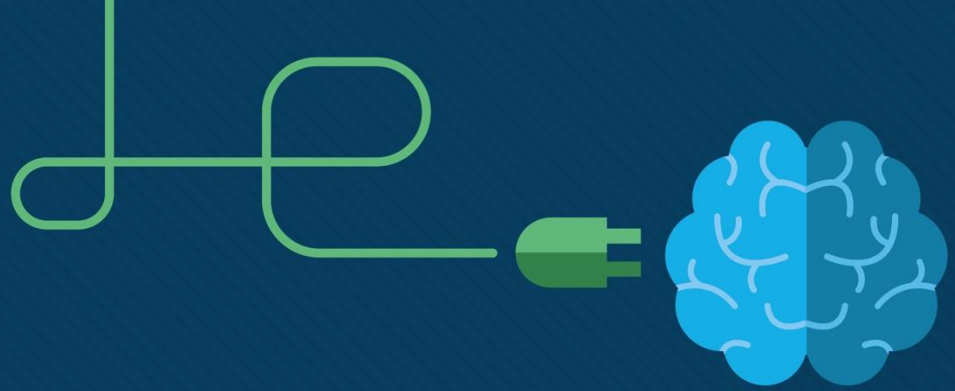

Chapter 14: Troubleshooting BGP

Instructor Materials

CCNP Enterprise: Advanced Routing

Chapter 14 Content

This chapter covers the following content:

- **Troubleshooting BGP Neighbor Adjacencies -** This section examines issues that may prevent a BGP neighbor relationship from forming and how to recognize and troubleshoot these issues. This section focuses primarily on IPv4 unicast BGP, the same issues arise with IPv6 unicast BGP.
- **Troubleshooting BGP Routes -** This section focuses on issues that may prevent BGP routes from being learned or advertised and how to recognize and troubleshoot these issues. This section focuses mostly on IPv4 unicast BGP, the same issues arise with IPv6 unicast BGP routes as well.
- **Troubleshooting BGP Path Selection -** This section explains how BGP determines the best path to reach a destination network and the importance of understanding how this process works for troubleshooting purposes.
- **Troubleshooting BGP for IPv6 -** This section discusses the methods used to successfully troubleshoot additional issues related to BGP for IPv6 that are not seen with BGP for IPv4.
- **BGP Trouble Tickets -** This section presents trouble tickets that demonstrate how to use a structured troubleshooting process to solve a reported problem.
- **MP-BGP Trouble Tickets -** This section presents a trouble ticket that demonstrates how to use a structured troubleshooting process to solve a reported problem.

Troubleshooting BGP Neighbor Adjacencies

- With BGP, you need to establish neighbor adjacencies manually, unlike EIGRP and OSPF, where the neighbor adjacencies form dynamically.
- BGP configurations are prone to human error, which means a greater effort is often needed during troubleshooting.
- There are two flavors of BGP: Internal BGP (iBGP) and External BGP (eBGP). Understanding the differences between the two and the issues relating to each of them is important for troubleshooting.

Verifying IPv4 Unicast BGP Neighbors

To verify IPv4 unicast BGP neighbors, you can use two **show** commands: **show bgp ipv4 unicast summary** and **show bgp ipv4 unicast neighbors.**

- For initial verification of neighbors, use **show bgp ipv4 unicast summary** because it provides condensed output.
- Example 14-1 shows sample output of the **show bgp ipv4 unicast summary** command, which indicates that R1 has two BGP neighbors: 10.1.12.2 and 10.1.13.3.
- Focus your attention on the State**/**PfxRcd column. If there is a number in this column (as there is in Example 14-1), it means you have successfully established a BGP neighbor relationship. If you see Idle or Active, there is a problem in the formation of the neighbor relationship.

Example 14-1 *Verifying BGP Neighbors with show bgp ipv4 unicast summary**

```
R1# show bgp ipv4 unicast summary
BGP router identifier 10.1.13.1, local AS number 65501
BGP table version is 1, main routing table version 1

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.1.12.2       4        65502      16      16        1    0    0 00:11:25        0
10.1.13.3       4        65502      15      12        1    0    0 00:09:51        0
```

Verifying IPv4 Unicast BGP Neighbors (Cont.)

The following are some of the reasons a BGP neighbor relationship might not form:

- **Interface is down -** The interface must be up/up.
- **Layer 3 connectivity is broken -** You need to be able to reach the IP address you are trying to form the adjacency with.
- **Path to the neighbor is through the default route -** You must be able to reach the neighbor using a route other than the default route.
- **Neighbor does not have a route to the local router -** The two routers forming a BGP peering must have routes to each other.
- **Incorrect neighbor statement -** The IP address and ASN in the **neighbor** *ip_address* **remote-as** *as_number* statement must be accurate.
- **ACLs -** An access control list (ACL) or a firewall may be blocking TCP (Transmission Control Protocol) port 179.
- **BGP packets sourced from the wrong IP address -** The source IP (Internet Protocol) address of an inbound BGP packet must match the local neighbor statement.

Verifying IPv4 Unicast BGP Neighbors (Cont.)

- **The TTL (time-to-live) of the BGP packet expires -** The peer may be further away than is permitted.
- **Mismatched authentication -** The two routers must agree on the authentication parameters.
- **Misconfigured peer group -** Peer groups simplify repetitive BGP configurations; however, if not carefully implemented, they can prevent neighbor relationships from forming or routes from being learned.
- **Timers -** Timers do not have to match; however, if the minimum holddown from neighbor option is set, it could prevent a neighbor adjacency.

When troubleshooting BGP neighbor adjacencies, you need to be able to identify these issues and understand why they occur.

Interface is Down or No Layer 3 Connectivity

Interface is Down - The physical or logical interface with the IP address that is being used to form BGP neighbor relationships must be up/up. Use **show ip interface brief** to verify the status of the interface.

Layer 3 Connectivity is Broken – BGP neighbors do not have to be directly connected or in the same subnet to form a neighbor relationship, but you do need to have Layer 3 connectivity. Use the **ping** command in order to determine if you have Layer 3 connectivity. When reviewing the output of **show bgp ipv4 unicast summary** in Example 14-2, notice that the State/PfxRcd field says Idle. This state occurs when the local router is not able to make a TCP connection with the neighbor.

Example 14-2 *Verifying BGP State with show bgp ipv4 unicast summary*

```
R5# show bgp ipv4 unicast summary
BGP router identifier 10.1.45.5, local AS number 65502
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        65502       0       0        1    0    0 never     Idle
```

Path to the Neighbor is Through Default Route

When a ping is successful, but no specific route to the neighbor exists in the routing table, the ping may have used the default route to reach the neighbor. Even though you can reach the neighbor using the default route, BGP does not consider a default route valid for forming an adjacency. In the output of **show bgp ipv4 unicast summary** on R5 in Example 14-6, notice that the state is idle, which indicates that a TCP session cannot be formed.

Example 14-6 *Verifying the BGP State on R5 with show bgp ipv4 unicast summary*

```
R5# show bgp ipv4 unicast summary
BGP router identifier 10.1.45.5, local AS number 65502
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        65502       0       0        1    0    0 never    Idle
```

Neighbor Does Not Have a Route to Local Router

The local router displays the state Idle when it does not have a route to the IP address it is trying to peer with. However, Idle also appears on a router when the neighbor does not have a route back to the local router.

In Example 14-7, you can see that R2, which is trying to form a BGP peering with R5, also displays the state Idle even though it has a route to 5.5.5.5. The Idle state appears because the routers cannot form the TCP session.

Example 14-7 *Verifying BGP State on R2 and Route to 5.5.5.5*

```
R2# show bgp ipv4 unicast summary
BGP router identifier 2.2.2.2, local AS number 65502
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        65502       0       0        1    0    0 00:00:13 Idle
10.1.12.1       4        65501       2       2        1    0    0 00:00:12        0

R2# show ip route 5.5.5.5 255.255.255.255
Routing entry for 5.5.5.5/32
  Known via "eigrp 100", distance 90, metric 131072, type internal
  Redistributing via eigrp 100
  Last update from 10.1.24.4 on GigabitEthernet2/0, 00:23:58 ago
  Routing Descriptor Blocks:
  * 10.1.24.4, from 10.1.24.4, 00:23:58 ago, via GigabitEthernet2/0
      Route metric is 131072, traffic share count is 1
      Total delay is 5020 microseconds, minimum bandwidth is 1000000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 2
```

Incorrect Neighbor Statement

To form a BGP peering, you use the **neighbor ip_address remote-as as_number** command in BGP configuration mode. Example 14-8 displays two **neighbor remote-as** commands on R2. The **neighbor 5.5.5.5 remote-as 65502** command forms an iBGP peering, and **neighbor 10.1.12.1 remote-as 65501** forms an eBGP peering.

There are two very important parts to this command: the address of the peer with which you form the peering and the autonomous system that the peer is in. If you make a mistake with either of these, you see either the Active or Idle state. You can verify the state of the TCP session on the routers by using the **show tcp brief all** command, shown in Example 14-9.

Example 14-8 *Verifying neighbor remote-as Commands on R2*

```
R2# show run | s router bgp
router bgp 65502
 bgp log-neighbor-changes
 neighbor 5.5.5.5 remote-as 65502
 neighbor 5.5.5.5 update-source Loopback0
 neighbor 10.1.12.1 remote-as 65501
```

Example 14-9 *Verifying the State of TCP Sessions*

```
R2# show tcp brief all
TCB        Local Address          Foreign Address        (state)
68DD357C   10.1.12.2.179           10.1.12.1.35780        ESTAB
68DD24DC   2.2.2.2.179             5.5.5.5.45723          ESTAB
```

BGP Packets Sourced from Wrong IP Address

In a redundant topology, a BGP router has multiple active IP addresses configured across its various interfaces. Figure 14-1 shows two BGP autonomous systems. Notice that R2, R3, and R4 could form a BGP peering with each other, using any physical interface, because of the multiple paths.

When you issue the **neighbor** *ip_address* **remote-as** *as_number* command on a router, the address specified is used by the router to determine whether the BGP open message came from a router it should establish a BGP peering with. The BGP open message has a source IP address, and the source IP address is compared with the address in the local **neighbor** *ip_address* **remote-as** *as_number* command. If they match, a BGP peering is formed; if not, no BGP peering is formed. By default, the source address is based on the exit interface of the router sending the BGP open message.

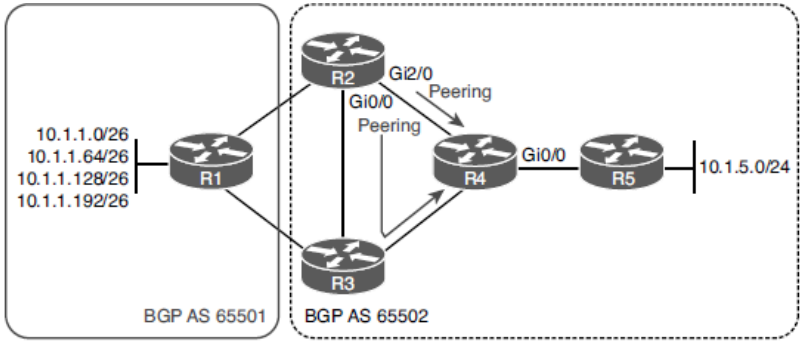

Figure 14-1 *Sample BGP Autonomous System with Redundancy*

BGP Packets Sourced from Wrong IP Address (Cont.)

To control the IP address that is used when sending BGP messages, you use the **neighbor** *ip_address* **update-source** *interface_type interface_number* command.

Example 14-10 shows the output of **show run | section router bgp** on R2.

Example 14-11 shows the appropriate configuration on R4 to ensure that a BGP peering is successful.

Example 14-10 *Verifying the Neighbor Statements and Loopback IP Address on R2*

```
R2# show run | section router bgp
router bgp 65502
 bgp log-neighbor-changes
 neighbor 4.4.4.4 remote-as 65502
 neighbor 4.4.4.4 update-source Loopback0
 neighbor 10.1.12.1 remote-as 65501

R2# show ip interface brief | include Loopback
Loopback0      2.2.2.2        YES         manual up          up
```

Example 14-11 *Verifying That R4's BGP Configuration Mirrors That of R2*

```
R4# show run | section router bgp
router bgp 65502
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 65502
 neighbor 2.2.2.2 update-source Loopback0

R4# show ip interface brief | include Loopback
Loopback0      4.4.4.4        YES         manual up          up
```

ACLs

BGP uses TCP port 179 to establish TCP sessions. If an access control list (ACL) is blocking TCP port 179 anywhere in the path between the routers attempting to form a BGP peering, the peering does not happen.

At the bottom of Example 14-12, the state is Idle on R5 because the TCP session cannot be established with the neighbor at 2.2.2.2 because R4 is denying TCP traffic related to port 179.

Example 14-12 *Verifying ACLs Blocking BGP Packets and the State of R5's Neighbor Relationship*

```
R4# show access-lists
Extended IP access list 100
    10 deny tcp any any eq bgp
    20 deny tcp any eq bgp any
    30 permit ip any any

R4# show ip interface gigabitEthernet 0/0 | include access list
  Outgoing access list is 100
  Inbound access list is not set

R5# show bgp ipv4 unicast summary
BGP router identifier 10.1.45.5, local AS number 65502
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        65502       0       0        1    0    0 00:02:24 Idle
```

ACLs (Cont.)

BGP sessions are server/client relationships. One router is using port 179 (server), and the other router is using an ephemeral port (client). By default, both routers try to establish a TCP session using the three-way handshake because both routers send a TCP syn packet sourced from an ephemeral port and destined to port 179. When both routers respond with an ACK to the request on port 179, two BGP sessions are created. This situation is called a BGP connection collision and the router with the higher BGP RID becomes the server.

To avoid BGP connection collisions, control the server and client roles right from the start by using the **neighbor** *ip_address* **transport connection-mode {active | passive}** command. In this command, active means client, passive means server.

In Example 14-13, the command **show bgp ipv4 unicast neighbors | i ^BGP neighbor|Local port|Foreign port** is used to display R2's neighbors along with the local port number and the foreign port number.

Example 14-13 *Verifying Local and Foreign BGP Port Numbers*

```
R2# show bgp ipv4 unicast neighbors | i ^BGP neighbor|Local port|Foreign port
BGP neighbor is 1.1.1.1, remote AS 65501, external link
Local host: 2.2.2.2, Local port: 23938
Foreign host: 1.1.1.1, Foreign port: 179
BGP neighbor is 3.3.3.3, remote AS 65502, internal link
Local host: 2.2.2.2, Local port: 179
Foreign host: 3.3.3.3, Foreign port: 45936
BGP neighbor is 4.4.4.4, remote AS 65502, internal link
Local host: 2.2.2.2, Local port: 34532
Foreign host: 4.4.4.4, Foreign port: 179
BGP neighbor is 5.5.5.5, remote AS 65502, internal link
Local host: 2.2.2.2, Local port: 49564
Foreign host: 5.5.5.5, Foreign port: 179
```

The TTL of the BGP Packet Expires

By default, an eBGP peering occurs between directly connected routers. With an iBGP peering, the routers can be up to 255 router hops from each other and still form a peering. Example 14-14 shows the output of **show bgp ipv4 unicast neighbors | include BGP neighbor|TTL**, which indicates that the eBGP neighbor at 10.1.12.1 must be reachable in 1 router hop, and the iBGP neighbor at 5.5.5.5 can be up to 255 hops away. If the TTL is not large enough to support the distance required to form a BGP peering, the packet is discarded and no neighbor relationship is formed.

Example 14-14 *Verifying the TTLs of eBGP and iBGP Packets*

```
R2# show bgp ipv4 unicast neighbors | include BGP neighbor|TTL
BGP neighbor is 5.5.5.5, remote AS 65502, internal link
Minimum incoming TTL 0, Outgoing TTL 255
BGP neighbor is 10.1.12.1, remote AS 65501, external link
Minimum incoming TTL 0, Outgoing TTL 1
```

The TTL of the BGP Packet Expires (Cont.)

Example 14-15 shows the configuration of R1 and R2. Notice that R1 is peering with R2, using the neighbor address 2.2.2.2 (R2 loopback) and the source address of Loopback 0 (1.1.1.1). R2 is peering with R1 using the neighbor address 1.1.1.1 (R1 loopback) and source address of Loopback 0 (2.2.2.2). Note that these loopback interfaces are not directly connected (one hop away), and because it is an eBGP neighbor relationship, you can expect the peering to fail.

Example 14-15 *Verifying the BGP Configurations on R1 and R2*

```
R1# show run | s router bgp
router bgp 65501
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 65502
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 10.1.13.3 remote-as 65502

R2# show run | s router bgp
router bgp 65502
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 65501
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 5.5.5.5 remote-as 65502
 neighbor 5.5.5.5 update-source Loopback0
```

The TTL of the BGP Packet Expires (Cont.)

Example 14-15 shows the configuration of R1 and R2. Notice that R1 is peering with R2, using the neighbor address 2.2.2.2 (R2 loopback) and the source address of Loopback 0 (1.1.1.1). R2 is peering with R1 using the neighbor address 1.1.1.1 (R1 loopback) and source address of Loopback 0 (2.2.2.2). Note that these loopback interfaces are not directly connected (one hop away), and because it is an eBGP neighbor relationship, you can expect the peering to fail.

To solve this issue with eBGP neighbors, you can modify the TTL of eBGP packets by using the neighbor ip_address ebgp-multihop [TTL] command. In this case, 2 would be enough to solve the issue. Therefore, on R1, you can type **neighbor 2.2.2.2 ebgp-multihop 2**, and on R2, you can type **neighbor 1.1.1.1 ebgp-multihop 2**.

Example 14-15 *Verifying the BGP Configurations on R1 and R2*

```
R1# show run | s router bgp
router bgp 65501
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 65502
 neighbor 2.2.2.2 update-source Loopback0
 neighbor 10.1.13.3 remote-as 65502

R2# show run | s router bgp
router bgp 65502
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 65501
 neighbor 1.1.1.1 update-source Loopback0
 neighbor 5.5.5.5 remote-as 65502
 neighbor 5.5.5.5 update-source Loopback0
```


Mismatched Authentication

BGP supports Message Digest 5 (MD5) authentication between peers. As is typical with authentication, if any of the parameters do not match, a peering does not form, as shown in Example 14-18.

A BGP authentication mismatch generates a syslog message like the following from the TCP facility:

```
%TCP-6-BADAUTH: No MD5 digest from 2.2.2.2(179) to 1.1.1.1(45577) tableid – 0
```

Example 14-18 *Verifying Neighbor State with Mismatched Authentication*

```
R1# show bgp ipv4 unicast summary
BGP router identifier 1.1.1.1, local AS number 65501
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        65502       0       0        1    0    0 00:02:49 Idle
10.1.13.3       4        65502       7       5        1    0    0 00:02:48        0
```

Misconfigured Peer Groups

When troubleshooting peer group issues, you need to look for the following possible culprits:

- **You forgot to associate the neighbor ip address with the peer group -** After the peer group is created, you need to use the **neighbor** *ip_address* **peer-group** *peer_group_ name* command to associate the neighbor with the configurations in the peer group.
- **The peer group is not configured correctly -** It is possible that you overlooked the fact that what works for one neighbor might not work for the other.
- **The route filter applied to the group is not appropriate for all the peers -** Be careful with filters and make sure they produce the desired results for all neighbors in the peer group.
- **Order of operations produces undesired results -** If there are conflicting entries between the peer group and a specific neighbor statement, the neighbor statement wins.

Timers

BGP timers do not have to match. This is because BGP uses the lowest timers set between the two neighbors. Notice in Example 14-20 that R3 is configured with a minimum hold time of 90 seconds; if a neighbor is using more aggressive timers, those timers will not be used. The situation is far worse than the timers simply not being used. The neighbor relationship does not form at all.

Example 14-20 *Verifying BGP Timers*

```
R1# show bgp ipv4 unicast neighbors 10.1.13.3 | include hold time|holdtime
  Last read 00:00:02, last write 00:00:29, hold time is 90, keepalive interval is
30 seconds
R3# show bgp ipv4 unicast neighbors 10.1.13.1 | include hold time|holdtime
  Last read 00:00:10, last write 00:00:23, hold time is 90, keepalive interval is
30 seconds
  Configured hold time is 90, keepalive interval is 30 seconds
  Minimum holdtime from neighbor is 90 seconds
```

Timers (Cont.)

Refer to Example 14-21. In this case, R1 has a hello interval set to 10 and hold time set to 30. R3 has the minimum hold time set to 90 seconds. Therefore, R3 does not agree with the 30-second hold time set by R1, and the neighbor relationship fails. You can see in the output that a BGP notification states that the hold time is not acceptable.

Example 14-21 *Modifying BGP Timers to Values That Are Not Acceptable on R1*

```
R1# config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# router bgp 65501
R1(config-router)# neighbor 10.1.13.3 timers 10 30
R1(config-router)# do clear ip bgp 10.1.13.3
R1(config-router)#
%BGP-5-ADJCHANGE: neighbor 10.1.13.3 Down User reset
%BGP_SESSION-5-ADJCHANGE: neighbor 10.1.13.3 IPv4 Unicast topology base removed from
session User reset
%BGP-3-NOTIFICATION: received from neighbor 10.1.13.3 active 2/6 (unacceptable hold
time) 0 bytes
R1(config-router)#
%BGP-5-NBR_RESET: Neighbor 10.1.13.3 active reset (BGP Notification received)
%BGP-5-ADJCHANGE: neighbor 10.1.13.3 active Down BGP Notification received
%BGP_SESSION-5-ADJCHANGE: neighbor 10.1.13.3 IPv4 Unicast topology base removed from
session BGP Notification received
R1(config-router)#
%BGP-3-NOTIFICATION: received from neighbor 10.1.13.3 active 2/6 (unacceptable hold
time) 0 bytes
R1#
```

Troubleshooting BGP Routes

- After a BGP adjacency is formed, BGP routers exchange their BGP routes with each other. For various reasons, BGP routes might be missing from either the BGP table or the routing table.
- This section explains those reasons and how to identify and troubleshoot them.

Missing Routes

Some common reasons BGP routes might be missing from either the BGP table or the routing table:

- **Missing or bad network mask command -** An accurate network command is needed to advertise routes.
- **Next-hop router not reachable -** To use a BGP route, the next hop must be reachable.
- **BGP split-horizon rule -** A router that learns BGP routes through an iBGP peering does not share those routes with another iBGP peer.
- **Better source of information -** If exactly the same network is learned from a more reliable source, it is used instead of the BGP-learned information.
- **Route filtering -** A filter might be preventing a route from being shared with neighbors or learned from neighbors.

Missing Routes (Cont.)

To verify the IPv4 unicast BGP-learned routes or routes locally injected into the BGP table, you use the **show bgp ipv4 unicast** command as shown in Example 14-22. Routes appear in this table for the following reasons:

- Another BGP router advertises them to the local router.
- The **network mask** command matches a route in the local routing table.
- A **redistribute** command is used to import the route from another local source.
- The **summary-address** command is used to create a summary route.

Example 14-22 *Examining the BGP Table*

```
R1# show bgp ipv4 unicast
BGP table version is 10, 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,
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 ?
 *>  10.1.1.0/26      0.0.0.0                  0         32768 i
 *>  10.1.1.0/24      0.0.0.0                            32768 i
 *>  10.1.1.64/26     0.0.0.0                  0         32768 i
 *>  10.1.1.128/26    0.0.0.0                  0         32768 i
 *>  10.1.1.192/26    0.0.0.0                  0         32768 i
 *   10.1.5.0/24      10.1.13.3             3328             0 65502 i
 *>                   2.2.2.2               3328             0 65502 i
 *>  10.1.12.0/24     0.0.0.0                  0         32768 ?
 *>  10.1.13.0/24     0.0.0.0                  0         32768 ?
```

Missing Routes (Cont.)

To display the routing table, you use the **show ip route** command. To view only the BGP routes, you issue the command **show ip route bgp**, as shown in Example 14-23. All BGP routes appear with the code B at the beginning of each entry.

Example 14-23 *Examining the BGP Routes in the Routing Table*

```
R2# show ip route bgp
...output omitted...

Gateway of last resort is 10.1.12.1 to network 0.0.0.0

      10.0.0.0/8 is variably subnetted, 15 subnets, 3 masks
B        10.1.1.0/24 [20/0] via 1.1.1.1, 00:19:11
B        10.1.1.0/26 [20/0] via 1.1.1.1, 00:41:04
B        10.1.1.64/26 [20/0] via 1.1.1.1, 00:36:45
B        10.1.1.128/26 [20/0] via 1.1.1.1, 00:36:15
B        10.1.1.192/26 [20/0] via 1.1.1.1, 00:36:15
B        10.1.13.0/24 [20/0] via 1.1.1.1, 00:20:23
```

Bad or Missing Network Mask

The **network mask** command is used to advertise routes into BGP. If you only remember one thing about this command, remember that it is extremely picky:

- The network/prefix you want to advertise with BGP must be in the routing table from some other source (connected, static, or some other routing protocol).
- The network mask command must be a perfect match to the network/prefix listed in the routing table.
- It is important that you be able to recognize a bad or missing network mask command as being the reason for missing routes.

Example 14-24 *Determining Whether the 10.1.1.0/26 Network Is Advertised*

```
R1# config t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# router bgp 65501
R1(config-router)# network 10.1.1.0 mask 255.255.255.192
R1(config-router)# end
R1# show ip route
...output omitted...

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0
      2.0.0.0/32 is subnetted, 1 subnets
S        2.2.2.2 [1/0] via 10.1.12.2
      10.0.0.0/8 is variably subnetted, 12 subnets, 3 masks
C        10.1.1.0/26 is directly connected, GigabitEthernet0/0.1
L        10.1.1.1/32 is directly connected, GigabitEthernet0/0.1
C        10.1.1.64/26 is directly connected, GigabitEthernet0/0.2
L        10.1.1.65/32 is directly connected, GigabitEthernet0/0.2
C        10.1.1.128/26 is directly connected, GigabitEthernet0/0.3
L        10.1.1.129/32 is directly connected, GigabitEthernet0/0.3
C        10.1.1.192/26 is directly connected, GigabitEthernet0/0.4
L        10.1.1.193/32 is directly connected, GigabitEthernet0/0.4
C        10.1.12.0/24 is directly connected, GigabitEthernet1/0
L        10.1.12.1/32 is directly connected, GigabitEthernet1/0
C        10.1.13.0/24 is directly connected, GigabitEthernet2/0
L        10.1.13.1/32 is directly connected, GigabitEthernet2/0
```

Next-Hop Router Not Reachable

For a BGP router to install a BGP route in the routing table, it must be able to reach the next-hop address listed for the network. A ping 1.1.1.1 command issued on router R5 fails, proving that the next hop to the 10.1.1.0/26 is not reachable.

There are many different ways to solve this problem. The key is to train R5 about how to get to the next hop. The following are a few examples:

- Create a static default route on R2 and R3 and advertise it into the Interior Gateway Protocol (IGP) routing protocol.
- Create a static default route on R5.
- Create a static route on R5.
- Advertise the next-hop address into the IGP routing protocol.
- In addition, BGP has a built-in option. It is the **neighbor** *ip_address* **next-hop-self** command.

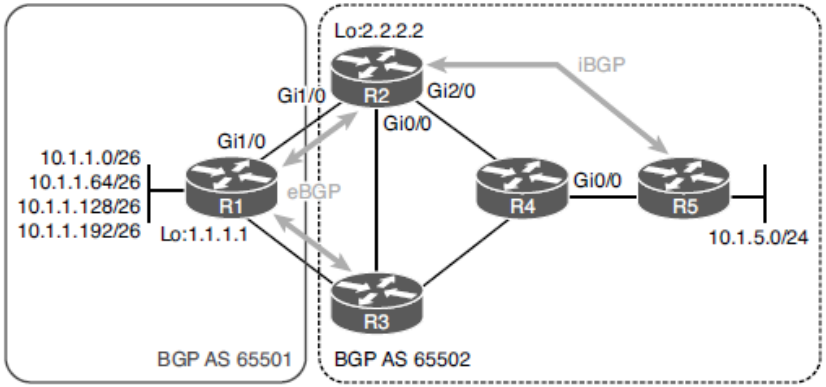

Figure 14-3 *Troubleshooting Next-Hop Address Behavior*

BGP Split-Horizon Rule

The BGP split-horizon rule states that a BGP router that receives a BGP route from an iBGP peering shall not advertise that route to another router that is an iBGP peer. For R5 to learn about the 10.1.1.0/26 network, it has to be an iBGP peer with the router that learned about the route from an eBGP peer or it has to be a peer with a route reflector.

Figure 14-5 indicates what the iBGP peerings should be to ensure that both R4 and R5 learn about 10.1.1.0/26 (as well as the other networks). This setup also ensures that redundancy is optimized in the BGP AS.

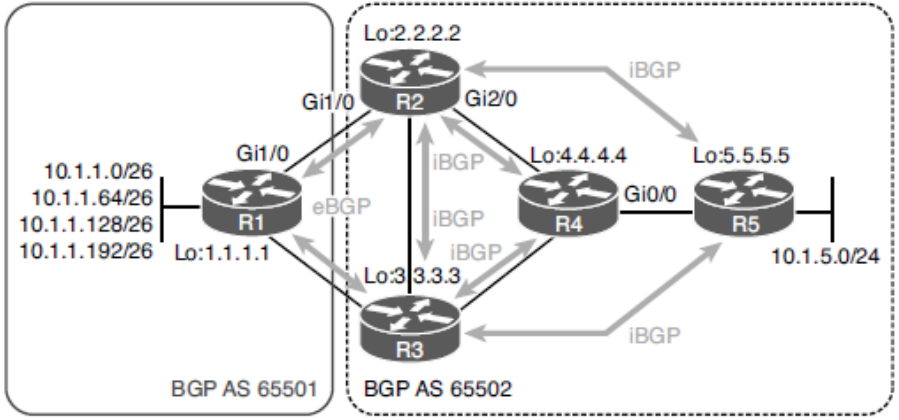

Figure 14-5 *Proper BGP Peerings to Avoid the BGP Split-Horizon Rule*

Better Source of Information

Example 14-30 shows the output of the IPv4 unicast BGP table on R5, using the **show bgp ipv4 unicast** command.

In the table, notice that the 10.1.5.0/24, 10.1.12.0/24, and 10.1.13.0/24 networks are best (installed in routing table), as indicated by the > symbol; however, they are not valid. They are listed as having a Routing Information Base (RIB) failure, as indicated by the *r*.

A RIB failure means that the BGP route was not able to be installed in the routing table; however, you can clearly see that the route is in the routing table because of the > symbol. In this case, the route in the routing table is from a better source.

Example 14-30 *Verifying BGP Routes*

```
R5# show bgp ipv4 unicast
BGP table version is 10, 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,
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      3.3.3.3                  0    100      0 65501 ?
 *>i                 2.2.2.2                  0    100      0 65501 ?
 * i 10.1.1.0/26     3.3.3.3                  0    100      0 65501 i
 *>i                 2.2.2.2                  0    100      0 65501 i
 * i 10.1.1.0/24     3.3.3.3                  0    100      0 65501 i
 *>i                 2.2.2.2                  0    100      0 65501 i
 * i 10.1.1.64/26    3.3.3.3                  0    100      0 65501 i
 *>i                 2.2.2.2                  0    100      0 65501 i
 * i 10.1.1.128/26   3.3.3.3                  0    100      0 65501 i
 *>i                 2.2.2.2                  0    100      0 65501 i
 * i 10.1.1.192/26   3.3.3.3                  0    100      0 65501 i
 *>i                 2.2.2.2                  0    100      0 65501 i
r i 10.1.5.0/24     3.3.3.3                3328    100      0 i
r>i                 2.2.2.2                3328    100      0 i
r i 10.1.12.0/24    3.3.3.3                  0    100      0 65501 ?
r>i                 2.2.2.2                  0    100      0 65501 ?
r i 10.1.13.0/24    3.3.3.3                  0    100      0 65501 ?
r>i                 2.2.2.2                  0    100      0 65501 ?
```

Better Source of Information (Cont.)

With regard to the 10.1.12.0/24 network, the output of **show bgp ipv4 unicast 10.1.12.0** in Example 14-32 indicates that it was learned from R2 and R3 using iBGP (internal), which has an AD of 200, much higher than EIGRP's AD.

You can verify why a route is experiencing a RIB failure by using the **show bgp ipv4 unicast rib-failure** command, as shown in Example 14-33. In this example, all three RIB failures are due to the BGP route having a higher AD.

Example 14-32 *Verifying Details of the BGP Routes*

```
R5# show bgp ipv4 unicast 10.1.12.0
BGP routing table entry for 10.1.12.0/24, version 50
Paths: (2 available, best #2, table default, RIB-failure(17))
  Not advertised to any peer
  Refresh Epoch 2
  65501
    3.3.3.3 (metric 131072) from 3.3.3.3 (3.3.3.3)
      Origin incomplete, metric 0, localpref 100, valid, internal
      rx pathid: 0, tx pathid: 0
  Refresh Epoch 2
  65501
    2.2.2.2 (metric 131072) from 2.2.2.2 (2.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      rx pathid: 0, tx pathid: 0x0
```

Example 14-33 *Verifying RIB Failures*

```
R5# show bgp ipv4 unicast rib-failure
  Network          Next Hop            RIB-failure   RIB-NH Matches
10.1.5.0/24        2.2.2.2          Higher admin distance        n/a
10.1.12.0/24       2.2.2.2          Higher admin distance        n/a
10.1.13.0/24       2.2.2.2          Higher admin distance        n/a
```

Route Filtering

When troubleshooting missing routes, you want to be able to determine whether a route filter is applied and whether it is the cause of the missing routes.

When a route is missing from the BGP table, check to see whether the route is being advertised before filters are applied.

As shown in Example 14-36, which displays the output of the **show bgp ipv4 unicast neighbors** *ip_address* **advertised routes** command, R2 and R3 are advertising the 10.1.13.0/24 network to R5.

Example 14-36 *Verifying Whether Routes Are Being Sent to R5*

```
R2# show bgp ipv4 unicast neighbors 5.5.5.5 advertised-routes
BGP table version is 10, 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,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 r> 1.1.1.1/32       1.1.1.1                  0             0 65501 ?
 *> 10.1.1.0/26      1.1.1.1                  0             0 65501 i
 *> 10.1.1.0/24      1.1.1.1                  0             0 65501 i
 *> 10.1.1.64/26     1.1.1.1                  0             0 65501 i
 *> 10.1.1.128/26    1.1.1.1                  0             0 65501 i
 *> 10.1.1.192/26    1.1.1.1                  0             0 65501 i
 *> 10.1.5.0/24      10.1.24.4             3328         32768 i
 r> 10.1.12.0/24     1.1.1.1                  0             0 65501 ?
 *> 10.1.13.0/24     1.1.1.1                  0             0 65501 ?

Total number of prefixes 9

R3# show bgp ipv4 unicast neighbors 5.5.5.5 advertised-routes
BGP table version is 10, 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,
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       10.1.13.1                0             0 65501 ?
 *> 10.1.1.0/26      10.1.13.1                0             0 65501 i
 *> 10.1.1.0/24      10.1.13.1                0             0 65501 i
 *> 10.1.1.64/26     10.1.13.1                0             0 65501 i
 *> 10.1.1.128/26    10.1.13.1                0             0 65501 i
 *> 10.1.1.192/26    10.1.13.1                0             0 65501 i
 *> 10.1.5.0/24      10.1.34.4             3328         32768 i
 *> 10.1.12.0/24     10.1.13.1                0             0 65501 ?
 r> 10.1.13.0/24     10.1.13.1                0             0 65501 ?

Total number of prefixes 9
```

Route Filtering (Cont.)

The **show ip protocols** command, as shown in Example 14-37, displays the incoming filter applied to the BGP autonomous system. It is a distribute list using the prefix list called FILTER_10.1.13.0/24. The prefix list is denying 10.1.13.0/24 and permitting all other routes.

This example focuses on a filter that applies to the entire BGP process. No matter which router the route 10.1.13.0/24 is received from, it is denied. You can apply a filter directly to a neighbor by using any one of the following commands:

- **neighbor** *ip_address* **distribute-list** *access_list_number* **{in | out}**
- **neighbor** *ip_address* **prefix-list** *prefix_list_name* **{in | out}**
- **neighbor** *ip_address* **route-map** *map_name* **{in | out}**
- **neighbor** *ip_address* **filter-list** *access_list_number* **{in | out}**

Example 14-37 *Verifying Whether Filters Are Applied to R5*

```
R5# show ip protocols
...output omitted...

Routing Protocol is "bgp 65502"
 Outgoing update filter list for all interfaces is not set
 Incoming update filter list for all interfaces is (prefix-list) FILTER_10.1.13.0/24
 IGP synchronization is disabled
 Automatic route summarization is disabled
 Neighbor(s):
   Address FiltIn FiltOut DistIn DistOut Weight RouteMap
   2.2.2.2
   3.3.3.3
 Maximum path: 1
 Routing Information Sources:...output omitted...

R5# show ip prefix-list
ip prefix-list FILTER_10.1.13.0/24: 2 entries
 seq 5 deny 10.1.13.0/24
 seq 10 permit 0.0.0.0/0 le 32

R5# show run | include bgp 65502|distribute-list
router bgp 65502
 distribute-list prefix FILTER_10.1.13.0/24 in
```


Route Filtering (Cont.)

In Example 14-38, an inbound distribute list is applied directly to the neighbor 2.2.2.2, as shown in the **show ip protocols** output. Notice that only the first six characters of the ACL are identified.

You can apply a route map, a prefix list, and a filter list directly to the neighbor command. The filter list appears under the FiltIn and FiltOut columns in the output of **show ip protocols**, and the route map appears under the RouteMap column in the **show ip protocols output**. If the prefix list is applied directly to a neighbor statement, it does not appear in the output of **show ip protocols**. You need to review the output of **show bgp ipv4 unicast neighbors**. To avoid the verbose output, use this shortcut:
show bgp ipv4 unicast neighbors *ip_address* **| include prefix|filter|Route map**

Example 14-38 *Verifying a Distribute List Applied to a Neighbor*

```
R5# show ip protocols
...output omitted...

Routing Protocol is "bgp 65502"
 Outgoing update filter list for all interfaces is not set
 Incoming update filter list for all interfaces is not set
 IGP synchronization is disabled
 Automatic route summarization is disabled
 Neighbor(s):
   Address FiltIn FiltOut DistIn DistOut Weight RouteMap
   2.2.2.2                FILTER
   3.3.3.3
 Maximum path: 1
 Routing Information Sources:
...output omitted...

R5# show run | include bgp 65502|distribute-list
router bgp 65502
 neighbor 2.2.2.2 distribute-list FILTER_10.1.13.0/24 in

R5# show ip access-lists
Standard IP access list FILTER_10.1.13.0/24
 10 deny 10.1.13.0, wildcard bits 0.0.0.255
 20 permit any
```


Troubleshooting BGP Path Selection

- Unlike OSPF and EIGRP, BGP does not consider a link's bandwidth when making a route decision.
- BGP uses various attributes when deciding which path is the best.
- This section discusses the BGP best-path decision-making process. In addition, it examines private ASNs.

The Best-Path Decision-Making Process

Cisco routers review BGP attributes in the following order when deciding which path is the best:

1. Prefer the highest weight
2. Prefer the highest local preference
3. Prefer the route originated by the local router
4. Prefer the path with the shorter Accumulated Interior Gateway Protocol (AIGP) metric attribute
5. Prefer the shortest AS_Path
6. Prefer the lowest origin code
7. Prefer the lowest multi-exit discriminator (MED)
8. Prefer an external path over an internal path
9. Prefer the path through the closest IGP neighbor
10. Prefer the oldest route for eBGP paths
11. Prefer the path with the lowest neighbor BGP RID
12. Prefer the path with the lowest neighbor IP address

The Best-Path Decision-Making Process (Cont.)

When BGP finds a match, it stops and uses that attribute as the reason for choosing the path as the best—and it looks no further. In addition, if the next-hop IP address is not reachable, the router does not even go through the following process because it considers the next hop inaccessible:

Step 1. BGP first looks at weight. Higher is better. If the weight is tied, the next attribute is checked.

Step 2. Local preference is checked next. Higher is better. If local preference is tied, the next attribute is checked.

Step 3. The router checks whether it generated the BGP route. If it did, it is preferred. If it did not generate any of the routes, the next attribute is checked.

Step 4. AIGP is checked next only if it's configured to be used, if not, then the next attribute is checked.

The Best-Path Decision-Making Process (Cont.)

Step 5. AS_Path is checked next. The shortest path is preferred. If the AS_Path is tied, the next attribute is checked.

Step 6. The origin code is checked next. IGP is better than EGP (the predecessor to BFP), which is better than incomplete. IGP means the route was generated with the **network mask** or **summary-address,** incomplete means the route was redistributed into BGP. If the origin code is the same, the next attribute is checked.

Step 7. MED (metric) is next. Lower is better. If the MED (metric) is the same for both, the next attribute has to be checked.

Step 8. Now eBGP is preferred over iBGP. If this attribute is tied as well, and the next has to be checked.

Step 9. The IGP path to the neighbor is compared now. If the metrics are the same, the next attribute has to be checked.

The Best-Path Decision-Making Process (Cont.)

Step 10. If they are eBGP paths, the ages of the routes are checked. If both paths are iBGP paths, the next attribute is checked.

Step 11. The BGP RIDs are now compared. Lower is better. If the RID is tied, the path through the neighbor with the lower IP address wins.

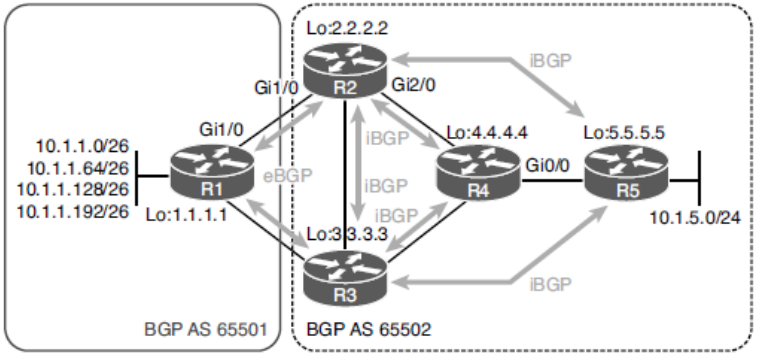

Figure 14-6 *BGP Best-Path Decision-Making Process Topology*

Private Autonomous System Numbers

- Like IPv4 addresses, BGP ASNs also have a private range. The 2-byte AS range is 64,512 to 65,534, and the 4-byte AS range is 4,200,000,000 to 4,294,967,294.
- These ASNs can be used for networks that are single-homed or dual-homed to the same ISP, thereby preserving the public ASNs for networks that are multihomed to multiple ISPs.
- It is imperative that the private ASN not be in the AS_Path attribute when the routes are advertised to the Internet (in the global BGP table) because multiple ASs could be using the same private ASN, which would cause issues on the Internet.
- If private ASNs are being sent into the global BGP table, they need to be stopped. You can accomplish this by using the **neighbor** *ip_address* **remove-private-as command**.

Using debug Commands

The majority of changes that occur with BGP generate syslog messages in real time. Therefore, you are notified through syslog if any neighbor issues occur. Avoid using the large number of debugs that are available because they place a lot of pressure on the routers' resources.

Here are a few debug commands that may be useful:

- **debug ip routing** – The output from this command shows updates to a router's IP routing table.
- **debug ip bgp** - This command can be useful in watching real-time state changes for IPv4 BGP peering relationships.
- **debug ip bgp updates** - This command produces more detailed output than the **debug ip bgp** command. Specifically, you can see the content of IPv4 BGP updates.

Troubleshooting BGP for IPv6

- BGP for IPv4 and BGP for IPv6 are configured in the same BGP autonomous system configuration mode, known as Multiprotocol BGP (MP-BGP).
- Implementing BGP for IPv4 and IPv6 on the same router requires the use of address families and the activation of neighbors for those address families.
- This section examines the additional issues that you might encounter when using MP-BGP with IPv4 and IPv6 unicast routes.

MP-BGP

There are two different ways to exchange IPv6 routes with BGP. You can exchange them over IPv4 TCP sessions or over IPv6 TCP sessions. Example 14-45 shows a sample BGP configuration in which IPv6 routes are exchanged over an IPv4 TCP session. Notice that there are two address families: one for IPv4 unicast, and one for IPv6 unicast.

The neighbors and remote ASNs are identified outside the address family (AF) configuration. You then activate the neighbor within the AF with the **neighbor** *ip_address* **activate** command.

In this example, the IPv6 AF is using an IPv4 neighbor address to establish the TCP session. Therefore, the TCP session is IPv4 based.

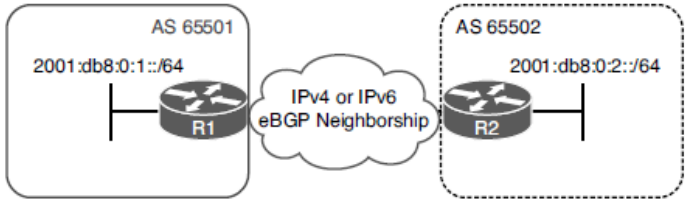

Figure 14-8 *MP-BGP Topology*

Example 14-45 *MP-BGP Configuration for IPv6 Routes over an IPv4 TCP Session*

```
R1# show run | s router bgp
router bgp 65501
 bgp log-neighbor-changes
 neighbor 2.2.2.2 remote-as 65502
 neighbor 2.2.2.2 ebgp-multihop 2
 neighbor 2.2.2.2 password CISCO
 neighbor 2.2.2.2 update-source Loopback0
!
 address-family ipv4
  network 10.1.1.0 mask 255.255.255.192
  network 10.1.1.64 mask 255.255.255.192
  network 10.1.1.128 mask 255.255.255.192
  network 10.1.1.192 mask 255.255.255.192
  aggregate-address 10.1.1.0 255.255.255.0
  redistribute connected
  neighbor 2.2.2.2 activate
  exit-address-family
 !
 address-family ipv6
  network 2001:DB8:1::/64
  neighbor 2.2.2.2 activate
  exit-address-family
```

MP-BGP (Cont.)

To verify the IPv6 unicast routes that have been learned from all neighbors, you can issue the **show bgp ipv6 unicast** command, as shown in Example 14-47. Its output displays the IPv6 BGP table.

Examine the 2001:db8:2::/64 route. This is the route that was learned from R2 (the 2.2.2.2 neighbor). It is not installed in the routing table, as indicated by the absence of the *>.

The address ::FFFF:2.2.2.2 is a dynamically generated next hop that was created to replace the original next hop of 2.2.2.2. An IPv6 route cannot have an IPv4 next-hop address.

To solve this issue, create a route map on the advertising router that changes the next hop to a valid IPv6 address and attach it to the neighbor statement, as shown in Example 14-48.

Example 14-47 *Verifying MP-BGP IPv6 Unicast Routes in the IPv6 BGP Table*

```
R1# show bgp ipv6 unicast
BGP table version is 2, 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,
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::/64  ::                        0         32768 i
 *   2001:DB8:2::/64  ::FFFF:2.2.2.2           0             0 65502 i
```

Example 14-48 *Modifying the BGP Next Hop*

```
R2# config t
Enter configuration commands, one per line. End with CNTL/Z.
R2(config)# route-map CHANGE_NH permit 10
R2(config-route-map)# set ipv6 next-hop 2001:db8:12::2
R2(config-route-map)# exit
R2(config)# router bgp 65502
R2(config-router)# address-family ipv6 unicast
R2(config-router-af)# neighbor 1.1.1.1 route-map CHANGE_NH out
```

MP-BGP (Cont.)

As shown in Example 14-50, to form the IPv6 TCP session, define the neighbor by using the **neighbor** *ipv6_address* **remote-as** *autonomous_system_number* command outside the AF configuration, and then you activate the neighbor in the IPv6 AF configuration by using the **neighbor** *ipv6_address* **activate** command.

Example 14-50 *MP-BGP Configuration for IPv6 Routes over an IPv6 TCP Session*

```
R1# show run | section router bgp
router bgp 65501
 bgp log-neighbor-changes
  neighbor 2.2.2.2 remote-as 65502
  neighbor 2.2.2.2 ebgp-multihop 2
  neighbor 2.2.2.2 password CISCO
  neighbor 2.2.2.2 update-source Loopback0
  neighbor 10.1.13.3 remote-as 65502
  neighbor 2001:DB8:12::2 remote-as 65502
  !
  address-family ipv4
   network 10.1.1.0 mask 255.255.255.192
   network 10.1.1.64 mask 255.255.255.192
   network 10.1.1.128 mask 255.255.255.192
   network 10.1.1.192 mask 255.255.255.192
   aggregate-address 10.1.1.0 255.255.255.0
   redistribute connected
   neighbor 2.2.2.2 activate
   neighbor 10.1.13.3 activate
   no neighbor 2001:DB8:12::2 activate
  exit-address-family
  !
  address-family ipv6
   network 2001:DB8:1::/64
   neighbor 2001:DB8:12::2 activate
  exit-address-family
```

MP-BGP (Cont.)

The output of **show bgp ipv6 unicast summary**, as shown in Example 14-51, indicates that R1 has formed an IPv6 BGP neighbor adjacency with the device at 2001:db8:12::2 using an IPv6 TCP session, and one prefix has been received.

The IPv6 BGP table, as displayed in the output of the **show bgp ipv6 unicast** command in Example 14-52, indicates that 2001:DB8:2::/64 can be reached with a next hop of 2001:DB8:12::2 and that it is installed in the routing table, as indicated by the *>.

Example 14-51 *MP-BGP Adjacencies with IPv6 TCP Sessions*

```
R1# show bgp ipv6 unicast summary
BGP router identifier 1.1.1.1, local AS number 65501
BGP table version is 5, main routing table version 5
2 network entries using 336 bytes of memory
2 path entries using 208 bytes of memory
2/2 BGP path/bestpath attribute entries using 272 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 840 total bytes of memory
BGP activity 12/1 prefixes, 22/10 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001:DB8:12::2  4        65502       5       5        4    0    0 00:00:05        1
```

Example 14-52 *Verifying the IPv6 BGP Table*

```
R1# show bgp ipv6 unicast
BGP table version is 5, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
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::/64  ::                        0         32768 i
 *>  2001:DB8:2::/64  2001:DB8:12::2           0             0 65502 i
```

BGP Trouble Tickets

- This section presents trouble tickets related to the topics discussed earlier in this chapter.
- The purpose of these trouble tickets is to show a process that you can use when troubleshooting in the real world or in an exam environment.

Trouble Ticket 14-1: BGP Routes

Problem: You are the administrator for BGP AS 65502. While you were away on vacation, the link between R1 and R2 failed. When the link between R1 and R2 fails, the link between R1 and R3 is supposed to forward traffic to BGP AS 65501. However, that did not occur while you were away. Your co-worker had to restore connectivity between R1 and R2, and complaints kept flowing in from the users in 10.1.5.0/24 about connectivity to the 10.1.1.0/24
networks being down.

Refer to your text for next steps and examples to troubleshoot and resolve this trouble ticket.

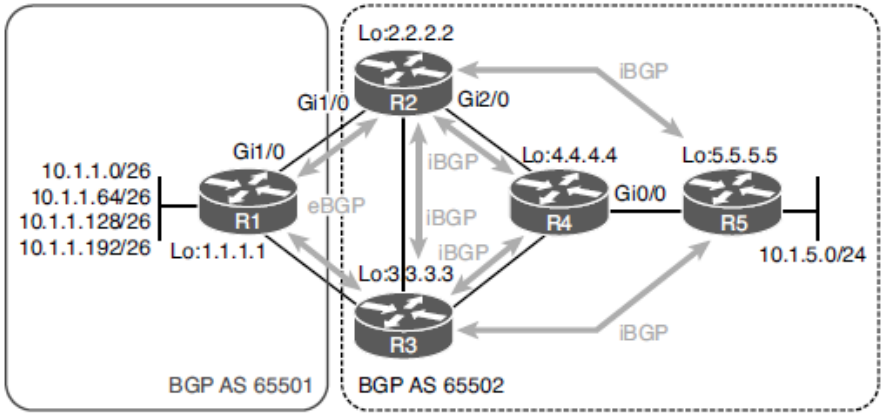

Figure 14-9 *BGP Trouble Tickers Topology*

Trouble Ticket 14-2: Layer 3 Connectivity

Problem: You are the administrator for BGP AS 65501. Users in the 10.1.1.0/26 and 10.1.1.64/26 networks have indicated that they are not able to access resources located at 10.1.5.5. However, they can access resources locally.

You begin troubleshooting by issuing two pings on R1 to 10.1.5.5 and sourcing them from 10.1.1.1 and 10.1.1.65.

Refer to your text for next steps and examples to troubleshoot and resolve this trouble ticket.

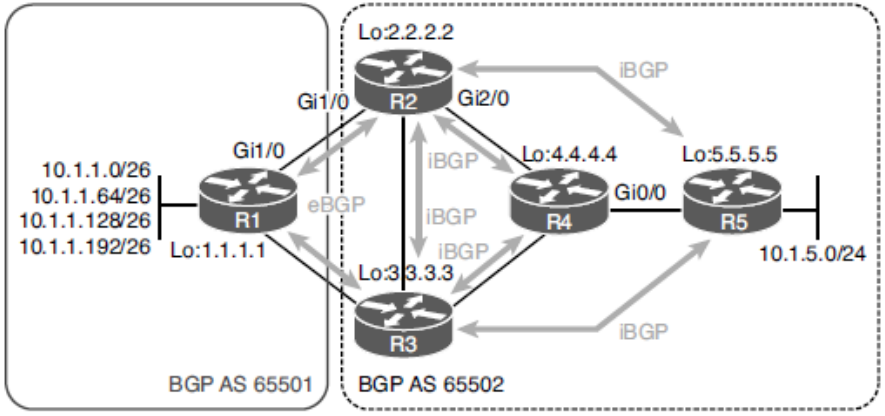

Figure 14-9 *BGP Trouble Tickers Topology*

Trouble Ticket 14-3: Backup Link

Problem: You are the administrator for BGP AS 65502. Traffic reports indicate that all traffic out of the autonomous system is flowing through R3 and across the backup link. This is undesirable unless the link between R2 and R1 fails.

To verify the issue, you use traceroute from R5.

Refer to your text for next steps and examples to troubleshoot and resolve this trouble ticket.

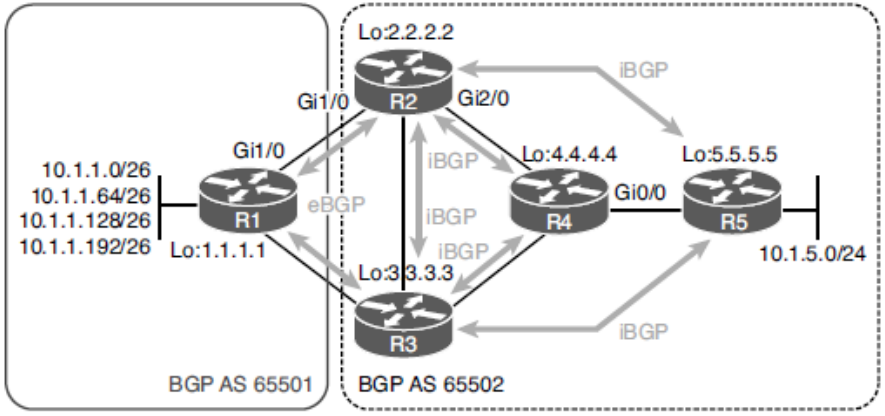

Figure 14-9 *BGP Trouble Tickers Topology*

MP-BGP Trouble Ticket

- This section presents trouble tickets related to the topics discussed earlier in this chapter.
- The purpose of these trouble tickets is to show a process that you can use when troubleshooting in the real world or in an exam environment.

Trouble Ticket 14-4: MP-eBGP

Problem: You are an administrator of BGP AS 65501. Another administrator in your AS has asked you for help. The default route from your ISP is not being learned by your router (R1) using BGP. As a result, no one in your AS is able to reach the Internet.

You start by confirming the issue by using the show ipv6 route command on R1.

Refer to your text for next steps and examples to troubleshoot and resolve this trouble ticket.

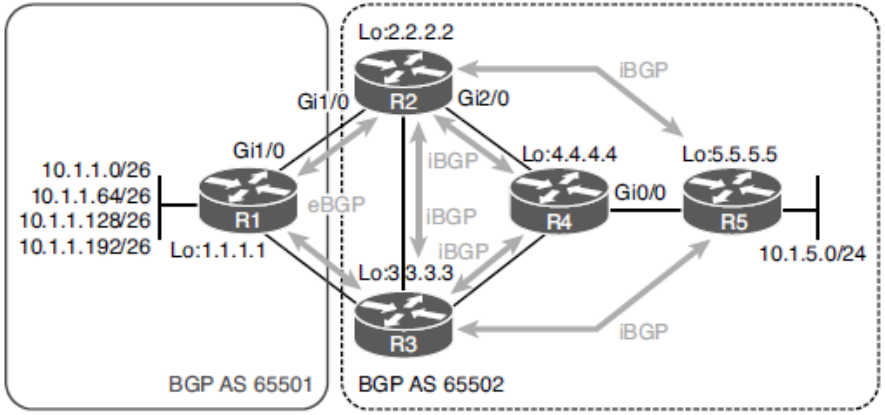

Figure 14-9 *BGP Trouble Tickers Topology*

Prepare for the Exam

Prepare for the Exam

Key Topics for Chapter 14

Description
Verifying BGP neighbors with **show bgp ipv4 unicast summary**
Considerations when troubleshooting BGP neighbor relationships
The path to the neighbor through the default route
Incorrect neighbor statement
How to control the source addresses of BGP packets
How BGP TCP sessions are formed and how you can control the server and client for the TCP session
Manipulating the TTL of an eBGP packet
How the minimum hold time parameter can prevent BGP neighbor relationships
The reasons a BGP route might be missing from the BGP table or the routing table

Key Topics for Chapter 14 (Cont.)

Description
Examining the BGP table
The requirements of the BGP network mask command
The BGP next-hop issue
Identifying BGP split-horizon issues
Troubleshooting filters that may be preventing BGP routes from being advertised or learned
The steps that BGP uses to successfully determine the best path to reach a given network
The next-hop issue that occurs when exchanging IPv6 BGP routes over IPv4 BGP TCP sessions
Solving the next-hop issue that occurs when IPv6 BGP routes are exchanged over IPv4 BGP TCP sessions

Key Terms for Chapter 14

Terms	
BGP	TTL
EGP	Peer group
eBGP	Split-horizon rule (iBGP)
iBGP	Weight
MP-BGP	Local Preference
ISP	AS_Path
Address family	MED

Command Reference for Chapter 14

Task	Command Syntax
Display a router's BGP RID, ASN, information about the BGP's memory usage, and summary information about IPv4/IPv6 unicast BGP neighbors	**show bgp {ipv4 | ipv6} unicast summary**
Display detailed information about all the IPv4/IPv6 BGP neighbors of a router	**show bgp {ipv4 | ipv6} unicast neighbors**
Display the IPv4/IPv6 network prefixes present in the IPv4/IPv6 BGP table	**show bgp {ipv4 | ipv6} unicast**
Show routes known to a router's IPv4/IPv6 routing table that were learned from BGP	**show {ipv4 | ipv6} route bgp**
Show real-time information about BGP events, such as the establishment of a peering relationship	**debug ip bgp**
Show real-time information about BGP updates sent and received by a BGP router	**debug ip bgp updates**
Display updates that occur in a router's IP routing table	**debug ip routing**


CISCO