BGP Professional Lab 1

Lesson Contents

Your government has asked you to help design its vision of govNet, a large private network that will be used to interconnect all of the region's public sector services. Schools, hospitals, universities, museums, government offices, and various other types of public buildings and services will be connected to this network. govNet will essentially function as the ISP for these public sector institutions. The network will provide Internet connectivity, telephony, and various other network services required for the public sector to operate.

govNet will be separated into several geographically defined administrative areas, each served by a BGP Autonomous System (AS). The whole network will connect to the Internet via a Tier 1 ISP.

Your job is to set up and configure all of the main BGP routers to fulfill the needs of the network. Your work will include:

- Basic BGP configuration of routers including eBGP peerings and iBGP peerings as shown in the topology diagram.
- Manipulation of various BGP attributes to achieve the required advertising of particular networks.
- The implementation of BGP communities to implement traffic engineering and dynamic routing policies.
- Additional advanced BGP configurations including confederations, route filtering, peer groups, and multipath among others, to achieve the required network behavior and capabilities.
- In preparation for migration to IPv6, you will introduce some IPv6 routes to be advertised by the BGP topology.

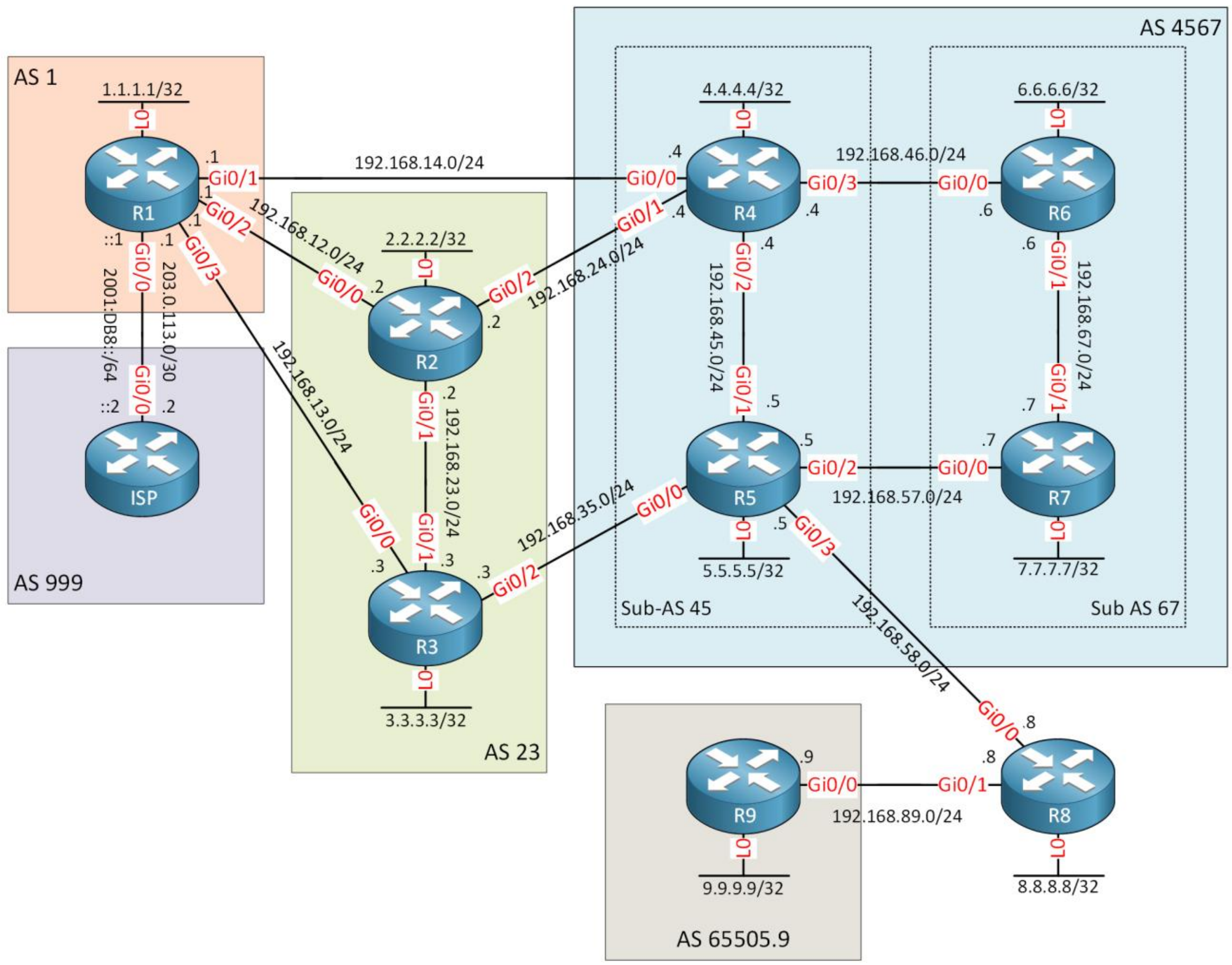
Your expertise will play a key role in advancing govNet to become one of the most sophisticated government networks in the world. This is your chance to demonstrate your potential and contribute to building a network that is not only scalable and reliable but also instrumental in achieving the regional government's ambitious goals.

This lab will give you tasks to configure, and it's up to you to figure out how to accomplish them. This helps you learn how to apply your theoretical knowledge to real-world scenarios. It's created to be challenging and deepen your understanding of BGP.

The lab also assumes you know what BGP is and how it operates. When you go through the lab and get stuck, you can follow the links to the different lessons in the solution section. You will get maximum value out of this when you try to solve everything on your own and only refer to the solution when you get stuck.

Topology

Here is the topology for this lab:



Let me explain this topology:

- Nine routers belong to govNet.
- govNet uses these AS numbers:
    - AS 1
    - AS 23
    - AS 4567
        - This is a confederation that contains sub-ASes 45 and 67.
    - AS 65505.9
- The ISP router belongs to the Tier 1 ISP.
- The ISP router contains several public networks in its BGP table.
- All routers except for R8 participate in BGP.
- AS 999 belongs to the Tier 1 ISP.

The following startup configs include the configuration of all hostnames, interfaces, IPv4 addresses, and IPv6 addresses. You can load these on your own hardware or your favorite emulator.

Want to take a look for yourself? Here, you will find the startup configuration of each device.

Tasks

The tasks are described as if you are on the job, and someone asks you to configure this network. It's not a cookbook where we tell you exactly what to do and in what order. You will have to decide for yourself. This will help you think through what is needed instead of just typing the commands we tell you to.

OSPF

We require an IGP such as OSPF for the iBGP peerings.

- Configure OSPF in ASes 23 and 4567 to prepare for the iBGP peerings.

Tunnel

R8 doesn't have an AS number yet, but R5 and R9 must establish a BGP neighbor adjacency. It's up to you to come up with a solution.

- Configure a tunnel solution so that R5 and R9 can establish a neighbor adjacency using their loopback interfaces.
- You are not allowed to make any changes to R8 except for two static routes.
- You are allowed to create two static routes on R5 and R9.

BGP Fundamentals

This section describes the fundamental BGP configurations that must be implemented.

Create iBGP and eBGP IPv4 peerings

Next, you will create the BGP peerings as shown in the network topology:

- Configure iBGP peerings with loopback interfaces. Do not use physical interfaces.
- Configure eBGP peerings with physical interfaces, with one exception.

Here is an overview of all required BGP peerings:

- eBGP peering between R1 and the ISP router.
    - eBGP peering between R1 and R2.
    - eBGP peering between R1 and R3.
    - eBGP peering between R1 and R4.
    - iBGP peering between R2 and R3.
    - eBGP peering between R2 and R4.
    - eBGP peering between R3 and R5.
    - Peerings between routers in the confederation AS 4567 and their sub ASes.
    - eBGP multi-hop peering between R5 and R9:
        - Use loopback interfaces as the source for eBGP.
        - Ensure that the 4-byte AS number is correctly interpreted by R5 and the rest of the BGP topology.

Create eBGP IPv6 peering

govNet wants to prepare for IPv6, luckily the ISP supports this.

- Configure a second eBGP peering between the ISP and R1 router using IPv6.
- Advertise the IPv6 networks on the ISP router in BGP.
- Use the IPv6 addresses on the physical interface for the BGP peering.

Advertise Routes

Next, you will ensure that routes are advertised within the BGP topology like so:

- Advertise the directly connected IPv4 networks on the ISP router using redistribution.
- Advertise the IPv4 network of Loopback 0 on the ISP router into BGP using the `network` command.
- Advertise the IPv6 prefixes on loopbacks 0 to 19 on the ISP router into BGP via the IPv6 peering using the `network` command.
- Configure AS23 to receive IPv6 routes over the IPv4 eBGP peerings with R1.
- Configure the network so that IPv6 routes are propagated into AS 1 and AS 23 but not beyond.
- Advertise the IPv4 address on the loopback interfaces of all BGP routers into BGP using the `network` command.

Next-Hop Self

- Configure next hop self for IPv4 BGP peerings on all routers that require this so that traffic can be routed outside of the local AS.

Advanced BGP Features

The next step is to configure some more advanced BGP features.

BGP Auto Summary

The R9 router is expected to contain multiple networks in the future that will be advertised to the rest of the network. These networks will fall within the 9.0.0.0/8 address range.

- Configure auto-summary on R9 to make BGP routing more efficient in the future.

BGP Summarization

The ISP router advertises several networks, which can be summarized. To make BGP tables smaller, you must summarize some of the ISP's networks:

- Summarize the networks on loopback 10,11 and 12 using the most specific summary address.
- Ensure that only the summary network is added to the BGP table. Other networks that fall within the range of the summary have to be suppressed.

BGP Weight Attribute

You are asked to ensure that the 12.34.0.0/16 be routed from R4 via AS 23 rather than directly via AS 1. To achieve this, you must use the weight attribute.

- Configure the weight attribute of the route with a route map configuration on R4.

BGP Local Preference Attribute

AS 23 is expected to grow in size in the coming months, and additional iBGP routers will be added in the future. We want to make sure traffic is routed in an optimal way.

- Configure BGP so that packets destined for network 23.45.0.0/16 from within AS 23 always exit AS 23 via router R2.
- Ensure no other networks are influenced.

BGP AS Path Prepending

You are asked to ensure that any traffic from R1 that is destined to the loopback address of R9 be routed via AS 23 rather than via AS 4567.

- Configure AS path prepending on R4 and make the path via AS 4567 three times as long as the path via AS 23.
- Ensure no other networks are influenced.

BGP Origin Code Attribute

Because the 66.77.0.0/17 network advertised by the ISP router may also be advertised elsewhere in the network, you have been asked to modify the way it is injected into BGP.

- Configure the ISP router so that other routers see network 66.77.0.0/17 with an origin code similar to a network that is not injected with redistribution.
- Ensure no other networks are influenced.

BGP MED Attribute

In a previous task, you configured AS path prepending so that traffic from R1 to R9's loopback would be routed via AS 23. There are two paths via AS 23 to reach that destination.

- Configure MED on R3 so that the path via R2 is preferred over the path via R3.
- Ensure that no other networks are influenced.

BGP Prefer eBGP over iBGP

Take a look at the 91.200.0.0/18 route, as it has been advertised by BGP to R3. Examine the various entries in the routing table. There should be three.

- Determine which of those possible BGP paths has been chosen as the best path and verify the reason for this using the appropriate verification commands.

BGP Communities

You are asked to adjust how various advertised routes are distributed throughout the network using BGP communities. Specifically, you are asked to use the appropriate BGP community to:

- Prevent the 102.64.0.0/18 network from being advertised further downstream by R2, R3, and R4. This configuration has to be applied to R1.
- Ensure that the 123.45.0.0/17 network is advertised from R1 to ASes 23 and 4567 so that the eBGP peers will not readvertise this route to other eBGP peers.
- Ensure that the 130.25.0.0/18 network is advertised into sub-AS 45 but not into sub-AS 67 or beyond.

BGP Route Filtering

You are asked to filter out some routes in the BGP topology. Specifically, you are asked to:

- Filter out all networks with a /18 prefix length:
    - These routes should not appear in the BGP table of R9.
    - All configurations should be applied using prefix lists configured on R5.
- Filter out any routes with a prefix length ranging between /22 and /32:
    - These routes should not appear in the BGP table of R9.
    - All configurations should be applied using BGP extended access list filtering prefix lists configured on R9.

BGP Transit AS

You are asked to ensure that AS 23 will never be used as a transit AS for the 175.45.200.0/21 network.

- Use distribute list filtering with an access list to achieve this.

BGP AS Path Filter

Employ AS Path Filtering to perform the following:

- Ensure that any routes that pass through AS 23 are filtered and thus are prevented from entering the BGP table of R6 using AS path filtering.
- Configure AS path filtering on R3 to remove any routers that have passed through AS 4567 but keep any routes that have originated from AS 4567.

BGP Route Dampening

To prevent many changes to the BGP tables, you must implement route dampening.

- Enable route dampening on R2 for the networks on loopback 13, 14, and 15 of the ISP route.
- Use the following parameters:
    - Half-life: 15
    - Reuse: 750
    - Suppress: 2000
    - Max-suppress-time: 60

BGP Peer Groups

R1 will connect to more routers in AS 23 in the future.

- Configure peer groups on R1 and group as many of R1's neighbors into peer groups as possible.

BGP Soft Reconfiguration and Route Refresh

- Configure routers R6 and R7 such that their inter-sub-AS peerings are enabled with soft reconfiguration in an inbound direction.

BGP Multipath

The 192.168.23.0/24 network between routers R2 and R3 must be advertised via BGP by both R2 and R3. This network is expected to be expanded in the near future, as more devices will be connected to it. The services running on that subnet must be accessible from the R1 router with redundant routing via both R2 and R3.

- Configure BGP multipath so that both paths (via R2 and R3) from R1 will be chosen as the best path.
- No more than two paths should be chosen as the best paths.
- Any other paths found in the BGP table of R1 that appear multiple times and have the same Weight, Local Preference, AS Path, Origin code, MED and IGP metric should also be routed redundantly using BGP.

BGP Next-Hop Tracking

- Configure BGP next hop tracking on R2 so that the next hop IP addresses of R2 and R3 are actively tracked.
- Modify the next hop trigger delay to 10 seconds for the IPv4 address family.

Solution

The lab was configured using the following router image for all devices:

```
Cisco IOS Software, IOSv Software (VIOS-ADVENTERPRISEK9-M), Version
15.9(3)M6, RELEASE SOFTWARE (fc1).
```

The following sections show how to achieve the requirements as described in the tasks section above.

OSPF

The iBGP routers in AS 23 and AS 4567 use loopback interfaces for the neighbor adjacency so we need an IGP such as OSPF to advertise these loopback interfaces. This is how to do it:

```
R2(config)#router ospf 1

R2(config-router)#network 2.2.2.2 0.0.0.0 area 0

R2(config-router)#network 192.168.23.0 0.0.0.255 area 0

R3(config)#router ospf 1

R3(config-router)#network 3.3.3.3 0.0.0.0 area 0

R3(config-router)#network 192.168.23.0 0.0.0.255 area 0

R4(config)#router ospf 1

R4(config-router)#network 4.4.4.4 0.0.0.0 area 0

R4(config-router)#network 192.168.45.0 0.0.0.255 area 0

R4(config-router)#network 192.168.46.0 0.0.0.255 area 0

R5(config)#router ospf 1

R5(config-router)#network 5.5.5.5 0.0.0.0 area 0
```

```
R5(config-router)#network 192.168.45.0 0.0.0.255 area 0

R5(config-router)#network 192.168.57.0 0.0.0.255 area 0

R6(config)#router ospf 1

R6(config-router)#network 6.6.6.6 0.0.0.0 area 0

R6(config-router)#network 192.168.46.0 0.0.0.255 area 0

R6(config-router)#network 192.168.67.0 0.0.0.255 area 0

R7(config)#router ospf 1

R7(config-router)#network 7.7.7.7 0.0.0.0 area 0

R7(config-router)#network 192.168.57.0 0.0.0.255 area 0

R7(config-router)#network 192.168.67.0 0.0.0.255 area 0
```

This takes care of all required OSPF neighbor adjacencies and advertises the loopback interfaces:

```
R2#show ip ospf neighbor



Neighbor ID     Pri   State           Dead Time   Address
Interface

3.3.3.3           1   FULL/DR         00:00:31    192.168.23.3
GigabitEthernet0/1

R3#show ip ospf neighbor



Neighbor ID     Pri   State           Dead Time   Address
Interface
```

```
2.2.2.2           1   FULL/BDR        00:00:39    192.168.23.2
GigabitEthernet0/1

R4#show ip ospf neighbor



Neighbor ID     Pri   State           Dead Time   Address
Interface

6.6.6.6           1   FULL/DR         00:00:35    192.168.46.6
GigabitEthernet0/3

5.5.5.5           1   FULL/DR         00:00:38    192.168.45.5
GigabitEthernet0/2

R5#show ip ospf neighbor



Neighbor ID     Pri   State           Dead Time   Address
Interface

7.7.7.7           1   FULL/DR         00:00:32    192.168.57.7
GigabitEthernet0/2

4.4.4.4           1   FULL/BDR        00:00:39    192.168.45.4
GigabitEthernet0/1

R6#show ip ospf neighbor



Neighbor ID     Pri   State           Dead Time   Address
Interface

7.7.7.7           1   FULL/DR         00:00:34    192.168.67.7
GigabitEthernet0/1

4.4.4.4           1   FULL/BDR        00:00:35    192.168.46.4
GigabitEthernet0/0

R7#show ip ospf neighbor
```

```
Neighbor ID     Pri   State           Dead Time   Address
Interface

6.6.6.6           1   FULL/BDR        00:00:34    192.168.67.6
GigabitEthernet0/1

5.5.5.5           1   FULL/BDR        00:00:36    192.168.57.5
GigabitEthernet0/0
```

Tunnel

We can create a GRE tunnel between R5 and R9 so they can establish a neighbor adjacency. Without a tunnel, R8 would drop traffic because it doesn't run BGP.

First, create two static routes so R5 and R9 know how to reach each others tunnel endpoints:

```
R5(config)#ip route 192.168.89.9 255.255.255.255 192.168.58.8

R9(config-if)#ip route 192.168.58.5 255.255.255.255 192.168.89.8
```

Now we can configure the tunnel interfaces:

```
R5(config)#interface Tunnel 0

R5(config-if)#tunnel source 192.168.58.5

R5(config-if)#tunnel destination 192.168.89.9

R5(config-if)#ip address 192.168.59.5 255.255.255.0

R9(config)#interface Tunnel 0

R9(config-if)#tunnel source 192.168.89.9

R9(config-if)#tunnel destination 192.168.58.5
```

```
R9(config-if)#ip address 192.168.59.9 255.255.255.0
```

The tunnel now works:

```
R5#show interfaces Tunnel 0 | include up

Tunnel0 is up, line protocol is up

  Tunnel linestate evaluation up
```

Let's make sure the tunnel works:

```
R5#ping 192.168.59.9

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.59.9, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

The last thing to do is to create two more static routes so that R5 and R9 can reach each other's loopback interfaces through the tunnel:

```
R5(config)#ip route 9.9.9.9 255.255.255.255 192.168.59.9

R9(config)#ip route 5.5.5.5 255.255.255.255 192.168.59.5
```

Let's send a quick ping:

```
R5#ping 9.9.9.9 source 5.5.5.5

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 9.9.9.9, timeout is 2 seconds:
```

```
Packet sent with a source address of 5.5.5.5

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

The tunnel is now ready, and R5 and R9 can reach each other's loopback interfaces through the tunnel. We can use this later for the BGP neighbor adjacency.

BGP Fundamentals

The following sections describe the solution to the requirements for the fundamental BGP configurations.

Create iBGP and eBGP IPv4 peerings

To fulfill the stated requirements for the configuration of eBGP and eBGP IPv4 peerings, perform the following configurations:

iBGP Peerings in AS 23

For the establishment of iBGP peerings in AS 23, use the Loopback 0 interfaces as the source of BGP messages.

```
R2(config)#router bgp 23

R2(config-router)#neighbor 3.3.3.3 remote-as 23

R2(config-router)#neighbor 3.3.3.3 update-source Loopback0

R3(config)#router bgp 23

R3(config-router)#neighbor 2.2.2.2 remote-as 23

R3(config-router)#neighbor 2.2.2.2 update-source Loopback0
```

To verify that the iBGP peerings have been established, issue the following command on each BGP peer:

```
R2#show ip bgp summary

BGP router identifier 2.2.2.2, local AS number 23

BGP table version is 1, main routing table version 1



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

3.3.3.3         4           23       6       6        1    0    0 00:02:00
0

R3#show ip bgp summary

BGP router identifier 3.3.3.3, local AS number 23

BGP table version is 1, main routing table version 1



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

2.2.2.2         4           23       5       5        1    0    0 00:01:29
0
```

Notice the AS 23 that appears as the AS of the remote peer in both cases. This indicates that the AS is the same as the one in the local router, which ensures that these are iBGP peerings.

iBGP peerings within the Confederation AS 4567

To create a confederation in AS 4567 as described in the task requirements, perform the following configuration steps. Remember to use the Loopback 0 interfaces as the source of BGP messages:

```
R4(config)#router bgp 45
```

```
R4(config-router)#bgp confederation identifier 4567

R4(config-router)#bgp confederation peers 67

R4(config-router)#neighbor 5.5.5.5 remote-as 45

R4(config-router)#neighbor 5.5.5.5 update-source Loopback0

R4(config-router)#neighbor 6.6.6.6 remote-as 67

R4(config-router)#neighbor 6.6.6.6 update-source Loopback0

R4(config-router)#neighbor 6.6.6.6 ebgp-multihop 2

R5(config)#router bgp 45

R5(config-router)#bgp confederation identifier 4567

R5(config-router)#bgp confederation peers 67

R5(config-router)#neighbor 4.4.4.4 remote-as 45

R5(config-router)#neighbor 4.4.4.4 update-source Loopback0

R5(config-router)#neighbor 7.7.7.7 remote-as 67

R5(config-router)#neighbor 7.7.7.7 update-source Loopback0

R5(config-router)#neighbor 7.7.7.7 ebgp-multihop 2

R6(config)#router bgp 67

R6(config-router)#bgp confederation identifier 4567

R6(config-router)#bgp confederation peers 45

R6(config-router)#neighbor 7.7.7.7 remote-as 67

R6(config-router)#neighbor 7.7.7.7 update-source Loopback0

R6(config-router)#neighbor 4.4.4.4 remote-as 45
```

```
R6(config-router)#neighbor 4.4.4.4 update-source Loopback0

R6(config-router)#neighbor 4.4.4.4 ebgp-multihop 2

R7(config)#router bgp 67

R7(config-router)#bgp confederation identifier 4567

R7(config-router)#bgp confederation peers 45

R7(config-router)#neighbor 6.6.6.6 remote-as 67

R7(config-router)#neighbor 6.6.6.6 update-source Loopback0

R7(config-router)#neighbor 5.5.5.5 remote-as 45

R7(config-router)#neighbor 5.5.5.5 update-source Loopback0

R7(config-router)#neighbor 5.5.5.5 ebgp-multihop 2
```

The above configurations have now created AS 4567 as a confederation:

- Sub-AS 45 with R4 and R5.
- Sub-AS 67 with R6 and R7.

To verify this, use the following command on each of the routers to determine the BGP peerings:

```
R4#show ip bgp summary

BGP router identifier 4.4.4.4, local AS number 45

BGP table version is 1, main routing table version 1



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

```
5.5.5.5         4          45      96      96        1    0    0 01:24:42
0

6.6.6.6         4          67      33      33        1    0    0 00:26:30
0

R5#show ip bgp summary

BGP router identifier 5.5.5.5, local AS number 45

BGP table version is 1, main routing table version 1



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

4.4.4.4         4          45     101     101        1    0    0 01:29:05
0

7.7.7.7         4          67      31      32        1    0    0 00:25:30
0
```

Notice the ASes that appear for each neighbor are those of the sub-ASes and not of the confederation AS. Similarly, you can see the results of these commands on routers R6 and R7:

```
R6#show ip bgp summary

BGP router identifier 6.6.6.6, local AS number 67

BGP table version is 1, main routing table version 1



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

4.4.4.4         4          45      41      40        1    0    0 00:32:52
0
```

```
7.7.7.7         4           67      34      35         1    0    0 00:28:09
0

R7#show ip bgp summary

BGP router identifier 7.7.7.7, local AS number 67

BGP table version is 1, main routing table version 1



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

5.5.5.5         4           45      35      34         1    0    0 00:27:55
0

6.6.6.6         4           67      35      35         1    0    0 00:28:36
0
```

eBGP Peerings

For the establishment of eBGP peerings, as with the iBGP peerings, use the Loopback 0 interfaces as the source of BGP messages. This means you need to configure eBGP Multihop to make this work.

There are two eBGP peerings that require a closer look:

- R5 and R9 where R9 uses a 4-byte ASN and is two hops away.
- R4 and R5 are within the BGP confederation.

Here is the configuration:

```
ISP(config)#router bgp 999

ISP(config-router)#neighbor 203.0.113.1 remote-as 1

R1(config)#router bgp 1

R1(config-router)#neighbor 203.0.113.2 remote-as 999
```

```
R1(config-router)#neighbor 192.168.12.2 remote-as 23

R1(config-router)#neighbor 192.168.13.3 remote-as 23

R1(config-router)#neighbor 192.168.14.4 remote-as 4567

R2(config)#router bgp 23

R2(config-router)#neighbor 192.168.12.1 remote-as 1

R2(config-router)#neighbor 192.168.24.4 remote-as 4567

R3(config)#router bgp 23

R3(config-router)#neighbor 192.168.13.1 remote-as 1

R3(config-router)#neighbor 192.168.35.5 remote-as 4567

R4(config)#router bgp 45

R4(config-router)#neighbor 192.168.14.1 remote-as 1

R4(config-router)#neighbor 192.168.24.2 remote-as 23

R5(config)#router bgp 45

R5(config-router)#neighbor 192.168.35.3 remote-as 23

R5(config-router)#neighbor 9.9.9.9 remote-as 65505.9

R5(config-router)#neighbor 9.9.9.9 update-source Loopback0

R5(config-router)#neighbor 9.9.9.9 ebgp-multihop 2

R9(config)#router bgp 65505.9

R9(config-router)#neighbor 5.5.5.5 remote-as 4567

R9(config-router)#neighbor 5.5.5.5 update-source Loopback0

R9(config-router)#neighbor 5.5.5.5 ebgp-multihop 2
```

At this point, all eBGP peerings should be established. To verify this, use the following command on each eBGP peer:

```
ISP#show ip bgp summary

BGP router identifier 221.25.0.1, local AS number 999

BGP table version is 1, main routing table version 1



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

203.0.113.1     4            1      43      43        1    0    0 00:36:02
0
```

Ensure that the neighbor Up/Down state has a time value that indicates that the peering is up. This should be done from both sides of the peering:

```
R1#show ip bgp summary

BGP router identifier 1.1.1.1, local AS number 1

BGP table version is 1, main routing table version 1



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

192.168.12.2    4           23      32      30        1    0    0 00:25:53
0

192.168.13.3    4           23      27      25        1    0    0 00:21:45
0

192.168.14.4    4         4567      23      20        1    0    0 00:17:20
0
```

```
203.0.113.2     4          999      46      46        1    0    0 00:38:30
0
```

Ensure that all expected eBGP peerings are up, including the peerings within the confederation AS of 4567 as well as between R5 and R9 which are not directly connected:

```
R4#show ip bgp summary

BGP router identifier 4.4.4.4, local AS number 45

BGP table version is 1, main routing table version 1



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

192.168.14.1    4            1      69      72        1    0    0 01:01:42
0

192.168.24.2    4           23      70      70        1    0    0 01:01:10
0

5.5.5.5         4           45     179     178        1    0    0 02:39:21
0

6.6.6.6         4           67     115     116        1    0    0 01:41:10
0

R6#show ip bgp summary

BGP router identifier 6.6.6.6, local AS number 67

BGP table version is 1, main routing table version 1



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

```
4.4.4.4         4           45     118     117        1    0    0 01:42:44
0

7.7.7.7         4           67     111     111        1    0    0 01:38:01
0

R7#show ip bgp summary

BGP router identifier 7.7.7.7, local AS number 67

BGP table version is 1, main routing table version 1



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

5.5.5.5         4           45     111     112        1    0    0 01:37:46
0

6.6.6.6         4           67     111     112        1    0    0 01:38:27
0

R5#show ip bgp summary

BGP router identifier 5.5.5.5, local AS number 45

BGP table version is 1, main routing table version 1



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

4.4.4.4         4           45    6329    6334        1    0    0 00:21:47
0

7.7.7.7         4           67    6269    6271        1    0    0 00:18:18
0

9.9.9.9         4   4292935689    6169    6174        1    0    0 00:12:19
0
```

```
192.168.35.3    4           23      16      16        1    0    0 00:13:45
0

R9#show ip bgp summary

BGP router identifier 9.9.9.9, local AS number 4292935689

BGP table version is 1, main routing table version 1



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

5.5.5.5         4         4567      22      23        1    0    0 00:17:51
0
```

At this point, all iBGP and eBGP peerings have been established, including within the BGP confederation 4567 and the multihop peering between R5 and R9.

It's important to note that at this moment, no routes have yet been exchanged by BGP. Only OSPF and static routes within each AS have been established such that BGP routers can reach their intended peers.

Create eBGP IPv6 peering

To create an [IPv6 eBGP peering]() between the ISP and R1 routers, perform the following configurations:

```
ISP(config)#ipv6 unicast-routing

R1(config)#ipv6 unicast-routing

ISP(config)#router bgp 999

ISP(config-router)#neighbor 2001:db8::1 remote-as 1

ISP(config-router)#address-family ipv4

ISP(config-router-af)#no neighbor 2001:db8::1 activate
```

```
ISP(config-router-af)#exit

ISP(config-router)#address-family ipv6

ISP(config-router-af)#neighbor 2001:db8::1 activate

R1(config)#router bgp 1

R1(config-router)#neighbor 2001:db8::2 remote-as 999

R1(config-router)#address-family ipv4

R1(config-router-af)#no neighbor 2001:db8::2 activate

R1(config-router-af)#exit

R1(config-router)#address-family ipv6

R1(config-router-af)#neighbor 2001:db8::2 activate
```

To verify this IPv6 eBGP peering, use the following command on both routers:

```
ISP#show bgp ipv6 unicast summary

BGP router identifier 221.25.0.1, local AS number 999

BGP table version is 1, main routing table version 1



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

2001:DB8::1     4            1    3150    3146        1    0    0 00:15:55
0

R1#show bgp ipv6 unicast summary

BGP router identifier 1.1.1.1, local AS number 1

BGP table version is 1, main routing table version 1
```

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

2001:DB8::2     4          999    3147    3151        1    0    0 00:14:31
0
```

It may be a good idea at this point to ensure that the IPv4 eBGP peering is still up an running, using the same verification commands as described in the previous section.

Advertise Routes in BGP

The following sections show how to fulfill the requirements as described in the task list.

IPv4 prefixes on ISP Router

To advertise the IPv4 address spaces on the loopback interfaces of the ISP router as described in the task list, perform the following configurations.

```
ISP(config)#router bgp 999

ISP(config-router)#address-family ipv4

ISP(config-router-af)#redistribute connected

ISP(config-router-af)#network 10.0.0.1 mask 255.255.255.255
```

To verify that these addresses are being advertised via eBGP, issue the following command:

```
R1#show ip bgp

BGP table version is 23, 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, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>   10.0.0.1/32      203.0.113.2              0             0 999 i
 *>   12.34.0.0/16     203.0.113.2              0             0 999 ?
 *>   23.45.0.0/16     203.0.113.2              0             0 999 ?
 *>   66.77.0.0/17     203.0.113.2              0             0 999 ?
 *>   89.100.0.0/18    203.0.113.2              0             0 999 ?
 *>   91.200.0.0/18    203.0.113.2              0             0 999 ?
 *>   102.64.0.0/18    203.0.113.2              0             0 999 ?
 *>   123.45.0.0/17    203.0.113.2              0             0 999 ?
 *>   130.25.0.0/18    203.0.113.2              0             0 999 ?
 *>   175.45.200.0/21  203.0.113.2              0             0 999 ?
 *>   183.77.220.0/22  203.0.113.2              0             0 999 ?
 *>   185.100.0.0/19   203.0.113.2              0             0 999 ?
 *>   190.30.128.0/17  203.0.113.2              0             0 999 ?
 *>   195.225.0.0/19   203.0.113.2              0             0 999 ?
```

```
*>   199.10.192.0/22  203.0.113.2              0             0 999 ?

r>   203.0.113.0/30   203.0.113.2              0             0 999 ?

*>   210.45.128.0/23  203.0.113.2              0             0 999 ?

*>   212.12.16.0/21   203.0.113.2              0             0 999 ?

*>   216.80.192.0/22  203.0.113.2              0             0 999 ?

*>   220.85.200.0/23  203.0.113.2              0             0 999 ?

*>   221.25.0.0/19    203.0.113.2              0             0 999 ?
```

Ensure that all routes have been learned via the redistribution of routes (within an origin code of "?") except for the network of Loopback0 of the ISP router (origin code of "i"). Also, note that the 203.0.113.0/30 network was learned via BGP, but it wasn't installed into the routing table (status code "r"). This is because R1 has a directly connected interface to that network, via which the network was added to the routing table. This directly connected route has a lower AD than eBGP.

Perform the same command on the ISP router to view the results there as well.

IPv6 prefixes on ISP Router

To advertise IPv6 addresses on the ISP router and fulfill the requirements of this task, perform the following configurations:

```
ISP(config)#router bgp 999

ISP(config-router)#address-family ipv6

ISP(config-router-af)#network 2001:0db8:23:45::/64

ISP(config-router-af)#network 2001:0db8:66:77::/64

ISP(config-router-af)#network 2001:0db8:89:100::/64

ISP(config-router-af)#network 2001:0db8:91:200::/64
```

```
ISP(config-router-af)#network 2001:0db8:102:64::/64

ISP(config-router-af)#network 2001:0db8:123:45::/64

ISP(config-router-af)#network 2001:0db8:130:25::/64

ISP(config-router-af)#network 2001:0db8:175:45::/64

ISP(config-router-af)#network 2001:0db8:183:77::/64

ISP(config-router-af)#network 2001:0db8:185:100::/64

ISP(config-router-af)#network 2001:0db8:190:32::/64

ISP(config-router-af)#network 2001:0db8:195:225::/64

ISP(config-router-af)#network 2001:0db8:199:10::/64

ISP(config-router-af)#network 2001:0db8:210:45::/64

ISP(config-router-af)#network 2001:0db8:212:12::/64

ISP(config-router-af)#network 2001:0db8:216:80::/64

ISP(config-router-af)#network 2001:0db8:220:85::/64

ISP(config-router-af)#network 2001:0db8:225:25::/64
```

To verify that these networks are being advertised to R1 via the IPv6 eBGP peering, issue the following command:

```
R1#show bgp ipv6 unicast

BGP table version is 21, 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, f RT-
Filter,
```

```
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>   2001:DB8:1::1/128
                       2001:DB8::2              0             0 999 i
 *>   2001:DB8:12:34::/64
                       2001:DB8::2              0             0 999 i
 *>   2001:DB8:23:45::/64
                       2001:DB8::2              0             0 999 i
 *>   2001:DB8:66:77::/64
                       2001:DB8::2              0             0 999 i
 *>   2001:DB8:89:100::/64
                       2001:DB8::2              0             0 999 i
 *>   2001:DB8:91:200::/64
                       2001:DB8::2              0             0 999 i
 *>   2001:DB8:102:64::/64
                       2001:DB8::2              0             0 999 i
 *>   2001:DB8:123:45::/64
```

```
                    2001:DB8::2              0              0 999 i
*>   2001:DB8:130:25::/64
                    2001:DB8::2              0              0 999 i
*>   2001:DB8:175:45::/64
                    2001:DB8::2              0              0 999 i
*>   2001:DB8:183:77::/64
                    2001:DB8::2              0              0 999 i
*>   2001:DB8:185:100::/64
                    2001:DB8::2              0              0 999 i
*>   2001:DB8:190:32::/64
                    2001:DB8::2              0              0 999 i
*>   2001:DB8:195:225::/64
                    2001:DB8::2              0              0 999 i
*>   2001:DB8:199:10::/64
                    2001:DB8::2              0              0 999 i
*>   2001:DB8:210:45::/64
                    2001:DB8::2              0              0 999 i
*>   2001:DB8:212:12::/64
                    2001:DB8::2              0              0 999 i
*>   2001:DB8:216:80::/64
                    2001:DB8::2              0              0 999 i
```

```
 *>   2001:DB8:220:85::/64

                        2001:DB8::2            0             0 999 i

 *>   2001:DB8:225:25::/64

                        2001:DB8::2            0             0 999 i
```

Notice that all learned routes have an origin code of "i" which indicates that they were added using the `network` command. Also note that the next hop router is indicated by an IPv6 address, which means that the routes have been learned via the IPv6 eBGP peering with the ISP router.

Issue this same command on the ISP router to see similar results.

IPv6 prefixes into AS 23

To prepare AS 23 to [receive IPv6 routes via IPv4 eBGP peerings](#) with R1, perform the following configurations:

```
R1(config)#router bgp 1

R1(config-router)#address-family ipv6

R1(config-router-af)#neighbor 192.168.12.2 activate

R1(config-router-af)#neighbor 192.168.13.3 activate

R2(config)#ipv6 unicast-routing

R2(config)#router bgp 23

R2(config-router)#address-family ipv6

R2(config-router-af)#neighbor 192.168.12.1 activate

R2(config-router-af)#neighbor 3.3.3.3 activate

R3(config)#ipv6 unicast-routing
```

```
R3(config)#router bgp 23

R3(config-router)#address-family ipv6

R3(config-router-af)#neighbor 192.168.13.1 activate

R3(config-router-af)#neighbor 2.2.2.2 activate
```

To verify that the IPv4 eBGP and iBGP peerings have been enabled to share IPv6 routes within AS 23, issue the following commands:

```
R2#show bgp ipv6 unicast summary

BGP router identifier 2.2.2.2, local AS number 23

BGP table version is 1, main routing table version 1

20 network entries using 3360 bytes of memory

20 path entries using 2160 bytes of memory

1/0 BGP path/bestpath attribute entries using 160 bytes of memory

4 BGP AS-PATH entries using 96 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 5776 total bytes of memory

BGP activity 42/0 prefixes, 218/132 paths, scan interval 60 secs



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

3.3.3.3         4           23      14      14        1    0    0 00:05:23
0
```

```
192.168.12.1    4            1      37      38         1    0    0 00:23:21
20

R3#show bgp ipv6 unicast summary

BGP router identifier 3.3.3.3, local AS number 23

BGP table version is 1, main routing table version 1

20 network entries using 3360 bytes of memory

20 path entries using 2160 bytes of memory

1/0 BGP path/bestpath attribute entries using 160 bytes of memory

2 BGP AS-PATH entries using 48 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 5728 total bytes of memory

BGP activity 42/0 prefixes, 152/88 paths, scan interval 60 secs



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

2.2.2.2         4           23      14      14         1    0    0 00:05:51
0

192.168.13.1    4            1      25      22         1    0    0 00:11:40
20

R1#show bgp ipv6 unicast summary

BGP router identifier 1.1.1.1, local AS number 1

BGP table version is 21, main routing table version 21
```

```
20 network entries using 3360 bytes of memory

20 path entries using 2160 bytes of memory

1/1 BGP path/bestpath attribute entries using 160 bytes of memory

1 BGP AS-PATH entries using 24 bytes of memory

0 BGP route-map cache entries using 0 bytes of memory

0 BGP filter-list cache entries using 0 bytes of memory

BGP using 5704 total bytes of memory

BGP activity 42/0 prefixes, 42/0 paths, scan interval 60 secs



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

2001:DB8::2     4          999    3423    3422       21    0    0 2d03h
20

192.168.12.2    4           23      40      40       21    0    0 00:25:23
0

192.168.13.3    4           23      23      26       21    0    0 00:13:14
0
```

Note that the entries with an IPv4 address in the neighbor column indicate that IPv6 routes are being shared across an IPv4 BGP peering.

To view the routes that have been advertised via BGP into AS 23, issue the following command:

```
R2#show bgp ipv6 unicast

BGP table version is 1, local router ID is 2.2.2.2
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *    2001:DB8:1::1/128

                       ::FFFF:192.168.12.1

                                                               0 1 999 i

 *    2001:DB8:12:34::/64

                       ::FFFF:192.168.12.1

                                                               0 1 999 i

 *    2001:DB8:23:45::/64

                       ::FFFF:192.168.12.1

                                                               0 1 999 i

 *    2001:DB8:66:77::/64

                       ::FFFF:192.168.12.1

                                                               0 1 999 i
```

```
*    2001:DB8:89:100::/64
                        ::FFFF:192.168.12.1
                                                             0 1 999 i
*    2001:DB8:91:200::/64
                        ::FFFF:192.168.12.1
                                                             0 1 999 i
*    2001:DB8:102:64::/64
                        ::FFFF:192.168.12.1
                                                             0 1 999 i
*    2001:DB8:123:45::/64
                        ::FFFF:192.168.12.1
                                                             0 1 999 i
*    2001:DB8:130:25::/64
                        ::FFFF:192.168.12.1
                                                             0 1 999 i
*    2001:DB8:175:45::/64
                        ::FFFF:192.168.12.1
                                                             0 1 999 i
*    2001:DB8:183:77::/64
                        ::FFFF:192.168.12.1
                                                             0 1 999 i
```

```
 *    2001:DB8:185:100::/64
                       ::FFFF:192.168.12.1
                                                            0 1 999 i
 *    2001:DB8:190:32::/64
                       ::FFFF:192.168.12.1
                                                            0 1 999 i
 *    2001:DB8:195:225::/64
                       ::FFFF:192.168.12.1
                                                            0 1 999 i
 *    2001:DB8:199:10::/64
                       ::FFFF:192.168.12.1
                                                            0 1 999 i
 *    2001:DB8:210:45::/64
                       ::FFFF:192.168.12.1
                                                            0 1 999 i
 *    2001:DB8:212:12::/64
                       ::FFFF:192.168.12.1
                                                            0 1 999 i
 *    2001:DB8:216:80::/64
                       ::FFFF:192.168.12.1
                                                            0 1 999 i
```

```
 *    2001:DB8:220:85::/64
                     ::FFFF:192.168.12.1
                                                             0 1 999 i
 *    2001:DB8:225:25::/64
                     ::FFFF:192.168.12.1
                                                             0 1 999 i
```

Notice that all IPv6 routes are valid routes, but none have been selected as the best path, nor have they been entered into the routing table. This is because there is no reachable next hop IP. The next hop IP address that appears in the BGP table is FFFF:192.168.12.1. This is a placeholder used to indicate that IPv4 is being used to propagate the IPv6 routes. To route IPv6 traffic, we will need to implement an IPv6 network within AS 23, which is outside the scope of this lab.

Loopback IPv4 prefixes on all BGP routers

To advertise the IPv4 addresses of the loopback interfaces of the rest of the BGP peers in the network as described in the task list, perform the following configurations:

```
R1(config)#router bgp 1

R1(config-router)#address-family ipv4

R1(config-router-af)#network 1.1.1.1 mask 255.255.255.255

R2(config)#router bgp 23

R2(config-router)#network 2.2.2.2 mask 255.255.255.255

R3(config)#router bgp 23

R3(config-router)#network 3.3.3.3 mask 255.255.255.255

R4(config)#router bgp 45
```

```
R4(config-router)#network 4.4.4.4 mask 255.255.255.255

R5(config)#router bgp 45

R5(config-router)#network 5.5.5.5 mask 255.255.255.255

R6(config)#router bgp 67

R6(config-router)#network 6.6.6.6 mask 255.255.255.255

R7(config)#router bgp 67

R7(config-router)#network 7.7.7.7 mask 255.255.255.255

R9(config)#router bgp 65505.9

R9(config-router)#network 9.9.9.9 mask 255.255.255.255
```

To verify that these networks are being advertised throughout the network, issue the following command on various BGP routers:

```
R9#show ip bgp

BGP table version is 119, local router ID is 9.9.9.9

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found
```

```
     Network          Next Hop            Metric LocPrf Weight Path

 *>   1.1.1.1/32       5.5.5.5                                0 4567 23 1
i

 *>   2.2.2.2/32       5.5.5.5                                0 4567 23 i

 *>   3.3.3.3/32       5.5.5.5                                0 4567 23 i

 *>   4.4.4.4/32       5.5.5.5                                0 4567 i

 r>   5.5.5.5/32       5.5.5.5                  0             0 4567 i

 *>   6.6.6.6/32       5.5.5.5                                0 4567 i

 *>   7.7.7.7/32       5.5.5.5                                0 4567 i

 *>   9.9.9.9/32       0.0.0.0                  0         32768 i

 *>   10.0.0.1/32      5.5.5.5                                0 4567 23 1
999 i

 ! >-- output omitted --<

R6#show ip bgp

BGP table version is 5, local router ID is 6.6.6.6

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found
```

```
     Network          Next Hop            Metric LocPrf Weight Path
 *    1.1.1.1/32       192.168.14.1             0    100      0 (45) 1 i
 *    2.2.2.2/32       192.168.24.2             0    100      0 (45) 23 i
 *    3.3.3.3/32       192.168.24.2             0    100      0 (45) 23 i
 r i  4.4.4.4/32       4.4.4.4                  0    100      0 (45) i
 r>                    4.4.4.4                  0    100      0 (45) i
 r i  5.5.5.5/32       5.5.5.5                  0    100      0 (45) i
 r>                    5.5.5.5                  0    100      0 (45) i
 *>   6.6.6.6/32       0.0.0.0                  0         32768 i
 r>i  7.7.7.7/32       7.7.7.7                  0    100      0 i
 *    10.0.0.1/32      192.168.14.1             0    100      0 (45) 1 999
i
 ! >-- output omitted --<
R3#show ip bgp
BGP table version is 120, 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, f RT-
Filter,
              x best-external, a additional-path, c RIB-compressed,
              t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
RPKI validation codes: V valid, I invalid, N Not found


     Network          Next Hop            Metric LocPrf Weight Path

 * i  1.1.1.1/32       192.168.12.1             0    100      0 1 i

 *>                    192.168.13.1             0             0 1 i

 r>i  2.2.2.2/32       2.2.2.2                  0    100      0 i

 *>   3.3.3.3/32       0.0.0.0                  0         32768 i

 *    4.4.4.4/32       192.168.13.1                           0 1 4567 i

 *>                    192.168.35.5                           0 4567 i

 * i                   192.168.24.4             0    100      0 4567 i

 * i  5.5.5.5/32       192.168.24.4             0    100      0 4567 i

 *                     192.168.13.1                           0 1 4567 i

 *>                    192.168.35.5             0             0 4567 i

 *>   6.6.6.6/32       192.168.35.5                           0 4567 i

 * i                   192.168.24.4             0    100      0 4567 i

 *                     192.168.13.1                           0 1 4567 i

 *    7.7.7.7/32       192.168.13.1                           0 1 4567 i

 * i                   192.168.24.4             0    100      0 4567 i

 *>                    192.168.35.5                           0 4567 i

 *>   9.9.9.9/32       192.168.35.5                           0 4567
4292935689 i

 * i  10.0.0.1/32      192.168.12.1             0    100      0 1 999 i
```

```
! >-- output omitted --<
```

Examine all of the above output and see if you can decipher the reasons behind the various indicators for these particular routes in the BGP table of each router.

Next-Hop-Self

To apply the next hop self feature within ASes 23 and 4567 so that the next hop IP for BGP routes is correct, issue the following commands:

```
R2(config)#router bgp 23

R2(config-router)#address-family ipv4

R2(config-router-af)#neighbor 3.3.3.3 next-hop-self

R3(config)#router bgp 23

R3(config-router)#address-family ipv4

R3(config-router-af)#neighbor 2.2.2.2 next-hop-self

R4(config)#router bgp 45

R4(config-router)#neighbor 5.5.5.5 next-hop-self

R4(config-router)#neighbor 6.6.6.6 next-hop-self

R5(config)#router bgp 45

R5(config-router)#neighbor 4.4.4.4 next-hop-self

R5(config-router)#neighbor 7.7.7.7 next-hop-self
```

To verify that the next hop self feature has been successfully configured, issue the following commands:

```
R2#show ip bgp 1.1.1.1
```

```
BGP routing table entry for 1.1.1.1/32, version 162

Paths: (3 available, best #1, table default)

  Advertised to update-groups:

     7          8

  Refresh Epoch 2

  1

    192.168.12.1 from 192.168.12.1 (1.1.1.1)

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

      rx pathid: 0, tx pathid: 0x0

  Refresh Epoch 1

  1

    3.3.3.3 (metric 2) from 3.3.3.3 (3.3.3.3)

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

      rx pathid: 0, tx pathid: 0

  Refresh Epoch 3

  4567 1

    192.168.24.4 from 192.168.24.4 (4.4.4.4)

      Origin IGP, localpref 100, valid, external

      rx pathid: 0, tx pathid: 0
```

Note that although the best route to the 1.1.1.1 network from R2 remains via a next hop of 192.168.12.1 (which is via the R1 router), the alternate path learned from its iBGP neighbor R3 has a next hop of 3.3.3.3. This means that R3 placed its own IP address as the next hop IP address for that route when it advertised it to R2.

```
R6#show ip bgp

BGP table version is 132, local router ID is 6.6.6.6

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 * i  1.1.1.1/32       5.5.5.5                  0    100      0 (45) 1 i

 *>                    4.4.4.4                  0    100      0 (45) 1 i

 * i  2.2.2.2/32       5.5.5.5                  0    100      0 (45) 23 i

 *>                    4.4.4.4                  0    100      0 (45) 23 i

 * i  3.3.3.3/32       5.5.5.5                  0    100      0 (45) 23 i

 *>                    4.4.4.4                  0    100      0 (45) 23 i

 ! >-- output omitted --<
```

Notice that each of these entries has been learned via an iBGP peer, indicated by "* i" at the beginning of the entry, and an eBGP peer, indicated by the absence of an "i" at the beginning. Each of the routes shown above was learned from the eBGP peering that has a next hop IP of 4.4.4.4 (R4).

This means that R4 replaced the IP address of its eBGP peer (R1) from which it learned the route (192.168.14.1) with its own IP address (4.4.4.4). So, the next hop self feature was configured correctly. You can see this on R5 as well:

```
R5#show ip bgp

BGP table version is 155, local router ID is 5.5.5.5

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

!

! >-- output omitted --< ! * 12.34.0.0/16 192.168.35.3 0 23 1 999 ? *>i
4.4.4.4                  0    100      0 1 999 ?

 *    23.45.0.0/16     192.168.35.3                          0 23 1 999 ?

 *>i                   4.4.4.4                  0    100      0 1 999 ?

 *    66.77.0.0/17     192.168.35.3                          0 23 1 999 ?

 *>i                   4.4.4.4                  0    100      0 1 999 ?
```

```
*    89.100.0.0/18    192.168.35.3                              0 23 1 999 ?
*>i                   4.4.4.4                0    100          0 1 999 ?
*    91.200.0.0/18    192.168.35.3                              0 23 1 999 ?
*>i                   4.4.4.4                0    100          0 1 999 ?
*    102.64.0.0/18    192.168.35.3                              0 23 1 999 ?
*>i                   4.4.4.4                0    100          0 1 999 ?
*    123.45.0.0/17    192.168.35.3                              0 23 1 999 ?
*>i                   4.4.4.4                0    100          0 1 999 ?
*    130.25.0.0/18    192.168.35.3                              0 23 1 999 ?
*>i                   4.4.4.4                0    100          0 1 999 ?
*    175.45.200.0/21  192.168.35.3                              0 23 1 999 ?
*>i                   4.4.4.4                0    100          0 1 999 ?
*    183.77.220.0/22  192.168.35.3                              0 23 1 999 ?
*>i                   4.4.4.4                0    100          0 1 999 ?
*    185.100.0.0/19   192.168.35.3                              0 23 1 999 ?
*>i                   4.4.4.4                0    100          0 1 999 ?
*    190.30.128.0/17  192.168.35.3                              0 23 1 999 ?
*>i                   4.4.4.4                0    100          0 1 999 ?
*    195.225.0.0/19   192.168.35.3                              0 23 1 999 ?
*>i                   4.4.4.4                0    100          0 1 999 ?
*    199.10.192.0/22  192.168.35.3                              0 23 1 999 ?
```

```
*>i                   4.4.4.4                 0     100       0 1 999 ?
*     203.0.113.0/30  192.168.35.3                            0 23 1 999 ?
*>i                   4.4.4.4                 0     100       0 1 999 ?
*     210.45.128.0/23 192.168.35.3                            0 23 1 999 ?
*>i                   4.4.4.4                 0     100       0 1 999 ?
*     212.12.16.0/21  192.168.35.3                            0 23 1 999 ?
*>i                   4.4.4.4                 0     100       0 1 999 ?
*     216.80.192.0/22 192.168.35.3                            0 23 1 999 ?
*>i                   4.4.4.4                 0     100       0 1 999 ?
*     220.85.200.0/23 192.168.35.3                            0 23 1 999 ?
*>i                   4.4.4.4                 0     100       0 1 999 ?
*     221.25.0.0/19   192.168.35.3                            0 23 1 999 ?
*>i                   4.4.4.4                 0     100       0 1 999 ?
```

Once again, notice the next hop IPs of all of the routes that have been learned via iBGP. You can see that R4 replaced the next hop IP address with its own.

Examine some more of the BGP tables of the various routers to ensure that the next hop self feature is functioning correctly.

Verification of Basic BGP Topology

To test this basic BGP topology, you can perform several checks to ensure that the appropriate routes are being advertised. Note that testing is primarily achieved by examining how the BGP routes have been advertised to each BGP.

When working with BGP, you have to think twice when you use commands like `ping` and `traceroute`. You may see something in the BGP table, but it's not installed in the routing table. Also, make sure you specify a source, because it's possible that your packet makes it to the destination, but the return traffic is dropped because one of the routers doesn't know about your source.

Examine BGP Table IPv4 R9

To view router R9's BGP table, issue the following command. You should see something similar to the following:

```
R9#show ip bgp

BGP table version is 141, local router ID is 9.9.9.9

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *>   1.1.1.1/32       5.5.5.5                                0 4567 1 i

 *>   2.2.2.2/32       5.5.5.5                                0 4567 23 i

 *>   3.3.3.3/32       5.5.5.5                                0 4567 23 i

 *>   4.4.4.4/32       5.5.5.5                                0 4567 i

 r>   5.5.5.5/32       5.5.5.5                  0             0 4567 i

 *>   6.6.6.6/32       5.5.5.5                                0 4567 i
```

```
 *>   7.7.7.7/32       5.5.5.5                                      0 4567 i

 *>   9.9.9.9/32       0.0.0.0                  0         32768 i

 *>   10.0.0.1/32      5.5.5.5                                      0 4567 1 999
i

 *>   12.34.0.0/16     5.5.5.5                                      0 4567 1 999
?

 *>   23.45.0.0/16     5.5.5.5                                      0 4567 1 999
?

 *>   66.77.0.0/17     5.5.5.5                                      0 4567 1 999
?

 *>   89.100.0.0/18    5.5.5.5                                      0 4567 1 999
?

 *>   91.200.0.0/18    5.5.5.5                                      0 4567 1 999
?

 *>   102.64.0.0/18    5.5.5.5                                      0 4567 1 999
?

 *>   123.45.0.0/17    5.5.5.5                                      0 4567 1 999
?

 *>   130.25.0.0/18    5.5.5.5                                      0 4567 1 999
?

 *>   175.45.200.0/21  5.5.5.5                                      0 4567 1 999
?

 *>   183.77.220.0/22  5.5.5.5                                      0 4567 1 999
?

 *>   185.100.0.0/19   5.5.5.5                                      0 4567 1 999
?

 *>   190.30.128.0/17  5.5.5.5                                      0 4567 1 999
?
```

```
 *>   195.225.0.0/19   5.5.5.5                                0 4567 1 999
?

 *>   199.10.192.0/22  5.5.5.5                                0 4567 1 999
?

 *>   203.0.113.0/30   5.5.5.5                                0 4567 1 999
?

 *>   210.45.128.0/23  5.5.5.5                                0 4567 1 999
?

 *>   212.12.16.0/21   5.5.5.5                                0 4567 1 999
?

 *>   216.80.192.0/22  5.5.5.5                                0 4567 1 999
?

 *>   220.85.200.0/23  5.5.5.5                                0 4567 1 999
?

 *>   221.25.0.0/19    5.5.5.5                                0 4567 1 999
?
```

Note that:

- All loopback0 interface IPv4 addresses of all routers (including the ISP router) have been successfully learned via BGP, except for R8, which does not participate in BGP.
- The next hop IP address for all loopback0 addresses is R5, which is R9's only eBGP neighbor.
- The 5.5.5.5/32 network was learned via eBGP from R5 and is considered the best BGP route. However, it indicates a RIB failure with the "r" status code. This is because a static route to 5.5.5.5 exists within the routing table, superseding the Administrative Distance (AD) of the eBGP learned route; thus, it is not installed in the routing table.
- The 9.9.9.9/32 route has a next hop of 0.0.0.0, which indicates that it was learned via the local router. This has been given a default Weight of 32768 as expected, unlike the rest of the routes, which are given a Weight of 0.
- The AS Paths of each route are correct, based on the location of each individual router and the actual ASes that must be traversed to reach each route.

- All of the loopback addresses advertised by the ISP router are also found within the BGP table, all learned via the AS Path 4567 1 999. This is as expected because this is the shortest AS Path. They are also learned via another AS Path, which is 4567 23 1 999, but that path is longer.
- Even though the routes have been learned through an AS, which is a Confederation (4567), all internal AS numbers have been stripped from the AS Path.
- All of the loopback addresses advertised via the ISP router (except for the 10.0.0.1/32 address) have been learned via redistribution. That is why we see the "?" origin code at the end of each entry. The 10.0.0.1/32 route was injected into BGP using the `network` command, and thus, its origin code is "i".

Examine BGP Table IPv4 R6

To view router R6's BGP table, issue the following command. You should see something similar to the following:

```
R6#show ip bgp

BGP table version is 132, local router ID is 6.6.6.6

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 * i  1.1.1.1/32       5.5.5.5                  0    100      0 (45) 1 i

 *>                    4.4.4.4                  0    100      0 (45) 1 i
```

```
 * i  2.2.2.2/32       5.5.5.5                  0    100      0 (45) 23 i

 *>                    4.4.4.4                  0    100      0 (45) 23 i

 * i  3.3.3.3/32       5.5.5.5                  0    100      0 (45) 23 i

 *>                    4.4.4.4                  0    100      0 (45) 23 i

 r i  4.4.4.4/32       5.5.5.5                  0    100      0 (45) i

 r>                    4.4.4.4                  0    100      0 (45) i

 r i  5.5.5.5/32       5.5.5.5                  0    100      0 (45) i

 r>                    4.4.4.4                  0    100      0 (45) i

 *>   6.6.6.6/32       0.0.0.0                  0         32768 i

 r>i  7.7.7.7/32       7.7.7.7                  0    100      0 i

 * i  9.9.9.9/32       5.5.5.5                  0    100      0 (45)
4292935689 i

 *>                    4.4.4.4                  0    100      0 (45)
4292935689 i

 * i  10.0.0.1/32      5.5.5.5                  0    100      0 (45) 1 999
i

 *>                    4.4.4.4                  0    100      0 (45) 1 999
i

 * i  12.34.0.0/16     5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  23.45.0.0/16     5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?
```

```
 * i  66.77.0.0/17     5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  89.100.0.0/18    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  91.200.0.0/18    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  102.64.0.0/18    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  123.45.0.0/17    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  130.25.0.0/18    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  175.45.200.0/21  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  183.77.220.0/22  5.5.5.5                  0    100      0 (45) 1 999
?
```

```
 *>                 4.4.4.4                  0    100      0 (45) 1 999
?

 * i  185.100.0.0/19   5.5.5.5                  0    100      0 (45) 1 999
?

 *>                 4.4.4.4                  0    100      0 (45) 1 999
?

 * i  190.30.128.0/17  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                 4.4.4.4                  0    100      0 (45) 1 999
?

 * i  195.225.0.0/19   5.5.5.5                  0    100      0 (45) 1 999
?

 *>                 4.4.4.4                  0    100      0 (45) 1 999
?

 * i  199.10.192.0/22  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                 4.4.4.4                  0    100      0 (45) 1 999
?

 * i  203.0.113.0/30   5.5.5.5                  0    100      0 (45) 1 999
?

 *>                 4.4.4.4                  0    100      0 (45) 1 999
?

 * i  210.45.128.0/23  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                 4.4.4.4                  0    100      0 (45) 1 999
?

 * i  212.12.16.0/21   5.5.5.5                  0    100      0 (45) 1 999
?

 *>                 4.4.4.4                  0    100      0 (45) 1 999
?
```

```
 * i  216.80.192.0/22  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  220.85.200.0/23  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  221.25.0.0/19    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?
```

You can use the following command to see from which routers the routes are being learned:

```
R6#show ip bgp 1.1.1.1

BGP routing table entry for 1.1.1.1/32, version 110

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

  Advertised to update-groups:

     2

  Refresh Epoch 1

  (45) 1

    5.5.5.5 (metric 3) from 7.7.7.7 (7.7.7.7)

      Origin IGP, metric 0, localpref 100, valid, confed-internal

      rx pathid: 0, tx pathid: 0
```

```
  Refresh Epoch 1

  (45) 1

    4.4.4.4 (metric 2) from 4.4.4.4 (4.4.4.4)

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

      rx pathid: 0, tx pathid: 0x0
```

Note that:

- All loopback0 interface IPv4 addresses of all routers (including the ISP router) have been successfully learned via BGP, except for R8 which is not participating in BGP.
- All of these loopback0 interface addresses have been learned via eBGP and via iBGP except for the 6.6.6.6/32 and 7.7.7.7/32 networks which belong to the local sub-AS.
- Note that the eBGP learned routes are being learned from R4 in the neighboring sub-AS which the iBGP routes are being learned from R7.
- In all cases, the eBGP routes are preferred as the best.
- The next hop IP address for all of the loopback0 addresses is either R4 or R5, where in all cases, R4 is chosen as the preferred path.
- The IP addresses of R4. R5, and R7 are in the BGP table, but indicate a RIB failure with the “r” status code. This is because these routes are found within the routing table using a preferred routing source, and thus a RIB failure is indicated.
- The 9.9.9.9/32 route uses a 4-byte AS number as expected.
- The AS Paths of each route are correct, based on the location of each individual router and the actual ASes that must be traversed to reach each route.
- All of the loopback addresses advertised by the ISP router are also found within the BGP table, all learned via the AS Path (45) 1 999. This is as expected, because internally in the Confederation, the sub-AS 45 appears in parentheses.
- All of the loopback addresses advertised via the ISP router (except for the 10.0.0.1/32 address) have been learned via redistribution. That is why we see the “?” origin code at the end of each entry. The 10.0.0.1/32 route was injected into BGP using the `network` command, and thus, its origin code is “i”.

Examine BGP Table IPv4 R2

To view router R2’s BGP table, issue the following command. You should see something similar to the following:

```
R2#show ip bgp

BGP table version is 208, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *>   1.1.1.1/32       192.168.12.1             0             0 1 i

 * i                   3.3.3.3                  0    100      0 1 i

 *                     192.168.24.4                           0 4567 1 i

 *>   2.2.2.2/32       0.0.0.0                  0         32768 i

 r>i  3.3.3.3/32       3.3.3.3                  0    100      0 i

 *    4.4.4.4/32       192.168.12.1                           0 1 4567 i

 * i                   3.3.3.3                  0    100      0 4567 i

 *>                    192.168.24.4             0             0 4567 i

 *    5.5.5.5/32       192.168.12.1                           0 1 4567 i

 * i                   3.3.3.3                  0    100      0 4567 i
```

```
 *>                     192.168.24.4                                0 4567 i

 *    6.6.6.6/32        192.168.12.1                                0 1 4567 i

 * i                    3.3.3.3                  0    100          0 4567 i

 *>                     192.168.24.4                                0 4567 i

 *    7.7.7.7/32        192.168.12.1                                0 1 4567 i

 * i                    3.3.3.3                  0    100          0 4567 i

 *>                     192.168.24.4                                0 4567 i

 *    9.9.9.9/32        192.168.12.1                                0 1 4567
4292935689 i

 *>                     192.168.24.4                                0 4567
4292935689 i

 * i                    3.3.3.3                  0    100          0 4567
4292935689 i

 *>   10.0.0.1/32       192.168.12.1                                0 1 999 i

 * i                    3.3.3.3                  0    100          0 1 999 i

 *                      192.168.24.4                                0 4567 1 999
i

 *>   12.34.0.0/16      192.168.12.1                                0 1 999 ?

 * i                    3.3.3.3                  0    100          0 1 999 ?

 *                      192.168.24.4                                0 4567 1 999
?

 *>   23.45.0.0/16      192.168.12.1                                0 1 999 ?

 * i                    3.3.3.3                  0    100          0 1 999 ?

 *                      192.168.24.4                                0 4567 1 999
?
```

```
 *>   66.77.0.0/17     192.168.12.1                           0 1 999 ?

 * i                   3.3.3.3                  0    100      0 1 999 ?

 *                     192.168.24.4                           0 4567 1 999
?

 *>   89.100.0.0/18    192.168.12.1                           0 1 999 ?

 * i                   3.3.3.3                  0    100      0 1 999 ?

 *                     192.168.24.4                           0 4567 1 999
?

 *>   91.200.0.0/18    192.168.12.1                           0 1 999 ?

 * i                   3.3.3.3                  0    100      0 1 999 ?

 *                     192.168.24.4                           0 4567 1 999
?

 *>   102.64.0.0/18    192.168.12.1                           0 1 999 ?

 * i                   3.3.3.3                  0    100      0 1 999 ?

 *                     192.168.24.4                           0 4567 1 999
?

 *>   123.45.0.0/17    192.168.12.1                           0 1 999 ?

 * i                   3.3.3.3                  0    100      0 1 999 ?

 *                     192.168.24.4                           0 4567 1 999
?

 *>   130.25.0.0/18    192.168.12.1                           0 1 999 ?

 * i                   3.3.3.3                  0    100      0 1 999 ?

 *                     192.168.24.4                           0 4567 1 999
?

 *>   175.45.200.0/21  192.168.12.1                           0 1 999 ?
```

```
 * i                 3.3.3.3                  0    100       0 1 999 ?
 *                   192.168.24.4                            0 4567 1 999
?
 *>   183.77.220.0/22  192.168.12.1                          0 1 999 ?
 * i                 3.3.3.3                  0    100       0 1 999 ?
 *                   192.168.24.4                            0 4567 1 999
?
 *>   185.100.0.0/19   192.168.12.1                          0 1 999 ?
 * i                 3.3.3.3                  0    100       0 1 999 ?
 *                   192.168.24.4                            0 4567 1 999
?
 *>   190.30.128.0/17  192.168.12.1                          0 1 999 ?
 * i                 3.3.3.3                  0    100       0 1 999 ?
 *                   192.168.24.4                            0 4567 1 999
?
 *>   195.225.0.0/19   192.168.12.1                          0 1 999 ?
 * i                 3.3.3.3                  0    100       0 1 999 ?
 *                   192.168.24.4                            0 4567 1 999
?
 *>   199.10.192.0/22  192.168.12.1                          0 1 999 ?
 * i                 3.3.3.3                  0    100       0 1 999 ?
 *                   192.168.24.4                            0 4567 1 999
?
 *>   203.0.113.0/30   192.168.12.1                          0 1 999 ?
 * i                 3.3.3.3                  0    100       0 1 999 ?
```

```
 *                    192.168.24.4                               0 4567 1 999
?

 *>   210.45.128.0/23  192.168.12.1                              0 1 999 ?

 * i                  3.3.3.3                  0    100        0 1 999 ?

 *                    192.168.24.4                               0 4567 1 999
?

 *>   212.12.16.0/21   192.168.12.1                              0 1 999 ?

 * i                  3.3.3.3                  0    100        0 1 999 ?

 *                    192.168.24.4                               0 4567 1 999
?

 *>   216.80.192.0/22  192.168.12.1                              0 1 999 ?

 * i                  3.3.3.3                  0    100        0 1 999 ?

 *                    192.168.24.4                               0 4567 1 999
?

 *>   220.85.200.0/23  192.168.12.1                              0 1 999 ?

 * i                  3.3.3.3                  0    100        0 1 999 ?

 *                    192.168.24.4                               0 4567 1 999
?

 *>   221.25.0.0/19    192.168.12.1                              0 1 999 ?

 * i                  3.3.3.3                  0    100        0 1 999 ?

 *                    192.168.24.4                               0 4567 1 999
?
```

Note that:

- Routes are learned from three sources:
    - One iBGP source which is R3.

    - Two eBGP sources which are R1 and R4.
- Except for the 3.3.3.3/32 network, all other networks choose the best path as the one learned via eBGP. And among the eBGP paths, the best path is the one with the shortest AS path.
- Much of the results that you can see are similar to what you saw in the previous routers.

Examine BGP Table IPv6 R2

To view router R2's BGP table containing IPv6 networks, issue the following command. You should see something similar to the following:

```
R2#show bgp ipv6 unicast

BGP table version is 1, local router ID is 2.2.2.2

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *    2001:DB8:1::1/128

                       ::FFFF:192.168.12.1

                                                            0 1 999 i


 *    2001:DB8:12:34::/64
```

```
                      ::FFFF:192.168.12.1
                                                    0 1 999 i
*    2001:DB8:23:45::/64
                      ::FFFF:192.168.12.1
                                                    0 1 999 i
*    2001:DB8:66:77::/64
                      ::FFFF:192.168.12.1
                                                    0 1 999 i
*    2001:DB8:89:100::/64
                      ::FFFF:192.168.12.1
                                                    0 1 999 i
*    2001:DB8:91:200::/64
                      ::FFFF:192.168.12.1
                                                    0 1 999 i
*    2001:DB8:102:64::/64
                      ::FFFF:192.168.12.1
                                                    0 1 999 i
*    2001:DB8:123:45::/64
                      ::FFFF:192.168.12.1
                                                    0 1 999 i
*    2001:DB8:130:25::/64
```

```
                    ::FFFF:192.168.12.1
                                                0 1 999 i
*    2001:DB8:175:45::/64
                    ::FFFF:192.168.12.1
                                                0 1 999 i
*    2001:DB8:183:77::/64
                    ::FFFF:192.168.12.1
                                                0 1 999 i
*    2001:DB8:185:100::/64
                    ::FFFF:192.168.12.1
                                                0 1 999 i
*    2001:DB8:190:32::/64
                    ::FFFF:192.168.12.1
                                                0 1 999 i
*    2001:DB8:195:225::/64
                    ::FFFF:192.168.12.1
                                                0 1 999 i
*    2001:DB8:199:10::/64
                    ::FFFF:192.168.12.1
                                                0 1 999 i
*    2001:DB8:210:45::/64
```

```
                    ::FFFF:192.168.12.1
                                                   0 1 999 i
 *   2001:DB8:212:12::/64
                    ::FFFF:192.168.12.1
                                                   0 1 999 i
 *   2001:DB8:216:80::/64
                    ::FFFF:192.168.12.1
                                                   0 1 999 i
 *   2001:DB8:220:85::/64
                    ::FFFF:192.168.12.1
                                                   0 1 999 i
 *   2001:DB8:225:25::/64
                    ::FFFF:192.168.12.1
                                                   0 1 999 i
```

Note that:

- R2's BGP table contains all of the IPv6 routes created in the ISP router and the address of the Loopback0 interface on R1.
- These routes are given a next hop IP address of ::FFFF:192.168.12.1 which indicates they are learned via IPv4 from R1.
- The AS path for all these routes is considered valid, but it is not considered the "best" because the next hop IP is not reachable.
- All the routes are considered valid, but they are not considered the "best" because the next hop IP is not reachable.

Advanced BGP Features

Once you have configured the above and verified their correct implementation, the next step is to tweak and adjust additional features

BGP Auto Summary

To fulfill the requirements of the implementation of BGP Auto-Summary on R9, perform the following configurations:

```
R9(config)#router bgp 65505.9

R9(config-router)#auto-summary

R9(config-router)#no network 9.9.9.9 mask 255.255.255.255

R9(config-router)#network 9.0.0.0
```

To verify that the classful network has entered the routing table of R9, use the following verification command:

```
R9#show ip bgp 9.0.0.0

BGP routing table entry for 9.0.0.0/8, version 143

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

  Advertised to update-groups:

     2

  Refresh Epoch 1

  Local

    0.0.0.0 from 0.0.0.0 (9.9.9.9)

      Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced,
local, best

      rx pathid: 0, tx pathid: 0x0
```

As you can see, the network in the local routing table appears as a class A network. To examine how this network has been advertised to the rest of the network, take a look at the BGP tables of other routers. The following shows the BGP table of R2 and indicates how it perceives the network advertised from R9:

```
R2#show ip bgp 9.0.0.0

BGP routing table entry for 9.0.0.0/8, version 212

Paths: (3 available, best #2, table default)

  Advertised to update-groups:

     7          8

  Refresh Epoch 2

  1 4567 4292935689

    192.168.12.1 from 192.168.12.1 (1.1.1.1)

      Origin IGP, localpref 100, valid, external

      rx pathid: 0, tx pathid: 0

  Refresh Epoch 3

  4567 4292935689

    192.168.24.4 from 192.168.24.4 (4.4.4.4)

      Origin IGP, localpref 100, valid, external, best

      rx pathid: 0, tx pathid: 0x0

  Refresh Epoch 1

  4567 4292935689

    3.3.3.3 (metric 2) from 3.3.3.3 (3.3.3.3)
```

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

      rx pathid: 0, tx pathid: 0
```

You can check the BGP tables of other routers to see that the table entry has been installed in a classful manner.

BGP Summarization

To perform the required route summarization and fulfill the requirements of this task, perform the following configurations:

You are asked to summarize the networks found on Loopbacks 10, 11, and 12 of the ISP router:

- Loopback10 183.77.220.1
- Loopback11 185.100.0.1
- Loopback12 190.30.128.1

The summary address for these networks is 176.0.0.0/4. You can attempt to calculate it yourself as well.

The task requires that only the summary address be installed in the ISP router's local routing table and that all member routes of the summary be marked as "suppressed" routes in the BGP routing table.

To fulfill these requirements, do the following:

```
ISP(config)#router bgp 999

ISP(config-router)#address-family ipv4

ISP(config-router-af)#aggregate-address 176.0.0.0 240.0.0.0 summary-only
```

Remember that because the ISP router is configured with multiple address families, the command must be applied under the IPv4 address family configuration mode.
The `summary-only` keyword is used to ensure that the member prefixes are suppressed within the BGP table. To verify this, take a look at the BGP table of the ISP router like so:

```
ISP#show ip bgp

BGP table version is 45, local router ID is 221.25.0.1

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *>   1.1.1.1/32       203.0.113.1              0             0 1 i

 *>   2.2.2.2/32       203.0.113.1                            0 1 23 i

 *>   3.3.3.3/32       203.0.113.1                            0 1 23 i

 *>   4.4.4.4/32       203.0.113.1                            0 1 4567 i

 *>   5.5.5.5/32       203.0.113.1                            0 1 4567 i

 *>   6.6.6.6/32       203.0.113.1                            0 1 4567 i

 *>   7.7.7.7/32       203.0.113.1                            0 1 4567 i

 *>   9.0.0.0          203.0.113.1                            0 1 4567
4292935689 i

 *>   10.0.0.1/32      0.0.0.0                  0         32768 i

 *>   12.34.0.0/16     0.0.0.0                  0         32768 ?
```

```
*>    23.45.0.0/16      0.0.0.0                  0          32768 ?
*>    66.77.0.0/17      0.0.0.0                  0          32768 ?
*>    89.100.0.0/18     0.0.0.0                  0          32768 ?
*>    91.200.0.0/18     0.0.0.0                  0          32768 ?
*>    102.64.0.0/18     0.0.0.0                  0          32768 ?
*>    123.45.0.0/17     0.0.0.0                  0          32768 ?
*>    130.25.0.0/18     0.0.0.0                  0          32768 ?
*>    175.45.200.0/21   0.0.0.0                  0          32768 ?
*>    176.0.0.0/4       0.0.0.0                             32768 i
s>    183.77.220.0/22   0.0.0.0                  0          32768 ?
s>    185.100.0.0/19    0.0.0.0                  0          32768 ?
s>    190.30.128.0/17   0.0.0.0                  0          32768 ?
*>    195.225.0.0/19    0.0.0.0                  0          32768 ?
*>    199.10.192.0/22   0.0.0.0                  0          32768 ?
*>    203.0.113.0/30    0.0.0.0                  0          32768 ?
*>    210.45.128.0/23   0.0.0.0                  0          32768 ?
*>    212.12.16.0/21    0.0.0.0                  0          32768 ?
*>    216.80.192.0/22   0.0.0.0                  0          32768 ?
*>    220.85.200.0/23   0.0.0.0                  0          32768 ?
*>    221.25.0.0/19     0.0.0.0                  0          32768 ?
```

Notice that the summary route has been installed, and that the three member routes of that summary route have been marked with an “s” which means “suppressed”. Let’s also ensure these routes have not been further advertised to routers outside of AS 999. Let’s take a look at the BGP table of R1:

```
R1#show ip bgp

BGP table version is 47, 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, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *>   1.1.1.1/32       0.0.0.0                  0         32768 i

 *    2.2.2.2/32       192.168.12.2             0             0 23 i

 *                     192.168.14.4                           0 4567 23 i

 *>                    192.168.13.3                           0 23 i

 *    3.3.3.3/32       192.168.12.2                           0 23 i

 *                     192.168.14.4                           0 4567 23 i

 *>                    192.168.13.3             0             0 23 i
```

```
 *    4.4.4.4/32       192.168.12.2                            0 23 4567 i
 *                     192.168.13.3                            0 23 4567 i
 *>                    192.168.14.4             0              0 4567 i
 *    5.5.5.5/32       192.168.12.2                            0 23 4567 i
 *                     192.168.13.3                            0 23 4567 i
 *>                    192.168.14.4                            0 4567 i
 *    6.6.6.6/32       192.168.12.2                            0 23 4567 i
 *                     192.168.13.3                            0 23 4567 i
 *>                    192.168.14.4                            0 4567 i
 *    7.7.7.7/32       192.168.12.2                            0 23 4567 i
 *                     192.168.13.3                            0 23 4567 i
 *>                    192.168.14.4                            0 4567 i
 *    9.0.0.0          192.168.13.3                            0 23 4567
4292935689 i
 *                     192.168.12.2                            0 23 4567
4292935689 i
 *>                    192.168.14.4                            0 4567
4292935689 i
 *>   10.0.0.1/32      203.0.113.2              0              0 999 i
 *>   12.34.0.0/16     203.0.113.2              0              0 999 ?
 *>   23.45.0.0/16     203.0.113.2              0              0 999 ?
 *>   66.77.0.0/17     203.0.113.2              0              0 999 ?
 *>   89.100.0.0/18    203.0.113.2              0              0 999 ?
```

```
*>  91.200.0.0/18    203.0.113.2              0             0 999 ?
*>  102.64.0.0/18    203.0.113.2              0             0 999 ?
*>  123.45.0.0/17    203.0.113.2              0             0 999 ?
*>  130.25.0.0/18    203.0.113.2              0             0 999 ?
*>  175.45.200.0/21  203.0.113.2              0             0 999 ?
*>  176.0.0.0/4      203.0.113.2              0             0 999 i
*>  195.225.0.0/19   203.0.113.2              0             0 999 ?
*>  199.10.192.0/22  203.0.113.2              0             0 999 ?
r>  203.0.113.0/30   203.0.113.2              0             0 999 ?
*>  210.45.128.0/23  203.0.113.2              0             0 999 ?
*>  212.12.16.0/21   203.0.113.2              0             0 999 ?
*>  216.80.192.0/22  203.0.113.2              0             0 999 ?
*>  220.85.200.0/23  203.0.113.2              0             0 999 ?
*>  221.25.0.0/19    203.0.113.2              0             0 999 ?
```

Notice that the summary route can be found in the BGP table of R1, but the member routes cannot be found. You can check other routers within the network as well to ensure that the summary route has been propagated throughout the network.

BGP Weight Attribute

To ensure that the 12.34.0.0/16 network is routed from R4 via AS 23 instead of AS 1, you must modify the [Weight attribute]() on R4 using a route map.

To check the current best path for this network, take a look at the BGP table of R4 for this route:

```
R4#show ip bgp 12.34.0.0

BGP routing table entry for 12.34.0.0/16, version 3

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

  Advertised to update-groups:

     6           7           10

  Refresh Epoch 5

  23 1 999

    192.168.24.2 from 192.168.24.2 (2.2.2.2)

      Origin incomplete, localpref 100, valid, external

      rx pathid: 0, tx pathid: 0

  Refresh Epoch 1

  1 999

    192.168.14.1 from 192.168.14.1 (1.1.1.1)

      Origin incomplete, localpref 100, valid, external, best

      rx pathid: 0, tx pathid: 0x0
```

As you can see, there are two routes to the destination from R4, via R1 and via R2. Via R1 is chosen as the best path because the AS Path Length is shorter.

To use the Weight attribute to change this behavior as the task describes, issue the following commands. First, configure the route map that will modify the weight of the route that matches ACL 1, and will leave all other routes unmodified:

```
R4(config)#route-map SETWEIGHT permit 10

R4(config-route-map)#match ip address 1
```

```
R4(config-route-map)#set weight 500

R4(config-route-map)#route-map SETWEIGHT permit 20
```

Next, configure the access list that will match the route in question:

```
R4(config)#access-list 1 permit 12.34.0.0 0.0.255.255
```

Finally, apply the route map in an inbound direction on the eBGP peer of R2:

```
R4(config)#router bgp 45

R4(config-router)#neighbor 192.168.24.2 route-map SETWEIGHT in
```

The changes will take some time to take effect. To speed up the process, use the following command on R4:

```
R4#clear ip bgp *
```

After a few moments, you can check to see the results of the change using the following command:

```
R4#show ip bgp 12.34.0.0

BGP routing table entry for 12.34.0.0/16, version 11

Paths: (3 available, best #3, table default)

  Advertised to update-groups:

     20          21          22

  Refresh Epoch 2

  1 999

    192.168.14.1 from 192.168.14.1 (1.1.1.1)
```

```
      Origin incomplete, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  23 1 999
    5.5.5.5 (metric 2) from 5.5.5.5 (5.5.5.5)
      Origin incomplete, metric 0, localpref 100, valid, confed-internal
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 3
  23 1 999
    192.168.24.2 from 192.168.24.2 (2.2.2.2)
      Origin incomplete, localpref 100, weight 500, valid, external, best
      rx pathid: 0, tx pathid: 0x0
```

You can see that the best path is now the one via R2, with a weight of 500.

BGP Local Preference Attribute

To ensure that BGP operates such that packets destined for the 23.45.0.0/16 network from within AS 23 will always exit that AS via the R2 router by modifying the Local Preference attribute, perform the following. First, configure the route map that will modify the Local Preference of the route that matches ACL 1, and will leave all other routes unmodified:

```
R2(config)#route-map LOCALPREF permit 10
R2(config-route-map)#match ip address 1
R2(config-route-map)#set local-preference 750
```

```
R2(config-route-map)#route-map LOCALPREF permit 20
```

Next, configure the access list that will match the route in question:

```
R2(config)#access-list 1 permit 23.45.0.0 0.0.255.255
```

Finally, apply the route map in an inbound direction on the eBGP peer of R1. Remember, R2 has an IPv4 address family within which the command must be applied:

```
R2(config)#router bgp 23

R2(config-router)#address-family ipv4

R2(config-router-af)#neighbor 192.168.12.1 route

R2(config-router-af)#neighbor 192.168.12.1 route-map LOCALPREF in
```

The changes will take some time to take effect. To speed up the process, use the following command on R2:

```
R2#clear ip bgp *
```

After a few moments, you can check to see the results of the change using the following command:

```
R2#show ip bgp 23.45.0.0

BGP routing table entry for 23.45.0.0/16, version 11

Paths: (2 available, best #1, table default)

  Advertised to update-groups:

     9          10

  Refresh Epoch 2
```

```
  1 999
    192.168.12.1 from 192.168.12.1 (1.1.1.1)
      Origin incomplete, localpref 750, valid, external, best
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 2
  4567 1 999
    192.168.24.4 from 192.168.24.4 (4.4.4.4)
      Origin incomplete, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
```

As you can see, the Local Preference of the 23.45.0.0/16 route via R1 is set to 750, which is the best route. Another important check, however is to see how this has affected the BGP table of R3:

```
R3#show ip bgp 23.45.0.0
BGP routing table entry for 23.45.0.0/16, version 168
Paths: (3 available, best #1, table default)
  Advertised to update-groups:
     7
  Refresh Epoch 1
  1 999
    2.2.2.2 (metric 2) from 2.2.2.2 (2.2.2.2)
      Origin incomplete, metric 0, localpref 750, valid, internal, best
```

```
      rx pathid: 0, tx pathid: 0x0

  Refresh Epoch 1

  4567 1 999

    192.168.35.5 from 192.168.35.5 (5.5.5.5)

      Origin incomplete, localpref 100, valid, external

      rx pathid: 0, tx pathid: 0

  Refresh Epoch 2

  1 999

    192.168.13.1 from 192.168.13.1 (1.1.1.1)

      Origin incomplete, localpref 100, valid, external

      rx pathid: 0, tx pathid: 0
```

R3 has learned about this route from three sources, however, because of the value of the Local Preference for this route that is advertised from R2 via iBGP, the path via R2 is chosen as the best path. Thus the configuration of Local Preference has caused this route to exit AS 23 only via R2, which fulfills the requirement of the task.

BGP AS Path Prepending

In this task, you must use AS Path Prepending to ensure that traffic coming from R1 that is destined for the 9.0.0.0/8 network will be routed via AS 23 rather than via AS 4567. Initially, let's check to see what path is being used with the following command:

```
R1#show ip bgp 9.0.0.0

BGP routing table entry for 9.0.0.0/8, version 93

Paths: (3 available, best #2, table default)

  Advertised to update-groups:
```

```
    1
  Refresh Epoch 1
  23 4567 4292935689
    192.168.12.2 from 192.168.12.2 (2.2.2.2)
      Origin IGP, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  4567 4292935689
    192.168.14.4 from 192.168.14.4 (4.4.4.4)
      Origin IGP, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
  Refresh Epoch 2
  23 4567 4292935689
    192.168.13.3 from 192.168.13.3 (3.3.3.3)
      Origin IGP, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
```

As you can see, the best path to this destination from R1 is via R4.

To fulfill the requirements of this task by employing AS Path Prepending, do the following. First, configure the route map that will prepend the AS Path values to the route that matches ACL 2, and will leave all other routes unmodified:

```
R4(config)#route-map PREPEND permit 10
```

```
R4(config-route-map)#match ip address 2

R4(config-route-map)#set as-path prepend 4567 4567 4567 4567

R4(config-route-map)#route-map PREPEND permit 20
```

Next, configure the access list that will match the route in question:

```
R4(config)#access-list 2 permit 9.0.0.0 0.255.255.255
```

Finally, apply the route map in an outbound direction on the eBGP peering with R1:

```
R4(config)#router bgp 45

R4(config-router)#neighbor 192.168.14.1 route-map PREPEND out
```

As before, the changes will take some time to take effect. To speed up the process, use the following command on R4:

```
R4#clear ip bgp *
```

After a few moments, you can check to see the results of the change using the following command on R1:

```
R1#show ip bgp

BGP table version is 106, 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, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *>   1.1.1.1/32       0.0.0.0                  0         32768 i

 *    2.2.2.2/32       192.168.14.4                           0 4567 23 i

 *                     192.168.13.3                           0 23 i

 *>                    192.168.12.2             0             0 23 i

 *    3.3.3.3/32       192.168.14.4                           0 4567 23 i

 *                     192.168.12.2                           0 23 i

 *>                    192.168.13.3             0             0 23 i

 *    4.4.4.4/32       192.168.12.2                           0 23 4567 i

 *                     192.168.13.3                           0 23 4567 i

 *>                    192.168.14.4             0             0 4567 i

 *>   5.5.5.5/32       192.168.14.4                           0 4567 i

 *                     192.168.12.2                           0 23 4567 i

 *                     192.168.13.3                           0 23 4567 i

 *>   6.6.6.6/32       192.168.14.4                           0 4567 i

 *                     192.168.12.2                           0 23 4567 i

 *                     192.168.13.3                           0 23 4567 i

 *>   7.7.7.7/32       192.168.14.4                           0 4567 i
```

```
 *                      192.168.12.2                                0 23 4567 i
 *                      192.168.13.3                                0 23 4567 i
 *    9.0.0.0           192.168.14.4                                0 4567 4567
4567 4567 4567 4292935689 i
 *                      192.168.12.2                                0 23 4567
4292935689 i
 *>                     192.168.13.3                                0 23 4567
4292935689 i
 *>   10.0.0.1/32       203.0.113.2              0                  0 999 i
 *>   12.34.0.0/16      203.0.113.2              0                  0 999 ?
 *>   23.45.0.0/16      203.0.113.2              0                  0 999 ?
 *>   66.77.0.0/17      203.0.113.2              0                  0 999 ?
 *>   89.100.0.0/18     203.0.113.2              0                  0 999 ?
 *>   91.200.0.0/18     203.0.113.2              0                  0 999 ?
 *>   102.64.0.0/18     203.0.113.2              0                  0 999 ?
 *>   123.45.0.0/17     203.0.113.2              0                  0 999 ?
 *>   130.25.0.0/18     203.0.113.2              0                  0 999 ?
 *>   175.45.200.0/21   203.0.113.2              0                  0 999 ?
 *>   176.0.0.0/4       203.0.113.2              0                  0 999 i
 *>   195.225.0.0/19    203.0.113.2              0                  0 999 ?
 *>   199.10.192.0/22   203.0.113.2              0                  0 999 ?
 r>   203.0.113.0/30    203.0.113.2              0                  0 999 ?
 *>   210.45.128.0/23   203.0.113.2              0                  0 999 ?
```

```
 *>   212.12.16.0/21   203.0.113.2              0             0 999 ?

 *>   216.80.192.0/22  203.0.113.2              0             0 999 ?

 *>   220.85.200.0/23  203.0.113.2              0             0 999 ?

 *>   221.25.0.0/19    203.0.113.2              0             0 999 ?
```

As you can see, the 9.0.0.0/8 route has been learned by R1 via three sources. The 4567 AS has been prepended four times due to the route map that was applied on R4, resulting in an AS Path length of 6, which is double that indicated via AS 23, which was a requirement of the task.

BGP Origin Code Attribute

In this task, you are asked to make a modification on the ISP router that will change the origin code of the 66.77.0.0/17 network. Specifically, the origin code for this route should be preferable to that given to a route that has not been redistributed into BGP. To see the current origin code of the route, take a look at the following command:

```
R1#show ip bgp 66.77.0.0

BGP routing table entry for 66.77.0.0/17, version 5

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

  Advertised to update-groups:

     1

  Refresh Epoch 1

  999

    203.0.113.2 from 203.0.113.2 (221.25.0.1)

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

      rx pathid: 0, tx pathid: 0x0
```

As you can see, the origin code for this route, as it has been advertised to R1, is "incomplete" which simply appears as a "?" at the end of the entry in the BGP table. This simply means that the route has been injected into BGP using redistribution. To change this to a code that is preferable to the "?" origin code, do the following:

```
ISP(config)#router bgp 999

ISP(config-router)#address-family ipv4

ISP(config-router-af)#network 66.77.0.0 mask 255.255.128.0
```

Remember, the ISP has the IPv4 address family configured, so you must apply this command under the address family configuration mode.

Let's reexamine the BGP table of R1 for this route:

```
R1#show ip bgp 66.77.0.0

BGP routing table entry for 66.77.0.0/17, version 107

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

  Advertised to update-groups:

     1

  Refresh Epoch 1

  999

    203.0.113.2 from 203.0.113.2 (221.25.0.1)

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

      rx pathid: 0, tx pathid: 0x0
```

You can see that the origin code has now changed to IGP, which is preferred over incomplete. This appears as an "i" at the end of the entry within the BGP table. Examine the BGP tables of other routers in the network to confirm that the origin code remains the same throughout. Note the BGP table of R9 below:

```
R9#show ip bgp

BGP table version is 366, local router ID is 9.9.9.9

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *>   1.1.1.1/32       5.5.5.5                                  0 4567 1 i

 *>   2.2.2.2/32       5.5.5.5                                  0 4567 23 i

 *>   3.3.3.3/32       5.5.5.5                                  0 4567 23 i

 *>   4.4.4.4/32       5.5.5.5                                  0 4567 i

 r>   5.5.5.5/32       5.5.5.5                  0              0 4567 i

 *>   6.6.6.6/32       5.5.5.5                                  0 4567 i

 *>   7.7.7.7/32       5.5.5.5                                  0 4567 i

 *>   9.0.0.0          0.0.0.0                  0          32768 i
```

```
 *>   10.0.0.1/32      5.5.5.5                                0 4567 1 999
i

 *>   12.34.0.0/16     5.5.5.5                                0 4567 23 1
999 ?

 *>   23.45.0.0/16     5.5.5.5                                0 4567 1 999
?

 *>   66.77.0.0/17     5.5.5.5                                0 4567 1 999
i

 *>   89.100.0.0/18    5.5.5.5                                0 4567 1 999
?

 *>   91.200.0.0/18    5.5.5.5                                0 4567 1 999
?

 *>   102.64.0.0/18    5.5.5.5                                0 4567 1 999
?

 *>   123.45.0.0/17    5.5.5.5                                0 4567 1 999
?

 *>   130.25.0.0/18    5.5.5.5                                0 4567 1 999
?

 *>   175.45.200.0/21  5.5.5.5                                0 4567 1 999
?

 *>   176.0.0.0/4      5.5.5.5                                0 4567 1 999
i

 *>   195.225.0.0/19   5.5.5.5                                0 4567 1 999
?

 *>   199.10.192.0/22  5.5.5.5                                0 4567 1 999
?

 *>   203.0.113.0/30   5.5.5.5                                0 4567 1 999
?

 *>   210.45.128.0/23  5.5.5.5                                0 4567 1 999
?
```

```
 *>   212.12.16.0/21   5.5.5.5                                0 4567 1 999
?

 *>   216.80.192.0/22  5.5.5.5                                0 4567 1 999
?

 *>   220.85.200.0/23  5.5.5.5                                0 4567 1 999
?

 *>   221.25.0.0/19    5.5.5.5                                0 4567 1 999
?
```

As you can see, the 66.77.0.0/17 route has an "i" at the end of the entry, which means the origin code is IGP, and the task is complete.

BGP MED Attribute

For this task, you are asked to configure MED within AS 23 to influence routing on R1 such that the path from R1 to the 9.0.0.0/8 network uses the path via R2. To achieve this, the MED value for this route must be modified using a route map on R3.

Initially, let's examine the path that R1 takes to reach the 9.0.0.0/8 network:

```
R1#show ip bgp 9.0.0.0

BGP routing table entry for 9.0.0.0/8, version 97

Paths: (3 available, best #3, table default)

  Advertised to update-groups:

     1

  Refresh Epoch 1

  4567 4567 4567 4567 4567 4292935689

    192.168.14.4 from 192.168.14.4 (4.4.4.4)

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

```
    rx pathid: 0, tx pathid: 0

Refresh Epoch 1

23 4567 4292935689

  192.168.12.2 from 192.168.12.2 (2.2.2.2)

    Origin IGP, localpref 100, valid, external

    rx pathid: 0, tx pathid: 0

Refresh Epoch 2

23 4567 4292935689

  192.168.13.3 from 192.168.13.3 (3.3.3.3)

    Origin IGP, localpref 100, valid, external, best

    rx pathid: 0, tx pathid: 0x0
```

As you can see, the path via R4 was not chosen because we performed AS Path Prepending to make it less attractive. The other two paths are via AS 23, and it is the path via R3 that is chosen as the best. Let's modify the MED to cause the path via R2 to be chosen as best. To do this, we will create a route map on R3 and apply it in an outbound direction on its BGP peering with R1 to increase the MED value of the route and make it less attractive.

First, configure the route map that will modify the MED value to the route that matches ACL 1, and will leave all other routes unmodified:

```
R3(config)#route-map MED permit 10

R3(config-route-map)#match ip address 1

R3(config-route-map)#set metric 800

R3(config-route-map)#route-map MED permit 20
```

Next, configure the access list that will match the route in question:

```
R3(config)#access-list 1 permit 9.0.0.0 0.255.255.255
```

Finally, apply the route map in an outbound direction on the eBGP peering with R1. Remember R3 has the address family configured for IPv4, so you must apply the route map under the address family configuration mode:

```
R3(config)#router bgp 23

R3(config-router)#address-family ipv4

R3(config-router-af)#neighbor 192.168.13.1 route-map MED out
```

As before, the changes will take some time to take effect. To speed up the process, use the following command on R3:

```
R3#clear ip bgp *
```

Now let's take a look at the BGP table on R1 to see if this has made a difference:

```
R1#show ip bgp 9.0.0.0

BGP routing table entry for 9.0.0.0/8, version 109

Paths: (3 available, best #3, table default)

  Advertised to update-groups:

     1

  Refresh Epoch 1

  23 4567 4292935689

    192.168.13.3 from 192.168.13.3 (3.3.3.3)
```

```
      Origin IGP, metric 800, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  4567 4567 4567 4567 4567 4292935689
    192.168.14.4 from 192.168.14.4 (4.4.4.4)
      Origin IGP, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  23 4567 4292935689
    192.168.12.2 from 192.168.12.2 (2.2.2.2)
      Origin IGP, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
```

From the above output, you can see that the best path was chosen via R2 since the path via R3 now has a metric or MED of 800. The task is complete.

BGP Prefer eBGP over iBGP

This task asks us to examine the BGP table on R3 for the 91.200.0.0/18 network, and to explore how eBGP routes are preferred over iBGP routes. To do so, issue the following command:

```
R3#show ip bgp 91.200.0.0
BGP routing table entry for 91.200.0.0/18, version 14
Paths: (3 available, best #3, table default)
  Advertised to update-groups:
```

```
    11          12
  Refresh Epoch 1
  1 999
    2.2.2.2 (metric 2) from 2.2.2.2 (2.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 2
  4567 1 999
    192.168.35.5 from 192.168.35.5 (5.5.5.5)
      Origin incomplete, localpref 100, valid, external
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 2
  1 999
    192.168.13.1 from 192.168.13.1 (1.1.1.1)
      Origin incomplete, localpref 100, valid, external, best
      rx pathid: 0, tx pathid: 0x0
```

From the above output, we can see three paths to this destination: Via the iBGP peer of R2, the eBGP peer of R1, and the eBGP peer of R5. Whether a route is learned via eBGP or iBGP can be determined by looking at the “internal” or “external” indicator.

Going through the BGP attributes, we see that the Weight, the Local Preference, and the Originate attributes are the same. However, the AS Path Length differs. We can see that the path via R5 has an AS Path length of 3, so that route is disregarded.

Now it's between the path via R1 and R2. The origin code is the same and the MED is the same, so we come to the comparison of eBGP and iBGP routes. We can see that the route via R1 is an eBGP-learned route, as it was learned from R1 which is an eBGP peer, while the route learned via R2 is an iBGP route, since R2 is an iBGP peer.

So, the eBGP-learned path was chosen as the best due to the eBGP over iBGP route tiebreaker.

BGP Communities

In this task, you are asked to use [BGP communities]() to be applied to specific routes using route maps. You will configure one route map that will be applied to the appropriate neighbors, modifying all communities with multiple statements.

Requirement prefix 102.64.0.0/18

The first part of this task is to prevent the 102.64.0.0/18 network from being advertised from R1 to R2, R3, or R4.

To achieve this, we can use the [No Advertise BGP community](). To do so, we can issue the following commands on R1. First, configure the route map that will modify the BGP community for the route that matches ACL 1, and will leave all other routes unmodified:

```
R1(config)#route-map COMMUNITIES permit 10

R1(config-route-map)#match ip address 1

R1(config-route-map)#set community no-advertise

R1(config-route-map)#route-map COMMUNITIES permit 20
```

Next, configure the access list that will match the route in question:

```
R1(config)#access-list 1 permit 102.64.0.0 0.0.63.255
```

Next, apply the route map in an outbound direction on the eBGP peerings with R2, R3, and R4. Remember R1 has the address family configured for IPv4, so you must apply the route maps under the address family configuration mode:

```
R1(config)#router bgp 1

R1(config-router)#address-family ipv4

R1(config-router-af)#neighbor 192.168.14.4 route-map COMMUNITIES out

R1(config-router-af)#neighbor 192.168.12.2 route-map COMMUNITIES out

R1(config-router-af)#neighbor 192.168.13.3 route-map COMMUNITIES out
```

Finally, we must enable the sending of BGP communities to these neighbors using the following commands:

```
R1(config-router-af)#neighbor 192.168.14.4 send-community

R1(config-router-af)#neighbor 192.168.12.2 send-community

R1(config-router-af)#neighbor 192.168.13.3 send-community
```

As before, the changes will take some time to take effect. To speed up the process, use the following command on R1:

```
R1#clear ip bgp *
```

You may need to apply this command to routers R2, R3, and R4 as well.

To verify that the community was applied correctly, issue the following commands:

```
R2#show ip bgp 102.64.0.0

BGP routing table entry for 102.64.0.0/18, version 29

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

```
  Not advertised to any peer

  Refresh Epoch 2

  1 999

    192.168.12.1 from 192.168.12.1 (1.1.1.1)

      Origin incomplete, localpref 100, valid, external, best

      Community: no-advertise

      rx pathid: 0, tx pathid: 0x0
```

As you can see, the BGP table in R2 states that this route is not advertised to any peer. The reason for this is the community of “no-advertise” that has been appended to the route. Also note that R2 has not learned about this route from any other source, such as R3 or R4. This indicates that those routers too have been informed of the no-advertise community and have not advertised the route to their neighbors. Let’s confirm this by looking at their BGP tables as well:

```
R3#show ip bgp 102.64.0.0

BGP routing table entry for 102.64.0.0/18, version 113

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

  Not advertised to any peer

  Refresh Epoch 1

  1 999

    192.168.13.1 from 192.168.13.1 (1.1.1.1)

      Origin incomplete, localpref 100, valid, external, best

      Community: no-advertise

      rx pathid: 0, tx pathid: 0x0
```

```
R4#show ip bgp 102.64.0.0

BGP routing table entry for 102.64.0.0/18, version 85

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

  Not advertised to any peer

  Refresh Epoch 1

  1 999

    192.168.14.1 from 192.168.14.1 (1.1.1.1)

      Origin incomplete, localpref 100, valid, external, best

      Community: no-advertise

      rx pathid: 0, tx pathid: 0x0
```

As you can see, both R3, which is an iBGP peer, and R4, which is an eBGP peer, have received the no-advertise BGP community for this route, and they do not advertise the network to any other peer. You can further confirm this by examining the BGP tables of the other routers in AS 4567 as well as that of R9.

Requirement prefix 123.45.0.0/17

The next task asks us that network 123.45.0.0/17 should be advertises from R1 to ASes 23 and 4567 but the eBGP peers should not propagate it further to other eBGP peers.

To achieve this, we can use the No Export BGP community. To do so, we can issue the following commands on the R1 router.

First, configure the route map that will modify the BGP community for the route that matches ACL 2, and will leave all other routes unmodified:

```
R1(config)#route-map COMMUNITIES permit 12

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

```
R1(config-route-map)#set community no-export
```

Next, configure the access list that will match the route in question:

```
R1(config)#access-list 2 permit 123.45.0.0 0.0.31.255
```

The route map is already applied in an outbound direction on the eBGP peerings with R2, R3, and R4. Similarly, the enabling of the ending of BGP communities to these neighbors has already been configured from the previous task. However, we must configure routers R2, R3, R4, and R5 to send the community attributes with their BGP updates to all their downstream BGP neighbors. To do this, apply the following commands on these routers:

```
R2(config)#router bgp 23

R2(config-router)#address-family ipv4

R2(config-router-af)#neighbor 3.3.3.3 send-community

R2(config-router-af)#neighbor 192.168.24.4 send-community

R3(config)#router bgp 23

R3(config-router)#address-family ipv4

R3(config-router-af)#neighbor 2.2.2.2 send-community

R3(config-router-af)#neighbor 192.168.35.5 send-community

R4(config)#router bgp 45

R4(config-router)#neighbor 5.5.5.5 send-community

R4(config-router)#neighbor 6.6.6.6 send-community

R5(config)#router bgp 45

R5(config-router)#neighbor 4.4.4.4 send-community
```

```
R5(config-router)#neighbor 7.7.7.7 send-community
```

As before, the changes will take some time to take effect. To speed up the process, use the following command on R1:

```
R1#clear ip bgp *
```

You may need to apply this command to routers R2, R3, and R4 as well.

To verify the correct behavior, examine the BGP tables of routers R2, R3, and R4:

```
R2#show ip bgp 123.45.0.0

BGP routing table entry for 123.45.0.0/17, version 128

Paths: (2 available, best #2, table default, not advertised to EBGP peer)

  Advertised to update-groups:

     13

  Refresh Epoch 1

  1 999

    3.3.3.3 (metric 2) from 3.3.3.3 (3.3.3.3)

      Origin incomplete, metric 0, localpref 100, valid, internal

      Community: no-export

      rx pathid: 0, tx pathid: 0

  Refresh Epoch 1

  1 999

    192.168.12.1 from 192.168.12.1 (1.1.1.1)
```

```
      Origin incomplete, localpref 100, valid, external, best

      Community: no-export

      rx pathid: 0, tx pathid: 0x0
```

As you can see on R2, the route learned via R1 has the no-export community. You can also see the statement “not advertised to EBGP peer”. However, you can also see that the route has been learned via R3 as well, and this route also has the no-export community set. So R3 which is an iBGP peer advertised the route to R2. Also note that R2 has no path to the destination via R4, which means that R4 has also been prevented from sharing this route to other eBGP peers.

```
R3#show ip bgp 123.45.0.0

BGP routing table entry for 123.45.0.0/17, version 194

Paths: (3 available, best #3, table default, not advertised to EBGP peer)

  Advertised to update-groups:

     14

  1 999

    2.2.2.2 (metric 2) from 2.2.2.2 (2.2.2.2)

      Origin incomplete, metric 0, localpref 100, valid, internal

      Community: no-export

      rx pathid: 0, tx pathid: 0

  Refresh Epoch 1

  1 999

    192.168.13.1 from 192.168.13.1 (1.1.1.1)

      Origin incomplete, localpref 100, valid, external, best
```

```
      Community: no-export

      rx pathid: 0, tx pathid: 0x0
```

On R3, you can see the statement “not advertised to EBGP peer” so this route is not advertised to other eBGP peers such as R5 for example. You can also see that the route has also been learned via R2, an iBGP peer, and it maintains the no-export community attribute.

```
R4#show ip bgp 123.45.0.0

BGP routing table entry for 123.45.0.0/17, version 113

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

  Advertised to update-groups:

     25          27

  Refresh Epoch 1

  1 999

    192.168.14.1 from 192.168.14.1 (1.1.1.1)

      Origin incomplete, localpref 100, valid, external, best

      Community: no-export

      rx pathid: 0, tx pathid: 0x0
```

Finally, we can see that this route has been learned by R4 via R1. It has not been learned via R2 because of the no-export community applied there. We can see the “not advertised to EBGP peer” confirming that the route was not readvertised to R2.

```
R5#show ip bgp 123.45.0.0

BGP routing table entry for 123.45.0.0/17, version 28
```

```
Paths: (1 available, best #1, table default, not advertised to EBGP peer)

Advertised to update-groups:

17

Refresh Epoch 1

1 999

4.4.4.4 (metric 2) from 4.4.4.4 (4.4.4.4)

Origin incomplete, metric 0, localpref 100, valid, confed-internal, best

 Community: no-export

rx pathid: 0, tx pathid: 0x0
```

As you can see, it has been advertised to R5 which is an iBGP peer, but with the no-export community intact.

```
R9#show ip bgp 123.45.0.0

% Network not in table
```

We can also confirm that the route was not further advertised to R9.

Requirement prefix 130.25.0.0/18

Ensure that the 130.25.0.0/18 network is advertised into the BGP confederation 4567 such that it remains within sub-AS 45, and is not further advertised into sub-AS 67 or beyond.

To achieve this, we can use the Local AS BGP community. There are several ways to achieve this, and each way may have a slightly different result. For this implementation, we will apply the local-AS community to the specific route in an outbound direction on router R1 in its peerings with R2, R3, and R4.

On R1, we will add an entry into the existing route map that will modify the BGP community for the route that matches ACL 3, and will leave all other routes unmodified:

```
R1(config)#route-map COMMUNITIES permit 14

R1(config-route-map)#match ip address 3

R1(config-route-map)#set community local-as
```

Next, configure the access list that will match the route in question:

```
R1(config)#access-list 3 permit 130.25.0.0 0.0.63.255
```

The route map is already applied in an outbound direction on the eBGP peerings with R2, R3, and R4. Similarly, the ability to send BGP communities to these neighbors has already been configured from the previous task. As before, the changes will take some time to take effect. To speed up the process, use the following command on R1:

```
R1#clear ip bgp *
```

You may need to apply this command to routers R2, R3, and R4 as well.

To verify the correct behavior, examine the BGP tables of routers R2, R3, and R4:

```
R2#show ip bgp 130.25.0.0

BGP routing table entry for 130.25.0.0/18, version 222

Paths: (2 available, best #2, table default, not advertised outside local AS)

  Advertised to update-groups:

     14

  Refresh Epoch 1
```

```
  1 999
    3.3.3.3 (metric 2) from 3.3.3.3 (3.3.3.3)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Community: local-AS
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 1
  1 999
    192.168.12.1 from 192.168.12.1 (1.1.1.1)
      Origin incomplete, localpref 100, valid, external, best
      Community: local-AS
      rx pathid: 0, tx pathid: 0x0
```

As you can see on R2, the route learned via R1 has the local-AS community. You can also see the statement "not advertised outside local AS". In addition, you see that the route has been learned via R3 as well with the local-AS community intact.

```
R3#show ip bgp 130.25.0.0
BGP routing table entry for 130.25.0.0/18, version 288
Paths: (2 available, best #2, table default, not advertised outside local AS)
  Advertised to update-groups:
     18
  Refresh Epoch 1
  1 999
```

```
    2.2.2.2 (metric 2) from 2.2.2.2 (2.2.2.2)

      Origin incomplete, metric 0, localpref 100, valid, internal

      Community: local-AS

      rx pathid: 0, tx pathid: 0

  Refresh Epoch 1

  1 999

    192.168.13.1 from 192.168.13.1 (1.1.1.1)

      Origin incomplete, localpref 100, valid, external, best

      Community: local-AS

      rx pathid: 0, tx pathid: 0x0
```

The output here is similar to that of R2. The route learned via R1 has the local-AS community. You can also see the statement "not advertised outside local AS". In addition, you see that the route has been learned via R2 as well with the local-AS community intact.

From the output on R2 and R3, we can conclude that AS 23 will not advertise this route to any eBGP neighbor. Thus, this route will only enter AS 4567 via R4's peering with R1. Let's take a look at R4:

```
R4#show ip bgp 130.25.0.0

BGP routing table entry for 130.25.0.0/18, version 155

Paths: (1 available, best #1, table default, not advertised outside local AS)

  Advertised to update-groups:

     33
```

```
  Refresh Epoch 1

  1 999

    192.168.14.1 from 192.168.14.1 (1.1.1.1)

      Origin incomplete, localpref 100, valid, external, best

      Community: local-AS

      rx pathid: 0, tx pathid: 0x0
```

Finally, we can see that this route has been learned by R4 via R1. It has not been learned via R2 because of the local-AS community applied to the routers in AS 23. And we can see the "not advertised outside local AS" confirming that the route was not readvertised to R2, nor was it advertised to any other sub-AS within the confederation. Let's confirm this:

```
R6#show ip bgp 130.25.0.0

% Network not in table

R7#show ip bgp 130.25.0.0

% Network not in table

R9#show ip bgp 130.25.0.0

% Network not in table
```

The task is complete.

BGP Route Filtering

For this task, you are asked to filter out all networks with a /18 prefix length from being advertised to router R9. This will be achieved by applying prefix lists on R5. First, we must create a prefix list that matches all prefixes of exactly /18 on router R5:

```
R5(config)#ip prefix-list BLOCK-18 deny 0.0.0.0/0 ge 18 le 18
```

```
R5(config)#ip prefix-list BLOCK-18 permit 0.0.0.0/0 le 32
```

Then we must apply this in an outbound direction to R5's eBGP peering with R9 like so:

```
R5(config)#router bgp 45

R5(config-router)#neighbor 9.9.9.9 prefix-list BLOCK-18 out
```

Let's see what effect this has on the BGP table of R9:

```
R9#show ip bgp

BGP table version is 543, local router ID is 9.9.9.9

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *>   1.1.1.1/32       5.5.5.5                                0 4567 1 i

 *>   2.2.2.2/32       5.5.5.5                                0 4567 23 i

 *>   3.3.3.3/32       5.5.5.5                                0 4567 23 i

 *>   4.4.4.4/32       5.5.5.5                                0 4567 i
```

```
 r>   5.5.5.5/32       5.5.5.5                  0             0 4567 i

 *>   6.6.6.6/32       5.5.5.5                                0 4567 i

 *>   7.7.7.7/32       5.5.5.5                                0 4567 i

 *>   9.0.0.0          0.0.0.0                  0         32768 i

 *>   10.0.0.1/32      5.5.5.5                                0 4567 1 999
i

 *>   12.34.0.0/16     5.5.5.5                                0 4567 23 1
999 ?

 *>   23.45.0.0/16     5.5.5.5                                0 4567 1 999
?

 *>   66.77.0.0/17     5.5.5.5                                0 4567 1 999
i

 *>   175.45.200.0/21  5.5.5.5                                0 4567 1 999
?

 *>   176.0.0.0/4      5.5.5.5                                0 4567 1 999
i

 *>   195.225.0.0/19   5.5.5.5                                0 4567 1 999
?

 *>   199.10.192.0/22  5.5.5.5                                0 4567 1 999
?

 *>   203.0.113.0/30   5.5.5.5                                0 4567 1 999
?

 *>   210.45.128.0/23  5.5.5.5                                0 4567 1 999
?

 *>   212.12.16.0/21   5.5.5.5                                0 4567 1 999
?

 *>   216.80.192.0/22  5.5.5.5                                0 4567 1 999
?
```

```
 *>   220.85.200.0/23  5.5.5.5                                0 4567 1 999
?

 *>   221.25.0.0/19    5.5.5.5                                0 4567 1 999
?
```

Notice that there are no prefixes in the BGP table with a prefix length of /18. The task is complete.

For the next task, we need to filter prefixes ranging from /22 to /32.

This must be applied using extended access lists on R9. First, we must create an extended access list that will filter out any prefixes that have a subnet ranging from /22 to /32. This can be done using the following command on R9:

```
R9(config)#ip access-list extended 110

R9(config-ext-nacl)#deny ip 0.0.0.0 255.255.255.255 255.255.252.0
0.0.3.255

R9(config-ext-nacl)#permit ip any any
```

Next, the ACL should be applied as a distribute list in an inbound direction within the BGP configuration of R9:

```
R9(config)#router bgp 65505.9

R9(config-router)#distribute-list 110 in
```

The changes will take some time to take effect. To speed up the process, use the following command on R9:

```
R9#clear ip bgp *
```

Let’s see what effect this has on the BGP table of R9:

```
R9#show ip bgp
```

```
BGP table version is 15, local router ID is 9.9.9.9

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *>   9.0.0.0          0.0.0.0                  0         32768 i

 *>   12.34.0.0/16     5.5.5.5                                0 4567 23 1
999 ?

 *>   23.45.0.0/16     5.5.5.5                                0 4567 1 999
?

 *>   66.77.0.0/17     5.5.5.5                                0 4567 1 999
i

 *>   175.45.200.0/21  5.5.5.5                                0 4567 1 999
?

 *>   176.0.0.0/4      5.5.5.5                                0 4567 1 999
i

 *>   192.168.23.0     5.5.5.5                                0 4567 23 i

 *>   195.225.0.0/19   5.5.5.5                                0 4567 1 999
?
```

```
 *>   212.12.16.0/21   5.5.5.5                                0 4567 1 999
?

 *>   221.25.0.0/19    5.5.5.5                                0 4567 1 999
?
```

Notice that all prefixes ranging from /22 to /32 have been removed from the BGP table. The task is complete.

BGP Transit AS

For this task, you are asked to ensure that AS 23 will never be used as a transit AS for the 175.45.200.0/21 network using distribute list filtering and a corresponding access list. We will prevent R2 and R3 in AS 23 from advertising this network to R4 and R5 in AS 4567. Thus, the route will enter AS 4567 only via the R1 –> R4 peering.

To do so, let's issue the following commands. First, we'll create an access list to deny the network in question, and allow all other networks:

```
R2(config)#ip access-list standard NO-TRANSIT

R2(config-std-nacl)#deny 175.45.200.0 0.0.7.255

R2(config-std-nacl)#permit any
```

Next, we apply that access list using a distribute list on the peering with R4 in an outbound direction:

```
R2(config)#router bgp 23

R2(config-router)#address-family ipv4

R2(config-router-af)#neighbor 192.168.24.4 distribute-list NO-TRANSIT out
```

Lets do the same on R3 with its peering with R5:

```
R3(config)#ip access-list standard NO-TRANSIT
```

```
R3(config-std-nacl)#deny 175.45.200.0 0.0.7.255

R3(config-std-nacl)#permit any

R3(config)#router bgp 23

R3(config-router)#address-family ipv4

R3(config-router-af)#neighbor 192.168.35.5 distribute-list NO-TRANSIT out
```

Now let's verify that AS 4567 has learned the 175.45.200.0/21 network only via R4's peering with R1, and there is no alternative route via AS 23:

```
R4#show ip bgp 175.45.200.0

BGP routing table entry for 175.45.200.0/21, version 161

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

  Advertised to update-groups:

     29          33          34

  Refresh Epoch 1

  1 999

    192.168.14.1 from 192.168.14.1 (1.1.1.1)

      Origin incomplete, localpref 100, valid, external, best

      rx pathid: 0, tx pathid: 0x0

R5#show ip bgp 175.45.200.0

BGP routing table entry for 175.45.200.0/21, version 17

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

  Advertised to update-groups:
```

```
     14          17          18

  Refresh Epoch 1

  1 999

    4.4.4.4 (metric 2) from 4.4.4.4 (4.4.4.4)

      Origin incomplete, metric 0, localpref 100, valid, confed-internal,
best

      rx pathid: 0, tx pathid: 0x0
```

As you can see from both R4 and R5, only the path via R1 has been learned. Thus, even in the event of a failure, AS 23 will never be used as a transit AS to route traffic to this destination.

BGP AS Path Filter

For this task, you are asked to use AS Path Filtering using BGP regular expressions to perform two specific route filtering objectives.

First, you are asked to ensure that routes that pass through AS 23 are prevented from entering the BGP table of R6. Let's examine the routes that R6 contains that include AS 23:

```
R6#show ip bgp

BGP table version is 774, local router ID is 6.6.6.6

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 * i  1.1.1.1/32       5.5.5.5                  0    100      0 (45) 1 i

 *>                    4.4.4.4                  0    100      0 (45) 1 i

 * i  2.2.2.2/32       5.5.5.5                  0    100      0 (45) 23 i

 *>                    4.4.4.4                  0    100      0 (45) 23 i

 * i  3.3.3.3/32       5.5.5.5                  0    100      0 (45) 23 i

 *>                    4.4.4.4                  0    100      0 (45) 23 i

 r i  4.4.4.4/32       5.5.5.5                  0    100      0 (45) i

 r>                    4.4.4.4                  0    100      0 (45) i

 r i  5.5.5.5/32       5.5.5.5                  0    100      0 (45) i

 r>                    4.4.4.4                  0    100      0 (45) i

 *>   6.6.6.6/32       0.0.0.0                  0         32768 i

 r>i  7.7.7.7/32       7.7.7.7                  0    100      0 i

 *>   9.0.0.0          4.4.4.4                  0    100      0 (45)
4292935689 i

 * i                   5.5.5.5                  0    100      0 (45)
4292935689 i

 * i  10.0.0.1/32      5.5.5.5                  0    100      0 (45) 1 999
i
```

```
 *>                  4.4.4.4                  0    100      0 (45) 1 999
i

 * i  12.34.0.0/16     5.5.5.5                  0    100      0 (45) 23 1
999 ?

 *>                  4.4.4.4                  0    100      0 (45) 23 1
999 ?

 * i  23.45.0.0/16     5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?

 * i  66.77.0.0/17     5.5.5.5                  0    100      0 (45) 1 999
i

 *>                  4.4.4.4                  0    100      0 (45) 1 999
i

 * i  89.100.0.0/18    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?

 * i  91.200.0.0/18    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?

 * i  123.45.0.0/17    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?

 * i  175.45.200.0/21  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?
```

```
 * i  176.0.0.0/4      5.5.5.5                  0    100      0 (45) 1 999
i
 *>                    4.4.4.4                  0    100      0 (45) 1 999
i
 * i  195.225.0.0/19   5.5.5.5                  0    100      0 (45) 1 999
?
 *>                    4.4.4.4                  0    100      0 (45) 1 999
?
 * i  199.10.192.0/22  5.5.5.5                  0    100      0 (45) 1 999
?
 *>                    4.4.4.4                  0    100      0 (45) 1 999
?
 * i  203.0.113.0/30   5.5.5.5                  0    100      0 (45) 1 999
?
 *>                    4.4.4.4                  0    100      0 (45) 1 999
?
 * i  210.45.128.0/23  5.5.5.5                  0    100      0 (45) 1 999
?
 *>                    4.4.4.4                  0    100      0 (45) 1 999
?
 * i  212.12.16.0/21   5.5.5.5                  0    100      0 (45) 1 999
?
 *>                    4.4.4.4                  0    100      0 (45) 1 999
?
 * i  216.80.192.0/22  5.5.5.5                  0    100      0 (45) 1 999
?
 *>                    4.4.4.4                  0    100      0 (45) 1 999
?
 * i  220.85.200.0/23  5.5.5.5                  0    100      0 (45) 1 999
?
```

```
 *>                4.4.4.4                  0    100      0 (45) 1 999
?

 * i  221.25.0.0/19    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                4.4.4.4                  0    100      0 (45) 1 999
?
```

Above, you can see the routes that contain AS 23 in their AS path. To filter those out, issue the following commands. First, you must create an AS Path access list that will deny any AS Path that contains the AS 23 and permits everything else:

```
R6(config)#ip as-path access-list 1 deny _23_

R6(config)#ip as-path access-list 1 permit .*
```

Next, create a route map that matches routes based on that AS Path access list:

```
R6(config)#route-map AS_PATH_FILTER permit 10

R6(config-route-map)#match as-path 1
```

Finally, the route map can be applied to the peerings with R4 and R7 in an inbound direction:

```
R6(config)#router bgp 67

R6(config-router)#neighbor 4.4.4.4 route-map AS_PATH_FILTER in

R6(config-router)#neighbor 7.7.7.7 route-map AS_PATH_FILTER in
```

The changes may take some time to propagate, so as before, to speed up the process, you can use the following command on R6:

```
R6#clear ip bgp *
```

Let's see how the BGP table of R6 has been affected:

```
R6#show ip bgp

BGP table version is 777, local router ID is 6.6.6.6

Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,

              r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 * i  1.1.1.1/32       5.5.5.5                  0    100      0 (45) 1 i

 *>                    4.4.4.4                  0    100      0 (45) 1 i

 r i  4.4.4.4/32       5.5.5.5                  0    100      0 (45) i

 r>                    4.4.4.4                  0    100      0 (45) i

 r i  5.5.5.5/32       5.5.5.5                  0    100      0 (45) i

 r>                    4.4.4.4                  0    100      0 (45) i

 *>   6.6.6.6/32       0.0.0.0                  0         32768 i

 r>i  7.7.7.7/32       7.7.7.7                  0    100      0 i
```

```
 *>   9.0.0.0          4.4.4.4                  0    100      0 (45)
4292935689 i

 * i                   5.5.5.5                  0    100      0 (45)
4292935689 i

 * i  10.0.0.1/32      5.5.5.5                  0    100      0 (45) 1 999
i

 *>                    4.4.4.4                  0    100      0 (45) 1 999
i

 * i  23.45.0.0/16     5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  66.77.0.0/17     5.5.5.5                  0    100      0 (45) 1 999
i

 *>                    4.4.4.4                  0    100      0 (45) 1 999
i

 * i  89.100.0.0/18    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  91.200.0.0/18    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  123.45.0.0/17    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  175.45.200.0/21  5.5.5.5                  0    100      0 (45) 1 999
?
```

```
 *>                  4.4.4.4                  0    100      0 (45) 1 999
?

 * i  176.0.0.0/4      5.5.5.5                  0    100      0 (45) 1 999
i

 *>                  4.4.4.4                  0    100      0 (45) 1 999
i

 * i  195.225.0.0/19   5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?

 * i  199.10.192.0/22  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?

 * i  203.0.113.0/30   5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?

 * i  210.45.128.0/23  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?

 * i  212.12.16.0/21   5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?

 * i  216.80.192.0/22  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                  4.4.4.4                  0    100      0 (45) 1 999
?
```

```
 * i  220.85.200.0/23  5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?

 * i  221.25.0.0/19    5.5.5.5                  0    100      0 (45) 1 999
?

 *>                    4.4.4.4                  0    100      0 (45) 1 999
?
```

You can see that all entries with AS Paths containing 23 have been removed.

The second route filtering objective is to ensure that R3 does not contain routes that have passed through AS 4567, but routes that have originated from AS 4567 remain in the BGP table.

First, let's examine the BGP table of R3:

```
R3#show ip bgp

BGP table version is 319, 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, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path
```

```
 *    1.1.1.1/32       192.168.35.5                           0 4567 1 i
 * i                   2.2.2.2                  0    100      0 1 i
 *>                    192.168.13.1             0             0 1 i
 r>i  2.2.2.2/32       2.2.2.2                  0    100      0 i
 *>   3.3.3.3/32       0.0.0.0                  0         32768 i
 *>   4.4.4.4/32       192.168.35.5                           0 4567 i
 *                     192.168.13.1                           0 1 4567 i
 * i                   2.2.2.2                  0    100      0 4567 i
 *>   5.5.5.5/32       192.168.35.5             0             0 4567 i
 *                     192.168.13.1                           0 1 4567 i
 * i                   2.2.2.2                  0    100      0 4567 i
 *>   6.6.6.6/32       192.168.35.5                           0 4567 i
 *                     192.168.13.1                           0 1 4567 i
 * i                   2.2.2.2                  0    100      0 4567 i
 *>   7.7.7.7/32       192.168.35.5                           0 4567 i
 *                     192.168.13.1                           0 1 4567 i
 * i                   2.2.2.2                  0    100      0 4567 i
 * i  9.0.0.0          2.2.2.2                  0    100      0 4567
4292935689 i
 *>                    192.168.35.5                           0 4567
4292935689 i
 *    10.0.0.1/32      192.168.35.5                           0 4567 1 999
i
```

```
 * i                  2.2.2.2                  0    100      0 1 999 i
 *>                   192.168.13.1                             0 1 999 i
 * i  12.34.0.0/16    2.2.2.2                  0    100      0 1 999 ?
 *>                   192.168.13.1                             0 1 999 ?
 *    23.45.0.0/16    192.168.35.5                             0 4567 1 999
?
 *>i                  2.2.2.2                  0    750      0 1 999 ?
 *                    192.168.13.1                             0 1 999 ?
 *    66.77.0.0/17    192.168.35.5                             0 4567 1 999
i
 * i                  2.2.2.2                  0    100      0 1 999 i
 *>                   192.168.13.1                             0 1 999 i
 *    89.100.0.0/18   192.168.35.5                             0 4567 1 999
?
 * i                  2.2.2.2                  0    100      0 1 999 ?
 *>                   192.168.13.1                             0 1 999 ?
 *    91.200.0.0/18   192.168.35.5                             0 4567 1 999
?
 * i                  2.2.2.2                  0    100      0 1 999 ?
 *>                   192.168.13.1                             0 1 999 ?
 *>   102.64.0.0/18   192.168.13.1                             0 1 999 ?
 * i  123.45.0.0/17   2.2.2.2                  0    100      0 1 999 ?
 *>                   192.168.13.1                             0 1 999 ?
```

```
 * i  130.25.0.0/18    2.2.2.2                  0    100      0 1 999 ?

 *>                    192.168.13.1                           0 1 999 ?

 *    175.45.200.0/21  192.168.35.5                           0 4567 1 999
?

 * i                   2.2.2.2                  0    100      0 1 999 ?

 *>                    192.168.13.1                           0 1 999 ?

 *    176.0.0.0/4      192.168.35.5                           0 4567 1 999
i

 * i                   2.2.2.2                  0    100      0 1 999 i

 *>                    192.168.13.1                           0 1 999 i

 *    195.225.0.0/19   192.168.35.5                           0 4567 1 999
?

 * i                   2.2.2.2                  0    100      0 1 999 ?

 *>                    192.168.13.1                           0 1 999 ?

 *    199.10.192.0/22  192.168.35.5                           0 4567 1 999
?

 * i                   2.2.2.2                  0    100      0 1 999 ?

 *>                    192.168.13.1                           0 1 999 ?

 *    203.0.113.0/30   192.168.35.5                           0 4567 1 999
?

 * i                   2.2.2.2                  0    100      0 1 999 ?

 *>                    192.168.13.1                           0 1 999 ?

 *    210.45.128.0/23  192.168.35.5                           0 4567 1 999
?

 * i                   2.2.2.2                  0    100      0 1 999 ?
```

```
 *>                    192.168.13.1                               0 1 999 ?

 *     212.12.16.0/21   192.168.35.5                              0 4567 1 999
?

 * i                   2.2.2.2                 0     100        0 1 999 ?

 *>                    192.168.13.1                               0 1 999 ?

 *     216.80.192.0/22  192.168.35.5                              0 4567 1 999
?

 * i                   2.2.2.2                 0     100        0 1 999 ?

 *>                    192.168.13.1                               0 1 999 ?

 *     220.85.200.0/23  192.168.35.5                              0 4567 1 999
?

 * i                   2.2.2.2                 0     100        0 1 999 ?

 *>                    192.168.13.1                               0 1 999 ?

 *     221.25.0.0/19    192.168.35.5                              0 4567 1 999
?

 * i                   2.2.2.2                 0     100        0 1 999 ?

 *>                    192.168.13.1                               0 1 999 ?
```

The red entries are those that have originated from 4567. Wherever else the 4567 ASN appears the routes should be removed.

To achieve this, issue the following commands. First, create the AS Path access list that will filter the required ASNs:

```
R3(config)#ip as-path access-list 2 permit 4567$

R3(config)#ip as-path access-list 2 deny 4567_
```

```
R3(config)#ip as-path access-list 2 permit .*
```

Next, create a route map that calls this AS Path access list and matches it.

```
R3(config)#route-map AS_PATH_FILTER permit 10

R3(config-route-map)#match as-path 2
```

Finally, apply the route map to the neighbor peering with R5 in an inbound direction.

```
R3(config)#router bgp 23

R3(config-router)#address-family ipv4

R3(config-router-af)#neighbor 192.168.35.5 route-map AS_PATH_FILTER in
```

The changes may take some time to propagate, so as before, to speed up the process, you can use the following command on R3:

```
R3#clear ip bgp *
```

Let's see the changes to R3's BGP table:

```
R3#show ip bgp

BGP table version is 1, 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, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found


     Network          Next Hop            Metric LocPrf Weight Path

 *    1.1.1.1/32       192.168.13.1             0             0 1 i

 * i                   2.2.2.2                  0    100      0 1 i

 * i  2.2.2.2/32       2.2.2.2                  0    100      0 i

 *    3.3.3.3/32       0.0.0.0                  0         32768 i

 *>   4.4.4.4/32       192.168.35.5                           0 4567 i

 *                     192.168.13.1                           0 1 4567 i

 * i                   2.2.2.2                  0    100      0 4567 i

 *>   5.5.5.5/32       192.168.35.5             0             0 4567 i

 *                     192.168.13.1                           0 1 4567 i

 * i                   2.2.2.2                  0    100      0 4567 i

 *>   6.6.6.6/32       192.168.35.5                           0 4567 i

 *                     192.168.13.1                           0 1 4567 i

 * i                   2.2.2.2                  0    100      0 4567 i

 *>   7.7.7.7/32       192.168.35.5                           0 4567 i

 *                     192.168.13.1                           0 1 4567 i

 * i                   2.2.2.2                  0    100      0 4567 i

 * i  9.0.0.0          2.2.2.2                  0    100      0 4567
4292935689 i
```

```
*    10.0.0.1/32      192.168.13.1                           0 1 999 i
* i                   2.2.2.2                  0    100      0 1 999 i
*    12.34.0.0/16     192.168.13.1                           0 1 999 ?
* i                   2.2.2.2                  0    100      0 1 999 ?
*    23.45.0.0/16     192.168.13.1                           0 1 999 ?
* i                   2.2.2.2                  0    750      0 1 999 ?
*    66.77.0.0/17     192.168.13.1                           0 1 999 i
* i                   2.2.2.2                  0    100      0 1 999 i
*    89.100.0.0/18    192.168.13.1                           0 1 999 ?
* i                   2.2.2.2                  0    100      0 1 999 ?
*    91.200.0.0/18    192.168.13.1                           0 1 999 ?
* i                   2.2.2.2                  0    100      0 1 999 ?
*    102.64.0.0/18    192.168.13.1                           0 1 999 ?
*    123.45.0.0/17    192.168.13.1                           0 1 999 ?
* i                   2.2.2.2                  0    100      0 1 999 ?
*    130.25.0.0/18    192.168.13.1                           0 1 999 ?
* i                   2.2.2.2                  0    100      0 1 999 ?
*    175.45.200.0/21  192.168.13.1                           0 1 999 ?
* i                   2.2.2.2                  0    100      0 1 999 ?
*    176.0.0.0/4      192.168.13.1                           0 1 999 i
* i                   2.2.2.2                  0    100      0 1 999 i
```

```
*    195.225.0.0/19   192.168.13.1                              0 1 999 ?
* i                   2.2.2.2                  0    100         0 1 999 ?
*    199.10.192.0/22  192.168.13.1                              0 1 999 ?
* i                   2.2.2.2                  0    100         0 1 999 ?
*    203.0.113.0/30   192.168.13.1                              0 1 999 ?
* i                   2.2.2.2                  0    100         0 1 999 ?
*    210.45.128.0/23  192.168.13.1                              0 1 999 ?
* i                   2.2.2.2                  0    100         0 1 999 ?
*    212.12.16.0/21   192.168.13.1                              0 1 999 ?
* i                   2.2.2.2                  0    100         0 1 999 ?
*    216.80.192.0/22  192.168.13.1                              0 1 999 ?
* i                   2.2.2.2                  0    100         0 1 999 ?
*    220.85.200.0/23  192.168.13.1                              0 1 999 ?
* i                   2.2.2.2                  0    100         0 1 999 ?
*    221.25.0.0/19    192.168.13.1                              0 1 999 ?
* i                   2.2.2.2                  0    100         0 1 999 ?
```

Note that all of the routes learned from R5 that used 4567 as a transit AS have been removed. However, the routes that originated in AS 4567 have remained in the BGP table. Only one route remains that used the 4567 as a transit AS, and that is the 9.0.0.0/8 destination. This however was not learned via R5, but via iBGP using the R2 peer, and this is why it remains in the BGP table.

BGP Route Dampening

In this task, you are asked to enable route dampening on R2 for the networks on Loopbacks 13, 14, and 15 of the ISP router. The dampening parameters should be: half-life: 15, reuse: 750, suppress: 2000, max-suppress-time: 60.

Since we need to specify the network for which dampening will be applied, we must use a route map rather than applying it globally.

To do so, issue the following commands. Create an access list that will match the networks in question:

```
R2(config)#ip access-list standard DAMP

R2(config-std-nacl)#permit 195.225.0.0 0.0.31.255

R2(config-std-nacl)#permit 199.10.192.0 0.0.3.255

R2(config-std-nacl)#permit 210.45.128.0 0.0.1.255
```

Next, create the route map, reference the ACL, and apply the dampening parameters:

```
R2(config)#route-map DAMPENING permit 10

R2(config-route-map)#match ip address DAMP

R2(config-route-map)#set dampening 15 750 2000 60
```

Finally, apply the route map to the IPv4 address family of the BGP configuration like so:

```
R2(config)#router bgp 23

R2(config-router)#address-family ipv4

R2(config-router-af)#bgp dampening route-map DAMPENING
```

To verify that dampening has been applied correctly, issue the following command:

```
R2#show ip bgp dampening parameters
```

```
 dampening 15 750 2000 60 (route-map DAMPENING 10)

  Half-life time      : 15 mins       Decay Time       : 2320 secs

  Max suppress penalty: 12000         Max suppress time: 60 mins

  Suppress penalty    :  2000         Reuse penalty    : 750
```

As you can see, the configured values have been applied via the specific route map.

BGP Peer Groups

You are asked to create a peer group on R1 to group identical configurations of BGP peers for simpler administration. To see which BGP peers can be grouped, take a look at the current BGP configuration on R1:

```
R1#show run | sec bgp

router bgp 1

 bgp log-neighbor-changes

 neighbor 2001:DB8::2 remote-as 999

 neighbor 192.168.12.2 remote-as 23

 neighbor 192.168.13.3 remote-as 23

 neighbor 192.168.14.4 remote-as 4567

 neighbor 203.0.113.2 remote-as 999

 !

 address-family ipv4

  network 1.1.1.1 mask 255.255.255.255

  no neighbor 2001:DB8::2 activate
```

```
  neighbor 192.168.12.2 activate

  neighbor 192.168.12.2 send-community

  neighbor 192.168.12.2 route-map COMMUNITIES out

  neighbor 192.168.13.3 activate

  neighbor 192.168.13.3 send-community

  neighbor 192.168.13.3 route-map COMMUNITIES out

  neighbor 192.168.14.4 activate

  neighbor 192.168.14.4 send-community

  neighbor 192.168.14.4 route-map COMMUNITIES out

  neighbor 203.0.113.2 activate

 exit-address-family

 !

 address-family ipv6

  neighbor 2001:DB8::2 activate

  neighbor 192.168.12.2 activate

  neighbor 192.168.13.3 activate

 exit-address-family
```

Notice that in the IPv4 address family configuration, R2, R3, and R4 all have the same configuration. Thus, these three can be consolidated into a single peer group. To do so, issue the following commands:

```
R1(config)#router bgp 1
```

```
R1(config-router)#neighbor R2R3R4 peer-group

R1(config-router)#neighbor 192.168.12.2 peer-group R2R3R4

R1(config-router)#neighbor 192.168.13.3 peer-group R2R3R4

R1(config-router)#neighbor 192.168.14.4 peer-group R2R3R4

R1(config-router)#address-family ipv4

R1(config-router-af)#neighbor R2R3R4 send-community

R1(config-router-af)#neighbor R2R3R4 route-map COMMUNITIES out
```

The above configuration consolidates the `send-community` and the `route-map COMMUNITIES` commands to a single command for all three BGP peers. The BGP configuration on R1 should now look like this:

```
R1#show run | section bgp

router bgp 1

 bgp log-neighbor-changes

 neighbor R2R3R4 peer-group

 neighbor 2001:DB8::2 remote-as 999

 neighbor 192.168.12.2 remote-as 23

 neighbor 192.168.12.2 peer-group R2R3R4

 neighbor 192.168.13.3 remote-as 23

 neighbor 192.168.13.3 peer-group R2R3R4

 neighbor 192.168.14.4 remote-as 4567

 neighbor 192.168.14.4 peer-group R2R3R4
```

```
neighbor 203.0.113.2 remote-as 999
!
address-family ipv4
 network 1.1.1.1 mask 255.255.255.255
 neighbor R2R3R4 send-community
 neighbor R2R3R4 route-map COMMUNITIES out
 no neighbor 2001:DB8::2 activate
 neighbor 192.168.12.2 activate
 neighbor 192.168.13.3 activate
 neighbor 192.168.14.4 activate
 neighbor 203.0.113.2 activate
exit-address-family
!
address-family ipv6
 neighbor 2001:DB8::2 activate
 neighbor 192.168.13.3 activate
exit-address-family
```

The peer group creation takes place outside of the address family configuration mode, but the actual application of parameters to the peer group is applied within the address family configuration mode.

BGP Soft Reconfiguration and Route Refresh

In this task, you are instructed to configure routers R6 and R7 so that their inter-sub-AS peerings are enabled with soft reconfiguration in an inbound direction.

To do so, issue the following commands:

```
R6(config)#router bgp 67

R6(config-router)#neighbor 4.4.4.4 soft-reconfiguration inbound

R7(config)#router bgp 67

R7(config-router)#neighbor 5.5.5.5 soft-reconfiguration inbound
```

BGP Multipath

For this task, you are asked to ensure that on R1, up to two paths are chosen as BGP best paths for the destination network of 192.168.23.0/24, as well as all other routes that R1 has learned that have the same Weight, Local Preference, AS Path, Origin code, MED and IGP metric. You will use the BGP multipath feature to fulfill the requirements of this task.

First, let's advertise the 192.168.23.0/24 network using BGP on routers R2 and R3:

```
R2(config)#router bgp 23

R2(config-router)#address-family ipv4

R2(config-router-af)#network 192.168.23.0 mask 255.255.255.0

R3(config)#router bgp 23

R3(config-router)#address-family ipv4

R3(config-router-af)#network 192.168.23.0 mask 255.255.255.0
```

Let's now take a look at the BGP table of R1:

```
R1#show ip bgp
```

```
BGP table version is 52, 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, f RT-Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *>   1.1.1.1/32       0.0.0.0                  0         32768 i

 *    2.2.2.2/32       192.168.14.4                           0 4567 23 i

 *                     192.168.13.3                           0 23 i

 *>                    192.168.12.2             0             0 23 i

 *    3.3.3.3/32       192.168.14.4                           0 4567 23 i

 *                     192.168.13.3             0             0 23 i

 *>                    192.168.12.2                           0 23 i

 *>   4.4.4.4/32       192.168.14.4             0             0 4567 i

 *                     192.168.13.3                           0 23 4567 i

 *                     192.168.12.2                           0 23 4567 i

 *>   5.5.5.5/32       192.168.14.4                           0 4567 i
```

```
 *                      192.168.13.3                                    0 23 4567 i

 *                      192.168.12.2                                    0 23 4567 i

 *>    6.6.6.6/32       192.168.14.4                                    0 4567 i

 *                      192.168.13.3                                    0 23 4567 i

 *                      192.168.12.2                                    0 23 4567 i

 *>    7.7.7.7/32       192.168.14.4                                    0 4567 i

 *                      192.168.13.3                                    0 23 4567 i

 *                      192.168.12.2                                    0 23 4567 i

 *     9.0.0.0          192.168.14.4                                    0 4567 4567
4567 4567 4567 4292935689 i

 *                      192.168.13.3           800                      0 23 4567
4292935689 i

 *>                     192.168.12.2                                    0 23 4567
4292935689 i

 *>    10.0.0.1/32      203.0.113.2              0                      0 999 i

 *>    12.34.0.0/16     203.0.113.2              0                      0 999 ?

 *>    23.45.0.0/16     203.0.113.2              0                      0 999 ?

 *>    66.77.0.0/17     203.0.113.2              0                      0 999 i

 *>    89.100.0.0/18    203.0.113.2              0                      0 999 ?

 *>    91.200.0.0/18    203.0.113.2              0                      0 999 ?

 *>    102.64.0.0/18    203.0.113.2              0                      0 999 ?

 *>    123.45.0.0/17    203.0.113.2              0                      0 999 ?

 *>    130.25.0.0/18    203.0.113.2              0                      0 999 ?
```

```
 *>   175.45.200.0/21  203.0.113.2              0             0 999 ?
 *>   176.0.0.0/4      203.0.113.2              0             0 999 i
 *    192.168.23.0     192.168.14.4                           0 4567 23 i
 *                     192.168.13.3             0             0 23 i
 *>                    192.168.12.2             0             0 23 i
 *>   195.225.0.0/19   203.0.113.2              0             0 999 ?
 *>   199.10.192.0/22  203.0.113.2              0             0 999 ?
 r>   203.0.113.0/30   203.0.113.2              0             0 999 ?
 *>   210.45.128.0/23  203.0.113.2              0             0 999 ?
 *>   212.12.16.0/21   203.0.113.2              0             0 999 ?
 *>   216.80.192.0/22  203.0.113.2              0             0 999 ?
 *>   220.85.200.0/23  203.0.113.2              0             0 999 ?
 *>   221.25.0.0/19    203.0.113.2              0             0 999 ?
```

Notice that the 192.168.23.0/24 network now appears in the BGP table, and it has three possible paths. Let's now enable BGP multipath on R1:

```
R1(config)#router bgp 1

R1(config-router)#address-family ipv4

R1(config-router-af)#maximum-paths 2
```

Let's look again at R1's BGP table:

```
R1#show ip bgp
```

```
BGP table version is 55, 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, f RT-
Filter,

              x best-external, a additional-path, c RIB-compressed,

              t secondary path,

Origin codes: i - IGP, e - EGP, ? - incomplete

RPKI validation codes: V valid, I invalid, N Not found



     Network          Next Hop            Metric LocPrf Weight Path

 *>   1.1.1.1/32       0.0.0.0                  0         32768 i

 *    2.2.2.2/32       192.168.14.4                           0 4567 23 i

 *m                    192.168.13.3                           0 23 i

 *>                    192.168.12.2             0             0 23 i

 *    3.3.3.3/32       192.168.14.4                           0 4567 23 i

 *m                    192.168.13.3             0             0 23 i

 *>                    192.168.12.2                           0 23 i

 *>   4.4.4.4/32       192.168.14.4             0             0 4567 i

 *                     192.168.13.3                           0 23 4567 i

 *                     192.168.12.2                           0 23 4567 i

 *>   5.5.5.5/32       192.168.14.4                           0 4567 i
```

```
 *                    192.168.13.3                             0 23 4567 i
 *                    192.168.12.2                             0 23 4567 i
 *>   6.6.6.6/32      192.168.14.4                             0 4567 i
 *                    192.168.13.3                             0 23 4567 i
 *                    192.168.12.2                             0 23 4567 i
 *>   7.7.7.7/32      192.168.14.4                             0 4567 i
 *                    192.168.13.3                             0 23 4567 i
 *                    192.168.12.2                             0 23 4567 i
 *    9.0.0.0         192.168.14.4                             0 4567 4567
4567 4567 4567 4292935689 i
 *                    192.168.13.3           800               0 23 4567
4292935689 i
 *>                   192.168.12.2                             0 23 4567
4292935689 i
 *>   10.0.0.1/32     203.0.113.2              0               0 999 i
 *>   12.34.0.0/16    203.0.113.2              0               0 999 ?
 *>   23.45.0.0/16    203.0.113.2              0               0 999 ?
 *>   66.77.0.0/17    203.0.113.2              0               0 999 i
 *>   89.100.0.0/18   203.0.113.2              0               0 999 ?
 *>   91.200.0.0/18   203.0.113.2              0               0 999 ?
 *>   102.64.0.0/18   203.0.113.2              0               0 999 ?
 *>   123.45.0.0/17   203.0.113.2              0               0 999 ?
 *>   130.25.0.0/18   203.0.113.2              0               0 999 ?
```

```
*>   175.45.200.0/21  203.0.113.2              0             0 999 ?
*>   176.0.0.0/4      203.0.113.2              0             0 999 i
*    192.168.23.0     192.168.14.4                           0 4567 23 i
*m                    192.168.13.3             0             0 23 i
*>                    192.168.12.2             0             0 23 i
*>   195.225.0.0/19   203.0.113.2              0             0 999 ?
*>   199.10.192.0/22  203.0.113.2              0             0 999 ?
r>   203.0.113.0/30   203.0.113.2              0             0 999 ?
*>   210.45.128.0/23  203.0.113.2              0             0 999 ?
*>   212.12.16.0/21   203.0.113.2              0             0 999 ?
*>   216.80.192.0/22  203.0.113.2              0             0 999 ?
*>   220.85.200.0/23  203.0.113.2              0             0 999 ?
*>   221.25.0.0/19    203.0.113.2              0             0 999 ?
```

Notice that for the 192.168.23.0/24 network, we now have one best path marked by ">" and a second path marked by "m," which indicates multipath. The result is that both are now valid as the best paths to the destination.

We also see two more routes affected by this configuration: the 2.2.2.2 and the 3.3.3.3 networks. These, too, have fulfilled their requirements for multipath, having the same identical attributes. Since the multipath command is global, it applies to all prefixes in the BGP table.

Let's take a closer look at the 192.168.23.0/24 network in the BGP table:

```
R1#show ip bgp 192.168.23.0

BGP routing table entry for 192.168.23.0/24, version 55
```

```
Paths: (3 available, best #3, table default)

Multipath: eBGP

  Advertised to update-groups:

     10         11

  Refresh Epoch 1

  4567 23

    192.168.14.4 from 192.168.14.4 (4.4.4.4)

      Origin IGP, localpref 100, valid, external

      rx pathid: 0, tx pathid: 0

  Refresh Epoch 1

  23

    192.168.13.3 from 192.168.13.3 (3.3.3.3)

      Origin IGP, metric 0, localpref 100, valid, external,
multipath(oldest)

      rx pathid: 0, tx pathid: 0

  Refresh Epoch 1

  23

    192.168.12.2 from 192.168.12.2 (2.2.2.2)

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

      rx pathid: 0, tx pathid: 0x0
```

From the above output, you can see that the path via R2 is the best path, and the path via R3 is marked as the multipath, which essentially means that it, too, is marked as a best path.

BGP Next-Hop Tracking

For this task, you must ensure that BGP next hop tracking is enabled for R1 and that the next hop IPs of R2 and R3 are being actively tracked and modify the next hop trigger delay to 10 seconds for the IPv4 address family.

To examine the current status of the next hop tracking feature, issue the following command:

```
R1#show run all | sec bgp

ipv6 multicast rpf use-bgp

 no bgp-policy accounting input

 no bgp-policy accounting output

 no bgp-policy accounting input source

 no bgp-policy accounting output source

 no bgp-policy source ip-prec-map

 no bgp-policy source ip-qos-map

 no bgp-policy destination ip-prec-map

 no bgp-policy destination ip-qos-map

 no bgp-policy accounting input

 no bgp-policy accounting output

 no bgp-policy accounting input source

 no bgp-policy accounting output source
```

```
no bgp-policy source ip-prec-map

no bgp-policy source ip-qos-map

no bgp-policy destination ip-prec-map

no bgp-policy destination ip-qos-map

no bgp-policy accounting input

no bgp-policy accounting output

no bgp-policy accounting input source

no bgp-policy accounting output source

no bgp-policy source ip-prec-map

no bgp-policy source ip-qos-map

no bgp-policy destination ip-prec-map

no bgp-policy destination ip-qos-map

no bgp-policy accounting input

no bgp-policy accounting output

no bgp-policy accounting input source

no bgp-policy accounting output source

no bgp-policy source ip-prec-map

no bgp-policy source ip-qos-map

no bgp-policy destination ip-prec-map

no bgp-policy destination ip-qos-map

no bgp-policy accounting input
```

```
 no bgp-policy accounting output

 no bgp-policy accounting input source

 no bgp-policy accounting output source

 no bgp-policy source ip-prec-map

 no bgp-policy source ip-qos-map

 no bgp-policy destination ip-prec-map

 no bgp-policy destination ip-qos-map

router bgp 1

 bgp fast-external-fallover

 bgp route-map-cache

 no bgp asnotation dot

 no bgp consistency-checker

 bgp client-to-client reflection

 bgp client-to-client reflection intra-cluster cluster-id any

 bgp transport path-mtu-discovery

 bgp enforce-first-as

 bgp log-neighbor-changes

 bgp dynamic-med-interval 600

 bgp listen limit 100

 bgp update-delay 120

 bgp graceful-restart restart-time 120
```

```
 bgp graceful-restart stalepath-time 360
 bgp refresh stalepath-time 0
 bgp refresh max-eor-time 0
 bgp regexp deterministic
 bgp default ipv4-unicast
 bgp default ipv6-nexthop
 bgp default local-preference 100
 bgp default route-target filter
 timers bgp 60 180 0
 neighbor R2R3R4 peer-group
 neighbor 2001:DB8::2 remote-as 999
 neighbor 192.168.12.2 remote-as 23
 neighbor 192.168.12.2 peer-group R2R3R4
 neighbor 192.168.13.3 remote-as 23
 neighbor 192.168.13.3 peer-group R2R3R4
 neighbor 192.168.14.4 remote-as 4567
 neighbor 192.168.14.4 peer-group R2R3R4
 neighbor 203.0.113.2 remote-as 999
 !
 address-family ipv4
  no synchronization
```

```
 bgp aggregate-timer 30

 bgp update-group split as-override

 bgp nexthop trigger enable

 bgp nexthop trigger delay 5

 bgp scan-time 60

 network 1.1.1.1 mask 255.255.255.255

 neighbor R2R3R4 send-community

 neighbor R2R3R4 weight 0

 neighbor R2R3R4 route-map COMMUNITIES out

 no neighbor 2001:DB8::2 activate

 neighbor 192.168.12.2 activate

 neighbor 192.168.13.3 activate

 neighbor 192.168.14.4 activate

 neighbor 203.0.113.2 activate

 maximum-paths 2

 distance bgp 20 200 200

 no auto-summary

exit-address-family

!

address-family ipv6

 distance bgp 20 200 200
```

```
  bgp aggregate-timer 30

  bgp update-group split as-override

  bgp nexthop trigger enable

  bgp nexthop trigger delay 5

  bgp scan-time 60

  neighbor 2001:DB8::2 activate

  neighbor 192.168.13.3 activate

 exit-address-family
```

The next hop feature commands are hidden commands and can only be displayed with the `show running-config all` command including the `all` keyword. As you can see, for address families IPv4 and IPv6, the next hop tracking feature is enabled. Let's change the next hop trigger delay to 10 seconds for the IPv4 address family:

```
R1(config)#router bgp 1

R1(config-router)#address-family ipv4

R1(config-router-af)#bgp nexthop trigger delay 10
```

Let's check the results of these commands. Note that some of the following output has been omitted.

```
R1#show run all | section bgp

router bgp 1

 !

 !>-- Output Omitted --

 !
```

```
address-family ipv4
 no synchronization
 bgp aggregate-timer 30
 bgp update-group split as-override
 bgp nexthop trigger enable
 bgp nexthop trigger delay 10
 bgp scan-time 60
 network 1.1.1.1 mask 255.255.255.255
 neighbor R2R3R4 send-community
 neighbor R2R3R4 weight 0
 neighbor R2R3R4 route-map COMMUNITIES out
 no neighbor 2001:DB8::2 activate
 neighbor 192.168.12.2 activate
 neighbor 192.168.13.3 activate
 neighbor 192.168.14.4 activate
 neighbor 203.0.113.2 activate
 maximum-paths 2
 distance bgp 20 200 200
 no auto-summary
exit-address-family
!
```

```
 address-family ipv6

  distance bgp 20 200 200

  bgp aggregate-timer 30

  bgp update-group split as-override

  bgp nexthop trigger enable

  bgp nexthop trigger delay 5

  bgp scan-time 60

  neighbor 2001:DB8::2 activate

  neighbor 192.168.13.3 activate

 exit-address-family
```

As you can see, the delay has been modified.

Want to take a look for yourself? Here you will find the final configuration of each device.

Conclusion

```
hostname ISP

!

ip cef

ipv6 unicast-routing

ipv6 cef

!

interface Loopback0
```

```
 ip address 10.0.0.1 255.255.255.255

 ipv6 address 2001:DB8:1::1/128

!

interface Loopback1

 ip address 12.34.0.1 255.255.0.0

 ipv6 address 2001:DB8:12:34::1/64

!

interface Loopback2

 ip address 23.45.0.1 255.255.0.0

 ipv6 address 2001:DB8:23:45::1/64

!

interface Loopback3

 ip address 66.77.0.1 255.255.128.0

 ipv6 address 2001:DB8:66:77::1/64

!

interface Loopback4

 ip address 89.100.0.1 255.255.192.0

 ipv6 address 2001:DB8:89:100::1/64

!

interface Loopback5

 ip address 91.200.0.1 255.255.192.0
```

```
 ipv6 address 2001:DB8:91:200::1/64

!

interface Loopback6

 ip address 102.64.0.1 255.255.192.0

 ipv6 address 2001:DB8:102:64::1/64

!

interface Loopback7

 ip address 123.45.0.1 255.255.128.0

 ipv6 address 2001:DB8:123:45::1/64

!

interface Loopback8

 ip address 130.25.0.1 255.255.192.0

 ipv6 address 2001:DB8:130:25::1/64

!

interface Loopback9

 ip address 175.45.200.1 255.255.248.0

 ipv6 address 2001:DB8:175:45::1/64

!

interface Loopback10

 ip address 183.77.220.1 255.255.252.0

 ipv6 address 2001:DB8:183:77::1/64
```

```
!
interface Loopback11
 ip address 185.100.0.1 255.255.224.0
 ipv6 address 2001:DB8:185:100::1/64
!
interface Loopback12
 ip address 190.30.128.1 255.255.128.0
 ipv6 address 2001:DB8:190:32::1/64
!
interface Loopback13
 ip address 195.225.0.1 255.255.224.0
 ipv6 address 2001:DB8:195:225::1/64
!
interface Loopback14
 ip address 199.10.192.1 255.255.252.0
 ipv6 address 2001:DB8:199:10::1/64
!
interface Loopback15
 ip address 210.45.128.1 255.255.254.0
 ipv6 address 2001:DB8:210:45::1/64
!
```

```
interface Loopback16

 ip address 212.12.16.1 255.255.248.0

 ipv6 address 2001:DB8:212:12::1/64

!

interface Loopback17

 ip address 216.80.192.1 255.255.252.0

 ipv6 address 2001:DB8:216:80::1/64

!

interface Loopback18

 ip address 220.85.200.1 255.255.254.0

 ipv6 address 2001:DB8:220:85::1/64

!

interface Loopback19

 ip address 221.25.0.1 255.255.224.0

 ipv6 address 2001:DB8:225:25::1/64

!

interface GigabitEthernet0/0

 ip address 203.0.113.2 255.255.255.252

 duplex auto

 speed auto

 media-type rj45
```

```
 ipv6 address 2001:DB8::2/64
!
router bgp 999
 bgp log-neighbor-changes
 neighbor 2001:DB8::1 remote-as 1
 neighbor 203.0.113.1 remote-as 1
 !
 address-family ipv4
  network 10.0.0.1 mask 255.255.255.255
  network 66.77.0.0 mask 255.255.128.0
  aggregate-address 176.0.0.0 240.0.0.0 summary-only
  redistribute connected
  no neighbor 2001:DB8::1 activate
  neighbor 203.0.113.1 activate
 exit-address-family
 !
 address-family ipv6
  network 2001:DB8:1::1/128
  network 2001:DB8:12:34::/64
  network 2001:DB8:23:45::/64
  network 2001:DB8:66:77::/64
```

```
  network 2001:DB8:89:100::/64

  network 2001:DB8:91:200::/64

  network 2001:DB8:102:64::/64

  network 2001:DB8:123:45::/64

  network 2001:DB8:130:25::/64

  network 2001:DB8:175:45::/64

  network 2001:DB8:183:77::/64

  network 2001:DB8:185:100::/64

  network 2001:DB8:190:32::/64

  network 2001:DB8:195:225::/64

  network 2001:DB8:199:10::/64

  network 2001:DB8:210:45::/64

  network 2001:DB8:212:12::/64

  network 2001:DB8:216:80::/64

  network 2001:DB8:220:85::/64

  network 2001:DB8:225:25::/64

  neighbor 2001:DB8::1 activate

 exit-address-family

!

end
```

```
hostname R1

!

ip cef

ipv6 unicast-routing

ipv6 cef

!

interface Loopback0

 ip address 1.1.1.1 255.255.255.255

!

interface GigabitEthernet0/0

 ip address 203.0.113.1 255.255.255.252

 ipv6 address 2001:DB8::1/64

!

interface GigabitEthernet0/1

 ip address 192.168.14.1 255.255.255.0

!

interface GigabitEthernet0/2

 ip address 192.168.12.1 255.255.255.0

!

interface GigabitEthernet0/3

 ip address 192.168.13.1 255.255.255.0
```

```
!
router bgp 1
 neighbor R2R3R4 peer-group
 neighbor 2001:DB8::2 remote-as 999
 neighbor 192.168.12.2 remote-as 23
 neighbor 192.168.12.2 peer-group R2R3R4
 neighbor 192.168.13.3 remote-as 23
 neighbor 192.168.13.3 peer-group R2R3R4
 neighbor 192.168.14.4 remote-as 4567
 neighbor 192.168.14.4 peer-group R2R3R4
 neighbor 203.0.113.2 remote-as 999
 !
 address-family ipv4
  bgp nexthop trigger delay 10
  network 1.1.1.1 mask 255.255.255.255
  neighbor R2R3R4 send-community
  neighbor R2R3R4 route-map COMMUNITIES out
  no neighbor 2001:DB8::2 activate
  neighbor 192.168.12.2 activate
  neighbor 192.168.13.3 activate
  neighbor 192.168.14.4 activate
```

```
  neighbor 203.0.113.2 activate
  maximum-paths 2
 exit-address-family
 !
 address-family ipv6
  neighbor 2001:DB8::2 activate
  neighbor 192.168.13.3 activate
 exit-address-family
!
route-map COMMUNITIES permit 10
 match ip address 1
 set community no-advertise
!
route-map COMMUNITIES permit 12
 match ip address 2
 set community no-export
!
route-map COMMUNITIES permit 14
 match ip address 3
 set community local-AS
!
```

```
route-map COMMUNITIES permit 20

!

access-list 1 permit 102.64.0.0 0.0.63.255

access-list 2 permit 123.45.0.0 0.0.31.255

access-list 3 permit 130.25.0.0 0.0.63.255

!

end
```

```
hostname R2

!

ip cef

ipv6 unicast-routing

ipv6 cef

!

interface Loopback0

 ip address 2.2.2.2 255.255.255.255

!

interface GigabitEthernet0/0

 ip address 192.168.12.2 255.255.255.0

!

interface GigabitEthernet0/1
```

```
 ip address 192.168.23.2 255.255.255.0
!
interface GigabitEthernet0/2
 ip address 192.168.24.2 255.255.255.0
!
router ospf 1
 network 2.2.2.2 0.0.0.0 area 0
 network 192.168.23.0 0.0.0.255 area 0
!
router bgp 23
 neighbor 3.3.3.3 remote-as 23
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 192.168.12.1 remote-as 1
 neighbor 192.168.24.4 remote-as 4567
 !
 address-family ipv4
  bgp dampening route-map DAMPENING
  network 2.2.2.2 mask 255.255.255.255
  network 192.168.23.0
  neighbor 3.3.3.3 activate
  neighbor 3.3.3.3 send-community
```

```
  neighbor 3.3.3.3 next-hop-self
  neighbor 192.168.12.1 activate
  neighbor 192.168.12.1 route-map LOCALPREF in
  neighbor 192.168.24.4 activate
  neighbor 192.168.24.4 send-community
  neighbor 192.168.24.4 distribute-list NO-TRANSIT out
 exit-address-family
 !
 address-family ipv6
  neighbor 3.3.3.3 activate
  neighbor 192.168.12.1 activate
 exit-address-family
!
ip access-list standard DAMP
 permit 195.225.0.0 0.0.31.255
 permit 199.10.192.0 0.0.3.255
 permit 210.45.128.0 0.0.1.255
ip access-list standard NO-TRANSIT
 deny   175.45.200.0 0.0.7.255
 permit any
!
```

```
route-map LOCALPREF permit 10

 match ip address 1

 set local-preference 750

!

route-map LOCALPREF permit 20

!

route-map DAMPENING permit 10

 match ip address DAMP

 set dampening 15 750 2000 60

!

access-list 1 permit 23.45.0.0 0.0.255.255

!

end
```

```
hostname R3

!

ip cef

ipv6 unicast-routing

ipv6 cef

!

interface Loopback0
```

```
 ip address 3.3.3.3 255.255.255.255
!
interface GigabitEthernet0/0
 ip address 192.168.13.3 255.255.255.0
!
interface GigabitEthernet0/1
 ip address 192.168.23.3 255.255.255.0
!
interface GigabitEthernet0/2
 ip address 192.168.35.3 255.255.255.0
!
router ospf 1
 network 3.3.3.3 0.0.0.0 area 0
 network 192.168.23.0 0.0.0.255 area 0
!
router bgp 23
 neighbor 2.2.2.2 remote-as 23
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 192.168.13.1 remote-as 1
 neighbor 192.168.35.5 remote-as 4567
 !
```

```
 address-family ipv4
  network 3.3.3.3 mask 255.255.255.255
  network 192.168.23.0
  neighbor 2.2.2.2 activate
  neighbor 2.2.2.2 send-community
  neighbor 2.2.2.2 next-hop-self
  neighbor 192.168.13.1 activate
  neighbor 192.168.13.1 route-map MED out
  neighbor 192.168.35.5 activate
  neighbor 192.168.35.5 send-community
  neighbor 192.168.35.5 distribute-list NO-TRANSIT out
  neighbor 192.168.35.5 route-map AS_PATH_FILTER in
 exit-address-family
 !
 address-family ipv6
  neighbor 2.2.2.2 activate
  neighbor 192.168.13.1 activate
 exit-address-family
!
ip as-path access-list 2 permit 4567$
ip as-path access-list 2 deny 4567_
```

```
ip as-path access-list 2 permit .*

!

ip access-list standard NO-TRANSIT

 deny   175.45.200.0 0.0.7.255

 permit any

!

route-map AS_PATH_FILTER permit 10

 match as-path 2

!

route-map MED permit 10

 match ip address 1

 set metric 800

!

route-map MED permit 20

!

access-list 1 permit 9.0.0.0 0.255.255.255

!

end
```

```
hostname R4

!
```

```
ip cef
!
interface Loopback0
 ip address 4.4.4.4 255.255.255.255
!
interface GigabitEthernet0/0
 ip address 192.168.14.4 255.255.255.0
!
interface GigabitEthernet0/1
 ip address 192.168.24.4 255.255.255.0
!
interface GigabitEthernet0/2
 ip address 192.168.45.4 255.255.255.0
!
interface GigabitEthernet0/3
 ip address 192.168.46.4 255.255.255.0
!
router ospf 1
 network 4.4.4.4 0.0.0.0 area 0
 network 192.168.45.0 0.0.0.255 area 0
 network 192.168.46.0 0.0.0.255 area 0
```

```
!
router bgp 45
 bgp confederation identifier 4567
 bgp confederation peers 67
 network 4.4.4.4 mask 255.255.255.255
 neighbor 5.5.5.5 remote-as 45
 neighbor 5.5.5.5 update-source Loopback0
 neighbor 5.5.5.5 next-hop-self
 neighbor 5.5.5.5 send-community
 neighbor 6.6.6.6 remote-as 67
 neighbor 6.6.6.6 ebgp-multihop 2
 neighbor 6.6.6.6 update-source Loopback0
 neighbor 6.6.6.6 next-hop-self
 neighbor 6.6.6.6 send-community
 neighbor 192.168.14.1 remote-as 1
 neighbor 192.168.14.1 route-map PREPEND out
 neighbor 192.168.24.2 remote-as 23
 neighbor 192.168.24.2 route-map SETWEIGHT in
!
route-map SETWEIGHT permit 10
 match ip address 1
```

```
 set weight 500
!
route-map SETWEIGHT permit 20
!
route-map PREPEND permit 10
 match ip address 2
 set as-path prepend 4567 4567 4567 4567
!
route-map PREPEND permit 20
!
access-list 1 permit 12.34.0.0 0.0.255.255
access-list 2 permit 9.0.0.0 0.255.255.255
!
end
```

```
hostname R5
!
ip cef
!
interface Loopback0
 ip address 5.5.5.5 255.255.255.255
```

```
!
interface Tunnel0
 ip address 192.168.59.5 255.255.255.0
 tunnel source 192.168.58.5
 tunnel destination 192.168.89.9
!
interface GigabitEthernet0/0
 ip address 192.168.35.5 255.255.255.0
!
interface GigabitEthernet0/1
 ip address 192.168.45.5 255.255.255.0
!
interface GigabitEthernet0/2
 ip address 192.168.57.5 255.255.255.0
!
interface GigabitEthernet0/3
 ip address 192.168.58.5 255.255.255.0
!
router ospf 1
 network 5.5.5.5 0.0.0.0 area 0
 network 192.168.45.0 0.0.0.255 area 0
```

```
 network 192.168.57.0 0.0.0.255 area 0
!
router bgp 45
 bgp confederation identifier 4567
 bgp confederation peers 67
 network 5.5.5.5 mask 255.255.255.255
 neighbor 4.4.4.4 remote-as 45
 neighbor 4.4.4.4 update-source Loopback0
 neighbor 4.4.4.4 next-hop-self
 neighbor 4.4.4.4 send-community
 neighbor 7.7.7.7 remote-as 67
 neighbor 7.7.7.7 ebgp-multihop 2
 neighbor 7.7.7.7 update-source Loopback0
 neighbor 7.7.7.7 next-hop-self
 neighbor 7.7.7.7 send-community
 neighbor 9.9.9.9 remote-as 4292935689
 neighbor 9.9.9.9 ebgp-multihop 2
 neighbor 9.9.9.9 update-source Loopback0
 neighbor 9.9.9.9 prefix-list BLOCK-18 out
 neighbor 192.168.35.3 remote-as 23
!
```

```
ip route 9.9.9.9 255.255.255.255 192.168.59.9

ip route 192.168.89.9 255.255.255.255 192.168.58.8

!

ip prefix-list BLOCK-18 seq 5 deny 0.0.0.0/0 ge 18 le 18

ip prefix-list BLOCK-18 seq 10 permit 0.0.0.0/0 le 32

!

end
```

```
hostname R6

!

ip cef

!

interface Loopback0

 ip address 6.6.6.6 255.255.255.255

!

interface GigabitEthernet0/0

 ip address 192.168.46.6 255.255.255.0

!

interface GigabitEthernet0/1

 ip address 192.168.67.6 255.255.255.0

!
```

```
router ospf 1
 network 6.6.6.6 0.0.0.0 area 0
 network 192.168.46.0 0.0.0.255 area 0
 network 192.168.67.0 0.0.0.255 area 0
!
router bgp 67
 bgp confederation identifier 4567
 bgp confederation peers 45
 network 6.6.6.6 mask 255.255.255.255
 neighbor 4.4.4.4 remote-as 45
 neighbor 4.4.4.4 ebgp-multihop 2
 neighbor 4.4.4.4 update-source Loopback0
 neighbor 4.4.4.4 soft-reconfiguration inbound
 neighbor 4.4.4.4 route-map AS_PATH_FILTER in
 neighbor 7.7.7.7 remote-as 67
 neighbor 7.7.7.7 update-source Loopback0
 neighbor 7.7.7.7 route-map AS_PATH_FILTER in
!
ip as-path access-list 1 deny _23_
ip as-path access-list 1 permit .*
!
```

```
route-map AS_PATH_FILTER permit 10

 match as-path 1

!

end
```

```
hostname R7

!

ip cef

!

interface Loopback0

 ip address 7.7.7.7 255.255.255.255

!

interface GigabitEthernet0/0

 ip address 192.168.57.7 255.255.255.0

!

interface GigabitEthernet0/1

 ip address 192.168.67.7 255.255.255.0

!

router ospf 1

 network 7.7.7.7 0.0.0.0 area 0

 network 192.168.57.0 0.0.0.255 area 0
```

```
 network 192.168.67.0 0.0.0.255 area 0

!

router bgp 67

 bgp confederation identifier 4567

 bgp confederation peers 45

 network 7.7.7.7 mask 255.255.255.255

 neighbor 5.5.5.5 remote-as 45

 neighbor 5.5.5.5 ebgp-multihop 2

 neighbor 5.5.5.5 update-source Loopback0

 neighbor 5.5.5.5 soft-reconfiguration inbound

 neighbor 6.6.6.6 remote-as 67

 neighbor 6.6.6.6 update-source Loopback0

!

end
```

```
hostname R8

!

ip cef

!

interface Loopback0

 ip address 8.8.8.8 255.255.255.255
```

```
!
interface GigabitEthernet0/0
 ip address 192.168.58.8 255.255.255.0
!
interface GigabitEthernet0/1
 ip address 192.168.89.8 255.255.255.0
!
ip route 5.5.5.5 255.255.255.255 192.168.58.5
ip route 9.9.9.9 255.255.255.255 192.168.89.9
!
end
```

```
hostname R9
!
ip cef
!
interface Loopback0
 ip address 9.9.9.9 255.255.255.255
!
interface Tunnel0
 ip address 192.168.59.9 255.255.255.0
```

```
 tunnel source 192.168.89.9

 tunnel destination 192.168.58.5

!

interface GigabitEthernet0/0

 ip address 192.168.89.9 255.255.255.0

!

router bgp 4292935689

 network 9.0.0.0

 neighbor 5.5.5.5 remote-as 4567

 neighbor 5.5.5.5 ebgp-multihop 2

 neighbor 5.5.5.5 update-source Loopback0

 distribute-list 110 in

 auto-summary

!

ip route 5.5.5.5 255.255.255.255 192.168.59.5

ip route 192.168.58.5 255.255.255.255 192.168.89.8

!

access-list 110 deny ip any 255.255.255.192 0.0.0.63

access-list 110 permit ip any any

!

end
```