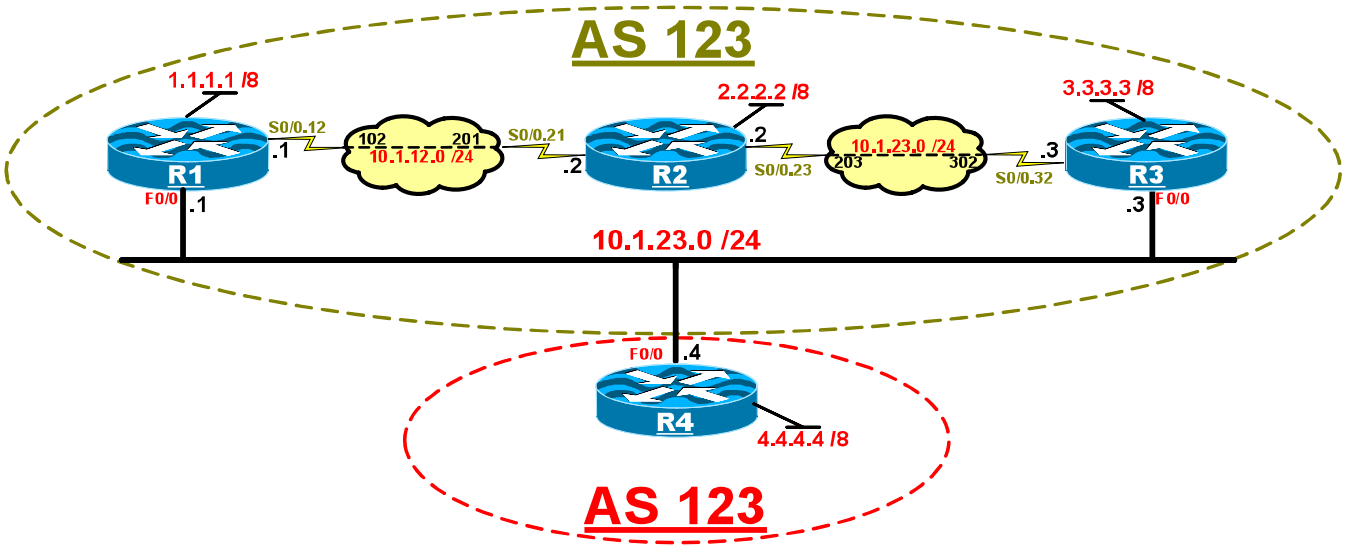
```
R2(config)#time-range TELNET
R2(config-time-range)#periodic weekdays 11:00 to 14:00

R2(config)#time-range HTTP
R2(config-time-range)#absol sta 00:00 19 August 2007 end 23:59 24 December 2007
R2(config-time-range)#periodic weekdays 9:00 to 14:00

R2(config)#access-list 100 deny tcp any any eq 23 time-range TELNET
```

```
R2(config)#access-list 100 permit tcp any any eq 23
R2(config)#access-list 100 deny tcp any any eq 80 time-range HTTP
R2(config)#access-list 100 permit tcp any any eq 80

R2(config)#int f0/0
R2(config-if)#ip access-group 100 out
```

Task 5

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 11 – Black Hole Filtering



Router	Interface	IP address
R1	Lo0 F0/0 S0/0.12	1.1.1.1/ 8 10.1.134.1/ 24 10.1.12.1 /24
R2	Lo0 S0/0.21 S0/0.23	2.2.2.2 /8 10.1.12.2/ 24 10.1.23.2 /24
R3	Lo0 S0/0.32 F0/0	3.3.3.3/ 8 10.1.23.3 /24 10.1.134.3/ 24
R4	Lo0 F0/0	4.4.4.4 /8 10.1.134.4/ 24

Lab Setup:

- Frame-relay should be configured between R2 and the R3, R4 serial links in a point to point manner.
- R1,R3 and R4 should participate in VLAN 134 on the F0/0 interfaces.

Task 1

Configure OSPF "123" for all routers in AS 123. R1,R2 and R3 should have IBGP peering in AS 123 while R4 is in AS 400 and has EBGP peering between R1 and R3.

On R1

```
R1(config)#ip bgp-community new-format

R1(config)#router ospf 1
R1(config-router)#netw 10.1.12.1 0.0.0.0 area 0
R1(config-router)#netw 10.1.134.1 0.0.0.0 area 0
R1(config-router)#netw 1.1.1.1 0.0.0.0 area 0

R1(config-router)#router bgp 123
R1(config-router)#No synchronization
R1(config-router)#bgp log-neighbor-changes
R1(config-router)#network 1.0.0.0
R1(config-router)#neighbor 2.2.2.2 remote-as 123
R1(config-router)#neighbor 2.2.2.2 update-source Loopback0
R1(config-router)#neighbor 10.1.134.4 remote-as 400
R1(config-router)#No auto-summary
```

On R2

```
R2(config)#ip bgp-community new-format

R2(config)#router ospf 1
R2(config-router)#netw 10.1.12.2 0.0.0.0 area 0
R2(config-router)#netw 10.1.23.2 0.0.0.0 area 0
R2(config-router)#netw 2.2.2.2 0.0.0.0 area 0

R2(config-router)#router bgp 123
R2(config-router)#No synchronization
R2(config-router)#bgp log-neighbor-changes
R2(config-router)#network 2.0.0.0
R2(config-router)#neighbor 1.1.1.1 remote-as 123
R2(config-router)#neighbor 1.1.1.1 update-source Loopback0
R2(config-router)#neighbor 1.1.1.1 route-reflector-client
R2(config-router)#neighbor 3.3.3.3 remote-as 123
R2(config-router)#neighbor 3.3.3.3 update-source Loopback0
R2(config-router)#neighbor 3.3.3.3 route-reflector-client
R2(config-router)#No auto-summary
```

On R3

```
R3(config)#ip bgp-community new-format

R3(config)#router ospf 1
R3(config-router)#netw 10.1.23.3 0.0.0.0 area 0
R3(config-router)#netw 10.1.134.3 0.0.0.0 area 0
R3(config-router)#netw 3.3.3.3 0.0.0.0 area 0

R3(config-router)#router bgp 123
R3(config-router)#No synchronization
R3(config-router)#bgp log-neighbor-changes
R3(config-router)#network 3.0.0.0
R3(config-router)#neighbor 2.2.2.2 remote-as 123
R3(config-router)#neighbor 2.2.2.2 update-source Loopback0
R3(config-router)#neighbor 10.1.134.4 remote-as 400
R3(config-router)#No auto-summary
R3(config-router)#end
```

On R4

```
R4(config)#ip bgp-community new-format

R4(config)#router bgp 400
R4(config-router)#No synchronization
R4(config-router)#bgp log-neighbor-changes
R4(config-router)#network 4.0.0.0
R4(config-router)#neighbor 10.1.134.1 remote-as 123
R4(config-router)#neighbor 10.1.134.3 remote-as 123
R4(config-router)#No auto-summary
```

Verify the configuration:

On R1

```
R1#Sh ip route ospf | i O

O        2.2.2.2/32 [110/65] via 10.1.12.2, 00:01:10, Serial0/0.12
O        3.3.3.3/32 [110/11] via 10.1.134.3, 00:01:10, FastEthernet0/0
O        10.1.23.0 [110/74] via 10.1.134.3, 00:01:10, FastEthernet0/0
```

R1#**Sh ip bgp s | b Neigh**

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

```
2.2.2.2         4   123         9       7       5    0     0 00:02:04        2
10.1.134.4      4   400         8       7       4    0     0 00:00:30        1
```

On R2

```
R2#Sh ip route ospf | i O

O        1.1.1.1/32 [110/65] via 10.1.12.1, 00:02:07, Serial0/0.21
O        3.3.3.3/32 [110/65] via 10.1.23.3, 00:02:07, Serial0/0.23
O        10.1.134.0 [110/74] via 10.1.23.3, 00:02:07, Serial0/0.23
```

R2#**Sh ip bgp s | b Neigh**

```
Neighbor        V    AS  MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.1         4   123         8       10       5    0     0 00:13:13        2
3.3.3.3         4   123         8       10       5    0     0 00:02:22        2
```

On R3

```
R3#Sh ip route ospf | i O

O        1.1.1.1/32 [110/11] via 10.1.134.1, 00:02:41, FastEthernet0/0
O        2.2.2.2/32 [110/65] via 10.1.23.2, 00:02:41, Serial0/0.32
O        10.1.12.0 [110/74] via 10.1.134.1, 00:02:41, FastEthernet0/0
```

R3#**Sh ip bgp s | b Neigh**

```
Neighbor        V    AS  MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2         4   123        10       8       5    0     0 00:02:51        3
10.1.134.4      4   400        10       9       5    0     0 00:02:15        1
```

On R4

R1#**Sh ip bgp s | b Neigh**

```
Neighbor        V    AS  MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.1.134.1      4   123         9       10       5    0     0 00:02:36        3
10.1.134.3      4   123         9       10       5    0     0 00:02:42        3
```

Task 2

Configure a filter that prevents Denial of Service to Server 66.66.66.66. All traffic should be dropped on R2 from any direction. You may only use an additional Network of 192.0.2.1 to assist in the Task.

On R2

```
R2(config)#int lo999
R2(config-if)#ip add 192.0.20.1 255.255.255.0
R2(config-if)#ip ospf 1 area 0
```

This loopback is added to allow the edge routers to find this network in the IGP

```
R2(config)#ip route 192.0.20.0 255.255.255.0 null 0
```

The static to a null interface is added to have routes destined for the network to be Black Holed.

```
R2(config)#route-map BLACKHOLE permit 10
R2(config-route-map)#match tag 666
R2(config-route-map)#set origin igp
R2(config-route-map)#set community 123:666 no-export
R2(config-route-map)#set ip next-hop 192.0.20.1
```

Any Route that is tagged with "666" will have a community tag added that only allows it to be advertised to IBGP but not EBGP peers.

```
R2(config-route-map)#router bgp 123
R2(config-router)#redistribute static route-map BLACKHOLE
R2(config-router)#neighbor 1.1.1.1 send-community
R2(config-router)#neighbor 3.3.3.3 send-community
```

In order for the Community value to be understood by All of the IBGP peers, the community attribute must be propagated. The static routes are redistributed into BGP that match the Criteria set forth in the Route MAP.

On R1

```
R1#Sh ip route 192.0.20.1

Routing entry for 192.0.20.1/32
  Known via "ospf 20", distance 110, metric 65, type intra area
  Last update from 10.1.12.2 on Serial0/0.12, 00:00:04 ago
  Routing Descriptor Blocks:
  * 10.1.12.2, from 2.2.2.2, 00:00:04 ago, via Serial0/0.12
```

```
      Route metric is 65, traffic share count is 1
```

R1 can see the 192.0.20.0 prefix in the routing table. This will be important when forwarding to the next hop addresss specified in the route-map on R2.

On R2

```
R2(config)#ip route 66.66.66.0 255.255.255.0 null 0 tag 666
```

The static route represents the network that is under attack.

```
R2#Sh ip bgp 66.66.66.0

BGP routing table entry for 66.66.66.0/24, version 10
Paths: (1 available, best #1, table Default-IP-Routing-Table, not
advertised to EBGP peer)
Flag: 0x820
  Advertised to update-groups:
     1
  Local
    192.0.20.1 from 0.0.0.0 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced,
best
      Community: 123:666 no-export
```

R1 can see the network under attack after it has been redistributed into BGP. The Community attribute has been successfully attached.

On R1

```
R1#Sh ip bgp

BGP table version is 6, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i2.0.0.0          2.2.2.2                  0    100      0 i
*> 4.0.0.0          10.1.134.4               0             0 400 i
*>i66.66.66.0/24    192.0.20.1               0    100      0 i
```

The network under attack has been bound to the 192.0.20.1 next hop address. All packets that are destined to the 66.66.66.66 prefix will be forwarded to R2 where the route is Blac Holed. This will continue while the static route for 66.66.66.0 still exists.

On R4

```
R4#Sh ip bgp 66.66.66.0

% Network not in table
```

The 66.66.66.0 network is not in the table as it was stripped off due to the "no-export" commnty that was attached to the route.

```
R4(config)#ip route 66.66.66.0 255.255.255.0 10.1.134.1
```

The attacking router has been provisioned with a static to continue the attack through R1.

```
R4#Trace 66.66.66.66

Type escape sequence to abort.
Tracing the route to 66.66.66.66

  1 10.1.134.1 8 msec 16 msec 24 msec
  2 10.1.12.2 12 msec 20 msec 28 msec
  3 10.1.12.2 !H  *  !H
```

A traceroute reveals that the packet reaches R2 but the is stopped there. The "!H" shows that packet is being being replied with Host Unreachable which proves the filter works. The attacker now knows there a possible filter in place.

On R2

```
R2(config)#int null 0
R2(config-if)#No ip unreachables
```

We configure the null interface to not respond to a packet that does not have a known destination.

On R4

```
R4#Trace 66.66.66.66

Type escape sequence to abort.
Tracing the route to 66.66.66.66

  1 10.1.134.1 28 msec 16 msec 28 msec
  2 10.1.12.2 16 msec 20 msec 28 msec
  3  *  *  *
  4  *  *  *
```

[Ctrl-6-X]

After repeating the traceroute, we see that the attacker is not given any extra unneccesary information.

Task 3

Configure the Black Hole Filter to be remotely triggered from R2. There should not be any filters configured on R1 or R3 to accomplish the Task. Attackers should see no response.

On R2

```
R2(config)#No int lo999

R2(config)#route-map BLACKHOLE permit 20
```

First we remove the loopback interface and add a second statement ro the Route-MAP.
This will prevent the 192.0.20.0 prefix from being learned via the IGP and will be redistributed into BGP instead.

On R1 and R2

```
Rx(config)#ip route 192.0.20.0 255.255.255.0 null 0
```

A static to a null for the next hop address is added to each of the edge routers. This will allow the nexthop to be local on ALL of the edge routers to allow the Black Hole filter to operate on ANY edge router that it is configured on. R2 triggers the attacked route with tag 666 to trigger "*next-hop = 192.0.20.0 = null.*" Sice the null is on all of the routers, the packet does not have to go to R2 to be destroyed.

On R4

```
R4#Trace 66.66.66.66

Type escape sequence to abort.
Tracing the route to 66.66.66.66

  1 10.1.134.1 16 msec 36 msec 8 msec
  2 10.1.134.1 !H  *  !H
```

Notice how we are receiving the Host Unreachable flag again. However the packet dies on R1.

On R1 and R3

```
Rx(config)#int null 0
Rx(config-if)#No ip unreachables
```

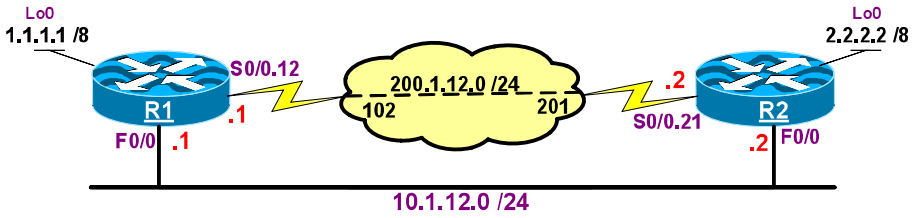
On R4

```
R4#Ping 66.66.66.66

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 66.66.66.66, timeout is 2 seconds:
…..
Success rate is 0 percent (0/5)

R4(config)#No ip route 66.66.66.0 255.255.255.0 10.1.134.1
R4(config)#ip route 66.66.66.0 255.255.255.0 10.1.134.3

R4#Ping 66.66.66.66

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 66.66.66.66, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 12 – uRPF



Lab Setup:

- Configure the F0/0 interface of R1 and R2 in VLAN 100.
- The Frame-Relay interface of these routers should be configured in a point-to-point manner.

Task 1

Configure the routers such that R2 uses its F0/0 interface to reaches network 1.0.0.0 /8, whereas, R1 uses the frame-Relay cloud to reach network 2.0.0.0 /8, .

On R1

```
R1(config)#ip route 2.0.0.0 255.0.0.0 s0/0.12
```

On R2

```
R2(config)#ip route 1.0.0.0 255.0.0.0 f0/0
```

To verify the configuration:

On R1

```
R1#Show ip route static

S    2.0.0.0/8 is directly connected, Serial0/0.12
```

On R2

```
R2#Show ip route static

S    1.0.0.0/8 is directly connected, FastEthernet0/0
```

Task 2

Configure URPF on R2's Frame-Relay interface

On R2

```
R2(config)#int s0/0.21
R2(config-subif)#ip verify unicast source reachable-via rx
```

Note that there is also an old form of this command, which does the same thing "ip verify unicast reverse-path". However, Cisco recommends using a new command when configuring uRPF.

To verify the configuration:

On R2

```
R2#Show ip inter s0/0.21 | b ip verify

  IP verify source reachable-via RX
  0 verification drops
  0 suppressed verification drops
```

Note with URPF enabled, the local router (R2) will compare the source IP address of all packets received to its routing table; this is done to ensure that they arrive on the best path interface, meaning that the closest interface back to the source. If the check fails, the packets are dropped, if the check passes, the packets are processed.

To test the configuration:

On R1

```
R1#Ping 2.2.2.2 source lo0


Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
.....
Success rate is 0 percent (0/5)

R2#Show ip inter s1/0.21 | b ip verify

  IP verify source reachable-via RX
  5 verification drops
  0 suppressed verification drops
```

Note the ping failed because the source ip address of the ICMP Echo message is 1.1.1.1, and from R2's perspective, these packets should have arrived through F0/0 interface and NOT the Frame-Relay interface.

To test the configuration further:

On R2

```
R2(config)#access-list 100 permit ip any any log-input

R2(config)#int f0/0
R2(config-if)#ip access-group 100 in

R2(config-if)#int s0/0.21
R2(config-subif)#ip access-group 100 in
```

In order to reveal the source/destination IP addresses and the interfaces, the above access-list is configured with "log-input" and applied to both S0/0.21 and F0/0.

On R1

```
R1#Ping 2.2.2.2 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
.....
Success rate is 0 percent (0/5)
```

You should see the following message on R2's console:

%SEC-6-IPACCESSLOGDP: list 100 permitted icmp 1.1.1.1 (Serial0/1.21) ->

2.2.2.2 (0/0), 4 packets

Note the output of the following command reveals that R2 expects the traffic from 1.1.1.1 to come through its F0/0 and NOT S0/0.21 interface; therefore, it drops the packets.

```
R2#Sh ip route static

S    1.0.0.0/8 is directly connected, FastEthernet0/0
```

Task 3

Configure R2 such that it allows ingress traffic from network 1.0.0.0 /8 through any interface. DO NOT modify the routing table or remove the "ip verify unicast source reachable-via rx" command to accomplish this task.

Note to accomplish this task, an access-list is configured to permit all traffic from network 1.0.0.0 /8 and tied to the "ip verify unicast source reachable-via rx" command. The condition of the access-list is checked ONLY when the condition of URPF fails, IF THE CONDITION OF URPF IS SUCCESSFUL, THE ACCESS-LIST IS NOT CHECKED.

On R2

```
R2(config)#access-list 101 permit ip 1.0.0.0 0.255.255.255 any

R2(config)#int s0/0.21
R2(config-subif)#ip verify unicast source reachable-via rx 101
```

To verify the configuration:

On R2

```
R2#Sh ip int s0/0.21 | b ip verify

  IP verify source reachable-via RX, ACL 101
  10 verification drops
  0 suppressed verification drops
```

To test the configuration:

On R1

```
R1#Ping 2.2.2.2 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/32/76 ms
```

Note you should see the following message on R2's console:

On R2

%SEC-6-IPACCESSLOGDP: list 100 permitted icmp 1.1.1.1 (Serial0/1.21) -> 2.2.2.2 (0/0), 5 packets

On R2

```
R2#Sh ip int S0/0.21 | b ip verify

  IP verify source reachable-via RX, ACL 101
  10 verification drops
  5 suppressed verification drops
```

Task 4

Reconfigure R2 based on the previous conditions such that ONLY ICMP packets are allowed through any interface, all other traffic from this network MUST come through F0/0 interface, if they do not, and they should be dropped.

On R2

```
R2(config)#NO access-list 101

R2(config)#access-list 101 permit icmp 1.0.0.0 0.255.255.255 any
```

To test the configuration:

On R1

```
R1#Ping 2.2.2.2 source lo0

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/41/72 ms
```

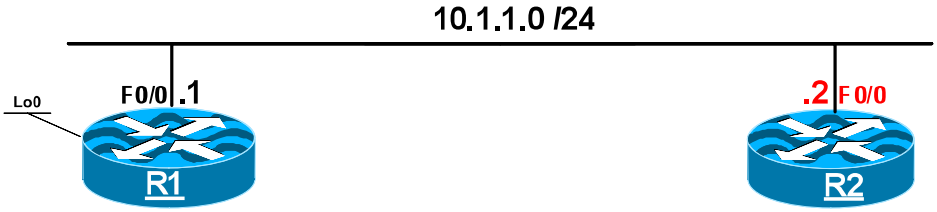
Note the ICMP traffic works fine, whereas, the following TCP traffic failed.

```
R1#Telnet 2.2.2.2 /source-interface lo0

Trying 2.2.2.2 ...
% Connection timed out; remote host not responding
```

Task 5

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 13 – Control Plane Policy



Lab Setup:

- The F0/0 interface of R1 and R2 should be configured in VLAN 100.
- Configure RIPv2 on all routers and advertise their directly connected interfaces in this routing protocol.
- Enable VTY logging on both devices.

IP Addressing:

Router	Interface	IP address
R1	F0/0 Lo0	10.1.1.1 /24 1.1.1.1/8
R2	**F0/0**	**10.1.1.2 /24**

Task 1

On R2, configure policing for ICMP echo request messages to rate limit it up to 50kbps using Control Plane Policy.

The Control Plane Policing feature allows to configure a quality of service (QoS) filter that manages the traffic flow of control plane packets to protect the control plane of Cisco IOS routers and switches against reconnaissance and denial-of-service (DoS) attacks. Control Plane is responsible for handling traffic like routing protocols, management protocols, event sending, authentication requests, etc. destined to or originated from the router.

On R2

Create an ACL that will match the traffic:

```
R2(config)#access-list 120 permit icmp any any echo
```

Match interesting traffic using class map:

```
R2(config)#class-map ICMP
R2(config-cmap)#match access-group 120
```

Create policy map and assign previously created class map. Police (rate limit) will drop the traffic, which exceeded 50k.

```
R2(config)#Policy-map TST
R2(config-pmap)#class ICMP
R2(config-pmap-c)#police 50000 conform transmit exceed drop
```

Assign policy to the Control Plane in the inbound direction.

```
R2(config)#control-plane
R2(config-cp)#service-policy input TST
```

%CP-5-FEATURE: Control-plane Policing feature enabled on Control plane aggregate path

You can verify this feature by issuing large ping from R1 towards R2 and see if the traffic is limited.

```
R2#Show policy-map control-plane
 Control Plane

  Service-policy input: TST

    Class-map: ICMP (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 120
      police:
      cir 50000 bps, bc 1562 bytes
    conformed 0 packets, 0 bytes; actions:
```

```
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any

R1#Ping 10.1.1.2 size 1500

Type escape sequence to abort.
Sending 5, 1500-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!.!.!
Success rate is 60 percent (3/5), round-trip min/avg/max = 132/154/184 ms
R1#
```

Note that only 3 pings are successful. The below command clearly shows that the traffic has been rate limited.

```
R2#Show policy-map control-plane
 Control Plane

  Service-policy input: TST

    Class-map: ICMP (match-all)
      5 packets, 7570 bytes
      5 minute offered rate 8000 bps, drop rate 6000 bps
      Match: access-group 120
      police:
          cir 50000 bps, bc 1562 bytes
        conformed 3 packets, 4542 bytes; actions:
          transmit
        exceeded 2 packets, 3028 bytes; actions:
          drop
        conformed 2000 bps, exceed 2000 bps

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
   Match: any
```

Task 2

Configure R2 such that it allows Telnet connections only to the IP address of 1.1.1.1. You should NOT apply an access-list to any of the interfaces or VTY lines of these two routers to accomplish this task.

This can be done using Control Plane Policing. Note that wording of this task does not mention Control Plane.

On R2

Configure an ACL that will match TELNET traffic from any routers interface to the host 10.1.1.1.

```
R2(config)#access-list 130 permit tcp any host 10.1.1.1 eq telnet
```

Match interesting traffic using class map:

```
R2(config)#class-map TELNET
R2(config-cmap)#match access-group 130
```

Configure a policy map and assign previously created class map. This policy map should be applied on the OUTBOUND direction.

```
R2(config)#policy-map TST
R2(config-pmap)#class TELNET
R2(config-pmap-c)#drop
```

Apply the policy map

```
R2(config)#control-plane
R2(config-cp)#service-policy output TST
```

To verify, initiate telnet connection from R2 toward R1's IP addresses and check the counters for the policy map.

```
R2#Show policy-map control-plane out

 Control Plane

  Service-policy output: CPP-OUT

    Class-map: TELNET (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
```

```
      Match: access-group 130
      drop
    Class-map: class-default (match-any)
      8 packets, 791 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
R2#

R2#Telnet 10.1.1.1

Trying 10.1.1.1 ...
% Connection timed out; remote host not responding
```

Note: this connection is NOT successful

```
R2#Telnet 1.1.1.1

Trying 1.1.1.1 ... Open

User Access Verification
Password:
R1>exit
[Connection to 1.1.1.1 closed by foreign host]
```

Note: this connection is successful

```
R2#Show policy-map control-plane out
 Control Plane

  Service-policy output: TST

    Class-map: TELNET (match-all)
      4 packets, 240 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 130
      drop

    Class-map: class-default (match-any)
      39 packets, 2951 bytes
      5 minute offered rate 1000 bps, drop rate 0 bps
      Match: any
```

Task 3

On R2, enable Control Plane logging feature for all dropped packets.

By default, Control Plane Policing silently performs all configured operations. However, a good security administrator should be aware what is going on in his/her network.

On R2

Create a special class map type and match all dropped packets.

```
R2(config)#class-map type logging match-any TEST
R2(config-cmap)#match packets dropped
```

Create a special policy map type and assign previously created class map. Enable logging of all packets matched that policy.

```
R2(config)#policy-map type logging TST
R2(config-pmap)#class TEST
R2(config-pmap-c)#log
```

Assign the policy map to the Control Plane.

```
R2(config)#control-plane
R2(config-cp)#service-policy type logging input TST
```

%CP-5-FEATURE: Control-plane Logging feature enabled on Control plane aggregate path

To verify perform the test from Task 1 and see if it generates the log message.

```
R1#Ping 10.1.1.2 size 1500

Type escape sequence to abort.
Sending 5, 1500-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!.!.!
Success rate is 60 percent (3/5), round-trip min/avg/max = 196/218/261 ms
```

%CP-6-IP: DROP Control-plane Policing 10.1.1.1 -> 10.1.1.2 icmp
%CP-6-IP: DROP Control-plane Policing 10.1.1.1 -> 10.1.1.2 icmp

Note: Two drops have been logged as two ICMP echo messages have been dropped.
To see what features on the Control Plane are configured, use the following command:

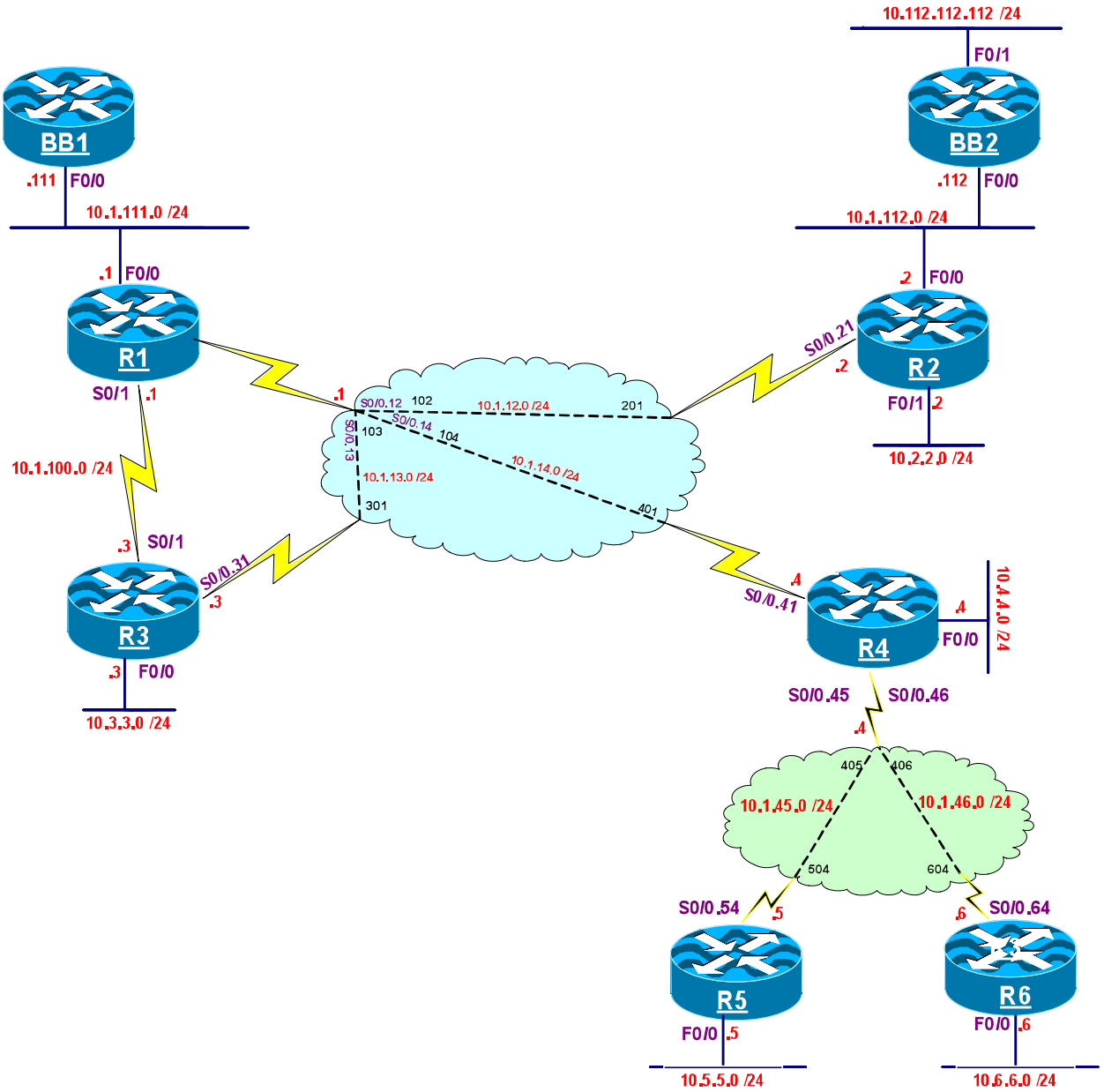
```
R2#Show control-plane features
```

```
Total 2 features configured
Control plane aggregate path features :
-------------------------------------------------------
Control-plane Logging activated Mar 01 2002 00:4
Control-plane Policing activated Mar 01 2002 00:2
```

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 14 – Attacks



Surinderjeet Singhh

IP Addressing chart:

Router	Interface	Connecting to:	IP Address
R1	S0/0.12 S0/0.13 S0/0.14 S0/1 F0/0	R2 R3 R4 R3 BB1	10.1.12.1 /24 10.1.13.1 /24 10.1.14.1 /24 10.1.100.1 /24 10.1.111.1 /24
R2	S0/0.21 F0/0 F0/1	R1 BB2 -	10.1.12.2 /24 10.1.112.2 /24 10.2.2.2 /24
R3	S0/0.31 S0/1 F0/0	R1 R1 -	10.1.13.3 /24 10.1.100.3 /24 10.3.3.3 /24
R4	S0/0.41 S0/0.45 S0/0.46 F0/0	R1 R5 R6 -	10.1.14.4 /24 10.1.45.4 /24 10.1.46.4 /24 10.4.4.4 /24
R5	S0/0.54 F0/0	R4 R6	10.1.45.5 /24 10.5.5.5 /24
R6	S0/0.64 F0/0	R4 R5	10.1.46.6 /24 10.6.6.6 /24
BB1	F0/0	R1	10.1.111.111 /24
BB2	F0/0 F0/1	R2 -	10.1.112.112 /24 10.112.112.112 /24

Lab Setup:

VLANs:

- F0/0 interface of BB1 and R1 should be configured in VLAN 11
- F0/0 interface of R3 should be configured in VLAN 3
- F0/0 interface of BB2 and R2 should be in VLAN 22
- F0/0 interface of R5 should be configured in VLAN 5
- F0/0 interface of R6 should be configured in VLAN 6
- F0/1 interface of R2 should be configured in VLAN 2
- F0/1 interface of BB2 should be configured in VLAN 112
- F0/0 interface of R4 should be configured in VLAN 4

Frame-Relay:

- All frame-Relay connections should be configured in a Point-to-point manner.

Trunking:

The trunking should be established between SW-1 and SW-2 using ports F0/19 and F0/20.

Routing:

Run RIPv2 on the routers and advertise their directly connected networks.

Task 1

Configure BB1 such that ONLY hosts 10.1.45.5 and 10.1.46.6 are allowed to Telnet into BB1. Do not use an extended access-list to accomplish this task.

On BB1

```
BB1(config)#access-list 1 permit host 10.1.45.5
BB1(config)#access-list 1 permit host 10.1.46.6

BB1(config)#line vty 0 871
BB1(config)#access-class 1 in
```

Note when the standard access-list is applied to the VTY line/s using the "access-class" command, the regular traffic is NOT checked against the access-list, whereas, if an extended access-list was applied to an interface, all traffic coming through that interface would be checked against the access-list before the traffic is allowed.

On R1

```
R1#Telnet 10.1.111.111

Trying 10.1.111.111 ...
% Connection refused by remote host
R1#
```

On R5

```
R5#Telnet 10.1.111.111

Trying 10.1.111.111 ... Open
User Access Verification
Password:
BB1>
```

On R6

```
R6#Telnet 10.1.111.111

Trying 10.1.111.111 ... Open
User Access Verification
Password:
BB1>
```

Task 2

Ensure that ONLY host with an IP address of 111.1.1.1 has the ability to access BB1's SNMP agent. The community string should be configured to be "Cisco" and this host should only have read only rights.

On BB1

```
BB1(config)#access-list 11 permit host 111.1.1.1
BB1(config)#snmp-server community Cisco ro 11
```

The above configuration ONLY allows SNMP host with an IP address of 111.1.1.1 to access BB1's SNMP agent. The community string that this host must use is "Cisco" and this host has READ ONLY rights.
Since SNMPv1 has no authentication capabilities, the following is recommended:

- **SNMPv1 should ONLY be used on the internal networks**
- **Access should be limited by using an access-list**

Task 3

You are worried about Smurf attacks through the frame-Relay cloud; configure R3 to block DoS Smurf attacks for its F0/0 segment. You should use an access-list to accomplish this task.

On R3

```
R3(config)#access-list 100 deny icmp any host 10.3.3.255 log
R3(config)#access-list 100 permit ip any any

R3(config)#interface S0/0.31
R3(config-if)#ip access-group 100 in
```

Smurf attack occurs when a large number of ICMP packets are sent to a router's subnet broadcast address with a spoofed source IP address of a host within that subnet. Post IOS 12.0 the routers are default with the "no ip directed-broadcast" command, which prevents this kind of attacks. However, if the "ip directed-broadcast" is enabled for whatever reason, then, the access-list will block these types of attacks.

Task 4

The administrator of R3 has requested that all the pings, icmp redirects and mask-requests to be blocked on its S0/1 interface. The other ICMP packet types should be permitted.

On R3

```
R3(config)#access-list 101 deny icmp any any echo log
R3(config)#access-list 101 deny icmp any any redirect log
R3(config)#access-list 101 deny icmp any any mask-request log
R3(config)#access-list 101 permit ip any any

R3(config)#int s0/1
R3(config-if)#ip access-group 101 in
```

Note ICMP echo, redirects and mask-requests packets can be used as a DoS and/or reconnaissance attacks, therefore, it is recommended to deny these types of ICMP packets from entering your internal networks.
Verification is a bit tricky, as there is a need to configure static route (or adjust RIP updates) to ensure R1 uses segment 10.1.100.0/24 as preferred path to 10.3.3.3.

On R1

```
R1(config)#ip route 10.3.3.3 255.255.255.255 10.1.100.3

R1#Ping 10.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.3, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

On R3

00:44:26: ICMP: dst (10.3.3.3) administratively prohibited unreachable sent to 10.1.100.1

Task 5

Configure R1 to block all inbound and outbound UDP Traceroute packets on its F0/0 interface.

On R1

```
R1(config)#access-list 100 deny udp any any range 33400 34400 log
R1(config)#access-list 100 permit ip any any
```

Note UDP Traceroute uses the above port range; therefore, they need to be blocked.

```
R1(config)#inter f0/0
R1(config)#ip access-group 100 in
R1(config)#ip access-group 100 out
```

Traceroute either sends out ICMP Echos (Typically in Windows) or UDP, which occurs in the most implementation of this command. This utility achieves its goal by gradually increasing the TTL value to examine the path through which a packet/s traverses.
The router begins by sending its first packet with a TTL value of one, the first hop router will discard this message and it will send back an ICMP TTL exceeded message. When the local router running the Traceroute application receives the "ICMP TTL exceeded" message, it can ascertain the hop via the source IP address. This process will continue until the destination is reached.

On R5

```
R5#Traceroute 10.1.111.111

Type escape sequence to abort.
Tracing the route to 10.1.111.111

  1 10.1.45.4 100 msec 88 msec 80 msec
  2 10.1.14.1 76 msec 160 msec 156 msec
  3 10.1.14.1 !A  !A  *
```

On R1

01:00:04: %SEC-6-IPACCESSLOGP: list 100 denied udp 10.1.45.5(38543) -> 10.1.111.111(33441), 1 packet

Note: Traceroute packet was destined on port 33441 in this case.

Task 6

You are experiencing DDoS attacks inbound F0/0 interface of BB2, after further investigations, you found that this router is under a Trin00 attack. Configure this router to block this attack coming in the network. You should also NOT allow infected internal hosts from sending messages out of the network to the vulnerable ports.

On BB2

```
BB2(config)#access-list 100 deny tcp any 10.112.112.0 0.0.0.255 eq 27665
BB2(config)#access-list 100 deny tcp any 10.112.112.0 0.0.0.255 eq 1524
BB2(config)#access-list 100 deny udp any 10.112.112.0 0.0.0.255 eq 31335
BB2(config)#access-list 100 deny udp any 10.112.112.0 0.0.0.255 eq 27444
BB2(config)#access-list 100 permit ip any any

BB2(config)#inter f0/0
BB2(config-if)#ip access-group 100 in
BB2(config-if)#ip access-group 100 out
```

The following explains the Trin00 attack:

- **Trin00 is a distributed SYN DoS attack**
- **It basically uses UDP floods**
- **Trin00 sets up communications between the clients, handlers and agents using two TCP (1524, 27665) and two UDP ports (27444, 31335)**
- **Should ONLY be blocked when under attack, because the high ports may be used by a host**

Task 7

You were just notified that network 10.2.2.0 /24 is under a Trinityv3 DDoS attack. The attack is coming through F0/0 interface; configure this router to block this attack.

On R4

```
R4(config)#access-list 100 deny tcp any 10.2.2.0 0.0.0.255 eq 33270 log
R4(config)#access-list 100 deny tcp any 10.2.2.0 0.0.0.255 eq 6667 log
R4(config)#access-list 100 permit ip any any

R4(config)#int f0/1
R4(config-if)#ip access-group 100 in
```

Trinityv3 can be used to launch few flood attacks on a given site; these include UDP, Fragment, SYN, RST, ACK and other floods. The communication from the intruder (Handler) to the agent is accomplished via Internet Relay Chat (IRC) or IRQ from AOL. This attack mainly uses port 6667 with a program that listens on TCP 33270.

Task 8

Configure R5 such that it blocks spoofing of the IP packets by blocking RFC 1918 and RFC 2827.

RFC 2827 states that in order to be a good citizen on the Internet you should prevent users of your network spoofing other network's IP address, by preventing any outbound traffic on your network that does not have a source IP address that's from your IP address space.

On R5

```
R5(config)#access-list 100 deny ip 10.5.5.0 0.0.0.255 10.5.5.0 0.0.0.255
R5(config)#access-list 100 deny ip 172.16.0.0 0.15.255.255 10.5.5.0 0.0.0.255
R5(config)#access-list 100 deny ip 192.168.0.0 0.0.255.255 10.5.5.0 0.0.0.255
R5(config)#access-list 100 deny ip 127.0.0.0 0.255.255.255 10.5.5.0 0.0.0.255
R5(config)#access-list 100 deny ip 169.254.0.0 0.0.255.255 10.5.5.0 0.0.0.255
R5(config)#access-list 100 deny ip 224.0.0.0 15.255.255.255 10.5.5.0 0.0.0.255
R5(config)#access-list 100 deny ip 255.255.255.255 0.0.0.0 10.5.5.0 0.0.0.255
R5(config)#access-list 100 deny ip 0.0.0.0 0.0.0.0 10.5.5.0 0.0.0.255
R5(config)#access-list 100 permit ip any any
```

Applying the Access-list:

```
R5(config)#int s0/0.54
R5(config-if)#ip access-group 100 in
```

Note if things are working correctly, you should NOT receive any packets with a source IP address that is defined in RFC 1918, 127.0.0.0 /8, the block that belongs to Microsoft (169.254.0.0 /16), Class D, and/or broadcast address.
RFC 2827 is performed on the second last line of this access-list.

Task 9

You are the administrator of R6 and you are told that network 10.6.6.0 /24 is under an HTTP Code Red and Nimda attack. These attacks are coming from the connection to the frame-Relay network; you must use NBAR to accomplish this task.

On R6

```
R6(config)#ip cef
```

This command is required for NBAR to work, nowadays it's enabled on most routers by default

```
R6(config)#class-map match-any Code-RED
```

The above command creates a Class map named "Code-RED".

```
R6(config-cmap)#match protocol http url *cmd.exe*
```

The above command matches the HTTP protocol and the string that is in the url.

```
R6(config-cmap)#match protocol http url *.ida*
```

The above command matches the HTTP protocol and the string that is in the url.

```
R6(config-cmap)#match protocol http url *root.exe*
```

The above command matches the HTTP protocol and the string that is in the url.

```
R6(config-cmap)#match protocol http url *readme.eml*
```

This is where Nimda is different to Code Red Nimda builds on Code Red.

Creating the Policy Map:

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

This command creates a Policy map called TEST.

```
R6(config-pmap)#class Code-RED
R6(config-pmap)#set ip precedence 4
```

Applying the Policy Map to the interface for inbound traffic:

```
R6(config)#Int s0/0.64
R6(config-if)#service-policy input TEST
```

Configuring the Access List to Drop the Code Red Packets:

Creating the Access List

```
R6(config)#access-list 100 deny ip any any precedence 4
```

This command specifies to drop all packets that have the IP Precedence 4.

```
R6(config)#access-list 100 permit ip any any
```

Applying the Access List:

```
R6(config)#int f0/0
R6(config-if)#ip access-group 100 out
```

There are few versions of Code Red, it attacks Microsoft IIS Web server version 4 and 5 product. Since this worm uses a common port number, it is very difficult to detect. However, if you see unusually high number of HTTP requests from the same IP address or the web server in the DMZ or elsewhere to internal IP addresses there could be a problem.

Task 10

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Advanced
CCIE SERVICE PROVIDER
v3.0

www.MicronicsTraining.com

Narbik Kocharians
CCIE #12410
R&S, Security, SP

Paul Negron
CCIE #14856
SP

Syslog, IP Accounting and IP SLA

Lab 1 – Syslog

Use any router to accomplish the following tasks.

Task 1

Configure R1 to locally store 4096 Bytes of severity 0 Syslog messages.

On R1

```
R1(config)#Logging buffered 4096 0
```

The "Logging buffer" command is used to store Syslog messages locally. When the severity level is configured, all lower level below the specified level is also included. If a Syslog severity level of 4 is configured, then, levels 0 – 4 is logged.

To verify the configuration:

On R1

```
R1#Show logging | Inc Buffer logging:

    Buffer logging:  level emergencies, 0 messages logged, xml
disabled,
```

Task 2

With minimum local memory reservation, configure R2 to store maximum possible syslog messages.

On R2

```
R2(config)#Logging buffered 4096 7
```

The question mark can be used to determine the minimum and/or the maximum values of a given parameter.

To verify the configuration:

On R2

```
R2#Show logging | Inc Buffer logging:
```

Buffer logging: **level debugging**, 2 messages logged, xml disabled,

To see all the levels:

```
R2(config)#Logging buffered ?

  <0-7>               Logging severity level
  <4096-2147483647>   Logging buffer size
  alerts              Immediate action needed           (severity=1)
  critical            Critical conditions               (severity=2)
  debugging           Debugging messages                (severity=7)
  discriminator       Establish MD-Buffer association
  emergencies         System is unusable                (severity=0)
  errors              Error conditions                  (severity=3)
  filtered            Enable filtered logging
  informational       Informational messages            (severity=6)
  notifications       Normal but significant conditions (severity=5)
  warnings            Warning conditions                (severity=4)
  xml                 Enable logging in XML to XML logging buffer
```

Task 3

Configure R3 to store all Syslog messages locally except the ones that include "10.1.1.1" IP address.

On R3

```
R3(config)#Logging discriminator TST msg-body drops "10.1.1.1"
R3(config)#Logging buffered discriminator TST 7
```

The "logging discriminator" can be used to inspect the patterns of a Syslog message. A message discriminator is a Syslog processor to inspect a Syslog message before its final message delivery. A discriminator can be configured to utilize a user-specified list of criteria to block some messages. As you can see in this task, once it's enabled, it can then be defined.

To verify the configuration:

On R3

```
R3#Show logging

Syslog logging: enabled (12 messages dropped, 0 messages rate-limited,
                0 flushes, 0 overruns, xml disabled, filtering disabled)

Active Message Discriminator:
TST       msg-body       drops    10.1.1.1

(The rest of the output is omitted)
```

Task 4

Configfure R4 such that all Syslog messages with a severity of emergencies through alerts are logged to a remote Syslog server located at 200.1.1.1.

On R4

```
R4(config)#Logging host 200.1.1.1
R4(config)#Logging trap 1
```

The "Logging host" command specifies the IP address or the name of the Syslog server. FreeBSD and Linux are two examples of the Unix servers that can be used to run a Syslog service. By default a Syslog server listens on UDP port 514.

Task 5

Configure R5 so that all XML formatted system logging messages are sent to a Syslog server located at 1.1.1.1, while all ESM-filetered logging messages with the stream 20 value are sent to the Syslog server located at 2.2.2.2.

On R5

```
R5(config)#Logging host 1.1.1.1 xml
R5(config)#Logging host 2.2.2.2 filtered stream 20
```

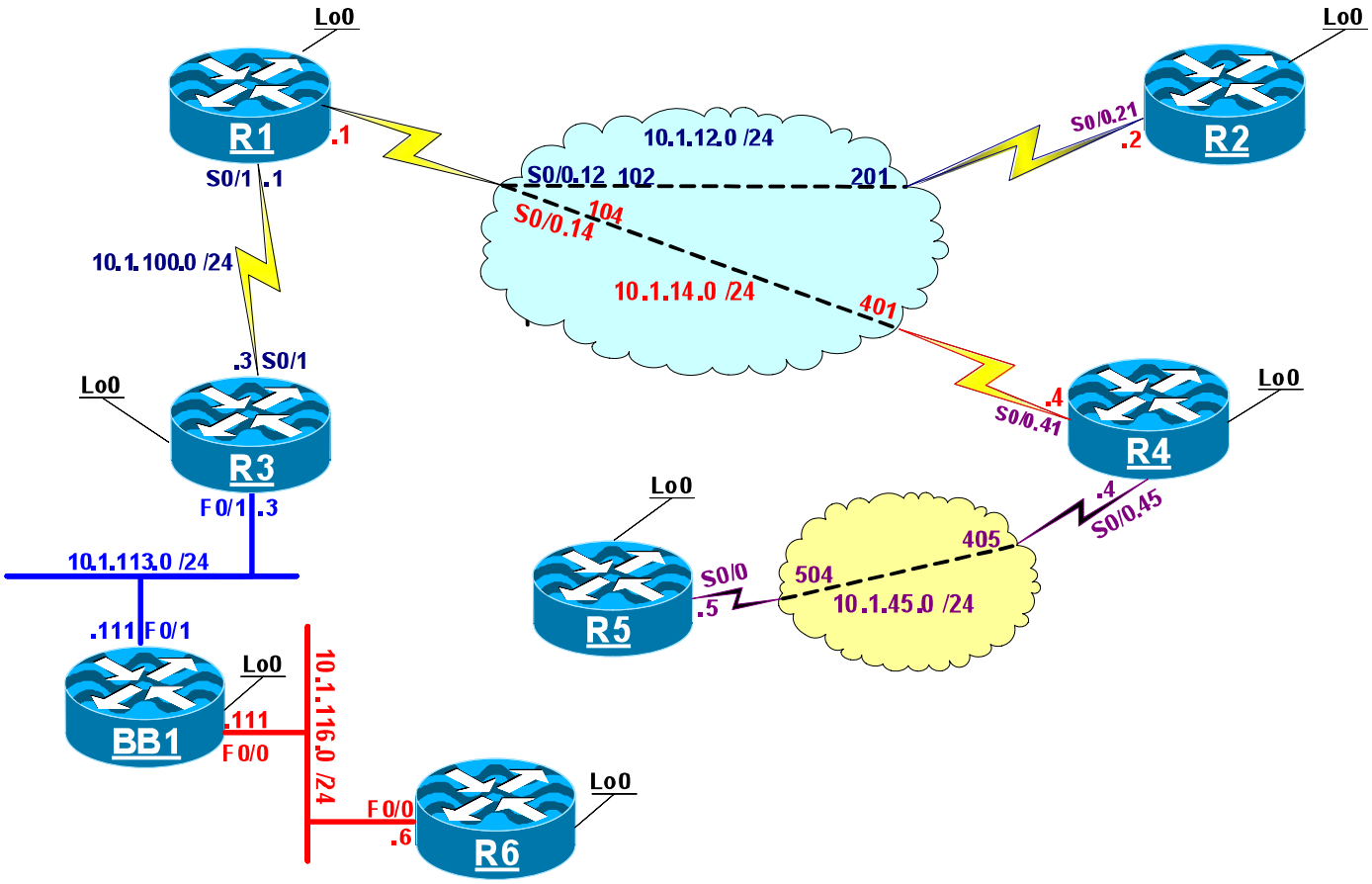
Starting IOS version 12.3(2)T, system messages can be logged independently as standard, XML-Formatted, or ESM-Filtered messages. For example, you could enable standard logging to the console connection, XML-formatted messages logging to the buffer and ESM-filtered messages logging to the monitor, and each type of output could be sent to a different destination.

The Embedded Syslog Manager (ESM) is a feature that is integrated with Cisco's IOS that allows you to have complete control over system messages at the source. For example: Customization, Severity Escalation for key messages, Message limiting and etc.

Task 6

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 2 – IP Accounting



Lab setup:

Configure the routers and switches according to the above diagram.

Run Eigrp AS 100 on all routers and advertise their doirectly connected interface/s in this routing domain.

Task 1

Enable IP accounting on R3 such that it takes a count of number of packets, Byte count, Source and destination IP address sent by any source destined to R1, R2, R4 and/or R5 in this topology.

IP accounting can collect statistics on traffic flows. Once IP accounting is enabled, packet's source and destination, number of packets and Byte counts are collected as packets are transmitted between two IP devices.

The following identifies two important aspects of IP accounting:

1. **ONLY occurs on Outbound interfaces**
2. **ONLY transit IP traffic is measured, traffic to and from the configured router is NOT measured.**
3. **IP accounting automatically disables Autonomous Switching, SSE (Silicon Switching Engine) switching, and DCEF on the interface that IP accounting is enabled on. This can cause peformance degradation.**

On R3

```
R3(config)#int s0/1
R3(config-if)#IP Accounting
```

If NO keywords are used, the "Output-packets" keyword is the default.

To test and verify the configuration:

On R6

```
R6#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

R6#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 128/128/128 ms

R6#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 = 80/80/84 ms
```

On R3

```
R3#Show ip accounting

   Source            Destination                Packets                Bytes
 10.1.116.6        2.2.2.2                          5                  500
 10.1.116.6        5.5.5.5                          5                  500
 10.1.116.6        1.1.1.1                          5                  500

Accounting data age is 28
```

We can see that R3 keeps a count of Source/Destination IP addresses, number of Packets, and Bytes count.

Remember that packets sourcing from R3 and/or destined to R3 are NOT measured, the following tests this condition:

On R3

```
R3#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 124/127/128 ms
```

On R6

```
R6#Ping 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

NOTE: The output of the following Show command reveals that traffic destined to or sourced from R3 is NOT measured:

```
R3#Show ip accounting

   Source            Destination                Packets                Bytes
```

```
 10.1.116.6       2.2.2.2                          5                  500
 10.1.116.6       5.5.5.5                          5                  500
 10.1.116.6       1.1.1.1                          5                  500

Accounting data age is 34
```

Task 2

Clear the accounting information on R3 and configure this router such that the number of accounting entries are limited to 3.

To clear the accounting entries on R3:

On R3

```
R3#Clear ip accounting
```

To verify the configuration:

On R3

```
R3#Show ip accounting
   Source            Destination                Packets                Bytes

Accounting data age is 0
```

To implement the new policy:

On R3

```
R3(config)#IP accounting-threshold 3
```

The above command sets the maximum number of accounting entries to be created to three.

To test the configuration:

Let's generate traffic from R6 and BB1:

On R6

```
R6#Ping 10.1.100.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.100.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

R6#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 124/127/128 ms
```

NOTE: ONLY two entries are created:

```
R3#Show ip accounting

   Source           Destination      Packets            Bytes
 10.1.116.6       10.1.100.1             5              500
 10.1.116.6       5.5.5.5                5              500

Accounting data age is 5
```

Let's create another entry:

On BB1

```
BB1#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 = 80/80/84 ms
```

On R3

```
R3#Show ip accounting

   Source           Destination      Packets            Bytes
 10.1.113.111     2.2.2.2                5              500
 10.1.116.6       10.1.100.1             5              500
 10.1.116.6       5.5.5.5                5              500

Accounting data age is 6
```

NOTE: We have reached the threshold because we have three entries in the accounting table, let's create one more and see the result:

On BB1

```
BB1#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 124/132/152 ms
```

On R3

```
R3#Show ip accounting

   Source             Destination                Packets                Bytes
 10.1.113.111       2.2.2.2                          5                  500
 10.1.116.6         10.1.100.1                       5                  500
 10.1.116.6         5.5.5.5                          5                  500

Accounting data age is 6
Accounting threshold exceeded for 5 packets and 500 bytes
```

The last ping generated from BB1 is NOT in the accounting table, and the output of the above show command states the reason.

Task 3

Enable IP accounting on R4 such that it takes a count of number of packets, Byte count, Source and destination IP addresses sent by any source from 10.1.113.0 /24 subnet destined to R5's directly connected interfaces.

The following command adds all hosts within the defined subnet to the list of hosts for which the accounting information will be kept:

On R4

```
R4(config)#IP accounting-list 10.1.113.0 0.0.0.255

R4(config)#Int S0/0.45
R4(config-subif)#IP accounting
```

To test and verify the configuration:

On BB1

```
BB1#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 124/127/128 ms

BB1#Ping 10.1.45.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.45.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 124/127/128 ms
```

To verify the result:

On R4

```
R4#Show ip accounting

   Source             Destination                Packets                Bytes
 10.1.113.111       5.5.5.5                          5                  500
 10.1.113.111       10.1.45.5                        5                  500

Accounting data age is 2
```

Since BB1 is using a source IP address from 10.1.113.0/24 network, and it matches the filter that you configured using the "IP Accounting-list", R4 will keep track of the Source/Destination IP, Number of packets and the Byte count.

To generate a Ping from R6 and test:

On R6

```
R6#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 128/128/128 ms
```

On R4

```
R4#Show ip accounting

   Source            Destination                Packets                Bytes
 10.1.113.111      5.5.5.5                          5                  500
 10.1.113.111      10.1.45.5                        5                  500

Accounting data age is 3
```

NOTE: R6 is NOT on the filtered subnet, therefore, the parameters of this traffic will NOT be in the accounting table.

Task 4

The policy of the company changed, without removing any command/s from R4, configure R4 so it overrides the policy that was implemented in the previous task. Limit the number of entries that does NOT match the filter configured in the previous task to 100.

This task can be accomplished using the "IP accounting-transits" global configuration command, transit entries are those that do NOT match any of the filters specified by R4:

```
R4(config)#IP accounting-transits 100
```

To test and verify the configuration:

On R6

```
R6#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 124/127/128 ms
```

To verify the accounting information:

On R4

```
R4#Show ip accounting
```

```
   Source           Destination           Packets               Bytes
 10.1.113.111     5.5.5.5                     5                 500
 10.1.113.111     10.1.45.5                   5                 500
 10.1.116.6       5.5.5.5                     5                 500
```

On R6

```
R6#Ping 10.1.45.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.45.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 124/128/132 ms
```

On R4

```
R4#Show ip accounting

   Source           Destination           Packets               Bytes
 10.1.113.111     5.5.5.5                     5                 500
 10.1.113.111     10.1.45.5                   5                 500
 10.1.116.6       10.1.45.5                   5                 500
 10.1.116.6       5.5.5.5                     5                 500

Accounting data age is 19
```

On R1

```
R1#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 96/98/100 ms

R4#Show ip accounting

   Source           Destination           Packets               Bytes
 10.1.14.1        5.5.5.5                     7                 700
 10.1.113.111     5.5.5.5                     5                 500
 10.1.113.111     10.1.45.5                   5                 500
 10.1.116.6       10.1.45.5                   5                 500
 10.1.116.6       5.5.5.5                     5                 500

Accounting data age is 21
```

Task 5

Enable IP accounting on BB1 such that it takes a count of Mac-address/es, number of packets, Byte count, of any traffic sourcing from R6 and destined to any interface on routers R1 – R5.

To configure this task, the "Mac-address" keyword can be used. When using the "Mac-address" keyword, there are two options:

1. **Input – Performs accounting based on the source Mac-address on received packets.**
2. **Output - Performs accounting based on the destination Mac-address on transmitted packets.**

On BB1

```
BB1(config)#Int F0/0
BB1(config-if)#IP accounting mac-address Input
```

To verify and test the configuration:

On R6

```
R6#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 124/127/128 ms
```

Let's check BB1:

On BB1

```
BB1#Show interfaces mac-accounting

FastEthernet0/0
  Input   (511 free)
    0000.6666.6666(0   ):   8 packets, 792 bytes, last: 1464ms ago
                  Total:   8 packets, 792 bytes
```

You are probably looking at the counters in the output of the above command and wondering as to why does it show 8 packets when we only sent five ICMP echos?

The answer is Eigrp hellos.

Let's repeat the show command few times and check the number of packets:

```
BB1#Show interfaces mac-accounting | i Total

                    Total:  48 packets, 3752 bytes

BB1#Show interfaces mac-accounting | i Total

                    Total:  49 packets, 3826 bytes
```

To prove this further, let's remove Eigrp from R6 and configure a static default route pointing to BB1:

On R6

```
R6(config)#NO router eigrp 100

R6(config)#IP route 0.0.0.0 0.0.0.0 10.1.116.111
```

Let's test the scenario again and check the results:

On R6

```
R6#Ping 5.5.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 5.5.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 124/127/128 ms

BB1#Show interfaces mac-accounting | i Total

                    Total:  71 packets, 5654 bytes

BB1#Show interfaces mac-accounting | i Total

                    Total:  71 packets, 5654 bytes

BB1#Show interfaces mac-accounting | i Total

                    Total:  71 packets, 5654 bytes

BB1#Show interfaces mac-accounting | i Total

                    Total:  71 packets, 5654 bytes
```

Let's remove the static default route and configure eigrp 100 on R6:

On R6

```
R6(config)#NO ip route 0.0.0.0 0.0.0.0

R6(config)#Router eigrp 100
R6(config-router)#No au
R6(config-router)#Network 0.0.0.0
```

Task 6

Configure R5 such that all Outbound ICMP traffic is marked with an IP Precedence value of 5.

On R5

```
R5(config)#Access-list 100 permit icmp host 10.1.45.5 any
R5(config)#Access-list 100 permit icmp host 5.5.5.5 any

R5(config)#Class-map R5
R5(config-cmap)#Match access-group 100

R5(config)#Policy-map TST
R5(config-pmap)#Class R5
R5(config-pmap-c)#Set ip precedence 5

R5(config)#Int S0/0
R5(config-if)#Service-policy Out TST
```

Task 7

Enable IP accounting on R4 such that it takes a count of number of packets, Byte count, Source and destination IP address sent by R5 to any IP address in this topology. Since R5 marks all of it's outbound traffic with an IP precedence level of 5, this accounting should be done based on the precedence level.

To configure this task, the "Precedence" keyword can be used. When using the "Precedence" keyword, there are two options:

1. **Input – Performs accounting based on the received packets.**

2. **Output - Performs accounting based on the transmitted packets.**

This option supports CEF, DCEF, flow and optimum switching.

On R4

```
R4(config)#Int S0/0.45
R4(config-subif)#IP Accounting precedence Input
```

NOTE: The above command could have been configured on R4's S0/0.41 sub-interface with an "Output" option.

On R5

```
R5#Ping 1.1.1.1 repeat 10

Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 96/99/100 ms
```

On R4

```
R4#Show interfaces precedence
Serial0/0.45
  Input
    Precedence 5:  10 packets, 1040 bytes
    Precedence 6:  2 packets, 128 bytes
```

What is Precedence 6? You guessed it, Eigrp.

Task 8

Configure an access-list on R1 so that it ONLY allows eigrp and Telnet traffic to any destination/s out of it's S0/1 interface.

```
R1(config)#Access-list 100 permit tcp any any eq Telnet
R1(config)#Access-list 100 permit eigrp any any
R1(config)#Access-list 100 deny ip any any log

R1(config)#Int S0/1
R1(config-if)#IP access-group 100 Out
```

Task 9

Configure R1 such that it identifies IP traffic that failed the access-list configured in the previous task.

This task can be accomplished by using the "Access-Violation" keyword. This keyword identifies the IP traffic that FAILED the access-list. A very handy tool that can alert you of some attacks.

On R1

```
R1(config)#Int S0/1
R1(config-if)#IP Accounting access-violations
```

On R2

```
R2#Ping 6.6.6.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)
```

On R1

```
R1#Show access-list

Extended IP access list 100
    10 permit tcp any any eq telnet
    20 permit eigrp any any
    30 deny ip any any log (5 matches)
```

You should also see the following console message:

```
%SEC-6-IPACCESSLOGDP: list 100 denied icmp 10.1.12.2 -> 6.6.6.6 (8/0), 4
packets
```

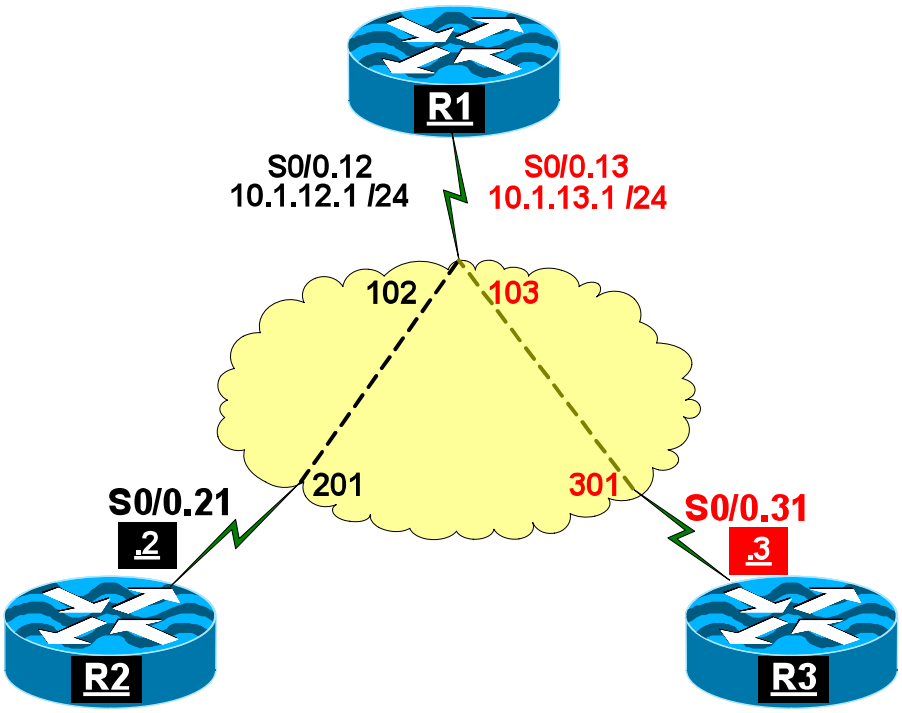
On R1

```
R1#Show ip accounting access-violations

   Source             Destination        Packets                 Bytes    ACL
 10.1.12.2          6.6.6.6                  5                   500    100

Accounting data age is 9
```

Task 10

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 3 – IP SLA



Lab Setup:

- Configure the Frame-Relay connection in a Point-to-Point manner.
- Use the IP addressing chart below for IP address assignment.

IP addressing Chart:

Router	Interface / IP addressing
R1	S0/0.12 = 10.1.12.1 /24 S0/0.13 = 10.1.13.1 /24
R2	S0/0.21 = 10.1.12.2 /24
R3	S0/0.31 = 10.1.13.3 /24

Task 1

Configure R1 and R2 so that they use UDP Echo packets to perform and determine the end-to-end response time. In this operation R1 should send these packets and R2 should be configured to respond to the packets with a time-stamp such that R1 can calculate the round trip time. This test should be performed for 30 seconds using UDP port 12000.

Using Cisco IP SLA, the performance of the network can be monitored; this can be performed without deploying a physical probe. A router can be configured to send a generated packet to the destination device and once the destination device receives this packet, the device will respond with time-stamp information for the source so the source can make the calculation on performance metric.
The UDP Echo operation measures end-to-end response time between a Cisco router and devices using IP.
In this task, R2 should be configured as an IP SLA responder, a responder actually responds to Cisco IP SLA's request packets.

On R2:

```
R2(config)#ip sla responder
```

To verify the configuration:

On R2:

```
R2#Sh ip sla responder

IP SLAs Responder is: Enabled
Number of control message received: 0 Number of errors: 0
Recent sources:
Recent error sources:
```

Note: The responder is enabled.

On R1:

```
R1(config)#IP sla 10
R1(config-sla)#udpecho 10.1.12.2 12000
R1(config-sla-monitor-udp)#frequency 5
```

Note: The above commands configure a UDP ECHO to be sent to destination IP address of 10.1.12.2 (R2) to UDP port number 12000 every 5 seconds.
The following configures the scheduling parameters for the SLA operation to start immediately and continue for 30 seconds ONLY. Note the numeric value (10) after the "IP SLA schedule" command should match the number configured in the "IP SLA" command above.

```
R1(config)#ip sla schedule 10 start-time now life 30
```

To test the configuration:

On R2:

```
R2#Show ip sla responder

IP SLAs Responder is: Enabled
Number of control message received: 6 Number of errors: 0
Recent sources:
        10.1.12.1 [13:46:45.007 UTC Tue Jul 13 2010]
        10.1.12.1 [13:46:40.007 UTC Tue Jul 13 2010]
        10.1.12.1 [13:46:35.007 UTC Tue Jul 13 2010]
        10.1.12.1 [13:46:30.007 UTC Tue Jul 13 2010]
        10.1.12.1 [13:46:25.007 UTC Tue Jul 13 2010]
Recent error sources:
```

On R1:

The RTT time may vary in your test.

```
R1#Sh ip sla statistics

Round Trip Time (RTT) for       Index 10
        Latest RTT: 25 milliseconds
Latest operation start time: *01:45:22.459 UTC Fri Mar 8 2002
Latest operation return code: OK
Number of successes: 6
Number of failures: 0
Operation time to live: 0
```

Task 2

Reconfigure the previous task to send packets with 1500 Bytes in size.

On R1:

```
R1(config)#No ip sla 10

R1(config)#ip sla 10
R1(config-sla-monitor)#udpecho 10.1.12.2 12000
R1(config-sla-monitor-udp)#frequency 5
```

```
R1(config-sla-monitor-udp)#request-data-size 1500
```

Note: The "request-data-size" command can be used to set the packet size.

Finally, the scheduling is invoked as follows:

```
R1(config)#ip sla schedule 10 start-time now life 30
```

To verify the configuration:

On R1:

```
R1#Sh ip sla statistics

Round Trip Time (RTT) for       Index 10
        Latest RTT: 686 milliseconds
Latest operation start time: *01:55:08.299 UTC Fri Mar 8 2002
Latest operation return code: OK
Number of successes: 6
Number of failures: 0
Operation time to live: 0
```

Note: The packet size was increased to 1500 bytes therefore, the RTT was affected; once again remember that your RTT may vary.

To verify the configuration:

On R1:

```
R1#Show ip sla configuration

IP SLAs Infrastructure Engine-II
Entry number: 10
Owner:
Tag:
Type of operation to perform: udp-echo
Target address/Source address: 10.1.12.2/0.0.0.0
Target port/Source port: 12000/0
Request size (ARR data portion): 1500
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Verify data: No
Data pattern:
Vrf Name:
Control Packets: enabled
```

```
Schedule:
   Operation frequency (seconds): 5  (not considered if randomly
scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 30
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 4294967295
Enhanced History:
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
```

Task 3

Configure R3 to measure the response time taken to perform a TCP Connect operation between R3 and R1. R3 should be configured to generate TCP Connect messages, whereas, R1 should be configured such that it enhances the accuracy of the connection response time.

This task can be accomplished by configuring the IP SLAs TCP Connect operation, this operation is used to measure the response time taken to perform a TCP connect operation. To enhance the accuracy of the response time R1 should be configured as an IP SLA responder.

On R1:

```
R1(config)#ip sla responder
```

On R3:

```
R3(config)#ip sla 30
R3(config-ip-sla)#tcp-connect 10.1.13.1 23
R3(config-ip-sla-tcp)#timeout 1000
R3(config-ip-sla-tcp)#frequency 5
```

```
R3(config)#ip sla schedule 30 life forever start-time now
```

To verify the configuration:

On R1:

```
R1#Sh ip sla responder

IP SLAs Responder is: Enabled
Number of control message received: 7 Number of errors: 0
Recent sources:
        10.1.13.3 [02:01:50.791 UTC Fri Mar 8 2002]
        10.1.13.3 [02:01:45.791 UTC Fri Mar 8 2002]
        10.1.13.3 [02:01:40.791 UTC Fri Mar 8 2002]
        10.1.13.3 [02:01:35.791 UTC Fri Mar 8 2002]
        10.1.13.3 [02:01:30.791 UTC Fri Mar 8 2002]
Recent error sources:
```

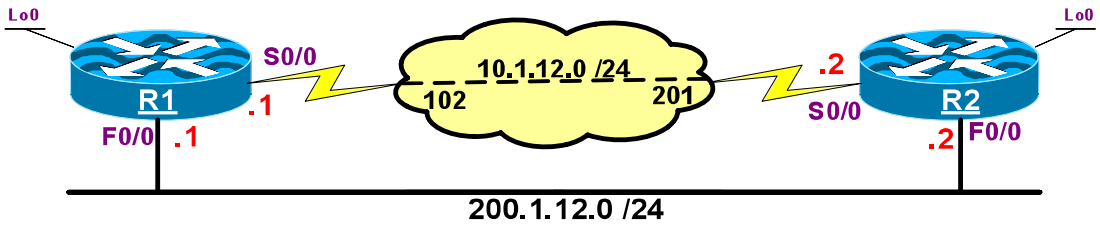
On R3:

```
R3#Show ip sla statistics

Round Trip Time (RTT) for       Index 30
        Latest RTT: 28 milliseconds
Latest operation start time: *19:49:08.671 UTC Wed Jan 14 2009
Latest operation return code: OK
Number of successes: 15
Number of failures: 0
Operation time to live: Forever
```

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab – 4
Reliable Static Routing using IP SLA



Lab Setup

- The frame-relay connection on R1 and R2 should be configured directly under the physical interface.
- The F0/0 interface of R1 and R2 should be configured in VLAN 12
- Run OSPF Area 0 on all interfaces of R1 and R2, loopback interfaces should be advertised using their correct mask.
- Use the IP addressing chart below for IP addressing assignment

IP addressing Chart:

Router	Interface / IP addressing
R1	S0/0 = 10.1.12.1 /24 F0/0 = 200.1.12.1 /24 Lo0 = 1.1.1.1 /8
R2	**S0/0 = 10.1.12.2 /24** **F0/0 = 200.1.12.2 /24** **Lo0 = 2.2.2.2 /8**

Task 1

Configure two static routes on R1 to reach R2's loopback. The configuration should be such that if R1's frame-relay connection to R2 is reliably working, it should be the preferred path, but if R1 cannot reach R2 through the frame-relay cloud, R1 should take the path through its F0/0 interface. **Do not** use EEK, backup interface, or PPP to accomplish this task.

To accomplish this task two floating static routes are configured as follows:

On R1:

```
R1(config)#ip route 2.0.0.0 255.0.0.0 10.1.12.2 50
R1(config)#ip route 2.0.0.0 255.0.0.0 200.1.12.2 100
```

To verify the configuration:

On R1:

```
R1#Show ip route | b Gate

Gateway of last resort is not set

C    1.0.0.0/8 is directly connected, Loopback0
S    2.0.0.0/8 [50/0] via 10.1.12.2
C    200.1.12.0/24 is directly connected, FastEthernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.12.0 is directly connected, Serial0/0
```

To test the configuration:

On R2:

To test this configuration, the S0/0 interface of R2 is shut down and the routing table of R1 is checked, then, a ping is generated from R1:

```
R2(config)#int s0/0
R2(config-if)#shutdown
```

On R1:

```
R1#Show ip route | b Gate

Gateway of last resort is not set
```

```
C    1.0.0.0/8 is directly connected, Loopback0
S    2.0.0.0/8 [50/0] via 10.1.12.2
C    200.1.12.0/24 is directly connected, FastEthernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.12.0 is directly connected, Serial0/0
```

On R1:

```
R1#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 0 percent (0/5)
```

Note: This configuration did not accomplish the requirements of this task, because shutting down the S0/0 interface of R2 did not affect R1 at ALL.

IP SLA ICMP ECHO can be used to monitor end-to-end response time between a Cisco router and another IP device, in this case, another Cisco router.

In the following configuration R1 is configured to generate "icmp-echo" messages to the destination IP address of 10.1.12.2, the source IP address of these messages is set to 10.1.12.1. The timeout keyword specifies the amount of time an IP SLA operation waits for a response from its request packets. In this case, the timeout is set to 500 milliseconds.

The "frequency" keyword sets the rate at which the specified IP SLAs operation is repeated.

On R1:

```
R1(config#ip sla 1
R1(config-ip-sla)#icmp-echo 10.1.12.2 source-ip 10.1.12.1
R1(config-ip-sla-echo)#timeout 500
R1(config-ip-sla-echo)#frequency 3
```

The above configuration is not enough for the router to generate the messages specified in the configuration; therefore, the router needs to be configured to start the above configuration operation.

The following configuration starts IP SLA operation 1, immediately with a life of the operation set to forever.

```
R1(config)#ip sla schedule 1 start-time now life forever
```

In the second last step of this configuration, the state of IP SLA operation is tracked for reachability:

```
R1(config)#track 2 rtr 1 reachability
```

The last step of this configuration, object tracking 2 is assigned to the primary static route:

```
R1(config)#No ip route 2.0.0.0 255.0.0.0 10.1.12.2 50

R1(config)#ip route 2.0.0.0 255.0.0.0 10.1.12.2 50 track 2
```

To verify the configuration of the two static routes:

On R1:

```
R1#Sh run | i ip route

ip route 2.0.0.0 255.0.0.0 10.1.12.2 50 track 2
ip route 2.0.0.0 255.0.0.0 200.1.12.2 100
```

Enable S0/0 interface of R2:

On R2:

```
R2(config)#int s0/0
R2(config-if)#No shutdown
```

To verify the configuration:

On R1:

Once the following console message is displayed:

```
%TRACKING-5-STATE: 2 rtr 1 reachability Down->Up

R1#Show track 2

Track 2
  Response Time Reporter 1 reachability
  Reachability is Up
    1 changes, last change 00:00:04
  Latest operation return code: OK
  Latest RTT (millisecs) 39
  Tracked by:

    STATIC-IP-ROUTING 0
```

Note: The rtr 1 is configured based on reachability, and the last operation was successful with a RTT of 39 ms.

```
R1#Show ip route | b Gate

Gateway of last resort is not set

C    1.0.0.0/8 is directly connected, Loopback0
S    2.0.0.0/8 [50/0] via 10.1.12.2
C    200.1.12.0/24 is directly connected, FastEthernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.12.0 is directly connected, Serial0/0
```

Since the frame-relay link is up and network 2.0.0.0 /8 is reachable through the frame-relay cloud, it is chosen as the best route.

```
R1#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 = 56/61/76 ms
```

To test the configuration:

On R2:

To test the configuration, the S0/0 interface of R2 is shut down:

```
R2(config)#int s0/0
R2(config-if)#shutdown
```

Note: Shutting down the S0/0 (the frame-relay) interface of R2 did not affect the Serial0/0 interface of R1:

On R1:

```
R1#Sh ip int br | inc Serial0/0_

Serial0/0              10.1.12.1       YES manual up                    up
```

You should see the following console message on R1

On R1:

```
%TRACKING-5-STATE: 2 rtr 1 reachability Up->Down
```

To test the operation of IP SLA/tracking:

On R1:

```
R1#Show track 2

Track 2
  Response Time Reporter 1 reachability
  Reachability is Down
    3 changes, last change 00:02:55
  Latest operation return code: Timeout
  Tracked by:
    STATIC-IP-ROUTING 0
```

Note: Because reachability is down (the IP SLA operation can not send icmp-echo messages to 10.1.12.2), the static route to network 2.0.0.0 /8 with an administrative distance of 50 is removed and the static route with an administrative distance of 100 is injected into the routing table.

To verify the configuration:

On R1:

```
R1#Show ip route | b Gateway

Gateway of last resort is not set

C    1.0.0.0/8 is directly connected, Loopback0
S    2.0.0.0/8 [100/0] via 200.1.12.2
C    200.1.12.0/24 is directly connected, FastEthernet0/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.12.0 is directly connected, Serial0/0
```

Note: The following traceroute & ping command reveals that the reachability is now through the fast ethernet interface.

```
R1#Traceroute 2.2.2.2

Type escape sequence to abort.
Tracing the route to 2.2.2.2
  1 200.1.12.2 0 msec *  0 msec

R1#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 = 1/2/4 ms
```

Note: Because of the floating static route and the IP SLA configuration, if the S0/0 interface of R2 is brought back up, the track 2 reachability will be UP, therefore, the primary static route will be injected back into the routing table and the backup static route will be removed.

To test the configuration:

On R2:

```
R2(config)#int s0/0
R2(config-if)#No shut
```

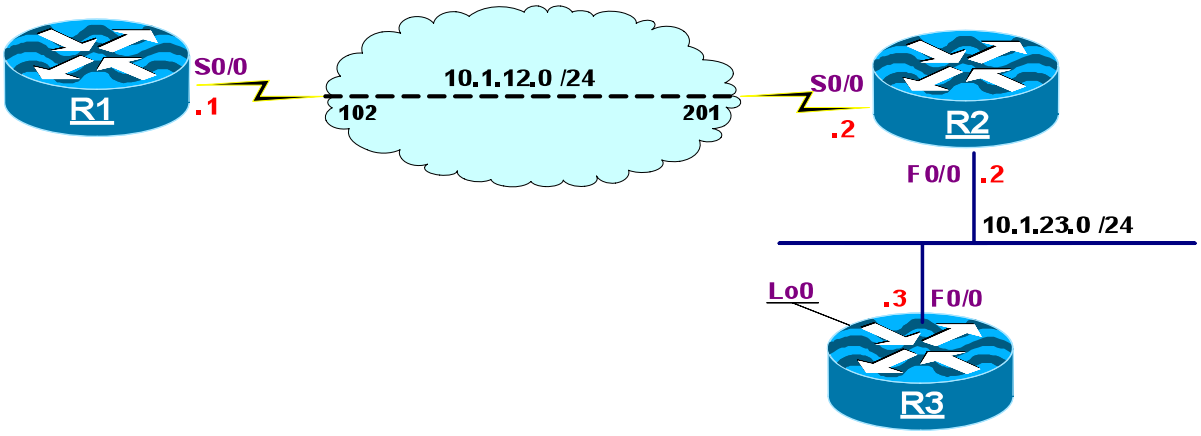
On R1:

```
R1#Sh ip route | inc 2.0.0.0

S    2.0.0.0/8 [50/0] via 10.1.12.2

R1#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 = 1/2/4 ms
```

Task 2

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab – 5
Reliable Conditional Default Route Injection Using IP SLA



Lab Setup:

- The frame-relay connection on R1 and R2 should be configured directly under the physical interface.
- The F0/0 interface of R2 and R3 should be configured in VLAN 23
- Use the IP addressing chart below for IP addressing assignment

IP addressing Chart:

Router	Interface / IP addressing
R1	S0/0 = 10.1.12.1 /24
R2	**S0/0 = 10.1.12.2 /24** **F0/0 = 10.1.23.2 /24**
R3	F0/0 = 10.1.23.3 /24 Lo0 = 3.3.3.3 /24

Task 1

Configure RIPv2 on the link that connects R1 to R2 and OSPF area 0 between R2 and R3. R3 should run OSPF on all of its directly connected interfaces.

On R1: & R2

```
Rx(config-if)#router rip
Rx(config-router)#No auto
Rx(config-router)#ver 2
Rx(config-router)#net 10.0.0.0
```

On R2:

```
R2(config)#router rip
R2(config-router)#passive-interface F0/0

R2(config)#router ospf 1
R2(config-router)#netw 10.1.23.2 0.0.0.0 area 0
```

On R3:

```
R3(config)#router ospf 1
R3(config-router)#netw 3.3.3.3 0.0.0.0 area 0
R3(config-router)#netw 10.1.23.3 0.0.0.0 area 0
```

To verify the configuration:

On R2:

```
R2#Show ip route ospf | inc O

O       3.3.3.3 [110/11] via 10.1.23.3, 00:07:02, FastEthernet0/0
```

On R1:

```
R1#Show ip route rip | inc R

R       10.1.23.0 [120/1] via 10.1.12.2, 00:00:26, Serial0/0
```

Task 2

Configure R2 to advertise a default route into OSPF routing domain. The default route should **only** be injected if R2 and R1 have reachability through the frame-relay cloud. You are **not** allowed to use static route/s or IP SLA to accomplish this task.

Since the use of IP SLA and/or static route/s are prohibited, PPP is used to accomplish this task; when configuring PPP on any link, a host route is injected, therefore, the host route can be identified by an access-list, and the access-list is referenced in the route-map, and finally the route-map is referenced by "default-information originate" router configuration command:

Step 1:
The following configures PPP on frame-relay:

On R1: and R2

```
R2(config)#int s0/0
R2(config-if)#No ip addr
```

On R1:

```
R1(config)#int virtual-template 12
R1(config-if)#ip addr 10.1.12.1 255.255.255.0

R1(config-if)#int s0/0
R1(config-if)#frame-relay interface-dlci 102 ppp virtual-template 12
```

On R2:

```
R2(config)#int virtual-template 21
R2(config-if)#ip addr 10.1.12.2 255.255.255.0

R2(config-if)#int S0/0
R2(config-if)#frame-relay interface-dlci 201 ppp virtual-template 21
```

To verify the configuration:

On R1:

Note: The output of the following show command reveals the host route that is injected by PPP:

```
R1#Sh ip route | inc /32

C       10.1.12.2/32 is directly connected, Virtual-Access2
```

On R2:

```
R2#Sh ip route | inc /32

     3.0.0.0/32 is subnetted, 1 subnets
C       10.1.12.1/32 is directly connected, Virtual-Access2
```

Routers R1 and R2 are exchanging RIPv2 routes:

On R1:

```
R1#Show ip route rip | inc R

R       10.1.23.0/24 [120/1] via 10.1.12.2, 00:00:17, Virtual-Access2
```

Step 2:
An access-list is configured to reference the host route generated by PPP:

```
R2(config)#access-list 1 permit host 10.1.12.1
```

Step 3:
A route-map is configured to reference the access-list:

```
R2(config)#route-map TST permit 10
R2(config-route-map)#match ip addr 1
```

Step 4:
In this final step a "default-information originate" is configured referencing the route-map:

```
R2(config-route-map)#router ospf 1
R2(config-router)#default-information originate route-map TST
```

To verify the configuration:

On R3:

```
R3#Show ip route ospf | inc O

O*E2 0.0.0.0/0 [110/1] via 10.1.23.2, 00:00:51, FastEthernet0/0
```

To test the configuration:

Note once R1's S0/0 goes down, the host route is removed and the condition of the route-map TST is no longer true, therefore, the default route is removed.

```
R1(config)#int s0/0
R1(config-if)#shut
```

On R3:

```
R3#Show ip route ospf | inc O
R3#
```

To test this condition further:

On R1:

```
R1(config)#int s0/0
R1(config-if)#No shut
```

On R3:

```
R3#Show ip route ospf | inc O

O*E2 0.0.0.0/0 [110/1] via 10.1.23.2, 00:00:08, FastEthernet0/0
```

Task 3

Re-configure R2 to advertise a default route into OSPF routing domain. The default route should **only** be injected if R2 and R1 have reachability through the frame-relay cloud. You should use IP SLA to accomplish this task.

On R2:

Step one:
The access-list, route-map and the default-information originate commands are removed:

```
R2(config)#No access-list 1
R2(config)#No route-map TST

R2(config)#router ospf 1
R2(config-router)#No default-information originate
```

Configuring IP SLA to generate IP ICMP Echo messages, the timeout and frequency can be set to any value:

```
R2(config)#ip sla 10
R2(config-sla-monitor)#icmp-echo 10.1.12.1 source-ip 10.1.12.2
R2(config-sla-monitor-echo)#timeout 250
R2(config-sla-monitor-echo)#frequency 5
```

Note: Even though the IP SLA is configured, it won't start unless it's configured to do so; when starting the operation, the start-time and life of these messages are defined:

```
R2(config)#ip sla schedule 10 start-time now life forever
```

The IP SLA operation is tracked in track 2 for reachability:

```
R2(config)#track 2 rtr 10 reachability
```

Since the track is referenced in the following default route, if the IP SLA operation fails, track 2 will go down and the default route is removed. Remember, in order to originate a default route in OSPF, the local router must have a default route or else it won't generate one:

```
R2(config)#ip route 0.0.0.0 0.0.0.0 null0 track 2

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

To test the configuration:

On R2:

```
R2#Show track 2

Track 2
  Response Time Reporter 10 reachability
  Reachability is Up
    1 change, last change 00:02:12

  Latest operation return code: OK
  Latest RTT (millisecs) 32
  Tracked by:
    STATIC-IP-ROUTING 0
```

Note: Reachability is UP, therefore, the default route should be present in R3's routing table:

On R3:

```
R3#Show ip route ospf
```

```
O*E2 0.0.0.0/0 [110/1] via 10.1.23.2, 00:17:38, FastEthernet0/0
```

To test this configuration:

1. Serial0/0 interface of R1 is Shutdown.
2. The state of track 2 is checked on R2; the state must be down.
3. The routing table of R3 is checked; it should **not** have a default route.
4. Serial0/0 interface of R1 is enabled.
5. The state of track 2 is checked on R2, the state must be UP.
6. The routing table of R3 is checked, it should have a default route.

Step one:

On R1:

```
R1(config)#int s0/0
R1(config-if)#shut
```

Step Two:

On R2:

Enter the following "Show track 2" command after the following console message is displayed:

%TRACKING-5-STATE: 2 rtr 10 reachability Up->Down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access2, changed state to down

```
R2#Show track 2

Track 2
  Response Time Reporter 10 reachability
  Reachability is Down
    2 changes, last change 00:01:36
  Latest operation return code: Timeout
  Tracked by:
    STATIC-IP-ROUTING 0
```

Step Three:

On R3:

```
R3#Show ip route ospf
R3#
```

Step Four:

On R1:

```
R1(config)#int s0/0
R1(config-if)#No shut
```

Step Five:

Wait until the following console message is displayed:

%LINK-3-UPDOWN: Interface Virtual-Access2, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access2, changed state to up

%TRACKING-5-STATE: 2 rtr 10 reachability Down->Up

```
R2#Show track 2

Track 2
  Response Time Reporter 10 reachability
  Reachability is Up
    3 changes, last change 00:00:49
  Latest operation return code: OK
  Latest RTT (millisecs) 32
  Tracked by:
    STATIC-IP-ROUTING 0
```

Step Six:

On R3:

```
R3#Show ip route ospf

O*E2 0.0.0.0/0 [110/1] via 10.1.23.2, 00:01:11, FastEthernet0/0
```

Task 4

Configure R2 to advertise a default route into RIPv2 routing domain. The default route should **only** be injected if R2 has reachability to R3 through the switched connection. You should use IP SLA, **Do not** configure a static default route to accomplish this task.

Note: R2 will not be aware if the F0/0 interface of R3 goes down, therefore, if the F0/0 interface of R3 is down, the F0/0 interface of R2 will remain in UP/UP state.

To configure this injection of default route reliably, once again, the IP SLA operation is configured like the previous tasks, but the difference in this configuration is the following:

Since a static default route can not be configured, a fake static route is created so it can be utilized to accomplish this task. This static route can be for any network, this network does not exist; this static route is tracked by the IP SLA operation and referenced in a route-map, the route-map is referenced in the "default-information originate" router configuration command. Therefore, if R2 fails to reach R3's F0/0 IP address through the IP SLA operation, this static route is removed, if the static route is removed the condition of the route-map will not be true, therefore, the default route is removed.

On R2:

The following configures a fake static route; in this case, 9.9.9.9 /32 IP address is chosen:

```
R2(config)#ip route 9.9.9.9 255.255.255.255 null0
```

The following access-list is created to identify the fake static route:

```
R2(config)#access-list 1 permit host 9.9.9.9
```

A route-map is configured and access-list 1 is referenced:

```
R2(config)#route-map TST permit 10
R2(config-route-map)#match ip address 1
```

The following configuration instructs the router to inject a default route only if the condition of the route-map is true; the condition of the route-map can only be true if 3.3.3.3 /32 exists:

```
R2(config)#router rip
R2(config-router)#default-information originate route-map TST
```

To verify the configuration:

On R1:

Note: The default route is injected:

```
R1#Show ip route rip | inc /0

R*   0.0.0.0/0 [120/1] via 10.1.12.2, 00:00:13, Virtual-Access2
```

To test the configuration:

On R3:

The F0/0 interface of R3 is shutdown:

```
R3(config)#int f0/0
R3(config-if)#shut
```

Note: Even though the F0/0 interface of R3 is in shutdown mode, the default route is still injected:

```
R1#Sh ip route rip | inc /0

R*   0.0.0.0/0 [120/1] via 10.1.12.2, 00:00:09, Virtual-Access2
```

To inject a reliable default route, an IP SLA is configured to track the reachability of R3's F0/0 interface, this is called a reliable conditional default gateway injection, as follows:

```
R2(config)#ip sla 20
R2(config-sla-monitor)#icmp-echo 10.1.23.3 source-ip 10.1.23.2
R2(config-ip-sla-echo)#timeout 250
R2(config-ip-sla-echo)#frequency 3

R2(config)#ip sla schedule 20 start-time now life forever

R2(config)#track 1 rtr 20 reachability
```

The following command tracks the static route created earlier:

```
R2(config)#NO ip route 9.9.9.9 255.255.255.255

R2(config)#ip route 9.9.9.9 255.255.255.255 null 0 track 1
```

The F0/0 interface of R3 is re-enabled:

```
R3(config)#int f0/0
R3(config-if)#No shut
```

To verify the configuration:

On R2:

You should see the following console message:

```
R2#Show track 1
```

```
Track 1
  Response Time Reporter 20 reachability
  Reachability is Up
    2 changes, last change 00:00:40
  Latest operation return code: OK
  Latest RTT (millisecs) 1
  Tracked by:
    STATIC-IP-ROUTING 0

R1#Show ip route rip | inc /0

R*   0.0.0.0/0 [120/1] via 10.1.12.2, 00:00:22, Virtual-Access2
```

To test the configuration:

The following is how the test will be conducted:

- **F0/0 interface of R3 is shutdown.**
- **A "debug track" on R2 and "debug ip icmp" on R3 is configured.**
- **The routing table of R1 is checked; if the configuration was performed properly, R2 should remove the fake static route to 3.3.3.3/32, once this happens, the condition of the route-map (TST) is no longer true, therefore, the default route is removed and it will not be in the routing table of R1.**
- **The F0/0 interface of R3 is enabled (no shut), if the configuration was performed properly, R2 should inject the default route back into RIP routing domain.**

On R2:

```
R2#Debug track
```

On R3:

```
R3#Debug ip icmp
```

The interface is shutdown:

```
R3(config)#int f0/0
R3(config-if)#shut
```

Note: On R2 you should receive the following messages:

Track: 1 Change #3 rtr 20, reachability Up->Down

%TRACKING-5-STATE: 1 rtr 20 reachability Up->Down

%OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on FastEthernet0/0 from FULL to DOWN, Neighbor Down: Dead timer expired

The routing table of R1 is verified:

On R1:

```
R1#Show ip route rip | inc /0
R1#
```

The output of the above command reveals that R1 no longer has the default route in its routing table.

The F0/0 interface of R3 is brought back up:

On R3:

```
R3(config)#int f0/0
R3(config-if)#No shut
```

On R2:

You should receive the following console message:

Track: 1 Change #4 rtr 20, reachability Down->Up

%TRACKING-5-STATE: 1 rtr 20 reachability Down->Up

```
R2#Show track 1

Track 1
  Response Time Reporter 20 reachability
  Reachability is Up
    4 changes, last change 00:00:39
  Latest operation return code: OK
  Latest RTT (millisecs) 1

  Tracked by:
    STATIC-IP-ROUTING 0
```

The routing table of R1 is checked:

```
R1#Sh ip route rip | inc /0

R*      0.0.0.0/0 [120/1] via 10.1.12.2, 00:00:18, Virtual-Access2
```

Note: The default route is in the routing table of R1. This may take few seconds.

Task 5

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Advanced
CCIE SERVICE PROVIDER
v3.0

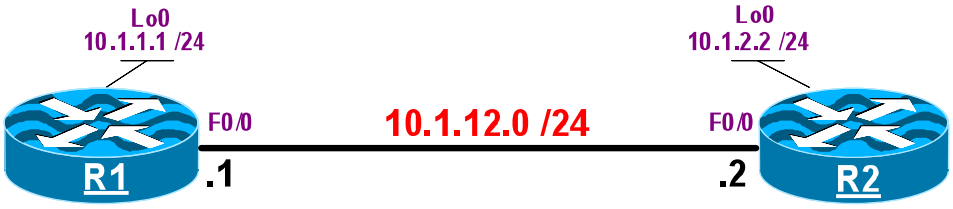
www.MicronicsTraining.com

Narbik Kocharians
CCIE #12410
R&S, Security, SP

Paul Negron
CCIE #14856
SP

QOS

Lab 1 – Priority Queuing



Lab setup:

- Configure the F0/0 interface of these routers in VLAN 12.
- Configure the routers based on the diagram.
- These routers should run RIPv2 and advertise their directly connected networks.

Task 1

Configure Priority Queuing On R1: using the following policy:

- All traffic generated from the Loopback0 interface of R2 should be assigned to the High queue.
- All IP traffic from R1's Loopback0 interface to R2's Loopback0 interface should be assigned to the Low queue.
- All HTTP traffic should be assigned to Medium queue.
- TFTP traffic should be assigned to the Normal queue.
- The rest of the traffic should be assign to Low queue.

Priority queuing uses 4 queues: high, medium, normal, and low. These queues have pre-assigned priority in the following order:

1. **High: This queue has the highest priority**
2. **Medium: This queue has the second highest priority**
3. **Normal: This queue has the third highest priority**

4. **Low: This queue has the lowest priority**

Packets in these queues are de-queued based on the following method:

De-queue the packets in the higher queue, until there are no more packets to de-queue. Only then, the system will service the next highest queue, this will continue, unless another packet arrives for a higher queue. Since PQ does not have a built-in policer, the traffic in a higher queue can consume the entire bandwidth, and therefore, starving the traffic in lower queues.

Priority queuing commands are processed like an access-list, they are read from top to the bottom. Once a match is found, the processing stops. Therefore, the order of the commands are extremely important. You should always configure the more specific items at the top and the general items toward the bottom.

On R1:

```
R1(config)#access-list 100 permit ip host 10.1.1.1 host 10.1.2.2
```

The above access-list identifies the IP communication between the Loopback0 interfaces of R1 and R2.

```
R1(config)#priority-list 1 protocol ip low list 100
R1(config)#priority-list 1 interface lo0 high
R1(config)#priority-list 1 protocol ip medium tcp 80
R1(config)#priority-list 1 protocol ip normal udp 69
R1(config)#priority-list 1 default low
```

The first line is assigning the more specific policy, which is the communication between Loopback0 interfaces of the routers.
The second line is the more general policy regarding the Loopback0 interface of R1. If this was configured in the reverse order, the communication between the Loopback0 interfaces would also be assigned to the High Queue.
The third and the fourth line could be entered in any order, because they are both identifying a specific protocol.
The last line is the catchall statement, which is applied to the rest of the traffic.

```
R1(config)#int f0/0
R1(config-if)#priority-group 1
```

Finally, the configuration is applied to the interface. Remember that queuing is always outbound.

To verify the configuration:

On R1:

```
R1#show queuing priority
```

```
Current DLCI priority queue configuration:
Current priority queue configuration:

List   Queue  Args
1      low    default
1      low    protocol ip          list 100
1      high   interface Loopback0
1      medium protocol ip          tcp port www
1      normal  protocol ip          udp port tftp
```

NOTE: The order may not be the order in which the commands were entered.

To test the configuration:

In order to reduce the amount of output in the debug, you should turn off CDP and the keepalives on the F0/0 interface.

On R1:

```
R1(config)#No cdp run

R1(config)#int f0/0
R1(config-if)#No keep
```

You should turn on debugging for the priority queuing and perform the following tests:

```
R1#Debug priority
```

In the first test, you should perform a ping command. This command should be sourced from the Loopback0 interface of R1 and destined for the Loopback0 interface of R2.

```
R1#ping
Protocol [ip]:→ press Enter
Target IP address: 10.1.2.2
Repeat count [5]: → press Enter
Datagram size [100]: → press Enter
Timeout in seconds [2]: → press Enter
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]: → press Enter
Set DF bit in IP header? [no]: → press Enter
Validate reply data? [no]: → press Enter
Data pattern [0xABCD]: → press Enter
Loose, Strict, Record, Timestamp, Verbose[none]: → press Enter
Sweep range of sizes [n]: → press Enter
```

```
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.2.2, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms

PQ: FastEthernet0/0: ip (s=10.1.1.1, d=10.1.2.2) -> low
PQ: FastEthernet0/0 output (Pk size/Q 114/3)
PQ: FastEthernet0/0: ip (s=10.1.1.1, d=10.1.2.2) -> low
PQ: FastEthernet0/0 output (Pk size/Q 114/3)
PQ: FastEthernet0/0: ip (s=10.1.1.1, d=10.1.2.2) -> low
PQ: FastEthernet0/0 output (Pk size/Q 114/3)
PQ: FastEthernet0/0: ip (s=10.1.1.1, d=10.1.2.2) -> low
PQ: FastEthernet0/0 output (Pk size/Q 114/3)
PQ: FastEthernet0/0: ip (s=10.1.1.1, d=10.1.2.2) -> low
```

NOTE: This traffic goes into the Low Queue as configured in the first line of our PQ configuration. Since in this lab we are using the local Loopback0, we will not be able to test the second line of this configuration. To test the third line of this configuration, a telnet using port 80 can be used in the following manner:

```
R1#Telnet 10.1.12.2 80

Trying 10.1.12.2, 80 ... Open
Quit → Type and Enter

PQ: FastEthernet0/0: ip (tcp 80) -> medium
PQ: FastEthernet0/0 output (Pk size/Q 60/1)
PQ: FastEthernet0/0: ip (tcp 80) -> medium
PQ: FastEthernet0/0 output (Pk size/Q 60/1)
PQ: FastEthernet0/0: ip (tcp 80) -> medium
```

Note the traffic is using port 80 and it is assigned to the Medium Queue. To test the fourth line, we can use the following method:

```
R1#Copy tftp nvram

Address or name of remote host? 10.1.12.2
Source filename? Abc

Destination filename [nvram]? → Press Enter
Accessing tftp://131.1.12.2/abc...

PQ: FastEthernet0/0: arp (defaulting) -> low
PQ: FastEthernet0/0 output (Pk size/Q 60/3)
```

```
PQ: FastEthernet0/0: ip (udp 69) -> normal
PQ: FastEthernet0/0 output (Pk size/Q 60/2)
PQ: FastEthernet0/0: ip (udp 69) -> normal
```

Note: ARP uses the default queue that is assigned to the Low Queue, and the traffic for UDP port 69 is assigned to the Normal Queue.

Task 2

Change the queue sizes based on the following policy:

High = 80, Medium = 60, Normal = 40, Low = 20

On R1:

```
R1(config)#priority-list 1 queue-limit 80 60 40 20
```

By default the maximum number of packets that the queues can hold is as follows:
High = 20, Medium = 40, Normal = 60 and Low is set to hold 80 packets.

To verify the configuration:

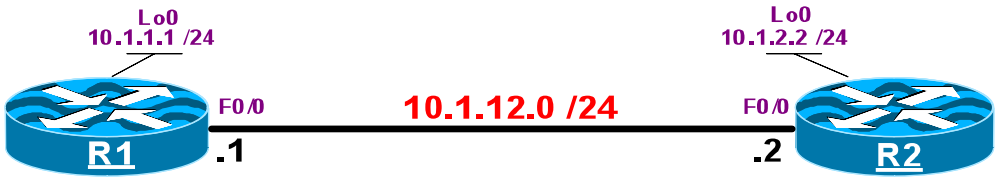
On R1:

```
R1#Show queuing priority

Current DLCI priority queue configuration:
Current priority queue configuration:

List   Queue  Args
1      low    default
1      low    protocol ip          list 100
1      high   interface Loopback0
1      medium protocol ip          tcp port www
1      normal protocol ip          udp port tftp
1      high   limit 80
1      medium limit 60
1      normal limit 40
1      low    limit 20
```

Task 3

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 2 – Custom Queuing



Lab setup:

- Configure the F0/0 interface of these routers in VLAN 12.
- Configure the routers based on the diagram.
- These routers should run RIPv2 and advertise their directly connected networks.

Task 1

Configure custom queuing On R1: using the following policy:

- All traffic generated from the Loopback0 interface should be assigned to Queue 1.
- All IP traffic from R1's Loopback0 interface to R2's Loopback0 interface should be assigned to Queue 2.
- Telnet traffic should be assigned to Queue 3.
- HTTP traffic should be assigned to Queue 4.
- The rest of the traffic should be assign to Queue 5.
- TFTP traffic should be assigned to Queue 6.

The configuration of Custom Queuing is identical to Priority Queuing with the following differences:

- **Priority Queuing defines four Queues: High, Normal, Medium and Low. The priority is based on the queues. If there are packets in the higher queue, they will be de-queued first before the lower queues are serviced. Therefore, the lower queues are subject to queue starvation.**

- **Custom Queuing can be used to create a bandwidth reservation in the output queue of a given interface. Custom queuing defines 17 queues, which includes a system or priority queue. The system queue (Queue 0) should NOT be altered and therefore, traffic should NOT be assigned to it. All user-defined queues have the same weight, meaning they are serviced in a round robin fashion and in every pass 1500 Bytes of data is de-queued. To assign weight (priority) to a given Queue, the Byte-count of that queue can be altered. By default, layer two keepalives and neighbor discovery protocol used by some of the routing protocols will go in the priority or the system queue. The routing updates do NOT go into the system queue.**

On R1:

```
R1(config)#access-list 100 permit ip host 10.1.1.1 host 10.1.2.2

R1(config)#queue-list 1 protocol ip 2 list 100
R1(config)#queue-list 1 interface lo0 1
R1(config)#queue-list 1 protocol ip 3 tcp 23
R1(config)#queue-list 1 protocol ip 4 tcp 80
R1(config)#queue-list 1 protocol ip 6 udp 69
R1(config)#queue-list 1 default 5
```

Note: like the configuration of Priority Queuing, the commands/statements are read in the order in which they are entered. The more specific policies should be toward the top of the configuration, whereas the more general ones should be toward the bottom of the configuration.

Once the statements are configured, they are applied to a given interface using the "custom-queue-list" command; note the direction is not specified because it is always outbound.

```
R1(config)#int f0/0
R1(config-if)#custom-queue-list 1
```

To verify the configuration:

On R1:

```
R1#Show queuing custom

Current custom queue configuration:

List    Queue   Args
1       5       default
1       2       protocol ip          list 100
```

```
1       1       interface Loopback0
1       3       protocol ip             tcp port telnet
1       4       protocol ip             tcp port www
1       6       protocol ip             udp port tftp
```

To test the configuration:

Use the same testing method as the previous lab.

Task 2

Configure the routers such that the bandwidth is allocated as follows:

- Queue 1, 2, and 6 should receive 10% of the bandwidth.
- Queue 3 and Queue 5 should receive 20% of the bandwidth.
- Queue 4 should receive 30% of the bandwidth.

On R1:

```
R1(config)#queue-list 1 queue 1 byte-count 1500
R1(config)#queue-list 1 queue 2 byte-count 1500
R1(config)#queue-list 1 queue 6 byte-count 1500
R1(config)#queue-list 1 queue 3 byte-count 3000
R1(config)#queue-list 1 queue 5 byte-count 3000
R1(config)#queue-list 1 queue 4 byte-count 4500
```

To verify the configuration:

On R1:

```
R1#Show queuing custom

Current custom queue configuration:

List    Queue   Args
1       5       default
1       2       protocol ip             list 100
1       1       interface Loopback0
1       3       protocol ip             tcp port telnet
1       4       protocol ip             tcp port www
1       6       protocol ip             udp port tftp
```

```
1       3       byte-count 3000
1       4       byte-count 4500
1       5       byte-count 3000
```

NOTE: Because the default byte-count value of all queues is 1500 bytes, Queue 1, 2, and 6 will not appear in the output of the show command. The byte count of the queues that are changed will show in the output of the "show queuing custom" command.

If the total number of byte count assigned to all queues are added, the total amount of byte count configured for all queues will equate to 15000 bytes, as follows:

1500 (Queue 1) +1500 (Queue 2) +3000 (Queue 3) + 4500 (Queue 4) +3000 (Queue 5) +1500 (Queue 6) = 15000 Bytes.

To find out the percentage, you should perform the following math:

1500 (for Queue 1 or 2 or 6) / 15000 (total number of Byte count) = 0.1 * 100 = 10%, therefore these queues will receive 10 percent of the bandwidth.

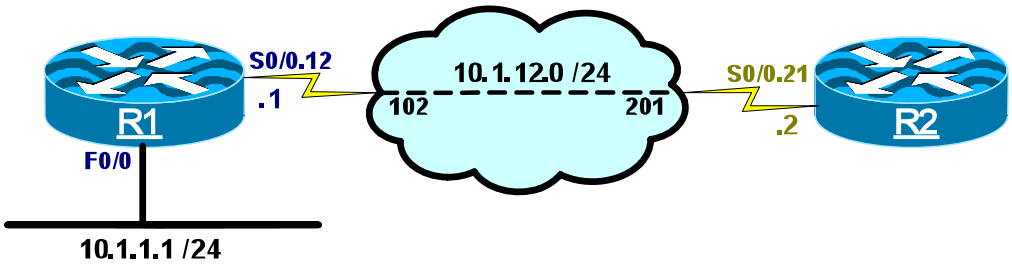
3000 (for Queue 3 and Queue 5) / 15000 (total number of Byte count) = 0.2 * 100 = 20%, these queues will receive 20 percent of the bandwidth.

4500 (for Queue 4) / 15000 (total number of Byte count) = 0.3 * 100 = 30%, queue 4 will receive 30 percent of the bandwidth.

Task 3

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 3 – WFQ



Lab setup:

- Configure the routers in a Frame-Relay point-to-point manner.
- The routers should be configured based on the diagram.
- Configure RIPv2 on both routers and advertise their directly connected interfaces in this routing protocol.

Task 1

Enable WFQ On R1:'s S0/0.12 interface using the following parameters:

The congestive discard threshold value should be set to 128 and ONLY 512 dynamic queues should be created. Ensure that the maximum number of packets that the WFQ system can hold for all queues is set to 1200.

Before these parameters are configured, you should verify the existing thresholds. This can be done using the following commands:

On R1:

```
R1#Show int s0/0 | b Output|Conversation

Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
```

Note the default values on this interface:

- **Hold-queue limit is set to 1000**
- **Congestive Discard Threshold is set to 64**
- **The maximum number of dynamic queues that can be created is set to 256.**

To verify the default values:

On R1:

```
R1#Show queuing fair

Current fair queue configuration:

  Interface             Discard    Dynamic   Reserved  Link      Priority
                        threshold  queues    queues    queues    queues
  Serial0/0             64         256       0         8         1
  Serial0/1             64         256       0         8         1
```

To implement the policy:

```
R1(config)#int s0/0
R1(config-if)#fair-queue 128 512
```

To verify the configuration:

On R1:

```
R1#Sh int s0/0 | b Output

  Output queue: 0/1000/128/0 (size/max total/threshold/drops)
     Conversations  0/1/512 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
```

Note: the CDT and Dynamic Queues are set to 128 and 512 respectively. Enter the following command to change the maximum "hold queue out" threshold:

On R1:

```
R1(config)#int s0/0
R1(config-if)#hold-queue 1200 out

R1#Show int S0/0 | inc Output|Conver
```

```
Output queue: 0/1200/128/0 (size/max total/threshold/drops)
     Conversations  0/1/512 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
```

Task 2

Two of the PCs that are connected to the F0/0 interface of R1 (PC1 and PC2) are sending traffic with equal size packets. Ensure that PC1's packets appear to be half the size they really are and because of this configuration, they receive twice as much bandwidth as the packets from PC2. You should use IP precedence to accomplish this task. PC1's IP address is 10.1.1.10 /24 and PC2's IP address is 10.1.1.20 /24

On R1:

```
R1(config)#access-list 1 permit host 10.1.1.10
R1(config)#access-list 2 permit host 10.1.1.20

R1(config)#class-map pc1
R1(config-cmap)#match access-group 1

R1(config-cmap)#class-map pc2
R1(config-cmap)#match access-group 2

R1(config)#policy-map TST
R1(config-pmap)#class pc1
R1(config-pmap-c)#set ip precedence 1

R1(config-pmap)#class pc2
R1(config-pmap-c)#set ip precedence 0

R1(config)#int f0/0
R1(config-if)#service-policy input TST
```

WFQ without the weight should be fair queuing; the weight aspect of the WFQ comes in when traffic is marked with IP Precedence levels. The formula that WFQ uses is as follows:

If packets are 1500 Bytes and the IP Precedence is set to 0, then:
[32384 / (IP Precedence + 1)] X 1500 = 32384 / (0 + 1) = 48,576,000

The 32384 is a set number.

If packets are 1500 Bytes and the IP Precedence is set to 1, then:

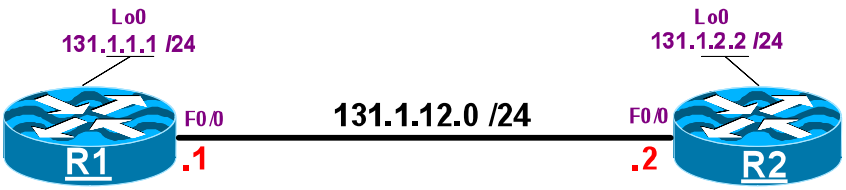
[32384 / (IP Precedence + 1)] X 1500 = 32384 / (1 + 1) = 24,288,000

Note the packets marked with IP Precedence of 1 appear to be half the size of packets marked with IP Precedence of 0; therefore, they will receive twice as much bandwidth as the packets that are marked with an IP Precedence level of 0.

Task 3

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 4 – CBWFQ



Lab setup:

- Configure the F0/0 interface of these routers in VLAN 12
- Configure the routers based on the diagram
- These routers should run RIPv2 and advertise their directly connected networks

Task 1

Configure R1 such that when congestion is experienced, the specified amount of bandwidth is allocated to the following protocols for the F0/0 interface. You should NOT use an access-list to accomplish this task. This QoS should be configured for the outbound traffic toward R2.

- TFTP traffic should have a minimum of 2 Mbps
- HTTP traffic should have a minimum of 5 Mbps
- FTP traffic should have a minimum of 3 Mbps
- Any outbound calls to URL www.MicronicsTraining.com should have a minimum of 6 Mbps

CBWFQ extends the WFQ functionality so it can provide support for user-defined classes.

Points to remember about CBWFQ:

- **CBWFQ is configured using MQC.**
- **A mechanism that enables users to guarantee a minimum amount of bandwidth.**
- **CBWFQ reserves multiple FIFO queues in the WFQ system.**
- **The default queue limit is 64, after which packets will be tail dropped.**
- **WRED can be configured in combination with CBWFQ to prevent congestion.**
- **CBWFQ guarantees bandwidth according to weights that are assigned to the different classes within the MQC.**
- **In CBWFQ, weights are defined based on bandwidth, bandwidth percent, and bandwidth remaining percent keywords.**
- **When applying the policy-map to a given interface using the "service-policy" command, weights assigned to the classes can NOT be mixed.**
- **By default, only 75 percent of the bandwidth can be defined. This threshold can be exceeded using the "max-reserved bandwidth" interface command.**

MQC provides a modular approach to the configuration of QoS. MQC deploys a three step approach for configuring QoS on a given interface, and they are:

1. **Define classes of traffic: in this step of the configuration, the traffic that is cared for is defined.**
2. **Define QoS policies for classes: in this step, the policy that should be assigned to the traffic is configured.**
3. **Apply a service policy: lastly, the policy is applied to a given interface in a particular direction.**

Note: In this case "MQC" is used to implement CBWFQ. With MQC a class-map is configured to identify the traffic, a policy-map is used to apply the desired policies to the class-map, and in the final step of this configuration, the policy-map is applied to the interface, either inbound or outbound using the "service-policy" command.

Remember that the bandwidth value is in units of Kbps.

On R1:

```
R1(config)#ip cef

R1(config)#class-map tftp
R1(config-cmap)#match protocol tftp

R1(config)#class-map url
R1(config-cmap)#match protocol http host www.MicronicsTraining.com

R1(config)#class-map http
R1(config-cmap)#match protocol http
```

```
R1(config)#class-map ftp
R1(config-cmap)#match protocol ftp

R1(config)#policy-map test
R1(config-pmap)#class tftp
R1(config-pmap-c)#bandwidth 2000

R1(config-pmap)#class url
R1(config-pmap-c)#bandwidth 6000

R1(config-pmap)#class http
R1(config-pmap-c)#bandwidth 5000

R1(config-pmap)#class ftp
R1(config-pmap-c)#bandwidth 3000

R1(config-pmap-c)#int f0/0
R1(config-if)#service-policy output test
```

To verify the configuration:

On R1:

```
R1#Show policy-map interface F0/0

 FastEthernet0/0

  Service-policy output: TEST

    Class-map: TFTP (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol tftp
      Queuing
        Output Queue: Conversation 265
        Bandwidth 2000 (kbps) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0

    Class-map: URL (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol http url " www.MicronicsTraining.com "
      Queuing
```

```
Output Queue: Conversation 268
     Bandwidth 6000 (kbps) Max Threshold 64 (packets)
     (pkts matched/bytes matched) 0/0
     (depth/total drops/no-buffer drops) 0/0/0

 Class-map: HTTP (match-all)
   0 packets, 0 bytes
   5 minute offered rate 0 bps, drop rate 0 bps
   Match: protocol http
   Queuing
     Output Queue: Conversation 266
     Bandwidth 5000 (kbps) Max Threshold 64 (packets)
     (pkts matched/bytes matched) 0/0
     (depth/total drops/no-buffer drops) 0/0/0

 Class-map: FTP (match-all)
   0 packets, 0 bytes
   5 minute offered rate 0 bps, drop rate 0 bps
   Match: protocol ftp
   Queuing
     Output Queue: Conversation 267
     Bandwidth 3000 (kbps) Max Threshold 64 (packets)
     (pkts matched/bytes matched) 0/0
     (depth/total drops/no-buffer drops) 0/0/0

 Class-map: class-default (match-any)
   80 packets, 7370 bytes
   5 minute offered rate 0 bps, drop rate 0 bps
   Match: any
```

Task 2

Configure the maximum queue size for the calls made to www.MicronicsTraining.com URL to 128 packets.

On R1:

```
R1(config)#policy-map test
R1(config-pmap)#class url
R1(config-pmap-c)#queue-limit 128
```

Note: Before this step was configured, the default maximum threshold was 64 packets. To verify the

configuration:

```
R1#Show policy-map interface f0/0

Class-map: URL (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: protocol http url " www.MicronicsTraining.com "
  Queuing
    Output Queue: Conversation 268
    Bandwidth 6000 (kbps) Max Threshold 128 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0
```

Task 3

Configure the remaining traffic for the above policy as follows:

- They should be configured to use fair-queuing
- Set the dynamic queues for the remaining traffic to 1024

On R2:

```
R2(config)#policy-map test
R2(config-pmap)#class class-default
R2(config-pmap-c)#fair-queue 1024
```

To verify the configuration:

On R2:

```
R2#Show policy-map interface F0/0

Class-map: class-default (match-any)
  112 packets, 11410 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
  Queuing
    Flow Based Fair Queuing
    Maximum Number of Hashed Queues 1024
    (total queued/total drops/no-buffer drops) 0/0/0
```

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 5 – CBWFQ-II

The topology and IP the addressing is based on the previous lab

Lab setup:

- Configure the F0/0 interface of these routers in VLAN 12
- Configure the routers based on the diagram
- These routers should run RIPv2 and advertise their directly connected networks

Task 1

Configure R1 to allocate the specified percentage of bandwidth to the following protocols for the F0/0 interface. You should NOT use an access-list to accomplish this task. This QoS should be configured for the outbound traffic toward R2.

TFTP to 25% of the available bandwidth
HTTP to 35% of the available bandwidth
FTP to 20% of the available bandwidth

On R1:

```
R1(config)#class-map tftp
R1(config-cmap)#match protocol tftp

R1(config)#class-map http
R1(config-cmap)#match protocol http

R1(config)#class-map ftp
R1(config-cmap)#match protocol ftp

R1(config)#policy-map test
R1(config-pmap)#class tftp
R1(config-pmap-c)#bandwidth percent 25

R1(config-pmap-c)#class http
R1(config-pmap-c)#bandwidth percent 35
```

```
R1(config-pmap-c)#class ftp
R1(config-pmap-c)#bandwidth 20
```

All classes with bandwidth should have consistent units

Note: Notice the error above: if the "bandwidth" argument is used the "bandwidth percent" argument CANNOT BE USED within the same policy-map.

```
R1(config-pmap-c)#bandwidth percent 20

R1(config-pmap-c)#int f0/0
R1(config-if)#service-policy output test
```

I/f FastEthernet0/0 class FTP requested bandwidth 20%, available only 15%

Note: The above error is displayed because you are allowed to allocate up to 75% of the bandwidth. The following command can be used to modify the default behavior:

```
R1(config-if)#max-reserved-bandwidth 85
R1(config-if)#service-policy output test
```

Note the "max-reserved-bandwidth" command is changed such that we can manually allocate up to 85% of the bandwidth.

Task 2

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 6 – Converting Custom Queuing to CBWFQ

The topology and IP the addressing is based on the previous lab

Lab setup:

- Configure the F0/0 interface of these routers in VLAN 12
- Configure the routers based on the diagram
- These routers should run RIPv2 and advertise their directly connected networks

Task 1

Convert the following Custom Queue list to CBWFQ for the F0/0 interface On R1:.

```
Queue-list 1 protocol ip 1 tcp www
Queue-list 1 protocol ip 2 tcp telnet
Queue-list 1 protocol ip 3 tcp smtp
Queue-list 1 default 4
Queue-list 1 queue 1 byte-count 3000
Queue-list 1 queue 2 byte-count 4500
Queue-list 1 queue 3 byte-count 4500
Queue-list 1 queue 4 byte-count 3000
```

To calculate the required bandwidth percentage, you should take the following steps:

Step 1:

Calculate the total configured Byte-count used by the Custom Queue:
3000+4500+4500+3000 = 15000

Step 2:

Calculate the percentage of the bandwidth assigned to each protocol based on the total Byte-count:

WWW protocol:
(3000 / 15000) X 100 = 20, this protocol is assigned 20% of the bandwidth.

Telnet protocol:
(4500 / 15000) X 100 = 30, this protocol is assigned 30% of the bandwidth.

SMTP protocol:
(4500 / 15000) X 100 = 30, this protocol is assigned 30% of the bandwidth.

The rest of the traffic:
(3000 / 15000) X 100 = 20, this protocol is assigned 20% of the bandwidth.

Last Step:

Since these protocols use a total of 100 percent, we must modify the maximum reserved bandwidth to match 100%; this is accomplished by using the "max-reserve-bandwidth" command configured in the F0/0 interface config mode.

On R2:

```
R2(config)#ip cef

R2(config)#class-map WWW
R2(config-cmap)#match protocol http

R2(config)#class-map TELNET
R2(config-cmap)#match protocol telnet

R2(config)#class-map SMTP
R2(config-cmap)#match protocol smtp

R2(config)#policy-map TEST
R2(config-pmap)#class WWW
R2(config-pmap-c)#bandwidth percent 20

R2(config-pmap-c)#class TELNET
R2(config-pmap-c)#bandwidth percent 30

R2(config-pmap-c)#class SMTP
R2(config-pmap-c)#bandwidth percent 30

R2(config-pmap-c)#class class-default
R2(config-pmap-c)#bandwidth percent 20

R2(config-pmap-c)#int f0/0
R2(config-if)#max-reserved-bandwidth 100
R2(config-if)#service-policy output TEST
```

Task 2

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 7 - LLQ

The topology and IP the addressing is based on the previous lab

Lab setup:

- Configure the F0/0 interface of these routers in VLAN 12.
- Configure the routers based on the diagram.
- These routers should run RIPv2 and advertise their directly connected networks.

Task 1

Configure R1 to allocate the specified percentage of bandwidth to the following protocols for the F0/0 interface. You should NOT use an access-list to accomplish this task. This QoS should be configured for the outbound traffic toward R2.

TFTP to 15% of the available bandwidth
HTTP to 25% of the available bandwidth
FTP to 20% of the available bandwidth
Traffic with IP Precedence of 5 should be allocated 25% of the bandwidth. This traffic type should be ensured of expedited forwarding. Traffic exceeding this threshold should be dropped.

This task calls for a LLQ (Low Latency Queuing) configuration. LLQ guarantees a maximum amount of bandwidth to be Low latency queued. LLQ brings strict priority queuing to CBWFQ. This allows time delay sensitive traffic such as voice to be de-queued and sent first before the other packets in the other queues. Unlike its legacy counterpart (priority-queuing), it ONLY uses a single queue and it is NOT subject to starvation.

LLQ utilizes two commands to provide low latency queuing for a given traffic and they are as follows:

1. **Priority: Guarantees a maximum amount of bandwidth to the priority traffic, the specified value is in kbps. If the priority traffic exceeds the configured value, the excess traffic will be dropped if congestion is experienced.**
2. **Priority percent *value*: With this command, we can specify the amount of bandwidth in percentage versus kbps. The value keyword is a number between 1 – 100. Remember by default only 75 percent of the bandwidth is used, unless it has specifically changed using the "max-reserved bandwidth" interface command.**

On R1:

```
R1(config)#ip cef

R1(config)#class-map tftp
R1(config-cmap)#match protocol tftp

R1(config)#class-map http
R1(config-cmap)#match protocol http

R1(config)#class-map ftp
R1(config-cmap)#match protocol ftp

R1(config)#class-map prec-5
R1(config-cmap)#match ip precedence 5

R1(config)#policy-map test
R1(config-pmap)#class tftp
R1(config-pmap-c)#bandwidth percent 15

R1(config-pmap-c)#class http
R1(config-pmap-c)#bandwidth percent 25

R1(config-pmap-c)#class ftp
R1(config-pmap-c)#bandwidth percent 20

R1(config-pmap-c)#class prec-5
R1(config-pmap-c)#priority percent 25

R1(config)#int f0/0
R1(config-if)#max-reserved-bandwidth 85
```

Note LLQ is configured using the "Priority percent" command

If the above command is not entered first you will receive an error.

```
R1(config-if)#service-policy output test
```

Note: LLQ is implemented by using the "priority" and the "priority percent" commands. The "priority" command is analogous to the "bandwidth" command, and the "priority percent" command is analogous to the "bandwidth percent" command.

To verify the configuration:

On R1:

```
R1#Show policy-map interface f0/0
```

```
FastEthernet0/0

 Service-policy output: TEST

   Class-map: TFTP (match-all)
     0 packets, 0 bytes
     5 minute offered rate 0 bps, drop rate 0 bps
     Match: protocol tftp
     Queuing
       Output Queue: Conversation 265
       Bandwidth 15 (%)
       Bandwidth 15000 (kbps) Max Threshold 64 (packets)
       (pkts matched/bytes matched) 0/0
       (depth/total drops/no-buffer drops) 0/0/0

   Class-map: HTTP (match-all)
     0 packets, 0 bytes
     5 minute offered rate 0 bps, drop rate 0 bps
     Match: protocol http
     Queuing
       Output Queue: Conversation 266
       Bandwidth 25 (%)
       Bandwidth 25000 (kbps) Max Threshold 64 (packets)
       (pkts matched/bytes matched) 0/0
       (depth/total drops/no-buffer drops) 0/0/0

   Class-map: FTP (match-all)
     0 packets, 0 bytes
     5 minute offered rate 0 bps, drop rate 0 bps
     Match: protocol ftp
     Queuing
       Output Queue: Conversation 267
       Bandwidth 20 (%)
       Bandwidth 20000 (kbps) Max Threshold 64 (packets)
       (pkts matched/bytes matched) 0/0
       (depth/total drops/no-buffer drops) 0/0/0

 Class-map: PREC-5 (match-all)
     0 packets, 0 bytes
     5 minute offered rate 0 bps, drop rate 0 bps
     Match: ip precedence 5
     Queuing
     Strict Priority
     Output Queue: Conversation 264
     Bandwidth 25 (%)
     Bandwidth 25000 (kbps) Burst 625000 (Bytes)
```

Note LLQ is enabled for this class.

```
(pkts matched/bytes matched) 0/0
     (total drops/bytes drops) 0/0

 Class-map: class-default (match-any)
   41 packets, 3869 bytes
   5 minute offered rate 0 bps, drop rate 0 bps
   Match: any
```

Task 2

Remove the configuration for this lab before proceeding to the next lab.

On R1:

```
R1(config)#No class-map tftp

% Class-map TFTP is being used
```

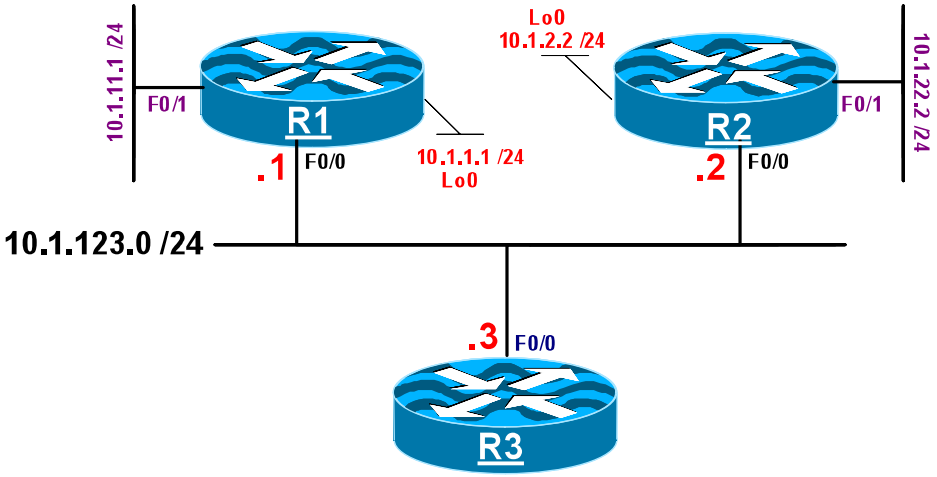
Note: To remove the commands, you must negate the commands in the reverse order.

```
R1(config)#int f0/0
R1(config-if)#NO service-policy output test
R1(config-if)#NO max-reserved-bandwidth 85
R1(config)#NO policy-map test
R1(config)#NO class-map prec-5
R1(config)#NO class-map tftp
R1(config)#NO class-map http
R1(config)#NO class-map ftp
```

Task 3

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 8 – Class Based Policing-I



Lab setup:

- Configure the F0/0 interface of these routers in VLAN 100.
- Configure the F0/1 interface of R1 in VLAN 11 and R2 in VLAN 22.
- Configure the routers based on the diagram.
- These routers should run RIPv2 and advertise their directly connected networks.

Task 1

Configure R1's F0/0 interface using the following policy:

- HTTP, FTP and ICMP traffic should be limited to 10 Mbps
- Telnet and SMTP traffic should be limited to 8 Mbps
- Traffic exceeding these thresholds should be dropped and traffic conforming to these thresholds should be transmitted
- DO NOT create an access-list to accomplish this task

Since access-lists cannot be used, NBAR is probably the ONLY other way to identify these traffic types.

With NBAR, you must have "IP CEF" enabled. All the new IOS releases have this enabled on most of the platforms.

When configuring Class Based Policing, if the "Bc" value is NOT specified, it will default to the CIR / 32 or 1500 Bytes, or whichever is the higher value, with a Tc of .25 seconds. Remember that in a single rate single bucket, the Be is disabled. If it is configured, the system will ignore it. The "Be" rate can ONLY be utilized when a violate action is configured.

On R1:

Note: To configure a match for the specified protocols, NBAR is used. NBAR uses the "match" command to match on different protocols. When the first match command is entered, there will be a little hesitation before the cursor is available again, but the subsequent match statements will not have this behavior. This happens because when NBAR is used for the first time, it needs to download the PDLMs (signature files) into the memory. Once they are loaded, they can be used and accessed very quickly.

```
R1(config)#class-map match-any qos-1
R1(config-cmap)#match protocol http
R1(config-cmap)#match protocol ftp
R1(config-cmap)#match protocol icmp
```

If the "match-any" is not used, a "match-all" option will be used.

```
R1(config)#class-map match-any qos-2
R1(config-cmap)#match protocol telnet
R1(config-cmap)#match protocol smtp

R1(config)#policy-map test
R1(config-pmap)#class qos-1
R1(config-pmap-c)#police 10000000 conform-action transmit exceed-action drop
```

Note: When configuring the "rate-limit" interface configuration command, the normal burst and maximum burst MUST be configured. When configuring the "police" command through the MQC, they do not have to be configured. If they are NOT configured the system will use the "CIR/32" or 1500 Bytes, whichever one is higher, as the normal burst.

```
R1(config-pmap)#class qos-2
R1(config-pmap-c)#police 8000000 conform-action transmit exceed-action drop

R1(config-pmap-c)#int f0/0
R1(config-if)#service-policy output test
```

To verify the configuration:

On R1:

```
R1#Show policy-map interface f0/0

 FastEthernet0/0

  Service-policy output: TEST

    Class-map: QOS-1 (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol http
   Match: protocol ftp
   Match: protocol icmp
   police:
     cir 10000000 bps, bc 312500 bytes
    conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps

    Class-map: QOS-2 (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol telnet
      Match: protocol smtp
      police:
          cir 8000000 bps, bc 250000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps

    Class-map: class-default (match-any)
      17 packets, 1586 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Note the Bc is not configured and the value of Bc defaults to CIR/32 10,000,000 / 32 = 312,500

Excess Burst is disabled.

To verify the configuration:

On R1:

```
R1#Show policy-map TST

  Policy Map TST
```

```
Class QOS-1
 police cir 10000000 bc 312500
   conform-action transmit
   exceed-action drop
Class QOS-2
 police cir 8000000 bc 250000
   conform-action transmit
   exceed-action drop
```

Task 2

Configure R2's F0/0 interface using the following policy:
Telnet traffic should be limited to 10 Mbps. This traffic should be configured with minimum amount of normal burst.
TFTP traffic should be limited to 8 Mbps with 40000 bps of normal bursts.
Both Telnet and Tftp traffic exceeding this policy should be dropped and if they conform to this policy, they should be transmitted.
DO NOT create an access-list to accomplish this task.

On R2:

```
R2(config)#class-map telnet
R2(config-cmap)#match protocol telnet

R2(config)#class-map tftp
R2(config-cmap)#match protocol tftp

R2(config)#policy-map test
R2(config-pmap)#class telnet
R2(config-pmap-c)#police 10000000 1000 conform-act transmit exceed-act drop
```

You should receive the following console message telling you that the minimum normal burst size for a CIR of 10Mbps should be 5000 Bytes so the system sets the normal burst to 5000:

Conform burst size increased to 5000

```
R2(config-pmap)#class tftp
R2(config-pmap-c)#police 8000000 5000 conform-act transmit exceed-act drop

R2(config-pmap-c-police)#int f0/0
R2(config-if)#service-policy out test
```

To verify the configuration:

On R2:

```
R2#Show policy-map interface f0/0

 FastEthernet0/0
  Service-policy output: TEST

    Class-map: Telnet (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol telnet
      police:
          cir 10000000 bps, bc 5000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps

    Class-map: TFTP (match-any)
      0 packets, 0 bytes
       5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol tftp
      police:
          cir 8000000 bps, bc 5000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps

    Class-map: class-default (match-any)
      38 packets, 3418 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

The last entry is always created automatically by the system. It is the "catch-all" condition. Basically, whatever traffic you did not specify will be taken care of by this condition.
Note: If a "?" is used after entering the access-rate, a range of (1000 – 512000000) is displayed. When you entered the minimum value (1000), the system will increase the value to the minimum value for the configured access-rate if the entered value is lower than minimum value possible for the configured CIR. Note the output of the "Show run" command reveals that the system changed the parameters automatically:

```
R2#Sh run | in police

    police 10000000 5000 conform-action transmit  exceed-action drop
    police 8000000 5000 conform-action transmit  exceed-action drop

R2#Sh policy-map TEST

  Policy Map TEST
    Class Telnet
     police cir 10000000 bc 5000
       conform-action transmit
       exceed-action drop
    Class TFTP
     police cir 8000000 bc 5000
       conform-action transmit
       exceed-action drop
```

Task 3

R1 has two servers connected to its F0/1 interface with the following MAC-addresses:
SRV1 = 0000.1111.1111
SRV2 = 0000.2222.2222

The traffic with a source MAC address that matches SRV1 should be policed to 1 Mb,whereas the traffic with a source mac address that matches SRV2 should be policed to 2 Mbps; traffic from these servers should ONLY be transmitted out of R1's F0/1 interface if they conform to this policy.

Note: Even though this task sounds very simple, it can be tricky. The reason it is tricky is that in the "Class-map Server1", if a match is made to the source MAC address of Server1 using the "Match source-address MAC 0000.1111.1111" command, then the policy-map cannot be applied outbound to the F0/1 interface. You will receive the following error message:

'match source-addr mac' is not allowed in an output policy

Because of this restriction, two MAC address access-lists should be created. One identifying Server1's MAC address and the second one identifying the Server2's MAC address as follows:

On R1:

```
R1(config)#class-map server1
R1(config-cmap)#match access-group 700
```

```
R1(config)#class-map server2
R1(config-cmap)#match access-group 701

R1(config)#policy-map tst
R1(config-pmap)#class server1
R1(config-pmap-c)#police 1000000 conform-action transmit exceed-action drop
R1(config-pmap)#class server2
R1(config-pmap-c)#police 8000000 conform-action transmit exceed-action drop

R1(config)#int f0/1
R1(config-if)#service-policy input tst
```

To verify the configuration:

On R1:

```
R1#Sh policy-map inter f0/1

 FastEthernet0/1

  Service-policy output: TST

    Class-map: Server1 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 700
      police:
          cir 1000000 bps, bc 31250 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps

    Class-map: Server2 (match-all)

      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 701
      police:
          cir 2000000 bps, bc 62500 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps
```

```
    Class-map: class-default (match-any)
      1 packets, 60 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any

R1#Show policy-map TST

  Policy Map TST
    Class Server1
     police cir 1000000 bc 31250
       conform-action transmit
       exceed-action drop
    Class Server2
     police cir 2000000 bc 62500
       conform-action transmit
       exceed-action drop
```

Task 4

Ensure that HTTP, FTP, and ICMP traffic On R3:'s F0/0 interface is policed to 10 Mbps on weekdays from 11:00 AM to 3:00 PM. Traffic exceeding this policy should be dropped and traffic conforming to this policy should be transmitted.

On R3:

```
R3(config)#time-range weekdays
R3(config-time-range)#periodic weekdays 11:00 to 15:00

R3(config)#access-list 100 permit tcp any any eq www time-range weekdays
R3(config)#access-list 100 permit icmp any any time-range weekdays
R3(config)#access-list 100 permit tcp any any eq 20 time-range weekdays
R3(config)#access-list 100 permit tcp any any eq 21 time-range weekdays

R3(config)#class-map qos
R3(config-cmap)#match access-group 100

R3(config)#policy-map TEST
R3(config-pmap)#class qos
R3(config-pmap-c)#police 10000000 conform-act transmit exceed-act drop

R3(config-pmap-c)#int f0/0
```

```
R3(config-if)#service-policy out TEST
```

To verify the configuration:

On R3:

```
R3#Show policy-map inter f0/0

 FastEthernet0/0

  Service-policy output: TEST
    Class-map: QOS (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 100
      police:
          cir 10000000 bps, bc 312500 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps

    Class-map: class-default (match-any)
      18 packets, 1663 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Task 5

Remove the configuration commands from this lab before proceeding to the next lab.

Lab 9 – Class Based Policing-II

The topology and IP the addressing is based on the previous lab

Lab setup:

- Configure the F0/0 interface of these routers in VLAN 12.
- Configure the F0/1 interface of R2 in VLAN 22
- Configure the routers based on the diagram.
- These routers should run RIPv2 and advertise their directly connected networks.

Task 1

Ensure that outbound HTTP traffic from R1's F0/0 interface is policed and marked based on the following policy:

- Traffic up to 1Mbps should be sent as is
- Traffic exceeding 1Mbps up to 2Mbps should be marked with IP Precedence 4 and transmitted
- Traffic exceeding 2Mbps should always be dropped

To accomplish this task, a dual rate dual bucket Class Based Pollicer should be configured. If "Bc" is NOT configured, the CIR / 32 or 1500 Bytes (whichever is higher) is chosen as the "Bc". However, if "Be" is NOT configured, the PIR / 32 or 1500 Bytes (whichever is higher) is chosen as the "Be" value. This feature was added to the IOS starting 12.2(4)T.

On R1:

```
R1(config)#class-map http
R1(config-cmap)#Match protocol http

R1(config)#policy-map tst
R1(config-pmap)#class http
R1(config-pmap-c)#police cir 1000000 pir 2000000 conform-act transmit
exceed-action set-prec-trans 4 violate-action drop
```

```
R1(config)#int f0/0
R1(config-if)#service-policy out tst
```

To verify the configuration:

On R1:

```
R1#Show policy-map tst
FastEthernet0/0

 Service-policy output: TST

   Class-map: HTTP (match-all)
     0 packets, 0 bytes
     5 minute offered rate 0 bps, drop rate 0 bps
     Match: protocol http
     police:
         cir 1000000 bps, bc 31250 bytes
         pir 2000000 bps, be 62500 bytes
       conformed 0 packets, 0 bytes; actions:
         transmit
       exceeded 0 packets, 0 bytes; actions:
         set-prec-transmit 4
       violated 0 packets, 0 bytes; actions:
         drop
       conformed 0 bps, exceed 0 bps, violate 0 bps

   Class-map: class-default (match-any)
     9 packets, 858 bytes
     5 minute offered rate 0 bps, drop rate 0 bps
     Match: any
```

Conform-action is transmitted

Exceed-action is tagged with IPP 4 & transmitted

Violated-action is dropped

Task 2

Users in VLAN 22 usually connect to the web server with an IP address of 10.1.12.100. Ensure that this traffic is policed to 30 percent of the bandwidth of the F0/1 interface.

The "Police percent" command is used to configure traffic policing based on a percentage of bandwidth available on an interface; this command should be used in the "policy-map" class configuration mode as follows:

On R2:

```
R2(config)#access-list 100 permit tcp 10.1.22.0 0.0.0.255 host 10.1.12.100 eq 80

R2(config)#class-map tst-www
R2(config-cmap)#match access-group 100

R2(config)#policy-map tst
R2(config-pmap)#Class tst-www
R2(config-pmap-c)#police cir percent 30

R2(config)#interface f0/1
R2(config-if)#service-policy output tst
```

Remember that the police cir percent is calculated based on the Bandwidth statement configured under the interface to which it is applied.

To verify the configuration:

On R2:

```
R2#Show policy-map int f0/1
 FastEthernet0/1

  Service-policy output: TST

    Class-map: TST-WWW (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 100
      police:
          cir 30 %
          cir 30000000 bps, bc 937500 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps

    Class-map: class-default (match-any)
      6 packets, 666 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Task 3

Configure R2's F0/0 interface using the following policy:

- Traffic up to 10 Mbps should be sent as is
- Traffic exceeding 10 Mbps up to 20 Mbps should be marked with IP Precedence 4 and transmitted
- Traffic exceeding 20 Mbps should always be dropped

On R2:

```
R2(config)#policy-map test
R2(config-pmap)#class class-default
R2(config-pmap-c)#police cir percent 10 pir percent 20 conform-action transmit
                   Exceed-action set-prec-transmit 4
                   Violate-action drop

R2(config)#int f0/0
R2(config-if)#service-policy out test
```

To verify the configuration:

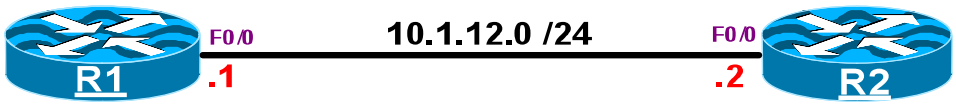
On R2:

```
R2#Sh policy-map int f0/0

 FastEthernet0/0
  Service-policy output: TEST

    Class-map: class-default (match-any)
      4 packets, 546 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      police:
          cir 10 %
          cir 10000000 bps, bc 312500 bytes
          pir 20 %
          pir 20000000 bps, be 625000 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          set-prec-transmit 4
        violated 0 packets, 0 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps, violate 0 bps
```

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 10 – WRED and CB-WRED



Lab setup:

- Configure F0/0 interface of these two routers in VLAN 12.
- The routers should be configured based on the diagram.

Task 1

Configure WRED On R1:'s F0/0 interface based on the following parameters and policies:

Name of the class-map	IP Precedence assigned to the class-map	Bandwidth guaranteed	WRED drop probability
Priority	4	35%	As per default for the Precedence values.
Bulk	2 and 3	25%	As per default for the Precedence values.
Best-effort	0 and 1	20%	As per default for the Precedence values.

On R1:

```
R1(config)#class-map best-effort
R1(config-cmap)#match ip precedence 0 1
```

```
R1(config)#class-map bulk
R1(config-cmap)#match ip precedence 2 3

R1(config)#class-map priority
R1(config-cmap)#match ip precedence 4

R1(config)#policy-map test
R1(config-pmap)#class best-effort
R1(config-pmap-c)#bandwidth percent 20
R1(config-pmap-c)#random-detect
R1(config-pmap-c)#random-detect precedence 0 20 40 10
R1(config-pmap-c)#random-detect precedence 1 22 40 10

R1(config-pmap-c)#class bulk
R1(config-pmap-c)#bandwidth percent 25
R1(config-pmap-c)#random-detect
R1(config-pmap-c)#random-detect precedence 2 24 40 10
R1(config-pmap-c)#random-detect precedence 3 26 40 10

R1(config-pmap-c)#class priority
R1(config-pmap-c)#bandwidth percent 35
R1(config-pmap-c)#random-detect
R1(config-pmap-c)#random-detect precedence 4 28 40 10

R1(config-pmap-c)#int f0/0
R1(config-if)#max-reserved-bandwidth 80
R1(config-if)#service-policy out test
```

Note: The default minimum threshold for the IP Precedence levels are as configured in the example.

Task 2

Remove the configuration from the previous task.

On R1:

```
R1(config-pmap-c)#int f0/0
R1(config-if)#No max-reserved-bandwidth 80
R1(config-if)#No service-policy out test

R1(config)#No policy-map test
R1(config)#No class-map best-effort
```

```
R1(config)#No class-map bulk
R1(config)#No class-map priority
```

Task 3

Enable DSCP-based WRED On R1:'s F0/0.

On R1:

R1(config)#**int f0/0**
R1(config-if)#**random-detect dscp-based**

By default, WRED is precedence-based, and uses eight default WRED profiles, one for each IP Precedence value. WRED can be changed to be DSCP-based; this will increase the number of profiles to 64. These profiles can be changed.

Task 4

Configure WRED On R1:'s F0/0 interface based on the following parameters and policies:

Name of the class-map	DSCP assigned to the class-map	Bandwidth guaranteed	WRED drop probability
Priority	AF21	35%	Enable ECN
Bulk	AF22	25%	Min-threshold = 30 Max-threshold = 40 Mark Probability Denominator = 1 out of 18
Best-effort	AF23	20%	Min-threshold = 26 Max-threshold = 40 Mark Probability Denominator = 1 out of 10

On R1:

```
R1(config)#int f0/0
R1(config-if)#No random-detect dscp-based
```

The "random-detect dscp-based" command must be removed before configuring this task.

On R1:

```
R1(config)#class-map best-effort
R1(config-cmap)#match ip dscp af23

R1(config)#class-map bulk
R1(config-cmap)#match ip dscp af22

R1(config)#class-map priority
R1(config-cmap)#match ip dscp af21

R1(config)#policy-map test
R1(config-pmap)#class best-effort
R1(config-pmap-c)#bandwidth percent 20
R1(config-pmap-c)#random-detect dscp-based
R1(config-pmap-c)#random-detect dscp af23 26 40 10

R1(config-pmap)#class bulk
R1(config-pmap-c)#bandwidth percent 25
R1(config-pmap-c)#random-detect dscp-based
R1(config-pmap-c)#random-detect dscp af22 30 40 18

R1(config-pmap)#class priority
R1(config-pmap-c)#bandwidth percent 35
R1(config-pmap-c)#random-detect
R1(config-pmap-c)#random-detect ecn

R1(config-pmap-c)#int f0/0
R1(config-if)#max-reserved-bandwidth 80
R1(config-if)#service-policy output test
```

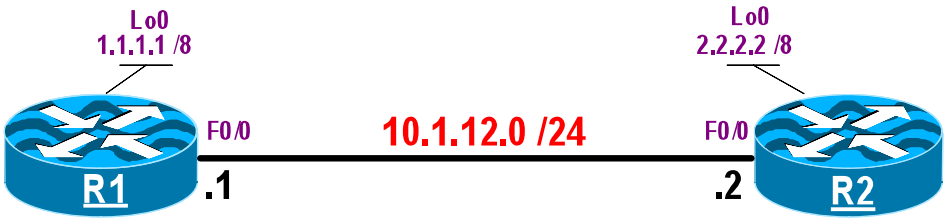
- **ECN is an extension to WRED.**
- **ECN marks packets instead of dropping them when the average queue length exceeds a configured threshold value.**
- **Routers and end hosts would use this marking as a signal that the network is congested and slow down sending the packets.**
- **RFC 3168 defines the addition of ECN to IP. These are two fields in the IP header ECT and CE. Explicit congestion transport and congestion experienced. These two bits can define four combinations (00, 01, 10 and11).**

The first bit is the ECT bit and the second bit is the CE bit. 00 in the IP header means that the end hosts are not ECN aware, whereas 01 or 10 means that the end hosts are ECN aware. The 11 means that congestion was experienced.

Task 5

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 11 – RSVP



Lab setup:

- Configure the F0/0 interface of these routers in VLAN 12.
- Configure the routers based on the diagram.
- These routers should run RIPv2 and advertise their directly connected networks.

Task 1

Configure RSVP from R1's Lo0 to R2's Lo0. This reservation should be restricted to 400 Kbps, but no single reservation may exceed 180 Kbps.

RSVP operation:

- **Sender sends a special RSVP packet called path messages to the network.**
- **Path message flows through the network, along the normal routed path of data from the sender to the receiver. The direction of the message is downstream.**
- **The path messages are propagated from the source to the destination on a periodic basis.**
- **When an RSVP enabled router receives the path message, it keeps a record of the information contained in the message, this information contains the following:**

From	To	Previous Hop	Bandwidth

- **After storing the information, the router forwards the message to the next router along the path to the receiver.**
- **Once the receiver receives the path message, the receiver inspects the path message and uses the information in the path message to formulate an RSVP reservation request to the network. This message is called a Reservation message.**
- **The Reservation message is sent by the receiver and propagated upstream along the exact reverse route of the path message.**
- **This message is a request to the router for a guaranteed level of QoS for the session.**
- **When a router receives a Reservation message it either accepts or rejects the Reservation message based on the available resources.**
- **If a router accepts the reservation, it sets aside router resources for the session.**
- **Once the Reservation message gets to the sender, it knows that the received QoS is in place and starts the transmission.**

On R1:

```
R1(config)#int lo0
R1(config-if)#ip rsvp bandwidth 400 180

R1(config-if)#int f0/0
R1(config-if)#ip rsvp bandwidth 400 180
```

On R2:

```
R2(config)#int lo0
R2(config-if)#ip rsvp bandwidth 400 180

R2(config-if)#int f0/0
R2(config-if)#ip rsvp bandwidth 400 180
```

The first value reserves a total of 400 kbps for all RSVP sessions, and the second value specifies that a single reservation may NOT exceed 180 kbps. If the two numbers are omitted, by default 75% of the bandwidth is reserved for the total and single reservation values.

Task 2

You must perform some testing of RSVP. Configure R1 to send RSVP path messages and R2 to send RSVP reservation messages using TCP; R1 should use a session bandwidth of 10 Kbps with a burst of 40 Kbps.

On R1:

```
R1(config)#ip rsvp sender-host 10.1.12.2 10.1.12.1 tcp 0 0 10 5
```

- **The "ip rsvp sender-host" command configures R1 to generate and send path messages.**
- **10.1.12.2 – This is the IP address of the receiver.**
- **10.1.12.1 – This is the IP address of the sender.**
- **TCP – TCP argument can be UDP or even IP, if IP is used a number in range of (1 – 255) must be used. This task states that TCP must be used.**
- **0 0 - The first zero is the destination port address and the second zero is the source port address. In this case 0 0 is used to tell the router to ignore the destination and the source port addresses.**
- **10 5 – The first value (10) is in Kbps, this specifies the session's bandwidth, and the second value 5 specifies the maximum of burst in Kilobytes per second.**

To verify the configuration:

On R2:

```
R2#Show ip rsvp sender

To              From            Pro DPort Sport Prev Hop        I/F      BPS
10.1.12.2       10.1.12.1       TCP 0     0     10.1.12.1       Fa0/0    10K
```

Enter the following commands to configure R2 to send the reservation messages:

```
R2(config)#ip rsvp reservation-host 10.1.12.2 10.1.12.1 tcp 0 0 ff rate 10 5
```

- **The "ip reservation-host" command configured the local router to send RSVP reservation messages.**
- **10.1.12.2 – This is the IP address of the receiver, in this case R2's F0/0 IP address.**
- **10.1.12.1 – This is the IP address of the sender, in this case R1's F0/0 IP address.**
- **TCP 0 0 – These arguments and values were explained earlier.**
- **FF – This specifies the reservation style. This argument can be FF (Fixed Filter) used for a single reservation, SE (Shared Explicit) which can be shared but limited scope, and WF (Wildcard Filter) which is shared reservation with unlimited scope.**
- **Rate 10 5 - Rate 10 5 – This argument defines the QoS that is requested by the receiver in the reserve message. This message requests a bandwidth of 10 kbps of bandwidth with a maximum burst of 5000 Bytes.**

To verify the configuration:

On R1:

```
R1#Show ip rsvp reservation

To              From            Pro DPort Sport Next Hop      I/F      Fi Serv BPS
10.1.12.2       10.1.12.1       TCP 0     0     10.1.12.2     Fa0/0    FF RATE 10K
```

Task 3

Remove the commands from the previous step before you proceed to the next task.

On R1:

```
R1(config)#No ip rsvp sender-host 10.1.12.2 10.1.12.1 tcp 0 0 10 5
```

On R2:

```
R2(config)#No ip rsvp reservation-host 10.1.12.2 10.1.12.1 tcp 0 0 ff rate 10 5
```

Task 4

R1 has a client (PC1) that is connected to its Lo0 interface and R2 has a client (PC2) that is connected to its Lo0 interface. This client does not have the capability to generate and send RSVP path messages, configure R1 to proxy for PC1 and send the RSVP path messages using TCP to the Lo0 interface of R2; this should use a 10Kbps with a burst of 5KBps.

On R1:

```
R1(config)#ip rsvp sender 2.2.2.2 1.1.1.1 tcp 0 0 2.2.2.2 loopback0 10 5
```

The "Ip rsvp sender" command is used to proxy for its client PC1, in this case, we are simulating a client, and if your router has two Ethernet/Fast Ethernet ports then the second Ethernet port can be used for this purpose.

- **2.2.2.2 - The first IP address is the IP address of the receiver (PC2), in this case 2.2.2.2.**
- **1.1.1.1 - The second IP address is the IP address of the sender (PC1), in this case 1.1.1.1.**

- **TCP - TCP argument can be UDP or even IP, if IP is used a number in range of (1 – 255) must be used. In this case, we have decided to use a TCP session.**
- **0 0 - The first zero is the destination port address and the second zero is the source port address. In this case 0 0 is used to tell the router to ignore the destination and the source port addresses.**
- **2.2.2.2 - The next IP address is the address of the sender or the router closest to the sender.**
- **Loopback0 - this is the previous hop interface, closest to the sender.**
- **10 – In Kbps, this specifies the session's bandwidth.**
- **5 – This specifies the maximum of burst in Kilo Bytes per second.**

To verify the configuration:

On R2:

```
R2#Show ip rsvp sender

To              From            Pro DPort Sport Prev Hop        I/F       BPS
2.2.2.2         1.1.1.1         TCP 0     0     10.1.12.1       Fa0/0     10K
```

Task 5

R2 has a client (PC2) that is connected to its Lo0 interface. This client does not have the capability to generate and send RSVP reserve messages, configure R2 to proxy for PC2 and send RSVP reserve messages.

On R2:

```
R2(config)#ip rsvp reservation 2.2.2.2 1.1.1.1 TCP 0 0 2.2.2.2 F0/0 FF RATE 10 5
```

- **2.2.2.2 – This IP address specifies the IP address of the receiver.**
- **1.1.1.1 – This IP address specifies the IP address of the sender.**
- **TCP - TCP argument can be UDP or even IP, if IP is used a number in range of (1 – 255) must be used. In this case, we have decided to use a TCP session.**
- **0 0 - The first zero is the destination port address and the second zero is the source port address. In this case 0 0 is used to tell the router to ignore the destination and the source port addresses.**
- **2.2.2.2 – This IP address specifies the receiver or the IP address of the router that is closest to the receiver.**
- **Fast Ethernet 0/0 – The interface on the local router that points to the next hop.**
- **FF – This specifies the reservation style. This argument can be FF (Fixed Filter) used for a single reservation, SE (Shared Explicit) which can be shared but limited scope, and WF (Wildcard Filter) which is shared reservation with unlimited scope.**
- **Rate 10 5 – This argument defines the QoS that is requested by the receiver in the reserve message. This message requests a bandwidth of 10 kbps of bandwidth with a maximum burst of 5000 Bytes.**

To verify the configuration:

On R1:

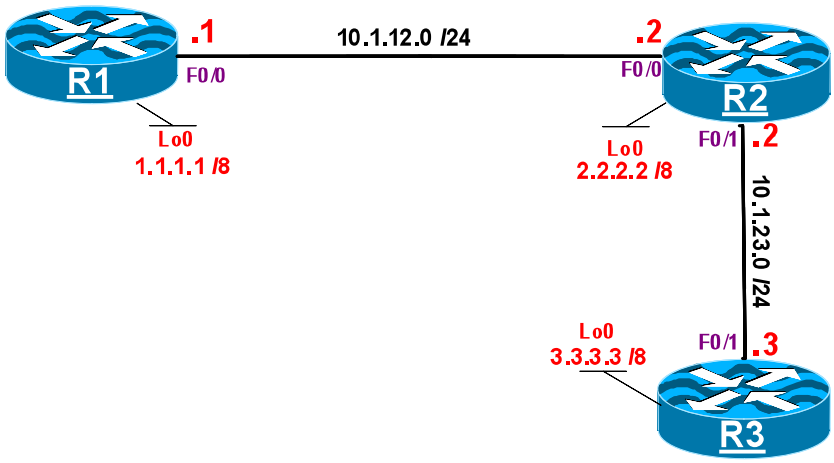
```
R1#Show ip rsvp reservation

To              From            Pro DPort Sport Next Hop      I/F      Fi Serv BPS
2.2.2.2         1.1.1.1         TCP 0     0     10.1.12.2     Fa0/0    FF RATE 10K
```

If this step does not work, you may have to save the config and reload the router to get this working.

Task 6

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 12 – Match Access-Group



Lab Setup:

- Configure the F0/0 interface of R1 and R2 to be in VLAN 12.
- Configure the F0/1 interface of R2 and R3 to be in VLAN 23.
- Use the chart below for IP addressing assignment.
- Configure static routes to provide reachability.

IP Addressing:

Router	Interface	IP address
R1	Lo0 F0/0	1.1.1.1 /8 10.1.12.1 /24
R2	Lo0 F0/0 F0/1	2.2.2.2 /8 10.1.12.2 /24 10.1.23.2 /24
R3	Lo0 F0/1	3.3.3.3 /8 10.1.23.3 /24

Task 1

Configure R2 to classify and mark all IP traffic sourced from R1 (10.1.12.1) and destined to R3 (10.1.23.3) with IP precedence level 2.

On R2:

```
R2(config)#ip cef

R2(config)#access-list 100 permit ip host 10.1.12.1 host 10.1.23.3

R2(config)#Class-map QOS
R2(config-cmap)#match access-group 100

R2(config)#policy-map TST
R2(config-pmap)#class QOS
R2(config-pmap-c)#set ip precedence 2

R2(config)#int F0/0
R2(config-if)#service-policy input TST
```

To verify the configuration:

On R2:

```
R2#Show policy-map TST

  Policy Map TST
    Class QOS
      set ip precedence 2

R2#Show policy-map interface F0/0

FastEthernet0/0

  Service-policy input: TST

    Class-map: QOS (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 100
      QoS Set
        precedence 2
          Packets marked 0
    Class-map: class-default (match-any)
```

```
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

To test the configuration:

On R3:

```
R3(config)#access-list 100 permit ip any any precedence 2 log
R3(config)#access-list 100 permit ip any any log

R3(config)#int F0/1
R3(config-if)#ip access-group 100 in
```

On R1:

```
R1#Ping 10.1.23.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/96/244 ms
```

On R3:

You should see the following console message:

```
%SEC-6-IPACCESSLOGDP: list 100 permitted icmp 10.1.12.1 -> 10.1.23.3 (0/0), 1
packet

R3#Show access-list 100

Extended IP access list 100
    10 permit ip any any precedence immediate log (5 matches)
    20 permit ip any any log
```

On R2:

```
R2#Sh policy-map inter | S QOS

  Class-map: QOS (match-all)
      5 packets, 570 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 100
      QoS Set
```

```
        precedence 2
          Packets marked 5
```

To test this further:

On R2:

```
R2#Clear counters
```

On R3:

```
R3#Clear access-list counter
```

On R1:

```
R1#Ping 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/104/260 ms
```

On R3:

NOTE: the 5 packets matched "permit ip any any" and NOT IP Precedence 2

```
R3#Show access-list 100

Extended IP access list 100
    10 permit ip any any precedence immediate log
    20 permit ip any any log (5 matches)
```

On R2:

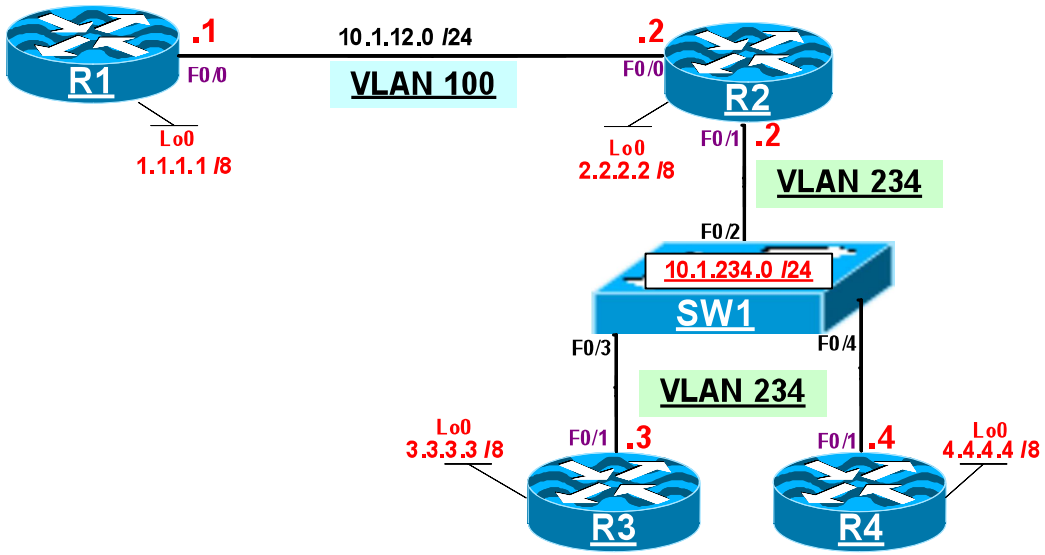
```
R2#Sh policy-map inter

FastEthernet0/0

  Service-policy input: TST

    Class-map: QOS (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group 100
```

```
    QoS Set
      precedence 2
        Packets marked 0

  Class-map: class-default (match-any)
    10 packets, 1140 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any
```

Task 2

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 13 – Match Destination & Source Address MAC



Lab Setup:

- Configure the F0/0 interface of R1 and R2 to be in VLAN 100.
- Configure the F0/1 interface of R2, R3 and R4 to be in VLAN 234.
- Use the chart below for IP addressing assignment.
- Configure static routes to provide reachability.
- Configure the MAC address of R3's F0/1 to be **0000.3333.3333**

IP Addressing:

Router	Interface	IP address
R1	Lo0 F0/0	1.1.1.1 /8 10.1.12.1 /24
R2	Lo0 F0/0 F0/1	2.2.2.2 /8 10.1.12.2 /24 10.1.234.2 /24
R3	Lo0 F0/1	3.3.3.3 /8 10.1.234.3 /24
R4	Lo0 F0/1	4.4.4.4 /8 10.1.234.4 /24

Task 1

Configure R2 to classify and mark all IP traffic from any source destined for the MAC address of R3's F0/1 interface with IP precedence level 1.

On R2:

```
R2(config)#class-map QOS
R2(config-cmap)#match destination-address mac 0000.3333.3333

R2(config)#policy-map TST
R2(config-pmap)#class QOS
R2(config-pmap-c)#set ip precedence 1

R2(config-pmap-c)#int F0/1
R2(config-if)#service-policy out TST
```

To verify the configuration:

On R2:

```
R2#Show policy-map TST

  Policy Map TST
    Class QOS
      set ip precedence 1

R2#Show policy-map interface S QOS

Class-map: QOS (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: destination-address mac 0000.3333.3333
      QoS Set
        precedence 1
          Packets marked 0
```

To test the configuration:

On R3:

```
R3(config)#access-list 100 permit ip any any precedence 1 log
R3(config)#access-list 100 permit ip any any  log
```

```
R3(config)#int F0/1
R3(config-if)#ip access-group 100 in
```

On R1:

```
R1#Ping 10.1.234.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.234.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/132/216 ms
```

On R2:

```
R2#Show policy-map interface

 FastEthernet0/1

  Service-policy output: TST

    Class-map: QOS (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: destination-address mac 0000.3333.3333
      QoS Set
        precedence 1
          Packets marked 0

    Class-map: class-default (match-any)
      6 packets, 630 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

On R3:

```
R3#Sh access-list 100

Extended IP access list 100
    10 permit ip any any precedence priority
    20 permit ip any any
```

On R1:

```
R1#Ping 10.1.234.3
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.234.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/136/260 ms
```

On R2:

```
R2#Clear counters

R2#Show policy-map interface | s qos

Class-map: QOS (match-all)
      5 packets, 570 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: destination-address mac 0000.3333.3333
      QoS Set
        precedence 1
          Packets marked 5
```

On R3:

```
R3#Sh access-list 100

Extended IP access list 100
    10 permit ip any any precedence priority log (5 matches)
    20 permit ip any any log
```

Task 2

Configure R2 to classify and mark all IP traffic from the MAC address of R3 to any destination with IP precedence level 2.

On R2:

```
R2(config)#class-map TASK2
R2(config-cmap)#match source-address mac 0000.3333.3333

R2(config)#policy-map TASK-2
R2(config-pmap)#class TASK2
R2(config-pmap-c)#set ip precedence 2

R2(config-pmap-c)#int F0/1
```

```
R2(config-if)#service-policy in TASK-2
```

On R4:

```
R4#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/80/220 ms
```

On R2:

```
R2#Sh policy-map interface | s TASK2

 FastEthernet0/1

  Service-policy input: TASK-2

    Class-map: TASK2 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: source-address mac 0000.3333.3333
      QoS Set
        precedence 2
          Packets marked 0
```

On R1:

```
R3#Ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/119/184 ms
```

On R2:

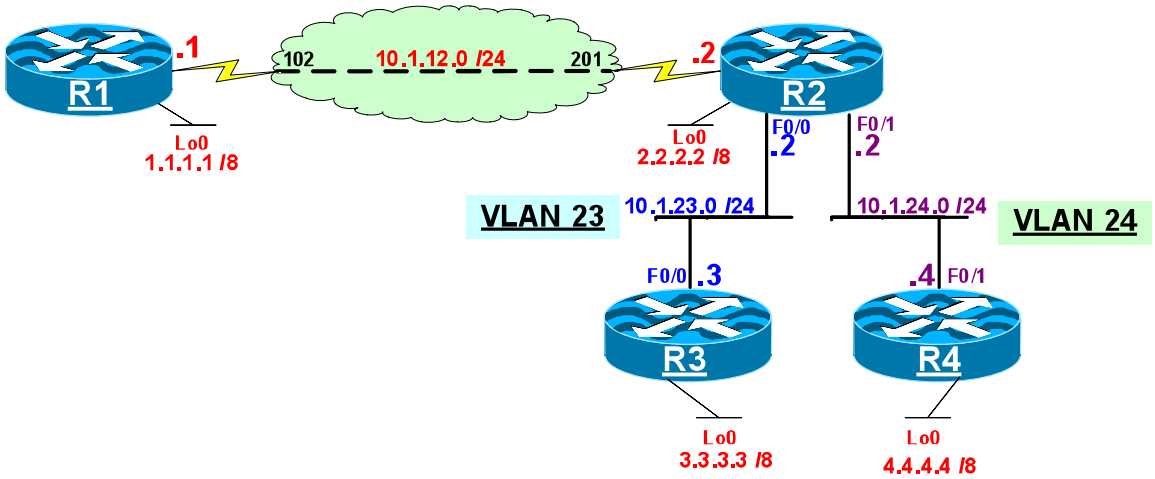
```
R2#Sh policy-map interface | s TASK2

 FastEthernet0/1

  Service-policy input: TASK-2
    Class-map: TASK2 (match-all)
      5 packets, 570 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
```

```
Match: source-address mac 0000.3333.3333
QoS Set
  precedence 2
    Packets marked 5
```

Task 3

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 14 – Match Input-Interface



Lab Setup:

- Configure the F0/0 interface of R2 and R3 to be in VLAN 23.
- Configure the F0/1 interface of R2 and R4 to be in VLAN 24.
- The Frame-Relay connection between R1 and R2 should be configured directly under their main interface.
- Configure static routes to provide reachability to all networks.
- Use the chart below for IP addressing assignment.

IP Addressing:

Router	Interface	IP address
R1	Lo0 S0/0	1.1.1.1 /8 10.1.12.1 /24
R2	Lo0 S0/0 F0/0 F0/1	2.2.2.2 /8 10.1.12.2 /24 10.1.23.2 /24 10.1.24.2 /24
R3	Lo0 F0/0	3.3.3.3 /8 10.1.23.3 /24
R4	Lo0 F0/1	4.4.4.4 /8 10.1.24.4 /24

Task 1

Configure R2 such that all traffic coming from any host connected to its F0/0 interface is marked with IP Precedence of 3, whereas, traffic coming from hosts that are connected to its F0/1 interface is marked with IP precedence 4. This should be verified On R1:.

On R2:

```
R2(config)#class-map QOS0
R2(config-cmap)#match input-interface f0/0
R2(config-cmap)#match access-group 100

R2(config)#class-map QOS1
R2(config-cmap)#match input-interface f0/1
R2(config-cmap)#match access-group 100

R2(config)#policy-map TST
R2(config-pmap)#class QOS0
R2(config-pmap-c)#set precedence 3

R2(config-pmap)#class QOS1
R2(config-pmap-c)#set precedence 4

R2(config-pmap-c)#int s0/0
R2(config-if)#service-policy out TST

R2(config)#access-list 100 permit ip any any
```

To verify & test the configuration:

On R1:

To verify the configuration, an access-list is configured to match any IP traffic destined to any IP address that matches IP Precedence 3 and 4 with a log option. It is applied to the S0/0 interface in the inbound direction, as follows:

```
R1(config)#access-list 100 permit ip any any precedence 3 log
R1(config)#access-list 100 permit ip any any precedence 4 log

R1(config)#int s0/0
R1(config-if)#ip access-group 100 in
```

On R3:

```
R3#Ping 10.1.12.1 repeat 3
```

```
Type escape sequence to abort.
Sending 3, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!
Success rate is 100 percent (3/3), round-trip min/avg/max = 52/52/52 ms
```

On R4:

```
R4#Ping 10.1.12.1 repeat 4

Type escape sequence to abort.
Sending 4, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (4/4), round-trip min/avg/max = 52/52/52 ms
```

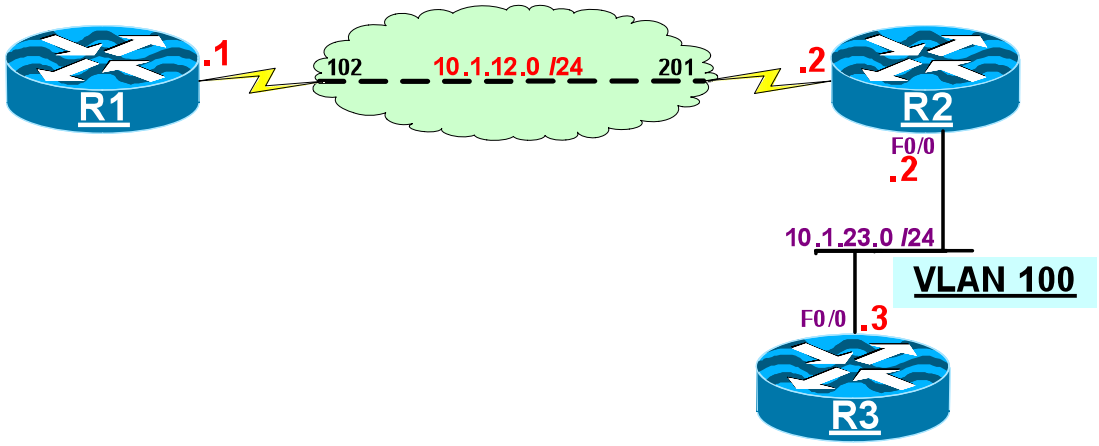
On R1:

```
R1#Show access-list

Extended IP access list 100
    10 permit ip any any precedence flash log (3 matches)
    20 permit ip any any precedence flash-override log (4 matches)
```

Task 3

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 15 – Match Fr-de & Packet Length



Lab Setup:

- Configure the Frame-Relay connection using the S0/0 interface
- Configure the F0/0 interface of R2 and R3 in VLAN 100
- Configure static routes to provide NLRI to all networks
- Use the following IP addressing chart for IP address assignment

IP addressing:

Router	Interface / IP addressing
R1	S0/0 = 10.1.12.1 /24
R2	S0/0 = 10.1.12.2 /24 F0/0 = 10.1.23.2 /24
R3	F0/0 = 10.1.23.3 /24

Task 1

Configure R2 to set the DE bit on all egress packets that are greater than 1000 bytes.

In the following example a class-map called "QOS" has been configured, and a layer 3 packet length has been specified as a match criterion. In the following configuration, packets with a minimum layer 3

packet length of 1000 bytes or bigger will match and are classified.

On R2:

```
R2(config)#class-map QOS
R2(config-cmap)#match packet length min 1000

R2(config-cmap)#policy-map TST
R2(config-pmap)#class QOS
R2(config-pmap-c)#set fr-de

R2(config-if)#int s0/0
R2(config-if)#service-policy output TST
```

On R1:

```
R1(config)#class-map FR
R1(config-cmap)#match fr-de

R1(config)#policy-map TST
R1(config-pmap)#class FR

R1(config-pmap-c)#int s0/0
R1(config-if)#service-policy in TST
```

To verify the configuration:

On R1:

```
R1#Show policy-map interface | s fr

Class-map: FR (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: fr-de
```

To test the configuration:

On R3:

```
R3#Ping 10.1.12.1 size 999 repeat 20

Type escape sequence to abort.
Sending 20, 999-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!
```

```
Success rate is 100 percent (20/20), round-trip min/avg/max = 452/452/457 ms
```

On R1:

```
R1#Show policy-map interface s0/0

 Serial0/0

  Service-policy input: TST
    Class-map: FR (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: fr-de
      Match: any

    Class-map: class-default (match-any)
      22 packets, 20724 bytes
      5 minute offered rate 4000 bps, drop rate 0 bps
      Match: any
```

On R3:

```
R3#Ping 10.1.12.1 size 1000 repeat 10

Type escape sequence to abort.
Sending 10, 1000-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 452/458/492 ms
```

On R1:

```
R1#Show policy-map interface | s fr

Class-map: FR (match-all)
      10 packets, 10040 bytes
      5 minute offered rate 0 bps
      Match: fr-de
```

Task 3

Reconfigure R2 to set the DE bit on all egress packets that are smaller than 1000 Bytes.

On R2:

```
R2(config)#class-map QOS
R2(config-cmap)#No match packet length min 1000
R2(config-cmap)#match packet length max 1000
```

To test the configuration:

On R1:

To clear all the counters:

```
R1#Clear counters

R1#Show policy-map interface s0/0

 Serial0/0

  Service-policy input: TST

    Class-map: FR (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: fr-de
      Match: any

    Class-map: class-default (match-any)
      2 packets, 664 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

To generate some traffic:

On R3:

```
R3#Ping 10.1.12.1 size 1001 repeat 10

Type escape sequence to abort.
Sending 10, 1001-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 452/454/457 ms
```

To see the result:

On R1:

```
R1#Show policy-map interface s0/0

 Serial0/0

  Service-policy input: TST

    Class-map: FR (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: fr-de
      Match: any

    Class-map: class-default (match-any)
      16 packets, 12042 bytes
      5 minute offered rate 1000 bps, drop rate 0 bps
      Match: any
```

To generate traffic with packet sizes less than 1000 Bytes:

On R3:

```
R3#Ping 10.1.12.1 size 999 repeat 10

Type escape sequence to abort.
Sending 10, 999-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 452/453/457 ms
```

On R1:

```
R1#Show policy-map interface s0/0

 Serial0/0

  Service-policy input: TST
    Class-map: FR (match-all)
      10 packets, 10030 bytes
      5 minute offered rate 0 bps
      Match: fr-de

      Match: any
    Class-map: class-default (match-any)
      19 packets, 13038 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Task 4

Reconfigure R2 to set the DE bit on all egress packets that are larger than 1000 and smaller than 1200 Bytes.

On R2:

```
R2(config)#class-map QOS
R2(config-cmap)#No match packet length max 1000
R2(config-cmap)#match packet length min 1000 max 1200
```

On R1:

```
R1#Clear counters

R1#Show policy-map interface s0/0

 Serial0/0

  Service-policy input: TST

    Class-map: FR (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: fr-de
      Match: any

    Class-map: class-default (match-any)
      2 packets, 664 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

On R3:

```
R3#Ping 10.1.12.1 size 999 repeat 10

Type escape sequence to abort.
Sending 10, 999-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 453/453/456 ms
```

On R1:

```
R1#Show policy-map interface s0/0
```

```
 Serial0/0

  Service-policy input: TST

    Class-map: FR (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: fr-de
      Match: any

    Class-map: class-default (match-any)
      14 packets, 11358 bytes
      5 minute offered rate 3000 bps, drop rate 0 bps
      Match: any
```

On R3:

```
R3#Ping 10.1.12.1 size 1201 repeat 10

Type escape sequence to abort.
Sending 10, 1201-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 540/543/545 ms
```

On R1:

```
R1#Show policy-map interface s0/0

 Serial0/0

  Service-policy input: TST

    Class-map: FR (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: fr-de
      Match: any

    Class-map: class-default (match-any)
      31 packets, 25732 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

On R3:

```
R3#Ping 10.1.12.1 size 1100 repeat 10
```

```
Type escape sequence to abort.
Sending 10, 1100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 496/497/501 ms
```

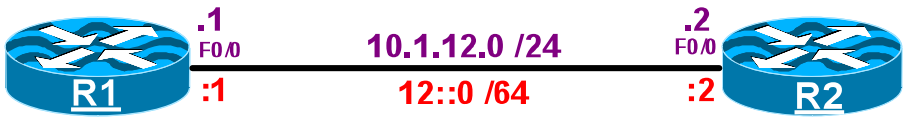
On R1:

```
R1#Show policy-map interface s0/0

 Serial0/0

  Service-policy input: TST

    Class-map: FR (match-all)
      10 packets, 11040 bytes
      5 minute offered rate 0 bps
      Match: fr-de
      Match: any

    Class-map: class-default (match-any)
      35 packets, 27060 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Task 5

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Lab 16 – Match IP Precedence vs. Match Precedence



Lab Setup:

- Configure the F0/0 interface of R1 and R2 to be in VLAN 12.
- Use the chart below for IP addressing assignment.

IP Addressing:

Router	Interface	IP address
R1	F0/0	10.1.12.1 /24 12::1/64
R2	F0/0	10.1.12.2 /24 12::2/64

Task 1

From R2, ping R1 using an extended ping, this traffic should be generated with an IP Precedence of 1 in the TOS Byte. You should use an MQC to verify the traffic's IP Precedence levels of IPv4 packets.

For verification, eight class-maps are configured On R1:, each matching different IP Precedence levels. Then, a policy-map called "TST" is configured. This policy-map references the different classes created earlier, and finally, the policy-map TST is applied to the F0/0 interface of R1 in the inbound direction.

On R1:

```
R1(config)#class-map QOS0
R1(config-cmap)#match ip precedence 0

R1(config)#class-map QOS1
R1(config-cmap)#match ip precedence 1
```

```
R1(config)#class-map QOS2
R1(config-cmap)#match ip precedence 2

R1(config)#class-map QOS3
R1(config-cmap)#match ip precedence 3

R1(config)#class-map QOS4
R1(config-cmap)#match ip precedence 4

R1(config)#class-map QOS5
R1(config-cmap)#match ip precedence 5

R1(config)#class-map QOS6
R1(config-cmap)#match ip precedence 6

R1(config)#class-map QOS7
R1(config-cmap)#match ip precedence 7

R1(config)#policy-map TST

R1(config-pmap)#class QOS0

R1(config-pmap)#class QOS1

R1(config-pmap)#class QOS2

R1(config-pmap)#class QOS3

R1(config-pmap)#class QOS4

R1(config-pmap)#class QOS5

R1(config-pmap)#class QOS6

R1(config-pmap)#class QOS7

R1(config)#int f0/0
R1(config-if)#service-policy input TST
```

To generate the traffic with an IP precedence of 1 from R2:

Note: An extended ping can be used to generate traffic with different IP Precedence levels. Remember that IP Precedence uses the three most significant bits of the TOS Byte, and the decimal value of these bits are: 128 (the most significant), 64 (the second most significant) and 32 (the third most significant). The table below identifies the TOS values and their corresponding IP Precedence values:

→ TOS Byte ←										
IPP	IPP	IPP	D	T	R	ECN	ECN	Decimal	IPP Level	TOS Levels
0	0	0	0	0	0	0	0	0	0	0 – 31
0	0	1	0	0	0	0	0	32	1	32 – 63
0	1	0	0	0	0	0	0	64	2	64 – 95
0	1	1	0	0	0	0	0	96	3	96 – 127
1	0	0	0	0	0	0	0	128	4	128 – 159
1	0	1	0	0	0	0	0	160	5	160 – 191
1	1	0	0	0	0	0	0	192	6	192 – 223
1	1	1	0	0	0	0	0	224	7	224 - 255
128	64	32	16	8	4	2	1	Decimal Conversion		

On R2:

```
R2#ping
Protocol [ip]: → Press Enter
Target IP address: 10.1.12.1
Repeat count [5]: → Press Enter
Datagram size [100]: → Press Enter
Timeout in seconds [2]: → Press Enter
Extended commands [n]: y
Source address or interface: 10.1.12.2
Type of service [0]: 32 ←
Set DF bit in IP header? [no]: → Press Enter
Validate reply data? [no]: → Press Enter
Data pattern [0xABCD]: → Press Enter
Loose, Strict, Record, Timestamp, Verbose[none]: → Press Enter
Sweep range of sizes [n]: → Press Enter
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.12.2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

On R1:

```
R1#Show policy-map interface f0/0

 FastEthernet0/0

  Service-policy input: TST
    Class-map: QOS0 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: ip precedence 0
```

```
    Class-map: QOS1 (match-all)
      5 packets, 570 bytes
      5 minute offered rate 0 bps
      Match: ip precedence 1

    Class-map: QOS2 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: ip precedence 2

    Class-map: QOS3 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: ip precedence 3

    Class-map: QOS4 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: ip precedence 4

    Class-map: QOS5 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: ip precedence 5

    Class-map: QOS6 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: ip precedence 6

    Class-map: QOS7 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: ip precedence 7

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Note: The output of the above Show command reveals that ONLY IP Precedence level 1 was matched.

To clear the counter:

```
R1#Clear counter
```

To generate IPv6 traffic:

On R2:

```
R2#ping ipv6
Target IPv6 address: 12::1
Repeat count [5]: → Press Enter
Datagram size [100]: → Press Enter
Timeout in seconds [2]: → Press Enter
Extended commands? [no]: Y
Source address or interface: 12::2
UDP protocol? [no]: → Press Enter
Verbose? [no]: → Press Enter
Precedence [0]: 4
Include hop by hop option? [no]: → Press Enter
Include destination option? [no]: → Press Enter
Sweep range of sizes? [no]: → Press Enter
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12::1, timeout is 2 seconds:
Packet sent with a source address of 12::2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/1/4 ms
```

To verify the configuration:

On R1:

```
R1#Show policy-map interface F0/0

 FastEthernet0/0

  Service-policy input: TST

    Class-map: QOS0 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: ip precedence 0

    Class-map: QOS1 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match: ip precedence 1

    Class-map: QOS2 (match-all)
      0 packets, 0 bytes
```

```
    5 minute offered rate 0 bps
    Match: ip precedence 2

  Class-map: QOS3 (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps
    Match: ip precedence 3

  Class-map: QOS4 (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps
    Match: ip precedence 4

  Class-map: QOS5 (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps
    Match: ip precedence 5

  Class-map: QOS6 (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps
    Match: ip precedence 6

  Class-map: QOS7 (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps
    Match: ip precedence 7

  Class-map: class-default (match-any)
    5 packets, 570 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any
```

Note: IPv6 traffic was matched to "class class-default" and the precedence level was not detected. This is because when the "class-maps" were configured, "match ip precedence" was used, when "match ip precedence" is used, it only matches on IPv4 traffic and NOT IPv6.

Task 2

Remove the service-policy, policy-map, and the class-map from the previous task.

On R1:

```
R1(config)#int f0/0
R1(config-if)#No service-policy input TST

R1(config)#No policy-map TST

R1(config)#No class-map QOS0
R1(config)#No class-map QOS1
R1(config)#No class-map QOS2
R1(config)#No class-map QOS3
R1(config)#No class-map QOS4
R1(config)#No class-map QOS5
R1(config)#No class-map QOS6
R1(config)#No class-map QOS7
```

Task 3

From R2, R1 should be pinged using an extended ping; this traffic should be generated with an IP Precedence of 4. You should use an MQC to verify the traffic's IP Precedence levels of IPv6 packets.

Note: Since the IP Precedence level of IPv6 packets should be matched, the match statement in the "class-map" should NOT include the "IP" keyword; remember that "IP" in IOS references IPv4 and NOT IPv6.

On R1:

```
R1(config)#class-map p0
R1(config-cmap)#match precedence 0

R1(config)#class-map p1
R1(config-cmap)#match precedence 1

R1(config)#class-map p2
R1(config-cmap)#match precedence 2

R1(config)#class-map p3
R1(config-cmap)#match precedence 3

R1(config)#class-map p4
R1(config-cmap)#match precedence 4

R1(config)#class-map p5
```

```
R1(config-cmap)#match precedence 5

R1(config)#class-map p6
R1(config-cmap)#match precedence 6

R1(config)#class-map p7
R1(config-cmap)#match precedence 7

R1(config)#policy-map TST
R1(config-pmap)#class p0
R1(config-pmap)#class p1
R1(config-pmap)#class p2
R1(config-pmap)#class p3
R1(config-pmap)#class p4
R1(config-pmap)#class p5
R1(config-pmap)#class p6
R1(config-pmap)#class p7

R1(config-pmap)#int f0/0
R1(config-if)#service-policy in TST
```

To test the configuration:

On R2:

```
R2#ping ipv6
Target IPv6 address: 12::1
Repeat count [5]: → Press Enter
Datagram size [100]: → Press Enter
Timeout in seconds [2]: → Press Enter
Extended commands? [no]: Y
Source address or interface: 12::2
UDP protocol? [no]: → Press Enter
Verbose? [no]: → Press Enter
Precedence [0]: 4
Include hop by hop option? [no]: → Press Enter
Include destination option? [no]: → Press Enter
Sweep range of sizes? [no]: → Press Enter
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12::1, timeout is 2 seconds:
Packet sent with a source address of 12::2
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/1/4 ms
```

To verify the configuration:

On R1:

```
R1#Sh policy-map inter

 FastEthernet0/0

  Service-policy input: TST

    Class-map: P0 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match:  precedence 0

    Class-map: P1 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match:  precedence 1

    Class-map: P2 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match:  precedence 2

    Class-map: P3 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match:  precedence 3

    Class-map: P4 (match-all)
      5 packets, 570 bytes
      5 minute offered rate 0 bps
      Match:  precedence 4

    Class-map: P5 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match:  precedence 5

    Class-map: P6 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match:  precedence 6

    Class-map: P7 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
```

```
      Match:  precedence 7

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Note: Precedence level 4 was matched in the IPv6 packets. The "match precedence" or "match dscp" command in the "class-map" can match IPv4 and/or IPv6, whereas, the "match ip precedence" or "match ip dscp" command in the "class-map" can ONLY match the precedence or dscp levels in IPv4 packets ONLY. The following test is conducted to reveal this fact:

```
R1#Sh policy-map interface

Ethernet0/0

 Service-policy input: TST

    Class-map: P0 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match:  precedence 0

    Class-map: P1 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match:  precedence 1

    Class-map: P2 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match:  precedence 2

    Class-map: P3 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
      Match:  precedence 3

    Class-map: P4 (match-all)
      5 packets, 570 bytes
      5 minute offered rate 0 bps
      Match:  precedence 4

    Class-map: P5 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps
```

```
    Match:  precedence 5

  Class-map: P6 (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps
    Match:  precedence 6

  Class-map: P7 (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps
    Match:  precedence 7

  Class-map: class-default (match-any)
    0 packets, 0 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any
```

Task 4

Stop all routers in the console window and exit. Stop the Server and press the "Erase Start" launcher before proceeding.

Advanced
CCIE SERVICE PROVIDER
v3.0

www.MicronicsTraining.com

Narbik Kocharians
CCIE #12410
R&S, Security, SP

Paul Negron
CCIE #14856
SP

Switching

Lab 1
Switching Port-Types

IP Addressing chart:

Router	Interface / IP Addressing	Connecting To:
R1	E0/0 = 10.1.1.1 /24	SW1's F0/2
R2	E0/0 = 10.1.1.2 /24	SW2's F0/2
SW3	F0/1 = 10.1.1.3/24	SW1's F0/1
SW4	F0/1 = 10.1.1.4 /24	SW2's F0/1

Task 1

Configure SW1 and SW2 to allow connectivity between the devices that are directly connected to each switch ONLY. The default port-type should be used with a vlan of 1234.

On SW1

```
SW1#sh ip int brie
Interface              IP-Address      OK? Method Status                Protocol
Vlan1                  unassigned      YES unset  up                    up
FastEthernet0          unassigned      YES unset  down                  down
FastEthernet0/1        unassigned      YES unset  administratively down down
FastEthernet0/2        unassigned      YES unset  administratively down down
FastEthernet0/3        unassigned      YES unset  administratively down down
FastEthernet0/4        unassigned      YES unset  administratively down down
```

```
FastEthernet0/5        unassigned      YES unset  administratively down down
FastEthernet0/6        unassigned      YES unset  administratively down down
FastEthernet0/7        unassigned      YES unset  administratively down down
FastEthernet0/8        unassigned      YES unset  administratively down down
FastEthernet0/9        unassigned      YES unset  administratively down down
FastEthernet0/10       unassigned      YES unset  administratively down down
FastEthernet0/11       unassigned      YES unset  administratively down down
FastEthernet0/12       unassigned      YES unset  administratively down down
FastEthernet0/13       unassigned      YES unset  administratively down down
FastEthernet0/14       unassigned      YES unset  administratively down down
FastEthernet0/15       unassigned      YES unset  administratively down down
FastEthernet0/16       unassigned      YES unset  administratively down down
FastEthernet0/17       unassigned      YES unset  administratively down down
FastEthernet0/18       unassigned      YES unset  administratively down down
FastEthernet0/19       unassigned      YES unset  administratively down down
FastEthernet0/20       unassigned      YES unset  administratively down down
FastEthernet0/21       unassigned      YES unset  administratively down down
FastEthernet0/22       unassigned      YES unset  administratively down down
FastEthernet0/23       unassigned      YES unset  administratively down down
FastEthernet0/24       unassigned      YES unset  administratively down down
GigabitEthernet0/1     unassigned      YES unset  up                    up
GigabitEthernet0/2     unassigned      YES unset  down                  down
```

Unlike normal Ethernet switches, the Metro Ethernet or ME3400 comes up with ALL ports disabled except for the 2 GE uplink ports. Port G0/1 has a device connected so it comes up, which is the only reason why the VLAN1 interface is also up.

```
SW1(config)#int f0/1
SW1(config-if)#no shut

SW1(config-if)#int f0/2
SW1(config-if)#no shut
SW1(config-if)#end

R1#ping 10.1.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

R1 cannot ping SW3 by default as it would do with a normal Ethernet switch. The port comes up with a default port-type of User Network Interface or UNI. A UNI port connects to customers and will not allow commiunication from one UNI port to another UNI port on the same switch. Even if they are in the same VLAN.

```
SW1(config)#int f0/1
SW1(config-if)#port-type uni

SW1(config-if)#int f0/2
```

```
SW1(config-if)#port-type uni
```

On SW2

```
SW2(config)#int f0/1
SW2(config-if)#port-type uni
SW2(config-if)#no shut

SW2(config-if)#int f0/2
SW2(config-if)#port-type uni
SW2(config-if)#no shut
```

Now that the ports have been labeled to CLEARLY show they are UNI ports, let's allow the connectivity to flow to the directly connected devices to each switch.

On SW1

```
SW1(config)#vlan 1234
SW1(config-vlan)#uni-vlan ?
  community  UNI/ENI community VLAN
  isolated   UNI/ENI isolated VLAN
```

A feature called uni-vlan can be implemented to establish connectivity between two UNI ports on the same switch. Although it is similar to Private VLANS, a uni-vlan cannot participate in PRIVATE VLANS if implemented on the switch. Only the isolated and community vlans are configurable. The isolated VLAN is the default type on a uni port. A community VLAN can only talk to other community vlans with the same number on the local switch.

```
SW1(config-vlan)#uni-vlan community

SW1(config)#int f0/1
SW1(config-if)#switchport access vlan 1234

SW1(config-if)#int f0/2
SW1(config-if)#switchport access vlan 1234
```

The newly created uni-vlan is then added to the port.

On SW2

```
SW1(config)#vlan 1234
SW1(config-vlan)#uni-vlan community

SW2#sh vlan uni-vlan
```

```
VLAN Type                Ports
---- ----------------- ------------------------------------------------
-----
1234 UNI community       Fa0/1, Fa0/2
```

VTP does not operate on the ME3400, therefore the VLANS must manually be created on each switch. The macro that allows a vlan to be applied and then created as a result also works like the standard Cisco Ethernet switch behavior.

On R1

```
R1#ping 10.1.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:
.!!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 1/1/1 ms
```

On R2

```
R2#ping 10.1.1.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.4, timeout is 2 seconds:
.!!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 1/2/4 ms

R2#ping 10.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R2#ping 10.1.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Although connectivity has been established on each of the local switches a different port-type is required to establish connectivity across the switches.

Task 2

Create vlan 20 for SW3 and SW4 called VLAN-TWENTY. Allow connectivity between these switches to flow along the f0/19 path. R1 and R2 should only use f0/20.

ON SW1

```
SW1(config)#vlan 20
SW1(config-vlan)#name VLAN-TWENTY

SW1(config-vlan)#int f0/1
SW1(config-if)#switchport access vlan 20

SW1(config-if)#int f0/19
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 20
SW1(config-if)#no shut
```

ON SW2

```
SW2(config)#int f0/1
SW2(config-if)#switchport access vlan 20
% Access VLAN does not exist. Creating vlan 20

SW2(config-if)#vlan 20
SW2(config-vlan)#name VLAN-TWENTY

SW2(config-if)#int f0/19
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk allowed vlan 20
SW2(config-if)#no shut
```

ON SW3

```
SW3#ping 10.1.1.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.4, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

As expressed in the previous task, a different port-type is required to allow switch to switch vlan communication. Port f0/19 on both switches defaults to a UNI port. UNI port to UNI port is no allowed on the same switch, even when configuring trunk ports.

ON SW1

```
SW1(config)#int f0/19
SW1(config-if)#port-type nni
```

ON SW2

```
SW1(config)#int f0/19
SW1(config-if)#port-type nni
```

ON SW3

```
SW3#ping 10.1.1.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.4, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms
```

A Network to Network Interface or NNI is required as the port type to allow vlans to span between switches across the trunk ports. The pings are successful between SW3 and SW4.

ON R1

```
R1#ping 10.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

The pings are not successful. Traffic from R1 and R2 cannot cross the f0/19 port due to the filter.

ON SW1

```
SW1(config)#int f0/20
SW1(config-if)#port-type nni
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 1234
```

ON SW2

```
SW2(config)#int f0/20
SW2(config-if)#port-type nni
SW2(config-if)#switchport mode trunk
```

```
SW2(config-if)#switchport trunk allowed vlan 1234
SW2(config-if)#no shut

R1#ping 10.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms
```

The pings are successful. R1 and R2 are still participating in a community VLAN and can reach each other through the NNI Trunk Port.

Task 3

Allow CDP to be seen between each switch and its'directly connected devices

On R1

```
R1#sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
```

On R2

```
R2# sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
```

On SW3

```
SW3#sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
```

On SW4

```
SW4#sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
```

On SW1

```
SW1#sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW2              Fas 0/20          149              S I   ME-3400E- Fas 0/20
SW2              Fas 0/19          131              S I   ME-3400E- Fas 0/19
```

On SW2

```
SW2#sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW1              Fas 0/20          176              S I   ME-3400E- Fas 0/20
SW1              Fas 0/19          169              S I   ME-3400E- Fas 0/19
```

The only port-type that is allowing the user to see cdp updates is the NNI between SW1 and SW2.

On SW1

```
SW1(config)#int f0/1
SW1(config-if)# cdp enable
                ^
% Invalid input detected at '^' marker.
```

CDP is not allowed to be configured on a UNI port. In fact Layer 2 control traffic such as STP, LLDP, LACP or PAgP are not allowed either.

```
SW1(config)#int f0/1
SW1(config-if)#port-type eni
SW1(config-if) cdp enable


SW1(config-if)#int f0/2
SW1(config-if)#port-type eni
SW1(config-if) cdp enable
```

The Enhanced Network Interface or ENI port-type offers the same behavior of UNI by isolating devices connected to them and the protocol stated above are disabled by default. However the Layer 2 control Traffic can be enabled on the port and passed to neighboring device.

On SW2

```
SW2(config)#int f0/1
SW2(config-if)#port-type eni
SW2(config-if) cdp enable


SW2(config-if)#int f0/2
SW2(config-if)#port-type eni
SW2(config-if) cdp enable
```

On R1

```
R1#sh cdp n | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
```

```
SW1              Eth 0/0          172           S I      ME-3400E- Fas 0/2
```

On R2

```
R2# sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW2              Eth 0/0           153           S I      ME-3400E- Fas 0/2
```

On SW3

```
SW3#sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW1              Fas 0/1           158           S I      ME-3400E- Fas 0/1
```

On SW4

```
SW4#sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW2              Fas 0/1           159           S I      ME-3400E- Fas 0/1
```

On SW1

```
SW1#sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW2              Fas 0/20          170              S I   ME-3400E- Fas 0/20
SW2              Fas 0/19          153              S I   ME-3400E- Fas 0/19
SW3              Fas 0/1           163              S I   WS-C3550- Fas 0/1
R1               Fas 0/2           154               R    7206VXR   Eth 0/0
```

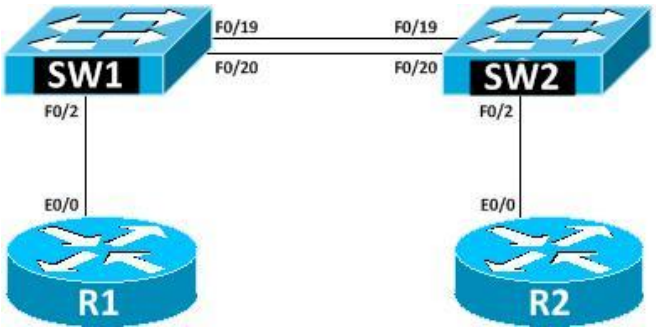
On SW2

```
SW2#sh cdp nei | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW4              Fas 0/1           175              S I   WS-C3550- Fas 0/1
SW1              Fas 0/20          160              S I   ME-3400E- Fas 0/20
SW1              Fas 0/19          153              S I   ME-3400E- Fas 0/19
R2               Fas 0/2           171               R    7206VXR   Eth 0/0
```

Task 4

Erase start and reload the devices upon completion of the devices.

Lab 2
Basic L2 Tunneling Mechanisms



IP Addressing chart:

Router	Interface / IP Addressing	Connecting To:
R1	E0/0 = 10.1.1.1 /24	SW1's F0/2
R2	E0/0 = 10.1.1.2 /24	SW2's F0/2
SW3	F0/1 = 10.1.1.3/24	SW1's F0/1
SW4	F0/1 = 10.1.1.4 /24	SW2's F0/1

Task 1

Provide connectivity between SW1 and SW2 using f0/19 and f0/20. The links should negotiate a trunk acting as a single link using a Standards based method.

On SW1

```
SW1(config)#int range f0/19-20
SW1(config-if-range)#port-type nni
SW1(config-if-range)#switchport mode trunk
SW1(config-if-range)#channel-group 1 mode active
SW1(config-if-range)#no shut
```

On SW2

```
SW2(config)#int range f0/19-20
SW2(config-if-range)#port-type nni
```

```
SW2(config-if-range)#switchport mode trunk
SW2(config-if-range)#channel-group 1 mode active
SW2(config-if-range)#no shut

SW2#show etherchannel summ | b Group
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SU)         LACP      Fa0/19(P)   Fa0/20(P)
```

Etherchannels should be used based on the instructions and an NNI port-type must be chosen since the default UNI port-type will not participate in negotiating an etherchannel. "On" is the mode supported for a UNI. The method chosen is Link Aggregation Control Protocol or LACP since it is standards based.

Task 2

Configure a protocol that allows connectivity between R1 and R2 vlan interfaces across the backbone switches. A single VLAN 88 should be allowed to cross the Port Channel.

On SW1

```
SW1(config)#int f0/2
SW1(config-if)#switchport mode dot1q-tunnel
SW1(config-if)#switchport access vlan 88
% Access VLAN does not exist. Creating vlan 88
SW1(config-if)#exit

SW1(config)#int port-channel 1
SW1(config-if)#switch trunk allowed vlan 88
```

On SW2

```
SW2(config)#int f0/2
SW2(config-if)#switchport mode dot1q-tunnel
SW2(config-if)#switchport access vlan 88
% Access VLAN does not exist. Creating vlan 88

SW2(config-if)#int port-channel 1
SW2(config-if)#switch trunk allowed vlan 88
```

QinQ encapsulation places a dot1q header with protocol ID of 0x8100 as the traffic enters the F0/2 port of the switch. Cisco has defined a different protocol ID to be placed on top of the vlan 10 or vlan 20 frame of 0x9100 to denote QinQ encapsulation. Only the top most label "88" needs to be allowed across the Port Channel as a result.

On R1

```
R1#ping 10.10.10.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

R1#ping 10.20.20.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.20.20.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

The pings work, which satisfies the Task requirement.

Task 3

Cleanup the the previous task by removing the QinQ Configuration.

On SW1

```
SW1(config)#int f0/2
SW1(config-if)#NO switchport mode dot1q-tunnel
SW1(config-if)#NO switchport access vlan 88
SW1(config-if)#exit
```

On SW2

```
SW2(config)#int f0/2
SW2(config-if)#NO switchport mode dot1q-tunnel
SW2(config-if)#NO switchport access vlan 88
```

Task 4

R1 and R2 are to be configured with OSPF to allow loopback reachabilty between the devices. CDP updates should allow the 2 routers to see each other as if directly connected.

On SW1

```
SW1(config)#int f0/2
SW1(config-if)#l2protocol-tunnel cdp
SW1(config-if)#switchport mode trunk
```

The Layer 2 Protocol Tunnel mechanism that will allow for the task to be completed. L2PT Tunneling enables the switch to identify a subset of well known Layer 2 Protocols that can be identified and tunneled by the switch. Unlike QinQ, the vlans being used by the trunk must also exist on the switch and be allowed to pass on an interiswitch trunk link.

On SW2

```
SW2(config)#int f0/2
SW2(config-if)#l2protocol-tunnel cdp
SW2(config-if)#switchport mode trunk
```

On R1

```
R1(config)#router ospf 1
R1(config-router)#network 10.10.10.1 0.0.0.0 area 0
R1(config-router)#network 10.20.20.1 0.0.0.0 area 0
R1(config-router)#netw 1.1.1.1 0.0.0.0 area 0
```

On R1

```
R1(config)#router ospf 1
R1(config-router)#network 10.10.10.1 0.0.0.0 area 0
R1(config-router)#network 10.20.20.1 0.0.0.0 area 0
R1(config-router)#netw 1.1.1.1 0.0.0.0 area 0

R1#sh ip ospf neighbor

R1#sh cdp neighbor
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
```

Notice how OSPF an CDP do not come up due to VLAN's 10 and 20 do not exist on the switch.

On SW1

```
SW1(config)#vlan 10
SW1(config-vlan)#exit

SW1(config)#vlan 20
SW1(config-vlan)#exit

SW1(config)#int port-channel 1
SW1(config-if)#switchport trunk allowed vlan 10,20
```

On SW2

```
SW1(config)#vlan 10
SW1(config-vlan)#exit

SW1(config)#vlan 20
SW1(config-vlan)#exit

SW1(config)#int port-channel 1
SW1(config-if)#switchport trunk allowed vlan 10,20
```

On R1

```
R1#sh ip route ospf | b Gateway
Gateway of last resort is not set

      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/11] via 10.20.20.2, 00:01:17, Ethernet0/0.20
                 [110/11] via 10.10.10.2, 00:01:17, Ethernet0/0.10

R1#sh cdp neighbor
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
```

Even though OSPF now comes up, CDP does not due to a different problem. On normal Ethernet switches, the vlan that carries cdp by default in 1, which is the Management Vlan. This vlan CANNOT be pruned from a trunk link. On the ME3400, when we allowed vlan 10 and 20 ONLY in the allowed vlan list, vlan 1 wa actually pruned. Therefore we must add vlan 1 to the allowed list to be able to see the CDP updates on R1 and R2.

On SW1

```
SW1#sh int port-channel 1 trunk
```

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

Port        Vlans allowed on trunk
Po1         10,20

Port        Vlans allowed and active in management domain
Po1         10,20

Port        Vlans in spanning tree forwarding state and not pruned
Po1         10,20

SW1(config)#int port-channel 1
SW1(config-if)#switchport trunk allowed vlan add 1
SW1(config-if)#end


SW1#sh int port-channel 1 trunk

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

Port        Vlans allowed on trunk
Po1         1,10,20

Port        Vlans allowed and active in management domain
Po1         1,10,20

Port        Vlans in spanning tree forwarding state and not pruned
Po1         1,10,20
```

On SW2

```
SW2(config)#int port-channel 1
SW2(config-if)#switchport trunk allowed vlan add 1
```

On R1

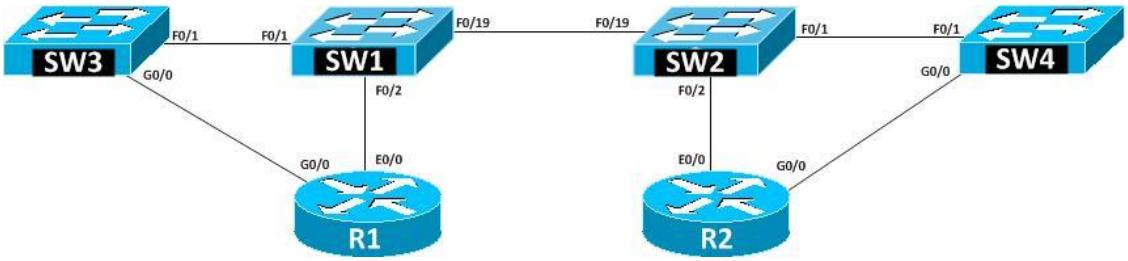
```
R1#sh ip route ospf | b Gateway
Gateway of last resort is not set

      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/11] via 10.20.20.2, 00:08:56, Ethernet0/0.20
                 [110/11] via 10.10.10.2, 00:08:56, Ethernet0/0.10

R1#sh cdp neighbor
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
```

```
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
R2               Eth 0/0           159           R        7206VXR   Eth 0/0


R1#ping 2.2.2.2 source lo0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Task 4

Erase start and reload all the devices used in the task

Lab 3
Provider Bridging (PB)



IP Addressing chart:

Router	Interface / IP Addressing	Connecting To:
R1	g0/0 = 10.20.20.1 /24 e0/0 = 10.10.10.1 /24	SW3's g0/1 SW1's F0/2
R2	g0/0 = 10.20.20.2 /24 e0/0 = 10.10.10.2 /24	SW4's g0/1 SW2's F0/2

Task 1

Configure connectivity between R1 and R2 through SW1 and SW2. They should see cdp updates directly from each other. The "l2protocol-tunnel" command is not permitted. All configurations should be performed on SW1 and SW2. Use VLAN 10.

On SW1

```
SW1(config)#int f0/2
SW1(config-if)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10

SW1(config-if)#ethernet dot1ad uni c-port
SW1(config-if)#no shut

SW1(config)#int f0/19
```

```
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 10
SW1(config-if)#port-type nni
SW1(config-if)#no shut
```

On SW2

```
SW2(config)#int f0/2
SW2(config-if)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10

SW2(config-if)#ethernet dot1ad uni c-port
SW2(config-if)#no shut

SW2(config-if)#int f0/19
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk allowed vlan 10
SW2(config-if)#port-type nni
SW2(config-if)#no shut
```

Verifying the Configuration:

On SW1

```
SW1#sh vlan brief | i 10
                                                Fa0/10, Fa0/11, Fa0/12,
Fa0/13
10   VLAN0010                         active    Fa0/2
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

SW1#sh int f0/19 trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/19      on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/19      10

Port        Vlans allowed and active in management domain
Fa0/19      10

Port        Vlans in spanning tree forwarding state and not pruned
```

```
Fa0/19      10
```

On SW2

```
SW2#sh vlan brief | i 10
                                                Fa0/10, Fa0/11, Fa0/12,
Fa0/13
10   VLAN0010                         active    Fa0/2
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

SW2#show int f0/19 trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/19      on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/19      10

Port        Vlans allowed and active in management domain
Fa0/19      10

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/19      10
```

On R1

```
R1#ping 10.10.10.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms

R1#show cdp neighbor
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW1              Eth 0/0           156           S I      ME-3400E- Fas 0/2
```

R1 does not see the cdp updates since the configuration is similar to the effects of QinQ with the exception that VLAN 10 must exist on the Core switches.

On SW1

```
SW1(config)#
SW1(config-if)#l2protocol forward cdp

SW1(config-if)#int f0/19
SW1(config-if)#ethernet dot1ad nni
```

Layer 2 forwarding is allowed on the port. The trunk must be converted to a dot1d nni to allow the forwarded protocol to be extended to the next switch.

On SW2

```
SW2(config-if)#int f0/2
SW2(config-if)#l2protocol forward cdp

SW2(config-if)#int f0/19
SW2(config-if)#ethernet dot1ad nni
```

On SW1

```
SW1#show port-type nni
Port       Name               Vlan       Port Type
---------- ------------------ ---------- ----------------------------
Fa0/2                         10         Network Node Interface           (nni)
Fa0/19                        trunk      Network Node Interface           (nni)
Gi0/1                         1          Network Node Interface           (nni)
Gi0/2                         1          Network Node Interface           (nni)
```

Notice how the port-type changed from the default uni to nni. This occurred as a result of configuring dot1ad on the port to allow the wellknown mac-address types to be permitted for tunneling. This does not count against the 4 configurable nni port-types that are allowed on this platform with this code.

On R1

```
R1#sh cdp n
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
R2               Eth 0/0           167           R        7206VXR   Eth 0/0
```

On R2

```
R2#sh cdp n | b Device
```

```
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
R1               Eth 0/0           171           R        7206VXR   Eth 0/0

R2#ping 10.10.10.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Task 2- (Lab Cleanup)

Shutdown the interface between the R1 to SW1 and R2 to SW2. The interface g0/0 interface on R1 and R2 should be activated. The routers should participate in vlan 20 on SW3 and SW4.

On SW1

```
SW1(config)#int f0/2
SW1(config-if)#shut
```

On R1

```
R1(config)#int e0/0
R1(config-if)#shut

R1(config-if)#int g0/0
R1(config-if)#no shut
```

On SW2

```
SW2(config)#int f0/2
SW2(config-if)#shut
```

On R2

```
R2(config)#int e0/0
R2(config-if)#shut

R2(config-if)#int g0/0
R2(config-if)#no shut
```

On SW3

```
SW3(config)#vlan 20
SW3(config-vlan)#exit

SW3(config)#int g0/1
SW3(config-if)#switchport mode access
SW3(config-if)#switchport access vlan 20
```

On SW4

```
SW4(config)#int g0/1
SW4(config-if)#switchport mode access
SW4(config-if)#switchport access vlan 20
% Access VLAN does not exist. Creating vlan 20
```

Task3

R1 and R2 should be connected through the following path: SW3> SW1 >SW2 > SW4. The switches should configure for trunking in the F0/1 port connected to the SW1 and SW2. SW3 should pass vtp and cdp updates to SW4. L2PT is not permitted.

On SW3

```
SW3(config-if)#int f0/1
SW3(config-if)#switchport trunk encap dot1q
SW3(config-if)#switchport mode trunk
SW3(config-if)#switchport trunk allowed vlan 20
SW3(config-if)#no shut
```

Sw3 is configured as trunk port that only passes vlan 20.

On SW4

```
SW4(config)#int f0/1
SW4(config-if)#switchport trunk encap dot1q
SW4(config-if)#switchport mode trunk
SW4(config-if)#switchport trunk allowed vlan 20
SW4(config-if)#no shut
```

On SW1

```
SW1(config)#vlan 20
```

```
SW1(config)#int f0/1
SW1(config-if)#ethernet dot1ad uni c-port
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 20,1
SW1(config-if)#l2protocol forward cdp
SW1(config-if)#l2protocol forward vtp
SW1(config-if)#no shut
SW1(config-if)#exit

SW1(config-vlan)#int f0/19
SW1(config-if)#switchport trunk allowed vlan 20,1
```

The f0/1 port on SW1 is configured for dot1ad as a trunk port. If it is not configured as a trunk it will cause the port to change to an inconsistent state for not matching the trunk configured on SW3. This configuration acts more like QinQ except the VTP and CDP updates can be forwarded. VLAN 1 must be allowed on the port connected to the customer switch and the trunk to allow for all of the management protocols to work.

On SW2

```
SW2(config)#int f0/1
SW2(config-if)#ethernet dot1ad uni c-port
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk allowed vlan 20,1
SW2(config-if)#l2protocol forward cdp
SW2(config-if)#l2protocol forward vtp
SW2(config-if)#no shut
SW2(config-if)#exit

SW2(config)#vlan 20

SW2(config-vlan)#int f0/19
SW2(config-if)#switchport trunk allowed vlan 20,1
```

On R1

```
R1#ping 10.20.20.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.20.20.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
R1#
```

On SW3

```
SW3#show vtp status
VTP Version                     : 2
Configuration Revision          : 1
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 6
VTP Operating Mode              : Server
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x05 0x05 0xF4 0x88 0xCB 0x99 0xA9 0x58
Configuration last modified by 0.0.0.0 at 3-3-93 17:35:42
Local updater ID is 0.0.0.0 (no valid interface found)
```

On SW4

```
SW4#show vtp status
VTP Version                     : running VTP1 (VTP2 capable)
Configuration Revision          : 1
Maximum VLANs supported locally : 1005
Number of existing VLANs        : 6
VTP Operating Mode              : Server
VTP Domain Name                 :
VTP Pruning Mode                : Disabled
VTP V2 Mode                     : Disabled
VTP Traps Generation            : Disabled
MD5 digest                      : 0x05 0x05 0xF4 0x88 0xCB 0x99 0xA9 0x58
Configuration last modified by 0.0.0.0 at 3-3-93 17:35:41
Local updater ID is 0.0.0.0 (no valid interface found)
```

VTP is first verified to be unconfigured and vlans will not be propagated.

On SW3

```
SW3#sh cdp n | b Device
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW4              Fas 0/1           163           S I      WS-C3550- Fas 0/1
R1               Gig 0/1           173            R       7206VXR   Gig 0/0

SW3(config)#vtp domain CCIE-SP
Changing VTP domain name from NULL to CCIE-SP
```

On SW4

```
SW4#sh vtp status | i VTP Domain
```

```
VTP Domain Name                 : CCIE-SP
```

After changing the domain, SW3 vtp undates are forwarded to SW4 through the dot1ad tunnel on both core switches. It is also verified to be working properly.

Task3

Reconfigure the Routers to allow connectivity between R1 and R2 using VLAN 88 on SW1 and SW2. Vlan 20 should not be configured on SW1 or SW2. The port-types should remain a UNI Type. The L2protocol forward command is not permitted.

On SW1

```
SW1(config)#no vlan 20

SW1(config)#int f0/1
SW1(config-if)#no l2protocol forward

SW1(config-if)#no ethernet dot1ad
SW1(config-if)#no switchport mode trunk
SW1(config-if)#no switchport trunk allowed vlan 1,20
SW1(config-if)#port-type uni
```

The port is recionfigured to be a uni port just to prove how a dot1ad tunnel does indeed modify it back to an nni port.

```
SW1(config-if)#switchport mode access
SW1(config-if)#ethernet dot1ad uni s-port
SW1(config-if)#switchport access vlan 88
% Access VLAN does not exist. Creating vlan 88

SW1(config)#interface FastEthernet0/19
SW1(config-if)# switchport trunk allowed vlan 88
```

This configuration is similar to configuring QinQ with L2PT Tunneling since the vlan from the customer has been removed and only one vlan is used regardless of the customer vlans.

On SW2

```
SW2(config)#no vlan 20

SW2(config)#int f0/1
```

```
SW2(config-if)#no l2protocol forward
SW2(config-if)#no ethernet dot1ad
SW2(config-if)#no switchport mode trunk
SW2(config-if)#no switchport trunk allowed vlan 1,20
SW2(config-if)#port-type uni

SW2(config-if)#switchport mode access
SW2(config-if)#ethernet dot1ad uni s-port

SW2(config-if)#switchport access vlan 88
% Access VLAN does not exist. Creating vlan 88

SW2(config)#interface FastEthernet0/19
SW2(config-if)# switchport trunk allowed vlan 88
```

On SW3

```
SW3#sh cdp nei | b D
Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
SW4              Fas 0/1           133           S I      WS-C3550- Fas 0/1

SW3(config)#vlan 30
SW3(config-vlan)#name PBB
```

On SW4

```
SW4#show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/2, Fa0/3, Fa0/4, Fa0/5
                                                Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                Fa0/14, Fa0/15, Fa0/16, Fa0/17
                                                Fa0/18, Fa0/19, Fa0/20, Fa0/21
                                                Fa0/22, Fa0/23, Fa0/24, Gi0/2
20   VLAN0020                         active    Gi0/1
30   PBB                              active
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup
```

CDP updates and VTP updates work without forwarding the updates on the core interfaces.