

CCIE Foundation

5.0

www.MicronicsTraining.com

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

VOL-II

Table of Content:

Section Eight: BGP

Subject	Page
Lab 1 – Establishing Neighbor Adjacency	5
Lab 2 – Route reflectors, Originator-ID and Cluster-ID	16
Lab 3 – Conditional Advertisement & BGP Backdoor	35
Lab 4 – The Community Attribute	54
Lab 5 – The AS-Path Attribute	66
Lab 6 – The Weight Attribute	74
Lab 7 – The Multi Exist Discriminator (MED) Attribute	83
Lab 8 – Filtering Using Access-lists and Prefix-lists	99
Lab 9 – Regular Expressions	113
Lab 10 – BGP Confederation	133

Section Nine: IPv6

Subject	Page
Lab 1 – Configuring Basic IPv6	141
Lab 2 – Configuring EIGRPv6	155
Lab 3 – OSPFv3 and the LSAs	168

Section Ten: QoS

Subject	Page
Lab 1 – MLS QoS	193
Lab 2 – DSCP-Mutation	208
Lab 3 – DSCP-CoS	220
Lab 4 – CoS-DSCP	228
Lab 5 – IP-Prec-to-DSCP	235
Lab 6 – Match input-interface & Match NOT	241
Lab 7 – Match destination-Address MAC & Source-address MAC	249
Lab 8 – Match IP-Prec/DSCP vs. Match DSCP	256
Lab 9 – Match protocol HTTP URL, MIME, and Host	264
Lab 10 – Configuring Class-Based Policing	272
Lab 11 – Configuring Class-Based Shaping	283

Section Eleven:

IP Services and Network Optimization & Advanced Features

Subject	Page
Lab 1 – HSRP	288
Lab 2 – VRRP	321
Lab 3 – GLBP	355
Lab 4 – NTP	373
Lab 5 – EEM	382

CCIE Foundation

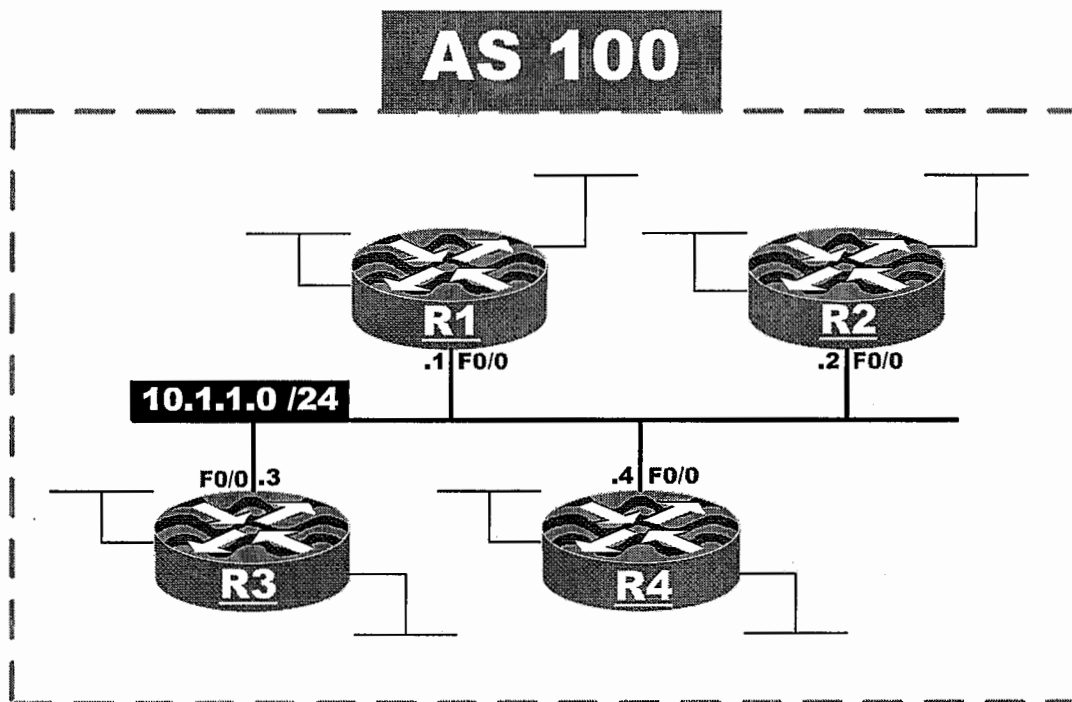
5.0

www.MicronicsTraining.com

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

BGP

Lab 1 – Establishing Neighbor Adjacency



Task 1

Configure the Routers and Switches based on the above topology. DO NOT configure BGP.

On R1:

```
R1(config)#Int Lo0
R1(config-if)#ip addr 1.1.1.1 255.0.0.0

R1(config)#int Lo1
R1(config-if)#IP addr 192.168.1.1 255.255.255.0

R1(config)#Int F0/0
R1(config-if)#IP addr 10.1.1.1 255.255.255.0
R1(config-if)#No shut
```

On R2:

```
R2(config)#Int Lo0
R2(config-if)#IP addr 2.2.2.2 255.0.0.0

R2(config)#Int Lo1
R2(config-if)#IP addr 192.168.2.2 255.255.255.0

R2(config)#Int f0/0
R2(config-if)#ip addr 10.1.1.2 255.255.255.0
R2(config-if)#No shut
```

On R3:

```
R3(config)#Int lo0
R3(config-if)#ip addr 3.3.3.3 255.0.0.0

R3(config)#int lo1
R3(config-if)#ip addr 192.168.3.3 255.255.255.0

R3(config)#Int f0/0
R3(config-if)#ip addr 10.1.1.3 255.255.255.0
R3(config-if)#No shut
```

On R4:

```
R4(config)#Int lo0
R4(config-if)#ip addr 4.4.4.4 255.0.0.0

R4(config)#Int lo1
R4(config-if)#ip addr 192.168.4.4 255.255.255.0

R4(config)#Int f0/0
R4(config-if)#ip addr 10.1.1.4 255.255.255.0
R4(config-if)#No shut
```

To verify and test the configuration:

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

```
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 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms

```
R1#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 = 1/1/1 ms

Task 2

Configure routers R1 – R4 in AS 100, these routers should create an IBGP peer sessions between them in a full mesh manner, ensure that these routers advertise their loopback 0 interface in this AS.

On R1:

```
R1(config)#router bgp 100
R1(config-router)#neighbor 10.1.1.2 remote-as 100
R1(config-router)#neighbor 10.1.1.3 remote-as 100
R1(config-router)#neighbor 10.1.1.4 remote-as 100
R1(config-router)#No syn
R1(config-router)#network 1.0.0.0
```

On R2:

```
R2(config)#router bgp 100
R2(config-router)#neighbor 10.1.1.1 remote-as 100
R2(config-router)#neighbor 10.1.1.3 remote-as 100
R2(config-router)#neighbor 10.1.1.4 remote-as 100
R2(config-router)#No syn
R2(config-router)#network 2.0.0.0
```

You should see the following console message stating that an adjacency is established with neighbor 10.1.1.1:

%BGP-5-ADJCHANGE: neighbor 10.1.1.1 Up

On R3:

```
R3(config)#router bgp 100
R3(config-router)#neighbor 10.1.1.1 remote-as 100
R3(config-router)#neighbor 10.1.1.2 remote-as 100
R3(config-router)#neighbor 10.1.1.4 remote-as 100
R3(config-router)#No syn
R3(config-router)#network 3.0.0.0
```

You should see the following console messages:

```
%BGP-5-ADJCHANGE: neighbor 10.1.1.1 Up
%BGP-5-ADJCHANGE: neighbor 10.1.1.2 Up
```

On R4:

```
R1(config)#router bgp 100
R4(config-router)#neighbor 10.1.1.1 remote-as 100
R4(config-router)#neighbor 10.1.1.2 remote-as 100
R4(config-router)#neighbor 10.1.1.3 remote-as 100
R4(config-router)#No syn
R4(config-router)#network 4.0.0.0
```

You should see the following console messages:

```
%BGP-5-ADJCHANGE: neighbor 10.1.1.1 Up
%BGP-5-ADJCHANGE: neighbor 10.1.1.3 Up
%BGP-5-ADJCHANGE: neighbor 10.1.1.2 Up
```

To verify the configuration:

On R1:

R1#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*>i2.0.0.0	10.1.1.2	0	100	0	i
*>i3.0.0.0	10.1.1.3	0	100	0	i
*>i4.0.0.0	10.1.1.4	0	100	0	i

The following explains the output of the above show command:

- The "*" means that the entry in the table is valid.
- The ">" means that this is the best entry for a given prefix.
- The "i" means that the entry was learned via an IBGP, this is the "i" to the right of the greater than sign (>) in the network column. The letter "i" under the path column, specifies the origin of the route.
- The "Network" column identifies the prefix entry for the network/s, if the mask is omitted, the default mask is assumed.
- The "Next Hop" column identifies the next hop's IP address to reach the specified network address, if its set to "0.0.0.0", it identifies a prefix that the local router has advertised.
- The "Metric" column identifies the inter-as metric, or the "MED" attribute which is set to zero by default.
- The "LocPrf" column identifies the "local preference" attribute; this is used in the route selection process carried within the local AS ONLY. With the local-pref attribute, the higher value has more preference. The prefixes that are received from a peer AS are tagged with a local-pref value of 100; this is the default behavior and this value can be changed to influence the best path selection process. The changed value is only advertised to IBGP peers. When the local router advertises a prefix, no local-pref value is seen in the output of the "Show ip bgp" command. The default value of 100 can be changed by the "BGP default local-preference" command.
- The "Weight" column identifies the weight attribute; the prefixes that are received via a neighbor (IBGP or EBGP) will have a weight attribute of 0, but the prefixes that are originated by the local router will have a weight value of 32768. This attribute overrides any other attribute for performing best path determination.
- The "Path" column identifies the path attribute. If the prefixes were originated or learned via an IBGP neighbor, the path column will have the letter "i" without any ASN. If the prefix was learned through another AS, then this column will have the AS number/s followed by the letter "i", the ASNs indicate which AS advertised the prefix. The maximum number of ASes that a prefix can traverse through is 255.

Task 3

Reconfigure the routers based on the following policy:

R2, R3 and R4 should be configured in AS 200, 300 and 400 respectively. Configure a full mesh peer session between these routers.

On All Routers:

```
Rx(config)#No Router bgp 100
```

On R1:

```
R1(config)#router bgp 100
R1(config-router)#neighbor 10.1.1.2 remote-as 200
R1(config-router)#neighbor 10.1.1.3 remote-as 300
R1(config-router)#neighbor 10.1.1.4 remote-as 400
R1(config-router)#No auto
R1(config-router)#network 1.0.0.0
```

On R2:

```
R2(config)#router bgp 200
R2(config-router)#neighbor 10.1.1.1 remote-as 100

R2(config-router)#neighbor 10.1.1.3 remote-as 300
R2(config-router)#neighbor 10.1.1.4 remote-as 400
R2(config-router)#No auto
R2(config-router)#network 2.0.0.0
```

On R3:

```
R3(config)#router bgp 300
R3(config-router)#neighbor 10.1.1.1 remote-as 100
R3(config-router)#neighbor 10.1.1.2 remote-as 200
R3(config-router)#neighbor 10.1.1.4 remote-as 400
R3(config-router)#No auto
R3(config-router)#network 3.0.0.0
```

On R4:

```
R1(config)#router bgp 400
R4(config-router)#neighbor 10.1.1.1 remote-as 100
R4(config-router)#neighbor 10.1.1.2 remote-as 200
R4(config-router)#neighbor 10.1.1.3 remote-as 300
R4(config-router)#No auto
R4(config-router)#network 4.0.0.0
```

To verify the configuration:

On R1:

```
R1#Show ip bgp | b Net
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
* 2.0.0.0	10.1.1.2			0 400	200 i
* 2.0.0.0	10.1.1.2			0 300	200 i
*> 2.0.0.0	10.1.1.2	0		0	200 i
* 3.0.0.0	10.1.1.3			0 400	300 i
* 3.0.0.0	10.1.1.3			0 200	300 i
*> 3.0.0.0	10.1.1.3	0		0	300 i
* 4.0.0.0	10.1.1.4			0 200	400 i
* 4.0.0.0	10.1.1.4			0 300	400 i
*> 4.0.0.0	10.1.1.4	0		0	400 i

Note: The local-preference attribute is not assigned on any of the prefixes; this is because the prefixes are advertised by an EBGp peer. The best selection in the above output is based on the shortest ASN.

The MED value (metric column) is zero for some of the prefixes, and it is not set on others, this is because when the prefix is advertised by the originating AS, the metric is set to "0", but when the same prefix is advertised by another AS, the MED value is removed.

NOTE: The next hop attribute does NOT change even though they are EBGp neighbors, this is because of the Multi-access broadcast network type (Ethernet) where each router can see the other router/s through the same interface.

Task 4

Reconfigure all the routers in AS 100; use the following policy for their IBGP peer sessions:

- Authentication must be enabled between the peers using "cisco" as the password.
- The peer session must be established based on the loopback 0's IP address.
- These routers should **only** advertise their loopback 1 in BGP.
- Provide NLRI to loopback0 interface using RIPv2.
- The peer session between the routers should only be established if they are running BGP version 4.
- Use peer-groups to accomplish this task.

Cisco's implementation of BGP in IOS 12.0(5)T or earlier releases supports BGP versions 2, 3, and 4, with dynamic negotiation down to Version 2. However, in IOS version 12.0(6)T or later, Cisco routers only support version 4 and they do not support dynamic negotiation down to Version 2. The reason you may see the "neighbor version" command configured on some Cisco routers is because may be the router is connecting and establishing a peer session with a Non-Cisco router, or the administrator is not aware of this fact.

On R1:

```
R1(config)#No Router bgp 100

R1(config)#Router bgp 100
R1(config-router)#No au
R1(config-router)#Net 192.168.1.0
R1(config-router)#Neighbor TST peer-group
R1(config-router)#Neighbor TST remote-as 100

R1(config-router)#Neighbor TST update-source lo0
R1(config-router)#Neighbor TST version 4
R1(config-router)#Neighbor TST password cisco

R1(config-router)#Neighbor 2.2.2.2 peer-group TST
R1(config-router)#Neighbor 3.3.3.3 peer-group TST
R1(config-router)#Neighbor 4.4.4.4 peer-group TST

R1(config-router)#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 R2:

```
R2(config)#No Router bgp 200

R2(config)#Router bgp 100
R2(config-router)#No au
R2(config-router)#Netw 192.168.2.0
R2(config-router)#Neighbor TST peer-group
R2(config-router)#Neighbor TST remote-as 100
R2(config-router)#Neighbor TST update-source lo0
R2(config-router)#Neighbor TST version 4
R2(config-router)#Neighbor TST password cisco

R2(config-router)#Neighbor 1.1.1.1 peer-group TST
```



```
R2(config-router)#Neighbor 3.3.3.3 peer-group TST
R2(config-router)#Neighbor 4.4.4.4 peer-group TST

R2(config-router)#Router rip
R2(config-router)#No au
R2(config-router)#Ver 2
R2(config-router)#Netw 10.0.0.0
R2(config-router)#Netw 2.0.0.0
```

On R3:

```
R3(config)#No Router bgp 300

R3(config)#Router bgp 100
R3(config-router)#No au
R3(config-router)#Network 192.168.3.0
R3(config-router)#Neighbor TST peer-group

R3(config-router)#Neighbor TST remote-as 100
R3(config-router)#Neighbor TST update-source lo0
R3(config-router)#Neighbor TST version 4
R3(config-router)#Neighbor TST password cisco

R3(config-router)#Neighbor 1.1.1.1 peer-group TST
R3(config-router)#Neighbor 2.2.2.2 peer-group TST
R3(config-router)#Neighbor 4.4.4.4 peer-group TST

R3(config-router)#Router rip
R3(config-router)#No au
R3(config-router)#Ver 2
R3(config-router)#Netw 10.0.0.0
R3(config-router)#Netw 3.0.0.0
```

On R4:

```
R4(config)#No Router bgp 400

R4(config)#Router bgp 100
R4(config-router)#No au
R4(config-router)#Network 192.168.4.0
R4(config-router)#Neighbor TST peer-group
R4(config-router)#Neighbor TST remote-as 100
R4(config-router)#Neighbor TST update-source lo0
R4(config-router)#Neighbor TST version 4
R4(config-router)#Neighbor TST password cisco
```

```
R4(config-router)#Neighbor 1.1.1.1 peer-group TST
R4(config-router)#Neighbor 2.2.2.2 peer-group TST
R4(config-router)#Neighbor 3.3.3.3 peer-group TST
```

```
R4(config-router)#Router rip
R4(config-router)#No au
R4(config-router)#Ver 2
R4(config-router)#Netw 10.0.0.0
R4(config-router)#Netw 4.0.0.0
```

To verify the configuration:

On R1:

```
R1#Show ip bgp | b Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.168.1.0	0.0.0.0	0		32768	i
*>i192.168.2.0	2.2.2.2	0	100	0	i
*>i192.168.3.0	3.3.3.3	0	100	0	i
*>i192.168.4.0	4.4.4.4	0	100	0	i

To verify the configuration:

On R1:

```
R1#Show ip bgp peer-group TST
```

```
BGP peer-group is TST, remote AS 100
  BGP version 4
  Default minimum time between advertisement runs is 0 seconds
  For address family: IPv4 Unicast
  BGP neighbor is TST, peer-group internal, members:
  2.2.2.2 3.3.3.3 4.4.4.4
  Index 0, Offset 0, Mask 0x0
  Update messages formatted 0, replicated 0
  Number of NLRI's in the update sent: max 0, min 0
```

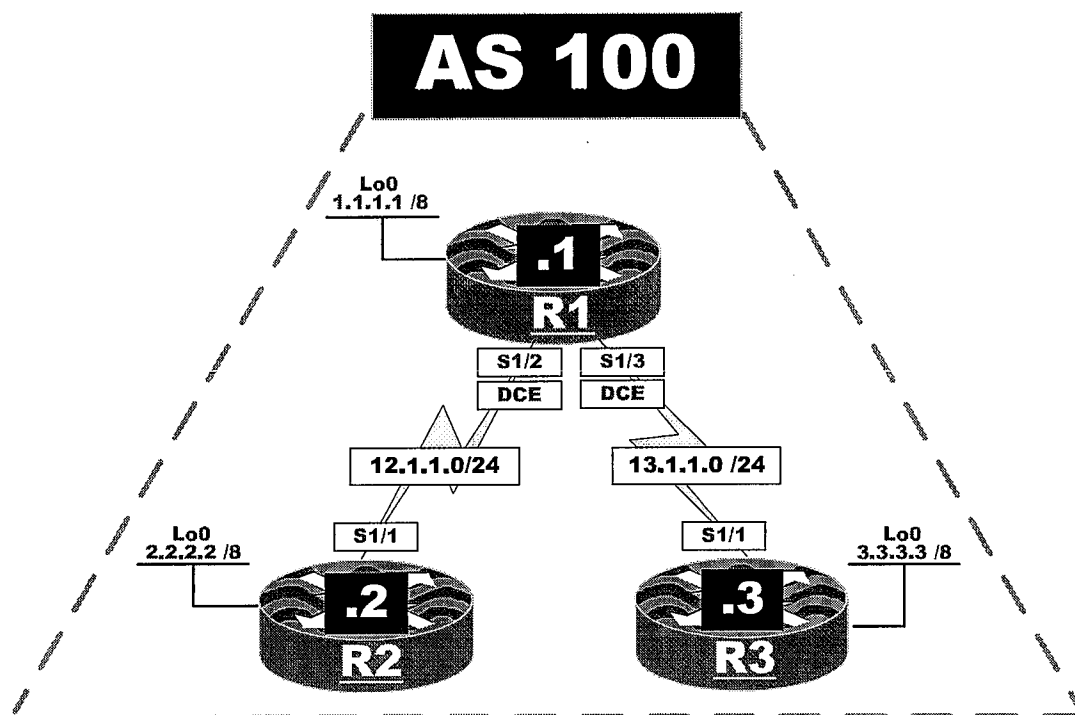
Note: The output of the “Show ip bgp peer-group TST” reveals the ip address of the members of the peer-group. One of the major benefits of a peer-group is that it can reduce the administrative overhead by cutting down redundant configuration on the routers.

Task 5

Erase the startup configuration and reload the routers before proceeding to the next lab.

Lab 2

Route Reflectors



Task 1

Configure the Routers and the Switches based on the above topology. DO NOT configure BGP.

On R1:

```
R1(config)#int s1/2
R1(config-if)#clock rate 64000
R1(config-if)#ip addr 12.1.1.1 255.255.255.0
R1(config-if)#No shut

R1(config)#int s1/3
R1(config-if)#clock rate 64000
R1(config-if)#ip addr 13.1.1.1 255.255.255.0
R1(config-if)#No shut

R1(config)#int lo0
```

```
R1(config-if)#ip addr 1.1.1.1 255.0.0.0
```

On R2:

```
R2(config)#int s1/1  
R2(config-if)#ip addr 12.1.1.2 255.255.255.0  
R2(config-if)#No shut
```

```
R2(config)#int lo0  
R2(config-if)#ip addr 2.2.2.2 255.0.0.0
```

On R3:

```
R3(config)#int s1/1  
R3(config-if)#ip addr 13.1.1.3 255.255.255.0  
R3(config-if)#No shut
```

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

To verify and test the configuration:

On R1:

```
R1#Ping 12.1.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.2, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms

```
R1#Ping 13.1.1.3
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 13.1.1.3, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms

Task 2

Configure BGP AS 100 on all routers and ensure that the routers can successfully establish an IBGP peer session with each other. These routers should only advertise their Loopback0 interface in BGP. Disable auto summary on all routers.

On R1:

```
R1(config)#Router bgp 100
R1(config-router)#Netw 1.0.0.0
R1(config-router)#Neighbor 12.1.1.2 remote-as 100
R1(config-router)#Neighbor 13.1.1.3 remote-as 100
R1(config-router)#No Auto
```

On R2:

```
R2(config)#Router bgp 100
R2(config-router)#Netw 2.0.0.0
R2(config-router)#Neighbor 12.1.1.1 remote-as 100
R2(config-router)#Neighbor 13.1.1.3 remote-as 100
R2(config-router)#No Auto
```

You should see the following console message:

```
%BGP-5-ADJCHANGE: neighbor 12.1.1.1 Up
```

On R3:

```
R3(config)#Router bgp 100
R3(config-router)#Netw 3.0.0.0
R3(config-router)#Neighbor 13.1.1.1 remote-as 100
R3(config-router)#Neighbor 12.1.1.2 remote-as 100
R3(config-router)#No Auto
```

You should see the following console message:

```
%BGP-5-ADJCHANGE: neighbor 13.1.1.1 Up
```

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*>i2.0.0.0	12.1.1.2	0	100	0	i
*>i3.0.0.0	13.1.1.3	0	100	0	i

R1#Show ip route bgp | Inc B

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

B 2.0.0.0/8 [200/0] via 12.1.1.2, 00:07:34
 B 3.0.0.0/8 [200/0] via 13.1.1.3, 00:05:27

On R2:

R2#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	12.1.1.1	0	100	0	i
*> 2.0.0.0	0.0.0.0	0		32768	i

R2#Show ip route bgp | Inc B

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

B 1.0.0.0/8 [200/0] via 12.1.1.1, 00:08:56

Network 3.0.0.0 is NOT in the BGP table of R2. Let's check their neighbor adjacency:

R2#Show ip bgp summary | B Neigh

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
12.1.1.1	4	100	16	16	4	0	0	00:10:33	1
13.1.1.3	4	100	0	0	1	0	0	never	Idle

The neighbor is in idle state. Do we have reachability to this neighbor? Let's verify using ping:

R2#Ping 13.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 13.1.1.3, timeout is 2 seconds:

.....

Success rate is 0 percent (0/5)

Let's check the routing table:

R2#Show ip route | B Gate

Gateway of last resort is not set

```

B      1.0.0.0/8 [200/0] via 12.1.1.1, 00:17:47
      2.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C      2.0.0.0/8 is directly connected, Loopback0
L      2.2.2.2/32 is directly connected, Loopback0
      12.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C      12.1.1.0/24 is directly connected, Serial1/1
L      12.1.1.2/32 is directly connected, Serial1/1

```

We can NOT see Network 13.1.1.0 /24 in the routing table, this is one reason that an IGP should be configured within an AS, this IGP should provide reachability to the links within the AS. Let's run OSPF on all the links within this topology:

On R1:

```

R1(config)#Router ospf 1
R1(config-router)#Router-id 0.0.0.1
R1(config-router)#Network 12.1.1.1 0.0.0.0 area 0
R1(config-router)#Network 13.1.1.1 0.0.0.0 area 0

```

On R2:

```

R2(config)#Router ospf 1
R2(config-router)#Router-id 0.0.0.2
R2(config-router)#Network 12.1.1.2 0.0.0.0 area 0

```

On R3:

```

R3(config)#Router ospf 1
R3(config-router)#Router-id 0.0.0.3
R3(config-router)#Network 13.1.1.3 0.0.0.0 area 0

```

```
R3#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	13.1.1.1	0	100	0	i
*>i2.0.0.0	12.1.1.2	0	100	0	i
*> 3.0.0.0	0.0.0.0	0		32768	i

```
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 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

Task 3

You received an e-mail from the management stating that within the next 12 months 20 additional routers will be added to this AS. In order to minimize the number of peer sessions within this AS, you decided to implement route reflectors. Configure R1 as a route reflector for this AS.

On R1:

```
R1(config)#Router bgp 100
R1(config-router)#Neighbor 12.1.1.2 route-reflector-client
R1(config-router)#Neighbor 13.1.1.3 route-reflector-client
```

On R2:

```
R2(config)#Router bgp 100
R2(config-router)#No Neighbor 13.1.1.3 remote-as 100
```

On R3:

```
R3(config)#Router bgp 100
R3(config-router)#No Neighbor 12.1.1.2 remote-as 100
```

To verify the configuration:

On R3:

```
R3#Show ip bgp | b Net
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	13.1.1.1	0	100	0	i
*>i2.0.0.0	12.1.1.2	0	100	0	i
*> 3.0.0.0	0.0.0.0	0		32768	i

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

In order for all IBGP speakers in an AS to exchange routes with one another, the IBGP speakers must be fully meshed (Every router must establish a peer session to every other router) this is because of a rule

called IBGP split-horizon, which states that when a BGP speaker receives an update from an IBGP peer, it will NEVER advertise that update to another IBGP peer. Route-reflectors can be configured to reduce the number of peer sessions that must be established between the routers within a given AS. If a route-reflector is used, all IBGP speakers need not be fully meshed. In this model, the router that is configured to be the route-reflector must have a peer session established to every client, the clients must establish a peer session with the route reflector ONLY. The route reflector will reflect routes learned from one client to the other client/s.

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*>i2.0.0.0	12.1.1.2	0	100	0	i
*>i3.0.0.0	13.1.1.3	0	100	0	i

The output of the following show commands reveal the fact that prefixes 2.0.0.0 and 3.0.0.0/8 were received from a route-Reflector Client:

```
R1#Show ip bgp 2.0.0.0/8
```

```
BGP routing table entry for 2.0.0.0/8, version 6
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
    2
  Local, (Received from a RR-client)
    10.1.12.2 from 10.1.12.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

```
R1#Show ip bgp 3.0.0.0/8
```

```
BGP routing table entry for 3.0.0.0/8, version 9
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    2
  Local, (Received from a RR-client)
    13.1.1.3 from 13.1.1.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

On R2:

```
R2#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	12.1.1.1	0	100	0	i
*> 2.0.0.0	0.0.0.0	0		32768	i
*>i3.0.0.0	13.1.1.3	0	100	0	i

```
R2#Ping 3.3.3.3
```

Type escape sequence to abort.

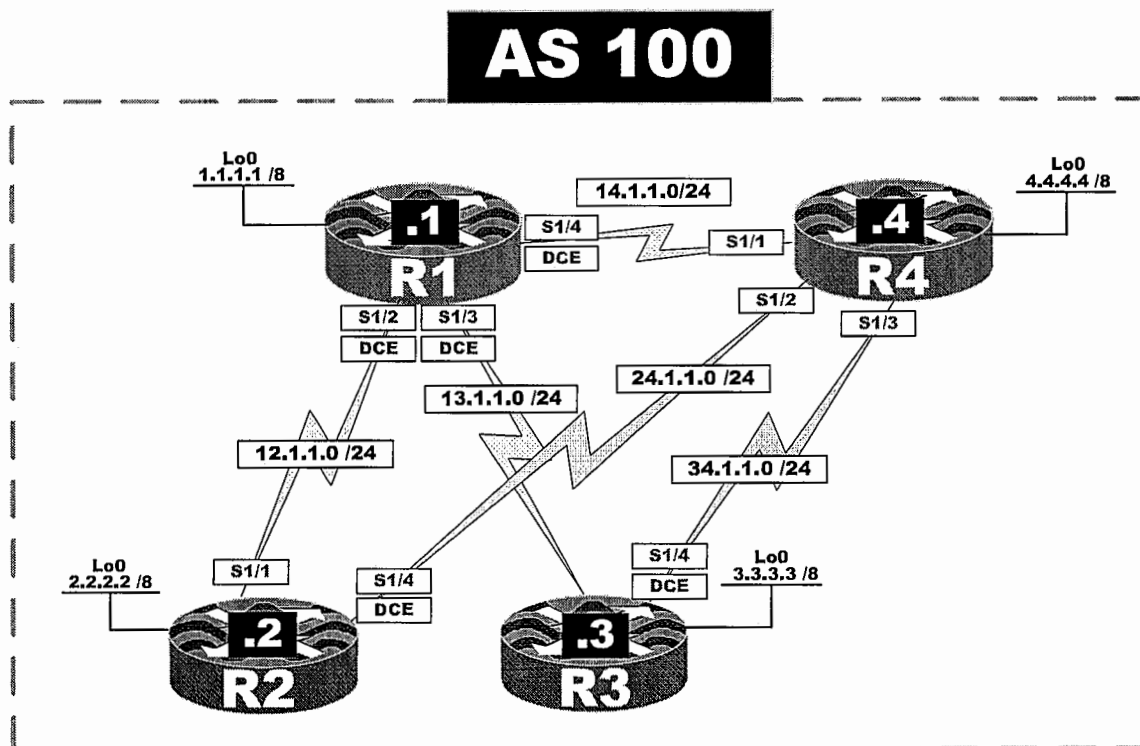
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:

!!!!!

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

Task 4

After implementing the route reflector, you realized that if the route reflector is down the entire network is dysfunctional; therefore, you decided to add R4 as the second route reflector for redundancy. Ensure that these routers can reach the advertised networks and the redundancy is operational. R4 should advertise its Lo0 interface in BGP.



DO NOT erase the existing configuration, add the following configuration to the existing configuration:

To copy and paste the initial configurations, go to “<http://www.micronics.nl>” → “Foundation” → “BGP” → “Lab-2-Task-4”.

R4 is the secondary route-reflector. R4 should be configured based on the following:

- **R4 should have a peer session with R1 – the route-reflectors should have full mesh peer sessions between them.**
- **R4 must have a peer session with R2 and R3.**
- **R4 must configure R2 and R3 as route-reflector clients.**

On R4:

```
R4 (config) #Int S1/1
R4 (config-if) #IP address 14.1.1.4 255.255.255.0
R4 (config-if) #No shut

R4 (config) #Int S1/2
R4 (config-if) #IP address 24.1.1.4 255.255.255.0
R4 (config-if) #No shut

R4 (config) #Int S1/3
R4 (config-if) #IP address 34.1.1.4 255.255.255.0
R4 (config-if) #No shut

R4 (config) #Int Lo0
R4 (config-if) #IP address 4.4.4.4 255.0.0.0

R4 (config) #Router bgp 100
R4 (config) #Network 4.0.0.0
R4 (config-router) #Neighbor 14.1.1.1 remote-as 100
R4 (config-router) #Neighbor 24.1.1.2 remote-as 100
R4 (config-router) #Neighbor 34.1.1.3 remote-as 100
R4 (config-router) #Neighbor 24.1.1.2 route-reflector client
R4 (config-router) #Neighbor 34.1.1.3 route-reflector client

R4 (config) #Router ospf 1
R4 (config-router) #Router-id 0.0.0.4
R4 (config-router) #Network 14.1.1.4 0.0.0.0 area 0
R4 (config-router) #Network 24.1.1.4 0.0.0.0 area 0
R4 (config-router) #Network 34.1.1.4 0.0.0.0 area 0
```

On R2:

```
R2(config)#Int S1/4
R2(config-if)#IP addr 24.1.1.2 255.255.255.0
R2(config-if)#Clock rate 64000
R2(config-if)#No shut

R2(config)#Router bgp 100
R2(config-router)#Neighbor 24.1.1.4 remote-as 100

R2(config)#Router ospf 1
R2(config-router)#Network 24.1.1.2 0.0.0.0 area 0
```

On R3:

```
R3(config)#Int S1/4
R3(config-if)#IP addre 34.1.1.3 255.255.255.0
R3(config-if)#Clock rate 64000
R3(config-if)#No shut

R3(config)#Router bgp 100
R3(config-router)#Neighbor 34.1.1.4 remote-as 100

R3(config)#Router ospf 1
R3(config-router)#Network 34.1.1.3 0.0.0.0 area 0
```

On R1:

```
R1(config)#Int S1/4
R1(config-if)#IP address 14.1.1.1 255.255.255.0
R1(config-if)#Clock rate 64000
R1(config-if)#No shut

R1(config)#Router bgp 100
R1(config-router)#Neighbor 14.1.1.4 remote-as 100

R1(config)#Router ospf 1
R1(config-router)#Network 14.1.1.1 0.0.0.0 area 0
```

Having a single RR can introduce a single point of failure, it's best to have multiple RRs incase the main RR fails, this redundancy is critical when there are many RR clients.

To verify the configuration:

On R1:

R1#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
* i2.0.0.0	24.1.1.2	0	100	0	i
*>i	12.1.1.2	0	100	0	i
* i3.0.0.0	34.1.1.3	0	100	0	i
*>i	13.1.1.3	0	100	0	i
*>i4.0.0.0	14.1.1.4	0	100	0	i

On R2:

R2#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
* i1.0.0.0	14.1.1.1	0	100	0	i
*>i	12.1.1.1	0	100	0	i
*> 2.0.0.0	0.0.0.0	0		32768	i
* i3.0.0.0	34.1.1.3	0	100	0	i
*>i	13.1.1.3	0	100	0	i
* i4.0.0.0	14.1.1.4	0	100	0	i
*>i	24.1.1.4	0	100	0	i

On R3:

R3#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
* i1.0.0.0	14.1.1.1	0	100	0	i
*>i	13.1.1.1	0	100	0	i
* i2.0.0.0	24.1.1.2	0	100	0	i
*>i	12.1.1.2	0	100	0	i
*> 3.0.0.0	0.0.0.0	0		32768	i
* i4.0.0.0	14.1.1.4	0	100	0	i
*>i	34.1.1.4	0	100	0	i

On R4:

R4#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	14.1.1.1	0	100	0	i

```

* i2.0.0.0          12.1.1.2          0      100      0 i
*>i                24.1.1.2          0      100      0 i
* i3.0.0.0          13.1.1.3          0      100      0 i
*>i                34.1.1.3          0      100      0 i
*> 4.0.0.0          0.0.0.0          0              32768 i

```

To test the configuration:

On R1:

```

R1(config)#int s1/2
R1(config-if)#shut

```

On R2:

You should wait to see the following console message before entering the following show command:

```
%LINK-3-UPDOWN: Interface Serial1/1, changed state to down
```

```
%OSPF-5-ADJCHG: Process 1, Nbr 0.0.0.1 on Serial1/1 from FULL to DOWN,
Neighbor Down: Interface down or detached
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state
to down
```

```
%BGP-5-ADJCHANGE: neighbor 12.1.1.1 Down BGP Notification sent
```

```
%BGP-3-NOTIFICATION: sent to neighbor 12.1.1.1 4/0 (hold time expired) 0
bytes
```

```
%BGP_SESSION-5-ADJCHANGE: neighbor 12.1.1.1 IPv4 Unicast topology base
removed from session BGP Notification sent * 28 msec
```

```
R2#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	14.1.1.1	0	100	0	i
*> 2.0.0.0	0.0.0.0	0		32768	i
*>i3.0.0.0	34.1.1.3	0	100	0	i
*>i4.0.0.0	24.1.1.4	0	100	0	i

```
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:

```
!!!!
```

2 34.1.1.3 28 msec * 28 msec

Task 6

Configure BGP on R1 – R6 based on the following policy:

- R1 should be configured as the route reflector for routers R2 and R3, whereas R4 should be configured to be the route-reflector for routers R5 and R6.
- R1 and R4 should be configured to have an IBGP peer session between them; these routers (R1-R6) should be configured in BGP AS 100.
- NLRI for the links should be provided through OSPF.

On R1:

```
R1(config)#Router bgp 100
R1(config-router)#No au
R1(config-router)#Network 1.0.0.0
R1(config-router)#Neighbor 14.1.1.4 remote-as 100
R1(config-router)#Neighbor 12.1.1.2 remote-as 100
R1(config-router)#Neighbor 13.1.1.3 remote-as 100
R1(config-router)#Neighbor 12.1.1.2 route-reflector-client
R1(config-router)#Neighbor 13.1.1.3 route-reflector-client

R1(config)#Router ospf 1
R1(config-router)#Router-id 0.0.0.1
R1(config-router)#Network 12.1.1.1 0.0.0.0 area 0
R1(config-router)#Network 14.1.1.1 0.0.0.0 area 0
R1(config-router)#Network 13.1.1.1 0.0.0.0 area 0
```

On R2:

```
R2(config)#Router bgp 100
R2(config-router)#No au
R2(config-router)#Network 2.0.0.0
R2(config-router)#Neighbor 12.1.1.1 remote-as 100

R2(config)#Router ospf 1
R2(config-router)#Router-id 0.0.0.2
R2(config-router)#Network 12.1.1.2 0.0.0.0 area 0
```

On R3:

```
R3(config)#Router bgp 100
R3(config-router)#No au
R3(config-router)#Network 3.0.0.0
```

```
R3(config-router)#Neighbor 13.1.1.1 remote-as 100
```

```
R3(config)#Router ospf 1
```

```
R3(config-router)#Router-id 0.0.0.3
```

```
R3(config-router)#Network 13.1.1.3 0.0.0.0 area 0
```

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*>i2.0.0.0	12.1.1.2	0	100	0	i
*>i3.0.0.0	13.1.1.3	0	100	0	i

On R2:

```
R2#Show ip route ospf | Inc 0
```

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP

```
O 13.1.1.0 [110/1562] via 12.1.1.1, 00:03:29, Serial1/1
```

```
O 14.1.1.0 [110/1562] via 12.1.1.1, 00:03:29, Serial1/1
```

```
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 100 percent (5/5), round-trip min/avg/max = 56/57/60 ms

On R4:

```
R4(config)#Router bgp 100
```

```
R4(config-router)#No au
```

```
R4(config-router)#Network 4.0.0.0
```

```
R4(config-router)#Neighbor 14.1.1.1 remote-as 100
```

```
R4(config-router)#Neighbor 45.1.1.5 remote-as 100
```

```
R4(config-router)#Neighbor 46.1.1.6 remote-as 100
```

```
R4(config-router)#Neighbor 45.1.1.5 route-reflector-client
```

```
R4(config-router)#Neighbor 46.1.1.6 route-reflector-client
```

```
R4(config)#Router ospf 1
R4(config-router)#Router-id 0.0.0.4
R4(config-router)#Network 14.1.1.4 0.0.0.0 area 0
R4(config-router)#Network 45.1.1.4 0.0.0.0 area 0
R4(config-router)#Network 46.1.1.4 0.0.0.0 area 0
```

On R5:

```
R5(config)#Router bgp 100
R5(config-router)#No au
R5(config-router)#Network 5.0.0.0
R5(config-router)#Neighbor 45.1.1.4 remote-as 100

R5(config)#Router ospf 1
R5(config-router)#Router-id 0.0.0.5
R5(config-router)#Network 45.1.1.5 0.0.0.0 area 0
```

On R6:

```
R6(config)#Router bgp 100
R6(config-router)#No au
R6(config-router)#Network 6.0.0.0
R6(config-router)#Neighbor 46.1.1.4 remote-as 100

R6(config)#Router ospf 1
R6(config-router)#Router-id 0.0.0.6
R6(config-router)#Network 46.1.1.6 0.0.0.0 area 0
```

To verify the configuration:

On R6:

```
R6#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	14.1.1.1	0	100	0	i
*>i2.0.0.0	12.1.1.2	0	100	0	i
*>i3.0.0.0	13.1.1.3	0	100	0	i
*>i4.0.0.0	46.1.1.4	0	100	0	i
*>i5.0.0.0	45.1.1.5	0	100	0	i
*> 6.0.0.0	0.0.0.0	0		32768	i

On R3:

```
R3#Show ip route bgp | B Gate
```

Gateway of last resort is not set

```
B      1.0.0.0/8 [200/0] via 13.1.1.1, 00:21:49
B      2.0.0.0/8 [200/0] via 12.1.1.2, 00:20:49
B      4.0.0.0/8 [200/0] via 14.1.1.4, 00:17:19
B      5.0.0.0/8 [200/0] via 45.1.1.5, 00:13:43
B      6.0.0.0/8 [200/0] via 46.1.1.6, 00:01:47
```

R3#**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:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 84/86/88 ms

R3#**Traceroute 6.6.6.6 numeric**

Type escape sequence to abort.

Tracing the route to 6.6.6.6

VRF info: (vrf in name/id, vrf out name/id)

```
 1 13.1.1.1 16 msec 12 msec 12 msec
 2 14.1.1.4 32 msec 28 msec 28 msec
 3 46.1.1.6 44 msec * 40 msec
```

Note: AS 100 has two route reflectors, each route-reflector has its own clients, when a given RR receives an update from one of its clients, it advertises that prefix to the other RR/s, the other RR/s in turn advertise that prefix to their clients.

There are some additional optional non-transitive attributes that can be used when RRs are configured and they are: "originator-id", "cluster-id" and "cluster-list".

- **Originator-id:** This attribute is created by the RR; this is the router-id of the router that originated the prefix. The originator-id created to avoid routing loops, a RR will NOT advertise a route back to the originator of the prefix, and if the originator of a prefix receives an update with its own router-id, it will ignore that prefix.
- **Cluster and Cluster-id:** A RR/s and its clients are collectively known as a cluster, each cluster must be uniquely identified, and the "cluster-id" is the router-id of the RR unless specifically configured.
- **Cluster-list:** This attribute is analogous to AS-path attribute, and it keeps track of the cluster-ids in the same way that the AS-path attribute keeps track of the AS numbers. When the RR advertises a prefix to a non-client, it appends the cluster-id to that prefix's cluster-list, if a RR receives an update and sees its own cluster-id in the cluster-list, it will ignore that update.

To see the attributes:

On R1:

```
R1#Show ip bgp 6.0.0.0
```

BGP routing table entry for 6.0.0.0/8, version 7

Paths: (1 available, best #1, table Default-IP-Routing-Table)

Flag: 0x820

Advertised to update-groups:

2

Local

46.1.1.6 (metric 1) from 14.1.1.4 (4.4.4.4)

Origin IGP, metric 0, localpref 100, valid, internal, best

Originator: 6.6.6.6, Cluster list: 4.4.4.4

Note: Prefix 6.0.0.0 was originated by R6 with a router-id of 6.6.6.6 (The originator of the prefix), and it came from cluster-id of 4.4.4.4 (The cluster-list).

On R3:

```
R3#Show ip bgp 6.0.0.0
```

BGP routing table entry for 6.0.0.0/8, version 8

Paths: (1 available, best #1, table default)

Not advertised to any peer

Local

46.1.1.6 (metric 2343) from 13.1.1.1 (1.1.1.1)

Origin IGP, metric 0, localpref 100, valid, internal, best

Originator: 6.6.6.6, Cluster list: 1.1.1.1, 4.4.4.4

Note: This prefix has gone through cluster-ids of 4.4.4.4 first and then it traversed through cluster-id 1.1.1.1 before it was received by the local router.

The originator-id is the router-id of the router that originated that prefix, the output of the following "Show" command reveals the router-id of the router that originated the route.

```
R6#Sh ip bgp | Inc router ID
```

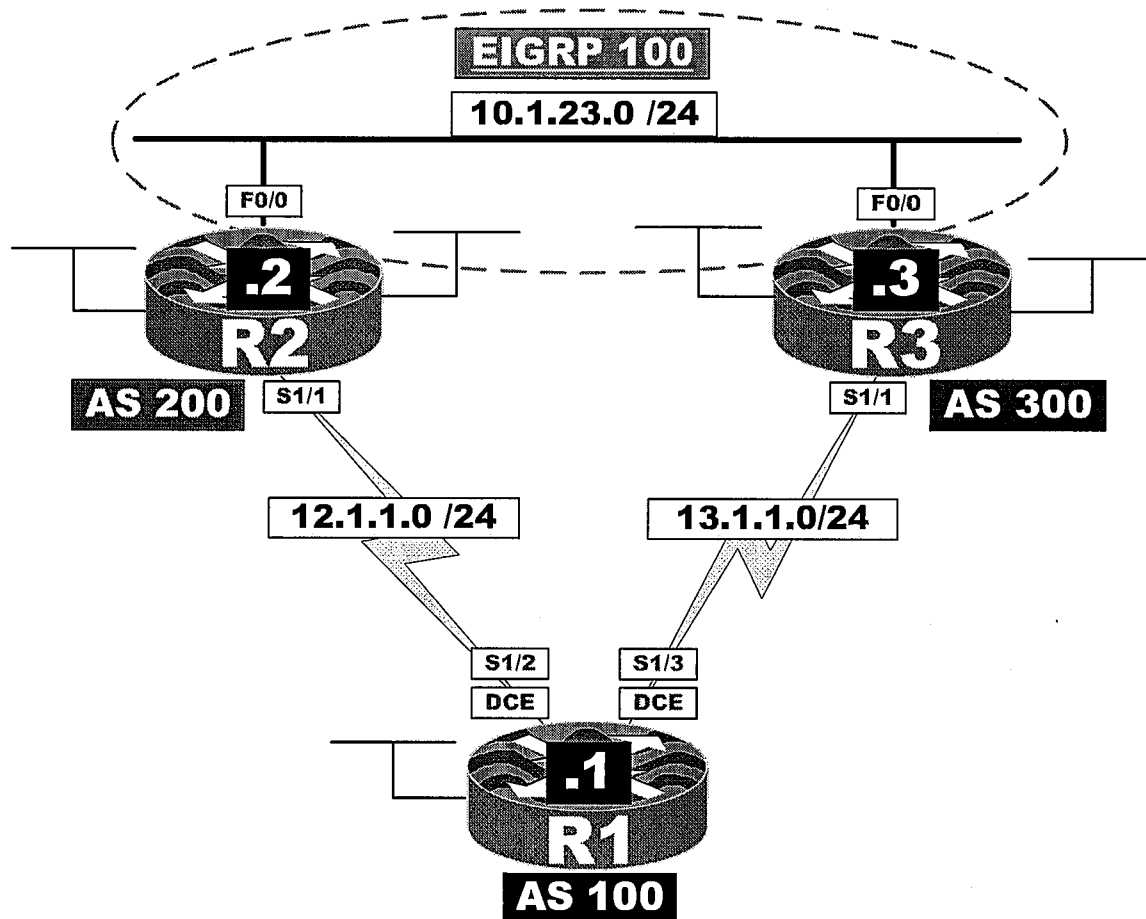
BGP table version is 8, **local router ID is 6.6.6.6**

Task 7

Erase the startup config and reload the routers before proceeding to the next lab.

Lab 3

Conditional Advertisement & BGP Backdoor



Task 1

Configure the Routers and the Switches according to the above diagram. DO NOT configure any routing protocol.

On R1:

```
R1(config)#int s1/2
```

```
R1(config-if)#clock rate 64000
R1(config-if)#ip addr 12.1.1.1 255.255.255.0
R1(config-if)#No shut
```

```
R1(config)#int s1/3
R1(config-if)#clock rate 64000
R1(config-if)#ip addr 13.1.1.1 255.255.255.0
R1(config-if)#No shut
```

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

On R2:

```
R2(config)#int s1/1
R2(config-if)#ip addr 12.1.1.2 255.255.255.0
R2(config-if)#No shut
```

```
R2(config)#int f0/0
R2(config-if)#ip addr 10.1.23.2 255.255.255.0
R2(config-if)#No shut
```

```
R2(config)#int lo0
R2(config-if)#ip addr 2.2.2.2 255.0.0.0
```

```
R2(config)#int lo1
R2(config-if)#ip addr 10.1.2.2 255.255.255.0
```

On R3:

```
R3(config)#int s1/1
R3(config-if)#ip addr 13.1.1.3 255.255.255.0
R3(config-if)#No shut
```

```
R3(config)#int f0/0
R3(config-if)#ip addr 10.1.23.3 255.255.255.0
R3(config-if)#No shut
```

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

```
R3(config)#int lo1
R3(config-if)#ip addr 10.1.3.3 255.255.255.0
```

To verify and test the configuration:

On R1:

R1#Ping 12.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

R1#Ping 13.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 13.1.1.3, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

Task 2

Configure R1 in AS 100 to establish an EBGp peer session with R2 and R3 in AS 200 and 300 respectively.

On R1:

R1(config)#Router bgp 100

R1(config-router)#No auto

R1(config-router)#Neighbor 12.1.1.2 remote-as 200

R1(config-router)#Neighbor 13.1.1.3 remote-as 300

On R2:

R2(config)#Router bgp 200

R2(config-router)#No au

R2(config-router)#Neighbor 12.1.1.1 remote-as 100

On R3:

R3(config)#Router bgp 300

R3(config-router)#No au

R3(config-router)#Neighbor 13.1.1.1 remote-as 100

To verify the configuration:

On R1:

R1#**Show ip bgp summary | B Neighbor**

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
10.1.12.2	4	200	3	3	1	0	0	00:00:39	0
10.1.13.3	4	300	3	3	1	0	0	00:00:32	0

Task 3

Configure R1, R2 and R3 to advertise their loopback0 interface in BGP.

On R1:

```
R1(config)#Router bgp 100  
R1(config-router)#Network 1.0.0.0
```

On R2:

```
R2(config)#Router bgp 200  
R2(config-router)#Network 2.0.0.0
```

On R3:

```
R3(config)#Router bgp 300  
R3(config-router)#Network 3.0.0.0
```

To verify the configuration:

On R3:

R3#**Show ip bgp | B Network**

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	13.1.1.1	0		0	100 i
*> 2.0.0.0	13.1.1.1			0	100 200 i
*> 3.0.0.0	0.0.0.0	0		32768	i

Task 4

Configure RIPv2 and Eigrp 100 on the routers as follows:

- Configure RIPv2 on networks 12.1.1.0 /24 and 13.1.1.0 /24; disable auto summarization.
- R2 and R3 should advertise their F0/0, and Loopback 1 interfaces in Eigrp AS 100. Disable auto summarization.

On R1:

```
R1(config)#Router rip
R1(config-router)#No au
R1(config-router)#Ver 2
R1(config-router)#Network 12.0.0.0
R1(config-router)#Network 13.0.0.0
```

On R2:

```
R2(config)#Router rip
R2(config-router)#No au
R2(config-router)#Ver 2
R2(config-router)#Network 12.0.0.0

R2(config)#Router eigrp 100
R2(config-router)#Network 10.1.23.2 0.0.0.0
R2(config-router)#Network 10.1.2.2 0.0.0.0
```

On R3:

```
R3(config)#Router rip
R3(config-router)#No au
R3(config-router)#Ver 2
R3(config-router)#Network 13.0.0.0

R3(config)#Router eigrp 100
R3(config-router)#Network 10.1.3.3 0.0.0.0
R3(config-router)#Network 10.1.23.3 0.0.0.0
```

To verify the configuration:

On R2:

```
R2#Show ip route eigrp | B Gate
Gateway of last resort is not set
```

```
10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D    10.1.3.0/24 [90/156160] via 10.1.23.3, 00:00:52, FastEthernet0/0
```

```
R2#Show ip route rip | B Gate
Gateway of last resort is not set
```

```
13.0.0.0/24 is subnetted, 1 subnets
R    13.1.1.0 [120/1] via 12.1.1.1, 00:00:15, Serial1/1
```

Task 5

Since network 10.1.23.0 is NOT advertised in BGP, if the link between R2 and R3 (The F0/0 interface) goes down, the Loopback1 network of these two routers won't have reachability to each other, even though there is a redundant link between these two routers through BGP, therefore, the administrator of R2 and R3 decided that Loopback 1 interfaces of R2 and R3 should be advertised in BGP for redundancy, configure these routers to accommodate this decision.

On R2:

```
R2(config)#Router bgp 200
R2(config-router)#Network 10.1.2.0 mask 255.255.255.0
```

On R3:

```
R3(config)#Router bgp 300
R3(config-router)#Network 10.1.3.0 mask 255.255.255.0
```

To verify the configuration:

On R2:

```
R2#Show ip route bgp | B Gate
Gateway of last resort is not set
```

```
B    1.0.0.0/8 [20/0] via 12.1.1.1, 00:16:27
B    3.0.0.0/8 [20/0] via 12.1.1.1, 00:15:57
10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
```

```
B      10.1.3.0/24 [20/0] via 12.1.1.1, 00:00:13
```

On R3:

```
R3#Show ip route bgp | B Gate
```

```
Gateway of last resort is not set
```

```
B      1.0.0.0/8 [20/0] via 13.1.1.1, 00:17:06
```

```
B      2.0.0.0/8 [20/0] via 13.1.1.1, 00:16:05
```

```
10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
```

```
B      10.1.2.0/24 [20/0] via 13.1.1.1, 00:01:22
```

Task 6

After implementing the previous task, the administrators realized that the traffic between networks 10.1.2.0 /24 and 10.1.3.0 /24 is taking a sub-optimal path and it is not using the direct path between routers R2 and R3.

Implement a BGP solution to fix this problem; you should **NOT** use the distance, PBR or any global config mode command to accomplish this task.

To see the suboptimal path:

On R3:

```
R3#Traceroute 10.1.2.2
```

```
Type escape sequence to abort.
```

```
Tracing the route to 10.1.2.2
```

```
VRF info: (vrf in name/id, vrf out name/id)
```

```
 1 13.1.1.1 16 msec 16 msec 12 msec
```

```
 2 12.1.1.2 32 msec * 28 msec
```

```
R3#Show ip route 10.1.2.2
```

```
Routing entry for 10.1.2.0/24
```

```
Known via "bgp 300", distance 20, metric 0
```

```
Tag 100, type external
```

```
Last update from 13.1.1.1 00:07:02 ago
```

```
Routing Descriptor Blocks:
```

```
* 13.1.1.1, from 13.1.1.1, 00:07:02 ago
```

```
Route metric is 0, traffic share count is 1
AS Hops 2
Route tag 100
MPLS label: none
```

NOTE: The BGP "Backdoor" option can help us to accomplish this task. The "Backdoor" keyword is added to the network command that is advertised to you, therefore, you should reference the network that is advertised to you and NOT the network that your local router is advertising:

On R2:

```
R2(config)#Router bgp 200
R2(config-router)#Network 10.1.3.0 mask 255.255.255.0 backdoor
```

To verify the configuration:

On R2:

```
R2#Show ip route 10.1.3.3
```

```
Routing entry for 10.1.3.0/24
  Known via "eigrp 100", distance 90, metric 156160, type internal
  Redistributing via eigrp 100
  Last update from 10.1.23.3 on FastEthernet0/0, 00:00:56 ago
  Routing Descriptor Blocks:
    * 10.1.23.3, from 10.1.23.3, 00:00:56 ago, via FastEthernet0/0
      Route metric is 156160, traffic share count is 1
      Total delay is 5100 microseconds, minimum bandwidth is 100000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

```
R2#Traceroute 10.1.3.3
```

```
Type escape sequence to abort.
Tracing the route to 10.1.3.3
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.23.3 4 msec * 0 msec
```

We can see that R2 uses its direct connection (F0/0 interface) to reach the Loopback 1 interface of R3. The "Backdoor" keyword increases the administrative distance through EBGP for the advertised network to 200 so the local route can use the IGP and NOT EBGP's advertisement. Let's test the redundancy:

On R2, let's shutdown its F0/0 interface and verify reachability:

On R2:

```
R2(config)#Int F0/0
```

```
R2(config-if)#Shut
```

```
R2#Show ip route 10.1.3.3
```

```
Routing entry for 10.1.3.0/24
```

```
Known via "bgp 200", distance 200, metric 0
```

```
Tag 100, type locally generated
```

```
Last update from 12.1.1.1 00:00:42 ago
```

```
Routing Descriptor Blocks:
```

```
* 12.1.1.1, from 12.1.1.1, 00:00:42 ago
```

```
Route metric is 0, traffic share count is 1
```

```
AS Hops 2
```

```
Route tag 100
```

```
MPLS label: none
```

```
R2#Traceroute 10.1.3.3
```

```
Type escape sequence to abort.
```

```
Tracing the route to 10.1.3.3
```

```
VRF info: (vrf in name/id, vrf out name/id)
```

```
1 12.1.1.1 16 msec 16 msec 12 msec
```

```
2 13.1.1.3 32 msec * 28 msec
```

Let's enable the F0/0 interface of R2 and configure the same on R3:

On R2:

```
R2(config)#Int F0/0
```

```
R2(config-if)#No shut
```

```
R2#Show ip route 10.1.3.3
```

```
Routing entry for 10.1.3.0/24
```

```
Known via "eigrp 100", distance 90, metric 156160, type internal
```

```
Redistributing via eigrp 100
```

```
Last update from 10.1.23.3 on FastEthernet0/0, 00:00:33 ago
```

```
Routing Descriptor Blocks:
```

```
* 10.1.23.3, from 10.1.23.3, 00:00:33 ago, via FastEthernet0/0
```

```
Route metric is 156160, traffic share count is 1
```

```
Total delay is 5100 microseconds, minimum bandwidth is 100000 Kbit
```

```
Reliability 255/255, minimum MTU 1500 bytes
```

```
Loading 1/255, Hops 1
```

On R3:

```
R3(config)#Router bgp 300
R3(config-router)#Network 10.1.2.0 mask 255.255.255.0 backdoor
```

To verify the configuration:

On R3:

```
R3#Sh ip rou eigrp | B Gate
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D       10.1.2.0/24 [90/156160] via 10.1.23.2, 00:00:20, FastEthernet0/0
```

On R2:

```
R2#Show ip route eigrp | B Gate
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
D       10.1.3.0/24 [90/156160] via 10.1.23.3, 00:07:07, FastEthernet0/0
```

NOTE: R2 and R3 were receiving routing information for networks 10.1.2.0 /24 and 10.1.3.0 /24 from two different sources, BGP and EIGRP.

R2 and R3 were using the routing information from BGP because it had a lower administrative distance (20 for EIGRP versus 90 for Eigrp).

The Network command with the "backdoor" option is a BGP solution to this problem; the BGP "backdoor" option assigns an administrative distance of 200 to networks 10.1.2.0 /24 and 10.1.3.0 /24, therefore, making the Eigrp more attractive.

Task 7

Remove the IP address from the F0/0 interfaces of R2 and R3 and ensure that the F0/0 interfaces of both routers are in administratively down state. You should also remove the Loopback1 interface from these two routers.

On R2 and R3:


```
Rx(config)#Default interface F0/0
```

```
Rx(config)#Interface F0/0
```

```
Rx(config-if)#Shutdown
```

```
Rx(config)#No int lol
```

Task 8

Configure R1 as follows:

- If network 2.0.0.0 /8 is up and it's advertised to R1, R1 should NOT advertise its network 1.0.0.0 /8 to R3.
- R1 should advertise network 1.0.0.0 /8 to R3 ONLY if network 2.0.0.0 /8 is down.

Before configuring this task you should verify the current BGP table of these routers:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0	200 i
*> 3.0.0.0	13.1.1.3	0		0	300 i

```
R2#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	12.1.1.1	0		0	100 i
*> 2.0.0.0	0.0.0.0	0		32768	i
*> 3.0.0.0	12.1.1.1			0	100 300 i

```
R3#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	13.1.1.1	0		0	100 i
*> 2.0.0.0	13.1.1.1			0	100 200 i
*> 3.0.0.0	0.0.0.0	0		32768	i

To implement conditional advertisement of selected prefixes, the following can be used:

- Advertise-map
- Non-exist-map
- Exist-map
- Inject-map

This situation calls for the use of the “advertise-map” and “non-exist-map”, basically using these two commands we are saying advertise network 1.0.0.0 ONLY if network 2.0.0.0 is down, if network 2.0.0.0 is NOT down, then don’t advertise network 1.0.0.0. To configure this task:

On R1:

Step #1 – Identify the prefixes using two access-list/prefix-list:

```
R1 (config) #Access-list 1 permit 1.0.0.0 0.255.255.255
R1 (config) #Access-list 2 permit 2.0.0.0 0.255.255.255
```

Step #2 – Configure two route-maps, one to reference access-list 1 and the second one to reference access-list 2. To prevent confusion you should select meaningful names for the route-maps:

```
R1 (config) #Route-map ADV permit 10
R1 (config-route-map) #match ip addr 1
R1 (config-route-map) #exit

R1 (config) #Route-map NotThere permit 10
R1 (config-route-map) #match ip addr 2
R1 (config-route-map) #exit
```

Final Step – the route-maps are referenced by the “Advertise-map” and “non-exist-map” options:

```
R1 (config) #Router bgp 100
R1 (config-router) #Neighbor 13.1.1.3 advertise-map ADV non-exist-map NotThere
```

The neighbor command has the following route-maps:

- The advertise-map – Specifies the name of the route-map that will be advertised if the condition of the non-exist-map is met.
- Non-exist-map – specifies the name of the route-map that will be compared to the advertise-map. If the condition is met and no match occurs, the route will be advertised. If a match occurs, then the condition is NOT met, and the route is withdrawn.

If network 2.0.0.0 is up, then network 1.0.0.0 should NOT be advertised to R3, since all the networks are up and advertised, R1 should withdraw its network (1.0.0.0 /8):

On R1:

NOTE: Network 2.0.0.0 is up so network 1.0.0.0 /8 should NOT be advertised to R3.

R1#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0	200 i
*> 3.0.0.0	13.1.1.3	0		0	300 i

The following show command reveals that R1 does NOT advertise its network (1.0.0.0 /8) to R3:

R1#Show ip bgp neighbors 13.1.1.3 advertised-routes | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2.0.0.0	12.1.1.2	0		0	200 i

To verify this configuration

On R3:

R3#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2.0.0.0	13.1.1.1			0	100 200 i
*> 3.0.0.0	0.0.0.0	0		32768	i

To test the condition:

On R2:

```
R2(config)#int lo0
R2(config-if)#Shut
```

The output of the following show command reveals that network 2.0.0.0 is DOWN; and R1 is advertising its network (1.0.0.0 /8) to R3. It may take few seconds for this policy to be implemented:

On R1:

R1#Show ip bgp neighbors 13.1.1.3 advertised-routes | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i

To see the test on R3:

On R3:

R3#**Show ip bgp | B Network**

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	13.1.1.1	0		0	100 i
*> 3.0.0.0	0.0.0.0	0		32768	i

Task 9

Remove the configuration commands entered in the previous task before you proceed to the next task. Ensure that the routers have the advertised networks in their BGP table.

On R1:

```
R1(config)#No access-list 1
R1(config)#No access-list 2

R1(config)#No route-map ADV

R1(config)#No route-map NotThere

R1(config)#Router bgp 100
R1(config-router)#No Neighbor 13.1.1.3 advertise-map ADV non-exist-map NotThere

R1#Clear ip bgp *
```

On R2:

```
R2(config)#int lo0
R2(config-if)#No shut
```

On R1:

R1#**Show ip bgp | B Network**

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i


```
*> 2.0.0.0          12.1.1.2          0          0 200 i
*> 3.0.0.0          13.1.1.3          0          0 300 i
```

On R2:

R2#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	12.1.1.1	0		0	100 i
*> 2.0.0.0	0.0.0.0	0		32768	i
*> 3.0.0.0	12.1.1.1			0	100 300 i

On R3:

R3#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	13.1.1.1	0		0	100 i
*> 2.0.0.0	13.1.1.1			0	100 200 i
*> 3.0.0.0	0.0.0.0	0		32768	i

Task 10

R1 should be configured based on the following policy:

1. If both networks (1.0.0.0 /8 and 2.0.0.0 /8) are up, then both networks should be advertised to R3.
2. If network 1.0.0.0 /8 is down, R1 should NOT advertise network 2.0.0.0 /8 to R3.
3. If network 2.0.0.0 /8 is down, then R1 should only advertise network 1.0.0.0 /8 to R3.

The logic in the following configuration says "ONLY advertise network 2.0.0.0/8 if network 1.0.0.0/8 is up, so if network 1.0.0.0/8 is NOT up, then DON'T advertise network 2.0.0.0/8.

On R1:

Step #1 - The following two access-lists identify the two networks (1.0.0.0 /8 and 2.0.0.0 /8):

```
R1(config)#Access-list 1 permit 1.0.0.0 0.255.255.255
```

```
R1 (config) #Access-list 2 permit 2.0.0.0 0.255.255.255
```

Step #2 - The following route-maps are configured to reference the two access-lists from the previous step:

```
R1 (config) #Route-map ADV permit 10
```

```
R1 (config-route-map) #match ip addr 2
```

```
R1 (config) #Route-map EXIST permit 10
```

```
R1 (config-route-map) #match ip addr 1
```

Final Step – With the following configuration, we are instructing BGP for the conditions of the task's requirements:

```
R1 (config) #Router bgp 100
```

```
R1 (config-router) #Neighbor 13.1.1.3 advertise-map ADV exist-map EXIST
```

To test the first condition

If both networks (1.0.0.0 /8 and 2.0.0.0 /8) are up, then both networks should be advertised to R3.

NOTE: Both prefixes are up:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0	200 i
*> 3.0.0.0	13.1.1.3	0		0	300 i

Let's see the prefixes that R1 is advertising to R3:

On R1:

```
R1#Show ip bgp neighbor 13.1.1.3 advertised-routes | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0	200 i

As you can see both prefixes are being advertised to R3, let's check R3's BGP table:

On R3:

```
R3#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	13.1.1.1	0		0	100 i
*> 2.0.0.0	13.1.1.1			0	100 200 i
*> 3.0.0.0	0.0.0.0	0		32768	i

To test the second condition

If network 1.0.0.0 /8 is down, R1 should NOT advertise network 2.0.0.0 /8 to R3.

Let's shutdown R1's Lo0 interface:

On R1:

```
R1(config)#Int lo0  
R1(config-if)#Shut
```

To force the change much faster:

On R1:

```
R1#Clear ip bgp *
```

Let's see the prefixes that R1 is advertising to R3:

```
R1#Show ip bgp neighbor 13.1.1.3 advertised-routes | B Network  
R1#
```

```
R1#Sh ip bgp 2.0.0.0
```

BGP routing table entry for 2.0.0.0/8, version 4

Paths: (1 available, best #1, table default)

Not advertised to any peer

200

12.1.1.2 from 12.1.1.2 (10.1.2.2)

Origin IGP, metric 0, localpref 100, valid, external, best

NOTE: The output of the above show command reveals that R1 is NOT advertising any prefixes to R3. Let's check R3's BGP table to verify:

On R3:

R3#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 3.0.0.0	0.0.0.0	0		32768	i

To test the third condition

If network 2.0.0.0 /8 is down, then R1 should only advertise network 1.0.0.0 /8 to R3.

Let's "NO shut" R1's Lo0 and shutdown R2's Lo0:

On R1:

```
R1 (config)#Int Lo0
R1 (config-if)#NO Shut
```

On R2:

```
R2 (config)#Int Lo0
R2 (config-if)#Shut
```

To force the change much faster:

On R1:

R1#Clear ip bgp *

Let's see which prefixes are advertised to R3 by R1:

R1#Show ip bgp neighbor 13.1.1.3 advertised-routes | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i

To verify the configuration

On R3:

R3#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path

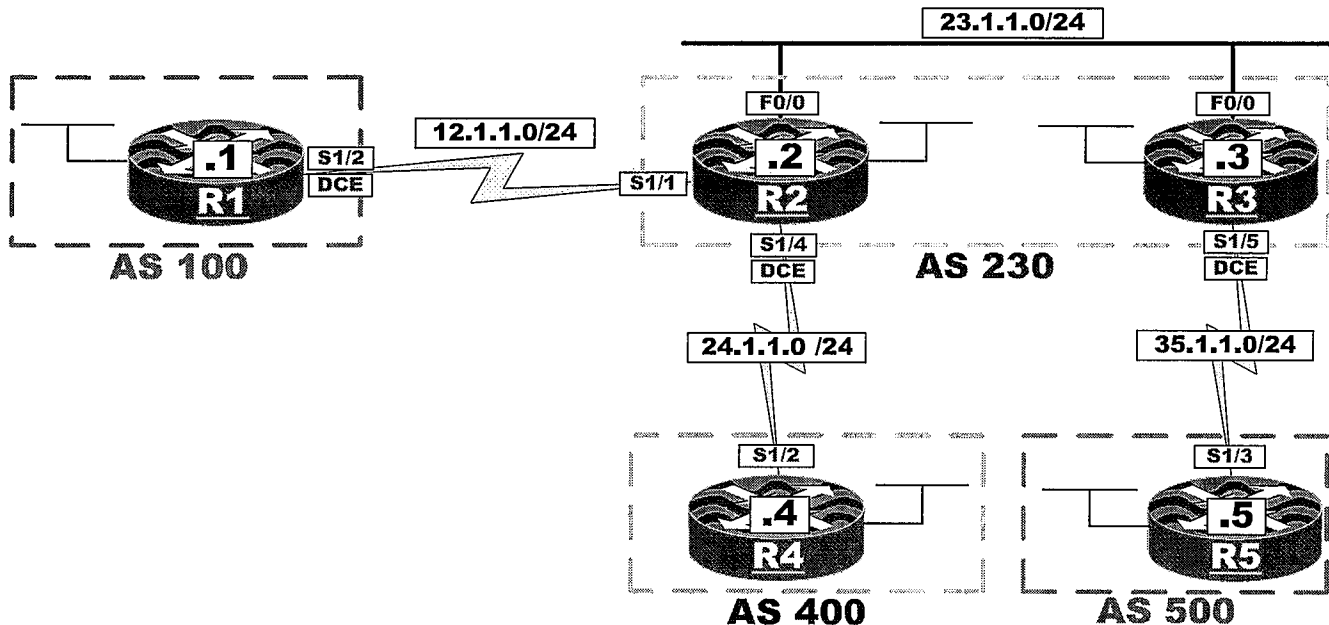
```
*> 1.0.0.0      13.1.1.1      0      0 100 i
*> 3.0.0.0      0.0.0.0      0      32768 i
```

Task 11

Erase the startup config and reload the routers before proceeding to the next lab.

Lab 4

The Community attribute



Lab Setup:

To copy and paste the initial configurations, go to "<http://www.micronics.nl>" → "Foundation" → BGP → "Lab-4".

Task 1

Configure BGP peering based on the above topology, these routers should advertise their loopback0 interfaces in BGP. Ensure reachability between the loopback interfaces. DO NOT configure an IGP, or default routes to accomplish this task.

On R1:

```
R1 (config) #Router bgp 100
R1 (config-router) #Netw 1.0.0.0
R1 (config-router) #No au
```

```
R1(config-router)#Neighbor 12.1.1.2 remote-as 230
```

On R2:

```
R2(config)#Router bgp 230
R2(config-router)#netw 2.0.0.0
R2(config-router)#No au
R2(config-router)#Neighbor 12.1.1.1 remote-as 100
R2(config-router)#Neighbor 23.1.1.3 remote-as 230
R2(config-router)#Neighbor 23.1.1.3 next-hop-self
R2(config-router)#Neighbor 24.1.1.4 remote-as 400
```

On R3:

```
R3(config)#Router bgp 230
R3(config-router)#Netw 3.0.0.0
R3(config-router)#No au
R3(config-router)#Neighbor 23.1.1.2 remote-as 230
R3(config-router)#Neighbor 23.1.1.2 next-hop-self
R3(config-router)#Neighbor 35.1.1.5 remote-as 500
```

On R4:

```
R4(config)#Router bgp 400
R4(config-router)#Netw 4.0.0.0
R4(config-router)#No au
R4(config-router)#Neighbor 24.1.1.2 remote-as 230
```

On R5:

```
R5(config)#Router bgp 500
R5(config-router)#Netw 5.0.0.0
R5(config-router)#No au
R5(config-router)#Neighbor 35.1.1.3 remote-as 230
```

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	1.0.0.0	0.0.0.0	0		32768	i
*>	2.0.0.0	12.1.1.2	0		0	230 i

```
*> 3.0.0.0          12.1.1.2          0 230 i
*> 4.0.0.0          12.1.1.2          0 230 400 i
*> 5.0.0.0          12.1.1.2          0 230 500 i
```

On R2:

R2#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	12.1.1.1	0		0	100 i
*> 2.0.0.0	0.0.0.0	0		32768	i
*>i3.0.0.0	23.1.1.3	0	100	0	i
*> 4.0.0.0	24.1.1.4	0		0	400 i
*>i5.0.0.0	23.1.1.3	0	100	0	500 i

On R3:

R3#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	23.1.1.2	0	100	0	100 i
*>i2.0.0.0	23.1.1.2	0	100	0	i
*> 3.0.0.0	0.0.0.0	0		32768	i
*>i4.0.0.0	23.1.1.2	0	100	0	400 i
*> 5.0.0.0	35.1.1.5	0		0	500 i

On R4:

R4#Show ip bgp | B Netw

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	24.1.1.2			0	230 100 i
*> 2.0.0.0	24.1.1.2	0		0	230 i
*> 3.0.0.0	24.1.1.2			0	230 i
*> 4.0.0.0	0.0.0.0	0		32768	i
*> 5.0.0.0	24.1.1.2			0	230 500 i

On R5:

R5#Show ip bgp | B Net

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	35.1.1.3			0	230 100 i
*> 2.0.0.0	35.1.1.3			0	230 i
*> 3.0.0.0	35.1.1.3	0		0	230 i


```
*> 4.0.0.0          35.1.1.3          0 230 400 i
*> 5.0.0.0          0.0.0.0          0      32768 i
```

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 = 28/29/32 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 = 28/30/32 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

!!!!!!

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

R1#Ping 5.5.5.5 source 1.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 5.5.5.5, 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 = 56/57/60 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 = 28/30/32 ms

On R2:

R2#**Ping 5.5.5.5 source 2.2.2.2**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 5.5.5.5, 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 = 28/29/32 ms

Task 2

Using the community attribute, configure R1 such that when it advertises network 1.0.0.0 /8 to R2 in AS 230, the network is not advertised to any of R2's IBGP or EBGP neighbor/s.

The community attribute is a numerical value that can be attached to a given prefix and advertised to a specific neighbor, once the neighbor receives the prefix, it will examine the community value and it will perform either filtering or use that value for route selection process.

By default, no community attribute/s is sent to any neighbor. To specify that a community attribute should be sent to a BGP neighbor, the "neighbor send-community" command is configured in the router config mode.

The well-known communities are as follows:

- Internet – If assigned to a network/s, that network/s should be advertised to all neighbors.
- Local-as – If assigned to a network/s, that network/s should only be advertised within that AS.
- No-advertise – If assigned to a network/s. that network/s should not be advertised to any BGP neighbor.
- No-export – If assigned to a network/s, that network/s should not be advertised to an EBGP neighbor.

On R1:

Note: Before configuring an access-list, always performs a "Show access-list" command to ensure that one does not exist.

```
R1(config)#Access-list 1 permit 1.0.0.0 0.255.255.255
```

The access-list is used to identify the network; Prefix-lists can also be used for this purpose.

```
R1(config)#Route-map TEST permit 10
R1(config-route-map)#match ip addr 1
R1(config-route-map)#set community no-advertise

R1(config-route-map)#Route-map TEST permit 90
```

Note: The above route-map matches on the access-list and sets the community to “no-advertise”, this well-known community attribute tells the receiving router not to advertise the prefix to any of its neighbor/s (IBGP or EBGP).

The “route-map TST permit 90” is the catchall route-map; it matches any network not matched with the match keyword in the “route-map TST permit 10”.

```
R1(config)#Router bgp 100
R1(config-router)#Neighbor 12.1.1.2 send-community
R1(config-router)#Neighbor 12.1.1.2 route-map TEST out
```

In the above commands, we are sending the community and assigning the route-map to a given neighbor. The direction of the route-map specifies which router’s decision should be influenced by this policy, if it should affect neighbor’s decision, then, the direction of the route-map should be “out”, but if the local router’s decision should be influenced, then, the direction of the route-map should be “in”.

To verify the configuration:

On R2:

```
R2#Sh ip bgp 1.0.0.0
```

```
BGP routing table entry for 1.0.0.0/8, version 18
Paths: (1 available, best #1, table default, not advertised to any peer)
  Not advertised to any peer
  100
    12.1.1.1 from 12.1.1.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: no-advertise
```

Since R2 does not advertise the network, R3 and the other EBGP and/or IBGP neighbor/s will not have any knowledge of this prefix.

On R3:

```
R3#Show ip bgp 1.0.0.0
```

```
% Network not in table
```

On R4:

```
R4#Show ip bgp 1.0.0.0
```

```
% Network not in table
```

On R5:

```
R5#Show ip bgp 1.0.0.0
```

```
% Network not in table
```

Task 3

Configure R5 such that when it advertises its network 5.0.0.0 to R3 in AS 230, the routers in AS 230 **do not** advertise that network to any of their EBGP peer/s. **Do not** configure filtering on R3 to accomplish this task.

On R5:

```
R5(config)#IP Prefix-list NET-5 permit 5.0.0.0/8
```

```
R5(config)#Route-map TST permit 10
```

```
R5(config-route-map)#match ip addr prefix NET-5
```

```
R5(config-route-map)#set community no-export
```

```
R5(config-route-map)#Route-map TST permit 90
```

```
R5(config)#Router bgp 500
```

```
R5(config-router)#Neighbor 35.1.1.3 send-community
```

```
R5(config-router)#Neighbor 35.1.1.3 route-map TST out
```

This is another well-known community. In this case network 5.0.0.0 will only be advertised to the routers in AS 230. The routers in AS 230 will not advertise this network to any of their EBGP neighbor/s.

Remember....by default routers will strip the community attribute when they send the community in the route-map to a given neighbor (IBGP, or EBGP), therefore, in this case, R3 should be configured to send community to R2, or else, R2 will advertise that network to its EBGP peers.

On R3:

```
R3(config)#Router bgp 230
R3(config-router)#Neighbor 23.1.1.2 send-community
```

To verify the configuration:

On R3:

```
R3#Sh ip bgp 5.0.0.0
```

```
BGP routing table entry for 5.0.0.0/8, version 8
Paths: (1 available, best #1, table default, not advertised to EBGp peer)
  Advertised to update-groups:
    10
  500
    35.1.1.5 from 35.1.1.5 (5.5.5.5)
      Origin IGP, metric 0, localpref 100, valid, external, best
      Community: no-export
```

On R2:

```
R2#Show ip bgp 5.0.0.0
```

```
BGP routing table entry for 5.0.0.0/8, version 19
Paths: (1 available, best #1, table default, not advertised to EBGp peer)
  Not advertised to any peer
  500
    23.1.1.3 from 23.1.1.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Community: no-export
```

On R4:

```
R4#Show ip bgp 5.0.0.0
% Network not in table
```

On R1:

```
R1#Show ip bgp 5.0.0.0
% Network not in table
```

Task 4

Configure R3 in AS 230 to advertise network 3.0.0.0/8 to the routers in its own AS **only**. YOU MUST use a "Network" command to accomplish this task.

On R3:

```
R3(config)#Route-map TST permit 10
R3(config-route-map)#set community local-as

R3(config)#Router bgp 230
R3(config-router)#Network 3.0.0.0 route-map TST
R3(config-router)#Neighbor 23.1.1.2 send-community
```

To verify the configuration:

On R3:

```
R3#Sh ip bgp 3.0.0.0
```

```
BGP routing table entry for 3.0.0.0/8, version 9
Paths: (1 available, best #1, table default, not advertised outside local
AS)
  Advertised to update-groups:
    10
  Local
    0.0.0.0 from 0.0.0.0 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced,
local, best
      Community: local-AS
```

On R5:

```
R5#Show ip bgp 3.0.0.0
```

```
% Network not in table
```

On R2:

```
R2#Show ip bgp 3.0.0.0
```

```
BGP routing table entry for 3.0.0.0/8, version 20
Paths: (1 available, best #1, table default, not advertised outside local
AS)
```

Not advertised to any peer

Local

23.1.1.3 from 23.1.1.3 (3.3.3.3)

Origin IGP, metric 0, localpref 100, valid, internal, best

Community: local-AS

On R1:

R1#Show ip bgp 3.0.0.0

% Network not in table

In this topology the "No-export" and the "Local-as" have the same behavior; the "No-export" community does not look at the "BGP Confederation identifier" command, therefore, it will keep the prefix/es local in the AS number that the BGP router is configured in, whereas, the "Local-as" community looks at the "BGP Confederation identifier" command first, if it's configured, it will keep the prefix/es within that AS, but if the "BGP Confederation identifier" is NOT configured, it will keep the prefix/es local to the AS that the BGP is configured on.

Task 5

R1 is advertising network 1.0.0.0 which has an attached community attribute of "no-advertise" to R2 (task 2). Router R2 should be configured to advertise network 1.0.0.0 to all of its IBGP and EBGP peers. You should utilize a community attribute to accomplish this task.

On R2:

Before configuring R2, we should display the prefix in the BGP table:

R2#Show ip bgp 1.0.0.0

BGP routing table entry for 1.0.0.0/8, version 18

Paths: (1 available, best #1, table default, **not advertised to any peer**)

Not advertised to any peer

100

12.1.1.1 from 12.1.1.1 (1.1.1.1)

Origin IGP, metric 0, localpref 100, valid, external, best

Community: no-advertise

R2 can be configured to assign the "Internet" community to this network; when the "Internet" community is assigned to a network, that network will be advertised to all peers (IBGP or EBGP).

To configure:

On R2:

```
R2(config)#Access-list 1 permit 1.0.0.0

R2(config)#Route-map TST permit 10
R2(config-route-map)#match ip addr 1
R2(config-route-map)#set community internet

R2(config)#Router bgp 230
R2(config-router)#Neighbor 12.1.1.1 route-map TST in
```

To verify the configuration:

On R2:

```
R2#Show ip bgp 1.0.0.0
```

```
R2#sh ip bgp 1.0.0.0
BGP routing table entry for 1.0.0.0/8, version 21
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    6          7
  100
    12.1.1.1 from 12.1.1.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best
  Community: internet
```

On R4:

```
R4#Show ip bgp | b Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	24.1.1.2			0	230 100 i
*> 2.0.0.0	24.1.1.2	0		0	230 i
*> 4.0.0.0	0.0.0.0	0		32768	i

On R5:

```
R5#Show ip bgp | b Network
```

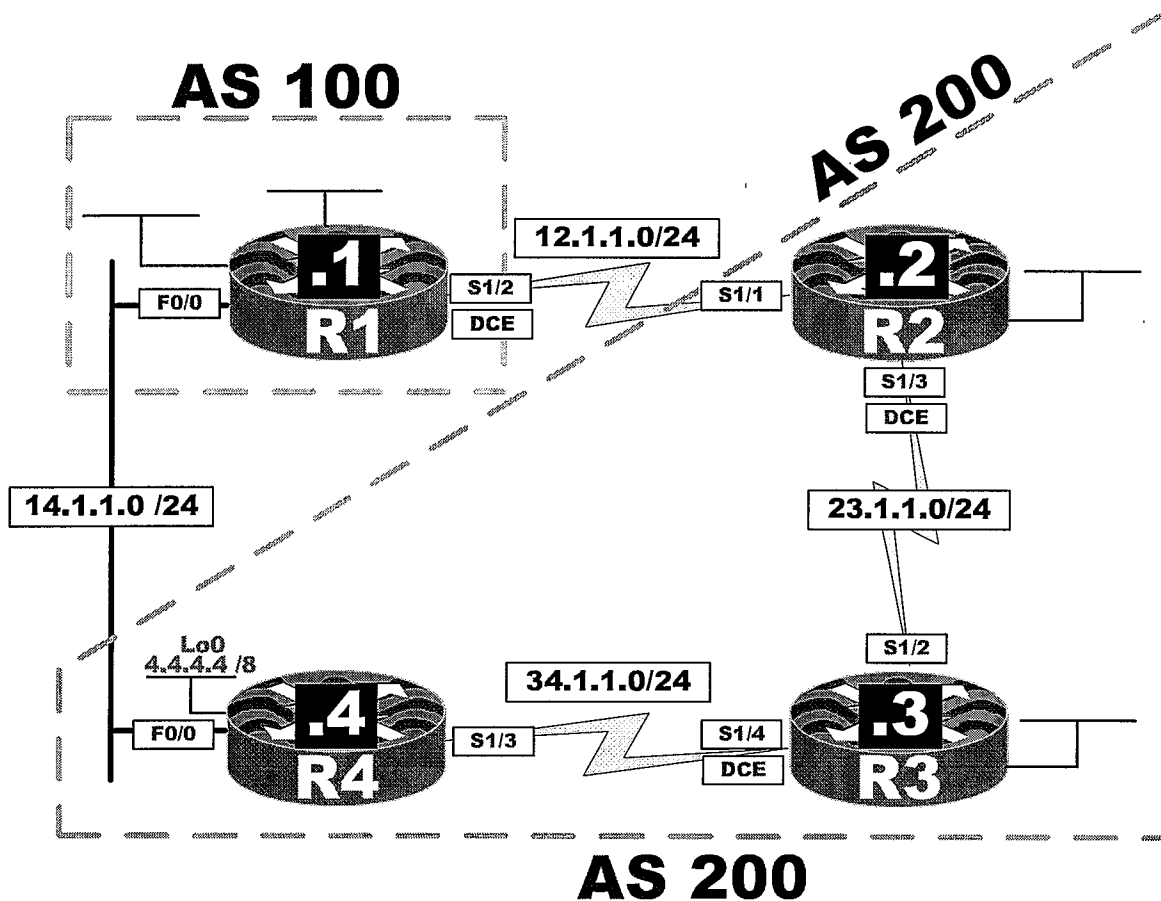
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	35.1.1.3			0	230 100 i

*> 2.0.0.0	35.1.1.3		0 230 i
*> 4.0.0.0	35.1.1.3		0 230 400 i
*> 5.0.0.0	0.0.0.0	0	32768 i

Task 6

Erase the startup config and reload the routers before proceeding to the next task.

Lab 5 - The AS-Path Attribute



Lab Setup:

To copy and paste the initial configurations, go to "<http://www.micronics.nl>" → "Foundation" → "BGP" → "Lab-5".

Task 1

Configure R2, R3 and R4 in AS 200, these routers should have full mesh peer session between them. R2 and R4 should have an EBGP peer session with R1 in AS 100. These routers should ONLY advertise their loopback interface/s in BGP. Provide NLRI for the links using OSPF.

On R1:

```
R1(config)#Router bgp 100
R1(config-router)#No au
R1(config-router)#Neighbor 12.1.1.2 remote-as 200
R1(config-router)#Neighbor 14.1.1.4 remote-as 200
R1(config-router)#Network 1.0.0.0
R1(config-router)#Network 11.0.0.0
```

On R2:

```
R2(config)#Router bgp 200
R2(config-router)#No au
R2(config-router)#Netw 2.0.0.0
R2(config-router)#Neighbor 12.1.1.1 remote-as 100
R2(config-router)#Neighbor 23.1.1.3 remote-as 200
R2(config-router)#Neighbor 23.1.1.3 next-hop-self
R2(config-router)#Neighbor 34.1.1.4 next-hop-self
R2(config-router)#Neighbor 34.1.1.4 remote-as 200

R2(config)#Router ospf 1
R2(config-router)#Network 23.1.1.2 0.0.0.0 area 0
```

On R3:

```
R3(config)#Router bgp 200
R3(config-router)#No au
R3(config-router)#Netw 3.0.0.0
R3(config-router)#Neighbor 34.1.1.4 remote-as 200
R3(config-router)#Neighbor 23.1.1.2 remote-as 200

R3(config)#Router ospf 1
R3(config-router)#Network 23.1.1.3 0.0.0.0 area 0
R3(config-router)#Network 34.1.1.3 0.0.0.0 area 0
```

On R4:

```
R4(config)#Router bgp 200
R4(config-router)#No au
R4(config-router)#Netw 4.0.0.0
R4(config-router)#Neighbor 34.1.1.3 remote-as 200
R4(config-router)#Neighbor 34.1.1.3 next-hop-self
R4(config-router)#Neighbor 23.1.1.2 remote-as 200
R4(config-router)#Neighbor 23.1.1.2 next-hop-self
R4(config-router)#Neighbor 14.1.1.1 remote-as 100
```

```
R4(config)#Router ospf 1
R4(config-router)#Network 34.1.1.4 0.0.0.0 area 0
```

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
* 2.0.0.0	14.1.1.4			0	200 i
*> 12.1.1.2		0		0	200 i
* 3.0.0.0	12.1.1.2			0	200 i
*> 14.1.1.4				0	200 i
* 4.0.0.0	14.1.1.4	0		0	200 i
*> 12.1.1.2				0	200 i
*> 11.0.0.0	0.0.0.0	0		32768	i

On R2:

```
R2#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i1.0.0.0	14.1.1.1	0	100	0	100 i
*> 12.1.1.1		0		0	100 i
*> 2.0.0.0	0.0.0.0	0		32768	i
*>i3.0.0.0	23.1.1.3	0	100	0	i
*>i4.0.0.0	34.1.1.4	0	100	0	i
* i11.0.0.0	14.1.1.1	0	100	0	100 i
*> 12.1.1.1		0		0	100 i

On R3:

```
R3#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i1.0.0.0	34.1.1.4	0	100	0	100 i
*>i 23.1.1.2		0	100	0	100 i
*>i2.0.0.0	23.1.1.2	0	100	0	i
*> 3.0.0.0	0.0.0.0	0		32768	i
*>i4.0.0.0	34.1.1.4	0	100	0	i
* i11.0.0.0	34.1.1.4	0	100	0	100 i
*>i 23.1.1.2		0	100	0	100 i

On R4:

R4#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
* i1.0.0.0	23.1.1.2	0	100	0	100 i
*>	14.1.1.1	0		0	100 i
*>i2.0.0.0	23.1.1.2	0	100	0	i
*>i3.0.0.0	34.1.1.3	0	100	0	i
*> 4.0.0.0	0.0.0.0	0		32768	i
* i11.0.0.0	23.1.1.2	0	100	0	100 i
*>	14.1.1.1	0		0	100 i

Task 2

Configure R1 in AS 100 such that routers in AS 200 use the link through R4-R1 to reach its network 1.0.0.0 /8. Use the AS-Path attribute to accomplish this task.

The AS-path attribute can be manipulated to accomplish this task. This attribute can be manipulated by prepending the AS numbers to existing AS paths. Typically, you perform AS-Path prepending on the outgoing EGBP updates over the non-desired return path, making the AS-Path over the non-desired link become longer than the desired path, and as a result making the desired path the primary.

On R1:

```
R1(config)#Access-list 1 permit 1.0.0.0 0.255.255.255
```

```
R1(config)#Route-map TST1 permit 10
```

```
R1(config-route-map)#match ip addr 1
```

```
R1(config-route-map)#set as-path prepend 100
```

```
R1(config)#route-map TST1 permit 90
```

```
R1(config)#Router bgp 100
```

```
R1(config-router)#Neighbor 12.1.1.2 route-map TST1 out
```

```
R1#Clear ip bgp * Out
```

To verify the configuration:

On R2:

R2#Sh ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	34.1.1.4	0	100	0	100 i
*	12.1.1.1	0		0	100 100 i
*> 2.0.0.0	0.0.0.0	0		32768	i
*>i3.0.0.0	23.1.1.3	0	100	0	i
*>i4.0.0.0	34.1.1.4	0	100	0	i
* i11.0.0.0	34.1.1.4	0	100	0	100 i
*>	12.1.1.1	0		0	100 i

To test the configuration:

On R2:

```
R2(config)#Router bgp 200
R2(config-router)#Neighbor 12.1.1.1 default-originate
```

R2#Clear ip bgp * Out

On R4:

```
R4(config)#Router bgp 200
R4(config-router)#Neighbor 14.1.1.1 default-originate
```

R4#Clear ip bgp * out

On R2:

R2#Traceroute 1.1.1.1 numeric

Type escape sequence to abort.

Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

1	23.1.1.3	12 msec	12 msec	16 msec
2	34.1.1.4	28 msec	32 msec	28 msec
3	14.1.1.1	20 msec	*	20 msec

On R3:

R3#Sh ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	34.1.1.4	0	100	0	100 i
*>i2.0.0.0	23.1.1.2	0	100	0	i

```
*> 3.0.0.0          0.0.0.0          0          32768 i
*>i4.0.0.0          34.1.1.4          0          100          0 i
* i11.0.0.0         34.1.1.4          0          100          0 100 i
*>i                 23.1.1.2          0          100          0 100 i
```

R3#Traceroute 1.1.1.1 numeric

Type escape sequence to abort.

Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

```
1 34.1.1.4 16 msec 16 msec 12 msec
2 14.1.1.1 20 msec * 24 msec
```

On R4:

R4#Sh ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	14.1.1.1	0		0	100 i
*>i2.0.0.0	23.1.1.2	0	100	0	i
*>i3.0.0.0	34.1.1.3	0	100	0	i
*> 4.0.0.0	0.0.0.0	0		32768	i
* i11.0.0.0	23.1.1.2	0	100	0	100 i
*>	14.1.1.1	0		0	100 i

R4#Traceroute 1.1.1.1 numeric

Type escape sequence to abort.

Tracing the route to 1.1.1.1

```
1 14.1.1.1 0 msec * 0 msec
```

Note the AS-path attribute is used to influence the degree of preference in another AS. R2, R3 and R4 will go through R4 to reach network 1.0.0.0 /8 because going through R4 they have to traverse through less AS Hops.

Task 3

Configure R1 in AS 100 such that the routers in AS 200 use the link through R2-R1 to reach network 11.0.0.0 /8. Use the AS-Path attribute and an IP Prefix-list to accomplish this task.

On R1:

```
R1(config)#IP Prefix-list NET11 permit 11.0.0.0/8
```

```
R1(config)#Route-map TST11 permit 10
```

```
R1(config-route-map)#match ip addr prefix NET11
```

```
R1(config-route-map)#set as-path prepend 100 100
```

```
R1(config)#route-map TST11 permit 90
```

```
R1(config-route-map)#Router bgp 100
```

```
R1(config-router)#Neighbor 14.1.1.4 route-map TST11 out
```

```
R1#Clear ip bgp * Out
```

To verify the configuration:

On R2:

```
R2#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
0.0.0.0	0.0.0.0				0 i
*>i1.0.0.0	34.1.1.4	0	100	0	100 i
*	12.1.1.1	0		0	100 100 i
*> 2.0.0.0	0.0.0.0	0		32768	i
*>i3.0.0.0	23.1.1.3	0	100	0	i
*>i4.0.0.0	34.1.1.4	0	100	0	i
*> 11.0.0.0	12.1.1.1	0		0	100 i

```
R2#Traceroute 11.1.1.1
```

Type escape sequence to abort.

Tracing the route to 11.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

```
1 12.1.1.1 16 msec * 12 msec
```

On R3:

```
R3#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	34.1.1.4	0	100	0	100 i
*>i2.0.0.0	23.1.1.2	0	100	0	i


```
*> 3.0.0.0          0.0.0.0          0          32768 i
*>i4.0.0.0          34.1.1.4          0          100          0 i
*>i11.0.0.0         23.1.1.2          0          100          0 100 i
```

R3#Traceroute 11.1.1.1

Type escape sequence to abort.
Tracing the route to 11.1.1.1

```
1 23.1.1.2 12 msec 12 msec 16 msec
2 12.1.1.1 28 msec * 28 msec
```

On R4:

R4#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	14.1.1.1	0			0 100 i
*>i2.0.0.0	23.1.1.2	0	100		0 i
*>i3.0.0.0	34.1.1.3	0	100		0 i
*> 4.0.0.0	0.0.0.0	0		32768	i
*>i11.0.0.0	23.1.1.2	0	100		0 100 i
*	14.1.1.1	0			0 100 100 100 i

R4#Traceroute 11.1.1.1

Type escape sequence to abort.
Tracing the route to 11.1.1.1

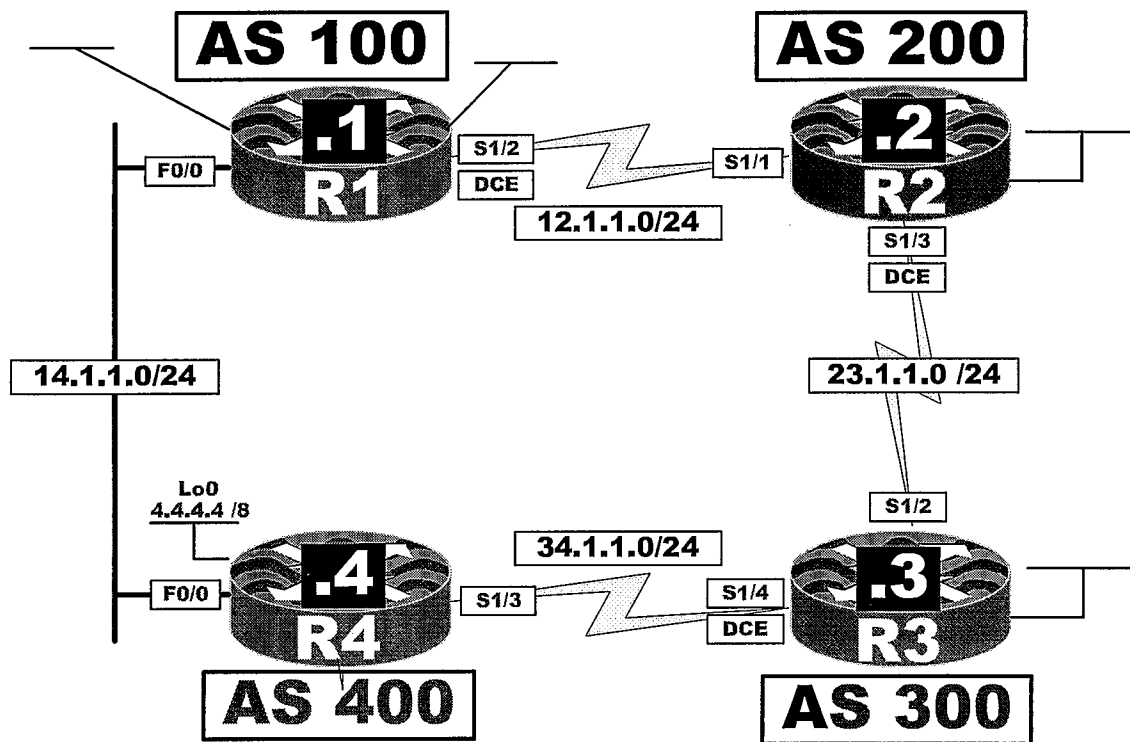
```
1 34.1.1.3 12 msec 12 msec 16 msec
2 23.1.1.2 28 msec 28 msec 32 msec
3 12.1.1.1 40 msec * 40 msec
```

Note all the routers will take the R2-R1 link to connect to network 11.0.0.0/8.

Task 4

Erase the startup configuration and reload the routers before proceeding to the next lab.

Lab 6 - The Weight Attribute



Lab Setup:

To copy and paste the initial configurations, go to "<http://www.micronics.nl>" → "Foundation" → "BGP" → "Lab-6".

The initial configuration file configures RIPv2 on all routers advertising the links to provide NLRI.

Task 1

Configure R1 in AS 100 to establish an EBGP peer session with R2 in AS 200 and R4 in AS 400. Router R2 should establish EBGP peer sessions with R1 in AS 100 and R3 in AS 300. Router R3 should establish EBGP peer sessions with R2 in AS 200 and R4 in AS 400. Router R4 should establish EBGP peer sessions with R1 in AS 100 and R3 in AS 300. The BGP routers should ONLY advertise their loopback/s in BGP.

On R1:

```
R1(config-if)#Router bgp 100
R1(config-router)#No au
R1(config-router)#Netw 1.0.0.0
R1(config-router)#Netw 11.0.0.0
R1(config-router)#Neighbor 14.1.1.4 remote-as 400
R1(config-router)#Neighbor 12.1.1.2 remote-as 200
```

On R2:

```
R2(config)#Router bgp 200
R2(config-router)#No au
R2(config-router)#Netw 2.0.0.0
R2(config-router)#Neighbor 12.1.1.1 remote-as 100
R2(config-router)#Neighbor 23.1.1.3 remote-as 300
```

On R3:

```
R3(config)#Router bgp 300
R3(config-router)#No au
R3(config-router)#Netw 3.0.0.0
R3(config-router)#Neighbor 34.1.1.4 remote-as 400
R3(config-router)#Neighbor 23.1.1.2 remote-as 200
```

On R4:

```
R4(config)#Router bgp 400
R4(config-router)#No au
R4(config-router)#Netw 4.0.0.0
R4(config-router)#Neighbor 34.1.1.3 remote-as 300
R4(config-router)#Neighbor 14.1.1.1 remote-as 100
```

To verify the configuration:

On R4:

```
R4#Sh ip bgp | B Network
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*	1.0.0.0	34.1.1.3			0	300 200 100 i
*>		14.1.1.1	0		0	100 i
*>	2.0.0.0	34.1.1.3			0	300 200 i
*		14.1.1.1			0	100 200 i

```

*> 3.0.0.0          34.1.1.3          0          0 300 i
*          14.1.1.1          0 100 200 300 i
*> 4.0.0.0          0.0.0.0          0          32768 i
* 11.0.0.0          34.1.1.3          0 300 200 100 i
*>          14.1.1.1          0 100 i

```

On R1:

R1#**Show ip bgp | B Network**

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
* 2.0.0.0	14.1.1.4			0	400 300 200 i
*>	12.1.1.2	0		0	200 i
* 3.0.0.0	14.1.1.4			0	400 300 i
*>	12.1.1.2			0	200 300 i
*> 4.0.0.0	14.1.1.4	0		0	400 i
*> 11.0.0.0	0.0.0.0	0		32768	i

Task 3

Configure R1 in AS 100 to use AS 200 to reach all the prefixes within this topology; you must use the "Weight" attribute to accomplish this task.

The weight attribute is a Cisco-defined attribute that is local to the router. This attribute is NOT advertised to any BGP neighbor/s. If there is more than one route to a given destination, the weight attribute can decide which path is the best path. The higher the value the better the preference.

The following assigns a weight attribute of 40000 to all existing and future prefixes:

On R1:

```

R1(config)#Router bgp 100
R1(config-router)#Neighbor 12.1.1.2 weight 40000

```

```

R1#Clear ip bgp *

```

To verify the configuration:

On R1:

R1#**Show ip bgp | B Network**

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
* 2.0.0.0	14.1.1.4			0	400 300 200 i
*> 12.1.1.2	12.1.1.2	0		40000	200 i
* 3.0.0.0	14.1.1.4			0	400 300 i
*> 12.1.1.2	12.1.1.2			40000	200 300 i
* 4.0.0.0	14.1.1.4	0		0	400 i
*> 12.1.1.2	12.1.1.2			40000	200 300 400 i
*> 11.0.0.0	0.0.0.0	0		32768	i

We can see that R1 has to traverse through R2 to reach all the networks within this topology.

R1#Traceroute 4.4.4.4

Type escape sequence to abort.

Tracing the route to 4.4.4.4

VRF info: (vrf in name/id, vrf out name/id)

```

  1 12.1.1.2 52 msec 16 msec 16 msec
  2 23.1.1.3 28 msec 28 msec 32 msec
  3 34.1.1.4 16 msec * 16 msec

```

R1#Traceroute 3.3.3.3

Type escape sequence to abort.

Tracing the route to 3.3.3.3

VRF info: (vrf in name/id, vrf out name/id)

```

  1 12.1.1.2 16 msec 12 msec 16 msec
  2 23.1.1.3 32 msec * 28 msec

```

R1#Traceroute 2.2.2.2

Type escape sequence to abort.

Tracing the route to 2.2.2.2

VRF info: (vrf in name/id, vrf out name/id)

```

  1 12.1.1.2 16 msec * 12 msec

```

Task 4

Configure and advertise the following Loopback interface on R3.

Loopback 1 – 33.3.3.3 /24

On R3:

```

R3(config)#Int Lo1
R3(config-if)#IP addr 33.3.3.3 255.255.255.0

R3(config)#Router bgp 300
R3(config-router)#Network 33.3.3.0 mask 255.255.255.0

```

To verify the configuration:

On R3:

```
R3#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* 1.0.0.0	34.1.1.4			0	400 100 i
*>	23.1.1.2			0	200 100 i
*> 2.0.0.0	23.1.1.2	0		0	200 i
*> 3.0.0.0	0.0.0.0	0		32768	i
*> 4.0.0.0	34.1.1.4	0		0	400 i
* 11.0.0.0	34.1.1.4			0	400 100 i
*>	23.1.1.2			0	200 100 i
*> 33.3.3.0/24	0.0.0.0	0		32768	i

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
* 2.0.0.0	14.1.1.4			0	400 300 200 i
*>	12.1.1.2	0		40000	200 i
* 3.0.0.0	14.1.1.4			0	400 300 i
*>	12.1.1.2			40000	200 300 i
* 4.0.0.0	14.1.1.4	0		0	400 i
*>	12.1.1.2			40000	200 300 400 i
*> 11.0.0.0	0.0.0.0	0		32768	i
*> 33.3.3.0/24	12.1.1.2			40000	200 300 i
*	14.1.1.4			0	400 300 i

NOTE: The "Weight" attribute configured in Task 3 affected the existing and the future prefix/es within the topology. To verify this information:

```
R1#Traceroute 33.3.3.3
```

```

Type escape sequence to abort.
Tracing the route to 33.3.3.3

```

```
VRF info: (vrf in name/id, vrf out name/id)
 1 12.1.1.2 12 msec 16 msec 16 msec
 2 23.1.1.3 28 msec *   28 msec
```

Task 5

Re-configure R1 based on the following policy, this policy may override the previously implemented policy:

- R1 in AS 100 should use R4 in AS 400 to reach networks 33.0.0.0 /8 and network 4.0.0.0 /8.
- R1 in AS 100 should use R2 in AS 200 to reach network 3.0.0.0 /8, and existing and future prefixes advertised in AS 200.

Let's remove the "Weight" configuration that was implemented in task 3:

On R1:

```
R1 (config) #Router bgp 100
R1 (config-router) #No Neigh 12.1.1.2 weight 40000

R1 #Clear ip bgp *
```

To configure this task, two access-lists are configured, the first access-list identifies networks 4.0.0.0 and 33.3.3.0/8, and the second one identifies network 4.0.0.0/8, but how do we identify the existing and the future prefixes advertised in AS 200? The easiest way to accomplish this is to configure an AS-Path access-list identifying the AS number, because the AS number remains the same for existing and the future routes in AS 200:

The following access-list identifies networks 33.3.3.0 /24 and 4.0.0.0 /8:

```
R1 (config) #Access-list 1 permit 33.3.3.0 0.0.0.255
R1 (config) #Access-list 1 permit 4.0.0.0 0.255.255.255
```

The following access-list identifies networks 3.0.0.0 /8:

```
R1 (config) #Access-list 2 permit 3.0.0.0 0.255.255.255
```

The following as-path access-list identifies AS 200:

```
R1(config)#Ip as-path access-list 1 permit ^200$
```

Once the prefixes are all identified, they are referenced in a route-map and the “Weight” attribute is assigned to them based on the requirements of this task.

The following route-map (TST) assigns a weight attribute of 45000 to the networks identified in the access-list 1.

```
R1(config)#Route-map TST permit 10
R1(config-route-map)#match ip addr 1
R1(config-route-map)#set weight 45000
R1(config)#Route-map TST permit 90
```

The following route-map (TEST) assigns a weight attribute of 50000 to the networks identified in the access-list 2 and AS 200.

```
R1(config)#Route-map TEST per 10
R1(config-route-map)#match as-path 1
R1(config-route-map)#set weight 50000
```

```
R1(config)#Route-map TEST per 20
R1(config-route-map)#match ip addr 2
R1(config-route-map)#set weight 50000
R1(config)#Route-map TEST per 90
```

```
R1(config)#Router bgp 100
R1(config-router)#Neighbor 14.1.1.4 route-map TST in
R1(config-router)#Neighbor 12.1.1.2 route-map TEST in
```

The above commands assign the attributes to neighbors R2 and R4.

```
R1#Clear ip bgp *
```

To verify the configuration:

On R1:

```
R1#Sh ip bgp | B Network
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	1.0.0.0	0.0.0.0	0		32768	i
*	2.0.0.0	14.1.1.4			0 400 300 200	i
*>		12.1.1.2	0		50000 200	i


```

* 3.0.0.0          14.1.1.4          0 400 300 i
*>                12.1.1.2          50000 200 300 i
*> 4.0.0.0          14.1.1.4          0 45000 400 i
*                12.1.1.2          0 200 300 400 i
*> 11.0.0.0         0.0.0.0          0 32768 i
*> 33.3.3.0/24      14.1.1.4          45000 400 300 i
*                12.1.1.2          0 200 300 i

```

NOTE: Network 3.0.0.0/8 and Networks that originated in AS 200 are assigned a weight attribute of 50000, and networks 4.0.0.0 and 33.3.3.0 /24 are assigned a weight attribute of 45000, therefore, R1 goes through R4 to reach networks 33.3.3.0/24 and 4.0.0.0/8, and it goes through R2 for all existing and future prefixes that originated in AS 200, and network 3.0.0.0/8. Let's verify:

R1#Traceroute 4.4.4.4 numeric

```

Type escape sequence to abort.
Tracing the route to 4.4.4.4
VRF info: (vrf in name/id, vrf out name/id)
 1 14.1.1.4 0 msec * 0 msec

```

R1#Traceroute 33.3.3.3 numeric

```

Type escape sequence to abort.
Tracing the route to 33.3.3.3
VRF info: (vrf in name/id, vrf out name/id)
 1 14.1.1.4 4 msec 0 msec 0 msec
 2 34.1.1.3 12 msec * 12 msec

```

R1#Traceroute 2.2.2.2 numeric

```

Type escape sequence to abort.
Tracing the route to 2.2.2.2
VRF info: (vrf in name/id, vrf out name/id)
 1 12.1.1.2 32 msec * 12 msec

```

R1#Traceroute 3.3.3.3 numeric

```

Type escape sequence to abort.
Tracing the route to 3.3.3.3
VRF info: (vrf in name/id, vrf out name/id)
 1 12.1.1.2 16 msec 16 msec 16 msec
 2 23.1.1.3 28 msec * 28 msec

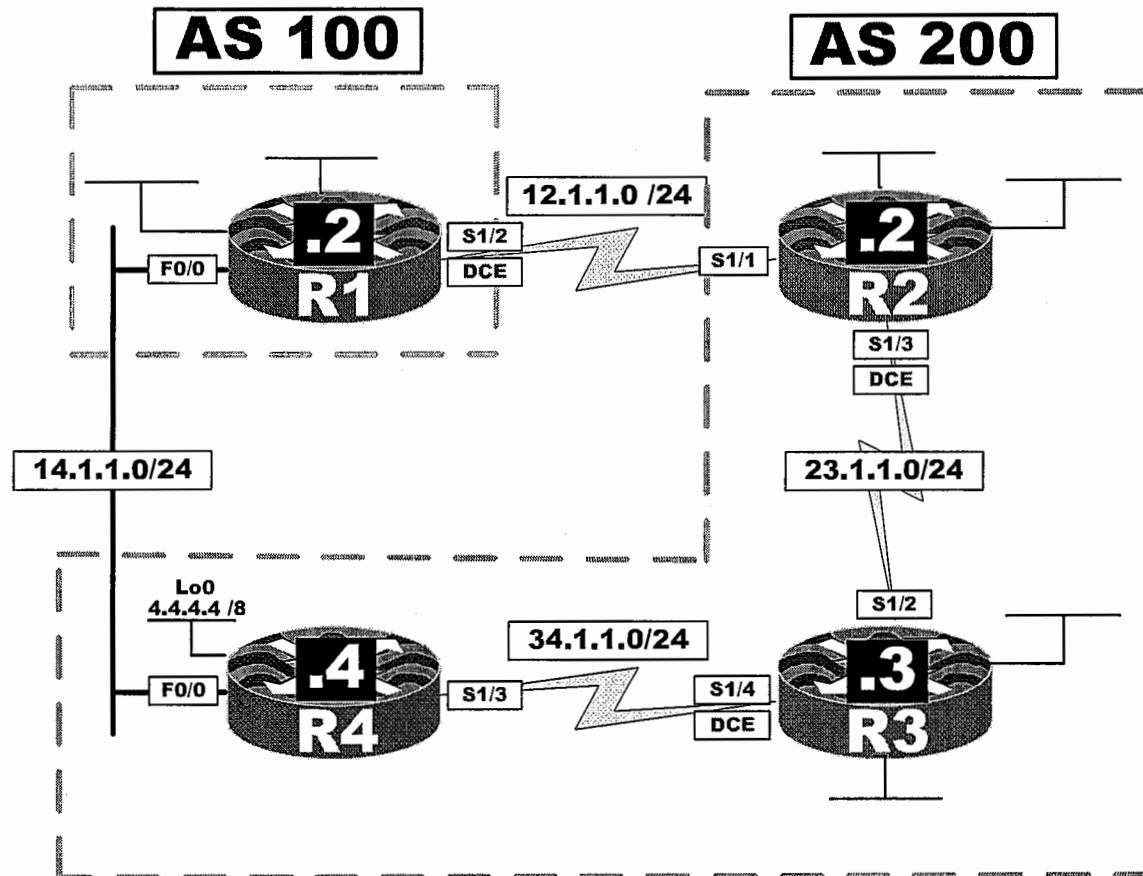
```

Task 6

Erase the startup config and reload the routers before proceeding to the next lab.

Lab 7

Multi Exit Discriminator Attribute



Lab Setup:

To copy and paste the initial configurations, go to "<http://www.micronics.nl>" → "Foundation" → "BGP" → "Lab-7".

Task 1

Configure routers R2, R3 and R4 in AS 200, these routers should have full mesh peer session between them. Routers R2 and router R4 should have EBGP peer session to R1 in AS 100. BGP routers should **only** advertise their loopback interface/s in BGP. Provide NLRI for the links in As 200 using OSPF.

On R1:

```
R1(config-if)#Router bgp 100
R1(config-router)#No au
R1(config-router)#Neighbor 14.1.1.4 remote-as 200
R1(config-router)#Neighbor 12.1.1.2 remote-as 200
R1(config-router)#Netw 11.0.0.0
R1(config-router)#Netw 1.0.0.0
```

On R2:

```
R2(config)#Router bgp 200
R2(config-router)#No au
R2(config-router)#Netw 2.0.0.0
R2(config-router)#Netw 22.0.0.0
R2(config-router)#Neighbor 12.1.1.1 remote-as 100
R2(config-router)#Neighbor 23.1.1.3 remote-as 200
R2(config-router)#Neighbor 23.1.1.3 next-hop-self
R2(config-router)#Neighbor 34.1.1.4 remote-as 200
R2(config-router)#Neighbor 34.1.1.4 next-hop-self

R2(config)#Router ospf 1
R2(config-router)#Router-id 0.0.0.2
R2(config-router)#Netw 23.1.1.2 0.0.0.0 area 0
```

On R3:

```
R3(config)#Router bgp 200
R3(config-router)#No au
R3(config-router)#Netw 3.0.0.0
R3(config-router)#Netw 33.0.0.0
R3(config-router)#Neighbor 34.1.1.4 remote-as 200
R3(config-router)#Neighbor 23.1.1.2 remote-as 200

R3(config)#Router ospf 1
R3(config-router)#Router-id 0.0.0.3
R3(config-router)#Netw 23.1.1.3 0.0.0.0 area 0
```

```
R3(config-router)#Netw 34.1.1.3 0.0.0.0 area 0
```

On R4:

```
R4(config)#Router bgp 200
```

```
R4(config-router)#No au
```

```
R4(config-router)#netw 4.0.0.0
```

```
R4(config-router)#Neighbor 34.1.1.3 remote-as 200
```

```
R4(config-router)#Neighbor 34.1.1.3 next-hop-self
```

```
R4(config-router)#Neighbor 23.1.1.2 remote-as 200
```

```
R4(config-router)#Neighbor 23.1.1.2 next-hop-self
```

```
R4(config-router)#Neighbor 14.1.1.1 remote-as 100
```

```
R4(config)#Router ospf 1
```

```
R4(config-router)#Router-id 0.0.0.4
```

```
R4(config-router)#Netw 34.1.1.4 0.0.0.0 area 0
```

To verify the configuration:

On R1:

```
R1#Sh ip bgp | b Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
* 2.0.0.0	14.1.1.4			0	200 i
*>	12.1.1.2	0		0	200 i
* 3.0.0.0	14.1.1.4			0	200 i
*>	12.1.1.2			0	200 i
* 4.0.0.0	12.1.1.2			0	200 i
*>	14.1.1.4	0		0	200 i
*> 11.0.0.0	0.0.0.0	0		32768	i
* 22.0.0.0	14.1.1.4			0	200 i
*>	12.1.1.2	0		0	200 i
* 33.0.0.0	14.1.1.4			0	200 i
*>	12.1.1.2			0	200 i

On R2:

```
R2#Sh ip bgp | b Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i1.0.0.0	34.1.1.4	0	100	0	100 i
*>	12.1.1.1	0		0	100 i

```

*> 2.0.0.0          0.0.0.0          0          32768 i
*>i3.0.0.0          23.1.1.3          0      100      0 i
*>i4.0.0.0          34.1.1.4          0      100      0 i
* i11.0.0.0         34.1.1.4          0      100      0 100 i
*>                 12.1.1.1          0          0 100 i
*> 22.0.0.0         0.0.0.0          0          32768 i
*>i33.0.0.0         23.1.1.3          0      100      0 i

```

On R3:

R3#Sh ip bgp | b Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i1.0.0.0	34.1.1.4	0	100	0	100 i
* i	23.1.1.2	0	100	0	100 i
*>i2.0.0.0	23.1.1.2	0	100	0	i
*> 3.0.0.0	0.0.0.0	0		32768	i
*>i4.0.0.0	34.1.1.4	0	100	0	i
*>i11.0.0.0	34.1.1.4	0	100	0	100 i
* i	23.1.1.2	0	100	0	100 i
*>i22.0.0.0	23.1.1.2	0	100	0	i
*> 33.0.0.0	0.0.0.0	0		32768	i

On R4:

R4#Sh ip bgp | b Network

Network	Next Hop	Metric	LocPrf	Weight	Path
* i1.0.0.0	23.1.1.2	0	100	0	100 i
*>	14.1.1.1	0		0	100 i
*>i2.0.0.0	23.1.1.2	0	100	0	i
*>i3.0.0.0	34.1.1.3	0	100	0	i
*> 4.0.0.0	0.0.0.0	0		32768	i
* i11.0.0.0	23.1.1.2	0	100	0	100 i
*>	14.1.1.1	0		0	100 i
*>i22.0.0.0	23.1.1.2	0	100	0	i
*>i33.0.0.0	34.1.1.3	0	100	0	i

Task 2

Configure AS 200 such that R1 in AS 100 takes R4 to reach any prefix advertised in AS 200.
Manipulate MED to accomplish this task.

MED is used as a suggestion to an external AS regarding the preferred path into the AS that is advertising the metric. The reason suggestion is used here is because the AS that is receiving MED attribute could use another attribute such as "weight" that will override the MED attribute. As far as MED is concerned, the lower value has better preference.

On R2:

```
R2(config)#Route-map TST permit 10
R2(config-route-map)#set metric 100

R2(config)#Router bgp 200
R2(config-router)#Neighbor 12.1.1.1 route-map TST out

R2#Clear ip bgp * Out
```

To verify the configuration:

On R1:

```
R1#Sh ip bgp | b Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	14.1.1.4			0	200 i
*	12.1.1.2	100		0	200 i
*> 3.0.0.0	14.1.1.4			0	200 i
*	12.1.1.2	100		0	200 i
* 4.0.0.0	12.1.1.2	100		0	200 i
*>	14.1.1.4	0		0	200 i
*> 11.0.0.0	0.0.0.0	0		32768	i
*> 22.0.0.0	14.1.1.4			0	200 i
*	12.1.1.2	100		0	200 i
*> 33.0.0.0	14.1.1.4			0	200 i
*	12.1.1.2	100		0	200 i

```
R1#Traceroute 2.2.2.2 source 1.1.1.1
```

```
Type escape sequence to abort.
Tracing the route to 2.2.2.2
VRF info: (vrf in name/id, vrf out name/id)
 1 14.1.1.4 0 msec 4 msec 0 msec
 2 34.1.1.3 16 msec 16 msec 16 msec
 3 23.1.1.2 20 msec * 20 msec
```

Note: R1 receives a MED value of 100 for all prefixes from R2, since R1 does not receive a MED value from R4; R1 considers the MED value coming from R4 as zero. When R1 compares the two MED values, it takes the path with a lower MED value, which is R4.

Task 3

Remove the configuration command from the previous task before proceeding to the next task.

On R2:

```
R2(config)#No route-map TST

R2(config)#Router bgp 200
R2(config-router)#No Neighbor 12.1.1.1 route-map TST out

R2#Clear ip bgp *
```

Task 4

Configure the appropriate router/s in AS 200 such that AS 100 goes through R4 to reach network 33.0.0.0 /8 and R2 to reach network 3.0.0.0 /8. Utilize MED to accomplish this task.

The following output shows the existing BGP table of R1 before manipulating the MED attribute:

On R1:

```
R1#Sh ip bgp | b Network
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	1.0.0.0	0.0.0.0	0		32768	i
*	2.0.0.0	14.1.1.4			0	200 i
*>		12.1.1.2	0		0	200 i
*	3.0.0.0	12.1.1.2			0	200 i
*>		14.1.1.4			0	200 i
*	4.0.0.0	12.1.1.2			0	200 i


```

*>                14.1.1.4                0                0 200 i
*> 11.0.0.0        0.0.0.0                0                32768 i
* 22.0.0.0        14.1.1.4                0 200 i
*>                12.1.1.2                0 200 i
* 33.0.0.0        12.1.1.2                0 200 i
*>                14.1.1.4                0 200 i

```

Note: Both networks are reachable via R4

On R2:

```

R2(config)#IP Prefix-list NET3 permit 3.0.0.0/8
R2(config)#IP Prefix-list NET33 permit 33.0.0.0/8

```

Note: It is very easy to remember that Prefix-list "NET3" is referencing network 3.0.0.0/8 and Prefix-list "NET33" is referencing network 33.0.0.0/8. If possible, you should choose a meaningful name.

On R2, the following configuration sets the "metric" which is the MED value to 50 for network 3.0.0.0/8 and 100 for network 33.0.0.0/8, whereas, on R4, the MED values are reversed, meaning that the MED value for network 3.0.0.0/8 is set to 100 and for network 33.0.0.0/8 it is set to 50.

```

R2(config)#Route-map TST permit 10
R2(config-route-map)#match ip addr prefix NET3
R2(config-route-map)#set metric 50

R2(config)#Route-map TST permit 20
R2(config-route-map)#match ip addr prefix NET33
R2(config-route-map)#set metric 100

R2(config)#Route-map TST permit 90

R2(config)#Router bgp 200
R2(config-router)#Neighbor 10.1.12.1 route-map TST out

R2#Clear ip bgp *

```

On R4:

```

R4(config)#IP Prefix-list NET3 permit 3.0.0.0/8
R4(config)#IP Prefix-list NET33 permit 33.0.0.0/8

```

```

R4(config)#Route-map TST permit 10
R4(config-route-map)#match ip addr prefix NET3
R4(config-route-map)#set metric 100

R4(config)#Route-map TST permit 20
R4(config-route-map)#match ip addr prefix NET33
R4(config-route-map)#set metric 50

R4(config)#Route-map TST permit 90

R4(config)#Router bgp 200
R4(config-router)#Neighbor 10.1.14.1 route-map TST out

R4#Clear ip bgp *

```

To verify the configuration:

On R1:

```
R1#Show ip bgp | b Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
* 2.0.0.0	14.1.1.4			0	200 i
*> 12.1.1.2	12.1.1.2	0		0	200 i
* 3.0.0.0	14.1.1.4	100		0	200 i
*> 12.1.1.2	12.1.1.2	50		0	200 i
* 4.0.0.0	12.1.1.2			0	200 i
*> 14.1.1.4	14.1.1.4	0		0	200 i
*> 11.0.0.0	0.0.0.0	0		32768	i
* 22.0.0.0	14.1.1.4			0	200 i
*> 12.1.1.2	12.1.1.2	0		0	200 i
*> 33.0.0.0	14.1.1.4	50		0	200 i
* 12.1.1.2	12.1.1.2	100		0	200 i

Note: Network 3.0.0.0/8 is reachable via R2 and network 33.0.0.0/8 is reachable via R4

To test the configuration:

On R1:

```
R1#Traceroute 3.3.3.3 source 1.1.1.1
```

Type escape sequence to abort.

```
Tracing the route to 3.3.3.3
VRF info: (vrf in name/id, vrf out name/id)
  1 12.1.1.2 16 msec 16 msec 12 msec \
  2 23.1.1.3 24 msec * 20 msec
```

```
R1#Traceroute 33.3.3.3 source 1.1.1.1
```

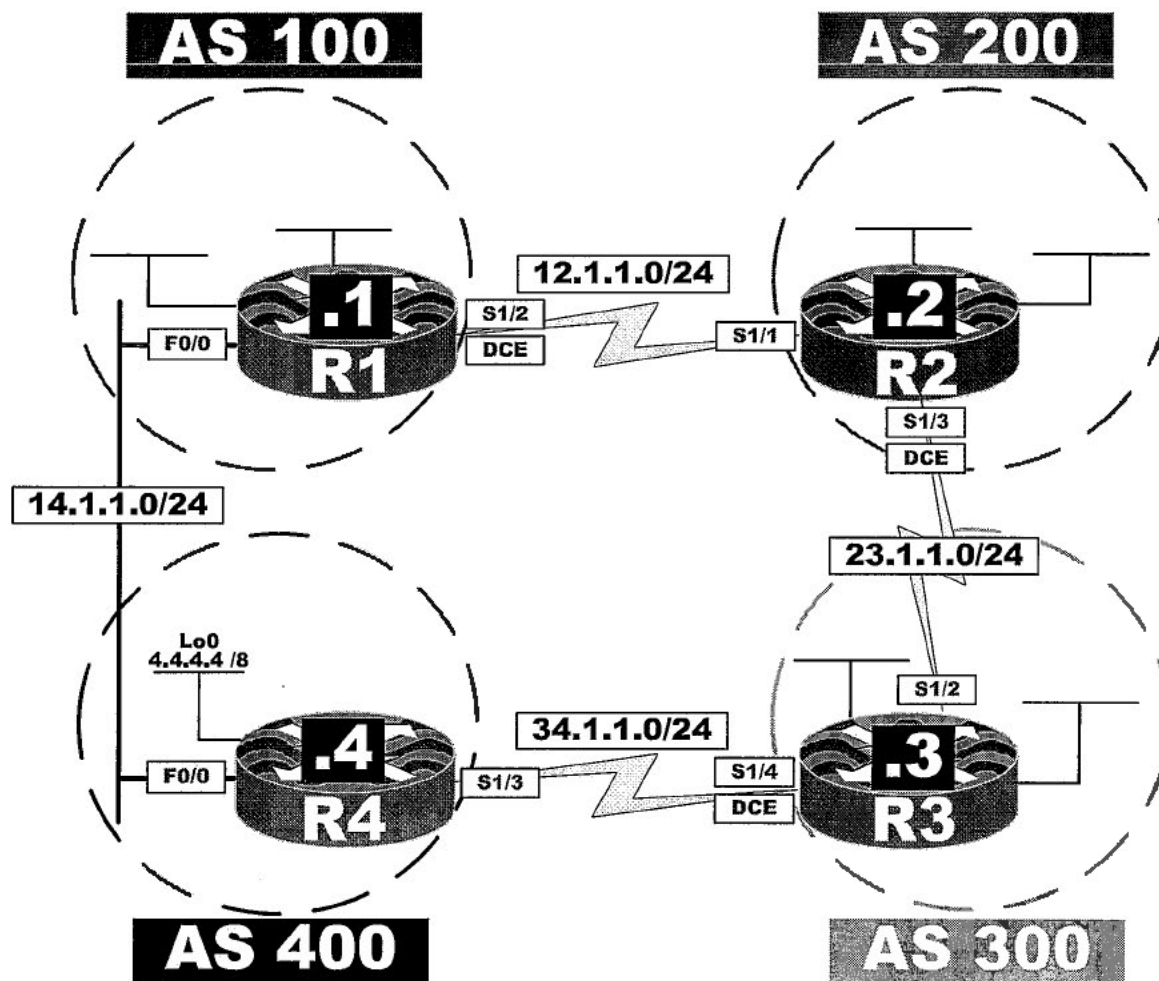
Type escape sequence to abort.

```
Tracing the route to 33.3.3.3
VRF info: (vrf in name/id, vrf out name/id)
  1 14.1.1.4 4 msec 0 msec 0 msec
  2 34.1.1.3 12 msec * 12 msec
```

Task 5

Remove BGP routing protocol and any previous BGP related command/s from the previous tasks and reconfigure them based on the chart below. These routers should advertise their loopback interface/s in their own AS. The BGP peering should be established as follows:

R1 should establish EBGP peer sessions with R2 and R4 in AS 200 and 400 respectively.
R2 should establish EBGP peer sessions with R1 and R3 in AS 100 and 300 respectively.
R3 should establish EBGP peer sessions with R2 and R4 in AS 200 and 400 respectively.
R4 should establish EBGP peer sessions with R3 and R1 in AS 300 and 100 respectively.



On R1:

```
R1(config)#No router bgp 100

R1(config)#Router bgp 100
R1(config-router)#No au
R1(config-router)#Netw 1.0.0.0
R1(config-router)#Netw 11.0.0.0
R1(config-router)#Neighbor 12.1.1.2 remote-as 200
R1(config-router)#Neighbor 14.1.1.4 remote-as 400
```

On R2:

```
R2(config)#No Router bgp 200
R2(config)#No IP Prefix-list NET3
```

```

R2(config)#No IP Prefix-list NET33
R2(config)#No Route-map TST

R2(config)#Router bgp 200
R2(config-router)#Netw 2.0.0.0
R2(config-router)#Netw 22.0.0.0
R2(config-router)#No au
R2(config-router)#Neighbor 12.1.1.1 remote-as 100
R2(config-router)#Neighbor 23.1.1.3 remote-as 300

```

On R3:

```

R3(config)#No Router bgp 200

R3(config)#Router bgp 300
R3(config-router)#No au
R3(config-router)#Netw 3.0.0.0
R3(config-router)#Netw 33.0.0.0
R3(config-router)#Neighbor 23.1.1.2 remote-as 200
R3(config-router)#Neighbor 34.1.1.4 remote-as 400

```

On R4:

```

R4(config)#No Router bgp 200
R4(config)#No IP Prefix-list NET3
R4(config)#No IP Prefix-list NET33
R4(config)#No Route-map TST

R4(config)#Router bgp 400
R4(config-router)#No au
R4(config-router)#Netw 4.0.0.0
R4(config-router)#Neighbor 14.1.1.1 remote-as 100
R4(config-router)#Neighbor 34.1.1.3 remote-as 300

```

To verify the configuration:

On R1:

```
R1#Show ip bgp | b Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
* 2.0.0.0	14.1.1.4			0	400 300 200 i
*>	12.1.1.2	0		0	200 i

```

*> 3.0.0.0          14.1.1.4          0 400 300 i
*                  12.1.1.2          0 200 300 i
* 4.0.0.0          12.1.1.2          0 200 300 400 i
*>                  14.1.1.4          0          0 400 i
*> 11.0.0.0         0.0.0.0          0          32768 i
* 22.0.0.0         14.1.1.4          0 400 300 200 i
*>                  12.1.1.2          0          0 200 i
*> 33.0.0.0         14.1.1.4          0 400 300 i
*                  12.1.1.2          0 200 300 i

```

Task 6

Configure R4 in AS 400 to pass a MED value of 100 to R1 in AS 100 and configure R2 in AS 200 to pass a MED of 120 to R1 in AS 100.

On R2:

```

R2(config)#Route-map TST permit 10
R2(config-route-map)#set metric 120

R2(config)#Router bgp 200
R2(config-router)#Neighbor 12.1.1.1 route-map TST out

R2#Clear ip bgp *

```

On R4:

```

R4(config)#Route-map TST permit 10
R4(config-route-map)#set metric 100

R4(config)#Router bgp 400
R4(config-router)#Neighbor 14.1.1.1 route-map TST out

R4#Clear ip bgp * out

```

To verify the configuration:

On R1:

```

R1#Clear ip bgp *

R1#Show ip bgp | b Network

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	120		0	200 i
* 3.0.0.0	12.1.1.2	120		0	200 300 i
*> 14.1.1.4		100		0	400 300 i
* 4.0.0.0	12.1.1.2	120		0	200 300 400 i
*> 14.1.1.4		100		0	400 i
*> 11.0.0.0	0.0.0.0	0		32768	i
*> 22.0.0.0	12.1.1.2	120		0	200 i
* 33.0.0.0	12.1.1.2	120		0	200 300 i
*> 14.1.1.4		100		0	400 300 i

Task 7

Ensure that R1 in AS 100 always takes R4 to reach all prefixes advertised in this topology. You should configure R1 to accomplish this task.

Note: R1 goes through R2 to reach network 22.0.0.0 /8 even though the MED value is lower through R4. The reason this happens is because the AS-PATH is longer through R4, and we know that the AS-Path is considered before MED.

There are two challenges in resolving this task:

The networks are being advertised by both ASes and by default the "MED" attribute is only considered among paths from the same AS.

AS-Path attribute is considered before MED.

To resolve the above challenges, we can change the default behavior of MED such that the local router looks at the MED attribute coming from two ASes, in this case AS 200 and AS 400; then, we MUST instruct the local router (R1) to ignore the AS-Path attribute when performing best path determination.

On R1:

```
R1(config)#Router bgp 100

R1(config-router)#Bgp always-compare-med
R1(config-router)#Bgp bestpath as-path ignore

R1#Clear ip bgp *
```

Note: The tab key will not work when entering the "bgp bestpath as-path ignore" command. This

command is a hidden one.

The MED as stated in RFC 1771, is an optional non-transitive attribute. The value of this attribute may be used by BGP speakers for best path selection process to discriminate among multiple exit points to a neighboring AS. The lower the value the better the path. The MED value comparison is done only among paths from the same AS. The "bgp always-compare-med" command is used to change this behavior by enforcing MED comparison between all paths, regardless of the AS from which the paths are received.

Note: The second command "bgp bestpath as-path ignore" is also needed as part of the solution in accomplishing this task, this command is needed because as-path and the origin attributes are looked at before the MED attribute. Since the origin attribute is identical for all routes and BGP routing protocol is told to ignore the as-path attribute in best path calculation, BGP will look at the MED value next.

To verify the configuration:

On R1:

R1#Sh ip bgp | b Network

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	1.0.0.0	0.0.0.0	0		32768	i
*	2.0.0.0	12.1.1.2	120		0	200 i
*>		14.1.1.4	100		0	400 300 200 i
*	3.0.0.0	12.1.1.2	120		0	200 300 i
*>		14.1.1.4	100		0	400 300 i
*	4.0.0.0	12.1.1.2	120		0	200 300 400 i
*>		14.1.1.4	100		0	400 i
*>	11.0.0.0	0.0.0.0	0		32768	i
*	22.0.0.0	12.1.1.2	120		0	200 i
*>		14.1.1.4	100		0	400 300 200 i
*	33.0.0.0	12.1.1.2	120		0	200 300 i
*>		14.1.1.4	100		0	400 300 i

Note: R1 takes R4 to reach all the networks in this topology

Task 8

Remove the BGP configuration performed in Task 6. Configure R4 to pass a MED value of 100 to R1; R2 in AS 200 should **NOT** pass any MED values to R1.

Ensure that R1 in AS 100 takes R4 to reach any prefix advertised in this topology except the ones originated by R2. **Do not** use any global config command/s as part of the solution to this task.

To remove the configuration performed in Task 6 and verify the configuration:

On R2:

```
R2(config)#Router bgp 200
R2(config-router)#No Neighbor 12.1.1.1 route-map TST out

R2(config)#No Route-map TST

R2#Clear ip bgp *
```

On R4:

```
R4(config)#Router bgp 400
R4(config-router)#No Neighbor 14.1.1.1 route-map TST out

R4(config)#No Route-map TST

R4#Clear ip bgp *
```

Let's configure R4 to pass a MED value of 100 to R1:

On R4:

```
R4(config)#Route-map TST per 10
R4(config-route-map)#set metric 100

R4(config)#Router bgp 400
R4(config-router)#Neigh 14.1.1.1 route-map TST out

R4#Clear ip bgp *
```

On R1:

```
R1#Sh ip bgp | b Network
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	1.0.0.0	0.0.0.0	0		32768	i
*>	2.0.0.0	12.1.1.2	0		0 200	i
*	3.0.0.0	14.1.1.4	100		0 400 300	i
*>		12.1.1.2			0 200 300	i

```

*> 4.0.0.0          12.1.1.2          0 200 300 400 i
*                   14.1.1.4          100      0 400 i
*> 11.0.0.0         0.0.0.0          0      32768 i
*> 22.0.0.0         12.1.1.2          0      0 200 i
* 33.0.0.0         14.1.1.4          100      0 400 300 i
*>                  12.1.1.2          0 200 300 i

```

Since R2 is not giving a MED value to R1; R1 consider the MED value coming from R2 as zero, therefore, R1 takes the path through R2 (The path with the lowest MED value). This behavior can be changed by instructing R1 to set the path with the missing MED value to the worst possible value.

On R1:

```

R1(config)#Router bgp 100
R1(config-router)#bgp bestpath med missing-as-worst

```

To verify the configuration:

On R1:

```

R1#Sh ip bgp | b Network

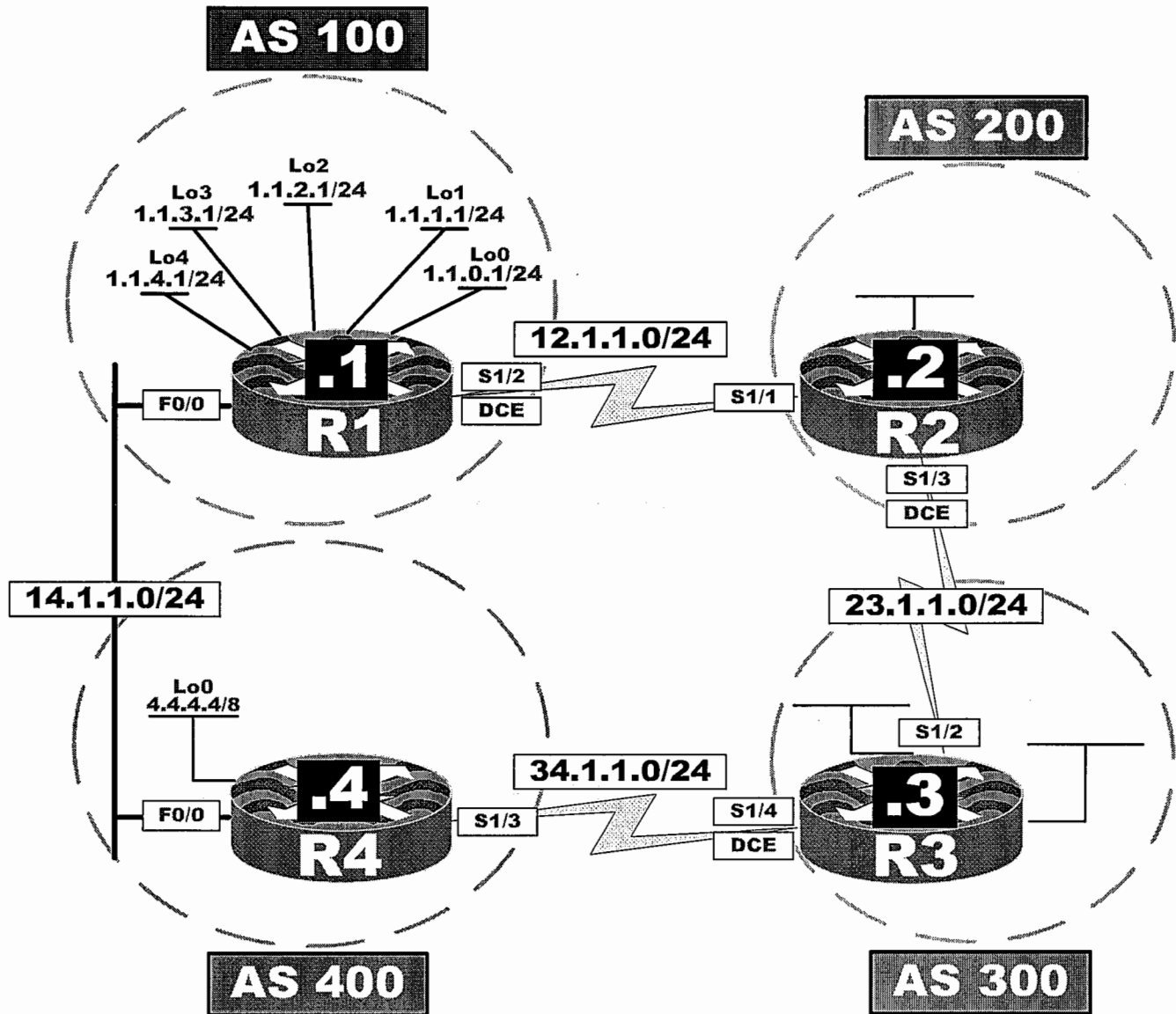
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0	200 i
*> 3.0.0.0	14.1.1.4	100		0	400 300 i
* 12.1.1.2	12.1.1.2	4294967295		0	200 300 i
* 4.0.0.0	12.1.1.2	4294967295		0	200 300 400 i
*> 14.1.1.4	14.1.1.4	100		0	400 i
*> 11.0.0.0	0.0.0.0	0		32768	i
*> 22.0.0.0	12.1.1.2	0		0	200 i
*> 33.0.0.0	14.1.1.4	100		0	400 300 i
* 12.1.1.2	12.1.1.2	4294967295		0	200 300 i

Task 9

Erase the startup config and reload the routers before proceeding to the next lab.

Lab 8 - Filtering Using Access-lists and Prefix-lists



Task 1

Configure the routers and the switches based on the above diagram. DO NOT configure BGP.

On SW1:

```
SW1(config)#Int Range f0/1,f0/4
SW1(config-if-range)#Swt mode acc
SW1(config-if-range)#Swt acc vlan 200
SW1(config-if-range)#No shut
```

On R1:

```
R1(config)#Int S1/2
R1(config-if)#IP address 12.1.1.1 255.255.255.0
R1(config-if)#Clock rate 64000
R1(config-if)#No shut
```

```
R1(config)#Int F0/0
R1(config-if)#IP addr 14.1.1.1 255.255.255.0
R1(config-if)#No shut
```

```
R1(config)#Int Lo0
R1(config-if)#IP address 1.1.0.1 255.255.255.0
```

```
R1(config)#Int Lo1
R1(config-if)#IP address 1.1.1.1 255.255.255.0
```

```
R1(config)#Int Lo2
R1(config-if)#IP address 1.1.2.1 255.255.255.0
```

```
R1(config)#Int Lo3
R1(config-if)#IP address 1.1.3.1 255.255.255.0
```

```
R1(config)#Int Lo4
R1(config-if)#IP address 1.1.4.1 255.255.255.0
```

On R2:

```
R2(config)#Int S1/1
R2(config-if)#IP address 12.1.1.2 255.255.255.0
R2(config-if)#No shut
```

```
R2(config)#Int S1/3
R2(config-if)#IP address 23.1.1.2 255.255.255.0
R2(config-if)#Clock rate 64000
R2(config-if)#No shut
```

```
R2(config)#Int Lo0
R2(config-if)#IP addr 2.2.2.2 255.0.0.0
```

```
R2(config)#Int Lo1
R2(config-if)#IP addr 22.2.2.2 255.0.0.0
```

On R3:

```
R3(config)#Int S1/4
R3(config-if)#IP address 34.1.1.3 255.255.255.0
R3(config-if)#Clock rate 64000
R3(config-if)#No shut
```

```
R3(config)#Int S1/2
R3(config-if)#IP address 23.1.1.3 255.255.255.0
R3(config-if)#No shut
```

```
R3(config)#Int Lo0
R3(config-if)#Ip addr 3.3.3.3 255.0.0.0
```

```
R3(config)#Int Lo1
R3(config-if)#Ip addr 33.3.3.3 255.0.0.0
```

On R4:

```
R4(config)#Int S1/3
R4(config-if)#IP addr 34.1.1.4 255.255.255.0
R4(config-if)#No shut
```

```
R4(config)#Int F0/0
R4(config-if)#IP addr 14.1.1.4 255.255.255.0
R4(config-if)#No shut
```

```
R4(config)#Int Lo0
R4(config-if)#IP addr 4.4.4.4 255.0.0.0
```

To verify the configuration:

On R2:

```
R2#Ping 12.1.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.2, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

R2#Ping 23.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 23.1.1.3, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

On R4:

R4#Ping 14.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.1, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R4#Ping 34.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 34.1.1.3, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms

Task 2

Configure the BGP peering based on the following policy:

R1 should establish EBGP peer sessions with R2 in AS 200, and R4 in AS 400.

R2 should establish EBGP peer sessions with R1 in AS 100, and R3 in AS 300.

R3 should establish EBGP peer sessions with R2 in AS 200, and R4 in AS 400.

R4 should establish EBGP peer sessions with R3 in AS 300, and R1 in AS 100.

On R1:

R1(config)#router bgp 100

R1(config-router)#No au

R1(config-router)#netw 1.1.0.0 mask 255.255.255.0

R1(config-router)#netw 1.1.1.0 mask 255.255.255.0

R1(config-router)#netw 1.1.2.0 mask 255.255.255.0

R1(config-router)#netw 1.1.3.0 mask 255.255.255.0

R1(config-router)#netw 1.1.4.0 mask 255.255.255.0

```
R1(config-router)#neighbor 12.1.1.2 remote-as 200
R1(config-router)#neighbor 14.1.1.4 remote-as 400
```

On R2:

```
R2(config)#router bgp 200
R2(config-router)#No au
R2(config-router)#netw 2.0.0.0
R2(config-router)#netw 22.0.0.0
R2(config-router)#neighbor 12.1.1.1 remote-as 100
R2(config-router)#neighbor 23.1.1.3 remote-as 300
```

On R3:

```
R3(config)#router bgp 300
R3(config-router)#No au
R3(config-router)#netw 3.0.0.0
R3(config-router)#netw 33.0.0.0
R3(config-router)#neighbor 23.1.1.2 remote-as 200
R3(config-router)#neighbor 34.1.1.4 remote-as 400
```

On R4:

```
R4(config)#router bgp 400
R4(config-router)#No au
R4(config-router)#netw 4.0.0.0
R4(config-router)#neighbor 14.1.1.1 remote-as 100
R4(config-router)#neighbor 34.1.1.3 remote-as 300
```

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.1.0.0/24	0.0.0.0	0		32768	i
*> 1.1.1.0/24	0.0.0.0	0		32768	i
*> 1.1.2.0/24	0.0.0.0	0		32768	i
*> 1.1.3.0/24	0.0.0.0	0		32768	i
*> 1.1.4.0/24	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0	200 i
* 3.0.0.0	14.1.1.4			0	400 300 i

```

*>          12.1.1.2          0 200 300 i
*> 4.0.0.0    14.1.1.4          0 0 400 i
*> 22.0.0.0    12.1.1.2          0 0 200 i
* 33.0.0.0    14.1.1.4          0 400 300 i
*>          12.1.1.2          0 200 300 i

```

On R2:

R2#Sh ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.1.0.0/24	12.1.1.1	0		0 100	i
*> 1.1.1.0/24	12.1.1.1	0		0 100	i
*> 1.1.2.0/24	12.1.1.1	0		0 100	i
*> 1.1.3.0/24	12.1.1.1	0		0 100	i
*> 1.1.4.0/24	12.1.1.1	0		0 100	i
*> 2.0.0.0	0.0.0.0	0		32768	i
*> 3.0.0.0	23.1.1.3	0		0 300	i
* 4.0.0.0	23.1.1.3			0 300 400	i
*>	12.1.1.1			0 100 400	i
*> 22.0.0.0	0.0.0.0	0		32768	i
*> 33.0.0.0	23.1.1.3	0		0 300	i

On R3:

R3#Sh ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
* 1.1.0.0/24	34.1.1.4			0 400 100	i
*>	23.1.1.2			0 200 100	i
* 1.1.1.0/24	34.1.1.4			0 400 100	i
*>	23.1.1.2			0 200 100	i
* 1.1.2.0/24	34.1.1.4			0 400 100	i
*>	23.1.1.2			0 200 100	i
* 1.1.3.0/24	34.1.1.4			0 400 100	i
*>	23.1.1.2			0 200 100	i
* 1.1.4.0/24	34.1.1.4			0 400 100	i
*>	23.1.1.2			0 200 100	i
* 2.0.0.0	34.1.1.4			0 400 100 200	i
*>	23.1.1.2	0		0 200	i
*> 3.0.0.0	0.0.0.0	0		32768	i
*> 4.0.0.0	34.1.1.4	0		0 400	i
*	23.1.1.2			0 200 100 400	i
* 22.0.0.0	34.1.1.4			0 400 100 200	i
*>	23.1.1.2	0		0 200	i


```
*> 33.0.0.0          0.0.0.0          0          32768 i
```

On R4:

```
R4#Sh ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* 1.1.0.0/24	34.1.1.3			0	300 200 100 i
*>	14.1.1.1	0		0	100 i
* 1.1.1.0/24	34.1.1.3			0	300 200 100 i
*>	14.1.1.1	0		0	100 i
* 1.1.2.0/24	34.1.1.3			0	300 200 100 i
*>	14.1.1.1	0		0	100 i
* 1.1.3.0/24	34.1.1.3			0	300 200 100 i
*>	14.1.1.1	0		0	100 i
* 1.1.4.0/24	34.1.1.3			0	300 200 100 i
*>	14.1.1.1	0		0	100 i
* 2.0.0.0	34.1.1.3			0	300 200 i
*>	14.1.1.1			0	100 200 i
*> 3.0.0.0	34.1.1.3	0		0	300 i
*	14.1.1.1			0	100 200 300 i
*> 4.0.0.0	0.0.0.0	0		32768	i
* 22.0.0.0	34.1.1.3			0	300 200 i
*>	14.1.1.1			0	100 200 i
*> 33.0.0.0	34.1.1.3	0		0	300 i
*	14.1.1.1			0	100 200 300 i

Task 3

Configure R2 to filter network 1.1.4.0 /24. Use distribute-list and access-list to accomplish this task.

Let's see the existing entry for this network:

On R2:

```
R2#Show ip bgp 1.1.4.0 255.255.255.0
```

BGP routing table entry for 1.1.4.0/24, version 6

Paths: (1 available, best #1, table default)

Advertised to update-groups:

1
100

```
12.1.1.1 from 12.1.1.1 (1.1.4.1)
  Origin IGP, metric 0, localpref 100, valid, external, best
```

To filter the prefix, an access-list is configured to deny the prefix that needs to be filtered and permit everything else:

```
R2(config)#access-list 4 deny 1.1.4.0 0.0.0.255
R2(config)#access-list 4 permit any
```

Two distribute-lists are configured to filter the prefix inbound from both neighbors:

```
R2(config)#router bgp 200
R2(config-router)#neighbor 12.1.1.1 distribute-list 4 in
R2(config-router)#neighbor 23.1.1.3 distribute-list 4 in
```

The tricky part was to understand the topology, if the topology is not understood, silly mistakes can occur which can cost points in the CCIE lab. The tricky part of this task is to block the prefix from both neighbors.

To verify the configuration:

On R2:

```
R2#Clear ip bgp * in
R2#Sh ip route 1.1.4.0
% Subnet not in table
R2#Show ip bgp 1.1.4.0
% Network not in table
```

You can see that the network is filtered.

Task 4

Remove the configuration command from previous task, and accomplish the same task using prefix-list and distribute-list.

On R2:

```
R2(config)#No access-list 4
```

```
R2(config)#router bgp 200
```

```
R2(config-router)#No neighbor 10.1.12.1 distribute-list 4 in
```

```
R2(config-router)#No neighbor 10.1.23.3 distribute-list 4 in
```

```
R2#Clear ip bgp * in
```

To verify the configuration:

On R2:

```
R2#Sh ip bgp 1.1.4.0
```

```
BGP routing table entry for 1.1.4.0/24, version 13
```

```
Paths: (2 available, best #2, table default)
```

```
Advertised to update-groups:
```

```
1
```

```
300 400 100
```

```
23.1.1.3 from 23.1.1.3 (33.3.3.3)
```

```
Origin IGP, localpref 100, valid, external
```

```
100
```

```
12.1.1.1 from 12.1.1.1 (1.1.4.1)
```

```
Origin IGP, metric 0, localpref 100, valid, external, best
```

```
R2#Show ip route 1.1.4.0
```

```
Routing entry for 1.1.4.0/24
```

```
Known via "bgp 200", distance 20, metric 0
```

```
Tag 100, type external
```

```
Last update from 12.1.1.1 00:00:57 ago
```

```
Routing Descriptor Blocks:
```

```
* 12.1.1.1, from 12.1.1.1, 00:00:57 ago
```

```
Route metric is 0, traffic share count is 1
```

```
AS Hops 1
```

```
Route tag 100
```

```
MPLS label: none
```

```
R2(config)#ip prefix-list TST seq 5 deny 1.1.4.0/24
```

```
R2(config)#ip prefix-list TST seq 10 permit 0.0.0.0/0 le 32
```

```
R2(config)#router bgp 200
```

```
R2(config-router)#neighbor 10.1.12.1 prefix-list TST in
```

```
R2(config-router)#neighbor 10.1.23.3 prefix-list TST in
```

```
R2#Clear ip bgp * in
```

Note there are multiple ways to accomplish a given task, understanding and remembering the different ways can be the key to success.

To verify the configuration:

On R2:

```
R2#Show ip bgp 1.1.4.0
```

```
% Network not in table
```

```
R2#Show ip route 1.1.4.0
```

```
% Subnet not in table
```

Task 5

Configure R3 in AS 300 to filter network 22.0.0.0 /8. DO NOT use distribute-list or prefix-list. A route-map and an access-list should be used to accomplish this task.

Let's verify the existing entry in BGP table:

On R3:

```
R3#Show ip bgp 22.0.0.0
```

```
BGP routing table entry for 22.0.0.0/8, version 8
```

```
Paths: (2 available, best #2, table default)
```

```
Advertised to update-groups:
```

```
1
```

```
400 100 200
```

```
34.1.1.4 from 34.1.1.4 (4.4.4.4)
```

```
Origin IGP, localpref 100, valid, external
```

```
200
```

```
23.1.1.2 from 23.1.1.2 (22.2.2.2)
```

```
Origin IGP, metric 0, localpref 100, valid, external, best
```

We can see that the local router is learning the prefix (22.0.0.0/8) from two neighbors, R2 and R4.

To filter the prefix, an access-list is configured to deny the prefix that needs to be filtered, and allow the rest:

```
R3(config)#access-list 22 deny 22.0.0.0 0.255.255.255
R3(config)#access-list 22 permit any
```

A route-map is configured to reference the access-list:

```
R3(config)#route-map TST permit 10
R3(config-route-map)#match ip addr 22
```

A neighbor route-map command is used for each neighbor:

```
R3(config)#router bgp 300
R3(config-router)#neighbor 23.1.1.2 route-map TST in
R3(config-router)#neighbor 34.1.1.4 route-map TST in

R3#Clear ip bgp * in
```

To verify the configuration:

On R3:

```
R3#Show ip route 22.0.0.0
% Network not in table

R3#Show ip bgp 22.0.0.0
% Network not in table
```

Task 6

Remove the Access-list and the route-map configuration from the previous task. Use minimum number of lines in the access-list to filter network 22.0.0.0 /8. You should use an access-list and a route-map to accomplish this task.

On R3:

```
R3(config)#No access-list 22
R3(config)#No route-map TST

R3(config)#access-list 22 permit 22.0.0.0 0.255.255.255

R3(config)#route-map TST deny 10
R3(config-route-map)#match ip addr 22
```

```
R3(config)#route-map TST permit 90
```

Remember that the neighbor command is already configured:

To verify the configuration:

On R3:

```
R3#Clear ip bgp * in
```

```
R3#Show ip route 22.0.0.0  
% Network not in table
```

```
R3#Show ip bgp 22.0.0.0  
% Network not in table
```

When we are asked to configure an access-list with minimum number of lines, we should always see if the task can be accomplished using a single statement in the access-list, sometimes a single line can accomplish the task and sometimes more lines are required in the access-list.

Task 7

R4 should NOT see networks 1.1.0.0 /24 – 1.1.4.0 /24 in its BGP or routing table; you should configure a prefix-list to accomplish this task. DO NOT configure R4, or a route-map.

Let's verify the existing BGP table of R4 for the stated prefixes:

```
R4#Show ip bgp regexp ^100$
```

```
BGP table version is 11, local router ID is 4.4.4.4  
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
               r RIB-failure, S Stale, m multipath, b backup-path, x best-  
external, f RT-Filter
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.1.0.0/24	14.1.1.1	0		0 100	i
*> 1.1.1.0/24	14.1.1.1	0		0 100	i
*> 1.1.2.0/24	14.1.1.1	0		0 100	i
*> 1.1.3.0/24	14.1.1.1	0		0 100	i
*> 1.1.4.0/24	14.1.1.1	0		0 100	i

Since R4 can NOT be configured to accomplish this task, R4's neighbors are configured to filter the prefixes outbound toward R4:

On R1 and R3:

```
Rx(config)#ip prefix-list TST seq 5 deny 1.1.0.0/24
Rx(config)#ip prefix-list TST seq 10 deny 1.1.1.0/24
Rx(config)#ip prefix-list TST seq 15 deny 1.1.2.0/24
Rx(config)#ip prefix-list TST seq 20 deny 1.1.3.0/24
Rx(config)#ip prefix-list TST seq 25 deny 1.1.4.0/24
Rx(config)#ip prefix-list TST seq 30 permit 0.0.0.0/0 le 32
```

On R1:

```
R1(config)#router bgp 100
R1(config-router)#neigh 14.1.1.4 prefix-list TST Out

R1#Clear ip bgp * out
```

On R3:

```
R3(config)#router bgp 100
R3(config-router)#neigh 34.1.1.4 prefix-list TST Out

R3#Clear ip bgp * out
```

To verify the configuration:

On R4:

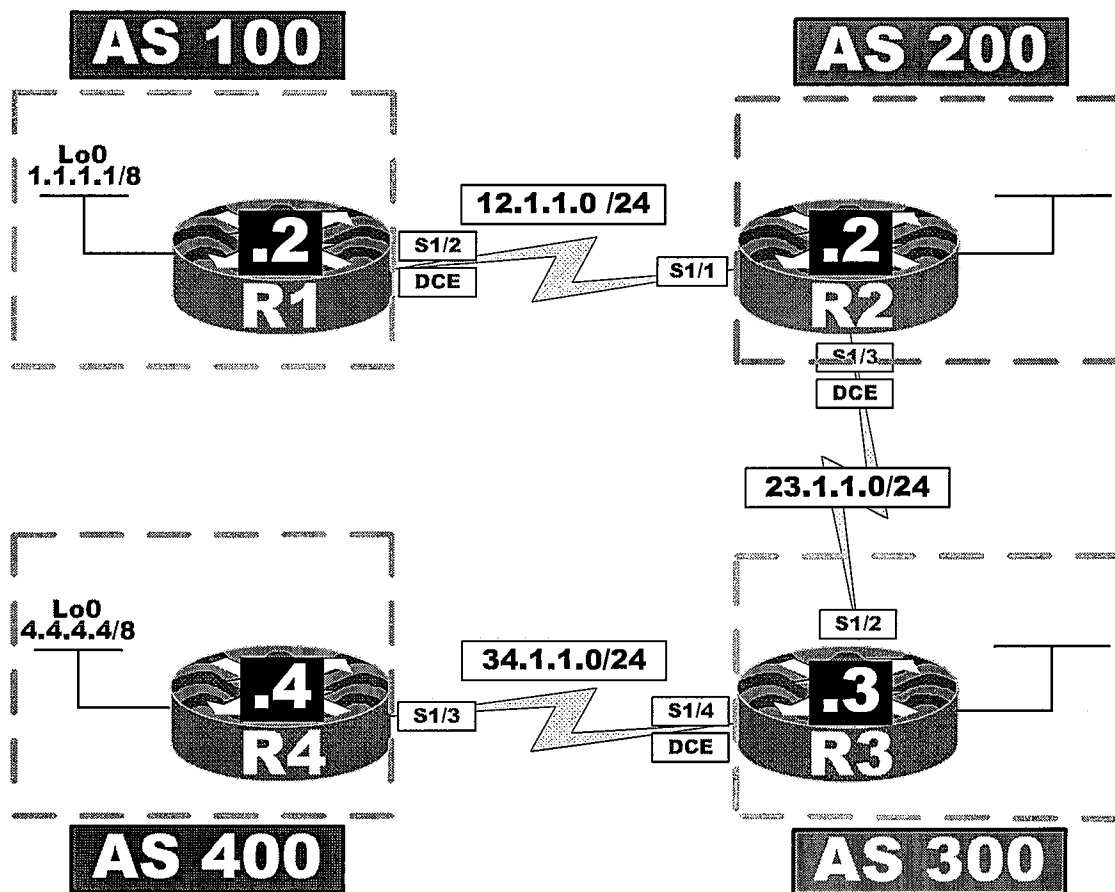
```
R4#Sh ip bgp | B Network
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*	2.0.0.0	34.1.1.3			0	300 200 i
*>		14.1.1.1			0	100 200 i
*>	3.0.0.0	34.1.1.3	0		0	300 i
*		14.1.1.1			0	100 200 300 i
*>	4.0.0.0	0.0.0.0	0		32768	i
*>	22.0.0.0	14.1.1.1			0	100 200 i
*>	33.0.0.0	34.1.1.3	0		0	300 i
*		14.1.1.1			0	100 200 300 i

Task 8

Erase the startup config and reload the routers before proceeding to the next lab.

Lab 9 - Regular Expressions



Task 1

Configure the routers and the switches based on the above diagram. DO NOT configure any routing protocol.

On R1:

```
R1(config)#Int S1/2
R1(config-if)#IP address 12.1.1.1 255.255.255.0
R1(config-if)#Clock rate 64000
```

```
R1(config-if)#No shut
R1(config)#Int Lo0
R1(config-if)#IP address 1.1.1.1 255.0.0.0
```

On R2:

```
R2(config)#Int S1/1
R2(config-if)#IP address 12.1.1.2 255.255.255.0
R2(config-if)#No shut
```

```
R2(config)#Int S1/3
R2(config-if)#Clock rate 64000
R2(config-if)#IP address 23.1.1.2 255.255.255.0
R2(config-if)#No shut
```

```
R2(config)#Int Lo0
R2(config-if)#IP addr 2.2.2.2 255.0.0.0
```

On R3:

```
R3(config)#Int S1/2
R3(config-if)#IP address 23.1.1.3 255.255.255.0
R3(config-if)#No shut
```

```
R3(config)#Int Lo0
R3(config-if)#Ip addr 3.3.3.3 255.0.0.0
```

```
R3(config)#Int S1/4
R3(config-if)#Clock rate 64000
R3(config-if)#IP address 34.1.1.3 255.255.255.0
R3(config-if)#No shut
```

On R4:

```
R4(config)#Int S1/3
R4(config-if)#IP addr 34.1.1.4 255.255.255.0
R4(config-if)#No shut
```

```
R4(config)#Int Lo0
R4(config-if)#IP addr 4.4.4.4 255.0.0.0
```

To verify the configuration:

On R1:

R1#Ping 12.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.2, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

On R3:

R3#Ping 34.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 34.1.1.4, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

R3#Ping 23.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 23.1.1.2, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms

Task 2

Configure BGP peering based on the following policy:

- R1 should establish EBGP peer sessions with R2 in AS 200.
- R2 should establish EBGP peer sessions with R1 and R3 in AS 100 and 300 respectively.
- R3 should establish EBGP peer sessions with R2 and R4 in AS 200 and 400 respectively.
- R4 should establish EBGP peer sessions with R3 in AS 300.
- Advertise the Loopback interfaces in BGP.

On R1:

R1(config)#router bgp 100

R1(config-router)#No au

R1(config-router)#netw 1.0.0.0

R1(config-router)#neighbor 12.1.1.2 remote-as 200

On R2:

```
R2(config)#router bgp 200
R2(config-router)#netw 2.0.0.0
R2(config-router)#No au
R2(config-router)#neighbor 12.1.1.1 remote-as 100
R2(config-router)#neighbor 23.1.1.3 remote-as 300
```

On R3:

```
R3(config)#router bgp 300
R3(config-router)#No au
R3(config-router)#netw 3.0.0.0
R3(config-router)#neighbor 23.1.1.2 remote-as 200
R3(config-router)#neighbor 34.1.1.4 remote-as 400
```

On R4:

```
R4(config)#router bgp 400
R4(config-router)#No au
R4(config-router)#netw 4.0.0.0
R4(config-router)#neighbor 34.1.1.3 remote-as 300
```

To verify the configuration:

On R1:

R1#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0 200	i
*> 3.0.0.0	12.1.1.2			0 200 300	i
*> 4.0.0.0	12.1.1.2			0 200 300 400	i

On R2:

R2#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	12.1.1.1	0		0 100	i
*> 2.0.0.0	0.0.0.0	0		32768	i
*> 3.0.0.0	23.1.1.3	0		0 300	i
*> 4.0.0.0	23.1.1.3			0 300 400	i

On R3:

R3#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	23.1.1.2			0	200 100 i
*> 2.0.0.0	23.1.1.2	0		0	200 i
*> 3.0.0.0	0.0.0.0	0		32768	i
*> 4.0.0.0	34.1.1.4	0		0	400 i

On R4:

R4#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	34.1.1.3			0	300 200 100 i
*> 2.0.0.0	34.1.1.3			0	300 200 i
*> 3.0.0.0	34.1.1.3	0		0	300 i
*> 4.0.0.0	0.0.0.0	0		32768	i

R1#Ping 4.4.4.4 source Lo0

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

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 84/86/88 ms

Task 3

Configure R1 such that it filters all the prefixes that have originated in AS 300.

Let's translate the regular expression:

Regular Expression	Meaning
_	The underscore matches a space, comma, left brace ({) or right brace (}), beginning and the end of an input string
\$	The dollar means the end of string

Basically you are referencing the first AS number in the path; or another way of looking at this is to say that we don't care about the prefixes that have been advertised by any AS/es in the path, but we like to match the prefix that has been advertised/originated by the last ASN in the path. In this task we are asked

to filter prefixes that have originated and NOT traversed through AS 300. Let's see the difference, in the following example:

Let's see the prefixes that have traversed through AS 300:

```
R1#Sh ip bgp regexp _300_
```

BGP table version is 7, 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, m multipath, b backup-path, x best-external, f RT-

Filter

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 3.0.0.0	12.1.1.2		0	200	300 i
*> 4.0.0.0	12.1.1.2		0	200	300 400 i

Let's see the prefixes that have originated in AS 300:

```
R1#Sh ip bgp regexp _300$
```

BGP table version is 7, 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, m multipath, b backup-path, x best-external, f RT-

Filter

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 3.0.0.0	12.1.1.2		0	200	300 i

To configure, an "AS-PATH" access-list is configured to deny any prefixes that have originated in AS 300 and permit everything else:

On R1:

```
R1(config)#ip as-path access-list 1 deny _300$
```

```
R1(config)#ip as-path access-list 1 permit .*
```

The "AS-PATH" access-list is referenced in the "Neighbor filter-list" command inbound through neighbor 12.1.1.2 (R2):

```
R1(config)#router bgp 100
```

```
R1(config-router)#neighbor 12.1.1.2 filter-list 1 in
```

```
R1#Clear ip bgp * in
```

To verify the configuration:

On R1:

```
R1#Show ip bgp regexp _300$
```

```
R1#
```

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0 200	i
*> 4.0.0.0	12.1.1.2			0 200 300 400	i

Task 4

Remove the configuration command/s from the previous task before proceeding to the next task.

On R1:

```
R1(config)#No ip as-path access-list 1
```

```
R1(config)#router bgp 100
```

```
R1(config-router)#No neighbor 12.1.1.2 filter-list 1 in
```

```
R1#Clear ip bgp * in
```

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0 200	i
*> 3.0.0.0	12.1.1.2			0 200 300	i
*> 4.0.0.0	12.1.1.2			0 200 300 400	i

Task 5

Configure R1 such that it blocks all the prefixes that have AS 300 in their AS-Path list.

Let's see all the prefixes that have traversed or have AS 300 in their AS-path:

R1#**Show ip bgp regexp _300_**

BGP table version is 5, 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, m multipath, b backup-path, x best-external, f RT-
Filter
Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 3.0.0.0	12.1.1.2			0 200 300	i
*> 4.0.0.0	12.1.1.2			0 200 300 400	i

On R1:

```
R1(config)#ip as-path access-list 1 deny _300_  
R1(config)#ip as-path access-list 1 permit .  
  
R1(config)#router bgp 100  
R1(config-router)#neighbor 12.1.1.2 filter-list 1 in  
  
R1#Clear ip bgp * in
```

To verify the configuration:

On R1:

R1#**Show ip bgp | B Network**

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0 200	i

Task 6

Remove the configuration command from the previous task before proceeding to the next task.

On R1:

```
R1(config)#No ip as-path access-list 1
```

```
R1(config)#router bgp 100
```

```
R1(config-router)#No neighbor 10.1.12.2 filter-list 1 in
```

```
R1#Clear ip bgp * in
```

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0	200 i
*> 3.0.0.0	12.1.1.2			0	200 300 i
*> 4.0.0.0	12.1.1.2			0	200 300 400 i

Task 7

Configure R3 such that it doesn't advertise the prefixes that have originated in its own AS to any of its neighbors.

Let's translate and test the regular expression before we configure it. Remember that the prefixes that have originated in the local AS will NOT have their own ASN in the AS-Path; they are only prepended when the prefixes are advertised to an EBGp neighbor. This method is used so the upstream ASes will NOT use the local AS as a transit AS.

Regular expression	Meaning
^	Start of string, in BGP it means directly connected ASes.
\$	End of string.

Let's test the regular expression:

On R3:

```
R3#Show ip bgp regexp ^$
```

```
BGP table version is 7, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-
Filter
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 3.0.0.0	0.0.0.0	0		32768	i

Now that we know the regular expression works, let's configure it:

On R3:

```
R3(config)#ip as-path access-list 1 deny ^$
R3(config)#ip as-path access-list 1 permit .*

R3(config)#router bgp 300
R3(config-router)#neighbor 23.1.1.2 filter-list 1 out
R3(config-router)#neighbor 34.1.1.4 filter-list 1 out
```

```
R3#Clear ip bgp * Out
```

To verify the configuration:

On R4:

```
R4#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	34.1.1.3			0 300 200	100 i
*> 2.0.0.0	34.1.1.3			0 300 200	i
*> 4.0.0.0	0.0.0.0	0		32768	i

On R2:

```
R2#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path

*> 1.0.0.0	12.1.1.1	0	0 100 i
*> 2.0.0.0	0.0.0.0	0	32768 i
*> 4.0.0.0	23.1.1.3		0 300 400 i

Task 8

Remove the configuration command from the previous task before proceeding to the next task.

On R3:

```
R3(config)#No ip as-path access-list 1 deny ^$
```

```
R3(config)#router bgp 300
```

```
R3(config-router)#No neighbor 23.1.1.2 filter-list 1 out
```

```
R3(config-router)#No neighbor 34.1.1.4 filter-list 1 out
```

```
R3#Clear ip bgp * out
```

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0 200	i
*> 3.0.0.0	12.1.1.2			0 200 300	i
*> 4.0.0.0	12.1.1.2			0 200 300 400	i

On R2:

```
R2#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	12.1.1.1	0		0 100	i
*> 2.0.0.0	0.0.0.0	0		32768	i
*> 3.0.0.0	23.1.1.3	0		0 300	i
*> 4.0.0.0	23.1.1.3			0 300 400	i

On R4:

R4#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	34.1.1.3			0	300 200 100 i
*> 2.0.0.0	34.1.1.3			0	300 200 i
*> 3.0.0.0	34.1.1.3	0		0	300 i
*> 4.0.0.0	0.0.0.0	0		32768	i

Task 9

Configure R3 such that it blocks all the prefixes that have originated in its neighboring AS 200.

We know that the "^" means start of the string or directly connected AS/es, therefore, the regular expression can start with the "^" followed by the ASN and then end of the string which is the "\$". Let's test this:

R3#Sh ip bgp regexp ^200\$

BGP table version is 7, local router ID is 3.3.3.3

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, x best-external, f RT-
Filter

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 2.0.0.0	23.1.1.2	0		0	200 i

On R3:

R3(config)#ip as-path access-list 1 deny ^200\$

R3(config)#ip as-path access-list 1 permit .*

R3(config)#router bgp 300

R3(config-router)#neighbor 23.1.1.2 filter-list 1 in

R3#Clear ip bgp * in

To verify the configuration:

On R3:

R3#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	23.1.1.2			0	200 100 i
*> 3.0.0.0	0.0.0.0	0		32768	i
*> 4.0.0.0	34.1.1.4	0		0	400 i

Task 10

Remove the configuration command from the previous task before proceeding to the next task.

On R3:

R3(config)#NO ip as-path access-list 1

R3(config)#router bgp 300

R3(config-router)#No neighbor 10.1.23.2 filter-list 1 in

R3#Clear ip bgp * in

To verify the configuration:

On R3:

R3#Show ip bgp | B Network

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	23.1.1.2			0	200 100 i
*> 2.0.0.0	23.1.1.2	0		0	200 i
*> 3.0.0.0	0.0.0.0	0		32768	i
*> 4.0.0.0	34.1.1.4	0		0	400 i

Task 11

Configure R3 such that it blocks all the prefixes from its existing directly connected neighbors.

Let's translate the following regular expression:

Regular expression	Meaning
<code>^</code>	Start of string, in BGP it means directly connected
<code>[0-9]</code>	Range of characters, in BGP it means any value from 0 to 9 Inclusively
<code>+</code>	Means one or more of the instance; in this configuration it means that one or more instance of <code>[0-9]</code> .
<code>\$</code>	End of string.

Therefore, the regexp used in this solution means any AS number as long as it's directly connected.

Let's implement the solution and test:

On R3:

```
R3(config)#ip as-path access-list 1 deny ^[0-9]+$
R3(config)#ip as-path access-list 1 permit .*
```

```
R3(config)#router bgp 300
R3(config-router)#neighbor 23.1.1.2 filter-list 1 in
R3(config-router)#neighbor 34.1.1.4 filter-list 1 in
```

```
R3#Clear ip bgp * in
```

To verify the configuration:

On R3:

```
R3#Show ip bgp | B Network
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	1.0.0.0	23.1.1.2			0 200 100	i
*>	3.0.0.0	0.0.0.0	0		32768	i

Task 12

Remove the configuration command from the previous task before proceeding to the next task.

On R3:

```
R3(config)#No ip as-path access-list 1
```

```
R3(config)#router bgp 300
```

```
R3(config-router)#No neighbor 10.1.23.2 filter-list 1 in
```

```
R3(config-router)#No neighbor 10.1.34.4 filter-list 1 in
```

```
R3#Clear ip bgp * in
```

To verify the configuration:

On R3:

```
R3#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	23.1.1.2			0 200 100	i
*> 2.0.0.0	23.1.1.2	0		0 200	i
*> 3.0.0.0	0.0.0.0	0		32768	i
*> 4.0.0.0	34.1.1.4	0		0 400	i

Task 13

Configure R1 such that it blocks all the prefixes that have originated in AS 300 and traversed through AS 200.

Let's test the regular expression:

```
R1#Show ip bgp regexp _200_300$
```

BGP table version is 13, 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, m multipath, b backup-path, x best-external, f RT-

Filter

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 3.0.0.0	12.1.1.2			0 200 300	i

On R1:

```

R1(config)#ip as-path access-list 1 deny _200_300$
R1(config)#ip as-path access-list 1 permit .*

R1(config)#router bgp 100
R1(config-router)#neighbor 12.1.1.2 filter-list 1 in

R1#Clear ip bgp * in

```

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0 200	i
*> 4.0.0.0	12.1.1.2			0 200 300 400	i

Task 14

Remove the configuration command from the previous task before proceeding to the next task.

On R1:

```

R1(config)#No ip as-path access-list 1

R1(config)#router bgp 100
R1(config-router)#No neighbor 12.1.1.2 filter-list 1 in

R1#Clear ip bgp * in

```

To verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0 200	i
*> 3.0.0.0	12.1.1.2			0 200 300	i


```
*> 4.0.0.0          12.1.1.2          0 200 300 400 i
```

Task 15

Configure R1 in AS 100 to filter prefixes from its directly connected AS 200 if they have prepended their AS number multiple times.

Before the “As-path access-list” is written and applied, a “Show ip bgp regexp” command should be issued, if the desired output is displayed, then, the “as-path access-list” should be written and applied.

Note in the following regular expression the “(_\1)” section can be thought of the memory button in a calculator, basically the expression before it “^([0-9]+)” is what you are putting in the memory location 1, and the “*” specifies zero or more of the expression that is in the memory location 1.

On R1:

```
R1#Sh ip bgp regexp ^([0-9]+)(_\1)*$
```

```
BGP table version is 19, 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
*> 2.0.0.0	12.1.1.2	0		0	200 i

The “*” before the “\$” means zero or more occurrence, therefore, the above regular expression will reference the prefixes if the neighboring AS 200 has prepended or not. Let’s replace the “*” with “+”, this will ONLY match if AS 200 has prepended. Let’s test:

```
R1#Sh ip bgp regexp ^([0-9]+)(_\1)+$
R1#
```

Obviously AS 200 is NOT prepending its AS number and that’s why we don’t see any prefixes in the AS-Path. Let’s configure and test:

```
R1(config)#ip as-path access-list 1 permit ^([0-9]+)(_\1)+$
```

Therefore there are two ways to test the regular expression before applying it to a given neighbor/s:

1. Show ip bgp regexp
2. Show ip bgp filter-list

The next step is to apply it to neighbor/s:

```
R1 (config) #Router bgp 100
```

```
R1 (config-router) #neighbor 10.1.12.2 filter-list 1 in
```

The following uses the refresh messages so the changes can apply to the existing and new prefixes.

```
R1 #Clear ip bgp * in
```

To verify the configuration:

On R1:

```
R1 #Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0	200 i
*> 3.0.0.0	12.1.1.2			0	200 300 i
*> 4.0.0.0	12.1.1.2			0	200 300 400 i

To properly test this, let's configure R2 to prepend its AS number multiple times.

```
R2 (config) #ip prefix-list NET per 2.0.0.0/8
```

```
R2 (config) #route-map TST per 10
```

```
R2 (config-route-map) #match ip addr prefix NET
```

```
R2 (config-route-map) #set as-path prepend 200
```

```
R2 (config-route-map) #route-map TST per 90
```

```
R2 (config) #router bgp 200
```

```
R2 (config-router) #neighbor 12.1.1.1 route-map TST out
```

```
R2 #Clear ip bgp * out
```

To verify and test the configuration:

On R1:

```
R1 #Show ip bgp | B Net
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 3.0.0.0	12.1.1.2			0	200 300 i
*> 4.0.0.0	12.1.1.2			0	200 300 400 i

Since R2 is prepending its AS number multiple times, R1 filters the prefix, let's remove the filtering configuration from R1 and verify:

On R1:

```
R1(config)#No ip as-path access-list 1
```

```
R1(config)#router bgp 100
```

```
R1(config-router)#No neigh 12.1.1.2 filter-list 1 in
```

```
R1#Clear ip bgp * in
```

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0	200 200 i
*> 3.0.0.0	12.1.1.2			0	200 300 i
*> 4.0.0.0	12.1.1.2			0	200 300 400 i

Let's remove the prepending configuration from R2:

On R2:

```
R2(config)#No ip prefix-list NET
```

```
R2(config)#No route-map TST
```

```
R2(config)#Router bgp 200
```

```
R2(config-router)#No neigh 12.1.1.1 route-map TST out
```

```
R2#Clear ip bgp * Out
```

To test and verify the configuration:

On R1:

```
R1#Show ip bgp | B Network
```

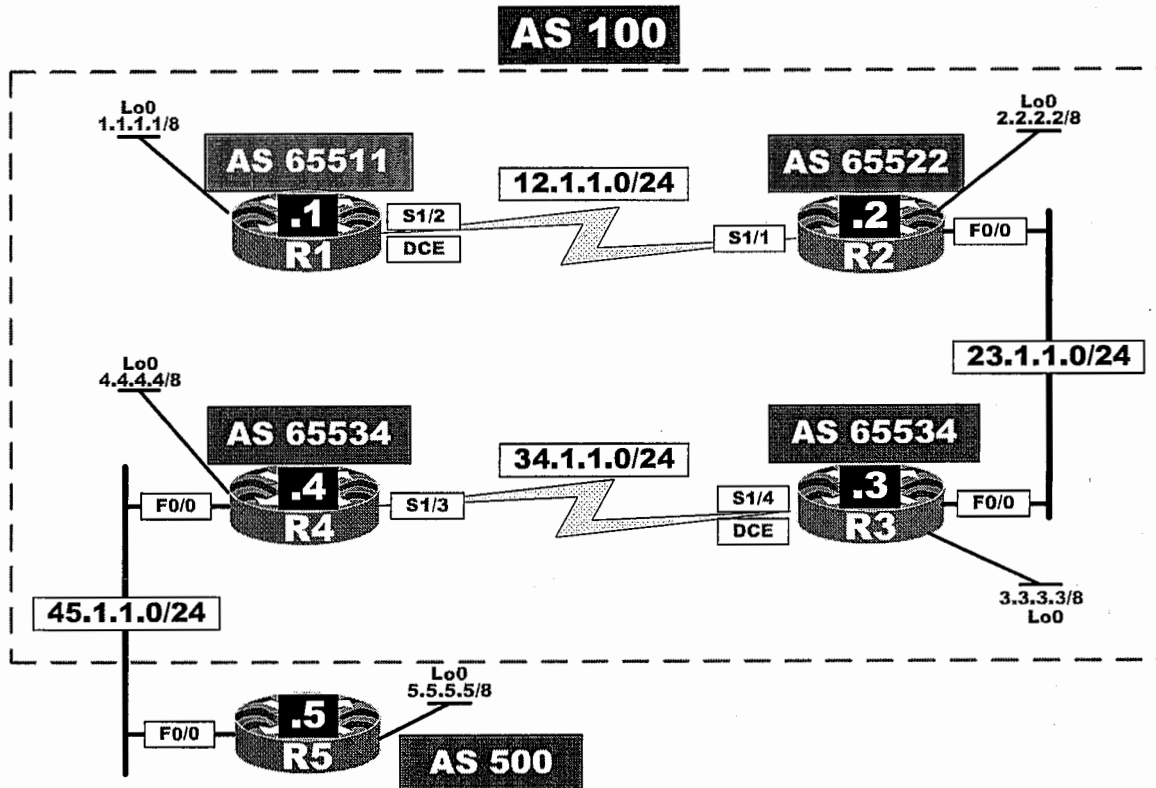
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	12.1.1.2	0		0	200 i
*> 3.0.0.0	12.1.1.2			0	200 300 i
*> 4.0.0.0	12.1.1.2			0	200 300 400 i

Task 16

Erase the startup config and reload the routers before proceeding to the next lab.

Lab 10

BGP Confederation



Task 1

Configure the routers and the switches based on the above diagram. DO NOT configure any routing protocol.

On SW1:

```
SW1(config)#Int Range f0/2-3
SW1(config-if-range)#Swi mode acc
SW1(config-if-range)#Swi acc vlan 100
SW1(config-if-range)#No shu
```

```
SW1(config)#Int Range f0/4-5
SW1(config-if-range)#Swi mode acc
SW1(config-if-range)#Swi acc vlan 200
SW1(config-if-range)#No shu
```

On R1:

```
R1(config)#int S1/2
R1(config-if)#Clock rate 64000
R1(config-if)#IP address 12.1.1.1 255.255.255.0
R1(config-if)#No shut

R1(config)#Int Lo0
R1(config-if)#IP address 1.1.1.1 255.0.0.0
```

On R2:

```
R2(config)#int S1/1
R2(config-if)#IP address 12.1.1.2 255.255.255.0
R2(config-subif)#No shut

R2(config)#Int F0/0
R2(config-if)#IP address 23.1.1.2 255.255.255.0
R2(config-if)#No shut

R2(config)#Int Lo0
R2(config-if)#IP addr 2.2.2.2 255.0.0.0
```

On R3:

```
R3(config)#int S1/4
R3(config-if)#IP address 34.1.1.3 255.255.255.0
R3(config-if)#Clock rate 64000
R3(config-if)#No shut

R3(config)#Int F0/0
R3(config-if)#IP address 23.1.1.3 255.255.255.0
R3(config-if)#No shu

R3(config)#Int Lo0
R3(config-if)#IP address 3.3.3.3 255.0.0.0
```

On R4:

```
R4(config)#int S1/3
```

```
R4(config-subif)#IP address 34.1.1.4 255.255.255.0
R4(config-if)#No shut
```

```
R4(config)#Int F0/0
R4(config-if)#IP address 45.1.1.4 255.255.255.0
R4(config-if)#No shu
```

```
R4(config)#Int Lo0
R4(config-if)#IP address 4.4.4.4 255.0.0.0
```

On R5:

```
R5(config)#Int Lo0
R5(config-if)#IP address 5.5.5.5 255.0.0.0
```

```
R5(config)#Int F0/0
R5(config-if)#IP address 45.1.1.5 255.255.255.0
R5(config-if)#No shut
```

To verify the configuration:

On R2:

```
R2#Ping 12.1.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

```
R2#Ping 23.1.1.3
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 23.1.1.3, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

On R4:

```
R4#Ping 34.1.1.3
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 34.1.1.3, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/29/32 ms

R4#**Ping 45.1.1.5**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 45.1.1.5, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

Task 2

Configure BGP peering on the routers as follows:

- Configure routers R1 – R4 in a confederation with their public AS number of 100.
- R1 in Sub-AS 65511, should establish an EBGP peer session with R2 in Sub-AS 65522.
- R2 in Sub-AS 65522, should establish EBGP peer sessions with R1 and R3 in Sub-ASes 65511 and 65534 respectively.
- R3 in Sub-AS 65534, should establish an EBGP peer sessions with R2 in Sub-AS 65522 and an IBGP peer session with R4 in AS 65534.
- R4 in Sub-AS 65534, should establish an IBGP peer sessions with R3 and an EBGP peer session with R5 in AS 500.
- R5 in AS 500 should establish an EBGP peer session with R4 in AS 100.
- These routers should advertise their Loopback interface in BGP.
- AS 100 should use OSPF for the links within the AS.

On R1:

```
R1(config)#Router ospf 1
R1(config-router)#Network 12.1.1.1 0.0.0.0 area 0

R1(config)#router bgp 65511
R1(config-router)#No au
R1(config-router)#netw 1.0.0.0
R1(config-router)#neighbor 12.1.1.2 remote-as 65522
R1(config-router)#bgp confederation identifier 100
```

The “**BGP confederation identifier**” command is used to configure a single AS number to identify a group of smaller ASes (Sub-ASes) as a single confederation. This command **MUST** be configured on all the routers within the confederation.

A confederation can be used to reduce the IBGP mesh by dividing a large single AS into multiple Sub-ASes and then grouping them into a single confederation.


```
R1(config-router)#bgp confederation peers 65522
```

The above command is used to identify the directly connected EBGP sub-confed peers.

On R2:

```
R2(config)#router ospf 1
R2(config-router)#Netw 12.1.1.2 0.0.0.0 area 0
R2(config-router)#Netw 23.1.1.2 0.0.0.0 area 0

R2(config)#router bgp 65522
R2(config-router)#No au
R2(config-router)#netw 2.0.0.0
R2(config-router)#neighbor 12.1.1.1 remote-as 65511
R2(config-router)#neighbor 23.1.1.3 remote-as 65534
R2(config-router)#bgp confederation identifier 100
R2(config-router)#bgp confederation peers 65511 65534
```

On R3:

```
R3(config)#router ospf 1
R3(config-router)#Netw 34.1.1.3 0.0.0.0 area 0
R3(config-router)#Netw 23.1.1.3 0.0.0.0 area 0

R3(config)#router bgp 65534
R3(config-router)#No au
R3(config-router)#neighbor 34.1.1.4 remote-as 65534
R3(config-router)#neighbor 23.1.1.2 remote-as 65522
R3(config-router)#netw 3.0.0.0
R3(config-router)#bgp confederation identifier 100
```

NOTE: R3 did not configure R4 (AS 65534) as a confederation peer, because the local router (R3) is in the same sub-confed as the R4.

```
R3(config-router)#bgp confederation peers 65522
```

On R4:

NOTE: R4 runs OSPF on the external link (45.1.1.0/24), this is configured so the routers within AS 100 can have reachability to the next-hop IP address for all the prefixes advertised by R5 in AS 500. The "Passive-interface F0/0" is configured so if R5 runs OSPF on the same link, it will not be able to establish an OSPF adjacency with R4 and have information on all the private IP addresses advertised in AS 100.

```
R4(config)#router ospf 1
R4(config-router)#Netw 34.1.1.4 0.0.0.0 area 0
```

```

R4(config-router)#Netw 45.1.1.4 0.0.0.0 area 0
R4(config-router)#Passive-interface F0/0

R4(config)#router bgp 65534

R4(config-router)#No au
R4(config-router)#netw 4.0.0.0
R4(config-router)#bgp confederation identifier 100
R4(config-router)#neighbor 45.1.1.5 remote-as 500
R4(config-router)#neighbor 34.1.1.3 remote-as 65534

```

NOTE: R4 does NOT have any directly connected EBGP sub-confed peers, therefore, the "BGP confederation peers" command is NOT configured.

On R5:

```

R5(config)#router bgp 500
R5(config-router)#No au
R5(config-router)#netw 5.0.0.0
R5(config-router)#neighbor 45.1.1.4 remote-as 100

```

To verify the configuration:

On R5:

```
R5#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	45.1.1.4			0 100	65511 i
*> 2.0.0.0	45.1.1.4			0 100	i
*> 3.0.0.0	45.1.1.4			0 100	i
*> 4.0.0.0	45.1.1.4	0		0 100	i
*> 5.0.0.0	0.0.0.0	0		32768	i

From R5's perspective, all the prefixes are from AS 100.

On R1:

```
R1#Show ip bgp | B Network
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 1.0.0.0	0.0.0.0	0		32768	i
*> 2.0.0.0	10.1.12.2	0	100	0	(65522) i
*> 3.0.0.0	10.1.23.3	0	100	0	(65522 65534) i
*> 4.0.0.0	10.1.34.4	0	100	0	(65522 65534) i

```
*> 5.0.0.0 10.1.45.5 0 100 0 (65522 65534) 500 i
```

Note the AS-Path/s in the parenthesis is the private AS numbers within the confederation; AS 500 is outside of the parenthesis because it's NOT part of the confederation.

Task 3

Erase the startup config and reload the routers before proceeding to the next lab.

CCIE Foundation

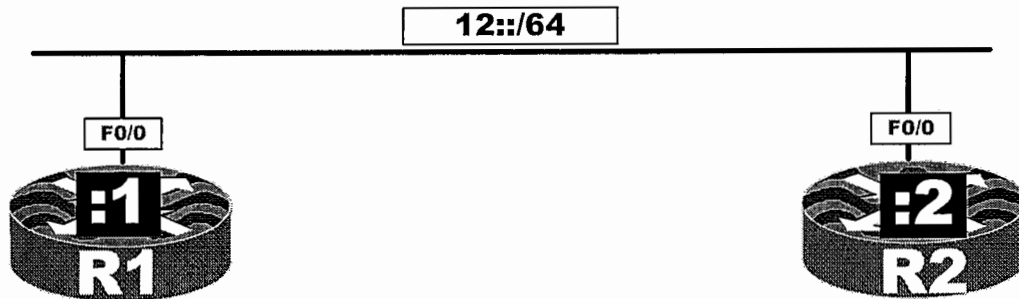
5.0

www.MicronicsTraining.com

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

IPv6

Lab 1 – Configuring Basic IPv6



In order to maximize hands on practice in IPv6, there are no initial config files for IPv6 labs.

Task 1

Configure SW1 to place the F0/0 interfaces of R1 and R2 in the same VLAN.

On SW1:

```
SW1(config)#Int range f0/1-2
SW1(config-if-range)#Swi mode acc
SW1(config-if-range)#Swi acc vlan 100
SW1(config-if-range)#No shut
```

Task 2

Enable IPv6 routing on R1.

On R1:

IPV6 routing is not enabled by default; therefore, it must be enabled statically.

```
R1(config)#ipv6 unicast-routing
```

Task 3

Configure the following MAC addresses on the F0/0 interfaces of these two routers:

R1 – 0000.1234.5678

R2 – 0000.dead.beef

On R1:

```
R1(config)#int f0/0
R1(config-if)#mac-address 0000.1234.5678
R1(config-if)#No shut
```

On R2:

```
R2(config)#int f0/0
R2(config-if)#mac-address 0000.dead.beef
R2(config-if)#No shut
```

Task 4

Configure the two routers such that they automatically assign a link-local address to their F0/0 interface. **Do not** configure IPv6 addressing of any kind to accomplish this task.

On Both Routers:

Before we enable IPv6 on the interfaces, let's enable ipv6 debug using the following command:

```
Router#Debug ipv6 nd
```

The above command enables debugging IPv6 neighbor discovery. Let's enable IPv6 on R1 first:

On R1:

```
R1(config-if)#int f0/0
R1(config-if)#ipv6 enable
R1(config-if)#shut
R1(config-if)#No shut
```

Layer 2 came up on the FastEthernet interface:

```
ICMPv6-ND: L2 came up on FastEthernet0/0
```

Trying to ensure that the Link Local IPv6 address is unique, but how did it come up with the following Link Local IPv6 address? This is explained later. DAD stands for Duplicate Address Detection.

```
IPv6-Addrmgr-ND: DAD request for FE80::200:12FF:FE34:5678 on
FastEthernet0/0
```

The local router sends a Neighbor Solicitation or "NS", to see if the link local address is unique:

```
ICMPv6-ND: Sending NS for FE80::200:12FF:FE34:5678 on FastEthernet0/0
```

Since the local router did not receive a response from any other device on the FastEthernet segment, the DAD process states that the Link local address is unique:

```
IPv6-Addrmgr-ND: DAD: FE80::200:12FF:FE34:5678 is unique.
```

The local router sends another Neighbor Advertisement stating that I am going to use the following link local address. This is like the last call.

```
ICMPv6-ND: Sending NA for FE80::200:12FF:FE34:5678 on FastEthernet0/0
```

Layer three on the FastEthernet interface comes up:

```
ICMPv6-ND: L3 came up on FastEthernet0/0
```

```
ICMPv6-ND: Linklocal FE80::200:12FF:FE34:5678 on FastEthernet0/0, Up
```

Since the "IPv6 unicast-routing" was enabled on R1, it sends a Router Advertisement for its link local IPv6 address:

```
ICMPv6-ND: Created RA context for FE80::200:12FF:FE34:5678
```

The local router will keep on sending the Router Advertisements to FF02::1 which is all the hosts on the local segment:

```
ICMPv6-ND: Request to send RA for FE80::200:12FF:FE34:5678
```

```
ICMPv6-ND: Sending RA from FE80::200:12FF:FE34:5678 to FF02::1 on
FastEthernet0/0
```

```
ICMPv6-ND:      MTU = 1500
ICMPv6-ND: ND output feature SEND executed on 3 - rc=0
```

Let's disable the debugging and test and verify the same process on R2:

```
R1#Undebug All
All possible debugging has been turned off
```

On R2:

```
R2#Debug ipv6 nd
  ICMP Neighbor Discovery events debugging is on
```

```
R2(config)#Int F0/0
R2(config-if)#Shut
R2(config-if)#IPv6 enable
R2(config-if)#No shut
```

NOTE: When IPv6 is enabled on the F0/0 interface (The first interface to have IPv6 enabled), a Link Local IPv6 address is assigned to the Null0 interface and a DAD states that the link local address is unique:

```
ICMPv6-ND: ND Module startup.
ICMPv6-ND: Initialise OL prefix database
ICMPv6-ND: IPv6 Opr Enabled on Null0
ICMPv6-ND: Allocate ND subblock on Null0 [1]
ICMPv6-ND: L2 came up on Null0
IPv6-Addrmgr-ND: DAD request for FE80::1 on Null0
IPv6-Addrmgr-ND: DAD: FE80::1 is unique.
ICMPv6-ND: L3 came up on Null0
ICMPv6-ND: Linklocal FE80::1 on Null0, Up
```

The next thing that happens is IPv6 is enabled on the F0/0 interface and the layer two protocol comes up on the interface:

```
ICMPv6-ND: IPv6 Opr Enabled on FastEthernet0/0
ICMPv6-ND: Allocate ND subblock on FastEthernet0/0 [3]
```

```
*May 22 18:01:55.467: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed
state to up
*May 22 18:01:56.467: %LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/0, changed state to up
ICMPv6-ND: L2 came up on FastEthernet0/0
```

DAD informs us that the link local address is unique.


```
IPv6-Addrmgr-ND: DAD request for FE80::200:DEFF:FEAD:BEEF on
FastEthernet0/0
ICMPv6-ND: Sending NS for FE80::200:DEFF:FEAD:BEEF on FastEthernet0/0

ICMPv6-ND: ND output feature SEND executed on 3 - rc=0
IPv6-Addrmgr-ND: DAD: FE80::200:DEFF:FEAD:BEEF is unique.
```

The local router sends a Neighbor Advertisement for the link local IPv6 address. The F0/0 interface comes up and the local router will keep on receiving Router Advertisements that are made by R1:

```
ICMPv6-ND: Sending NA for FE80::200:DEFF:FEAD:BEEF on FastEthernet0/0
ICMPv6-ND: L3 came up on FastEthernet0/0
ICMPv6-ND: Linklocal FE80::200:DEFF:FEAD:BEEF on FastEthernet0/0, Up
ICMPv6-ND: ND output feature SEND executed on 3 - rc=0

ICMPv6-ND: Received RA from FE80::200:12FF:FE34:5678 on FastEthernet0/0
```

Let's see how the link local IPv6 address was determined:

Once IPv6 is enabled, a Link Local address is automatically assigned, in this case the MAC address of the first Ethernet/FastEthernet interface is used. Let's see how the Link Local address was derived from the MAC address. On R2:

On R2:

```
R2#Show int F0/0 | Inc bia
```

```
Hardware is MV96340 Ethernet, address is 0000.dead.beef (bia
001c.58b9.f778)
```

MAC addresses are 48 bits long, whereas, the node/host portion of an IPv6 address is 64 bits long, to fill-in the difference they decided to add "FFFE" in the middle of the MAC address, let's try this manually:

0000.DEAD.BEEF



0000.DEFF.FEAD.BEEF

Then, the 7th bit in inverted:

0000.DEFF.FEAD.BEEF

↓ ↘
0000 0000
1234 5678

↓
The 7th bit is inverted, in this case it's 1

0000 0010

The new Link Local Address is:

0200.DEFF.FEAD.BEEF

OR

200.DEFF.FEAD.BEEF

Finally, the FE80 is concatenated to the above address to form the link local address of:

FE80::200.DEFF.FEAD.BEEF

To verify the configuration:

On R2:

```
R2#Show ipv6 interface brief | Exc admini|unass
FastEthernet0/0          [up/up]
    FE80::200:DEFF:FEAD:BEEF
```

```
R2#Show ipv6 inter f0/0 | Inc FF
```

```
IPv6 is enabled, link-local address is FE80::200:DEFF:FEAD:BEEF
FF02::1
FF02::1:FFAD:BEEF
Default router is FE80::200:12FF:FE34:5678 on FastEthernet0/0
```

NOTE: IPv6 is enabled, and the Link Local Address is FE80::200:DEFF:FEAD:BEEF

The joined groups are as follows:

FF02::1 – The “FF” means that this is a multicast address, the “0” after the FF means that the address is permanently assigned, if you see a “1” as the third character, it means that it is temporarily assigned. The “:1” at the end means all hosts on the local link.

FF02::1:FFAD:BEEF— This is the solicited-node-multicast address. The solicited-node-multicast always starts with "FF02::1:FF", and the last 24 bits or 6 hex characters of the link local address is added to complete the IPv6 address, in this case "AD:BEEF".

To see the link local IPv6 address for Null0 interface:

```
R2#Show ipv6 interface null0 | Inc IPv6
```

```
IPv6 is enabled, link-local address is FE80::1
```

Task 4

Configure the F0/0 interface of R1 with an IPv6 address of 2001:12:1::1/64; R2 should acquire the network portion of its IPv6 address from R1 through the SLAAC process. R1's Link Local address should be changed to "FE80::1".

On R1:

```
R1(config)#int f0/0  
R1(config-if)#ipv6 address 2001:12:1::1/64
```

NOTE: Whenever the IPv6 address of an interface is changed or a new one is assigned, the local router sends a Neighbor Solicitation for that IPv6 address, followed by a DAD to ensure that the IPv6 address is unique locally.

The following assigns an IPv6 Link Local address to the F0/0 interface:

```
R1(config-if)#ipv6 address FE80::1 link-local
```

On R2:

```
R2(config)#int f0/0  
R2(config-if)#IPv6 address autoconfig default
```

Note: The above configuration instructs the local router to acquire the network portion of its IPv6 address from a router, in this case R1. This process is called SLAAC (Stateless Address Auto Configuration), You should see the following debug output:

The local router sends a Router Solicitation on its F0/0 interface:

```
ICMPv6-ND: Sending RS on FastEthernet0/0
ICMPv6-ND: ND output feature SEND executed on 3 - rc=0
ICMPv6-ND: ND input feature SEND executed on 3 - rc=0
```

It immediately receives a Router Advertisement from FE80::1 (R1) with the network portion of the segment that R2 is on, in this case 2001:12:1::/64:

```
ICMPv6-ND: Received RA from FE80::1 on FastEthernet0/0
IPv6-Addrmgr-ND: Received prefix PI-flag change notification: prefix
2001:12:1::/64 onlink (was not-onlink)
```

The local router adds the network portion to the mode-ID or the host portion that it determined earlier and it wants to ensure that the new IPv6 address is unique:

```
IPv6-Addrmgr-ND: DAD request for 2001:12:1:0:200:DEFF:FEAD:BEEF on
FastEthernet0/0
```

The local router sends a Neighbor Solicitation to ensure it is unique:

```
ICMPv6-ND: Sending NS for 2001:12:1:0:200:DEFF:FEAD:BEEF on
FastEthernet0/0
```

Since the local router did not receive a Neighbor Advertisement for the solicitation, it autoconfigures the new IPv6 address on its F0/0 interface:

```
ICMPv6-ND: Autoconfiguring 2001:12:1:0:200:DEFF:FEAD:BEEF on
FastEthernet0/0
ICMPv6-ND: ND output feature SEND executed on 3 - rc=0
R2(config-if)#
```

DAD ensures that the IPv6 address is unique, and the local router proudly advertises the fact that it is using the new "2001:12:1:0:200:DEFF:FEAD:BEEF" IPv6 address.

```
IPv6-Addrmgr-ND: DAD: 2001:12:1:0:200:DEFF:FEAD:BEEF is unique.
ICMPv6-ND: Sending NA for 2001:12:1:0:200:DEFF:FEAD:BEEF on
FastEthernet0/0
ICMPv6-ND: ND output feature SEND executed on 3 - rc=0
```

To verify the configuration:

On R1:

```
R2#Show ipv6 interface brief | Exc admini|unass
```

```
FastEthernet0/0 [up/up]
FE80::200:DEFF:FEAD:BEEF
2001:12:1:0:200:DEFF:FEAD:BEEF
```

To verify the configuration:

On R2:

```
R2#clear ipv6 neighbors
```

```
R2#Ping fe80::1
```

```
Output Interface: fastethernet0/0
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to FE80::1, timeout is 2 seconds:
```

```
Packet sent with a source address of
```

```
FE80::200:DEFF:FEAD:BEEF%FastEthernet0/0
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/2/8 ms
```

Since the same Link Local IPv6 address can be assigned to multiple interfaces, when pinging a neighbor's Link local IPv6 address, the IOS asks which interface it should exit to reach that Link Local IPv6 address.

NOTE: The entire interface must be configured, meaning that an F0/0 will not work.

```
R1#Show ipv6 neighbors
```

IPv6 Address	Age	Link-layer Addr	State	Interface
FE80::1	3	0000.1234.5678	STALE	Fa0/0

The state, in this case is "STALE", Stale means that the particular entry was not used within the neighbor discovery reachable-time and therefore it transitioned from "REACH" to "STALE" state.

Task 5

Configure Loopback0 interfaces on both routers as follows:

- R1 – Loopback0 - 1:1:1:1:: /64, the router should assign the host portion of the IPv6 address automatically.
- R2 – Loopback0 – 2:2:2:2:: /64. Statically assign the host portion of this interface, the host portion should be based on the EUI-64 addressing scheme. Use the first FastEthernet interface's MAC address to accomplish this task.

On R1:

```
R1(config)#int lo0
R1(config-if)#ipv6 address 1:1:1:1::/64 eui-64
```

Note: On R1 we assigned the network address with the “eui-64” argument, which instructs the router to formulate the host portion of the IPv6 address automatically using the MAC address of the first LAN interface.

To verify the configuration:

On R1:

```
R1#Show ipv6 interface Lo0
```

```
Loopback0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::200:12FF:FE34:5678
No Virtual link-local address(es):
Global unicast address(es):
  1:1:1:1:200:12FF:FE34:5678, subnet is 1:1:1:1::/64 [EUI]
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FE34:5678
MTU is 1514 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ICMP unreachable are sent
ND DAD is not supported
ND reachable time is 30000 milliseconds (using 30000)
ND RAs are suppressed (periodic)
Hosts use stateless autoconfig for addresses.
```

On R2:

To configure R2's IPv6 address we must first find out the MAC address of the F0/0 interface:

```
R2#Show interface F0/0 | Inc bia
```

```
Hardware is MV96340 Ethernet, address is 0000.dead.beef (bia
001c.58b9.f778)
```

The second step is to insert “FFFE” in the middle of the MAC address, as follows:

0000.dead.beef → → This is the MAC address of R2's F0/0 interface

0000.deFF.FEad.beef → → The "FFFE" is inserted in the middle

200.deFF.FEad.beef → → The 7th bit is inverted, leading zero/s are dropped, and the new host portion is determined.

```
R2(config)#int lo0
```

```
R2(config-if)#ipv6 address 2:2:2:2:200:deff:fead:beef/64
```

To verify the configuration:

On R2:

```
R2#Show ipv6 interface Lo0
```

Loopback0 is up, line protocol is up

IPv6 is enabled, link-local address is FE80::200:DEFF:FEAD:BEEF

No Virtual link-local address(es):

Global unicast address(es):

2:2:2:2:200:DEFF:FEAD:BEEF, subnet is 2:2:2:2::/64

Joined group address(es):

FF02::1

FF02::1:FFAD:BEEF

MTU is 1514 bytes

ICMP error messages limited to one every 100 milliseconds

ICMP redirects are enabled

ICMP unreachable are sent

ND DAD is not supported

ND reachable time is 30000 milliseconds (using 30000)

Default router is FE80::1 on FastEthernet0/0

Task 6

Ensure that these routers can ping each other's loopback 0 interface. You should use static routes to accomplish this task.

On R1:

```
R1(config)#ipv6 route 2:2:2:2::/64 2001:12:1:0:200:DEFF:FEAD:BEEF
```

To verify this configuration:

On R1:

```
R1#Ping 2:2:2:2:200:DEFF:FEAD:BEEF
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2:2:2:2:200:DEFF:FEAD:BEEF, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms

The reason the Ping was successful is because the source IPv6 address that was used was the IPv6 address of R1's F0/0 interface.

```
R1#Show ipv6 route 2:2:2:2::/64
```

Routing entry for 2:2:2:2::/64

Known via "static", distance 1, metric 0

Route count is 1/1, share count 0

Routing paths:

2001:12:1:0:200:DEFF:FEAD:BEEF

Last updated 00:01:42 ago

Note: In the above example, the IPv6 address of the F0/0 interface of R2 was used as the next-hop IPv6 address to reach the 2:2:2:2::/64 network. The link-local IPv6 address can also be used for this purpose, remember that IGP's in IPv6 use the link-local IPv6 address as the next hop IPv6 address to reach the advertised networks.

Let's configure and verify:

```
R1(config)#No ipv6 route 2:2:2:2::/64 2001:12:1:0:200:DEFF:FEAD:BEEF
```

```
R1(config)#IPv6 route 2:2:2:2::/64 FE80::200:DEFF:FEAD:BEEF
```

% Interface has to be specified for a link-local nexthop

```
R1(config)#ipv6 route 2:2:2:2::/64 F0/0 FE80::200:DEFF:FEAD:BEEF
```

Note: On Multiaccess networks if the link-local IPv6 address is used as the next-hop, the interface through which the IPv6 network is reachable must be specified. This is because the link-local of all interfaces of a given router can be identical; therefore, we must specify the interface.

```
R1#Ping 2:2:2:2:200:DEFF:FEAD:BEEF
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to Ping 2:2:2:2:200:DEFF:FEAD:BEEF, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms

A default route can also be used, the following example configures a default route on R1:

```
R1(config)#No ipv6 route 2:2:2:2::/64 F0/0 FE80::200:DEFF:FEAD:BEEF
```

```
R1(config)#ipv6 route ::/0 F0/0 FE80::200:DEFF:FEAD:BEEF
```

To verify the configuration:

On R1

```
R1#Ping 2:2:2:2:200:DEFF:FEAD:BEEF
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2:2:2:2:200:DEFF:FEAD:BEEF, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/4 ms

To verify the configuration:

On R1:

```
R1#Show ipv6 route ::/0
```

Routing entry for ::/0

Known via "static", distance 1, metric 0

Route count is 1/1, share count 0

Routing paths:

FE80::200:DEFF:FEAD:BEEF, FastEthernet0/0

Last updated 00:00:50 ago

On R2:

```
R2(config)#ipv6 route ::/0 f0/0 FE80::1
```

Task 7

Create a host entry for the loopback0 interface of both routers such that when R1 needs to communicate with R2's loopback0 it uses "R2Lo0" instead of its IPv6 address.

On R1:

```
R1(config)# ipv6 host R2Lo0 2:2:2:2:200:DEFF:FEAD:BEEF
```

To verify the configuration:

On R1:

```
R1#Ping R2Lo0
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2:2:2:2:200:DEFF:FEAD:BEEF, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms

```
R1#Show hosts | B Host
```

Host	Port	Flags	Age	Type	Address(es)
R2Lo0	None	(perm, OK)	0	IPv6	2:2:2:2:200:DEFF:FEAD:BEEF

On R2:

```
R2(config)#ipv6 host R1Lo0 1:1:1:1:200:12FF:FE34:5678
```

```
R2#Ping R1Lo0
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 1:1:1:1:200:12FF:FE34:5678, timeout is 2 seconds:

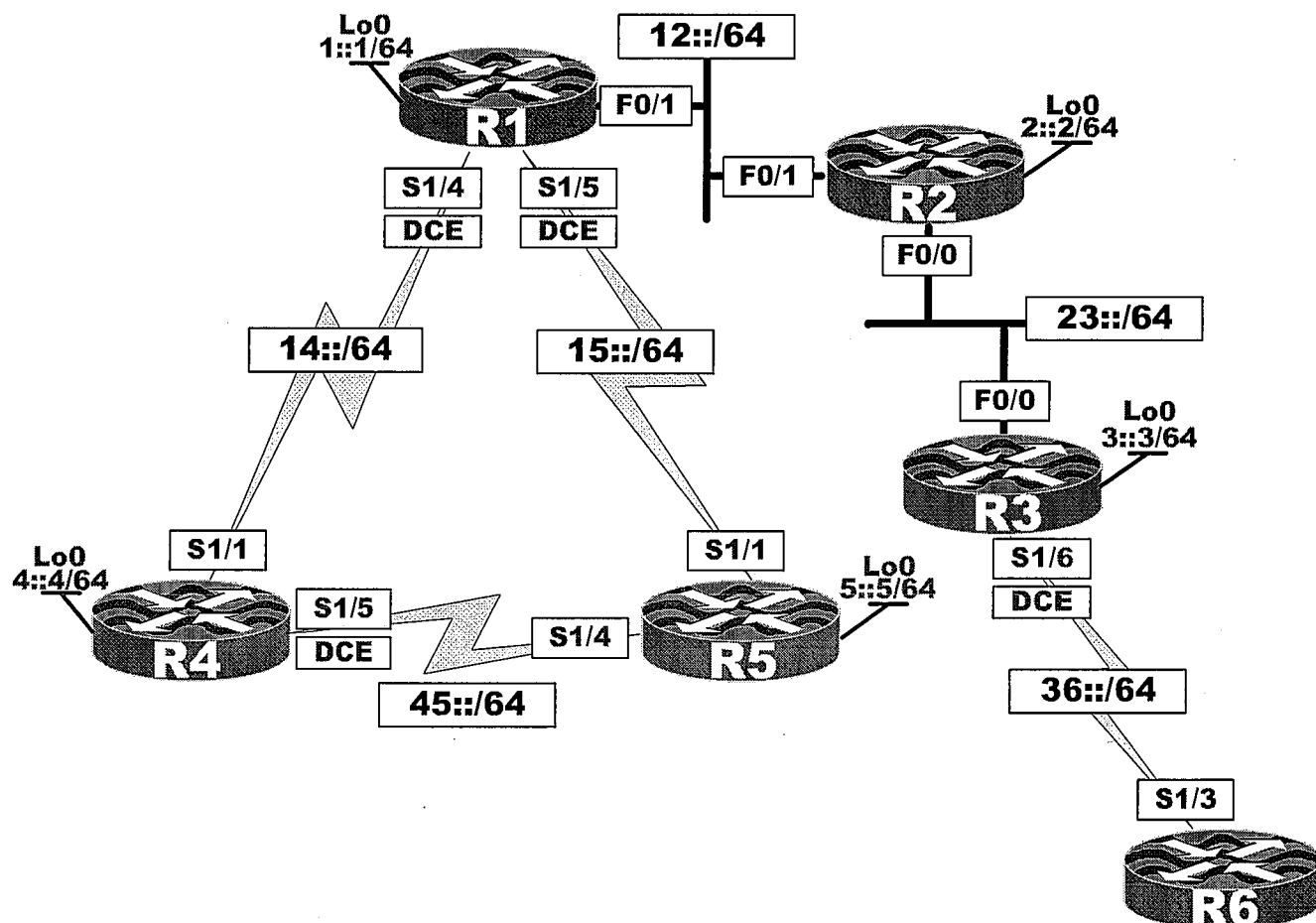
!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/4 ms

Task 8

Erase the startup config and reload the routers before proceeding to the next lab.

Lab 2 – Configuring EIGRPv6



Task 1

Configure the Routers and the Switches based on the above diagram. DO NOT configure any routing protocol. Configure fe80::1, fe80::2, fe80::3, fe80::4, fe80::5, and fe80::6 for R1, R2, R3, R4, R5 and R6 respectively.

On R1:

```
R1(config)#Int f0/1
R1(config-if)#ipv6 address 12::1/64
```

```
R1(config-if)#ipv6 address FE80::1 Link
R1(config-if)#No shut
```

```
R1(config-if)#int s1/4
R1(config-if)#clock rate 64000
R1(config-if)#ipv6 address 14::1/64
R1(config-if)#ipv6 address fe80::1 Link
R1(config-if)#No shut
```

```
R1(config)#Int s1/5
R1(config-if)#clock rate 64000
R1(config-if)#ipv6 address 15::1/64
R1(config-if)#ipv6 address fe80::1 Link
R1(config-if)#No shu
```

```
R1(config-if)#int lo0
R1(config-if)#ipv6 address 1::1/64
```

On R2:

```
R2(config)#int F0/1
R2(config-if)#ipv6 address 12::2/64
R2(config-if)#ipv6 address fe80::2 Link
R2(config-if)#No shut
```

```
R2(config)#int f0/0
R2(config-if)#ipv6 address 23::2/64
R2(config-if)#ipv6 address fe80::2 Link
R2(config-if)#No shut
```

```
R2(config)#int lo0
R2(config-if)#ipv6 address 2::2/64
```

On R3:

```
R3(config)#int f0/0
R3(config-if)#ipv6 addr 23::3/64
R3(config-if)#ipv6 addr fe80::3 Link
R3(config-if)#No shut
```

```
R3(config)#int s1/6
R3(config-if)#clock rate 64000
R3(config-if)#ipv6 address 36::3/64
R3(config-if)#ipv6 address fe80::3 Link
R3(config-if)#No shut
```

```
R3(config)#Int lo0
R3(config-if)#ipv6 address 3::3/64
```

On R4:

```
R4(config)#int s1/1
R4(config-if)#ipv6 address 14::4/64
R4(config-if)#ipv6 address fe80::4 Link
R4(config-if)#No shut
```

```
R4(config)#int s1/5
R4(config-if)#clock rate 64000
R4(config-if)#ipv6 address 45::4/64
R4(config-if)#ipv6 address fe80::4 Link
R4(config-if)#No shut
```

```
R4(config)#int lo0
R4(config-if)#Ipv6 address 4::4/64
```

On R5:

```
R5(config)#int s1/1
R5(config-if)#ipv6 address 15::5/64
R5(config-if)#ipv6 address fe80::5 Link
R5(config-if)#No shut
```

```
R5(config)#int s1/4
R5(config-if)#ipv6 address 45::5/64
R5(config-if)#ipv6 address fe80::5 Link
R5(config-if)#No shu
```

```
R5(config-if)#int lo0
R5(config-if)#ipv6 address 5::5/64
```

On R6:

```
R6(config)#Int s1/3
R6(config-if)#IPv6 address 36::6/64
R6(config-if)#ipv6 address fe80::6 Link
R6(config-if)#No shut
```

```
R6(config)#int Lo0
R6(config-if)#IPv6 address 6::6/64
```

On SW1:

```
SW1(config)#int range f0/2-3
SW1(config-if-range)#swi mode acc
```

```
SW1(config-if-range)#swi acc v 23
SW1(config-if-range)#No shut
```

On SW2:

```
SW2(config)#int range f0/1-2
SW2(config-if-range)#swi mode acc
SW2(config-if-range)#swi acc v 12
SW2(config-if-range)#No shu
```

Task 2

Configure Eigrpv6 AS 100 on R3 and R6 and advertise their Loopback and Serial interfaces in this routing domain.

In EIGRPv6 a router id and a “No Shut” router configuration command must be explicitly configured or else the process will NOT start; once the router-id and the “No Shut” command is configured, EIGRPv6 must be enabled on the interface/s. The Eigrp router id is NOT an IPv4 address; it is a 32 bit dotted decimal value, just like OSPFv2.

On R6:

```
R6(config)#IPv6 unicast-routing
```

```
R6(config)#Int S1/3
R6(config-if)#IPv6 eigrp 100
```

```
R6(config)#int Lo0
R6(config-if)#IPv6 eigrp 100
```

```
R6(config)#IPv6 router eigrp 100
R6(config-rtr)#No shut
R6(config-rtr)#router-id 0.0.0.6
```

```
R6#Show ipv protocol
```

```
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "connected"
```

IPv6 Routing Protocol is "eigrp 100"

EIGRP-IPv6 Protocol for AS(100)

Metric weight K1=1, K2=0, K3=1, K4=0, K5=0

NSF-aware route hold timer is 240

Router-ID: 0.0.0.6

Topology : 0 (base)

Active Timer: 3 min

Distance: internal 90 external 170

Maximum path: 16

Maximum hopcount 100

Maximum metric variance 1

Interfaces:

Serial1/3

Loopback0

Redistribution:

None

On R3:

```
R3(config)#IPv6 unicast-routing
```

```
R3(config)#Int S1/6
```

```
R3(config-if)#ipv eigrp 100
```

```
R3(config)#Int Lo0
```

```
R3(config-if)#ipv eigrp 100
```

```
R3(config)#IPv6 router eigrp 100
```

```
R3(config-rtr)#No shut
```

```
R3(config-rtr)#router-id 0.0.0.3
```

You should see the following console message:

```
EIGRP-IPv6 100: Neighbor FE80::6 (Serial1/6) is up: new adjacency
```

To verify the configuration:

On R3:

```
R3#Show ipv6 route Eigrp
```

IPv6 Routing Table - default - 8 entries

Codes: C - Connected, L - Local, S - Static, U - Per-user Static route

B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP

IL - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

D - EIGRP, EX - EIGRP external, NM - NEMO, ND - Neighbor Discovery
 1 - LISP
 O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
 ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D 6::/64 [90/20640000]
 via FE80::6, Serial1/6

R6#Show ipv6 eigrp neighbors
 EIGRP-IPv6 Neighbors for AS(100)

H	Address	Interface	Hold Uptime (sec)	SRTT (ms)	RTO	Q Cnt	Seq Num
0	Link-local address: FE80::6	Se1/6	12 00:01:51	1285	5000	0	3

Task 3

Configure EIGRPv6 AS 100 on the F0/1 and the loopback 0 interfaces of R1 and R2, the F0/0 interfaces of R2 and R3.

On R1:

```
R1(config)#IPv6 unicast-routing

R1(config)#Int F0/1
R1(config-if)#IPv6 Eigrp 100

R1(config)#Int Lo0
R1(config-if)#IPv6 Eigrp 100

R1(config)#IPv6 router eigrp 100
R1(config-rtr)#router-id 0.0.0.1
R1(config-rtr)#No shut
```

On R2:

```
R2(config)#Int F0/1
R2(config-if)#IPv6 eigrp 100

R2(config)#Int Lo0
R2(config-if)#IPv6 Eigrp 100

R2(config)#Int F0/0
```



```
R2(config-if)#IPv6 Eigrp 100
```

```
R2(config)#IPv6 router eigrp 100
```

```
R2(config-rtr)#router-id 0.0.0.2
```

```
R2(config-rtr)#No shut
```

You should see the following console message:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv6 100: Neighbor FE80::1 (FastEthernet0/1) is up: new adjacency
```

On R3:

```
R3(config)#Int F0/0
```

```
R3(config-if)#IPv6 Eigrp 100
```

You should see the following console message:

```
%DUAL-5-NBRCHANGE: EIGRP-IPv6 100: Neighbor FE80::2 (FastEthernet0/0) is up: new adjacency
```

To verify the configuration:

On R6:

```
R6#Show IPv6 route Eigrp
```

IPv6 Routing Table - default - 10 entries

Codes: C - Connected, L - Local, S - Static, U - Per-user Static route

B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

D - EIGRP, EX - EIGRP external, NM - NEMO, ND - Neighbor Discovery

l - LISP

O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

```
D 1::/64 [90/20645120]
  via FE80::3, Serial1/3
D 2::/64 [90/20642560]
  via FE80::3, Serial1/3
D 3::/64 [90/20640000]
  via FE80::3, Serial1/3
D 12::/64 [90/20517120]
  via FE80::3, Serial1/3
D 23::/64 [90/20514560]
  via FE80::3, Serial1/3
```

Task 4

Configure Eigrpv6 AS 100 on the Serial interfaces of R1, and all directly connected interfaces of R4 and R5.

On R1:

```
R1(config)#Int S1/4  
R1(config-if)#IPv6 Eigrp 100
```

```
R1(config)#Int S1/5  
R1(config-if)#IPv6 Eigrp 100
```

On R4:

```
R4(config)#IPv6 unicast-routing
```

```
R4(config)#Int S1/1  
R4(config-if)#IPv6 Eigrp 100
```

```
R4(config)#Int S1/5  
R4(config-if)#IPv6 Eigrp 100
```

```
R4(config)#Int Lo0  
R4(config-if)#IPv6 Eigrp 100
```

```
R4(config)#Ipv6 router eigrp 100  
R4(config-rtr)#router-id 0.0.0.4  
R4(config-rtr)#No shut
```

You should see the following console message:

```
EIGRP-IPv6 100: Neighbor FE80::1 (Serial1/1) is up: new adjacency
```

On R5:

```
R5(config)#IPv6 unicast-routing
```

```
R5(config)#Int S1/1  
R5(config-if)#IPv6 Eigrp 100
```

```
R5(config)#Int S1/4  
R5(config-if)#IPv6 Eigrp 100
```

```
R5(config)#Int Lo0
```

```
R5(config-if)#IPv6 Eigrp 100
```

```
R5(config)#IPv6 router eigrp 100
```

```
R5(config-rtr)#router-id 0.0.0.5
```

```
R5(config-rtr)#No shut
```

You should see the following console message:

```
EIGRP-IPv6 100: Neighbor FE80::1 (Serial1/1) is up: new adjacency
```

```
EIGRP-IPv6 100: Neighbor FE80::4 (Serial1/4) is up: new adjacency
```

To verify the configuration:

On R6:

```
R6#Show ipv6 route Eigrp
```

IPv6 Routing Table - default - 15 entries

Codes: C - Connected, L - Local, S - Static, U - Per-user Static route

B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

D - EIGRP, EX - EIGRP external, NM - NEMO, ND - Neighbor Discovery

1 - LISP

O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

```
D 1::/64 [90/20645120]
    via FE80::3, Serial1/3
D 2::/64 [90/20642560]
    via FE80::3, Serial1/3
D 3::/64 [90/20640000]
    via FE80::3, Serial1/3
D 4::/64 [90/21157120]
    via FE80::3, Serial1/3
D 5::/64 [90/21157120]
    via FE80::3, Serial1/3
D 12::/64 [90/20517120]
    via FE80::3, Serial1/3
D 14::/64 [90/21029120]
    via FE80::3, Serial1/3
D 15::/64 [90/21029120]
    via FE80::3, Serial1/3
D 23::/64 [90/20514560]
    via FE80::3, Serial1/3
D 45::/64 [90/21541120]
    via FE80::3, Serial1/3
```

Task 5

Configure the frame-relay of R1, R4 and R5 to exchange hellos every 90 seconds and a holddown timer of 270 seconds.

Let's see the default timers before changing them:

On R1:

```
R1#Show ipv6 Eigrp interface detail | s Se1/4
```

```
Se1/4          1          0/0          75          7/190          530
0
```

```
Hello-interval is 5, Hold-time is 15
```

```
Split-horizon is enabled
```

```
Next xmit serial <none>
```

```
Un/reliable mcasts: 0/0 Un/reliable ucasts: 4/5
```

```
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
```

```
Retransmissions sent: 1 Out-of-sequence rcvd: 0
```

```
Topology-ids on interface - 0
```

```
Authentication mode is not set
```

We can see that on the S1/4 interface the Hello interval is auto-set to 5 Seconds and the hold-time is set to 15 seconds. Let's change the timers:

On R4 and R5:

```
R1(config)#int S1/1
```

```
R1(config-if)#IPv6 hello-interval eigrp 100 90
```

```
R1(config-if)#IPv6 hold-time eigrp 100 270
```

On R5:

```
R5(config)#int s1/4
```

```
R5(config-if)#IPv6 hello-interval eigrp 100 90
```

```
R5(config-if)#IPv6 hold-time eigrp 100 270
```

On R4:

```
R4(config)#int s1/5
```

```
R4(config-if)#IPv6 hello-interval eigrp 100 90
```

```
R4(config-if)#IPv6 hold-time eigrp 100 270
```

To verify the configuration:

On R4:

```
R4#Show ipv6 Eigrp interface det | S Se1/1
```

```
Se1/1          1          0/0          105          7/190          638
0
```

```
Hello-interval is 90, Hold-time is 270
```

```
Split-horizon is enabled
```

```
Next xmit serial <none>
```

```
Un/reliable mcasts: 0/0 Un/reliable ucasts: 6/9
```

```
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
```

```
Retransmissions sent: 1 Out-of-sequence rcvd: 1
```

```
Topology-ids on interface - 0
```

```
Authentication mode is not set
```

Task 6

Authenticate the link between R3 and R6 using "Cisco" as the password.

On R3 and R6:

```
Rx(config)#Key chain TST
```

```
Rx(config-keychain)#Key 1
```

```
Rx(config-keychain-key)#Key-string Cisco
```

On R3:

```
R3(config)#Int S1/6
```

```
R3(config-if)#IPv6 authentication key-chain eigrp 100 TST
```

```
R3(config-if)#IPv6 authentication mode eigrp 100 md5
```

On R6:

```
R6(config)#Int S1/3
```

```
R6(config-subif)#IPv6 authentication key-chain eigrp 100 TST
```

```
R6(config-subif)#IPv6 authentication mode eigrp 100 md5
```

To verify the configuration:

On R3:

R3#Show ipv6 eigrp 100 interfaces det | S Se1/6

Se1/6 1 0/0 1310 7/190 6650
0

Hello-interval is 5, Hold-time is 15
Split-horizon is enabled
Next xmit serial <none>
Un/reliable mcasts: 0/0 Un/reliable ucasts: 18/20
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 0 Out-of-sequence rcvd: 2
Topology-ids on interface - 0
Authentication mode is md5, key-chain is "TST"

R3#Show ipv6 eigrp neighbors

IPv6-EIGRP neighbors for process 100

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
0	Link-local address: FE80::6	Se1/6	10	00:01:56	1310	5000	0	21
1	Link-local address: FE80::2	Fa0/0	12	00:24:14	1	200	0	35

Task 7

Configure R6 to filter network 1::/64.

Let's verify the routing table of R6 for network 1::/64

On R6:

R6#Show ipv6 route 1::/64

Routing entry for 1::/64

Known via "eigrp 100", distance 90, metric 20645120, type internal

Route count is 1/1, share count 0

Routing paths:

FE80::3, Serial1/3

Last updated 00:02:50 ago

To filter network 1::/64, a prefix-list is configured to deny network 1::/64 and permit everything else:

R6(config)#IPv6 prefix-list TST deny 1::/64

```
R6(config)#IPv6 prefix-list TST permit ::/0 LE 128
```

The distribute-list is configured to reference the prefix-list TST inbound on its S0/0/0.63 sub-interface:

```
R6(config)#IPv6 router eigrp 100
```

```
R6(config-rtr)#Distribute-list prefix TST in S1/3
```

To verify the configuration:

On R6:

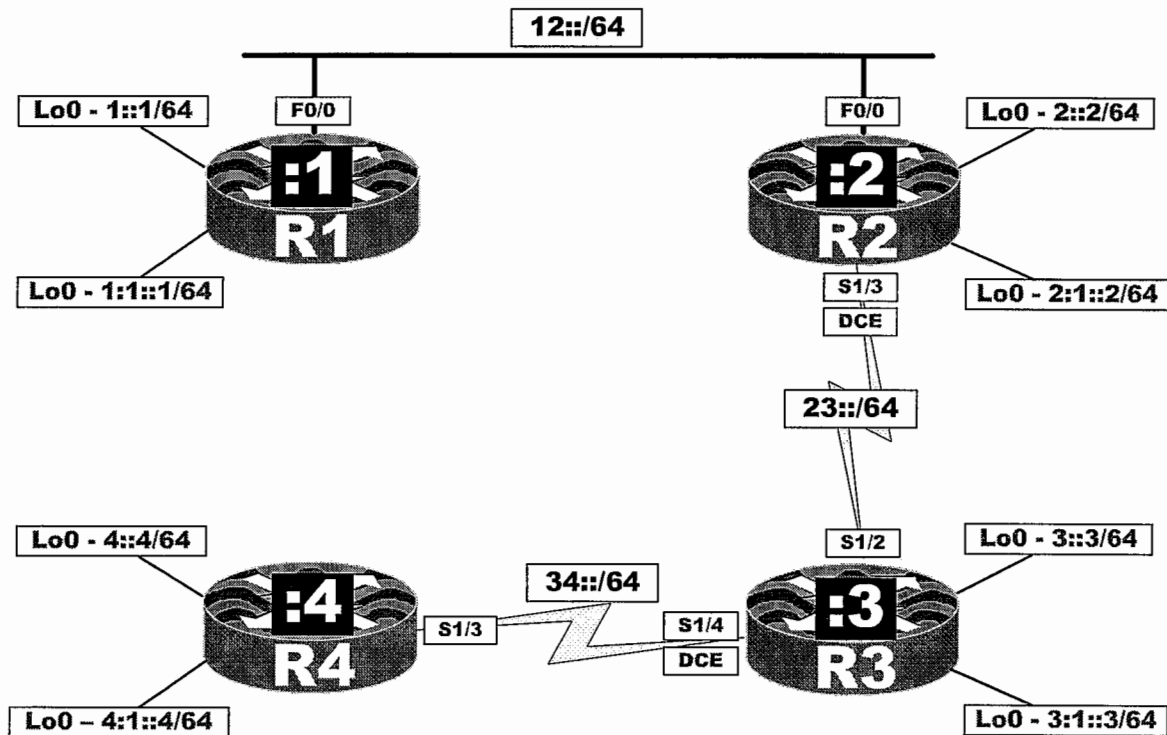
```
R6#Show ipv6 route 1::/64
```

```
% Route not found
```

Task 8

Erase the routers and the switches and reload these devices before proceeding to the next lab.

Lab 5 – OSPFv3 & The LSAs



Task 1

Configure the Routers based on the above topology. DO NOT configure any routing protocols.

On SW1:

```
SW1(config)#int range f0/1-2
SW1(config-if-range)#Swi mode acc
SW1(config-if-range)#swi acc v 12
SW1(config-if-range)#No shut
```

On R1:


```
R1(config)#int lo0
R1(config-if)#ipv6 addr 1::1/64

R1(config-if)#int lo1
R1(config-if)#ipv6 addr 1:1::1/64

R1(config-if)#int F0/0
R1(config-if)#ipv6 addr 12::1/64
R1(config-if)#ipv6 addr fe80::1 Link
R1(config-if)#No shut
```

On R2:

```
R2(config)#int lo0
R2(config-if)#ipv6 addr 2::2/64

R2(config)#int lo1
R2(config-if)#ipv6 addr 2:1::2/64

R2(config)#int F0/0
R2(config-if)#ipv6 address 12::2/64
R2(config-if)#ipv6 address fe80::2 Link
R2(config-if)#No shut

R2(config)#int s1/3
R2(config-if)#clock rate 64000
R2(config-if)#ipv6 addr 23::2/64
R2(config-if)#ipv6 addr fe80::2 Link
R2(config-if)#No shut
```

On R3:

```
R3(config)#int lo0
R3(config-if)#ipv6 addr 3::3/64

R3(config)#int lo1
R3(config-if)#ipv6 addr 3:1::3/64

R3(config)#int s1/2
R3(config-if)#ipv6 addr 23::3/64
R3(config-if)#ipv6 addr fe80::3 Link
R3(config-if)#No shut

R3(config)#int s1/4
R3(config-if)#clock rate 64000
R3(config-if)#ipv6 addr 34::3/64
```

```
R3(config-if)#ipv6 addr fe80::3 Link
R3(config-if)#No shut
```

On R4:

```
R4(config)#int lo0
R4(config-if)#ipv6 addr 4::4/64

R4(config)#int lo1
R4(config-if)#ipv6 addr 4:1::4/64

R4(config)#int s1/3
R4(config-if)#ipv6 addr 34::4/64
R4(config-if)#ipv6 addr fe80::4 Link
R4(config-if)#No shut
```

To verify the configuration:

On R2:

```
R2#Ping 12::1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12::1, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms

```
R2#Ping 23::3
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 23::3, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/30/36 ms

On R3:

```
R3#Ping 23::2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 23::2, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms

```
R3#Ping 34::4
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 34::4, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms

Task 2

Configure R1 and R2 based on the following policy:

Router	Interfaces	Area	RID
R1	F0/0	1	0.0.0.1
	Lo0	1	
	Lo1	1	
R2	F0/0	1	0.0.0.2
	Lo1	1	

On R1:

```
R1(config)#IPv6 unicast-routing
```

NOTE: When configuring OSPFv3, if the local router does NOT have an active IP address for OSPFv3 to use as a router-id, a router-id MUST be configured manually or else the OSPF process will NOT start.

Remember that in OSPFv2, OSPFv3 and in EIGRPv6 the router-id is a 32-bit dotted decimal value and NOT an IP address.

```
R1(config)#IPv6 router ospf 1
% IPv6 routing not enabled
R1(config-rtr)#router-id 0.0.0.1
```

```
R1(config)#Int Lo0
R1(config-if)#IPv6 ospf 1 area 1
```

```
R1(config)#Int Lo1
R1(config-if)#IPv6 ospf 1 area 1
```

```
R1(config)#int F0/0
R1(config-if)#IPv6 ospf 1 area 1
```

On R2:

```
R2(config)#IPv6 unicast-routing
```

```
R2(config)#Ipv6 router ospf 1  
R2(config-rtr)#router-id 0.0.0.2
```

```
R2(config)#Int Lo1
```

```
R2(config-if)#IPv6 ospf 1 area 1
```

```
R2(config)#Int F0/0  
R2(config-if)#IPv6 ospf 1 area 1
```

You should see the following console message:

```
%OSPFv3-5-ADJCHG: Process 1, Nbr 0.0.0.1 on FastEthernet0/0 from LOADING  
to FULL, Loading Done
```

To verify the configuration:

On R2:

```
R2#Show ipv6 ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Interface ID	Interface
0.0.0.1	1	FULL/BDR	00:00:35	3	FastEthernet0/0

```
R2#Show ipv6 route ospf
```

IPv6 Routing Table - 9 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route, M - MIPv6

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary

O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2

ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2

D - EIGRP, EX - EIGRP external

```
O 1::1/128 [110/1]  
  via FE80::1, FastEthernet0/0  
O 1:1::1/128 [110/1]  
  via FE80::1, FastEthernet0/0
```

Task 3

Identify, display and explain the LSAs in Area 1

Let's see the different types of LSAs in this area:

On R2:

```
R2#Show ipv6 ospf database | Inc States
```

```
Router Link States (Area 1)
Net Link States (Area 1)
Link (Type-8) Link States (Area 1)
Intra Area Prefix Link States (Area 1)
```

We can see four types of LSAs in Area 1: Type 1, 2, 8 and 9. Let's view more specific information about these LSAs, let's start with LSA Type-1s:

```
R2#Show ipv6 ospf database router
```

```
OSPFv3 Router with ID (0.0.0.2) (Process ID 1)
```

```
Router Link States (Area 1)
```

```
LS age: 138
Options: (V6-Bit, E-Bit, R-bit, DC-Bit)
LS Type: Router Links
Link State ID: 0
Advertising Router: 0.0.0.1
LS Seq Number: 80000004
Checksum: 0xA74B
Length: 40
Number of Links: 1
```

```
Link connected to: a Transit Network
Link Metric: 1
Local Interface ID: 3
Neighbor (DR) Interface ID: 3
Neighbor (DR) Router ID: 0.0.0.2
```

```
LS age: 137
Options: (V6-Bit, E-Bit, R-bit, DC-Bit)
LS Type: Router Links
Link State ID: 0
Advertising Router: 0.0.0.2
LS Seq Number: 80000004
Checksum: 0xA150
Length: 40
Number of Links: 1
```

```
Link connected to: a Transit Network
Link Metric: 1
Local Interface ID: 3
Neighbor (DR) Interface ID: 3
Neighbor (DR) Router ID: 0.0.0.2
```

Router LSAs (Type-1) describes the link state and the cost of the local router's link to the area. Just like the previous version, LSA type-1s are flooded within an area **ONLY**. This LSA also describes the router's role, meaning if the router is an ABR, ASBR and if it's one end of a virtual-link. These LSAs define the topology and have **NO ADDRESSING INFORMATION**.

NOTE: The output of the above show command reveals that LSA type 1s do not carry the IPv6 addressing, they **ONLY** identify the following:

- Options field, the "V6-Bit" – identifies that the router has the capability of forwarding transit IPv6 packets, the "E-Bit" – identifies the capability of the local router to accept external LSAs, the "R-bit" – this bit indicates that the router is active and the "DC-Bit" identifies that the router is capable of running demand circuit.
- Advertising routers RID
- Number of links in this area through the interface.
- The network type on HDLC, PPP links are point-to-point, and Ethernet links are Transit
- The cost of this router to the area
- The local routers interface-id and the interface-id to which it's connected.
- In OSPFv3, interface ids are used instead of an IPv6 address; every interface on a router is assigned an interface-id and the local router can see its neighbors interface-id.
- Their RID and Neighbor's RID to which they are connected.

Let's examine LSA Type-2s:

```
R2#Show ipv6 ospf database Network
```

```
OSPFv3 Router with ID (0.0.0.2) (Process ID 1)
```

```
Net Link States (Area 1)
```

```
LS age: 390
Options: (V6-Bit, E-Bit, R-bit, DC-Bit)
LS Type: Network Links
Link State ID: 3 (Interface ID of Designated Router)
Advertising Router: 0.0.0.2
LS Seq Number: 80000001
Checksum: 0xE818
Length: 32
Attached Router: 0.0.0.2
```

Attached Router: 0.0.0.1

On Multi-access (Broadcast and/or Non-broadcast) networks a DR/BDR is elected to cut down on the adjacency and if a router connected to that segment has a topology change, it notifies the DR and the DR floods the change onto that segment.

We can see that the advertising router has a router-id of 0.0.0.2. On a Multi-access networks the router with the highest priority will become the DR and in case of a tie, the router with the highest router-id will become the DR for that segment:

We can see that R2 is the DR. This LSA also identifies the router-ids of the routers that are connected to that multi-access network.

Let's check LSA Type-8s:

R2#Show ipv6 ospf database link

OSPFv3 Router with ID (0.0.0.2) (Process ID 1)

Link (Type-8) Link States (Area 1)

LS age: 481
Options: (V6-Bit E-Bit R-bit DC-Bit)
LS Type: Link-LSA (Interface: FastEthernet0/0)
Link State ID: 4 (Interface ID)
Advertising Router: 0.0.0.1
LS Seq Number: 80000001
Checksum: 0xB977
Length: 56
Router Priority: 1
Link Local Address: FE80::1
Number of Prefixes: 1
Prefix Address: 12::
Prefix Length: 64, Options: None

LS age: 490
Options: (V6-Bit E-Bit R-bit DC-Bit)
LS Type: Link-LSA (Interface: FastEthernet0/0)
Link State ID: 4 (Interface ID)
Advertising Router: 0.0.0.2
LS Seq Number: 80000001
Checksum: 0xC965
Length: 56
Router Priority: 1
Link Local Address: FE80::2

Number of Prefixes: 1
Prefix Address: 12::
Prefix Length: 64, Options: None

The output of the above show command displays the "Link LSAs" in OSPF's database. The Link LSAs are LSA type-8; these LSAs are flooded within the local link and are never flooded beyond the local link. Link LSAs provide the Link-Local IPv6 address of the local router to all other routers attached to the link. Link LSAs convey all IPv6 address(s) associated to the Link-Local IPv6 address.

Let's view LSA Type-9s:

R2#Show ipv6 ospf database Prefix

OSPFv3 Router with ID (0.0.0.2) (Process ID 1)

Intra Area Prefix Link States (Area 1)

Routing Bit Set on this LSA

LS age: 239

LS Type: Intra-Area-Prefix-LSA

Link State ID: 0

Advertising Router: 0.0.0.1

LS Seq Number: 80000009

Checksum: 0x9243

Length: 72

Referenced LSA Type: 2001

Referenced Link State ID: 0

Referenced Advertising Router: 0.0.0.1

Number of Prefixes: 2

Prefix Address: 1::1

Prefix Length: 128, Options: LA , Metric: 0

Prefix Address: 1:1::1

Prefix Length: 128, Options: LA , Metric: 0

Routing Bit Set on this LSA

LS age: 1699

LS Type: Intra-Area-Prefix-LSA

Link State ID: 0

Advertising Router: 0.0.0.2

LS Seq Number: 80000005

Checksum: 0x600F

Length: 52

Referenced LSA Type: 2001

Referenced Link State ID: 0

Referenced Advertising Router: 0.0.0.2

Number of Prefixes: 1

Prefix Address: 2:1::2

Prefix Length: 128, Options: LA , Metric: 0

Routing Bit Set on this LSA

LS age: 1701

LS Type: Intra-Area-Prefix-LSA

Link State ID: 4096

Advertising Router: 0.0.0.2

LS Seq Number: 80000001

Checksum: 0xD8C2

Length: 44

Referenced LSA Type: 2002

Referenced Link State ID: 4

Referenced Advertising Router: 0.0.0.2

Number of Prefixes: 1

Prefix Address: 12::

Prefix Length: 64, Options: None, Metric: 0

LSA Type-9s convey all the intra-area routes to the routers in that area.

Task 4

Configure R2 and R3 based on the following policy:

Router	Interfaces	Area	RID
R2	S1/3	0	0.0.0.2
	Lo0	0	
R3	S1/2	0	0.0.0.3
	Lo0	0	

On R2:

```
R2(config)#Int S1/3
```

```
R2(config-fr-dlci)#IPv6 ospf 1 area 0
```

```
R2(config)#Int Lo0
```

```
R2(config-if)#IPv6 ospf 1 area 0
```

To verify the configuration:

On R2:

R2#Show ipv6 ospf inter brief

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Lo0	1	0	22	1	LOOP	0/0	
Se1/3	1	0	8	781	P2P	0/0	
Lo1	1	1	23	1	LOOP	0/0	
Fa0/0	1	1	3	1	DR	1/1	

On R3:

R3(config)#IPv6 unicast-routing

R3(config)#Int S1/2

R3(config-fr-dlci)#IPv6 ospf 1 area 0

R3(config)#Int Lo0

R3(config-if)#IPv6 ospf 1 area 0

R3(config)#IPv6 router ospf 1

R3(config-rtr)#router-id 0.0.0.3

You should see the following console message:

%OSPFv3-5-ADJCHG: Process 1, Nbr 0.0.0.2 on Serial1/2 from LOADING to FULL, Loading Done

To verify the configuration:

On R3:

R3#Show ipv6 ospf inter brief

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Lo0	1	0	22	1	LOOP	0/0	
Se1/2	1	0	7	781	P2P	1/1	

R3#Show ipv6 route ospf

IPv6 Routing Table - 10 entries

Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP

U - Per-user Static route, M - MIPv6

I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
D - EIGRP, EX - EIGRP external

```
OI 1::1/128 [110/782]
    via FE80::2, Serial1/2
OI 1:1::1/128 [110/782]
    via FE80::2, Serial1/2
O 2::2/128 [110/781]
    via FE80::2, Serial1/2
OI 2:1::2/128 [110/781]
    via FE80::2, Serial1/2
OI 12::/64 [110/782]
    via FE80::2, Serial1/2
```

We can see four Inter-area (OI) and one intra-area (O) routes.

We should have additional LSAs because we have inter-area routes, let's see the LSAs types in the OSPF database:

On R2:

```
R2#Show ipv6 ospf database | Inc States
```

```
Router Link States (Area 0)
Inter Area Prefix Link States (Area 0)
Link (Type-8) Link States (Area 0)
Intra Area Prefix Link States (Area 0)
Router Link States (Area 1)
Net Link States (Area 1)
Inter Area Prefix Link States (Area 1)
Link (Type-8) Link States (Area 1)
Intra Area Prefix Link States (Area 1)
```

We can see two areas, area 0, and area 1. From area 0's perspective we can see LSAs Type 1, 3, 8, and 9; from area 1's perspective we can see five different LSA types: 1, 2, 3, 8 and 9.

Since we have examined area 1, let's view the LSDB from area 0's perspective:

```
R2#Show IPv6 ospf database router
```

```
OSPFv3 Router with ID (0.0.0.2) (Process ID 1)
```

```
Router Link States (Area 0)
```

```
LS age: 1183
```

Options: (V6-Bit E-Bit R-bit DC-Bit)

LS Type: Router Links

Link State ID: 0

Advertising Router: 0.0.0.2

LS Seq Number: 80000006

Checksum: 0xA5F1

Length: 40

Area Border Router

Number of Links: 1

Link connected to: another Router (point-to-point)

Link Metric: 781

Local Interface ID: 8

Neighbor Interface ID: 7

Neighbor Router ID: 0.0.0.3

LS age: 1189

Options: (V6-Bit E-Bit R-bit DC-Bit)

LS Type: Router Links

Link State ID: 0

Advertising Router: 0.0.0.3

LS Seq Number: 80000003

Checksum: 0x9407

Length: 40

Number of Links: 1

Link connected to: another Router (point-to-point)

Link Metric: 781

Local Interface ID: 7

Neighbor Interface ID: 8

Neighbor Router ID: 0.0.0.2

(The rest of the output is omitted)

We can see that it states that the local router is the Area Border Router, and it also shows the link connected is point-to-point and the neighboring router-id is 0.0.0.3.

The following Show command displays the LSA Type-3s from Area 0's perspective:

R2#Show ipv6 ospf database inter-area prefix

OSPFv3 Router with ID (0.0.0.2) (Process ID 1)

Inter Area Prefix Link States (**Area 0**)

LS age: 464

LS Type: Inter Area Prefix Links
Link State ID: 0
Advertising Router: 0.0.0.2
LS Seq Number: 80000002
Checksum: 0x693F
Length: 44
Metric: 1
Prefix Address: 1::1
Prefix Length: 128, Options: None

LS age: 464
LS Type: Inter Area Prefix Links
Link State ID: 1
Advertising Router: 0.0.0.2
LS Seq Number: 80000002
Checksum: 0x6D39
Length: 44
Metric: 1
Prefix Address: 1:1::1
Prefix Length: 128, Options: None

LS age: 466
LS Type: Inter Area Prefix Links
Link State ID: 2
Advertising Router: 0.0.0.2
LS Seq Number: 80000002
Checksum: 0x8321
Length: 44
Metric: 0
Prefix Address: 2:1::2
Prefix Length: 128, Options: None

LS age: 467
LS Type: Inter Area Prefix Links
Link State ID: 3
Advertising Router: 0.0.0.2
LS Seq Number: 80000002
Checksum: 0x2CB1
Length: 36
Metric: 1
Prefix Address: 12::
Prefix Length: 64, Options: None

NOTE: This information is what is injected into the routing table of R2, let's verify:

On R2:

```
R2#Show ipv6 route ospf | Inc O
```

```
      O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
O  1::1/128 [110/1]
O  1:1::1/128 [110/1]
O  3::3/128 [110/781]
```

NOTE: Loopback 0 interface of R1 (1::1/128) is advertised by R2 with a metric of 1. This should have been advertised with a metric of 2, the cost of the F0/0 interface (100/100=1) plus the cost of the loopback 0 interface which is also one, this should have been advertised with a metric of 2, what happened? Since OSPF can NOT establish an adjacency with another router through its loopback 0 interface it sees it as a single IP address and it totally ignores the mask and advertises it as a host route. Let's change the network type of the loopback0 interface to point-to-point and verify:

On R1:

```
R1(config)#Int Lo0
R1(config-if)#IPv6 ospf network point-to-point
```

To verify the configuration:

On R2:

```
R2#Show ipv6 ospf database inter-area prefix 1::/64
```

```
      OSPFv3 Router with ID (0.0.0.2) (Process ID 1)
LS age: 467
LS Type: Inter Area Prefix Links
Link State ID: 4
Advertising Router: 0.0.0.2
LS Seq Number: 80000001
Checksum: 0x5D90
Length: 36
Metric: 2
Prefix Address: 1::
Prefix Length: 64, Options: None
```

```
R2#Sh ipv route ospf | Inc O
```

```
      D - EIGRP, EX - EIGRP external, NM - NEMO, ND - Neighbor Discovery
      O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
      ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
```

```
O  1::/64 [110/2]
```

- O 1:1::1/128 [110/1]
- O 3::3/128 [110/781]

Let's examine LSA Type-9s:

On R2:

R2#Show ipv6 ospf database prefix

OSPFv3 Router with ID (0.0.0.2) (Process ID 1)

Intra Area Prefix Link States (Area 0)

Routing Bit Set on this LSA

LS age: 121

LS Type: Intra-Area-Prefix-LSA

Link State ID: 0

Advertising Router: 0.0.0.2

LS Seq Number: 80000006

Checksum: 0x348A

Length: 64

Referenced LSA Type: 2001

Referenced Link State ID: 0

Referenced Advertising Router: 0.0.0.2

Number of Prefixes: 2

Prefix Address: 2::2

Prefix Length: 128, Options: LA , Metric: 0

Prefix Address: 23::

Prefix Length: 64, Options: None, Metric: 781

Routing Bit Set on this LSA

LS age: 2033

LS Type: Intra-Area-Prefix-LSA

Link State ID: 0

Advertising Router: 0.0.0.3

LS Seq Number: 80000004

Checksum: 0x14A8

Length: 64

Referenced LSA Type: 2001

Referenced Link State ID: 0

Referenced Advertising Router: 0.0.0.3

Number of Prefixes: 2

Prefix Address: 23::

Prefix Length: 64, Options: None, Metric: 781

Prefix Address: 3::3

Prefix Length: 128, Options: LA , Metric: 0

(The rest of the output is omitted)

Task 4

Configure R3 and R4 based on the following policy:

Router	Interfaces	Area	RID
R3	Lo1	2	0.0.0.3
	S1/4	2	
R4	S1/3	2	0.0.0.4
	Lo0	2	
	Lo1	Redistribute into OSPF	

On R3:

```
R3(config)#Int S1/4
R3(config-if)#IPv6 ospf 1 area 2
```

```
R3(config)#int lo1
R3(config-if)#IPv6 ospf 1 area 2
```

To verify the configuration:

On R3:

```
R3#Show ipv6 ospf inter brie
```

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Lo0	1	0	22	1	LOOP	0/0	
Se1/2	1	0	7	781	P2P	1/1	
Lo1	1	2	23	1	LOOP	0/0	
Se1/4	1	2	9	781	P2P	0/0	

On R4:

```
R4(config)#IPv6 unicast-routing
```

```
R4(config)#Int S1/3
```



```

R4(config-if)#IPv6 ospf 1 area 2

R4(config)#Int Lo0
R4(config-if)#IPv6 ospf 1 area 2

R4(config)#Route-map TST permit 10
R4(config-route-map)#Match interface Lo1

R4(config)#IPv6 router ospf 1
R4(config-rtr)#router-id 0.0.0.4
R4(config-rtr)#Redistribute connected route-map TST

```

You should see the following console message:

```
%OSPFv3-5-ADJCHG: Process 1, Nbr 0.0.0.3 on Serial1/3 from LOADING to FULL, Loading Done
```

To verify the configuration:

On R4:

```
R4#Show ipv6 ospf inter brief
```

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Lo0	1	2	22	1	LOOP	0/0	
Se1/3	1	2	8	781	P2P	1/1	

Let's check the LSAs in area 2:

On R4:

```
R4#Show ipv6 ospf database | Inc States
```

```

Router Link States (Area 2)
Inter Area Prefix Link States (Area 2)
Link (Type-8) Link States (Area 2)
Intra Area Prefix Link States (Area 2)
Type-5 AS External Link States

```

So far we have covered LSA Types 1, 2, 3, 8, 9 but we can see the external link Which are Type-5s, let's examine Type-5 LSAs:

On R4:

```
R4#Show ipv6 ospf database external
```

OSPFv3 Router with ID (0.0.0.4) (Process ID 1)

Type-5 AS External Link States

LS age: 149

LS Type: AS External Link

Link State ID: 0

Advertising Router: 0.0.0.4

LS Seq Number: 80000001

Checksum: 0x268D

Length: 36

Prefix Address: 4:1::

Prefix Length: 64, Options: None

Metric Type: 2 (Larger than any link state path)

Metric: 20

We can see that just like OSPFv2 when routes are redistributed they are assigned a cost of 20 and a metric-type of 2.

Task 5

Re-configure the routers based on the following policy and ensure that the loopback interfaces are configured with their correct mask.

Router	Interfaces	Area	RID
R1	F0/0	0	0.0.0.1
	Lo0	0	
	Lo1	0	
R2	F0/0	0	0.0.0.2
	Lo0	0	
	S1/3	1	
	Lo1	1	
R3	S1/2	1	0.0.0.3
	Lo0	1	
	S1/4	2	
	Lo1	2	
R4	S1/3	2	0.0.0.4
	Lo0	2	
	Lo1	2	

On R1 – R4:

```
R1(config)#No IPv6 router ospf 1
```

On R1:

```
R1(config)#Int Lo0
R1(config-if)#IPv6 ospf network point-to-point
R1(config-if)#IPv6 ospf 1 area 0
```

```
R1(config-if)#Int Lo1
R1(config-if)#IPv6 ospf network point-to-point
R1(config-if)#IPv6 ospf 1 area 0
```

```
R1(config)#Int F0/0
R1(config-if)#IPv6 ospf 1 area 0
```

```
R1(config)#IPv6 router ospf 1
R1(config-rtr)#router-id 0.0.0.1
```

To verify the configuration:

On R1:

```
R1#Show ipv6 ospf inter brief
```

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Lo0	1	0	22	1	P2P	0/0	
Lo1	1	0	23	1	P2P	0/0	
Fa0/0	1	0	3	1	DR	0/0	

On R2:

```
R2(config)#Int Lo0
R2(config-if)#IPv6 ospf 1 area 0
R2(config-if)#IPv6 ospf network point-to-point
```

```
R2(config)#Int Lo1
R2(config-if)#IPv6 ospf 1 area 1
R2(config-if)#IPv6 ospf network point-to-point
```

```
R2(config)#Int F0/0
R2(config-if)#IPv6 ospf 1 area 0
```

```
R2(config)#Int S1/3
```

```
R2(config-subif)#IPv6 ospf 1 area 1
```

```
R2(config)#IPv6 router ospf 1
```

```
R2(config-rtr)#router-id 0.0.0.2
```

You should see the following console message:

```
%OSPFv3-5-ADJCHG: Process 1, Nbr 0.0.0.1 on FastEthernet0/0 from LOADING to FULL, Loading Done
```

```
R2#Sh ipv ospf int br
```

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Lo0	1	0	22	1	P2P	0/0	
Fa0/0	1	0	3	1	BDR	1/1	
Lo1	1	1	23	1	P2P	0/0	
Se1/3	1	1	8	781	P2P	0/0	

On R3:

```
R3(config)#Int S1/2
```

```
R3(config-subif)#IPv6 ospf 1 area 1
```

```
R3(config)#Int Lo0
```

```
R3(config-if)#IPv6 ospf network point-to-point
```

```
R3(config-if)#IPv6 ospf 1 are 1
```

```
R3(config)#Int lo1
```

```
R3(config-if)#IPv6 ospf network point-to-point
```

```
R3(config-if)#IPv6 ospf 1 are 2
```

```
R3(config)#Int S1/4
```

```
R3(config-if)#IPv6 ospf 1 area 2
```

```
R3(config)#IPv6 router ospf 1
```

```
R3(config-rtr)#router-id 0.0.0.3
```

You should see the following console message:

```
%OSPFv3-5-ADJCHG: Process 1, Nbr 0.0.0.2 on Serial1/2 from LOADING to FULL, Loading Done
```

To verify the configuration:

On R3:

R3#Show IPv6 ospf inter brief

Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
Lo0	1	1	22	1	P2P	0/0	
Se1/2	1	1	7	781	P2P	1/1	
Lo1	1	2	23	1	P2P	0/0	
Se1/4	1	2	9	781	P2P	0/0	

On R4:

```
R4(config)#Int Lo0
R4(config-if)#IPv6 ospf network point-to-point
R4(config-if)#IPv6 ospf 1 area 2
```

```
R4(config)#Int Lo1
R4(config-if)#IPv6 ospf network point-to-point
R4(config-if)#IPv6 ospf 1 area 2
```

```
R4(config)#Int F0/0
R4(config-if)#IPv6 ospf 1 area 2
```

```
R4(config)#IPv6 router ospf 1
R4(config-rtr)#router-id 0.0.0.4
```

You should see the following console message:

```
%OSPFv3-5-ADJCHG: Process 1, Nbr 0.0.0.3 on Serial1/3 from LOADING to
FULL, Loading Done
```

To verify the configuration:

On R4:

R4#Show ipv6 route ospf

```
IPv6 Routing Table - 8 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
O    3:1::/64 [110/782]
    via FE80::3, Serial1/3
```

This should NOT be a surprise since area 2 does NOT have a connection to area 0. We must configure a

virtual-link to logically connect area 2 to area 0, let's configure this virtual-link and verify:

On R2:

```
R2#Show ipv6 ospf | Inc ID
```

```
Routing Process "ospfv3 1" with ID 0.0.0.2
```

On R3:

```
R3#Show ipv6 ospf | Inc ID
```

```
Routing Process "ospfv3 1" with ID 0.0.0.3
```

Now that we know the router-id of the ABRs we can configure the virtual-link:

On R2:

```
R2(config)#IPv6 router ospf 1
R2(config-rtr)#Area 1 virtual-link 0.0.0.3
```

On R3:

```
R3(config)#IPv6 router ospf 1
R3(config-rtr)#Area 1 virtual-link 0.0.0.2
```

You should see the following console message:

```
%OSPFv3-5-ADJCHG: Process 1, Nbr 0.0.0.2 on OSPFv3_VL0 from LOADING to
FULL, Loading Done
```

To verify the configuration:

On R4:

```
R4#Show ipv6 route ospf
```

```
IPv6 Routing Table - 17 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route, M - MIPv6
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
       D - EIGRP, EX - EIGRP external
OI  1::/64 [110/1564]
```

```

    via FE80::3, Serial1/3
OI 1:1::/64 [110/1564]
    via FE80::3, Serial1/3
OI 2::/64 [110/1563]
    via FE80::3, Serial1/3
OI 2:1::/64 [110/1563]
    via FE80::3, Serial1/3
OI 2:1::2/128 [110/1562]
    via FE80::3, Serial1/3
OI 3::/64 [110/782]
    via FE80::3, Serial1/3
OI 3::3/128 [110/781]
    via FE80::3, Serial1/3
O 3:1::/64 [110/782]
    via FE80::3, Serial1/3
OI 12::/64 [110/1563]
    via FE80::3, Serial1/3
OI 23::/64 [110/1562]
    via FE80::3, Serial1/3

```

Task 6

Erase the startup configuration of all Routers and Switches and reload them before proceeding to the next lab.

CCIE Foundation

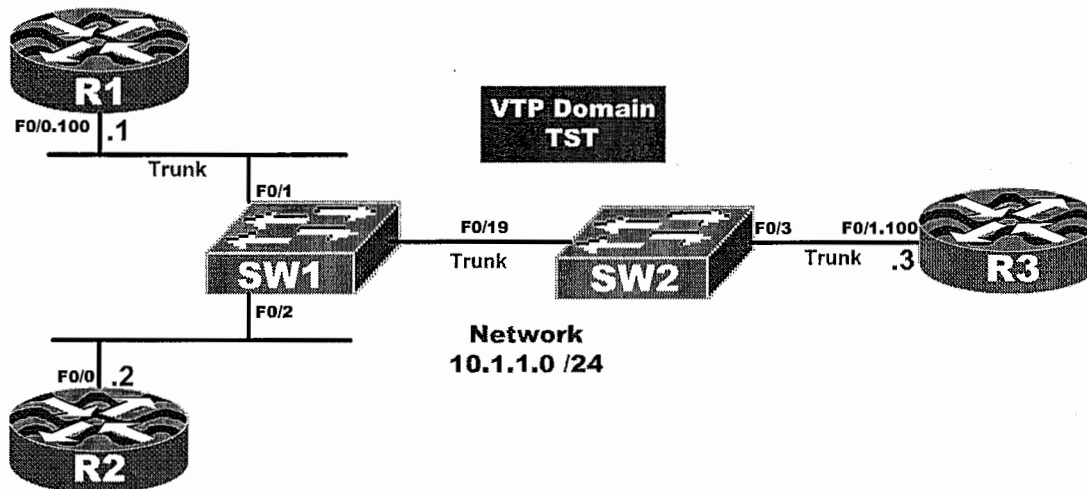
5.0

www.MicronicsTraining.com

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

QOS

Lab 1 – MLS QOS



Task 1

Configure the routers and the switches based on the above topology.

On SW1 and SW2:

```
SWx(config)#Int Range f0/1-24
SWx(config-if-range)#Shut
```

On SW1:

```
SW1(config)#Int F0/1
SW1(config-if-range)#Swi trunk encap dot1q
SW1(config-if-range)#Swi mode trunk
SW1(config-if-range)#No shut
```

```
SW1(config)#VTP domain TST
Changing VTP domain name from NULL to TST
```

```
SW1(config)#Int F0/2
SW1(config-if-range)#Swi mode acc
```

```
SW1(config-if-range)#Swt access vlan 100
SW1(config-if-range)#No shut
```

```
SW1(config)#Int F0/19
SW1(config-if)#Swt trunk encap dot
SW1(config-if)#Swt mode trunk
SW1(config-if)#No shu
```

On SW2:

```
SW2(config)#Int F0/3
SW2(config-if)#Swt trunk enc dot
SW2(config-if)#Swt mode trunk
SW2(config-if)#No shu
```

```
SW2(config)#Int F0/19
SW2(config-if)#Swt trunk enc dot
SW2(config-if)#Swt mode trunk
SW2(config-if)#No shut
```

On R1:

```
R1(config)#Int F0/0
R1(config-if)#No shut

R1(config-if)#Int F0/0.100
R1(config-subif)#Encap dot1q 100
R1(config-subif)#IP address 10.1.1.1 255.255.255.0
```

On R2:

```
R2(config)#Int F0/0
R2(config-if)#IP address 10.1.1.2 255.255.255.0
R2(config-if)#No shut
```

On R3:

```
R3(config)#Int F0/1
R3(config-if)#No shut

R3(config-if)#Int F0/1.100
R3(config-subif)#Encap dot1q 100
R3(config-subif)#IP address 10.1.1.3 255.255.255.0
```

To verify the configuration:

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

Task 2

Configure R1 to send all traffic with a CoS marking of 1. R3 should be used for verification purposes, R3 should be configured to match on CoS values of 0 – 7 ingress on its F0/1.100 sub-interface.

In this step, R3 is configured to match on incoming traffic with CoS values of 0 – 7, this is done so the policy can be tested and verified.

On R3:

R3(config)#class-map cos0

R3(config-cmap)#match cos 0

R3(config)#class-map cos1

R3(config-cmap)#match cos 1

R3(config)#class-map cos2

R3(config-cmap)#match cos 2

R3(config)#class-map cos3

R3(config-cmap)#match cos 3

R3(config)#class-map cos4

R3(config-cmap)#match cos 4

```
R3(config)#class-map cos5
R3(config-cmap)#match cos 5
```

```
R3(config)#class-map cos6
R3(config-cmap)#match cos 6
```

```
R3(config)#class-map cos7
R3(config-cmap)#match cos 7
```

```
R3(config)#policy-map TST
R3(config-pmap)#class cos0
R3(config-pmap)#class cos1
R3(config-pmap)#class cos2
R3(config-pmap)#class cos3
R3(config-pmap)#class cos4
R3(config-pmap)#class cos5
R3(config-pmap)#class cos6
R3(config-pmap)#class cos7
```

```
R3(config)#Int F0/1.100
R3(config-subif)#service-policy in TST
```

The following configures R1 to generate all traffic with a CoS value of 1:

On R1:

```
R1(config)#Policy-map tst
R1(config-pmap)#class class-default
R1(config-pmap-c)#set cos 1
```

```
R1(config-pmap-c)#Int F0/0.100
R1(config-subif)#Service-policy Out tst
```

To test the configuration:

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 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

On R3:

```
R3#Show policy-map Interface | S cos1
```

```
Class-map: cos1 (match-all)  
  5 packets, 590 bytes  
  5 minute offered rate 0 bps  
Match: cos 1
```

You can see that the test was successful. Let's verify the configuration of "MLS QoS" on SW1 and SW2:

On SW1:

```
SW1#Show mls qos
```

```
QoS is disabled
```

```
QoS ip packet dscp rewrite is enabled
```

On SW2:

```
SW2#Show mls qos
```

```
QoS is disabled
```

```
QoS ip packet dscp rewrite is enabled
```

The reason the test was successful is because "MLS QoS" is disabled, when MLS QoS is disabled, the switch does NOT drop or change the marking of incoming traffic.

Task 3

Enable "MLS QoS" on SW1 and ensure that the test in the previous task is still successful.

On SW1:

```
SW1(config)#MLS Qos
```

To verify the configuration:

On SW1:

```
SW1#Show mls qos
```

QoS is enabled

```
QoS ip packet dscp rewrite is enabled
```

When "MLS QoS" is enabled, the switch drops the marking of all incoming traffic, unless that marking was trusted on the interface through which the traffic was received.

If ONLY "mls qos" was configured on SW1, and R1 was generating traffic with a CoS value of "1", the switch (SW1) will drop the CoS marking and R3 will see all traffic generated by R1 with a CoS value of "0". If the F0/1 interface of SW1 is configured to trust CoS, when SW1 receives the traffic, it will NOT rewrite or drop the layer two marking. The switch will also consult the "CoS-to-DSCP" mapping. Since SW1's default "CoS-to-DSCP" mapping is NOT changed, and by default it maps CoS 0 to DSCP 0, R3 will also see a DSCP value of "0" in the ToS byte of all incoming traffic from R1.

On R3:

```
R3#Clear counters
```

```
Clear "show interface" counters on all interfaces [confirm]
```

Press Enter to confirm

To test the configuration:

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

On R3:

```
R3#Show policy-map Interface | S cos1
```

```
Class-map: cos1 (match-all)
```

```
0 packets, 0 bytes
```

```
5 minute offered rate 0 bps
```

```
Match: cos 1
```

NOTE: Because the marking was dropped by SW1, R3 no longer sees the traffic marked with CoS value of 1, but if the marking is dropped, it should match CoS 0, correct?

Let's verify:

On R3:

```
R3#Show policy-map Interface | S cos0
```

```
Class-map: cos0 (match-all)
  5 packets, 590 bytes
  5 minute offered rate 0 bps
  Match: cos 0
```

In order for SW1 to trust the CoS marking of incoming traffic, the switchport that R1 is connected to must trust the CoS marking in incoming traffic. Let's configure SW1's F0/1 interface to trust the CoS marking of all incoming traffic:

On SW1:

```
SW1(config)#Int F0/1
SW1(config-if)#MLS Qos trust cos
```

To test the configuration:

On R3:

```
R3#Clear counters
Clear "show interface" counters on all interfaces [confirm]
Press Enter to confirm
```

On R1:

```
R1#Ping 10.1.1.3 repeat 10
```

```
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:
!!!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 1/1/4 ms
```

On R3:

```
R3#Show policy-map Interface | S cos1
```

```
Class-map: cos1 (match-all)
  10 packets, 1180 bytes
  5 minute offered rate 0 bps
```

Match: cos 1

Task 4

Configure SW1's F0/2 interface such that it marks all ingress traffic with a CoS marking of 2, DO NOT configure MQC for this purpose.

On SW1:

The following command assigns a default CoS value of 2 to untagged traffic received through this interface.

```
SW1(config)#Int F0/2
SW1(config-if)#mls qos cos 2
```

To verify the configuration:

On SW1:

```
SW1#Show mls qos inter f0/2

FastEthernet0/2
trust state: not trusted
trust mode: not trusted
trust enabled flag: ena
COS override: dis
default COS: 2
DSCP Mutation Map: Default DSCP Mutation Map
Trust device: none
qos mode: port-based
```

To test the configuration:

On R2:

```
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 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms

To verify the test:

On R3:

```
R3#Show policy-map interface | s cos0
```

```
Class-map: cos0 (match-all)
  4 packets, 472 bytes
  5 minute offered rate 0 bps
  Match: cos 0
```

```
R3#Show policy-map interface | s cos2
```

```
Class-map: cos2 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps
  Match: cos 2
```

The "mls qos cos" command on its own does NOTHING. It should be combined with either "mls qos cos override" or "mls qos trust cos". When it's combined with "MLS qos trust cos", ONLY the untagged traffic is effected, but if it is combined with "mls qos cos override", then, all traffic (tagged or untagged) is effected. Tagged traffic is traffic that contains the VLAN ID.

NOTE: Even though the interface is configured with "mls qos cos 2" the traffic coming in on that interface is NOT effected. To mark ALL traffic with a CoS marking of 2, which means all traffic regardless of their markings, the port must be configured to override the existing CoS.

The following command configures the switch port to trust the CoS value in ALL incoming untagged traffic through F0/2 interface. The "mls qos cos override" command will be tested later:

```
SW1(config)#Int F0/2
SW1(config-if)#mls qos trust cos
```

To verify the configuration:

On SW1:

```
SW1#Sh mls qos interface f0/2
```

```
FastEthernet0/2
trust state: trust cos
trust mode: trust cos
```

```
trust enabled flag: ena
COS override: dis
default COS: 2
DSCP Mutation Map: Default DSCP Mutation Map
Trust device: none
qos mode: port-based
```

The output of the above command reveals that the CoS marking is trusted and the default CoS value is set to 2.

To test the configuration:

On R3:

```
R3#Clear counters
```

```
Clear "show interface" counters on all interfaces [confirm]
```

```
Press Enter to confirm
```

On R2:

```
R2#Ping 10.1.1.3 repeat 20
```

Type escape sequence to abort.

Sending 20, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

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

Success rate is 100 percent (20/20), round-trip min/avg/max = 1/1/4 ms

To verify the test:

On R3:

```
R3#Sh policy-map inter | s cos0
```

```
Class-map: cos0 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps
  Match: cos 0
```

```
R3#Show policy-map interface | s cos2
```

```
Class-map: cos2 (match-all)
  20 packets, 2360 bytes
  5 minute offered rate 0 bps
  Match: cos 2
```

The output of the above show command reveals that all traffic (in this case untagged) that is sourced from R2 is marked with a CoS value of 2. However, tagged traffic will retain its CoS marking.

To test tagged traffic, R2's F0/0 interface is configured as trunk and it's also configured to mark all egress traffic with a CoS value of 7:

On R2:

```
R2 (config) #Default interface F0/0

R2 (config) #Int F0/0.100
R2 (config-subif) #Encap dot1q 100
R2 (config-subif) #Ip addr 10.1.1.2 255.255.255.0

R2 (config) #Policy-map tst
R2 (config-pmap) #class class-default
R2 (config-pmap-c) #set CoS 7

R2 (config) #Int F0/0.100
R2 (config-subif) #Service-policy out tst
```

SW1's F0/2 interface should also be configured as trunk, or else R2 will NOT be able to communicate with R3:

On SW2:

```
SW1 (config) #Int F0/2
SW1 (config-if) #Swi trunk encap dot1q
SW1 (config-if) #Swi mode trunk
SW1 (config-if) #Mls qos trust cos
SW1 (config-if) #Mls qos cos 2
```

Let's test by pinging R3 and verifying the result:

On R3:

```
R3 #Clear counters
Clear "show interface" counters on all interfaces [confirm]
Press Enter to confirm
```

On R2:

```
R2 #Ping 10.1.1.3 rep 20
```

Type escape sequence to abort.
Sending 20, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:
.!!!!!!!!!!!!!!!!!!!!!!
Success rate is 95 percent (19/20), round-trip min/avg/max = 1/1/4 ms

On R3:

```
R3#Show policy-map Interface | S cos2
```

```
Class-map: cos2 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps
Match: cos 2
```

```
R3#Show policy-map Interface | S cos0
```

```
Class-map: cos0 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps
Match: cos 0
```

```
R3#Show policy-map Interface | S cos7
```

```
Class-map: cos7 (match-all)
  19 packets, 2242 bytes
  5 minute offered rate 0 bps
Match: cos 7
```

You can see that the tagged traffic retained its marking, and it was NOT affected by the configuration of SW1's F0/2 interface. To see the configuration of SW1's F0/2 interface:

On SW1:

```
SW1#Show run int f0/2 | B interface
```

```
interface FastEthernet0/2
  switchport trunk encapsulation dot1q
  switchport mode trunk
  mls qos cos 2
  mls qos trust cos
end
```

Task 5

SW2 should be configured such that it marks all traffic ingress traffic from SW1 (tagged or untagged) with a CoS value of 5. DO NOT configure R1, R2, or SW1 to accomplish this task.

Let's test the existing condition:

On R3:

R3#**Clear counters**

Clear "show interface" counters on all interfaces [confirm]

Press Enter to confirm

On SW2:

SW2 (config)#**mls qos**

SW2 (config)#**int F0/19**

SW2 (config-if)#**Mls qos trust cos**

To test the configuration:

On R1:

R1#**Ping 10.1.1.3 repeat 10**

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

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

Success rate is 100 percent (10/10), round-trip min/avg/max = 1/1/4 ms

On R2:

R2#**Ping 10.1.1.3 repeat 20**

Type escape sequence to abort.

Sending 20, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

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

Success rate is 100 percent (20/20), round-trip min/avg/max = 1/1/4 ms

To test the configuration:

On R3:

We can see that R1's traffic matched correctly to CoS value of 1:

```
R3#Show policy-map Interface | S cos1
```

```
Class-map: cos1 (match-all)
  10 packets, 1180 bytes
  5 minute offered rate 0 bps
  Match: cos 1
```

We can also see that the traffic from R2 matched correctly to CoS value of 7:

```
R3#Show policy-map Interface | S cos7
```

```
Class-map: cos7 (match-all)
  20 packets, 2360 bytes
  5 minute offered rate 0 bps
  Match: cos 7
```

Let's configure all traffic tagged or untagged traffic with a CoS value of 5:

On SW2:

```
SW2(config)#Mls qos
SW2(config)#Int F0/19
SW2(config-if)#Mls qos cos 5
SW2(config-if)#Mls qos cos override
```

To test and verify the configuration:

On R3:

```
R3#Clear counters
Clear "show interface" counters on all interfaces [confirm]
Press Enter to confirm
```

On R1:

```
R1#Ping 10.1.1.3 rep 60
```

```
Type escape sequence to abort.
Sending 60, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (60/60), round-trip min/avg/max = 1/1/4 ms
```

On R2:

```
R2#Ping 10.1.1.3 rep 70
```

Type escape sequence to abort.

Sending 70, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
Success rate is 100 percent (70/70), round-trip min/avg/max = 1/1/4 ms
```

On R3:

```
R3#Show policy-map interface | s cos1|cos7|cos5
```

```
Class-map: cos1 (match-all)  
  0 packets, 0 bytes  
  5 minute offered rate 0 bps  
  Match: cos 1
```

```
Class-map: cos5 (match-all)  
  130 packets, 15340 bytes  
  5 minute offered rate 0 bps  
  Match: cos 5
```

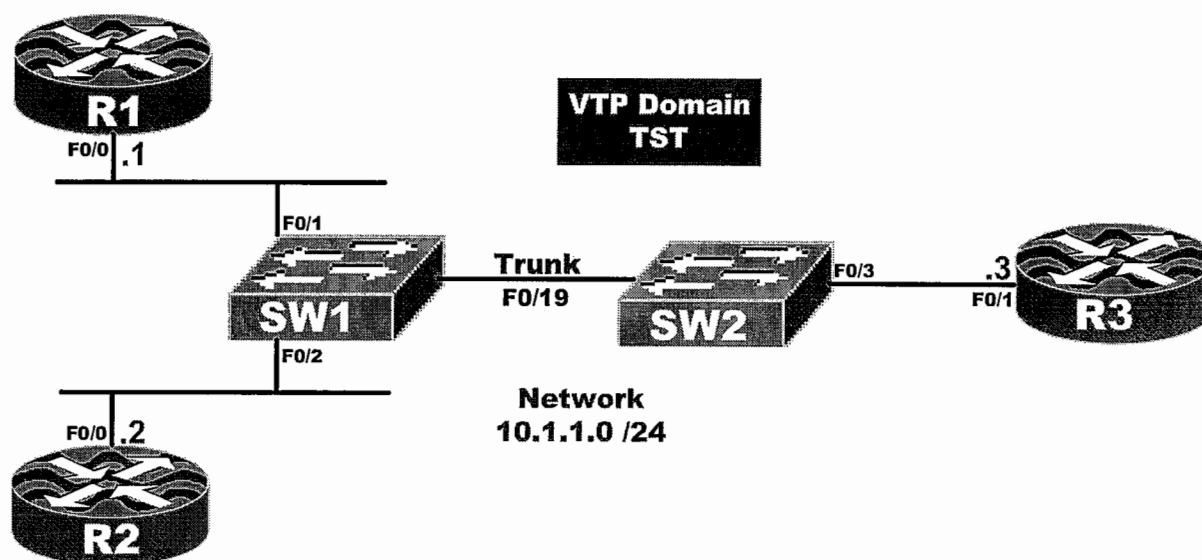
```
Class-map: cos7 (match-all)  
  0 packets, 0 bytes  
  5 minute offered rate 0 bps  
  Match: cos 7
```

NOTE: All traffic regardless of their markings are marked with a CoS value of 5.

Task 6

Erase the startup configuration On R1:-3 and SW1 & SW2 and reload these routers and switches before proceeding to the next lab.

Lab 2 – DSCP-Mutation



Task 1

Configure the routers and the switches based on the above topology.

On SW1 and SW2:

```
SWx(config)#Int Range f0/1-24  
SWx(config-if-range)#Shut
```

On SW1:

```
SW1(config)#Int range F0/1-2  
SW1(config-if-range)#Swt mode acc  
SW1(config-if-range)#Swt access vlan 100  
SW1(config-if-range)#No shut
```

```
SW1(config)#Int F0/19  
SW1(config-if)#Swt trunk encap dot  
SW1(config-if)#Swt mode trunk  
SW1(config-if)#No shu
```


SW1(config)#VTP domain TST
Changing VTP domain name from NULL to TST

On SW2:

```
SW2(config)#Int F0/3
SW2(config-if)#Swt mode acc
SW2(config-if)#Swt acc vlan 100
SW2(config-if)#No shu

SW2(config)#Int F0/19
SW2(config-if)#Swt trunk enc dot
SW2(config-if)#Swt mode trunk
SW2(config-if)#No shut
```

On R1:

```
R1(config)#Int F0/0
R1(config-subif)#IP address 10.1.1.1 255.255.255.0
R1(config-if)#No shut
```

On R2:

```
R2(config)#Int F0/0
R2(config-if)#IP address 10.1.1.2 255.255.255.0
R2(config-if)#No shut
```

On R3:

```
R3(config)#Int F0/1
R3(config-subif)#IP address 10.1.1.3 255.255.255.0
R3(config-if)#No shut
```

To verify the configuration:

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

Task 2

Configure an MQC on R1 such that all packets going out of its F0/0 interface are marked with a DSCP value of 1. For verification purposes, R3's F0/1 interface should be configured to match on DSCP values of 0-7 for all ingress traffic. Ensure that "mls qos" is disabled on both switches.

On both switches:

```
SWx#Sh mls qos
```

QoS is disabled

```
QoS ip packet dscp rewrite is enabled
```

The following configuration marks all egress traffic with a DSCP value of 1:

On R1:

```
R1(config)#Policy-map TST
R1(config-pmap)#class class-default
R1(config-pmap-c)#set ip dscp 1
```

```
R1(config)#Int F0/0
R1(config-if)#Service-policy out TST
```

On R3:

The following configuration is done for verification and testing purposes:

```
R3(config)#class-map dscp0
R3(config-cmap)#match ip dscp 0

R3(config)#class-map dscp1
R3(config-cmap)#match ip dscp 1
```

```

R3(config)#class-map dscp2
R3(config-cmap)#match ip dscp 2

R3(config)#class-map dscp3
R3(config-cmap)#match ip dscp 3

R3(config)#class-map dscp4
R3(config-cmap)#match ip dscp 4

R3(config)#class-map dscp5
R3(config-cmap)#match ip dscp 5

R3(config)#class-map dscp6
R3(config-cmap)#match ip dscp 6

R3(config)#class-map dscp7
R3(config-cmap)#match ip dscp 7

R3(config)#policy-map TST
R3(config-pmap)#class dscp0
R3(config-pmap)#class dscp1
R3(config-pmap)#class dscp2
R3(config-pmap)#class dscp3
R3(config-pmap)#class dscp4
R3(config-pmap)#class dscp5
R3(config-pmap)#class dscp6
R3(config-pmap)#class dscp7

R3(config)#Int F0/1
R3(config-if)#Service-policy in TST

```

To test the configuration:

On R1:

```
R1#Ping 10.1.1.3 rep 10
```

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

.!!!!!!!

Success rate is 90 percent (9/10), round-trip min/avg/max = 1/1/4 ms

On R3:

```
R3#Sh policy-map inter | s dscp1
```

```
Class-map: DSCP1 (match-all)
  9 packets, 1026 bytes
  5 minute offered rate 0 bps
Match: ip dscp 1
```

Task 3

Configure SW2 such that if the incoming traffic is marked with a DSCP value of 1, it is overwritten to a DSCP value of 60. DO NOT configure a class-map or a policy-map to accomplish this task. Use R3 to verify the configuration.

DSCP-Mutation can be configured on SW2 to accomplish this task; there are four steps in configuring DSCP-mutation, and they are as follows:

Step 1:

Mls qos MUST be enabled:

On SW2:

```
SW2(config)#mls qos
```

Remember, once the "mls qos" is enabled, the marking of all traffic is zeroed out, meaning that incoming traffic that is marked with any DSCP value will match to DSCP value of "0" on R3. The following proves this point:

On SW2:

```
SW2#Show mls qos
```

```
QoS is enabled
```

```
QoS ip packet dscp rewrite is enabled
```

To test the configuration:

On R3:

```
R3#Clear counters
```

```
Clear "show interface" counters on all interfaces [confirm]
```

Press Enter to confirm

On R1:

```
R1#Ping 10.1.1.3 rep 100
```

Type escape sequence to abort.

Sending 100, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

!!
!!

Success rate is 100 percent (100/100), round-trip min/avg/max = 1/1/4 ms

```
R3#Sh policy-map inter | s dscp1
```

```
Class-map: DSCP1 (match-all)  
  0 packets, 0 bytes  
  5 minute offered rate 0 bps  
  Match: dscp 1
```

```
R3#Sh policy-map inter | s dscp0
```

```
Class-map: DSCP0 (match-all)  
  100 packets, 11400 bytes  
  5 minute offered rate 2000 bps  
  Match: dscp default (0)
```

Step 2:

Remember, if “mls qos trust dscp” is NOT configured; the configuration will NOT have any effect on the packets, because SW2 will drop the marking of the incoming traffic. To verify this information:

On SW2:

```
SW2(config)#int f0/19  
SW2(config-if)#mls qos trust dscp
```

To verify this information:

On SW2:

```
SW2#Show mls qos int f0/19 | inc trust state
```

```
trust state: trust dscp
```

NOTE: If CoS was trusted, the output of the above command would have stated “trust state: trust CoS”, since ONLY DSCP is trusted, the trust state is DSCP.

To test this information:

On R3:

R3#**Clear counters**

Clear "show interface" counters on all interfaces [confirm]

Press Enter to confirm

On R1:

R1#**Ping 10.1.1.3 rep 100**

Type escape sequence to abort.

Sending 100, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

!!
!!

Success rate is 100 percent (100/100), round-trip min/avg/max = 1/1/4 ms

On R3:

R3#**Sh policy-map inter | s dscp0**

```
Class-map: DSCP0 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps
  Match: dscp default (0)
```

R3#**Sh policy-map inter | s dscp1**

```
Class-map: DSCP1 (match-all)
  100 packets, 11400 bytes
  5 minute offered rate 0 bps
  Match: dscp 1
```

Step 3:

In this step, a custom DSCP-Mutation map is configured. Remember that if this custom mapping is NOT configured, the default DSCP-Mutation map will be used.

The default DSCP-Mutation map cannot be changed and it is configured as one to one, meaning that the DSCP marking in the incoming traffic will always match the DSCP marking in the outgoing traffic.

In this step, a custom DSCP-Mutation map named TST is configured.

This custom DSCP-Mutation map maps the incoming DSCP value (in this case 1) to an outgoing DSCP value of 60:

To see the default DSCP-Mutation map:

```
SW2#Show mls qos map dscp-mutation
```

Dscp-dscp mutation map:

Default DSCP Mutation Map:

d1 :	d2	0	1	2	3	4	5	6	7	8	9
0 :		01	02	03	04	05	06	07	08	09	
1 :		10	11	12	13	14	15	16	17	18	19
2 :		20	21	22	23	24	25	26	27	28	29
3 :		30	31	32	33	34	35	36	37	38	39
4 :		40	41	42	43	44	45	46	47	48	49
5 :		50	51	52	53	54	55	56	57	58	59
6 :		60	61	62	63						

Note the "d1:" column (highlighted in yellow) specifies the most significant digit of the DSCP value of incoming packets, whereas, the "d2:" row (highlighted in blue) specifies the least significant digit of the DSCP value in incoming packets.

The intersection of the "d1" and "d2" values (This is the body of the output) provides the DSCP value of the outgoing packets.

NOTE: The output of the above show command reveals that by default, the incoming DSCP value of 1, is re-written to the outgoing DSCP value of 1.

Let's configure a custom DSCP-Mutation map called "TST" that maps the incoming DSCP value of 1 to an outgoing DSCP value of 60:

```
SW2(config)#mls qos map dscp-mutation TST 1 to 60
```

To verify the configuration:

On SW2:

```
SW2#Show mls qos map dscp-mutation TST
```

Dscp-dscp mutation map:

TST:

d1 :	d2	0	1	2	3	4	5	6	7	8	9
0 :		60	02	03	04	05	06	07	08	09	
1 :		10	11	12	13	14	15	16	17	18	19

```

2 :    20 21 22 23 24 25 26 27 28 29
3 :    30 31 32 33 34 35 36 37 38 39
4 :    40 41 42 43 44 45 46 47 48 49
5 :    50 51 52 53 54 55 56 57 58 59
6 :    60 61 62 63

```

Once the custom DSCP-Mutation map is configured, it must be applied to the F0/19 interface (trunk interface) of SW2.

```

SW2(config)#Int f0/19
SW2(config-if)#Mls qos dscp-mutation TST

```

To verify the configuration:

On SW2:

```
SW2#Show mls qos int f0/19 | inc DSCP
```

```
DSCP Mutation Map: TST
```

Step 4:

In the final step of this configuration, you MUST ensure that DSCP re-writes are enabled. If this is disabled, then the DSCP marking will NOT be re-written. Let's verify this information:

To verify if the DSCP re-writes are enabled:

On SW2:

```
SW2#Show mls qos
```

```
QoS is enabled
```

```
QoS ip packet dscp rewrite is enabled
```

By default, DSCP rewrites are enabled ONLY if "mls qos" is enabled, I recommend entering the following command and NOT to rely on memory to check to see if it is enabled:

```
SW2(config)#Mls qos rewrite ip dscp
```

Let's test and see if the incoming DSCP value of 1 is rewritten to a DSCP value of 60: Remember that on R3 we need to match on DSCP 60 so we can test and verify the configuration:

On R3:

```

R3(config)#Class-map dscp60
R3(config-cmap)#match ip dscp 60

```



```
R3(config)#Policy-map TST
R3(config-pmap)#class dscp60
```

When we add class/es to an existing Policy-map, the new class/es are added to the end of all the classes within that policy-map. To see this:

On R3:

```
R3#Show policy-map TST
```

```
Policy Map TST
  Class dscp0
  Class dscp1
  Class dscp2
  Class dscp3
  Class dscp4
  Class dscp5
  Class dscp6
  Class dscp7
  Class dscp60
```

To test the configuration:

On R3:

```
R3#Clear counters
Clear "show interface" counters on all interfaces [confirm]
Press Enter to confirm
```

On R1:

```
R1#Ping 10.1.1.3 rep 60
```

```
Type escape sequence to abort.
Sending 60, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (60/60), round-trip min/avg/max = 1/1/4 ms
```

On R3:

```
R3#Sh policy-map inter | s dscp60

Class-map: DSCP60 (match-all)
  60 packets, 6840 bytes
```

5 minute offered rate 2000 bps
Match: ip dscp 60

Let's disable the rewrites and verify and test the configuration:

On SW2:

SW2(config)#No mls qos rewrite ip dscp

To verify the configuration:

On SW2:

SW2#Show mls qos

QoS is enabled

QoS ip packet dscp rewrite is disabled

To test the configuration:

On R3:

R3#Clear counters

Clear "show interface" counters on all interfaces [confirm]

Press Enter to confirm

On R1:

R1#Ping 10.1.1.3 rep 10

Type escape sequence to abort.

Sending 100, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

!!!!!!!!!!

Success rate is 100 percent (10/10), round-trip min/avg/max = 1/1/4 ms

On R3:

R3#Sh policy-map inter | s dscp1

Class-map: DSCP1 (match-all)

100 packets, 11400 bytes

5 minute offered rate 0 bps

Match: dscp 1

NOTE: The incoming DSCP value was NOT modified.

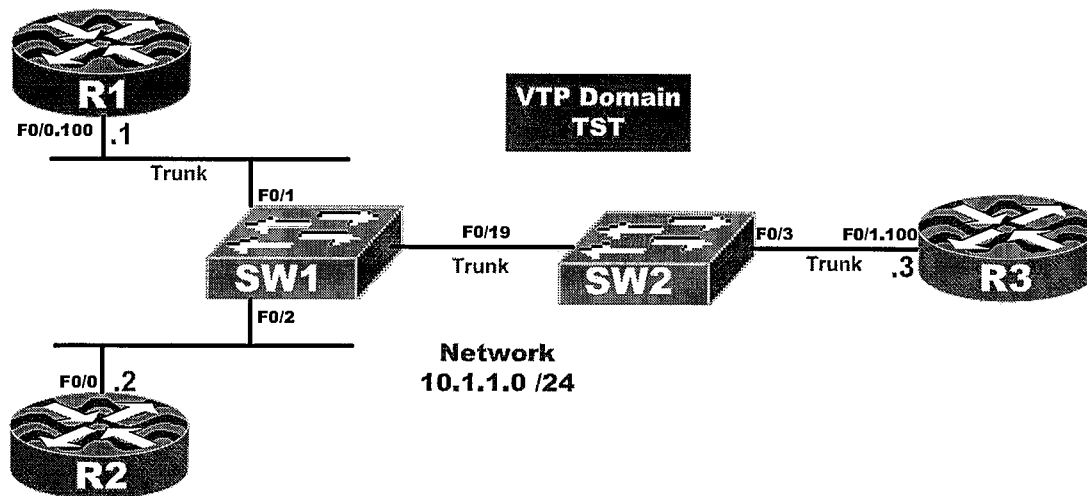
On SW2:

```
SW2 (config) #mls qos rewrite ip dscp
```

Task 4

Erase the startup configuration on R1-3 and SW1 & SW2 and reload these routers and switches before proceeding to the next lab.

Lab 3 – DSCP-CoS Mapping



Task 1

Configure the routers and the switches based on the above topology.

On SW1 and SW2:

```
SWx(config)#Int Range f0/1-24  
SWx(config-if-range)#Shut
```

On SW1:

```
SW1(config)#Int F0/1  
SW1(config-if-range)#Swi trunk encap dot1q  
SW1(config-if-range)#Swi mode trunk  
SW1(config-if-range)#No shut
```

```
SW1(config)#Int F0/2  
SW1(config-if-range)#Swi mode acc  
SW1(config-if-range)#Swi access vlan 100  
SW1(config-if-range)#No shut
```

```
SW1(config)#Int F0/19
SW1(config-if)#Swt trunk encap dot

SW1(config-if)#Swt mode trunk
SW1(config-if)#No shu

SW1(config)#VTP domain TST
Changing VTP domain name from NULL to TST
```

On SW2:

```
SW2(config)#Int F0/3
SW2(config-if)#Swt trunk enc dot
SW2(config-if)#Swt mode trunk
SW2(config-if)#No shu

SW2(config)#Int F0/19
SW2(config-if)#Swt trunk enc dot
SW2(config-if)#Swt mode trunk
SW2(config-if)#No shut
```

On R1:

```
R1(config)#Int F0/0
R1(config-if)#No shut

R1(config-if)#Int F0/0.100
R1(config-subif)#Encap dot1q 100
R1(config-subif)#IP address 10.1.1.1 255.255.255.0
```

On R2:

```
R2(config)#Int F0/0
R2(config-if)#IP address 10.1.1.2 255.255.255.0
R2(config-if)#No shut
```

On R3:

```
R3(config)#Int F0/1
R3(config-if)#No shut

R3(config-if)#Int F0/1.100
R3(config-subif)#Encap dot1q 100
R3(config-subif)#IP address 10.1.1.3 255.255.255.0
```

To verify the configuration:

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

Task 2

For testing and verification of this lab, configure R3 to match on incoming CoS markings of 0-7 using an MQC; this policy should be applied inbound to R3's F0/1.100 sub-interface.

On R3:

R3(config)#**class-map cos0**
R3(config-cmap)#**match cos 0**

R3(config)#**class-map cos1**
R3(config-cmap)#**match cos 1**

R3(config)#**class-map cos2**
R3(config-cmap)#**match cos 2**

R3(config)#**class-map cos3**
R3(config-cmap)#**match cos 3**

R3(config)#**class-map cos4**
R3(config-cmap)#**match cos 4**

R3(config)#**class-map cos5**

```

R3(config-cmap)#match cos 5

R3(config)#class-map cos6
R3(config-cmap)#match cos 6

R3(config)#class-map cos7
R3(config-cmap)#match cos 7

R3(config)#Policy-map TST
R3(config-pmap)#class cos0
R3(config-pmap)#class cos1
R3(config-pmap)#class cos2
R3(config-pmap)#class cos3
R3(config-pmap)#class cos4
R3(config-pmap)#class cos5
R3(config-pmap)#class cos6
R3(config-pmap)#class cos7

R3(config)#int f0/1.100
R3(config-subif)#service-policy in TST

```

Task 3

Configure R1 such that it marks all outgoing traffic with a DSCP value of 5.

On R1:

```

R1(config)#Policy-map TST
R1(config-pmap)#class class-default
R1(config-pmap-c)#set ip dscp 5

R1(config)#Int F0/0
R1(config-if)#service-policy out TST

```

Task 4

Configure SW2 such that it maps the DSCP value of 5 in incoming packets to a CoS value of 6.

Before configuring this task, the default "DSCP-CoS" mapping should be displayed, using the following command:

On SW2:

```
SW2#Sh mls qos map dscp-cos
```

```
Dscp-cos map:
d1 : d2 0  1  2  3  4  5  6  7  8  9
-----
0 :    00 00 00 00 00 00 00 00 01 01
1 :    01 01 01 01 01 01 02 02 02 02
2 :    02 02 02 02 03 03 03 03 03 03
3 :    03 03 04 04 04 04 04 04 04 04
4 :    05 05 05 05 05 05 05 05 06 06
5 :    06 06 06 06 06 06 07 07 07 07
6 :    07 07 07 07
```

Note the output of the above show command displays the default DSCP to CoS mapping, it means that if the "mls qos trust dscp" command is configured, then the DSCP marking of incoming packets are mapped to outgoing CoS value according to the DSCP-CoS map.

The incoming DSCP values are shown in the "d1" column and the "d2" row, whereas, the outgoing CoS values are identified in the body of this display, this is the intersection of the "d1" column and the "d2" row.

NOTE: Every eight DSCP values are mapped to a single CoS value. This mapping can affect the entire switch, and a custom mapping cannot be configured.

By default, an incoming DSCP value of 5 is mapped to an outgoing CoS value of 0. To accomplish this task, we have to modify this mapping so the incoming DSCP value of 5 is mapped to an outgoing CoS value of 6.

To test & verify the default configuration:

The following command **MUST** be configured, so the incoming DSCP values are trusted. If this is NOT configured, the incoming DSCP values will NOT be mapped to an outgoing CoS value.

On SW2:

```
SW2(config)#Mls qos
```

```
SW2(config)#int f0/19
```

```
SW2(config-if)#mls qos trust dscp
```


To verify the configuration:

On SW2:

```
SW2#Show mls qos int f0/19 | inc trust state
```

```
trust state: trust dscp
```

To test the configuration:

A ping is generated from R1 with a repeat count of 60 and verified On R3:

On R1:

```
R1#Ping 10.1.1.3 repeat 60
```

Type escape sequence to abort.

Sending 60, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

.....

Success rate is 98 percent (59/60), round-trip min/avg/max = 1/1/4 ms

To verify the configuration:

On R3:

```
R3#Sh policy-map interface | s cos0
```

```
Class-map: COS0 (match-all)
```

```
59 packets, 6962 bytes
```

```
5 minute offered rate 0 bps
```

```
Match: cos 0
```

Note: DSCP 5 is mapped to a CoS value of 0, this is because of the default mapping that is in use.

In the next step the default "DSCP-CoS" mapping is changed to map an incoming DSCP value of 5 to an outgoing CoS value of 6:

On SW2:

```
SW2(config)#Mls qos map dscp-cos 5 to 6
```

NOTE: The first value (5) is the DSCP value in the incoming packets and the second value (6) is the CoS value in the outgoing packets.

To test & verify the configuration:

On R3:

R3#**Clear counters**

Clear "show interface" counters on all interfaces [confirm]

Press Enter to confirm

On R1:

R1#**Ping 10.1.1.3 repeat 60**

Type escape sequence to abort.

Sending 60, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:

!!

Success rate is 100 percent (60/60), round-trip min/avg/max = 1/1/4 ms

On R3:

NOTE: Incoming packets that have a marking of DSCP 5 are mapped to an outgoing CoS value of 6 and NOT 0:

R3#**Sh policy-map interface | s COS0**

```
Class-map: COS0 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps
  Match: cos 0
```

R3#**Sh policy-map interface | s COS6**

```
Class-map: COS6 (match-all)
  60 packets, 7080 bytes
  5 minute offered rate 0 bps
  Match: cos 6
```

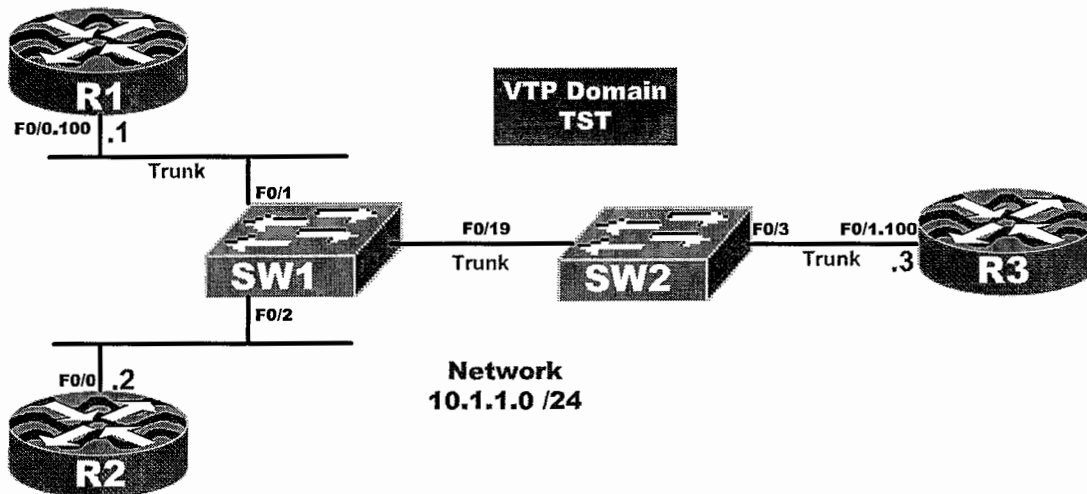
IMPORTANT POINT:

THE DSCP VALUE IS NOT REWRITTEN TO A COS VALUE, WHEN THE SWITCH SEES THE DSCP VALUE, IT SETS THE COS VALUE ACCORDING TO THE CONFIGURATION. THIS MEANS THAT THE LAYER THREE HEADER WILL RETAIN ITS DSCP VALUE, BUT THE LAYER TWO HEADER WILL BE MARKED WITH THE CONFIGURED COS VALUE.

Task 5

Erase the startup configuration On R1-3 and SW1 & SW2 and reload these routers and switches before proceeding to the next lab.

Lab 4 – CoS-DSCP Mapping



Task 1

Configure the routers and the switches based on the above topology.

On SW1 and SW2:

```
SWx(config)#Int Range f0/1-24
SWx(config-if-range)#Shut
```

On SW1:

```
SW1(config)#Int F0/1
SW1(config-if-range)#Swi trunk encap dot1q
SW1(config-if-range)#Swi mode trunk
SW1(config-if-range)#No shut

SW1(config)#Int F0/2
SW1(config-if-range)#Swi mode acc
SW1(config-if-range)#Swi access vlan 100
SW1(config-if-range)#No shut

SW1(config)#Int F0/19
```

```
SW1(config-if)#Swt trunk encap dot
SW1(config-if)#Swt mode trunk
SW1(config-if)#No shu
```

```
SW1(config)#VTP domain TST
Changing VTP domain name from NULL to TST
```

On SW2:

```
SW2(config)#Int F0/3
SW2(config-if)#Swt trunk enc dot
SW2(config-if)#Swt mode trunk
SW2(config-if)#No shu
```

```
SW2(config)#Int F0/19
SW2(config-if)#Swt trunk enc dot
SW2(config-if)#Swt mode trunk
SW2(config-if)#No shut
```

On R1:

```
R1(config)#Int F0/0
R1(config-if)#No shut

R1(config-if)#Int F0/0.100
R1(config-subif)#Encap dot1q 100
R1(config-subif)#IP address 10.1.1.1 255.255.255.0
```

On R2:

```
R2(config)#Int F0/0
R2(config-if)#IP address 10.1.1.2 255.255.255.0
R2(config-if)#No shut
```

On R3:

```
R3(config)#Int F0/1
R3(config-if)#No shut

R3(config-if)#Int F0/1.100
R3(config-subif)#Encap dot1q 100
R3(config-subif)#IP address 10.1.1.3 255.255.255.0
```

To verify the configuration:

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

Task 2

Configure the F0/0 interface of R2 to match the incoming traffic with markings of DSCP 0 – 7 and 56 using an MQC.

On R2:

R2(config)#**class-map D0**

R2(config-cmap)#**match ip dscp 0**

R2(config-cmap)#**class-map d1**

R2(config-cmap)#**match ip dscp 1**

R2(config-cmap)#**class-map d2**

R2(config-cmap)#**match ip dscp 2**

R2(config-cmap)#**class-map d3**

R2(config-cmap)#**match ip dscp 3**

R2(config-cmap)#**class-map d4**

R2(config-cmap)#**match ip dscp 4**

R2(config-cmap)#**class-map d5**

R2(config-cmap)#**match ip dscp 5**

```
R2(config-cmap)#class-map d6
R2(config-cmap)#match ip dscp 6

R2(config-cmap)#class-map d7
R2(config-cmap)#match ip dscp 7

R2(config-cmap)#class-map d56
R2(config-cmap)#match ip dscp 56

R2(config)#policy-map TST
R2(config-pmap)#class d0
R2(config-pmap-c)#class d1
R2(config-pmap-c)#class d2
R2(config-pmap-c)#class d3
R2(config-pmap-c)#class d4
R2(config-pmap-c)#class d5
R2(config-pmap-c)#class d6
R2(config-pmap-c)#class d7
R2(config-pmap-c)#class d56

R2(config-pmap-c)#int f0/0
R2(config-if)#service-policy in TST
```

Task 3

Configure the F0/3 interface of SW2 so that it marks all incoming traffic with a cos value of 7 regardless of existing markings; this should apply to tagged and untagged traffic.

On SW2:

```
SW2(config)#Mls qos

SW2(config)#int f0/3
SW2(config-if)#mls qos cos 7
SW2(config-if)#mls qos cos override
```

Task 4

Configure SW1 so that it maps the CoS value of 7 in incoming packets to a DSCP value of 7 in the outgoing packets.

Before configuring this task we should display the default mapping. The following command reveals the default mapping of CoS to DSCP:

On SW1:

```
SW1#Show mls qos map cos-dscp
```

```
Cos-dscp map:
```

```
cos:    0   1   2   3   4   5   6   7 ← the Cos value of incoming packets
```

```
-----  
dscp:    0   8  16  24  32  40  48  56 ← the DSCP value of outgoing packets
```

NOTE: The CoS value of 0 is mapped to an outgoing DSCP value of 0, and an incoming CoS value of 1 is mapped to a DSCP value of 8, and so forth.

To test this task, R3 pings R2. Since the switchport that R3 is connected to is configured to assign a CoS value of 7 to all incoming traffic, it should be matched to a DSCP value of 56 on R2. Before this can be accomplished, "mls qos" must be enabled at SW1, and the F0/20 interface of SW1 MUST be configured to trust CoS:

On SW1:

```
SW1(config)#mls qos
```

```
SW1(config)#int f0/20
```

```
SW1(config-if)#mls qos trust cos
```

To verify the configuration:

On SW1:

```
SW1#Show mls qos
```

```
QoS is enabled
```

```
QoS ip packet dscp rewrite is enabled
```

```
SW1#Show mls qos int f0/20 | inc trust state
```


trust state: trust cos

To test the configuration:

On R3:

R3#Ping 10.1.1.2 rep 70

Type escape sequence to abort.

Sending 70, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

.!!

Success rate is 98 percent (69/70), round-trip min/avg/max = 1/1/4 ms

On R2:

R2#Sh policy-map interface | s d56

```
Class-map: D56 (match-all)
  69 packets, 7866 bytes
  5 minute offered rate 0 bps
  Match: ip dscp cs7 (56)
```

To configure the task:

To change the default mapping, SW1 should be configured as follows:

On SW1:

CoS	DSCP
CoS 0	0
CoS 1	8
CoS 2	16
CoS 3	24
CoS 4	32
CoS 5	40
CoS 6	48
CoS 7	7

SW1(config)#mls qos map cos-dscp 0 8 16 24 32 40 48 7

When configuring this command, you MUST configure all 8 DSCP values; the values entered are the DSCP values, the position of the DSCP values identifies the CoS values. The first DSCP value (0) maps to CoS value of 0, and the second DSCP value (8) entered maps to CoS value of 1, the third DSCP value (16) maps to CoS value of 2, the fourth DSCP value (24) maps to CoS 3, the fifth DSCP value (32) maps to CoS value of 4, the sixth DSCP value (40) maps to CoS value of 5, the seventh DSCP value (48) maps to CoS value of 6 and the last DSCP value of 7 is mapped to a CoS value of 7.

To test and verify the configuration:

On R2:

R2#Clear counters

Clear "show interface" counters on all interfaces [confirm]

Press Enter to confirm

On R3:

```
R3#Ping 10.1.1.2 rep 70
```

Type escape sequence to abort.

Sending 70, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!

Success rate is 100 percent (70/70), round-trip min/avg/max = 1/1/4 ms

On R2:

```
R2#Show policy-map inter | s d56
```

```
Class-map: D56 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps
  Match: ip dscp cs7 (56)
```

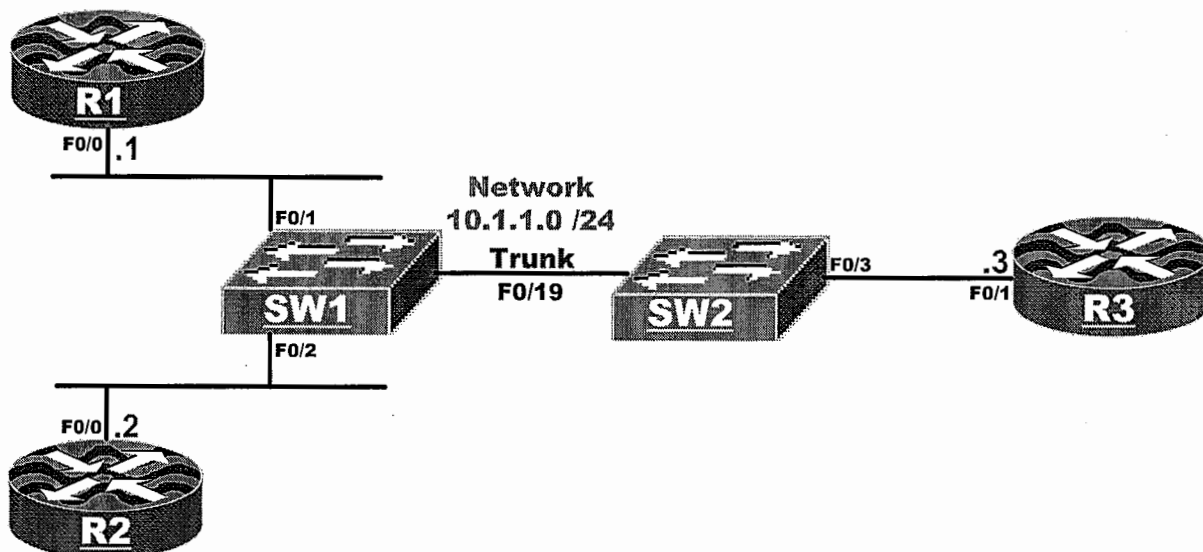
```
R2#Show policy-map inter | s d7
```

```
Class-map: D7 (match-all)
  70 packets, 7980 bytes
  5 minute offered rate 0 bps
  Match: ip dscp 7
```

Task 5

Erase the startup configuration On R1:3 and SW1 & SW2 and reload these routers and switches before proceeding to the next lab.

Lab 5 – IP-Precedence-DSCP Mapping



Task 1

Configure the routers and the switches based on the above topology.

On SW1 and SW2:

```
SWx(config)#Int Range f0/1-24  
SWx(config-if-range)#Shut
```

On SW1:

```
SW1(config)#Int range F0/1-2  
SW1(config-if-range)#Swt mode acc  
SW1(config-if-range)#Swt access vlan 100  
SW1(config-if-range)#No shut
```

```
SW1(config)#Int F0/19  
SW1(config-if)#Swt trunk encap dot  
SW1(config-if)#Swt mode trunk  
SW1(config-if)#No shu
```

On SW2:

```
SW2(config)#Int F0/3
SW2(config-if)#Swi mode acc
SW2(config-if)#Swi acc vlan 100
SW2(config-if)#No shu

SW2(config)#Int F0/19
SW2(config-if)#Swi trunk enc dot
SW2(config-if)#Swi mode trunk
SW2(config-if)#No shut
```

On R1:

```
R1(config)#Int F0/0
R1(config-subif)#IP address 10.1.1.1 255.255.255.0
R1(config-if)#No shut
```

On R2:

```
R2(config)#Int F0/0
R2(config-if)#IP address 10.1.1.2 255.255.255.0
R2(config-if)#No shut
```

On R3:

```
R3(config)#Int F0/1
R3(config-subif)#IP address 10.1.1.3 255.255.255.0
R3(config-if)#No shut
```

To verify the configuration:

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

Task 2

Configure SW1 so that it overwrites the IP Precedence value of 5 in incoming packets to a DSCP value of 5 in the outgoing packets. Configure R1 to match on DSCP markings of 0, 40, and 5 in incoming packets; this should be done for testing purpose.

On R1:

```
R1(config)#Class-map d0
R1(config-cmap)#match ip dscp 0

R1(config)#Class-map d40
R1(config-cmap)#match ip dscp 40

R1(config)#Class-map d5
R1(config-cmap)#match ip dscp 5

R1(config)#Policy-map TST
R1(config-pmap)#class d0
R1(config-pmap)#class d40
R1(config-pmap)#class d5

R1(config)#Int F0/0
R1(config-if)#Service-policy in TST
```

Before this task is configured the default IP-Precedence to DSCP mapping should be examined:

On SW1:

```
SW1#Show mls qos map ip-prec-dscp
```

```
IpPrecedence-dscp map:
```

```
  ipprec:   0  1  2  3  4  5  6  7  ← the IPP value of incoming packets
```

```
-----
```

```
  dscp:   0  8 16 24 32 40 48 56 ← the DSCP value of Outgoing packets
```

Note the IP-Precedence value of 0 is mapped to an outgoing DSCP value of 0, and an incoming IP-Precedence value of 1 is mapped to a DSCP value of 8, and so forth.

Configure the following to test this default mapping:

- Configure R3 to mark all egress traffic with an IP Precedence value of 5
- Enable "Mls qos" on both switches
- The F0/3 interface of SW2, and the F0/19 interface of SW1 should be configured to trust IP Precedence.

Ping R1 from R3, if the devices are configured correctly, R1 should see the Ping traffic from R3 match to a DSCP value of 40.

On R3:

```
R3(config)#Policy-map TST
R3(config-pmap)#class class-default
R3(config-pmap-c)#Set ip precedence 5
```

```
R3(config)#Int F0/1
R3(config-if)#Service-policy output TST
```

On SW2:

```
SW2(config)#Mls qos
SW2(config)#int f0/3
SW2(config-if)#mls qos trust ip-precedence
```

On SW1:

```
SW1(config)#Mls qos
SW1(config)#int f0/19
SW1(config-if)#mls qos trust ip-precedence
```

To test and verify the configuration:

On R1:

```
R1#Clear counters
Clear "show interface" counters on all interfaces [confirm]
Press Enter to confirm
```

On R3:

```
R3#Ping 10.1.1.1 rep 50
```

Type escape sequence to abort.

Sending 50, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:

!!

Success rate is 100 percent (50/50), round-trip min/avg/max = 1/2/40 ms

On R1:

R1#Show policy-map interface | s d40

```
Class-map: D40 (match-all)
  50 packets, 5700 bytes
  5 minute offered rate 0 bps
  Match: ip dscp cs5 (40)
```

Note: ALL traffic is matched to a DSCP value of 40 on R1.

Let's configure this task:

SW1(config)#Mls qos map ip-prec-dscp 0 8 16 24 32 5 48 56

To verify the configuration:

On SW1:

SW1#Show mls qos map ip-prec-dscp

```
IpPrecedence-dscp map:
  ipprec:   0   1   2   3   4   5   6   7
  -----
  dscp:    0.  8 16 24 32   5 48 56
```

To test the configuration:

On R1:

R1#Clear counters

Clear "show interface" counters on all interfaces [confirm]

Press Enter to confirm

On R3:

R3#Ping 10.1.1.1 rep 5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

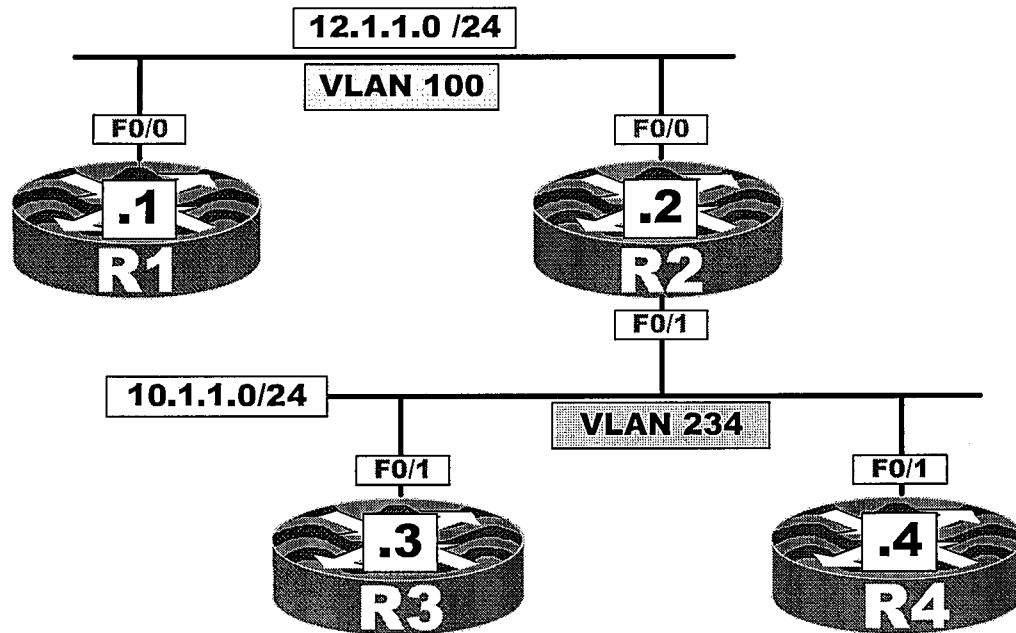
On R1:

```
R1#Sh policy-map interface | sd5  
  
Class-map: D5 (match-all)  
  5 packets, 570 bytes  
  5 minute offered rate 0 bps  
Match: ip dscp 5
```

Task 3

Erase the startup config and reload these devices with a clean configuration before proceeding to the next lab.

Lab 6 – Match Input-Interface & Match NOT



Task 1

Configure the routers based on the above diagram.

On R1:

```
R1(config)#int f0/0
R1(config-if)#ip addr 12.1.1.1 255.255.255.0
R1(config-if)#No shut
```

On R2:

```
R2(config)#int f0/0
R2(config-if)#ip addr 12.1.1.2 255.255.255.0
R2(config-if)#No shut

R2(config)#int f0/1
R2(config-if)#ip addr 10.1.1.2 255.255.255.0
```

R2(config-if)#No shut

On R3:

```
R3(config)#int f0/1
R3(config-if)#ip addr 10.1.1.3 255.255.255.0
R3(config-if)#No shut
```

On R4:

```
R4(config)#int f0/1
R4(config-if)#ip addr 10.1.1.4 255.255.255.0
R4(config-if)#No shut
```

On SW1:

```
SW1(config)#int range f0/1-2
SW1(config-if-range)#swi mode acc
SW1(config-if-range)#swi acc v 100
SW1(config-if-range)#No shu
```

On SW2:

```
SW2(config)#int range f0/2-4
SW2(config-if-range)#swi mode acc
SW2(config-if-range)#swi acc v 200
SW2(config-if-range)#No shut
```

To verify and test the configuration:

On R2:

R2#Ping 12.1.1.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

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 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms
```

```
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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

Task 2

Configure R4 such that any traffic that it generates out of its F0/1 interface is marked with a DSCP value of 40.

On R4:

```
R4(config)#Policy-map tst
R4(config-pmap)#class class-default
R4(config-pmap-c)#Set ip dscp 40

R4(config)#int f0/1
R4(config-if)#service-policy out tst
```

To verify and test the configuration:

On R4:

```
R4#Show policy-map interface
FastEthernet0/1
```

Service-policy output: tst

```
Class-map: class-default (match-any)
  12 packets, 1304 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: any
QoS Set
  dscp cs5
  Packets marked 0
```

To test the configuration, a class-map is configured to match on DSCP value of 40, and a policy-map is configured that references the class-map and the policy-map is applied to the F0/1 interface of R2 inbound.

```

R2(config)#Class-map DSCP40
R2(config-cmap)#match ip dscp 40

R2(config)#policy-map tst
R2(config-pmap)#class DSCP40

R2(config)#int f0/1
R2(config-if)#service-policy in tst

```

To test this configuration, we can use pings that are generated by R4 and verify the DSCP value on R2:

On R2:

```

R2#sh policy-map inter
FastEthernet0/1

Service-policy input: tst

Class-map: DSCP40 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps
  Match: ip dscp cs5 (40)

Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any

```

NOTE: The number of packets is set to zero.

On R4:

```

R4#Ping 10.1.1.2 rep 40

Type escape sequence to abort.
Sending 40, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (40/40), round-trip min/avg/max = 1/3/4 ms

```

On R2:

```

R2#Show policy-map interface
FastEthernet0/1

Service-policy input: tst

```

```
Class-map: DSCP40 (match-all)
```

```
40 packets, 4560 bytes
```

```
5 minute offered rate 0 bps
```

```
Match: ip dscp cs5 (40)
```

```
Class-map: class-default (match-any)
```

```
0 packets, 0 bytes
```

```
5 minute offered rate 0 bps, drop rate 0 bps
```

```
Match: any
```

We can see that 40 packets matched on the class that matches DSCP value of 40. Let's remove the MQC configured on R2 for testing purposes.

On R2:

```
R2(config)#int f0/1
```

```
R2(config-if)#No service-policy in tst
```

```
R2(config)#No policy-map tst
```

```
R2(config)#No class-map DSCP40
```

Task 3

Configure R2 based on the following policy:

- Traffic coming through the F0/0 interface should be classified and marked with a DSCP value of 10.
- Traffic coming through the F0/1 interface should be classified and marked with a DSCP value of 20, this policy should NOT affect traffic that is marked with a DSCP value of 40. DO NOT configure an access-list to accomplish this task.

On R2:

```
R2(config)#Class-map F0/0
```

```
R2(config-cmap)#Match input-interface F0/0
```

```
R2(config)#Class-map F0/1
```

```
R2(config-cmap)#Match NOT dscp 40
```

```
R2(config-cmap)#Match input-interface F0/1
```

```
R2(config)#Policy-map F0/0
```

```

R2(config-pmap)#Class F0/0
R2(config-pmap-c)#set ip dscp 10

R2(config-pmap)#int f0/0
R2(config-if)#Service-policy in F0/0

R2(config)#policy-map F0/1
R2(config-pmap)#class F0/1
R2(config-pmap-c)#Set ip dscp 20

R2(config-pmap-c)#int f0/1
R2(config-if)#Service-policy in F0/1

R2#Show policy-map interface f0/0
FastEthernet0/0

Service-policy input: F0/0

Class-map: F0/0 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: input-interface FastEthernet0/0
QoS Set
  dscp af11
  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

R2#Show policy-map interface f0/1
FastEthernet0/1

Service-policy input: F0/1

Class-map: F0/1 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: not dscp cs5 (40)
Match: input-interface FastEthernet0/1
QoS Set
  dscp af22
  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 verify and test the configuration:

On R1:

R1#**Ping 12.1.1.2 rep 10**

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 12.1.1.2, timeout is 2 seconds:

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

Success rate is 100 percent (10/10), round-trip min/avg/max = 1/2/4 ms

R2#**Show policy-map interface f0/0**

FastEthernet0/0

Service-policy input: F0/0

Class-map: F0/0 (match-all)

10 packets, 1140 bytes

5 minute offered rate 0 bps, drop rate 0 bps

Match: input-interface FastEthernet0/0

QoS Set

dscp af11

Packets marked 10

Class-map: class-default (match-any)

0 packets, 0 bytes

5 minute offered rate 0 bps, drop rate 0 bps

Match: any

R3#**Ping 10.1.1.2 rep 30**

Type escape sequence to abort.

Sending 30, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

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

Success rate is 100 percent (30/30), round-trip min/avg/max = 1/2/4 ms

R2#**Show policy-map interface f0/1**

FastEthernet0/1

Service-policy input: F0/1

Class-map: F0/1 (match-all)

```
30 packets, 3420 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: not dscp cs5 (40)
Match: input-interface FastEthernet0/1
QoS Set
  dscp af22
  Packets marked 30
```

```
Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: any
```

R4#**Ping 10.1.1.2 rep 40**

Type escape sequence to abort.

Sending 40, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:

!!

Success rate is 100 percent (40/40), round-trip min/avg/max = 1/3/4 ms

R2#**Show policy-map interface f0/1**
FastEthernet0/1

Service-policy input: F0/1

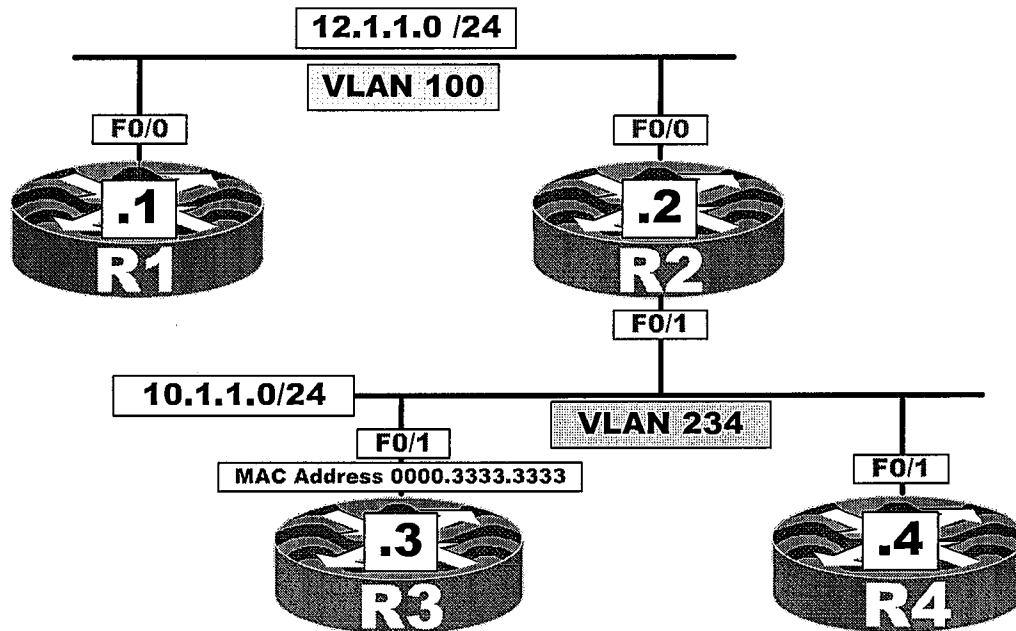
```
Class-map: F0/1 (match-all)
  30 packets, 3420 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: not dscp cs5 (40)
Match: input-interface FastEthernet0/1
QoS Set
  dscp af22
  Packets marked 30

Class-map: class-default (match-any)
  40 packets, 4560 bytes
  5 minute offered rate 2000 bps, drop rate 0 bps
Match: any
```

Task 4

Erase the startup configuration on the routers and reload them before proceeding to the next task.

Lab 7 – Match Destination & Source Address MAC



Task 1

Configure the Routers based on the above diagram.

On SW1:

```
SW1(config)#int range f0/1-2
SW1(config-if-range)#swi mode acc
SW1(config-if-range)#swi acc v 12
SW1(config-if-range)#No shu
```

On SW2:

```
SW2(config)#int range f0/2-4
SW2(config-if-range)#swi mode acc
SW2(config-if-range)#swi acc v 234
SW2(config-if-range)#No shu
```

On R1:

```
R1(config)#int f0/0
R1(config-if)#ip addr 12.1.1.1 255.255.255.0
R1(config-if)#No shut
```

On R2:

```
R2(config)#int f0/0
R2(config-if)#ip addr 12.1.1.2 255.255.255.0
R2(config-if)#No shut
```

```
R2(config)#int f0/1
R2(config-if)#ip addr 10.1.1.2 255.255.255.0
R2(config-if)#No shut
```

On R3:

```
R3(config)#int f0/1
R3(config-if)#mac-address 0000.3333.3333
R3(config-if)#ip addr 10.1.1.3 255.255.255.0
R3(config-if)#No shut
```

On R4:

```
R4(config)#int f0/1
R4(config-if)#ip addr 10.1.1.4 255.255.255.0
R4(config-if)#No shut
```

To verify and test the configuration:

On R1:

```
R1#Ping 12.1.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.2, timeout is 2 seconds:

.!!!!

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

On R2:

```
R2#Ping 12.1.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

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 80 percent (4/5), round-trip min/avg/max = 1/1/4 ms

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

R1#Show arp

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	-	001c.58b9.f779	ARPA	FastEthernet0/1
Internet	10.1.1.3	0	0000.3333.3333	ARPA	FastEthernet0/1
Internet	10.1.1.4	0	0017.59ce.02b9	ARPA	FastEthernet0/1
Internet	12.1.1.1	2	fcfb.fba1.1520	ARPA	FastEthernet0/0
Internet	12.1.1.2	-	001c.58b9.f778	ARPA	FastEthernet0/0

Task 2

Configure RIPv2 on all routers and advertise their directly connected interface/s.

On All Routers:

```
Rx(config)#Router rip
Rx(config-router)#No au
Rx(config-router)#ver 2
Rx(config-router)#Network 0.0.0.0
```

To verify the configuration:

On R4:

```
R4#Show ip rou rip | B Gate
Gateway of last resort is not set

    12.0.0.0/24 is subnetted, 1 subnets
R       12.1.1.0 [120/1] via 10.1.1.2, 00:00:07, FastEthernet0/1
```

Task 3

Configure R2 to classify and mark all IP routed traffic from any source destined to the MAC address of R3's F0/1 interface with an IP precedence level 1.

On R2:

R2 is configured to match any packet destined to the MAC address of 0000.3333.3333, which is R3's configured MAC address:

```
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
```

If the "destination-address Mac" is configured in the class-map, the policy-map can NOT be applied inbound, it must be applied outbound.

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:

The following configures an access-list with the "log" option to match on packets that are marked with an IP precedence level 1:

```
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 10.1.1.3 rep 10
```

Type escape sequence to abort.

Sending 10, 100-byte ICMP Echos to 10.1.1.3, timeout is 2 seconds:
!!!!!!!!!!

Success rate is 100 percent (10/10), round-trip min/avg/max = 1/2/4 ms

On R2:

NOTE: It did not match class-map "QOS" because R4's MAC address is NOT 0000.3333.3333

```
R2#Show policy-map interface
FastEthernet0/1
```

Service-policy output: TST

```
Class-map: QOS (match-all)
  10 packets, 1140 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: destination-address mac 0000.3333.3333
  QoS Set
    precedence 1
```

Packets marked 10

```
Class-map: class-default (match-any)
  35 packets, 3333 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: any
```

On R3:

We can see that 10 packets matched the access-list:

```
R3#Show access-list 100
```

```
Extended IP access list 100
 10 permit ip any any precedence priority log (10 matches)
 20 permit ip any any log (6 matches)
```

Task 4

Configure R2 to classify and mark all IP routed traffic from the MAC address of R3 to any destination with IP precedence level 2.

On R2:

```
R2(config)#Class-map Task4
R2(config-cmap)#match source-address mac 0000.3333.3333
```

```
R2(config)#Policy-map QOS
R2(config-pmap)#class Task4
R2(config-pmap-c)#set ip precedence 2
```

```
R2(config-pmap-c)#int F0/1
R2(config-if)#service-policy in QOS
```

On R2:

```
R2#clear counters
```

On R3:

```
R3#Ping 12.1.1.1 rep 100
```

Type escape sequence to abort.

Sending 100, 100-byte ICMP Echos to 12.1.1.1, timeout is 2 seconds:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Success rate is 100 percent (100/100), round-trip min/avg/max = 1/2/4 ms

On R2:

```
R2#Sh policy-map inter | S Task4
```

```
Service-policy input: QoS
```

```
Class-map: Task4 (match-all)
```

```
  100 packets, 11400 bytes
```

```
    5 minute offered rate 0 bps, drop rate 0 bps
```

```
Match: source-address mac 0000.3333.3333
```

```
QoS Set
```

```
  precedence 2
```

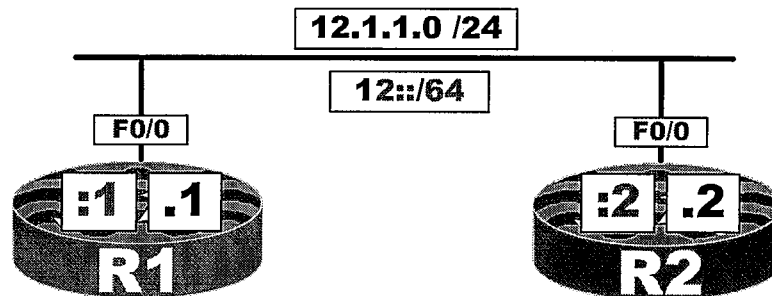
```
    Packets marked 100
```

We can see that 100 packets matched and therefore, they were marked.

Task 5

Erase the startup configuration on the routers and reload them before proceeding to the next task.

Lab 8 – Match IP DSCP/Precedence vs. Match DSCP



Task 1

Configure the Routers based on the above diagram.

On SW1:

```
SW1 (config)#int range f0/1-2
SW1 (config-if-range)#swi mode acc
SW1 (config-if-range)#swi acc v 12
SW1 (config-if-range)#No shu
```

On R1:

```
R1 (config)#int f0/0
R1 (config-if)#ip addr 12.1.1.1 255.255.255.0
R1 (config-if)#ipv6 address 12::1/64
R1 (config-if)#No shut
```

On R2:

```
R2 (config)#int f0/0
R2 (config-if)#ip address 12.1.1.2 255.255.255.0
R2 (config-if)#ipv6 address 12::2/64
R2 (config-if)#No shut
```

To verify the configuration:

On R1:

```
R1#Ping 12.1.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.2, timeout is 2 seconds:

.!!!!

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

```
R1#Ping 12::2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12::2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 0/2/8 ms

Task 2

R2 should be configured to ping R1's F0/0 interface (12.1.1.1) with an IP Precedence value of 1. You should use an MQC to verify the traffic's IP Precedence levels on R1. DO NOT configure an MQC on R2 to accomplish this task.

For verification eight class-maps are configured on R1, each matching different IP Precedence levels; a policy-map called "TST" is configured to reference the eight class-maps, 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 P0
```

```
R1 (config-cmap) #match ip precedence 0
```

```
R1 (config) #class-map P1
```

```
R1 (config-cmap) #match ip precedence 1
```

```
R1 (config) #class-map P2
```

```
R1 (config-cmap) #match ip precedence 2
```

```
R1 (config) #class-map P3
```

```
R1 (config-cmap) #match ip precedence 3
```

```
R1 (config) #class-map P4
```

```

R1(config-cmap)#match ip precedence 4

R1(config)#class-map P5
R1(config-cmap)#match ip precedence 5

R1(config)#class-map P6
R1(config-cmap)#match ip precedence 6

R1(config)#class-map P7
R1(config-cmap)#match ip 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)#int f0/0
R1(config-if)#service-policy input TST

```

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 Byte 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: 12.1.1.1
Repeat count [5]: → Press Enter
Datagram size [100]: → Press Enter
Timeout in seconds [2]: → Press Enter
Extended commands [n]: y
Source address or interface: 12.1.1.2
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 80 percent (4/5), round-trip min/avg/max = 1/2/4 ms

On R1:

R1#Show policy-map interface | s P1

```
Class-map: P1 (match-all)
  4 packets, 456 bytes
  5 minute offered rate 0 bps
  Match: ip precedence 1
```

Note: The output of the above show command reveals that four packets were matched to IP Precedence level 1.

To clear the counter:

R1#Clear counters

Clear "show interface" counters on all interfaces [confirm]
Press Enter to confirm

Let's test 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 inter | S P4

Class-map: P4 (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps
  Match: ip precedence 4
```

Note: IPv6 traffic was NOT matched to "Class P4".

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 P0
R1(config)#No class-map P1
```

```
R1 (config) #No class-map P2
R1 (config) #No class-map P3
R1 (config) #No class-map P4
R1 (config) #No class-map P5
R1 (config) #No class-map P6
R1 (config) #No class-map P7
```

Task 3

R2 should Ping R1's G0/0's IPv6 address (12::1) address; this traffic should be generated with a DSCP value of 20. You should use an MQC to verify the traffic's DSCP value ONLY for IPv6 packets on R1.

Since the DSCP value of IPv6 packets must be matched, the match statement in the class-map includes two statements: match dscp and match protocol ipv6.

On R1:

```
R1 (config) #Class-map D20
R1 (config-cmap) #Match protocol ipv6
R1 (config-cmap) #Match dscp 20

R1 (config) #Policy-map TST
R1 (config-pmap) #class D20

R1 (config-pmap) #Int F0/0
R1 (config-if) #Service-policy in TST
```

To test the configuration:

On R2:

```
R2 (config) #Policy-map tst
R2 (config-pmap) #class class-default
R2 (config-pmap-c) #set dscp 20

R2 (config-pmap-c) #Int F0/0
R2 (config-if) #Service-policy output tst

R2 #Ping ipv6 12::1 repeat 100

Type escape sequence to abort.
```

```
Sending 100, 100-byte ICMP Echos to 12::1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 0/0/0 ms
```

To verify the configuration:

On R1:

```
R1#Sh policy-map inter
FastEthernet0/0
```

```
Service-policy input: TST
```

```
Class-map: D20 (match-all)
  102 packets, 11596 bytes
  5 minute offered rate 2000 bps
  Match: protocol ipv6
  Match: dscp af22 (20)
```

```
Class-map: class-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
```

We can see that the DSCP value was matched in the IPv6 packets. Let's ping R1's IP address and see if it matches the "D20" class-map:

On R1:

```
R1#Clear counters
Clear "show interface" counters on all interfaces [confirm]
Press Enter to confirm
```

On R2:

```
R2#Ping 12.1.1.1 repeat 100
```

```
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 10.1.12.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/1/4 ms
```

To verify the test:

On R1:

```
R1#Show policy-map interface  
FastEthernet0/0
```

```
Service-policy input: TST
```

```
Class-map: D20 (match-all)
```

```
0 packets, 0 bytes
```

```
5 minute offered rate 0 bps
```

```
Match: protocol ipv6
```

```
Match: dscp af22 (20)
```

```
Class-map: class-default (match-any)
```

```
0 packets, 0 bytes
```

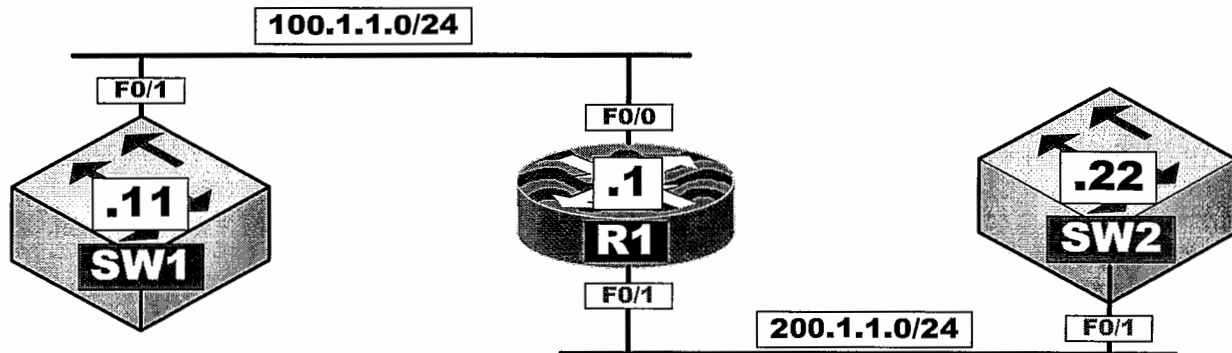
```
5 minute offered rate 0000 bps, drop rate 0000 bps
```

```
Match: any
```

Task 4

Erase the startup configuration on the routers and reload them before proceeding to the next task.

Lab 9 – Match Protocol HTTP URL, MIME & Host



Task 1

Configure the Router and the Switches based on the above diagram.

On SW1:

```
SW1(config)#Int f0/1
SW1(config-if)#No swi
SW1(config-if)#ip addr 100.1.1.11 255.255.255.0
SW1(config-if)#No shut
```

On R1:

```
R1(config)#Int f0/0
R1(config-if)#ip addr 100.1.1.1 255.255.255.0
R1(config-if)#No shut

R1(config)#int f0/1
R1(config-if)#ip addr 200.1.1.1 255.255.255.0
R1(config-if)#No shut
```

On SW2:


```
SW2(config)#Int f0/1
SW2(config-if)#No swi
SW2(config-if)#ip addr 200.1.1.22 255.255.255.0
SW2(config-if)#No shut
```

To verify the configuration:

On R1:

```
R1#Ping 100.1.1.11
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 100.1.1.11, timeout is 2 seconds:
..!!!

Success rate is 60 percent (3/5), round-trip min/avg/max = 1/2/4 ms

```
R1#Ping 200.1.1.22
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 200.1.1.22, timeout is 2 seconds:
..!!!

Success rate is 60 percent (3/5), round-trip min/avg/max = 1/2/4 ms

Task 2

Save the configuration of SW1 and copy the startup configuration of this switch (**config.text**) to flash as "config.mpeg", "config.jpeg", and "config.txt".

On SW1:

```
SW1#wr
```

```
SW1#Dir
```

Directory of flash:/

```
 2  -rw-   7899301   Mar 1 1993 00:19:35 +00:00  c3560-advipservicesk9-mz.122-25.SEE.bin
 3  -rw-      1320   Mar 1 1993 00:11:17 +00:00  config.text
 4  -rw-       24   Mar 1 1993 00:11:17 +00:00  private-config.text
```

```
SW1#Copy config.text config.mpeg
```

Destination filename [config.mpeg]? → Press Enter

```
SW1#Copy config.text config.txt
Destination filename [config.txt]? → Press Enter
```

```
SW1#Copy config.text config.jpeg
Destination filename [config.jpeg]? → Press Enter
```

To verify the configuration:

On SW1:

```
Switch#dir
```

```
Directory of flash:/
```

2	-rwx	7899301	Mar 1 1993 00:19:35 +00:00	c3560-advipservicesk9-mz.122-25.SEE.bin
3	-rwx	1320	Mar 1 1993 00:11:17 +00:00	config.text
4	-rwx	24	Mar 1 1993 00:11:17 +00:00	private-config.text
5	-rwx	1320	Mar 1 1993 00:13:35 +00:00	config.mpeg
6	-rwx	1320	Mar 1 1993 00:13:44 +00:00	config.txt
7	-rwx	1320	Mar 1 1993 00:13:55 +00:00	config.jpeg

```
15998976 bytes total (8091648 bytes free)
```

Task 3

Enable the Cisco web browser interface on SW1 and specify the HTTP path used to locate the files created in the previous step.

On SW1:

```
SW1(config)#ip http server
SW1(config)#ip http path flash:
```

To verify the configuration:

On SW1:

```
SW1#Sh ip http server status
```

```
HTTP server status: Enabled
```

```
HTTP server port: 80
HTTP server authentication method: enable
HTTP server access class: 0
HTTP server base path: flash:
Maximum number of concurrent server connections allowed: 16
```

(The rest of the output is omitted)

Task 4

Configure R1's F0/1 interface such that it blocks any URL that contains any file type with an "mpeg" extension.

On R1:

```
R1(config)#Class-map QOS
R1(config-cmap)#match protocol http url *.mpeg

R1(config)#Policy-map TST
R1(config-pmap)#class QOS
R1(config-pmap-c)#drop

R1(config-pmap-c)#int f0/1
R1(config-if)#Service-policy input TST
```

To verify the configuration:

On R1:

```
R1#Show class-map QOS

Class Map match-all QOS (id 1)
  Match protocol http url "*.mpeg"

R1#Show policy-map TST

Policy Map TST
  Class QOS
    drop

R1#Show policy-map interface f0/1
```

FastEthernet0/1

Service-policy input: TST

Class-map: QOS (match-all)
0 packets, 0 bytes

5 minute offered rate 0 bps, drop rate 0 bps
Match: protocol http url "*.mpeg"
drop

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 SW2:

For testing purposes, the "config.txt" is copied first, you should be successful.

NOTE: Your file size may be different.

```
SW2#Copy http://100.1.1.11/config.txt Null:
Loading http://100.1.1.11/config.txt !
1711 bytes copied in 0.026 secs (65808 bytes/sec)
```

The following operation fails because of the "mpeg" extension, these extensions are blocked on R2:

```
SW2#Copy http://100.1.1.11/config.mpeg NULL:
%Error opening http://100.1.1.11/config.mpeg (I/O error)
```

Task 5

Reconfigure R2 such that it blocks any URL that contains any file type with an "mpeg" or "jpeg" extension.

On R1:

```
R1(config)#Class-map QOS
R1(config-cmap)#No match protocol http url *.mpeg
```

```
R1(config-cmap)#match protocol http url *.mpeg|*.jpeg
```

To verify the configuration:

On R1:

```
R1#Show class-map QOS
```

```
Class Map match-all QOS (id 1)
  Match protocol http url "*.mpeg|*.jpeg"
```

```
R1#Show policy-map interface f0/1
FastEthernet0/1
```

```
Service-policy input: TST
```

```
Class-map: QOS (match-all)
  8 packets, 1320 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: protocol http url "*.mpeg|*.jpeg"
  drop
```

```
Class-map: class-default (match-any)
  18 packets, 1481 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
```

To test the configuration:

On SW2:

Note: The following two copy operations fail because the file extensions are blocked
On R2::

```
SW2#Copy http://100.1.1.11/config.mpeg NULL:
```

You should receive the following console message:

```
%Error opening http://100.1.1.11/config.mpeg (I/O error)
```

```
SW2#Copy http://100.1.1.11/config.jpeg NULL:
```

You should receive the following console message:

```
%Error opening http://100.1.1.11/config.jpeg (I/O error)
```

Note: The following copy operation is successful, because the "TXT" extension types are NOT blocked by R2:

```
SW2#Copy http://100.1.1.11/config.txt NULL:
Loading http://100.1.1.11/config.txt !
1711 bytes copied in 0.017 secs (100647 bytes/sec)
```

Task 6

Reconfigure R2 such that any HTTP call to the server located at "100.1.1.11" is dropped.

On R2:

```
R1(config-if) #Class-map QOS
R1(config-cmap) #NO match protocol http url *.mpeg|*.jpeg
R1(config-cmap) #match protocol http host 100.1.1.11
```

To test the configuration:

On SW2:

```
SW2#Copy http://100.1.1.11/config.txt NULL:
```

You should see the following console message:

```
%Error opening http://12.1.1.10/config.txt (I/O error)
```

```
SW2#Copy http://12.1.1.10/config.mpeg NULL:
```

You should see the following console message:

```
%Error opening http://12.1.1.10/config.mpeg (I/O error)
```

```
SW2#Copy http://12.1.1.10/config.jpeg NULL:
```

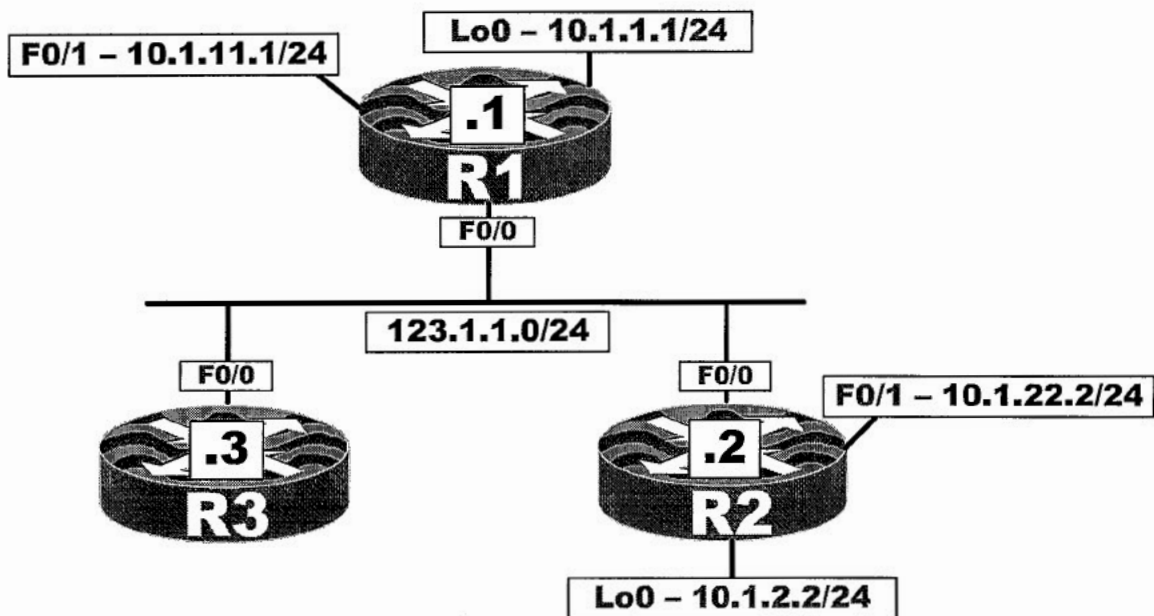
You should see the following console message:

```
%Error opening http://12.1.1.10/config.jpeg (I/O error)
```

Task 7

Erase the startup configuration on the routers and reload them before proceeding to the next task.

Lab 10 – Class-Based Policing



Task 1

Configure the Routers and the Switch based on the above diagram.

On SW1:

```
SW1(config)#Int range f0/1-3
SW1(config-if-range)#Swt mode acc
SW1(config-if-range)#Swt acc vlan 100
SW1(config-if-range)#No shut
```

On SW2:

```
SW2(config)#int f0/1
SW2(config-if)#swt mode acc
SW2(config-if)#swt acc v 11
SW2(config-if)#No shu
```



```
SW2(config-if)#int f0/2
SW2(config-if)#swi mdoe acc
SW2(config-if)#swi acc v 22
SW2(config-if)#No shu
```

On R1:

```
R1(config)#Int F0/0
R1(config-if)#IP addr 123.1.1.1 255.255.255.0
R1(config-if)#No shut
```

```
R1(config)#Int Lo0
R1(config-if)#IP addr 10.1.1.1 255.255.255.0
```

```
R1(config)#int F0/1
R1(config-if)#IP addr 10.1.11.1 255.255.255.0
R1(config-if)#No shut
```

On R2:

```
R2(config)#Int F0/0
R2(config-if)#IP addr 123.1.1.2 255.255.255.0
R2(config-if)#No shut
```

```
R2(config)#Int F0/1
R2(config-if)#IP addr 10.1.22.2 255.255.255.0
R2(config-if)#No shut
```

```
R2(config)#Int Lo0
R2(config-if)#IP addr 10.1.2.2 255.255.255.0
```

On R3:

```
R3(config)#Int F0/0
R3(config-if)#IP addr 123.1.1.3 255.255.255.0
R3(config-if)#No shut
```

To verify the configuration:

On R1:

```
R1#Ping 123.1.1.2
```

Type escape sequence to abort.

```
Sending 5, 100-byte ICMP Echos to 123.1.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

R1#Ping 123.1.1.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 123.1.1.3, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

Task 2

Configure R1's F0/0 interface using the following policy:

- HTTP, FTP and ICMP traffic should be rate limited to 10 Mbps outbound.
- 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 CEF 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. Remember that in a single rate single bucket, the "Be" is disabled, and 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 protocol" 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 configured, a "match-all" option will be the default.

```
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 1000000 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
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: protocol ftp
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: protocol icmp
    0 packets, 0 bytes
```

```

5 minute rate 0 bps

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: qos-2 (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: protocol telnet
    0 packets, 0 bytes
    5 minute rate 0 bps
  Match: protocol smtp
    0 packets, 0 bytes
    5 minute rate 0 bps
  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)
  2 packets, 120 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:

- The outgoing Telnet traffic should be limited to 10 Mbps. This traffic should be configured with minimum amount of normal burst.
- The outgoing 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-action transmit exceed-
action 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-action transmit exceed-
action 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 4

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 on 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)#access-list 700 permit 0000.1111.1111
```

```
R1(config)#class-map server1
```

```
R1(config-cmap)#match access-group 700
```

```
R1(config)#access-list 701 permit 0000.2222.2222
```

```
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 2000000 conform-action transmit exceed-action drop
```

```
R1(config)#int f0/1
```

```
R1(config-if)#service-policy Output 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)
  0 packets, 0 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 5

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

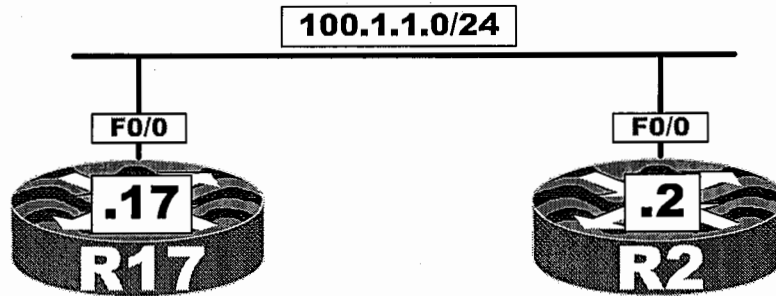
Service-policy output: TEST
Class-map: QOS (match-all)
  0 packets, 0 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: access-group 100
```

```
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 6

Erase the startup configuration and reload the routers before proceeding to the next lab.

Lab 11 – Class-Based Shaping



Task 1

Configure the routers based on the above diagram.

On SW1:

```
SW1(config)#int range f0/17,f0/2
SW1(config-if-range)#swi mode acc
SW1(config-if-range)#swi acc v 100
SW1(config-if-range)#No shu
```

On R17:

```
R1(config)#int f0/0
R1(config-if)#ip addr 100.1.1.17 255.255.255.0
R1(config-if)#No shut
```

On R2:

```
R2(config)#int f0/0
R2(config-if)#ip addr 100.1.1.2 255.255.255.0
R2(config-if)#No shut
```

To verify the configuration:

On R17:

```
R1#Ping 100.1.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 100.1.1.2, timeout is 2 seconds:

.!!!!

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

Task 2

Traffic from R17 to R2 should be shaped to 16000 bps, configure the Tc to be 4 seconds, R1 should send 8000 Bytes (Bc) per Tc.

When we send Bc's worth of data we are performing shape average, but if we send Bc + Be every Tc, we are performing shape peak. The task states that the Tc is 4 seconds, since the CIR is 16000, and the "Tc=Bc/CIR" if the formula is re-arranged, all we need to do is multiply the CIR by four, and that should give us the Bc value. Since the task did not specify the Be value, Be MUST be configured as zero, if the Be value is NOT configured at all (Not even a zero), Be will equal to Bc.

On R17:

```
R17(config)#policy-map tst
R17(config-pmap)#class class-default
R17(config-pmap-c)#shape average 16000 64000 0

R17(config)#int f0/0
R17(config-if)#service-policy out tst
```

To verify the configuration:

On R17:

```
R17#Show policy-map interface
FastEthernet0/0

Service-policy output: tst

Class-map: class-default (match-any)
  103 packets, 69005 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
Match: any
Queueing
```

```
queue limit 64 packets
(queue depth/total drops/no-buffer drops) 0/0/0
(pkts output/bytes output) 103/69005
shape (average) cir 16000, bc 64000, be 0
target shape rate 16000
```

Based on the shaping parameters configured, if we divide Bc/CIR, it should give us the Tc value, in this case:

$64000/16000 = 4$ seconds.

Therefore, every 4 seconds R17 will be sending Bc's worth of data, because "Shape average" is configured. So, every 4 seconds R17 will send 64000 bps. Since we are going to test this using a ping command, and the size of the packets in ping is defined in Bytes, let's divide 64000 by 8, this will give us 8000 Bytes, the amount of data that R17 will send every 4 seconds or Tc. Let's configure and test:

On R17:

```
R17#Debug ip icmp
ICMP packet debugging is on
```

```
R17#Ping 100.1.1.2 size 1000 timeout 0 repeat 32
```

Type escape sequence to abort.

Sending 32, 1000-byte ICMP Echos to 100.1.1.2, timeout is 0 seconds:

.....
Success rate is 0 percent (0/32)

```
*Sep 27 21:29:34.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:34.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:34.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:34.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:34.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:34.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:34.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:34.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17

*Sep 27 21:29:38.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:38.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:38.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:38.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:38.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:38.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:38.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:38.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17

*Sep 27 21:29:42.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:42.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:42.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
```

```
*Sep 27 21:29:42.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:42.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:42.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:42.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:42.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17

*Sep 27 21:29:46.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:46.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:46.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:46.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:46.739: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:46.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:46.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
*Sep 27 21:29:46.743: ICMP: echo reply rcvd, src 100.1.1.2, dst 100.1.1.17
```

The Ping command is sending 1000 Byte packets and since the shape average is configured to send 8000 Bytes every Tc which is 4 seconds, we can see that every 4 seconds the local router is sending eight 1000 Byte packets.

Task 3

Erase the startup configuration of the Routers and reload them before proceeding to the next lab.

CCIE Foundation

5.0

www.MicronicsTraining.com

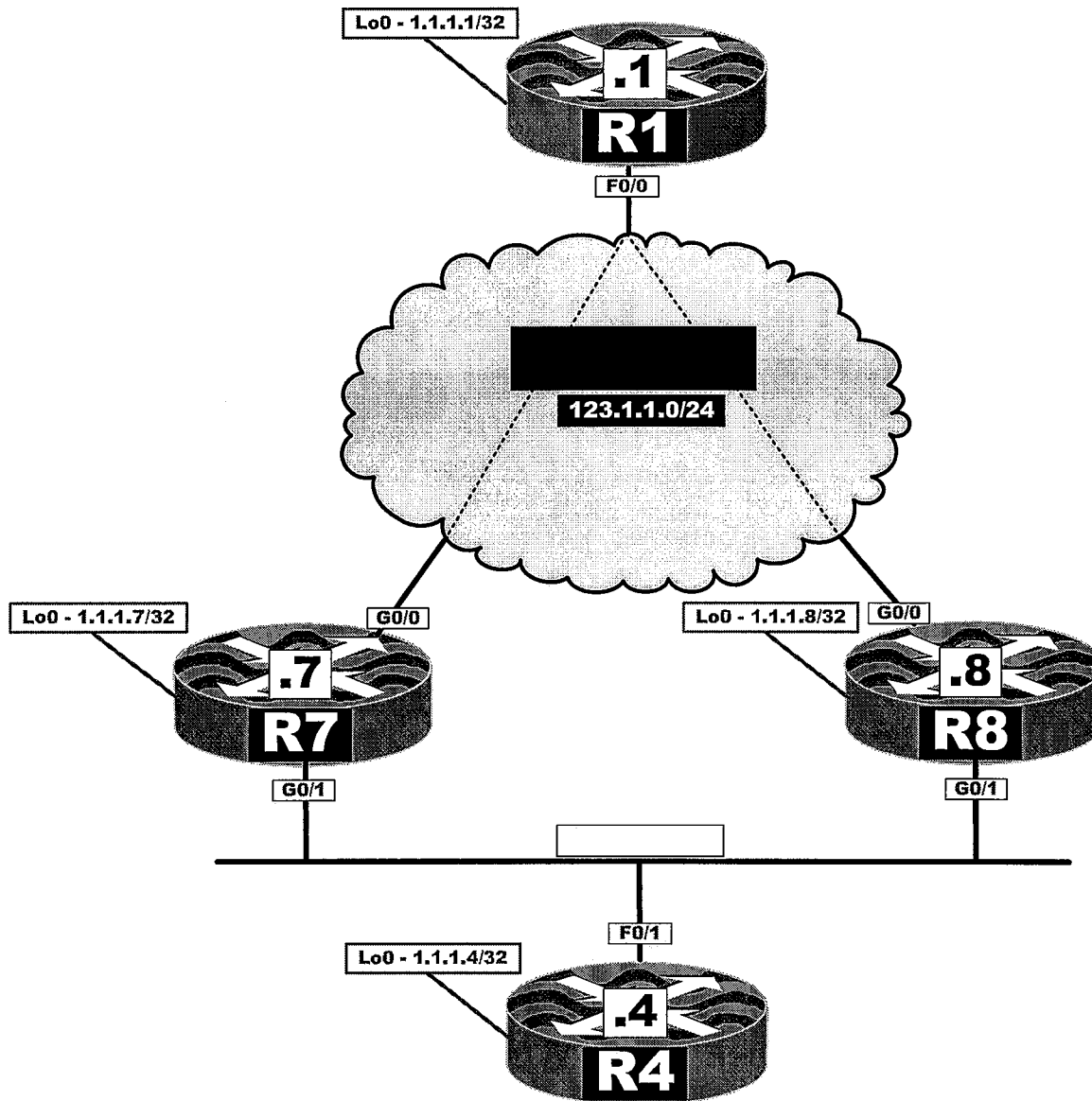
**Narbik Kocharians
CCIE #12410
R&S, Security, SP**

IP Services, Network

Optimization and Advanced

Services

Lab 1 – HSRP Configuration



Task 1

Configure the routers based on the following policy:

- Configure DMVPN Phase 1 such that R1 is configured as the hub, and routers R2 and R3 are configured as the spokes.
- In the above diagram, SW1 should be configured to represent the Internet and provide routing to R1's F0/0, R7 and R8's G0/0 interfaces for the DMVPN Network.
- Configure 192.1.1.10/24, 192.1.7.10/24, and 192.1.8.10 on SW1 for ports F0/1, F0/7 and F0/8 respectively.
- R1, R7 and R8 should be configured with static routes to reach the IP addresses of the routers connected to this DMVPN network; these routers should use the IP address of the Switchport to which they are connected to as their gateway. The static routes should be configured for each IP address, DO NOT configure a static default route to accomplish this task.
- R1 should NOT be configured with static mapping for the DMVPN endpoints, provide Multicast capability.
- Configure R7 and R8's G0/1 interfaces, and R4's F0/1 interface in VLAN 234.

On SW2:

```
SW2(config)#Int F0/23
SW2(config-if)#Switchport trunk encap dot1q
SW2(config-if)#Switchport mode trunk
SW2(config-if)#No shut

SW2(config)#Int range F0/4, F0/7
SW2(config-if-range)#Switchport mode access
SW2(config-if-range)#Switchport access vlan 478
SW2(config-if)#No shut
```

On SW3:

```
SW3(config)#int f0/23
SW3(config-if)#Swi trunk enc dot
SW3(config-if)#Swi mode trunk
SW3(config-if)#No shut
```

```
SW3(config)#int f0/8
SW3(config-if)#Swi mode access
SW3(config-if)#Swi access vlan 478
SW3(config-if)#No shut
```

On R7:

```
R7(config)#int G0/1
R7(config-if)#ip addr 10.1.1.7 255.255.255.0
R7(config-if)#No shut

R7(config)#Int lo0
R7(config-if)#ip addr 1.1.1.7 255.255.255.255
```

On R8:

```
R8(config)#int G0/1
R8(config-if)#ip addr 10.1.1.8 255.255.255.0
R8(config-if)#No shut

R8(config-if)#Int lo0
R8(config-if)#Ip addr 1.1.1.8 255.255.255.255
```

On R4:

```
R4(config)#int f0/1
R4(config-if)#ip addr 10.1.1.4 255.255.255.0
R4(config-if)#No shut

R4(config)#Int lo0
R4(config-if)#ip addr 1.1.1.4 255.255.255.255
```

To verify and test the configuration:

On R4:

```
R4#Ping 10.1.1.7
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.7, timeout is 2 seconds:

.....

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

```
R4#Ping 10.1.1.8
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.8, timeout is 2 seconds:

.!!!!

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

Let's configure the DMVPN phase 1; before the tunnel is configured, the switch is configured to represent the Internet:

On SW1:

```
SW1(config)#ip routing
```

```
SW1(config)#int f0/1
```

```
SW1(config-if)#No swi
```

```
SW1(config-if)#ip addr 192.1.1.10 255.255.255.0
```

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

```
SW1(config)#int f0/7
```

```
SW1(config-if)#No swi
```

```
SW1(config-if)#ip addr 192.1.7.10 255.255.255.0
```

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

```
SW1(config)#int f0/8
```

```
SW1(config-if)#No swi
```

```
SW1(config-if)#ip addr 192.1.8.10 255.255.255.0
```

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

On R1:

```
R1(config-if)#Int Lo0
```

```
R1(config-if)#Ip addr 1.1.1.1 255.255.255.255
```

```
R1(config)#Int F0/0
```

```
R1(config-if)#Ip addr 192.1.1.1 255.255.255.0
```

```
R1(config-if)#No shut
```

```
R1(config)#ip route 192.1.7.7 255.255.255.255 192.1.1.10
```

```
R1(config)#ip route 192.1.8.8 255.255.255.255 192.1.1.10
```

On R7:

```
R7(config)#Int G0/0
```

```
R7(config-if)#Ip addr 192.1.7.7 255.255.255.0
```

```
R7(config-if)#No shut
```

```
R7(config)#ip route 192.1.1.1 255.255.255.255 192.1.7.10
R7(config)#ip route 192.1.8.8 255.255.255.255 192.1.7.10
```

On R8:

```
R8(config)#Int G0/0
R8(config-if)#Ip addr 192.1.8.8 255.255.255.0
R8(config-if)#No shut

R8(config)#ip route 192.1.1.1 255.255.255.255 192.1.8.10
R8(config)#ip route 192.1.7.7 255.255.255.255 192.1.8.10
```

To verify the configuration:

On R1:

```
R1#Ping 192.1.7.7
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.1.7.7, timeout is 2 seconds:
..!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 1/2/4 ms
```

```
R1#Ping 192.1.8.8
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.1.8.8, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

Let's configure the tunnel interfaces:

On R1:

```
R1(config)#Int tunnel 178
R1(config-if)#IP addr 178.1.1.1 255.255.255.0
R1(config-if)#Tunnel source F0/0
R1(config-if)#Tunnel mode gre multipoint
R1(config-if)#IP nhrp network-id 111
R1(config-if)#IP nhrp map multicast dynamic
```

On R7:

```
R7(config)#Int tunnel 178
```

```

R7(config-if)#IP addr 178.1.1.7 255.255.255.0
R7(config-if)#Tunnel source G0/0
R7(config-if)#Tunnel destination 192.1.1.1
R7(config-if)#IP nhrp network-id 777
R7(config-if)#IP nhrp map 178.1.1.1 192.1.1.1
R7(config-if)#IP nhrp nhs 178.1.1.1

```

On R8:

```

R8(config)#int tunnel 178
R8(config-if)#ip addr 178.1.1.8 255.255.255.0
R8(config-if)#Tunnel source G0/0
R8(config-if)#Tunnel destination 192.1.1.1
R8(config-if)#IP nhrp network-id 888
R8(config-if)#IP nhrp map 178.1.1.1 192.1.1.1
R8(config-if)#IP nhrp nhs 178.1.1.1

```

To verify and test the configuration:

On R1:

R1#Show dmvpn

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
 N - NATed, L - Local, X - No Socket
 # Ent --> Number of NHRP entries with same NBMA peer
 NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting
 UpDn Time --> Up or Down Time for a Tunnel

Interface: Tunnel178, IPv4 NHRP Details

Type:Hub, NHRP Peers:2,

# Ent	Peer NBMA Addr	Peer Tunnel Add	State	UpDn Tm	Attrb
1	192.1.7.7	178.1.1.7	UP	00:01:42	D
1	192.1.8.8	178.1.1.8	UP	00:00:18	D

R1#Show ip nhrp

```

178.1.1.7/32 via 178.1.1.7
  Tunnel178 created 00:03:08, expire 01:56:51
  Type: dynamic, Flags: unique registered
  NBMA address: 192.1.7.7
178.1.1.8/32 via 178.1.1.8
  Tunnel178 created 00:01:43, expire 01:58:16
  Type: dynamic, Flags: unique registered

```

NBMA address: 192.1.1.8

R1#**Ping 178.1.1.7**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 178.1.1.7, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R1#**Ping 178.1.1.8**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 178.1.1.8, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

On R7:

R7#**Show dmvpn | B Interface**

Interface: Tunnel178, IPv4 NHRP Details

Type:Spoke, NHRP Peers:1,

#	Ent	Peer NBMA Addr	Peer Tunnel Add	State	UpDn	Tm	Attrb
1	192.1.1.1		178.1.1.1	UP	00:04:31		S

R7#**Show ip nhrp**

178.1.1.1/32 via 178.1.1.1

Tunnel178 created 00:05:07, never expire

Type: static, Flags:

NBMA address: 192.1.1.1

R7#**Ping 178.1.1.1**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 123.1.1.1, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R7#**Ping 178.1.1.8**

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 178.1.1.8, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms

R7#Traceroute 178.1.1.8 Numeric

Type escape sequence to abort.

Tracing the route to 178.1.1.8

VRF info: (vrf in name/id, vrf out name/id)

1 178.1.1.1 4 msec 0 msec 4 msec

2 178.1.1.8 0 msec * 0 msec

On R8:

R8#Show dmvpn | B Interface

Interface: Tunnel178, IPv4 NHRP Details

Type:Spoke, NHRP Peers:1,

#	Ent	Peer NBMA Addr	Peer Tunnel Add	State	UpDn Tm	Attrb
1		192.1.1.1	178.1.1.1	UP	00:05:35	S

R8#Show ip nhrp

178.1.1.1/32 via 178.1.1.1

Tunnel178 created 00:06:26, never expire

Type: static, Flags:

NBMA address: 192.1.1.1

R8#Ping 178.1.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 178.1.1.1, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms

R8#Ping 178.1.1.7

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 178.1.1.7, timeout is 2 seconds:

!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms

R8#Traceroute 178.1.1.7 Numeric

Type escape sequence to abort.

Tracing the route to 178.1.1.7

VRF info: (vrf in name/id, vrf out name/id)

```
1 178.1.1.1 0 msec 4 msec 0 msec
2 178.1.1.7 0 msec * 0 msec
```

Task 2

Configure RIPv2 on R1, R7 and R8. Advertise their Loopback 0 and their tunnel interfaces in this routing domain. Disable the auto-summarization.

R7 and R8 should be configured to redistribute network 10.1.1.0 /24 into the RIP routing domain.

On R1:

```
R1(config)#Router rip
R1(config-router)#No au
R1(config-router)#ver 2
R1(config-router)#Netw 1.0.0.0
R1(config-router)#Netw 178.1.0.0

R1(config)#int tunnel 178
R1(config-if)#No ip split-horizon
```

On R7:

```
R7(config)#Router rip
R7(config-router)#No au
R7(config-router)#ver 2
R7(config-router)#Netw 1.0.0.0
R7(config-router)#Netw 178.1.0.0
```

On R8:

```
R8(config)#Router rip
R8(config-router)#No au
R8(config-router)#ver 2
R8(config-router)#Netw 1.0.0.0
R8(config-router)#Netw 178.1.0.0
```

To verify the configuration:

On R7:

R7#**Show ip route rip | B Gate**
Gateway of last resort is not set

```
1.0.0.0/32 is subnetted, 3 subnets
R      1.1.1.1 [120/1] via 178.1.1.1, 00:00:23, Tunnel178
R      1.1.1.8 [120/2] via 178.1.1.8, 00:00:01, Tunnel178
```

On R8:

R8#**Show ip route rip | B Gate**
Gateway of last resort is not set

```
1.0.0.0/32 is subnetted, 3 subnets
R      1.1.1.1 [120/1] via 178.1.1.1, 00:00:03, Tunnel178
R      1.1.1.7 [120/2] via 178.1.1.7, 00:00:03, Tunnel178
```

R8#**Traceroute 1.1.1.7 Numeric**

Type escape sequence to abort.
Tracing the route to 1.1.1.7
VRF info: (vrf in name/id, vrf out name/id)
 1 178.1.1.1 4 msec 0 msec 4 msec
 2 178.1.1.7 0 msec * 0 msec

R8#**Ping 1.1.1.7**

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.7, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

On R7 and R8:

The following redistributes the G0/1 interfaces of R7 and R8 into the RIP routing domain. The purpose of this configuration is for the return traffic back to R4. In the later tasks R4 will Ping 1.1.1.1/32 prefix, and if R1 does NOT have a return path back to that 10.1.1.0/24 segment, the test will fail.

```
Rx(config)#Route-map tst permit 10
Rx(config-route-map)#match inter G0/1

Rx(config)#Router rip
Rx(config-router)#redistribute connected route-map tst
```

To verify the configuration:

On R1:

```
R1#Show ip route rip | B Gate
```

```
Gateway of last resort is not set
```

```
1.0.0.0/32 is subnetted, 3 subnets
```

```
R      1.1.1.7 [120/1] via 178.1.1.7, 00:00:13, Tunnel178
```

```
R      1.1.1.8 [120/1] via 178.1.1.8, 00:00:25, Tunnel178
```

```
10.0.0.0/24 is subnetted, 1 subnets
```

```
R      10.1.1.0 [120/1] via 178.1.1.8, 00:00:25, Tunnel178  
[120/1] via 178.1.1.7, 00:00:13, Tunnel178
```

Task 3

Configure VLAN 478 such that the host/s (In this case R4) in this VLAN use R8 as the primary and R7 as the backup default gateway. Use the following policy to accomplish this task:

- You must use 10.1.1.78 IP address as the gateway.
- ONLY R4 should be configured with a single static route, the other routers should NOT be configured with any static route/s.
- DO NOT change the IP address of any router.
- Use HSRP.

The Hot Standby Router Protocol or HSRP is designed to allow for transparent failover of the First-Hop IP router (The default gateways).

With HSRP, HSRP's virtual IP address (VIP) is configured as the default gateway on the hosts, and the primary router is the responsible router for this VIP.

Once HSRP is configured on a network segment, it provides a VIP and a Virtual MAC Address (VMAC) that is shared among a group of routers running HSRP. Only one of the routers within the group is chosen as the primary or the active router. The active/Primary router receives and routes packets destined for the MAC address of the group, this is the VMAC.

HSRP detects when the active/Primary router fails and selects another router from the HSRP Group as the active/primary router.

The active and Standby (The Backup) election is based on the configured priority. By default, all routers within the group have a priority of 100, and the router with a higher priority value will be elected as the active/primary.

The routers running HSRP use a hello mechanism to detect router failure; this mechanism uses UDP-based Multicast. When the active/primary router fails to send a hello message within a configured period of time, the standby router with the highest priority or next highest priority will transition into active/primary router. This process is completely transparent to the hosts on that given segment.

Let's configure HSRP on R7 and R8:

On R7 and R8:

```
Rx(config)#Int G0/1
Rx(config-if)#Standby 1 ip 10.1.1.78
```

You should see the following console message:

```
%HSRP-5-STATECHANGE: GigabitEthernet0/1 Grp 1 state Speak -> Standby
```

Let's verify HSRP On R7:

On R7:

```
R7#Show standby
```

```
GigabitEthernet0/1 - Group 1
  State is Standby
    1 state change, last state change 00:00:15
  Virtual IP address is 10.1.1.78
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.256 secs
  Preemption disabled
  Active router is 10.1.1.8, priority 100 (expires in 9.344 sec)
  Standby router is local
  Priority 100 (default 100)
  Group name is "hsrp-Gi0/1-1" (default)
```

The output of the above show command reveals the following:

- HSRP group is 1
- The state of the local router is Standby
- VIP address is 10.1.1.78
- VMAC address id 0000.0c07.ac01
- Hellos are set to 3 seconds, and the Hold time is set to three times as much as the hello interval
- Preemption is disable

- The IP address of the active/primary router is 10.1.1.8
- The local router is in standby mode
- The name of the HSRP group is "hsrp-Gi0/1-1". The (default) at the end of this line means that the default name is being used, meaning that we have not configured the name for this group.

In HSRP we can have multiple groups; this will be configured and discussed in later tasks.

The local router is in Standby mode. In this case, the priorities of these two routers were not configured, why did R3 become the active router? If the priority of the routers within a group is identical, the router with the highest IP address will transition into Active/Primary state.

When HSRP is configured, the group number is specified, the VMAC address is derived from the configured group number. The VMAC address is "0000.0c07.acxx", where xx is the group number. If a group number is not specified in the configuration, the IOS will assign "group 0" automatically, in which case the VMAC address will be 0000.0c07.ac00".

By default, the hello intervals are set to 3 seconds and the Hold time is set to 10 seconds.

The preemption is disabled by default; this feature will be discussed and configured in later tasks.

The IP address of the active/Primary.

The Group-name, the group name of the configured HSRP

On R7:

R2#**Show standby brief**

P indicates configured to preempt.

|

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	100		Standby	10.1.1.8	local	10.1.1.78

The above show command reveals the important aspects of the configured HSRP.

To complete the configuration, R4's default gateway MUST be configured to point to the VIP address, in this case 10.1.1.23.

On R4:

R4(config)#**Ip route 0.0.0.0 0.0.0.0 10.1.1.78**

To test the configuration:

On R4:

To clear the ARP table:

```
R4(config)#Int F0/1
R4(config-if)#Shut
```

Wait for the interface to go down, then:

```
R4(config-if)#No shut
```

```
R4#Traceroute 1.1.1.1 Numeric
```

```
Type escape sequence to abort.
Tracing the route to 1.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
 1 10.1.1.8 0 msec 4 msec 0 msec
 2 178.1.1.1 4 msec * 0 msec
```

The output of the above traceroute shows that R8 is the next-hop to reach the 1.1.1.1/32 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:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
```

```
R4#Show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.4	-	0017.59ce.02b9	ARPA	FastEthernet0/1
Internet	10.1.1.78	0	0000.0c07.ac01	ARPA	FastEthernet0/1

You can see that the VMAC (The Virtual MAC) address is used instead of the real MAC address of R8.

```
R4#Ping 10.1.1.78
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.23, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

Let's test by shutting down the G0/1 interface of the active router (R8), and check if the failover occurs:

On R8:

```
R8(config)#Int G0/1
R8(config-if)#Shut
```

You should see the following console message:

```
%HSRP-5-STATECHANGE: FastEthernet0/1 Grp 1 state Active -> Init
```

```
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to  
administratively down
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed  
state to down
```

Let's Traceroute to 1.1.1.1 IP address on R4 again and verify the output:

On R4:

```
R4#Traceroute 1.1.1.1 Numeric
```

Type escape sequence to abort.

Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

```
 1 10.1.1.7 4 msec 0 msec 0 msec  
 2 178.1.1.1 4 msec * 0 msec
```

```
R4#Show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.4	-	0017.59ce.02b9	ARPA	FastEthernet0/1
Internet	10.1.1.78	0	0000.0c07.ac01	ARPA	FastEthernet0/1

We can see that the failover worked, but the default gateway of R4 was not changed and the ARP table verifies that R4 is still using 10.1.1.78 with a MAC address of 0000.0c07.ac01.

Let's "No shut" the G0/1 interface of R8:

On R8:

```
R8(config)#Int G0/1
```

```
R8(config-if)#No shut
```

You should see the following console message:

```
%HSRP-5-STATECHANGE: FastEthernet0/1 Grp 1 state Standby -> Active
```

To verify the configuration:

On R8:

You should see the following console message:

```
%HSRP-5-STATECHANGE: GigabitEthernet0/1 Grp 1 state Speak -> Standby
```

```
R8#Show standby brief
```

```
                P indicates configured to preempt.
                |
Interface      Grp  Pri P State   Active      Standby      Virtual IP
Gi0/1          1    100 Standby 10.1.1.7    local        10.1.1.78
```

Why didn't R8 become the active router?

Because preemption is NOT enabled by default, let's verify this information:

```
R8#Show standby | Inc Pre
```

```
Preemption disabled
```

Task 4

Configure the appropriate router/s such that as long as R7 is up, it is the active router, if R7 goes down, R8 should take the active role and become the active router, but if R7 comes back up, R7 should become the active router and R8 should transition into the Standby mode.

Let's test and verify the result before configuring the routers:

On R7:

```
R7#Show standby brief
```

```
                P indicates configured to preempt.
                |
Interface      Grp  Pri P State   Active      Standby      Virtual IP
Gi0/1          1    100 Active 10.1.1.8    local        10.1.1.78
```

On R8:

```
R8#Show standby brief
```

```
                P indicates configured to preempt.
                |
Interface      Grp  Pri P State   Active      Standby      Virtual IP
```

```
Gi0/1      1      100      Standby 10.1.1.7      local      10.1.1.78
```

Let's verify the configuration of HSRP:

On R7:

```
R7#Show run int g0/1 | B interface
```

```
interface GigabitEthernet0/1
 ip address 10.1.1.7 255.255.255.0
 standby 1 ip 10.1.1.78
 duplex auto
 speed auto
end
```

```
R7#Show standby
```

```
GigabitEthernet0/1 - Group 1
  State is Active
    2 state changes, last state change 02:10:21
  Virtual IP address is 10.1.1.78
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.992 secs
  Preemption disabled
  Active router is local
  Standby router is 10.1.1.8, priority 100 (expires in 8.448 sec)
  Priority 100 (default 100)
  Group name is "hsrp-Gi0/1-1" (default)
```

On R8:

```
R8#Show run int g0/1 | B interface
```

```
interface GigabitEthernet0/1
 ip address 10.1.1.8 255.255.255.0
 standby 1 ip 10.1.1.78
 duplex auto
 speed auto
end
```

```
R8#Show standby
```

```
GigabitEthernet0/1 - Group 1
```


State is Standby

4 state changes, last state change 02:10:04
Virtual IP address is 10.1.1.78
Active virtual MAC address is 0000.0c07.ac01
Local virtual MAC address is 0000.0c07.ac01 (v1 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 0.256 secs

Preemption disabled

Active router is 10.1.1.7, priority 100 (expires in 7.968 sec)
Standby router is local
Priority 100 (default 100)
Group name is "hsrp-Gi0/1-1" (default)

To configure this task, preemption must be configured. Preemption enables the HSRP router with the highest priority to immediately become the active router.

Priority is determined first by the configured priority value, in which case the router with the highest priority value will become the active router, and in an event of a tie, the router with the highest IP address will have the highest priority.

On R7:

```
R7(config)#Int G0/1
R7(config-if)#Standby 1 priority 101
```

NOTE: R3 is still the active router, to verify this:

```
R7(config-if)#Do Show standby brief
```

P indicates configured to preempt.

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	101		Active	local	10.1.1.8	10.1.1.78

NOW.....let's configure the "Standby preempt" command so as long as the local router (R7) is up, it is the active router, and verify the result:

On R7:

```
R7(config)#int g0/1
R7(config-if)#Standby 1 preempt
```

To test the configuration:

On R7:

```
R7(config)#Int G0/1
R7(config-if)#Shut
```

On R8:

```
R8#Show standby brie
```

P indicates configured to preempt.

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Fa0/0	1	100		Active	local	unknown	10.1.1.23

NOTE: The Standby router is unknown, because the G0/1 interface of R7 is shutdown.

Let's enable the G0/1 interface of R7:

On R7:

```
R7(config)#Int G0/1
R7(config-if)#No shut
```

You should see the following console messages:

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
```

It may take few seconds to see the following console message:

```
%HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Listen -> Active
```

To verify the configuration:

On R8:

```
R8#Show standby brie
```

P indicates configured to preempt.

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	101	P	Active	local	10.1.1.8	10.1.1.78

Task 5

Configure the appropriate router/s such that if the R7's connection to R1 goes down, R8 will become the active router.

If R7's connection to R1 goes down, R7 should not remain as the active router because it will blackhole all the destinations, therefore, let's configure R8 to become the active router if R7's connection to R1 is down.

On R7:

```
R7(config)#Track 10 interface g0/0 line-protocol
```

```
R7(config)#Int G0/1
```

```
R7(config-if)#Standby 1 track 10 decrement 2
```

On R8:

```
R8(config)#Int G0/1
```

```
R8(config-if)#Standby 1 preempt
```

The above command configures interface tracking for the HSRP group 1. This command instructs the router for group 1 to track interface G0/1, and if it goes down, the process should reduce the priority of this router by 2, which means:

101 (The configured priority) minus 2 (From the Standby track command) = 99

Since the default priority is 100. If R8 is configured with the "Standby 1 preempt" command, it will become the active router.

To test the configuration:

On R7:

```
R7(config)#Int G0/1
```

```
R7(config-if)#Shut
```

You should also see the following console message:

```
%TRACK-6-STATE: 10 interface Gi0/0 line-protocol Up -> Down
```

```
%HSRP-5-STATECHANGE: GigabitEthernet0/1 Grp 1 state Active -> Speak
```

%HSRP-5-STATECHANGE: GigabitEthernet0/1 Grp 1 state Speak -> Standby

R7#Show standby brief

P indicates configured to preempt.

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	99	P	Standby	10.1.1.8	local	10.1.1.78

To test the configuration further:

On R7:

R7(config)#Int G0/0

R7(config-if)#No shut

%TRACK-6-STATE: 10 interface Gi0/0 line-protocol Down -> Up

%LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0,
changed state to up

%HSRP-5-STATECHANGE: GigabitEthernet0/1 Grp 1 state Standby -> Active

%LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel178, changed state
to up

R7#Show standby brief

P indicates configured to preempt.

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	101	P	Active	local	10.1.1.8	10.1.1.78

Task 6

Configure the hello and hold time interval for groups 1 to 5 and 15 seconds respectively.

A "Show Standby" command reveals that the default hello and hold timer is set to 3 and 10 seconds respectively, to change these timers perform the following:

On R7 and R8:

Rx#Show Standby | Inc Hello

```
Hello time 3 sec, hold time 10 sec
```

To change the timers:

```
Rx(config)#Int G0/1  
Rx(config-if)#Standby 1 timers 5 15
```

To verify the configuration:

On R8:

```
R8#Show Standby | Inc Hello
```

```
Hello time 5 sec, hold time 15 sec
```

Task 7

Configure the name of the HSRP group 1 to "R7-8-HSRP-G1".

On R7 and R8:

```
Rx(config)#int g0/1  
Rx(config-if)#Standby 1 name R7-8-HSRP-G1
```

To verify the configuration:

On R8:

```
R8#Show standby | Inc name
```

```
Group name is "R7-8-HSRP-G1" (cfgd)
```

NOTE: The "default" is replaced with (cfgd).

Task 8

Ensure that the routers send SNMP traps to the NMS located at 1.2.3.4 for HSRP.

On Both routers:

```
Rx (config) #snmp-server enable traps hsrp
Rx (config) #snmp-server host 1.2.3.4 public hsrp
```

The first command enables the router to send SNMP traps, informs, and HSRP notification.

The second command specifies the recipient of an SNMP notification operation, and that hsrp notification is sent to the host with an IP address of 1.2.3.4.

Task 9

Configure VLAN 478 such that half of the users use R7 as the primary default gateway and R8 as the backup, the other half should use R8 as the primary and R7 as their backup default gateway. Use the following IP addresses to accomplish this task:

10.1.1.77 and 10.1.1.88

Multiple HSRP groups enables load-sharing with a given network, with this feature configured, redundancy can also be fully utilized. With this feature R7 can be the active router for group 1 and standby for group 2, whereas, R8 can be the active router for Group 2 and the standby router for Group 1. This feature can be used for multiple VLANs or a single VLAN.

On R7:

```
R7 (config) #Int G0/1
R7 (config-if) #Standby 1 ip 10.1.1.77
R7 (config-if) #Standby 2 ip 10.1.1.88
```

On R8:

```
R8 (config) #Int G0/1
R8 (config-if) #Standby 1 ip 10.1.1.77
R8 (config-if) #Standby 2 priority 101
R8 (config-if) #Standby 2 preempt
R8 (config-if) #Standby 2 ip 10.1.1.88
```

To see the configuration of these routers:

On R7:

R7#Show run int g0/1 | B interface

```
interface FastEthernet0/0
ip address 10.1.1.7 255.255.255.0
standby 1 ip 10.1.1.77
standby 1 timers 5 15
standby 1 priority 101
standby 1 preempt
standby 1 name R7-8-HSRP-G1
standby 1 track 10 decrement 2
standby 2 ip 10.1.1.88
end
```

On R8:

R8#Show run int F0/0 | B interface

```
interface FastEthernet0/0
ip address 10.1.1.8 255.255.255.0
standby 1 ip 10.1.1.77
standby 1 timers 5 15
standby 1 preempt
standby 1 name R7-8-HSRP-G1
standby 2 ip 10.1.1.88
standby 2 priority 101
standby 2 preempt
end
```

To verify the configuration:

On R7:

R7#Show Standby brief

P indicates configured to preempt.

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	101	P	Active	local	10.1.1.8	10.1.1.77
Gi0/1	2	100		Standby	10.1.1.8	local	10.1.1.88

On R8:

R8#Show standby brief

P indicates configured to preempt.

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP

Gi0/1	1	100	P	Standby	10.1.1.7	local	10.1.1.77
Gi0/1	2	101	P	Active	local	10.1.1.7	10.1.1.88

Task 10

Configure HSRP group 1 to use plain text authenticated using "Cisco" as the password.

HSRP ignores unauthenticated HSRP protocol messages. The default authentication type is test authentication, and the string is "cisco". This means that both ends **MUST** be configured with the same string, unless **ONLY** one end is configured using "cisco" as the string.

HSRP authentication protects against false HSRP hello packets causing a DoS attacks. A host can send HSRP hellos with a higher priority to become the active router.

HSRP offers two kinds of authentications:

- Plain text
- MD5

MD5 authentication provides greater security than the alternative plain text authentication scheme. This authentication can be applied to a given group, meaning that different groups can have different authentication strings.

In order to test the default string, let's configure authentication on R7 and use "cisco" as the string:

On R7:

```
R7(config)#Int G0/1
R7(config-if)#Standby 1 authentication cisco
```

Nothing happened. Let's check the status of the HSRP groups:

```
R7#Show standby brief
```

```

                P indicates configured to preempt.
                |
Interface    Grp  Pri  P State   Active      Standby      Virtual IP
Gi0/1        1    101 P Active  local      10.1.1.8     10.1.1.77
Gi0/1        2    100 Standby 10.1.1.8   local      10.1.1.88

```

On R8:

R8#Show standby brief

P indicates configured to preempt.

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	100	P	Standby	10.1.1.7	local	10.1.1.77
Gi0/1	2	101	P	Active	local	10.1.1.7	10.1.1.88

Let's verify the configuration in detail:

On R7:

R7#Show Standby | Inc Authentication

R7#

The reason we do not see authentication in the output of any show command is because the "cisco" string is the default, to verify this information, let's configure the string to be "Cisco" as the task stated. NOTE: The letter "C" in "Cisco" is now upper case.

On R7:

R7(config)#Int G0/1

R7(config-if)#Standby 1 authentication Cisco

You should see the following console message:

%HSRP-4-BADAUTH: Bad authentication from 10.1.1.8, group 1, remote state Standby

To verify the configuration

On R7:

R7#Show Standby brief

P indicates configured to preempt.

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	101	P	Active	local	unknown	10.1.1.77
Gi0/1	2	100		Standby	10.1.1.8	local	10.1.1.88

On R8:

R8#Show Standby brief

P indicates configured to preempt.

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	100	P	Active	local	unknown	10.1.1.77
Gi0/1	2	101	P	Active	local	10.1.1.7	10.1.1.88

Let's verify the configuration in detail:

On R7:

R7#Show Standby | Inc Authentication

Authentication text "Cisco"

Let's configure R8 to authenticate using "Cisco" as the string:

On R8:

R8(config)#Int G0/1

R8(config-if)#Standby 1 authentication Cisco

You should see the following console message:

%HSRP-5-STATECHANGE: GigabitEthernet0/1 Grp 1 state Active -> Speak

%HSRP-5-STATECHANGE: GigabitEthernet0/1 Grp 1 state Speak -> Standby

To verify the configuration

On R8:

R8#Show standby brief

P indicates configured to preempt.

|

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	100	P	Standby	10.1.1.7	local	10.1.1.77
Gi0/1	2	101	P	Active	local	10.1.1.7	10.1.1.88

On R2:

R2#Show Standby brief

P indicates configured to preempt.

|

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	101	P	Active	local	10.1.1.8	10.1.1.77
Gi0/1	2	100		Standby	10.1.1.8	local	10.1.1.88

Task 11

Configure HSRP group 2 to be MD5 authenticated using "HSRP" as the password.

On R7 and R8:

```
Rx(config)#Key chain tst
Rx(config-keychain)#Key 1
Rx(config-keychain-key)#Key-string HSRP
```

```
Rx(config)#Int F0/0
Rx(config-if)#Standby 2 authentication md5 key-chain tst
```

To verify the configuration

On R8:

R8#Show Standby brief

P indicates configured to preempt.
|

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	100	P	Standby	10.1.1.7	local	10.1.1.77
Gi0/1	2	101	P	Active	local	10.1.1.7	10.1.1.88

On R7:

R7#Show Standby brief

P indicates configured to preempt.
|

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Gi0/1	1	101	P	Active	local	10.1.1.8	10.1.1.77
Gi0/1	2	100		Standby	10.1.1.8	local	10.1.1.88

To verify the configuration:

On R8:

R8#Show Standby | Inc Authentication|Group

```
GigabitEthernet0/1 - Group 1
Authentication text, string "Cisco"
Group name is "R7-8-HSRP-G1" (cfgd)
```

```
GigabitEthernet0/1 - Group 2
Authentication MD5, key-chain "tst"
Group name is "hsrp-Gi0/1-2" (default)
```

Task 12

The G0/1 interfaces of R7 and R8 are connected to SW2's port F0/7 and SW3's port F0/8. Configure SW2's F0/8 and SW3's F0/8 interfaces with "Port-Security" using the default parameters. Configure HSRP to accommodate this request.

The default parameters of "Port-Security" only allows a single MAC address to be attached, how are we going to configure this task, since HSRP will also use a virtual MAC address. Let's see the Mac-address-Table of SW2:

On SW2:

```
SW2#Show mac-address-table dynamic vlan 478
      Mac Address Table
```

Vlan	Mac Address	Type	Ports
478	0000.0c07.ac01	DYNAMIC	Fa0/7 → The VMA for G1
478	0000.0c07.ac02	DYNAMIC	Fa0/23
478	000c.858b.7a17	DYNAMIC	Fa0/23
478	0017.59ce.02b9	DYNAMIC	Fa0/4
478	24e9.b3ab.4b21	DYNAMIC	Fa0/7 → R7's G0/1's MAC Address
478	3c08.f6a2.bc81	DYNAMIC	Fa0/23

Total Mac Addresses for this criterion: 6

On SW3:

```
SW3#Show mac address-table dynamic vlan 478
      Mac Address Table
```

Vlan	Mac Address	Type	Ports
478	0000.0c07.ac01	DYNAMIC	Fa0/23
478	0000.0c07.ac02	DYNAMIC	Fa0/8 → The VMA for G2
478	0017.59ce.02b9	DYNAMIC	Fa0/23
478	24e9.b3ab.4b21	DYNAMIC	Fa0/23

```
478      3c08.f6a2.bc81      DYNAMIC      Fa0/8 → R8's G0/1's MAC address
Total Mac Addresses for this criterion: 5
```

You can see that each port on the switch has two MAC addresses, the HSRP's VMAC and the MAC address of the router. Therefore, if the "Port-Security" is configured on F0/7 interface of SW2 and the SW3's F0/8 interface, the ports will transition into "err-disable" state.

On R7 and R8:

```
Rx(config)#int G0/1
Rx(config-if)#Standby use-bia
Rx(config-if)#Shut
Rx(config-if)#No Shut
```

To verify the configuration:

On SW2:

```
SW2#Show mac-address-table dynamic vlan 478
      Mac Address Table
```

Vlan	Mac Address	Type	Ports
478	0000.0c07.ac01	DYNAMIC	Fa0/23
478	0000.0c07.ac02	DYNAMIC	Fa0/23
478	000c.858b.7a17	DYNAMIC	Fa0/23
478	0017.59ce.02b9	DYNAMIC	Fa0/4
478	24e9.b3ab.4b21	DYNAMIC	Fa0/7
478	3c08.f6a2.bc81	DYNAMIC	Fa0/23

```
Total Mac Addresses for this criterion: 6
```

NOTE: HSRP uses the MAC addresses of the routers instead of the default HSRP MAC addresses. To verify:

```
R7#Show int g0/1 | Inc bia
```

```
Hardware is CN Gigabit Ethernet, address 24e9.b3ab.4b21 (bia
24e9.b3ab.4b21)
```

Let's enable port-security on the F0/7 of SW2 and the F0/8 interfaces of SW3:

On SW2:

```
SW2 (config)#Int F0/7
SW2 (config-if)#Switchport port-security
```

To verify the configuration:

On SW2:

SW2#**Show port-security interface F0/7**

```
Port Security           : Enabled
Port Status             : Secure-up
Violation Mode          : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses : 0
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 24e9.b3ab.4b21:478
Security Violation Count : 0
```

The MAC address of R7's G0/1 interface

The VLAN ID

On SW3:

SW3(config)#**Int F0/8**

SW3(config-if)#**Switchport port-security**

SW3#**Show port-security interface F0/8**
Mac Address Table

Vlan	Mac Address	Type	Ports
478	0017.59ce.02b9	DYNAMIC	Fa0/23
478	24e9.b3ab.4b21	DYNAMIC	Fa0/23
478	3c08.f6a2.bc81	DYNAMIC	Fa0/8

Total Mac Addresses for this criterion: 3

SW3#**Show port-security int f0/8**

```
Port Security           : Enabled
Port Status             : Secure-up
Violation Mode          : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
```

```
Configured MAC Addresses      : 0
Sticky MAC Addresses          : 0
Last Source Address:Vlan      : 3c08.f6a2.bc81:478
Security Violation Count      : 0
```

To test this feature properly, let's remove the "Standby use-bia", and verify the result:

On R7:

```
R7(config)#int G0/1
R7(config-if)#No Standby use-bia
R7(config-if)#Shut
R7(config-if)#No Shut
```

You should see the following console messages:

```
%LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to down
```

NOTE: The interface came up and went down and stayed down, let's see why:

On SW2:

```
SW2#Show port-security interface F0/7
```

```
Port Security                : Enabled
Port Status                   : Secure-shutdown
Violation Mode                : Shutdown
Aging Time                    : 0 mins
Aging Type                    : Absolute
SecureStatic Address Aging    : Disabled
Maximum MAC Addresses         : 1
Total MAC Addresses           : 0
Configured MAC Addresses      : 0
Sticky MAC Addresses          : 0
Last Source Address:Vlan      : 0000.0c07.ac01:478
Security Violation Count      : 1
```

```
SW2#Show interface F0/7 status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Fa0/7		err-disabled	478	auto	auto	10/100BaseTX

That's exactly what we expected to see. Let's re-configure the "Standby use-bia" command.

On R7:

```
R7(config)#int G0/1
R7(config-if)#Standby use-bia
R7(config-if)#Shut
R7(config-if)#No Shut
```

On SW2:

```
SW2(config)#int f0/7
SW2(config-if)#shu
SW2(config-if)#No shu
```

To verify the configuration:

On SW2:

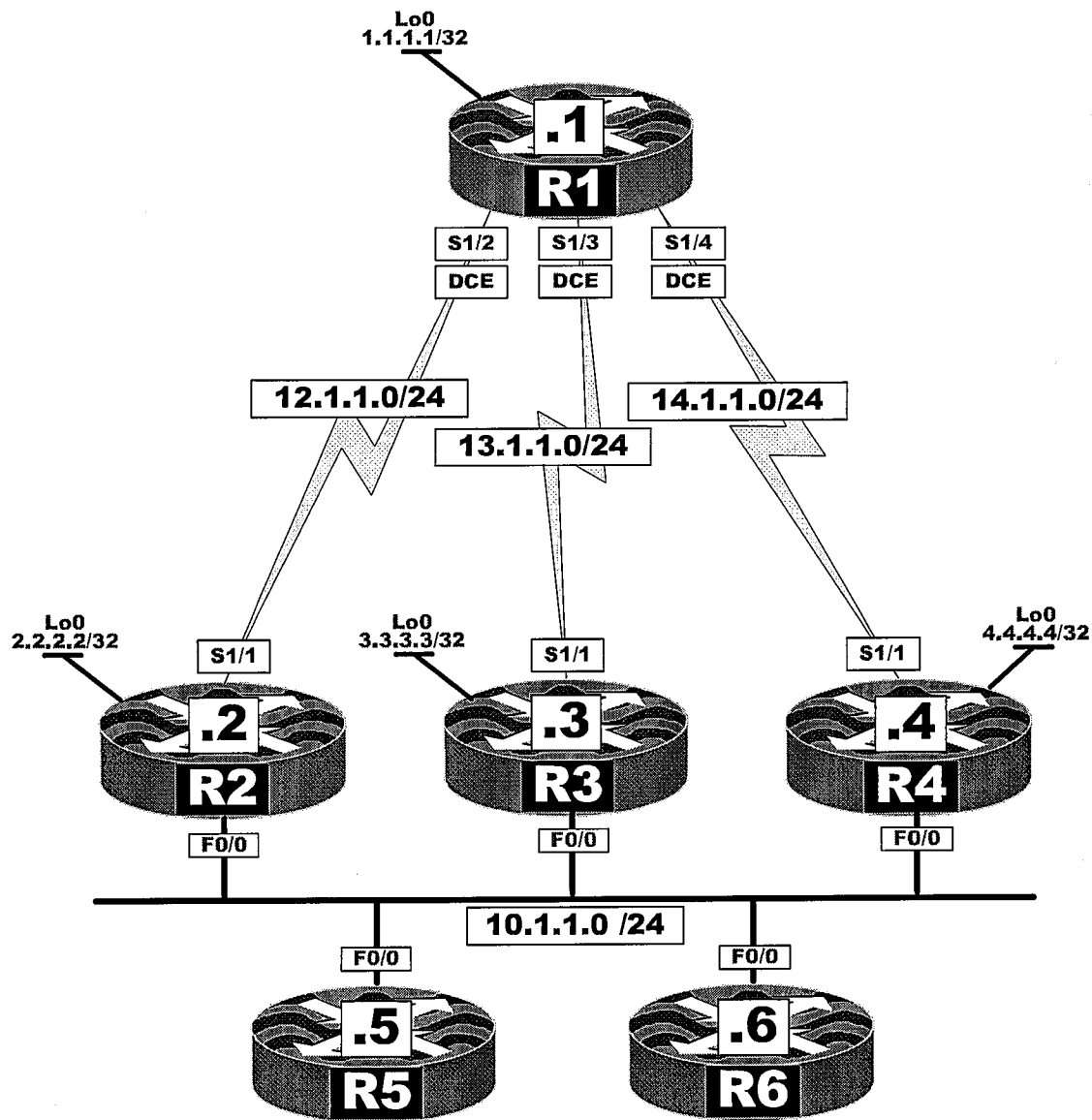
```
SW2#Show port-security inter f0/7
```

```
Port Security           : Enabled
Port Status             : Secure-up
Violation Mode          : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses : 0
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 24e9.b3ab.4b21:478
Security Violation Count : 0
```

Task 13

Erase the startup config and reload the routers before proceeding to the next lab.

Lab 2 – VRRP Configuration



Task 1

Configure the routers based on the above diagram and the following policy:

- Assign 0000.2222.2222, 0000.3333.3333, and 0000.4444.4444 MAC addresses to R2, R3 and R4 respectively.

On SW1:

```
SW1 (config) #Int range F0/2-6
SW1 (config-if-range) #Switchport mode access
SW1 (config-if-range) #Switchport access vlan 100
SW1 (config-if-range) #No shu
```

On R1:

```
R1 (config) #int s1/2
R1 (config-if) #clock rate 64000
R1 (config-if) #ip addr 12.1.1.1 255.255.255.0
R1 (config-if) #No shut

R1 (config) #Int s1/3
R1 (config-if) #clock rate 64000
R1 (config-if) #ip addr 13.1.1.1 255.255.255.0
R1 (config-if) #No shut

R1 (config) #int s1/4
R1 (config-if) #clock rate 64000
R1 (config-if) #ip addr 14.1.1.1 255.255.255.0
R1 (config-if) #No shut

R1 (config-if) #Int Lo0
R1 (config-if) #Ip addr 1.1.1.1 255.255.255.255
```

On R2:

```
R2 (config) #int s1/1
R2 (config-if) #ip addr 12.1.1.2 255.255.255.0
R2 (config-if) #No shut

R2 (config) #Int F0/0
R2 (config-if) #Ip addr 10.1.1.2 255.255.255.0
R2 (config-if) #mac-address 0000.2222.2222
R2 (config-if) #No shut
```

```
R2(config)#Int lo0
R2(config-if)#ip addr 2.2.2.2 255.255.255.255
```

On R3:

```
R3(config)#Int S1/1
R3(config-if)#Ip addr 13.1.1.3 255.255.255.0
R3(config-if)#No shut
```

```
R3(config)#int F0/0
R3(config-if)#Ip addr 10.1.1.3 255.255.255.0
R3(config-if)#mac-address 0000.3333.3333
R3(config-if)#No shut
```

```
R3(config)#Int Lo0
R3(config-if)#Ip addr 3.3.3.3 255.255.255.255
```

On R4:

```
R4(config)#Int S1/1
R4(config-if)#ip addr 14.1.1.4 255.255.255.0
R4(config-if)#No shut
```

```
R4(config)#Int F0/0
R4(config-if)#Ip addr 10.1.1.4 255.255.255.0
R4(config-if)#mac-address 0000.4444.4444
R4(config-if)#No shut
```

```
R4(config)#Int Lo0
R4(config-if)#Ip addr 4.4.4.4 255.255.255.255
```

On R5:

```
R5(config)#Int F0/0
R5(config-if)#Ip addr 10.1.1.5 255.255.255.0
R5(config-if)#No shut
```

On R6:

```
R6(config)#Int F0/0
R6(config-if)#Ip addr 10.1.1.5 255.255.255.0
R6(config-if)#No shut
```

To verify the configuration:

On R1:

R1#Ping 12.1.1.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.2, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 52/52/56 ms

R1#Ping 13.1.1.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 13.1.1.3, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 52/53/56 ms

R1#Ping 14.1.1.4

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 14.1.1.4, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 52/53/56 ms

On R2:

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

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 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms

R2#Ping 10.1.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.5, timeout is 2 seconds:

.!!!!

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

Task 2

Configure RIPv2 on R1, R2, R3 and R4. Advertise their Loopback 0 and their serial interfaces in this routing domain. Disable the auto-summarization.

R2, R3 and R4 should be configured to redistribute network 10.1.1.0 /24 into the RIP routing domain.

On R1:

```
R1(config)#Router rip
R1(config-router)#No au
R1(config-router)#ver 2
R1(config-router)#Netw 1.0.0.0
R1(config-router)#Netw 12.0.0.0
R1(config-router)#Netw 13.0.0.0
R1(config-router)#Netw 14.0.0.0
```

On R2:

```
R2(config)#Route-map tst permit 10
R2(config-route-map)#match inter f0/0

R2(config)#Router rip
R2(config-router)#No au
R2(config-router)#ver 2
R2(config-router)#Netw 2.0.0.0
R2(config-router)#Netw 12.0.0.0
R2(config-router)#redistribute connected route-map tst
```

On R3:

```
R3(config)#Route-map tst permit 10
R3(config-route-map)#match inter f0/0

R3(config)#Router rip
R3(config-router)#No au
R3(config-router)#ver 2
R3(config-router)#Netw 3.0.0.0
R3(config-router)#Netw 13.0.0.0
R3(config-router)#redistribute connected route-map tst
```

On R4:

```
R4(config)#Route-map tst permit 10
R4(config-route-map)#match inter f0/0
```

```
R4(config)#Router rip
R4(config-router)#No au
R4(config-router)#ver 2
R4(config-router)#Netw 4.0.0.0
R4(config-router)#Netw 14.0.0.0
R4(config-router)#redistribute connected route-map tst
```

To verify the configuration:

On R3:

```
R3#Show ip route rip | B Gate
Gateway of last resort is not set

    1.0.0.0/32 is subnetted, 1 subnets
R       1.1.1.1 [120/1] via 13.1.1.1, 00:00:25, Serial1/1
    2.0.0.0/32 is subnetted, 1 subnets
R       2.2.2.2 [120/2] via 13.1.1.1, 00:00:25, Serial1/1
    4.0.0.0/32 is subnetted, 1 subnets
R       4.4.4.4 [120/2] via 13.1.1.1, 00:00:25, Serial1/1
    12.0.0.0/24 is subnetted, 1 subnets
R       12.1.1.0 [120/1] via 13.1.1.1, 00:00:25, Serial1/1
    14.0.0.0/24 is subnetted, 1 subnets
R       14.1.1.0 [120/1] via 13.1.1.1, 00:00:25, Serial1/1
```

To verify the configuration:

On R3:

```
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 = 52/53/56 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 100 percent (5/5), round-trip min/avg/max = 104/105/108 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 100 percent (5/5), round-trip min/avg/max = 104/105/108 ms

Task 3

Configure VLAN 100 such that the host/s (R2 - 6) in this VLAN use R2 as the VR (Virtual Router) Master, and R3 and R4 as the VR Backup default gateway. Use the following policy to accomplish this task:

- R2 should be configured as the VR Master.
- R3 and R4 should be configured as the VR Backup routers.
- R3 should be the first backup, and R4 should be the second backup.
- DO NOT change the IP address of any router.

VRRP enables a group of routers to form a single virtual router. The hosts on that segment can then be configured with the IP address of the Virtual router (VR) as their default gateway.

VRRP is very flexible because it is supported on Ethernet, Fast Ethernet, BVI, Gigabit Ethernet, and MPLS VPNs.

Let's say there are three routers in the VR group, R1 (1.1.1.1), R2 (1.1.1.2) and R3 (1.1.1.3). The IP address of the VR is the same as that configured for the Ethernet interface of R1 (1.1.1.1), because R1 is the VR Master, and R2 and R3 are VR Backups.

Since the IP address of R1's Ethernet interface is used, R1 assumes the role of the VR Master; R1, the VR Master, is responsible for forwarding packets sent to this IP address.

Hosts H1 – 3 are configured with the default gateway IP address of R1's Ethernet's IP address. R2 and R3 function as the VR Backups. If the VR Master fails, the router configured with the higher priority will become the VR Master and provide uninterrupted service for the hosts on that LAN. The priority determines which VR Backup will assume the role of VR Master, the priority is configured using the "VRRP Priority" command and the range is 1 – 254. The higher value has more preference. When R1 recovers, it becomes the VR Master again because in VRRP preemption is on by default.

You can configure two routers in a group, VR1, R1 is the owner of the IP address 1.1.1.1, therefore, it's the VR Master, and R2 is the VR Backup. You can configure Hosts H1 and H2 to use that IP address as their default gateway.

For VR2, R2 is the owner of IP address 1.1.1.2 therefore, it's the VR Master, and R1 is the VR Backup to R2. You can configure Hosts H3 and H4 to use that IP address as their default gateway.

VRRP can support up to 255 VRRP Groups on a physical interface. This enables us to have a granular scalability, redundancy and Load sharing; it also allows the use of secondary IP addresses, Preemption, Authentication, and Object Tracking.

VRRP uses a registered Multicast address of 224.1.1.18 and IP protocol number of 112 for its VRRP Advertisements.

On R2:

```
R2 (config) #Int F0/0
R2 (config-if) #vrrp 1 ip 10.1.1.2
```

You should see the following console message:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Init -> Master
```

The console message announces R2 as the VR Master, NOT because it is the only router configured so far; R2 is chosen as the VR Master router because its Ethernet IP address is used as the VIP (Virtual IP address).

```
R2 (config-if) #vrrp 1 prio 250
```

```
% Priority change will have no effect whilst interface is VRRP address owner
```

NOTE: Because the IP address configured is one of the IP addresses configured on R2, R2 is said to be the owner of that IP address, and because of that, R2 will be the VR Master and its priority will be set to 255. We will see later in this lab how the configured priority can play a major role.

On R3:

```
R3 (config) #Int F0/0
R3 (config-if) #vrrp 1 ip 10.1.1.2
```

You can see that the local router is now set as the VR Backup

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Init -> Backup
```

```
R3 (config-if) #vrrp 1 priority 200
```

On R4:


```
R4(config)#Int F0/0
R4(config-if)#vrrp 1 ip 10.1.1.2
```

Based on the following console message, R4 is also a VR Backup router.

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Init -> Backup
```

```
R4(config-if)#vrrp 1 priority 150
```

To verify the configuration:

On R2:

```
R2#Show vrrp
```

```
FastEthernet0/0 - Group 1
  State is Master
  Virtual IP address is 10.1.1.2
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 255 (cfgd 250)
  Master Router is 10.1.1.2 (local), priority is 255
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.003 sec
```

You can see that the local router is the VR Master, and the VIP is 10.1.1.2 and the VMAC is 0000.5e00.01xx, where xx is the group ID.

One major difference between VRRP and HSRP is the preemption, in HSRP the preemption is disabled by default and it can be enabled manually, whereas, in VRRP, the preemption is enabled by default and it can be disabled manually.

Since the local router is the VR Master its priority will be set to 255 unless it is statically configured to be something else, in this case, it is configured to be 250.

The above output also displays the IP address of the master router, and some timers which will be discussed and configured later.

On R3:

```
R3#Show vrrp
```

```
FastEthernet0/0 - Group 1
```

```
State is Backup
Virtual IP address is 10.1.1.2
Virtual MAC address is 0000.5e00.0101
Advertisement interval is 1.000 sec
Preemption enabled
Priority is 200
Master Router is 10.1.1.2, priority is 255
Master Advertisement interval is 1.000 sec
Master Down interval is 3.218 sec (expires in 2.430 sec)
```

The output of the above show command reveals the following:

- The local router is a VR Backup
- VIP is 10.1.1.2
- VMAC is 0000.5e00.0101
- Advertisement interval is set to 1 second
- Preemption is enabled; unlike HSRP, the preemption is always enabled by default in VRRP.
- The configured priority is 200
- In the next three lines, it identifies the VR Master router's IP address, default priority and some timers.

On R4:

R4#**Show vrrp**

FastEthernet0/0 - Group 1

```
State is Backup
Virtual IP address is 10.1.1.2
Virtual MAC address is 0000.5e00.0101
Advertisement interval is 1.000 sec
Preemption enabled
Priority is 150
Master Router is 10.1.1.2, priority is 255
Master Advertisement interval is 1.000 sec
Master Down interval is 3.414 sec (expires in 3.022 sec)
```

The output of the above show command reveals the following:

- The local router is a VR Backup
- VIP is 10.1.1.2
- VMAC is 0000.5e00.0101
- Advertisement interval is set to 1 second
- Preemption is enabled; unlike HSRP, the preemption is always enabled by default in VRRP.
- The configured priority is 150. The configured priority on the other VR Back router (R3) is

configured to be 200, this means that R3 is the first backup and R4 is the second backup, because R4's priority is lower than R3's priority.

- In the next three lines, it identifies the VR Master router's IP address, default priority and some timers.

To test the configuration, R5 is configured with a default gateway pointing to the VIP address (10.1.1.2):

On R5:

```
R5(config)#IP route 0.0.0.0 0.0.0.0 10.1.1.2
```

To verify the configuration:

Since Traceroute will be used in this lab to verify the configuration, and the ICMP unreachable packets are rate-limited, let's disable the rate-limiting so we can get a quicker response.

On All Routers:

```
Rx(config)#No ip icmp rate-limit unreachable
```

On R5:

```
R5#Traceroute 1.1.1.1 Numeric
```

Type escape sequence to abort.

Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

```
 1 10.1.1.2 4 msec 0 msec 4 msec
```

```
 2 12.1.1.1 16 msec * 12 msec
```

```
R5#Show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	14	0000.5e00.0101	ARPA	FastEthernet0/0
Internet	10.1.1.3	0	0000.3333.3333	ARPA	FastEthernet0/0
Internet	10.1.1.5	-	001b.d4be.69d0	ARPA	FastEthernet0/0

We can see that R5 took the path through R2; its default gateway to reach 1.1.1.1/32 prefix. Let's shutdown the F0/0 interface of R2 and test again:

On R2:

```
R2(config)#Int F0/0
```

```
R2(config-if)#Shut
```

You should see the following console message:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Master -> Init
```

```
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to  
administratively down
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed  
state to down
```

Let's Traceroute to 1.1.1.1/32 prefix again:

On R5:

```
R5#Traceroute 1.1.1.1 Numeric
```

Type escape sequence to abort.

Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

```
 1 10.1.1.3 0 msec 4 msec 0 msec  
 2 13.1.1.1 20 msec * 12 msec
```

```
R5#Show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	0	0000.5e00.0101	ARPA	FastEthernet0/0
Internet	10.1.1.3	2	0000.3333.3333	ARPA	FastEthernet0/0
Internet	10.1.1.4	0	0000.4444.4444	ARPA	FastEthernet0/0
Internet	10.1.1.5	-	001b.d4be.69d0	ARPA	FastEthernet0/0

We can clearly see that R5 is going through R3. To clear the ARP table and repeat the Traceroute on R5:

On R5:

```
R5(config)#Int F0/0
```

```
R5(config-if)#Shut
```

Wait for the link to go down before issuing the "No shut" command.

```
R5(config-if)#No shut
```

On R5:

```
R5#Traceroute 1.1.1.1 Numeric
```

Type escape sequence to abort.
Tracing the route to 1.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
1 10.1.1.3 0 msec 0 msec 0 msec
2 13.1.1.1 16 msec * 12 msec

R5#Show arp

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	0	0000.5e00.0101	ARPA	FastEthernet0/0
Internet	10.1.1.5	-	001b.d4be.69d0	ARPA	FastEthernet0/0

R5#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 = 52/52/52 ms

R5#Show arp

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	1	0000.5e00.0101	ARPA	FastEthernet0/0
Internet	10.1.1.5	-	001b.d4be.69d0	ARPA	FastEthernet0/0

On R3:

R3#Show vrrp

FastEthernet0/0 - Group 1
State is Master
Virtual IP address is 10.1.1.2
Virtual MAC address is 0000.5e00.0101
Advertisement interval is 1.000 sec
Preemption enabled
Priority is 200
Master Router is 10.1.1.3 (local), priority is 200
Master Advertisement interval is 1.000 sec
Master Down interval is 3.218 sec

Sure enough R3 is the VR Master, let's check R4:

On R4:

R4#Show vrrp

FastEthernet0/0 - Group 1

State is Backup

Virtual IP address is 10.1.1.2

Virtual MAC address is 0000.5e00.0101

Advertisement interval is 1.000 sec

Preemption enabled

Priority is 150

Master Router is 10.1.1.3, priority is 200

Master Advertisement interval is 1.000 sec

Master Down interval is 3.414 sec (expires in 3.366 sec)

R4 is the VR backup, because it has a lower vrrp priority than R3; R3's priority is 200, and R4's priority is 150. Let's see what happened to R2:

On R2:

R2#Show vrrp

FastEthernet0/0 - Group 1

State is Init

Virtual IP address is 10.1.1.2

Virtual MAC address is 0000.5e00.0101

Advertisement interval is 1.000 sec

Preemption enabled

Priority is 255 (cfgd 250)

Master Router is unknown, priority is unknown

Master Advertisement interval is unknown

Master Down interval is unknown

Since the X-VR Master's F0/0 interface is down, it is stuck in the "init" state.

Let's Shutdown the F0/0 interface of R3, if VRRP works as advertised, R4 should transition into the new VR Master:

On R3:

R3(config)#Int F0/0

R3(config-if)#Shut

We can see from the console messages on R3 that it is also stuck in the "init" state:

%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Master -> Init

Let's verify the VRRP on R4:

On R4:

You should see the following console message stating that the local router is the Master:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Backup -> Master
```

```
R4#Show vrrp
```

```
FastEthernet0/0 - Group 1
```

```
State is Master
```

```
Virtual IP address is 10.1.1.2
```

```
Virtual MAC address is 0000.5e00.0101
```

```
Advertisement interval is 1.000 sec
```

```
Preemption enabled
```

```
Priority is 150
```

```
Master Router is 10.1.1.4 (local), priority is 150
```

```
Master Advertisement interval is 1.000 sec
```

```
Master Down interval is 3.414 sec
```

Let's configure a static default route on R6 and point it to the VR's IP address.

On R6:

```
R6(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.2
```

```
R6#Traceroute 1.1.1.1 Numeric
```

```
Type escape sequence to abort.
```

```
Tracing the route to 1.1.1.1
```

```
VRF info: (vrf in name/id, vrf out name/id)
```

```
1 10.1.1.4 0 msec 4 msec 0 msec
```

```
2 14.1.1.1 16 msec * 16 msec
```

```
R6#Show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	5	0000.5e00.0101	ARPA	FastEthernet0/0
Internet	10.1.1.4	2	0000.4444.4444	ARPA	FastEthernet0/0
Internet	10.1.1.6	-	0017.5aad.52aa	ARPA	FastEthernet0/0

Let's test the preemption which is enabled by default. Let's "No Shut" the F0/0 interface of R3:

On R3:

```
R3(config)#Int F0/0
```

```
R3(config-if)#No shut
```

You should see the following console messages:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Init -> Backup
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Backup -> Master
```

Yes, as predicted, R3 is the VR Master. If R4's console messages are checked, we should see that it has transitioned into VR Backup:

On R4:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Master -> Backup
```

To verify and test the configuration:

On R6:

```
R6(config)#Int F0/0
```

```
R6(config-if)#Shut
```

Wait for the link to go down before issuing the "No shut" command.

```
R6(config-if)#No shut
```

```
R6#Traceroute 1.1.1.1 Numeric
```

Type escape sequence to abort.

Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

```
 1 10.1.1.3 0 msec 4 msec 0 msec
 2 13.1.1.1 16 msec * 12 msec
```

```
R6#Show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	1	0000.5e00.0101	ARPA	FastEthernet0/0
Internet	10.1.1.6	-	0017.5aad.52aa	ARPA	FastEthernet0/0

Finally, let's "No shut" the F0/0 interface of R2:

On R2:

```
R2 (config) #Int F0/0
R2 (config-if) #No shut
```

You should see the following console messages on R2:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Init -> Master
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
```

IMPORTANT.....if the priority was not configured on the VR Backup routers (R3 and R4), the router with the highest IP address will become the first backup, in this case R4 will become the first backup.

Task 4

Configure the appropriate router/s such that if the R2's connection to R1 goes down, the first backup router (R3) will become the active router.

On R2:

```
R2 (config) #Track 21 int S1/1 line-protocol
```

```
R2 (config) #Int F0/0
R2 (config-if) #vrrp 1 track 21 decrement 55
```

```
% tracking not supported on IP address owner
```

WOW..... Tjhis didn't work, how do we accomplish this task?

Since the error message states that the tracking is NOT supported on IP address owner, we can use an unused IP address from the same subnet? Let's configure and verify:

```
R2 (config) #Int F0/0
R2 (config-if) #No vrrp 1
R2 (config-if) #vrrp 1 ip 10.1.1.100
R2 (config-if) #vrrp 1 priority 254
R2 (config-if) #vrrp 1 track 21 decrement 55
```

On R3:

```
R3(config)#Int F0/0
R3(config-if)#No vrrp 1
R3(config-if)#Vrrp 1 ip 10.1.1.100
R3(config-if)#Vrrp 1 priority 200
```

You should see the following console message:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Init -> Backup
```

On R4:

```
R4(config)#Int F0/0
R4(config-if)#No vrrp 1
R4(config-if)#Vrrp 1 ip 10.1.1.100
R4(config-if)#Vrrp 1 priority 150
```

You should see the following console message:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Init -> Backup
```

On R5 and R6:

We should remove the previous static route on R5 and R6 that was pointing to 10.1.1.2 and configure a new one that points to the new default gateway IP address (10.1.1.100):

```
Rx(config)#No ip route 0.0.0.0 0.0.0.0
Rx(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.100
```

To verify and test the configuration:

On R2:

```
R2#Show vrrp
```

```
FastEthernet0/0 - Group 1
```

```
State is Master
```

```
Virtual IP address is 10.1.1.100
```

```
Virtual MAC address is 0000.5e00.0101
```

```
Advertisement interval is 1.000 sec
```

```
Preemption enabled
```

```
Priority is 254
```

```
Track object 21 state Up decrement 55
```

Master Router is 10.1.1.2 (local), priority is 254
Master Advertisement interval is 1.000 sec
Master Down interval is 3.007 sec

On R2:

```
R2(config)#Int S1/1  
R2(config-subif)#Shut
```

You should see the following console messages:

%TRACKING-5-STATE: 21 interface Se0/0.21 line-protocol Up->Down

%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Master -> Backup

To verify the configuration:

On R2:

```
R2#Show vrrp brief
```

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	199	3007		Y	Backup	10.1.1.3	10.1.1.100

We can see that the priority was decremented by 55, and the local router transitioned to VR Backup, and R3 (10.1.1.3) became the VR Master.

To test the configuration:

On R5:

To clear the ARP table:

```
R5(config)#Int F0/0  
R5(config-if)#Shut
```

Wait for the link to go down, and then:

```
R5(config-if)#No shut
```

```
R5#Traceroute 1.1.1.1 Numeric
```

Type escape sequence to abort.
Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

```
1 10.1.1.3 0 msec 0 msec 4 msec
2 13.1.1.1 16 msec * 12 msec
```

R5#Show arp

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.5	-	001b.d4be.69d0	ARPA	FastEthernet0/0
Internet	10.1.1.100	0	0000.5e00.0101	ARPA	FastEthernet0/0

What will happen if the F0/0 interface of R3 goes down? Let's test this:

On R3:

```
R3(config)#Int F0/0
R3(config-if)#Shut
```

You should see the following console message:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Master -> Init
```

Who is the VR Master? Let's check:

On R4:

R4#Show vrrp brief

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	150	3414		Y	Backup	10.1.1.2	10.1.1.100

R4 is the backup; if R4 is the VR Backup, then what is the state of R2? Let's check:

On R2:

Yes, we can see the following console message:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Backup -> Master
```

Let's verify:

R2#Show vrrp brief

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	199	3007		Y	Master	10.1.1.2	10.1.1.100

Sure enough, R2 is the master.

To test the configuration:

On R5:

R5#**Traceroute 1.1.1.1 Numeric**

Type escape sequence to abort.
Tracing the route to 1.1.1.1

```
 1 10.1.1.2 0 msec 4 msec 0 msec
 2 10.1.1.2 !H  !H  !H
```

R5#**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:
UUUUU
Success rate is 0 percent (0/5)

What does the letter “H” mean?

Character	Description
*	The prob timed out
A	Administratively prohibited (Example access-list)
Q	Source quench (The destination is busy)
I	User interrupted test
U	Port Unreachable
H	Host unreachable
N	Network Unreachable
P	Protocol Unreachable
T	Timeout
?	Unknown Packet Type

So this means that none of the hosts on 10.1.1.0/24 segment have reachability to any network outside of this segment. How can we fix this problem? Why did this happen?

When R2 was configured with Object Tracking, it should have decremented its priority to something below R4's priority which is set to 150, so R4 will take over the role of the VR Master when R3 is down.

Let's reconfigure and test again:

On R3:

```
R3(config)#Int F0/0
R3(config-if)#No shut
```

On R2:

```
R2(config)#int S1/1
R2(config-subif)#No shut
```

The value should be decremented by 105, because $254 - 149 = 105$

The 254 in the above simple math is the existing priority of R2, and the priority of R4 is 150, therefore, in order for R4 to become the second backup, the priority of R2 should be less than R4's priority.

```
R2(config)#Int F0/0
R2(config-if)#Vrrp 1 track 21 decrement 105
```

Let's test again:

On R5:

```
R5#Traceroute 1.1.1.1
```

```
Type escape sequence to abort.
Tracing the route to 1.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.1.2 0 msec 0 msec 4 msec
  2 12.1.1.1 16 msec * 12 msec
```

```
R5#Show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.3	0	0000.3333.3333	ARPA	FastEthernet0/0
Internet	10.1.1.5	-	001b.d4be.69d0	ARPA	FastEthernet0/0
Internet	10.1.1.100	2	0000.5e00.0101	ARPA	FastEthernet0/0

On R2:

```
R2(config)#int S1/1
R2(config-subif)#Shut
```

You should see the following console messages:

```
%TRACKING-5-STATE: 21 interface Se1/1 line-protocol Up->Down
```

```
%LINK-5-CHANGED: Interface Serial1/1, changed state to administratively
```

down

%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Master -> Backup

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to down

R3 should be the master, let's verify:

On R2:

R2#Show vrrp brief

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	149	3007		Y	Backup	10.1.1.3	10.1.1.100

On R3:

R3#Show vrrp brief

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	200	3218		Y	Master	10.1.1.3	10.1.1.100

To test the configuration:

On R5:

To clear the ARP table:

R5(config)#Int F0/0

R5(config-if)#Shut

Wait for the link to go down, and then:

R5(config-if)#No shut

R5#Traceroute 1.1.1.1 Numeric

Type escape sequence to abort.

Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

1	10.1.1.3	4 msec	0 msec	4 msec
2	13.1.1.1	16 msec	*	12 msec

R5#Show arp

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.5	-	001b.d4be.69d0	ARPA	FastEthernet0/0
Internet	10.1.1.100	0	0000.5e00.0101	ARPA	FastEthernet0/0

Let's shutdown the F0/0 interface of R3 and see if R4 takes over as the VR Master:

On R3:

```
R3(config)#Int F0/0
R3(config-if)#Shut
```

You should see the following console messages:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Master -> Init
```

```
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to
administratively down
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to down
```

Let's verify that R4 is the VR Master:

On R4:

Sure enough, we see the console message stating that the local router transitioned from the VR Backup to the VR Master, let's verify:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Backup -> Master
```

```
R4#Show vrrp brief
```

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	150	3414		Y	Master	10.1.1.4	10.1.1.100

Let's test the configuration:

On R6:

```
R6#Traceroute 1.1.1.1 Numeric
```

```
Type escape sequence to abort.
```

```
Tracing the route to 1.1.1.1
```

```
VRF info: (vrf in name/id, vrf out name/id)
```

```
 1 10.1.1.4 0 msec 0 msec 0 msec
```


2 14.1.1.1 16 msec * 12 msec

R6#Show arp

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	22	0000.5e00.0101	ARPA	FastEthernet0/0
Internet	10.1.1.6	-	0017.5aad.52aa	ARPA	FastEthernet0/0
Internet	10.1.1.100	2	0000.5e00.0101	ARPA	FastEthernet0/0

Let's bring up the S1/1, F0/0 interfaces on R2, and the F0/0 interface of R3 before proceeding further:

On R2:

```
R2(config)#Int S1/1
R2(config-subif)#No shut
```

On R3:

```
R3(config)#Int F0/0
R3(config-if)#No shut
```

To verify the configuration:

On R2:

Once we console into R2 we should see the following console message stating that the local router has transitioned into the Master VR:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Backup -> Master
```

R2#Show vrrp brief

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	254	3007		Y	Master	10.1.1.2	10.1.1.100

On R3:

R3#Show vrrp brief

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	200	3218		Y	Backup	10.1.1.2	10.1.1.100

On R4:

R4#Show vrrp brief

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	150	3414		Y	Backup	10.1.1.2	10.1.1.100

Task 5

Ensure that the routers send SNMP traps to the NMS located at 1.2.3.4.

On R2, R3 and R4:

The following command enables the router to send SNMP VRRP notification traps and informs:

```
R4(config)#Snmp-server enable traps vrrp
```

The following command specifies the recipient of an SNMP notification operation:

```
R4(config)#Snmp-server host 1.2.3.4 Public vrrp
```

Task 6

Configure VRRP based on the following policy:

Group 1:

- R2 should be the VR Master, R3 should be the first VR Backup and R4 should be configured as the Second VR Backup.
- This group should use 10.1.1.100 as its VIP.
- R5 should be configured with 10.1.1.100 as its default gateway
- Configure the interval between successive advertisements by the VR Master router for to 1/4th of a second.

Group 2:

- R3 should be the VR Master, R2 should be the first VR Backup and R4 should be configured as the second VR Backup.
- This group should use 100.1.1.200 as its VIP.
- R6 should be configured with 10.1.1.200 as its default gateway

- Configure the interval between successive advertisements by the VR Master router for Group 2 to 1/2 of a second. The VR Backup routers should learn the advertisement interval set by the VR Master.

To configure VRRP for the stated policy, ONLY the policy for Group 2 needs to be configured, since the first policy is already configured:

On R2:

```
R2(config)#Int F0/0
R2(config-if)#Vrrp 2 ip 10.1.1.200
R2(config-if)#Vrrp 2 priority 200
R2(config-if)#Vrrp 2 timers learn
```

You should see the following console message:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 2 state Init -> Backup
%VRRP-6-STATECHANGE: Fa0/0 Grp 2 state Backup -> Master
```

To see the configuration on the F0/0 interface of R2:

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

```
interface FastEthernet0/0
  mac-address 0000.2222.2222
  ip address 10.1.1.2 255.255.255.0
  duplex auto
  speed auto
  vrrp 1 ip 10.1.1.100
  vrrp 1 priority 254
  vrrp 1 track 21 decrement 105
  vrrp 2 ip 10.1.1.200
  vrrp 2 timers learn
  vrrp 2 priority 200
end
```

On R3:

```
R3(config)#Int F0/0
R3(config-if)#Vrrp 2 ip 10.1.1.200
R3(config-if)#Vrrp 2 priority 250
R3(config-if)#Vrrp 2 timers advertise 5
```

You should see the following console message:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 2 state Init -> Backup
%VRRP-6-STATECHANGE: Fa0/0 Grp 2 state Backup -> Master
```

NOTE: If the console message of R2 is checked, you should see that R2 transitioned into the Backup state:

On R2:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 2 state Master -> Backup
```

On R4:

```
R4(config)#Int F0/0
R4(config-if)#Vrrp 2 ip 10.1.1.200
R4(config-if)#Vrrp 2 priority 150
R4(config-if)#Vrrp 2 timers learn
```

You should see the following console message:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 2 state Init -> Backup
%VRRP-6-STATECHANGE: Fa0/0 Grp 2 state Backup -> Master
%VRRP-6-STATECHANGE: Fa0/0 Grp 2 state Master -> Backup
```

To verify the configuration:

On R2:

R2#Show vrrp brief

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	254	3007		Y	Master	10.1.1.2	10.1.1.100
Fa0/0	2	200	3218		Y	Backup	10.1.1.3	10.1.1.200

R2#Show vrrp

FastEthernet0/0 - Group 1

State is Master

Virtual IP address is 10.1.1.100

Virtual MAC address is 0000.5e00.0101

Advertisement interval is 1.000 sec

Preemption enabled

Priority is 254

Track object 21 state Up decrement 105

Master Router is 10.1.1.2 (local), priority is 254

Master Advertisement interval is 1.000 sec

Master Down interval is 3.007 sec

FastEthernet0/0 - Group 2

State is Backup

Virtual IP address is 10.1.1.200

Virtual MAC address is 0000.5e00.0102

Advertisement interval is 1.000 sec

Preemption enabled

Priority is 200

Master Router is 10.1.1.3, priority is 250

Master Advertisement interval is 5.000 sec

Master Down interval is 15.218 sec (expires in 12.818 sec) Learning

On R3:

R3#Show vrrp brief

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	200	3218		Y	Backup	10.1.1.2	10.1.1.100
Fa0/0	2	250	3023		Y	Master	10.1.1.3	10.1.1.200

On R4:

R4#Show vrrp brief

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	150	3414		Y	Backup	10.1.1.2	10.1.1.100
Fa0/0	2	150	3414		Y	Backup	10.1.1.3	10.1.1.200

Let's configure a default gateway on R6 and point it to 10.1.1.200 :

On R6:

R6(config)#No ip route 0.0.0.0 0.0.0.0

R6(config)#IP route 0.0.0.0 0.0.0.0 10.1.1.200

To test the configuration:

On R6:

R6#Traceroute 1.1.1.1 Numeric

Type escape sequence to abort.

Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

1 10.1.1.3 0 msec 0 msec 0 msec

2 13.1.1.1 16 msec * 12 msec

R6#**Show arp**

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.2	50	0000.5e00.0101	ARPA	FastEthernet0/0
Internet	10.1.1.6	-	0017.5aad.52aa	ARPA	FastEthernet0/0
Internet	10.1.1.100	27	0000.5e00.0101	ARPA	FastEthernet0/0
Internet	10.1.1.200	0	0000.5e00.0102	ARPA	FastEthernet0/0

On R5:

R5#**Traceroute 1.1.1.1**

Type escape sequence to abort.

Tracing the route to 1.1.1.1

VRF info: (vrf in name/id, vrf out name/id)

```
1 10.1.1.2 0 msec 4 msec 0 msec
2 12.1.1.1 20 msec * 12 msec
```

R5#**Show arp**

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.3	0	0000.3333.3333	ARPA	FastEthernet0/0
Internet	10.1.1.5	-	001b.d4be.69d0	ARPA	FastEthernet0/0
Internet	10.1.1.100	29	0000.5e00.0101	ARPA	FastEthernet0/0

Task 7

Configure VRRP group 1 to be plain text authenticated using "Cisco" as the password.

In VRRP there are two types of authentications, clear text and MD5. When configuring MD5 authentication, there are two choices, using the "Key-chain" or using the "Key-string". The key for the MD5 hash can either be given directly in the configuration using the "Key string" or supplied indirectly through a "key-chain".

On R2:

R2(config)#**Int F0/0**

R2(config-if)#**Vrrp 1 authentication Cisco**

To verify the configuration:

On R2:

```
R2#Show vrrp int f0/0 group 1
```

```
FastEthernet0/0 - Group 1
  State is Master
  Virtual IP address is 10.1.1.100
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 254
  Track object 21 state Up decrement 105
  Authentication text, string "Cisco"
  Master Router is 10.1.1.2 (local), priority is 254
  Master Advertisement interval is 1.000 sec
  Master Down interval is 3.007 sec
```

You should see the following console messages. Let's see what happened to R3 and R4:

```
%VRRP-4-BADAUTHTYPE: Bad authentication from 10.1.1.3, group 1, type 0,
expected 1
```

On R3:

You should see the following console message:

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Backup -> Master
```

```
R3#Show vrrp brief
```

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	200	3218		Y	Master	10.1.1.3	10.1.1.100
Fa0/0	2	250	15023		Y	Master	10.1.1.3	10.1.1.200

On R4:

```
R4#Show vrrp brief
```

Interface	Grp	Pri	Time	Own	Pre	State	Master addr	Group addr
Fa0/0	1	150	3414		Y	Backup	10.1.1.3	10.1.1.100
Fa0/0	2	150	15414		Y	Backup	10.1.1.3	10.1.1.200

It looks like R3 and R4 are in a different world altogether. Once the authentication was enabled, R2 stopped communicating with R3 and R4. So which router will R5 use as its default gateway?

Let's clear the ARP table and try again:

On R5:

```
R5(config)#Int F0/0
R5(config-if)#Shut
R5(config-if)#No shut
```

```
R5#Traceroute 1.1.1.1
```

```
Type escape sequence to abort.
Tracing the route to 1.1.1.1
VRF info: (vrf in name/id, vrf out name/id)
  1 10.1.1.2 4 msec 0 msec 4 msec
  2 12.1.1.1 16 msec *
  13.1.1.1 12 msec
```

```
R5#Show arp
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.5	-	001b.d4be.69d0	ARPA	FastEthernet0/0
Internet	10.1.1.100	0	0000.5e00.0101	ARPA	FastEthernet0/0

Very unpredictable result. Let's continue with the configuration of authentication on R3 and R4:

On R3 and R4:

```
Rx(config)#Int F0/0
Rx(config-if)#Vrrp 1 authentication Cisco
```

You should see the following console message, once the authentication command is entered on these routers, as predicted.

```
%VRRP-6-STATECHANGE: Fa0/0 Grp 1 state Master -> Backup
```

Task 8

Configure MD5 authentication using "VRRP" as the password for Group 2. You should use a key chain called "tst" to accomplish this task.

On R2, R3 and R4:

```
Rx(config)#Key chain tst
```



```
Rx(config-keychain)#Key 1
Rx(config-keychain-key)#Key-string VRRP

Rx(config)#Int F0/0
Rx(config-if)#Vrrp 2 authentication md5 key-chain tst
```

To verify the configuration:

On R2:

```
R2#Show vrrp | s Group 2
```

```
FastEthernet0/0 - Group 2
  State is Backup
  Virtual IP address is 10.1.1.200
  Virtual MAC address is 0000.5e00.0102
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 200
  Authentication MD5, key-chain "tst"
  Master Router is 10.1.1.3, priority is 250
  Master Advertisement interval is 5.000 sec
  Master Down interval is 15.218 sec (expires in 12.190 sec) Learning
```

On R3:

```
R3#Show vrrp | s Group 2
```

```
FastEthernet0/0 - Group 2
  State is Master
  Virtual IP address is 10.1.1.200
  Virtual MAC address is 0000.5e00.0102
  Advertisement interval is 5.000 sec
  Preemption enabled
  Priority is 250
  Authentication MD5, key-chain "tst"
  Master Router is 10.1.1.3 (local), priority is 250

  Master Advertisement interval is 5.000 sec
  Master Down interval is 15.023 sec
```

On R4:

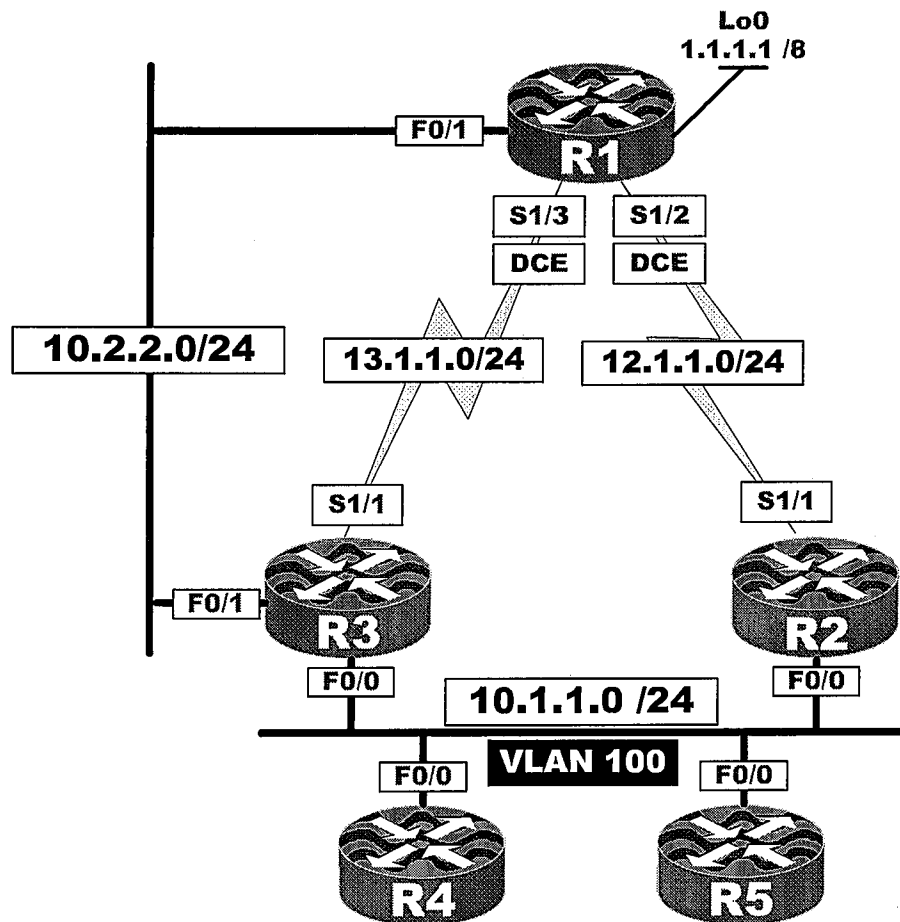
```
R4#Show vrrp | s Group 2
```

```
FastEthernet0/0 - Group 2
  State is Backup
  Virtual IP address is 10.1.1.200
  Virtual MAC address is 0000.5e00.0102
  Advertisement interval is 1.000 sec
  Preemption enabled
  Priority is 150
  Authentication MD5, key-chain "tst"
  Master Router is 10.1.1.3, priority is 250
  Master Advertisement interval is 5.000 sec
  Master Down interval is 15.414 sec (expires in 13.862 sec) Learning
```

Task 9

Erase the startup config and reload the routers before proceeding to the next lab.

Lab 3 – GLBP Configuration



Task 1

Configure the Routers and the Switch based on the above diagram.

On SW1:

```
SW1(config)#Int range f0/2-5
```

```
SW1(config-if-range)#Swi mode acc
SW1(config-if-range)#Swi acc vlan 100
SW1(config-if-range)#No shu
```

On SW2:

```
SW2(config)#Int range f0/1,f0/3
SW2(config-if-range)#Swi mode acc
SW2(config-if-range)#SWi acc vlan 200
SW2(config-if-range)#No shut
```

On R1:

```
R1(config)#int s1/2
R1(config-if)#clock rate 64000
R1(config-if)#ip addr 12.1.1.1 255.255.255.0
R1(config-if)#No shut
```

```
R1(config)#int s1/3
R1(config-if)#clock rate 64000
R1(config-if)#ip addr 13.1.1.1 255.255.255.0
R1(config-if)#No shut
```

```
R1(config)#Int F0/1
R1(config-if)#IP address 10.2.2.1 255.255.255.0
R1(config-if)#No shut
```

```
R1(config)#Int Lo0
R1(config-if)#IP address 1.1.1.1 255.0.0.0
```

On R2:

```
R2(config)#int s1/1
R2(config-if)#ip addr 12.1.1.2 255.255.255.0
R2(config-if)#No shut
```

```
R2(config)#Int F0/0
R2(config-if)#IP address 10.1.1.2 255.255.255.0
R2(config-if)#No shut
```

On R3:

```
R3(config)#int s1/1
R3(config-if)#ip addr 13.1.1.3 255.255.255.0
R3(config-if)#No shut
```

```
R3(config)#Int F0/0
R3(config-if)#IP address 10.1.1.3 255.255.255.0
R3(config-if)#No shut
```

```
R3(config)#Int F0/1
R3(config-if)#IP address 10.2.2.3 255.255.255.0
R3(config-if)#No shut
```

On R4:

```
R4(config)#Int F0/0
R4(config-if)#IP address 10.1.1.4 255.255.255.0
R4(config-if)#No shut
```

On R5:

```
R5(config)#Int F0/0
R5(config-if)#IP address 10.1.1.5 255.255.255.0
R5(config-if)#No shut
```

To verify the configuration:

On R1:

```
R1#Ping 10.2.2.3
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
```

```
R1#Ping 12.1.1.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 12.1.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
```

```
R1#Ping 13.1.1.3
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 13.1.1.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/56 ms
```

On R2:

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 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

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 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

R2#Ping 10.1.1.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.5, timeout is 2 seconds:

!!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

Task 2

Configure GLBP on R2 and R3 based on the following policy:

1. R2 should be configured to be the AVG
2. The GLBP group number should be set to 1.
3. Devices on VLAN 100 should use 10.1.1.100 IP address as their default gateway

- GLBP provides gateway redundancy by using a single IP address and multiple virtual MAC addresses.
- In GLBP there can be upto 4 routers/gateways in a group.
- Group members will communicate with each other using hello messages, these messages will be destined to 224.0.0.102 Multicast address and to UDP port 3222 and they will be sent every 3 seconds by default.
- Initially group members will elect one active virtual gateway (AVG); the other router/s will be the backup VG in case the active AVG fails.
- The AVG will assign virtual mac addresses to the other routers, known to be active virtual

forwarders (AVF).

- Each AVF assumes responsibility for forwarding packets sent to the virtual MAC address assigned to it by the AVG.
- The AVG is responsible for answering to ARP requests for the virtual IP address.

On R2:

```
R2 (config) #Int F0/0
```

The following commands enable GLBP on an interface and identify the primary IP address of the virtual gateway and sets the local router's priority to 101, the default priority is set to 100. If you are planning to customize GLBP, this command should be configured last. It is possible that if you first enable a GLBP group before customizing GLBP, the router could take over control of the group and become the AVG before you have finished customizing the feature. Therefore, if you plan to customize GLBP, it is a good idea to do so before enabling GLBP.

```
R2 (config-if) #Glbp 1 priority 101
R2 (config-if) #Glbp 1 ip 10.1.1.100
```

On R3:

```
R3 (config) #Int F0/0
R3 (config-if) #Glbp 1 ip 10.1.1.100
```

To verify the configuration:

On R2:

```
R2#Show glbp brief
```

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	101	Active	10.1.1.100	local	10.1.1.3
Fa0/0	1	1	-	Active	0007.b400.0101	local	-
Fa0/0	1	2	-	Listen	0007.b400.0102	10.1.1.3	-

The output of the above show command reveals that the local router is the AVG, and the standby router is 10.1.1.3 (R3).

There are two Active Virtual Forwarders (AVFs) the first AVF is the local router and it is responsible for the 0007.b400.0101 MAC address and the second AVF is responsible for forwarding traffic for 0007.b400.0102 MAC address with an IP address 10.1.1.3 (This is R3).

On R3:

R3#Show glbp brief

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	100	Standby	10.1.1.100	10.1.1.2	local
Fa0/0	1	1	-	Listen	0007.b400.0101	10.1.1.2	-
Fa0/0	1	2	-	Active	0007.b400.0102	local	-

The output of the above show command reveals the following:

- The local router is in Standby state and the active router is 10.1.1.2 (R2).
- The local router is responsible for forwarding traffic for 0007.b400.0102 MAC address and it is backing up 0007.b400.0101 MAC address.

To test the configuration:

On R2:

R2(config)#Int F0/0

R2(config-if)#Shut

You should see the following console messages:

%GLBP-6-FWDSTATECHANGE: FastEthernet0/0 Grp 1 Fwd 1 state Active -> Init

%GLBP-6-STATECHANGE: FastEthernet0/0 Grp 1 state Active -> Init

To verify the configuration:

On R2:

R2#Show glbp brief

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	101	Init	10.1.1.100	unknown	unknown
Fa0/0	1	1	-	Init	0007.b400.0101	unknown	-
Fa0/0	1	2	-	Init	0007.b400.0102	unknown	-

On R3:

Once you console into R3, you should see the following console message:

%GLBP-6-FWDSTATECHANGE: FastEthernet0/0 Grp 1 Fwd 1 state Listen -> Active

%GLBP-6-STATECHANGE: FastEthernet0/0 Grp 1 state Standby -> Active

R3#Show glbp brief

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	100	Active	10.1.1.100	local	unknown
Fa0/0	1	1	-	Active	0007.b400.0101	local	-
Fa0/0	1	2	-	Active	0007.b400.0102	local	-

We can see that R3 is the active router and the AVF for both V-MAC addresses.

Let's "No Shut" the F0/0 interface of R2 and verify the result.

On R2:

```
R2(config)#Int F0/0
R2(config-if)#No shut
```

We should see the following console message:

```
%GLBP-6-FWDSTATECHANGE: FastEthernet0/0 Grp 1 Fwd 1 state Listen ->
Active
```

To verify the test:

On R2:

```
R2#Show glbp brief
```

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	101	Standby	10.1.1.100	10.1.1.3	local
Fa0/0	1	1	-	Active	0007.b400.0101	local	-
Fa0/0	1	2	-	Listen	0007.b400.0102	10.1.1.3	-

Why is R2 NOT the active router?

Let's configure preemption and see what happens:

On R2:

```
R2(config)#Int F0/0
R2(config-if)#Glb 1 preempt
```

You should see the following console message:

```
%GLBP-6-STATECHANGE: FastEthernet0/0 Grp 1 state Standby -> Active
```

```
R2#Show glbp brief
```

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router

Fa0/0	1	-	101	Active	10.1.1.100	local	10.1.1.3
Fa0/0	1	1	-	Active	0007.b400.0101	local	-
Fa0/0	1	2	-	Listen	0007.b400.0102	10.1.1.3	-

So by default the preemption is disabled and it needs to be enabled if a particular router must be the AVG.

On R3:

R3#Show glbp brief

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	100	Standby	10.1.1.100	10.1.1.2	local
Fa0/0	1	1	-	Listen	0007.b400.0101	10.1.1.2	-
Fa0/0	1	2	-	Active	0007.b400.0102	local	-

Task 3

Configure the Hello interval to be 250 msec and the hold time to be 750 msec.

On R2 and R3:

```
Rx(config)#Int F0/0
Rx(config-if)#Glb 1 timers msec 250 msec 750
```

The above command sets the hello intervals to 250 msec and the dead interval to 750 msec.

To verify the configuration:

On R2:

R2#Show glbp detail | Inc Hello

```
Hello time 250 msec, hold time 750 msec
```

Task 4

Configure RIPv2 on R1, R2 and R3; these routers should advertise all their directly connected interfaces in this routing domain. R4 and R5 should have a static default route pointing to the VIP address.

On R1, R2 and R3:

```
Rx(config)#Router rip
Rx(config-router)#No au
Rx(config-router)#ver 2
Rx(config-router)#Network 10.0.0.0
```

On R1:

```
R1(config)#Router rip
R1(config-router)#Network 1.0.0.0
R1(config-router)#Netw 12.0.0.0
R1(config-router)#Netw 13.0.0.0
```

On R2:

```
R2(config)#Router rip
R2(config-router)#Netw 12.0.0.0
```

On R3:

```
R3(config)#Router rip
R3(config-router)#Netw 13.0.0.0
```

On R4 and R5:

```
Rx(config)#IP route 0.0.0.0 0.0.0.0 10.1.1.100
```

To verify the configuration:

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 = 28/28/28 ms

On R5:

```
R5#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 = 16/16/16 ms

On R2:

```
R2#Show ip route rip | B Gate
```

Gateway of last resort is not set

```
R      1.0.0.0/8 [120/1] via 12.1.1.1, 00:00:10, Serial1/1
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
R      10.2.2.0/24 [120/1] via 12.1.1.1, 00:00:10, Serial1/1
              [120/1] via 10.1.1.3, 00:00:19, FastEthernet0/0
      13.0.0.0/24 is subnetted, 1 subnets
R      13.1.1.0 [120/1] via 12.1.1.1, 00:00:10, Serial1/1
              [120/1] via 10.1.1.3, 00:00:19, FastEthernet0/0
```

Task 5

Configure and test load balancing in a round robin fashion.

On R2:

```
R2#Show glbp det | Inc Load
```

Load balancing: round-robin

We can see that the default load balancing is performed in a round robin fashion. Load balancing can be set to host-dependent, round robin or weighted.

HOST DEPENDANT

- The MAC address of a host is used to determine which AVF's MAC address the host is directed towards.
- A given host is guaranteed to use the same virtual MAC address as long as the number of VFs in the GLBP group is constant.
- Host dependent load balancing will need to be used when using SNAT. Because it requires each

host to be returned the same virtual MAC address each time it sends an ARP request for the virtual IP address.

- Host dependent load balancing is not recommended for situations where there are a small number of end hosts, for example less than 20. Unless specified in a CCIE lab.
- The "GLBP 1 load-balancing host-dependent" command can be used to set the load balancing to host-dependent.

WEIGHTED

- In this algorithm, GLBP places a weight on each device when calculating the amount of load sharing that will occur through MAC assignment.
- Each GLBP router in a given group will advertise its weighting and assignment; the AVG will act based on that value.
- For example, if there are two routers in a group and router A has double the forwarding capacity of router B, the weighting value of router A should be configured to be double the amount of router B.
- The "GLBP 1 weighting" and "GLBP 1 load-balancing weighted" command can be used to set the load balancing to weighted.

ROUND ROBIN

- With round robin, the VF MAC addresses are used sequentially in ARP replies for the virtual IP address.
- Round robin load balancing is suitable for any number of end hosts.
- This is the default
- The "GLBP 1 load-balancing round robin" command can be used to set the load balancing to round robin.

To test the configuration:

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 = 28/28/32 ms

```
R4#Show arp 10.1.1.100
```

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.100	10	0007.b400.0102	ARPA	FastEthernet0/0

On R5:

```
R5#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 = 12/14/16 ms

R5#Show arp 10.1.1.100

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	10.1.1.100	11	0007.b400.0101	ARPA	FastEthernet0/0

To verify the configuration:

On R2:

R2#Show glbp det | s Forwarder 1

Forwarder 1

State is Active

3 state changes, last state change 00:42:37

MAC address is 0007.b400.0101 (default)

Owner ID is 001c.58b9.f778

Redirection enabled

Preemption enabled, min delay 30 sec

Active is local, weighting 100

Client selection count: 1

We can see that forwarder 1 is responsible for forwarding traffic for the 0007.b400.0101 Mac address and it is owned by 001c.58b9.f778. Remember that the owner id MAC address may be different in your rack of routers. Let's see which router's MAC address is 001c.58b9.f778?

R2#Show int f0/0 | Inc bia

Hardware is MV96340 Ethernet, address is 001c.58b9.f778 (bia 001c.58b9.f778)

Sure enough it's R2's MAC address.

R2#Show glbp det | s Forwarder 2

Forwarder 2

State is Listen

MAC address is 0007.b400.0102 (learnt)

Owner ID is 001c.58f6.f660

Redirection enabled, 599.844 sec remaining (maximum 600 sec)

Time to live: 14399.844 sec (maximum 14400 sec)

Preemption enabled, min delay 30 sec

Active is 10.1.1.3 (primary), weighting 100 (expires in 0.594 sec)
Client selection count: 1

On R3:

```
R3#Show int f0/0 | Inc bia
```

```
Hardware is MV96340 Ethernet, address is 001c.58f6.f660 (bia 001c.58f6.f660)
```

Task 6

Configure clear text authentication with a password of "Cisco" for GLBP group 1.

On R2:

```
R2(config)#Int F0/0  
R2(config-if)#Glbp 1 authentication text Cisco
```

Once we enter the above command we should see the following console messages:

```
%GLBP-4-BADAUTH: Bad authentication received from 10.1.1.3, group 1  
%GLBP-6-FWDSTATECHANGE: FastEthernet0/0 Grp 1 Fwd 2 state Listen -> Active  
%GLBP-4-BADAUTH: Bad authentication received from 10.1.1.3, group 1
```

The above console messages are expected because R3 does NOT have authentication configured yet. Let's configure authentication on R3.

To verify the configuration:

On R2:

```
R2#Show glbp det | Inc Auth
```

```
Authentication text "Cisco"
```

On R3:

You should also see the following bad authentication console messages on R3 as well:

```
%GLBP-4-BADAUTH: Bad authentication received from 10.1.1.2, group 1
```

```
R3(config)#Int F0/0
R3(config-if)#Glbp 1 authentication text Cisco
```

Now you should see the following console messages:

```
%GLBP-6-STATECHANGE: FastEthernet0/0 Grp 1 state Active -> Speak
%GLBP-6-FWDSTATECHANGE: FastEthernet0/0 Grp 1 Fwd 1 state Active -> Listen
```

To verify the configuration:

On R3:

```
R3#Show glbp brief
```

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	100	Standby	10.1.1.100	10.1.1.2	local
Fa0/0	1	1	-	Listen	0007.b400.0101	10.1.1.2	-
Fa0/0	1	2	-	Active	0007.b400.0102	local	-

Task 7

Re-configure the password using the strongest authentication mechanism.

On R2 and R3:

```
Rx(config)#Key chain TST
Rx(config-keychain)#Key 1
Rx(config-keychain-key)#Key-string Cisco

Rx(config)#Int F0/0
R2(config-if)#No glbp 1 authentication
Rx(config-if)#Glbp 1 authentication md5 key-chain TST
```

To verify the configuration:

On R2:

```
R2#Show glbp detail | Inc Auth
```

```
Authentication MD5, key-chain "TST"
```


Task 8

Configure the routers in this GLBP group such that if an AVF fails, the virtual mac address that the AVF was responsible for is still given out in ARP-replies for another 600 seconds. The virtual mac address of the AVF that failed should be inactive after 4 hours.

On R2 and R3:

```
Rx(config)#int f0/0
Rx(config-if)#Glbp 1 timers redirect 600 14400
```

A virtual forwarder that is assigned a virtual mac address by the AVG is known as the primary VF.

If the VF has learned the virtual mac address from hello messages, it is referred to as the secondary VF.

The redirect timer (the first value configured "600") sets the time delay between a forwarder failing and the AVG assuming that the forwarder will not return. The virtual mac address that the forwarder was responsible for is still given out in ARP-replies, but the forwarding task is still being handled by another router in the group.

The second value is called the timeout interval. This interval is the time delay between a forwarder failing and the mac address for which the forwarder was responsible becoming inactive on all routers in the GLBP group. After the timeout interval, packets sent to this virtual mac address will be lost.

This value must be long enough to allow all hosts to refresh their ARP cache entry that contained the virtual mac address, typically 4 hours.

To verify the configuration:

On R2:

```
R2#Show glbp detail | Inc Redirect time
```

```
Redirect time 600 sec, forwarder timeout 14400 sec
```

Task 9

Configure interface tracking such that if the F0/1 **and** S1/1 interfaces of R3 go down, R2 transitions into AVG and R3 is **not** used as the AVG, **but if these two interfaces** come back up, once again R3 resumes the role of AVG.

On R3:

```
R3(config)#track 1 interface f0/1 line-protocol
R3(config)#track 2 interface s1/1 line-protocol
```

The above two lines track F0/1 and serial S1/1 interface of R3

```
R3(config)#int f0/0
R3(config-if)#glbp 1 weighting track 1
R3(config-if)#glbp 1 weighting track 2
R3(config-if)#glbp 1 priority 119
R3(config-if)#glbp 1 preem
```

The above two statements tie the tracked interfaces to GLBP such that if the line-protocol of any of the tracked interfaces is in down state, the weight will be decremented by 10 (The default value). If the line protocols of both interfaces are in down state, the weight will be decremented by 20. The decrement value can be individually configured using the "GLBP 1 weighting track [1 or 2] decrement" command.

```
R3(config-if)#glbp 1 weighting 110 lower 95 upper 105
```

The above command sets the maximum weight to 110, if the line protocols of BOTH interfaces are in down state, this value will be decremented by 20 which will be lower than the lower threshold and the router will no longer be an active forwarder. To resume its role as an active forwarder, the router must have both tracked interfaces back up. This will set the weight value higher than the upper threshold.

To verify the configuration:

On R3:

```
R3#Show glbp br
```

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	119	Active	10.1.1.100	local	10.1.1.2
Fa0/0	1	1	-	Listen	0007.b400.0101	10.1.1.2	-
Fa0/0	1	2	-	Active	0007.b400.0102	local	-

On R2:

R2#Show glbp brief

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	101	Standby	10.1.1.100	10.1.1.3	local
Fa0/0	1	1	-	Active	0007.b400.0101	local	-
Fa0/0	1	2	-	Listen	0007.b400.0102	10.1.1.3	-

To test the configuration:

On R3:

R3(config)#Int f0/1

R3(config-if)#shu

You should see the following console messages:

%TRACKING-5-STATE: 2 interface Fa0/1 line-protocol Up->Down

%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down

R3(config)#Int S1/1

R3(config-subif)#Shut

You should see the following console messages within 10 seconds:

%TRACKING-5-STATE: 1 interface Se0/0.21 line-protocol Up->Down

%GLBP-6-FWDSTATECHANGE: FastEthernet0/0 Grp 1 Fwd 1 state Active -> Listen

To verify the configuration:

On R2:

You should see the following console message:

%GLBP-6-FWDSTATECHANGE: FastEthernet0/0 Grp 1 Fwd 2 state Listen -> Active

R2#Show glbp brief

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	101	Standby	10.1.1.100	10.1.1.3	local
Fa0/0	1	1	-	Active	0007.b400.0101	local	-
Fa0/0	1	2	-	Active	0007.b400.0102	local	-

We can see that R2 takes responsibility for forwarding packets for both V-MAC addresses.

To test the configuration:

On R3:

```
R3(config)#Int S1/1
R3(config-subif)#No shut
```

```
R3(config)#int f0/1
R3(config-if)#No shu
```

You should see the following console messages:

```
%TRACKING-5-STATE: 2 interface Se0/0 line-protocol Down->Up
%LINK-3-UPDOWN: Interface Serial0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0, changed state
to up
%TRACKING-5-STATE: 1 interface Fa0/1 line-protocol Down->Up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up
%GLBP-6-FWDSTATECHANGE: FastEthernet0/0 Grp 1 Fwd 2 state Listen -> Active
```

To verify the configuration:

On R3

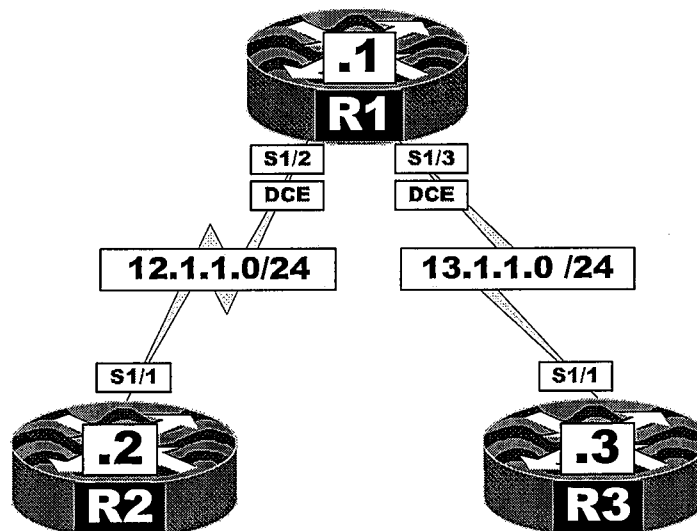
```
R3#Show glbp brief
```

Interface	Grp	Fwd	Pri	State	Address	Active router	Standby router
Fa0/0	1	-	119	Active	10.1.1.100	local	10.1.1.2
Fa0/0	1	1	-	Listen	0007.b400.0101	10.1.1.2	-
Fa0/0	1	2	-	Active	0007.b400.0102	local	-

Task 10

Erase the startup configuration of the Routers and the Switches and reload them before proceeding to the next lab.

Lab 4 – Configuring NTP



Task 1

Configure the Routers based on the above diagram.

On R1:

```
R1(config)#int s1/2
R1(config-if)#clock rate 64000
R1(config-if)#ip addr 12.1.1.1 255.255.255.0
R1(config-if)#No shut
```

```
R1(config)#int s1/3
R1(config-if)#clock rate 64000
R1(config-if)#ip addr 13.1.1.1 255.255.255.0
R1(config-if)#No shut
```

On R3:

```
R3(config)#int s1/1
```

```
R3(config-if)#ip addr 13.1.1.3 255.255.255.0
R3(config-if)#No shut
```

On R2:

```
R2(config)#int s1/1
R2(config-if)#ip addr 12.1.1.2 255.255.255.0
R2(config-if)#No shut
```

To verify the configuration

On R1

```
R1#Ping 12.1.1.2
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 12.1.1.2, timeout is 2 seconds:

!!!!!!

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

```
R1#Ping 13.1.1.3
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 13.1.1.3, timeout is 2 seconds:

!!!!!!

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

Task 2

Ensure that R1's time and date is set on May 17th 2014.

On R1:

```
R1#Clock set 11:00:00 17 May 2014
```

You should see the following console message:

```
%SYS-6-CLOCKUPDATE: System clock has been updated from 02:42:25 UTC Tue
May 20 2014 to 11:00:00 UTC Sat May 17 2014, configured from console by
console.
```

To verify the configuration:

On R1:

```
R1#Show clock
```

```
11:01:36.575 UTC Sat May 17 2014
```

Task 3

Configure R1 to be the NTP server using a stratum of 2.

On R1:

```
R1(config)#NTP master 2
```

The above command is used to configure the router as an NTP clock master. This command should be used when an external clock source is not available.

To verify the configuration:

On R1:

```
R1#show ntp status
```

```
Clock is synchronized, stratum 2, reference is 127.127.1.1  
nominal freq is 250.0000 Hz, actual freq is 249.9968 Hz, precision is  
2**24  
reference time is D721C067.3DA369A9 (11:03:03.240 UTC Sat May 17 2014)  
clock offset is 0.0000 msec, root delay is 0.00 msec  
root dispersion is 7937.51 msec, peer dispersion is 7937.50 msec  
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000012578 s/s  
system poll interval is 16, last update was 1 sec ago.
```

Task 4

R2 and R3 should be configured to synchronize their time and date to R1's time and date.

Let's see what R2's clock is set to:

On R2:

R2#**Show clock**

***02:57:05.415 UTC Tue May 20 2014**

The following configures R2 to use R1 as their NTP server:

R2(config)#**Ntp server 12.1.1.1**

To configure R3 to use R1 as its NTP server:

On R3:

R3(config)#**Ntp server 13.1.1.1**

To verify the configuration:

On R2:

R2#**Show ntp status**

Clock is synchronized, stratum 3, reference is 12.1.1.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24
reference time is D721C149.21FB8ECA (11:06:49.132 UTC Sat May 17 2014)
clock offset is 0.7582 msec, root delay is 24.29 msec
root dispersion is 7938.65 msec, peer dispersion is 7937.50 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000001 s/s
system poll interval is 64, last update was 1 sec ago.

R2#**Show clock**

11:07:50.408 UTC Sat May 17 2014

Note: The NTP master started with a stratum of 2 and every downstream router that gets the clock from the master adds one to the stratum, therefore R2 will have a stratum of $2 + 1 = 3$

On R3:

R3#**show ntp status**

Clock is synchronized, stratum 3, reference is 13.1.1.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24

reference time is D721C175.532FF29E (11:07:33.324 UTC Sat May 17 2014)
clock offset is 1.3237 msec, root delay is 24.18 msec
root dispersion is 7939.81 msec, peer dispersion is 7937.50 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000004 s/s
system poll interval is 64, last update was 45 sec ago.

R3#Show clock

11:08:59.660 UTC Sat May 17 2014

Task 5

Ensure that these routers authenticate using "Cisco" as their password.

Cisco's NTP implementation includes a powerful authentication capability that ensures that NTP updates are being sent from a trusted source.

On R1:

This command defines an authentication key for use with NTP.

```
R1 (config) #Ntp authentication-key 1 md5 Cisco
R1 (config) #Ntp authenticate
R1 (config) #Ntp trusted-key 1
```

On R2:

This command defines an authentication key for use with NTP.

```
R2 (config) #Ntp authentication-key 1 md5 Cisco
```

The following command enables NTP authentication.

```
R2 (config) #Ntp authenticate
```

This command is used to authenticate the router to a specific authentication key.

```
R2 (config) #ntp trusted-key 1
```

The following command defines the key number to use with the specified server

```
R2(config)#ntp server 12.1.1.1 key 1
```

On R3:

```
R3(config)#Ntp authentication-key 1 md5 Cisco  
R3(config)#Ntp authenticate  
R3(config)#Ntp trusted-key 1  
R3(config)#Ntp server 13.1.1.1 key 1
```

To verify the configuration:

On R2:

```
R2#Sh ntp status
```

```
Clock is synchronized, stratum 3, reference is 12.1.1.1  
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24  
reference time is D721C88E.437E2D57 (11:37:50.263 UTC Sat May 17 2014)  
clock offset is 0.4630 msec, root delay is 29.65 msec  
root dispersion is 7938.44 msec, peer dispersion is 7937.50 msec  
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000000 s/s  
system poll interval is 64, last update was 11 sec ago.
```

On R3:

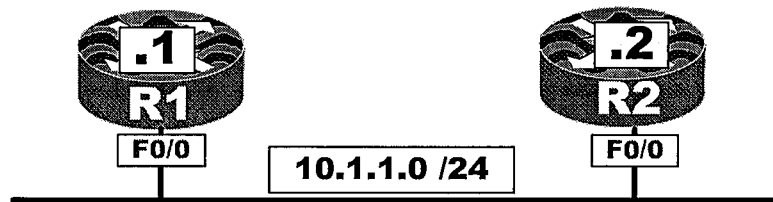
```
R3#Sh ntp status
```

```
Clock is synchronized, stratum 3, reference is 13.1.1.1  
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24  
reference time is D721C8A9.9CBC8CA1 (11:38:17.612 UTC Sat May 17 2014)  
clock offset is 4.9467 msec, root delay is 29.63 msec  
root dispersion is 7943.00 msec, peer dispersion is 7937.50 msec  
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is 0.000000013 s/s  
system poll interval is 64, last update was 21 sec ago.
```

NOTE: It may take a long time for these routers to synchronize their time.

Task 6

Configure the routers based on the following topology and IP addressing scheme:



On SW1:

```
SW1(config)#Int range f0/1-2
SW1(config-if-range)#Swi mode acc
SW1(config-if-range)#Swi acc vlan 12
SW1(config-if-range)#No shut
```

On R1:

```
R1(config)#Int F0/0
R1(config-if)#IP address 10.1.1.1 255.255.255.0
R1(config-if)#No shut
```

On R2:

```
R2(config)#Int F0/0
R2(config-if)#IP address 10.1.1.2 255.255.255.0
R2(config-if)#No shut
```

To verify the configuration:

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

Task 7

Ensure that R1's time and date is set to May 17th 2014.

On R1:

```
R1#Clock set 12:00:00 17 may 2014
```

To verify the configuration:

On R1:

```
R1#Show clock
```

```
12:00:16.347 UTC Sat May 17 2014
```

Task 8

R1 should be configured as the NTP server; this router should broadcast updates on its F0/0.

On R1:

```
R1(config)#Int F0/0  
R1(config-if)#Ntp broadcast
```

The above command specifies that the interface should send NTP broadcast packets and the following command configures the local router to be the NTP master.

```
R1(config)#Ntp master
```

To verify the configuration:

On R1:

```
R1#Show ntp status | Inc stratum
```

```
Clock is synchronized, stratum 8, reference is 127.127.7.1
```

We can see that if the stratum is NOT configured on the NTP Master, the stratum is auto-set to 8.

Task 9

Ensure that R2 synchronizes its time to R1.

On R2:

```
R2(config)#int f0/0  
R2(config-if)#ntp broadcast client
```

This command allows the router to receive NTP broadcast packets through this interface.

To verify the configuration:

```
R2#Show ntp status | Inc stratum
```

```
Clock is unsynchronized, stratum 16, no reference clock
```

```
R2#Show ntp status | Inc stratum
```

```
Clock is synchronized, stratum 9, reference is 10.1.1.1
```

Task 10

Erase the startup config and reload the routers before proceeding to the next lab.

Lab 6 – EEM

Task 1

Configure an applet named TST on R1 that assigns the F0/0 interface an IP address of 10.1.1.1 /24 and enables the interface.

On R1:

```
R1 (config) #Event manager applet TST
R1 (config-applet) #event none sync yes
R1 (config-applet) #action 1.0 cli command "enable"
R1 (config-applet) #action 1.1 cli command "config t"
R1 (config-applet) #action 1.2 cli command "interface f0/0"
R1 (config-applet) #action 1.3 cli command "ip address 10.1.1.1 255.255.255.0"
R1 (config-applet) #action 1.4 cli command "no shut"
R1 (config-applet) #action 1.5 cli command "end"
R1 (config-applet) #end
```

EEM applets are divided into three main elements: the name, the trigger (or event) and the action(s). In this example, the command “event manager applet TST” is used to name the applet “TST”.

We define our trigger with the command “event none sync yes”. A trigger is used to tell the applet to execute.

“Event none” states that there is to be a manual trigger (i.e. that the applet needs to be executed from the CLI with the “event manager run” command). “Sync yes” tells the router to run the commands in the applet synchronously with the router CLI, rather than execute them in a batch. This helps prevent command overruns.

When the applet executes the commands it uses “action” statements. These must be numbered in ascending order but do not have to be consecutive. Generally, it is wise to use the number before the period to represent code sections and after the period the line within the sections that make a function. In our example we have the following action statements:

```
action 1.0 cli command "enable"
action 1.1 cli command "config terminal"
action 1.2 cli command "interface F0/0"
action 1.3 cli command "ip address 10.1.1.1 255.255.255.0"
action 1.4 cli command "no shut"
```

action 1.5 cli command "end"

The syntax here dictates each statement to be a CLI (command line interface) command and in double quotes we specify which regular IOS command needs to be run. You'll notice here we include every command needed to get from enable mode to the F0/0 interface, the applet then executes our configuration commands and then exits back to privileged exec mode. Because this is all one function we chose to number them 1.x for each command line within the function.

To test the configuration:

On R1:

```
R1#event manager run TST
```

You should see the following console messages:

```
%SYS-5-CONFIG_I: Configured from console by on vty0 (EEM:TST)
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
```

```
R1#Show run int f0/0 | B interface
```

```
interface FastEthernet0/0
 ip address 10.1.1.1 255.255.255.0
 duplex auto
 speed auto
end
```

Task 2

Configure an applet named TST on R2 that gives the F0/0 interface an IP address of 10.1.1.2 /24 and enables the interface. The applet should then log a console message that states "IP address assigned successfully".

On R2:

```
R2(config)#Event manager applet TST
R2(config-applet)#event none sync yes
R2(config-applet)#action 3.0 cli command "enable"
R2(config-applet)#action 3.1 cli command "config t"
```

```
R2 (config-applet)#action 3.2 cli command "interface f0/0"
R2 (config-applet)#action 3.3 cli command "ip addr 10.1.1.2 255.255.255.0"
R2 (config-applet)#action 3.4 cli command "no shut"
R2 (config-applet)#action 3.5 cli command "end"

R2 (config-applet)#action 3.6 syslog msg "IP Address assigned successfully"
```

To test the configuration:

On R2:

```
R2#Event manager run TST
```

You should see the following console messages:

```
%SYS-5-CONFIG_I: Configured from console by on vty0 (EEM:TST)
%HA_EM-6-LOG: TST: IP Address assigned successfully
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed
state to up
```

This example is very similar to the first task. The major difference (besides it being used to configure R2 rather than R1) is that there is an additional action to log a syslog message after the IP address of R2 is configured and enabled. Syslog is a protocol used to manage events and alerts. By default syslog messages are sent straight to the router console, so we see the following message on the router console after the script executes:

```
%HA_EM-6-LOG: TST: IP Address assigned successfully
```

To enable a syslog message we used the action statement in the applet of:

```
action 3.6 syslog msg "IP Address assigned successfully"
```

This caused our string of text to be displayed on the console after our other commands executed successfully.

Task 3

Configure R2 or any other router such that when a "ping 10.1.1.1" command is entered the following console message is displayed:

"The command is bypassed, try again later."

On R2:

```
R2(config)#Event manager applet Task3
R2(config-applet)#event cli pattern "ping 10.1.1.1" sync no skip yes
R2(config-applet)#action 8.0 syslog msg "The command is bypassed, try
again later."
```

```
R3(config-applet)#end
```

To test the configuration:

On R2:

```
R2#Ping 10.1.1.1
R2#
```

```
%HA_EM-6-LOG: Task3: The command is bypassed, try again later.
```

In the above example the "Sync NO" option is used to stop the policy from running synchronously, this gave us the ability to use the "Skip" keyword; the "Skip" keyword prevents the issued command (ping 10.1.1.1) from running. Then, the ONLY action is taken which generates a console message.

Task 4

Erase the startup configuration of the routers and switches and reload them before proceeding to the lab.

