A Case Study in BGP
The Border Gateway Protocol (BGP) is an inter-Autonomous System (AS) routing protocol and its primary function is to exchange network reachability information with other BGP speakers. The following case study provides complex BGP configuration scenarios to aid the student preparing for the CCIE Lab.

**Overview**

BGP is a path vector inter-AS routing protocol that is based on distance vector algorithms. (For the purposes of this chapter, *inter* means routing between entities and *intra* means routing within an entity.) An AS is a collection of routers or end stations that are under the same administrative control and viewed as a single entity. The reason BGP is called a path vector protocol is that the BGP routing information carries a sequence of AS numbers, which indicate the path the route has traversed. This information is used to construct a graph of AS connectivity from which routing loops can be pruned.

The interior gateway routing protocols (RIP, OSPF, IGRP, EIGRP) were designed to operate in a single AS that is under a single administrative control. BGP was introduced to facilitate a loop-free exchange of routing information between ASs while controlling the expansion of routing tables through Classless Inter-Domain Routing (CIDR) and providing a structured view of the Internet through the use of ASs.

In a sense, the Internet could have been a large Open Shortest Path First (OSPF) network. If this were the case, then all organizations that participated in it would have to adhere to the same administrative policies. By segregating the Internet into multiple ASs, one can create one large network that consists of smaller, more manageable networks. Within these smaller networks, or ASs, an organization's unique rules and administrative policies can be applied. A unique number assigned by an Internet registry identifies each AS.

Figure 2-1 displays two Internet service providers (ISPs), Xnet and Ynet, each of which consists of multiple networks running multiple Interior Gateway Protocols (IGPs). Each service provider is assigned an AS number by an Internet registry that represents its entire network. When Company Xnet and Ynet want to exchange routing information, they do so using BGP.

Company Ynet advertises networks 2.0.0.0 and 3.0.0.0 to company Xnet, and the routes are marked as originating from AS 200. Company Xnet does not need to have a full topological view of company Ynet nor does it need to understand Ynet's internal routing or policies. It simply knows that networks 2.0.0.0 and 3.0.0.0 are in AS 200.
BGP Terminology

Before diving into the intricate details of BGP, it is important to have a clear understanding of key terms and concepts, some of which are used interchangeably.

**External BGP (EBGP) versus Internal BGP (IBGP)** Although BGP was designed to be used between ASs (EBGP), it is often used within an AS (IBGP) to carry information between border routers running EBGP to other ASs. This enables all BGP path attributes to be maintained across the AS.

By definition, an EBGP neighbor is a router with an administrative and policy control that is outside of your AS. An IBGP neighbor is a router that is under the same administrative control (see Figure 2-2).

**Classless Interdomain Routing (CIDR)** This was developed to address the explosive growth of IP addresses present in IP routing tables on Internet routers and the exhaustion of IP address space. CIDR is an address allocation scheme that eliminates the concept of a network class within BGP. In CIDR, an IP network is represented by a prefix, which is the IP address and a number that indicates the leftmost contiguous significant bits in the address. For example, in Figure 2-3, 256 Class C networks are present on service provider A’s network. Without CIDR, the service provider must advertise each network individually. Using CIDR, service provider A can advertise all of these networks with one classless advertisement (200.10.0.0 /16), as shown in Figure 2-4.

**Supernet** A supernet is a network advertisement whose prefix boundary contains fewer bits than the network’s natural mask. For example, in Figure 2-4, the natural network mask for the Class C network 200.10.1.0 is 255.255.255.0. However, when you represent it as 200.10.0.0 /16, the mask is 16, which is less than 24. Hence, it is termed a supernet advertisement.
IP Prefix  An IP prefix is an IP network address along with an indication of the number of bits that makes up the network number. For example, 10.0.0.0 /8 is an IP prefix.

Network Layer Reachability Information (NLRI)  This is how BGP supports classless routing (CIDR). The NLRI is part of the BGP update message and is used to list a set of destinations that is reachable. The NLRI field in the BGP update message contains two tuples: [length, prefix]. The length is the number of bits in the mask and the prefix is the IP address. The two combined represent the network number. For example, the network 10.0.0.0 /8 would be advertised in the NLRI field of a BGP update message as<8,10.0.0.0>.
A Case Study in BGP

**Autonomous System (AS)**  An AS is a group of routers or hosts that are under the same administrative control and policies. AS numbers are assigned by an Internet registry.

**Synchronization**  Before BGP can announce a route, the route must be present in the IP routing table. In other words, BGP and IGP must be in sync before the networks can be advertised. Cisco enables BGP to override the synchronization requirement with the command `no synchronization`. This allows BGP to announce routes that are known via BGP but are not in the routing table. The reason that this rule exists is that it is important for the AS to be consistent with the routes it advertises.

In Figure 2-5, RouterA and RouterB are the only routers running BGP. If synchronization is disabled on RouterB, it will advertise network 1.0.0.0 /8 to AS 200. When RouterD wants to send traffic to network 1.0.0.0, it sends the packet to RouterB, which does a recursive lookup in its IP routing table and forwards the packet to RouterC. Because RouterC is not running BGP, it has no visibility to network 1.0.0.0 and therefore drops the packet. This is why BGP requires synchronization between BGP and IGP. Care must be taken when disabling synchronization. If an AS is a transit AS, all routes should be running fully meshed IBGP before synchronization is disabled.

---

**Figure 2-5**
Synchronization within an AS

![Figure 2-5 Synchronization within an AS](image-url)
Technology Overview

BGP is an inter-AS routing protocol whose primary function is to exchange network reachability information with other BGP speakers (a BGP speaker is any device that is configured for BGP). BGP uses the Transmission Control Protocol (TCP) as its transport protocol (port 179) that provides reliable data transfer between the BGP speakers.

Two BGP routers form a transport protocol connection between one another. The two routers are called neighbors or peers. Once the transport connection is formed, the peer routers exchange messages to open and confirm connection parameters. It is in this stage that the routers exchange information on the BGP version number, the AS number, the hold time, the BGP identifier, and other optional parameters. If the peers disagree on any of the parameters, a notification error is sent and the peer connection does not get established.

If the peer routers agree upon the parameters, then the entire BGP routing table is exchanged using UPDATE messages. The UPDATE messages contain a list of destinations reachable via each system (NLRI) along with path attributes for each route. The path attributes contain information such as the origin of the route and the degree of preference. Path attributes will be covered in great detail later in this chapter.

The BGP table is valid for each peer for the duration of the BGP connection. If any routing information changes, the neighbor router uses incremental updates to convey this information. BGP does not require that routing information be periodically refreshed. If no routing changes occur, BGP peers only exchange keepalive packets, which are sent periodically to ensure that the connection is kept active.

Case Study 1: BGP

IOS Requirements

BGP-4 first became available in IOS release 10.0, but this case study was performed using IOS 11.2.

Equipment Needed

The following equipment is needed to perform this case study:

- One Catalyst 1900 Ethernet switch
- One terminal server with one Ethernet and one serial port
A Case Study in BGP

- One Cisco router with one serial port and one Ethernet port
- Three Cisco routers with two serial ports and one Ethernet port
- One Cisco router with one Ethernet port and three serial ports
- One Cisco router with one Ethernet port and four serial ports
- One peripheral router with one Ethernet port and three serial ports. This router (R8) will be used as the Frame Relay switch as well as the BGP neighbor. (The student does not configure this router and should use the configuration provided on the CD-ROM.) Table 2-1 displays each router’s interface requirements.
- Cisco IOS 11.2
- A PC running a terminal emulation program for connecting to the console port of the terminal server
- Six Ethernet cables
- Seven Cisco DTE/DCE crossover cables
- One Cisco Rolled cable

<table>
<thead>
<tr>
<th>Table 2-1</th>
<th>Interface requirements for each router</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>1 Serial, 2 Ethernet</td>
</tr>
<tr>
<td>R2</td>
<td>2 Serial, 1 Ethernet</td>
</tr>
<tr>
<td>R3</td>
<td>4 Serial, 1 Ethernet</td>
</tr>
<tr>
<td>R4</td>
<td>2 Serial</td>
</tr>
<tr>
<td>R5</td>
<td>2 Serial, 1 Ethernet</td>
</tr>
<tr>
<td>R6</td>
<td>1 Serial, 1 Ethernet</td>
</tr>
<tr>
<td>R7</td>
<td>1 Serial, 1 Ethernet (terminal server)</td>
</tr>
<tr>
<td>R8</td>
<td>3 Serial, 1 Ethernet (Frame Relay switch)</td>
</tr>
</tbody>
</table>
Physical Connectivity Diagram

Figure 2-6 shows the physical connectivity for the routers in this case study.
A Case Study in BGP

Notes

- All IP addresses will be assigned from the class B network 152.1.0.0.
- No static or default routes should be used in this case study unless explicitly stated.
- Use Process ID 64 for all OSPF configurations.
- Do not use the network point-to-multipoint command under OSPF.
- No configuration is needed on R8, load the configuration that is provided on the CD.
- Read through the entire case study before beginning.
- Make sure you save your configurations regularly.

Questions

1. Cable the routers according to Figure 2-6 and load the configuration provided on the CD to R8. The configuration provided is from a 4500 with an Ethernet interface and a four-port serial network module. If you plan on using a router other than a 4500 or one that has a different way of numbering its interfaces, such as a 3620, the interfaces provided in the configuration will need to be renamed.

2. Erase any existing configuration from the routers and set each router’s hostname. The name will be the corresponding router number. For example, router 1 will be R1.

3. Set the enable password on each router to be *cisco*.

4. Configure the terminal server (R7) so that all routers can be accessed by name through the terminal server. The port number of the terminal server attaches to the corresponding router. That is, port 1 of the terminal server goes to the console port of R1, and the Catalyst switch goes to port 9. (Use the Ethernet IP address as the IP address for reverse telneting.)

5. Place Ethernet interface 0 of R5, R6, R7, and R8 in Virtual Local Area Network (VLAN) 1. The routers should be attached in ascending order, starting with port 1 on the Catalyst switch.

6. Configure the router so that if an incorrect command is entered, the router does not try to translate it to an IP address.

7. The Mac addresses of the Ethernet interfaces attached to the VLAN should have the following format (routernumber.ASnumber.0000). For example, R7 would be set to 0007.0400.0000.
8. Configure the Catalyst so that only the designated router can attach to the specific port.
9. Verify that step 8 works by swapping the cables between port 1 and port 2 of the Catalyst switch. The ports should go into blocking mode.
10. Normalize the port.
11. Assign IP addresses to Ethernet 0 of R5, R6, R7, and R8 from network 152.1.1.0. Do not allocate more addresses than needed; use the first available subnet and assign the addresses in ascending order starting with R5.
12. Place Ethernet 0 of R1 and R3 in VLAN 2; place Ethernet 1 of R1 and Ethernet 0 of R2 in VLAN 3.
13. Connect the rest of the routers serially as per Figure 2-6. Make sure that the correct ports are configured for DTE/DCE. The DCE clock will be set to 64K on all routers.
14. R3 and R2 are connected via Frame Relay to R1. Use physical interfaces on R3 and R2 and a logical interface on R1. Assign addresses to each interface using subnet 152.1.10.0. Do not allocate more addresses than needed; use the first subnet available and assign the addresses in ascending order, starting with R1. Configure the interfaces for Frame Relay. Make sure that the only DLCIs that are used are the ones provided. Make sure that all routers can reach each other.
The following is the physical connectivity of R1, R2, and R3 to the Frame Relay Switch (R8).

<table>
<thead>
<tr>
<th>Frame Switch (R8) interface S0</th>
<th>R1 Interface S0/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Switch (R8) interface S0</td>
<td>R2 Interface S0</td>
</tr>
<tr>
<td>Frame Switch (R8) interface S0</td>
<td>R3 Interface S0/1</td>
</tr>
</tbody>
</table>

15. Assign IP addresses to Ethernet 0 of R3 and R1 using the next available subnet in 152.1.8.0. The subnet should contain 48 host addresses, R1 should use the first address in the subnet, and R3 should use the last one.
16. Assign IP addresses to Ethernet 1 of R1 and Ethernet 0 of R2 using the first available subnet in 152.1.9.0. The subnet should contain 128 host addresses. R1 should use the first address in the subnet and R2 should use the last one.
17. Assign IP addresses to Serial 0 on R7 and serial S0/0 on R3 using the first available subnet in 152.1.20.0. The address should be assigned out in ascending order, starting with R3.
18. Assign IP address to the serial interfaces connecting R3 and R4, use the first available subnet in 152.1.12.0. The subnet should contain the minimal hosts needed and the IP addresses should be assigned in ascending order, staring with R3.
19. Assign IP addresses to the serial interfaces connecting R3 and R5, use the next available subnet in 152.1.12.0. The subnet should contain the minimal hosts needed and the IP addresses should be assigned in ascending order, staring with R3.
A Case Study in BGP

20. Assign an IP address to Serial 1 on R4 and R5 using the next available subnet in 152.1.12.0. The subnet should contain the minimal hosts needed and the IP addresses should be assigned in ascending order, starting with R4.

21. Assign an IP address to Serial 1 of R2 and Serial 0 of R6 using the first available subnet in 152.1.11.0. The subnet should contain the minimal hosts needed and the IP addresses should be assigned in ascending order, starting with R2.

22. Assign the following loopback addresses to the corresponding routers:
   - R7 Loopback 0 152.1.20.7 /32
   - R7 Loopback 1 152.1.20.17 /28
   - R4 Loopback 0 152.1.12.4 /32
   - R4 Loopback 0 152.1.13.1 /24
   - R5 Loopback 0 152.1.12.5 /32
   - R5 Loopback 1 152.1.14.1 /29
   - R3 Loopback 0 152.1.10.3 /32
   - R2 Loopback 0 152.1.10.2 /32
   - R1 Loopback 0 152.1.10.1 /32
   - R6 Loopback 0 152.1.11.6 /32
   - R6 Loopback 1 152.1.11.129 /26

IGP Protocols

23. Place the serial connections between R3, R2, and R1 in OSPF Area 0.

24. Place VLAN 2 in OSPF area 1 and VLAN 3 in OSPF area 2. R1 should be the DR for Area 1 and R2 should be the DR in Area 2.

25. R1 should prefer the Frame Relay link over the Ethernet connection. All other connections should prefer the interface with the highest bandwidth.

26. Change the Hello interval on R3 to two minutes.

27. Configure EIGRP on the interface connecting R3 to R7. Do not send advertisements out any other interface. Enable EIGRP on all interfaces on R7, but do not send updates out the interface connecting to VLAN1.

28. R7 should be able to see all the networks behind R3. R3 only needs to see network 152.1.20.16/28 via R7.

29. Make sure R7 does not advertise any routes back to R3 that were learned from R3 and vice versa. Do not rely on split horizons!

30. Enable EIGRP using process 56 on R4 and R5. Don’t run EIGRP on the interfaces that are connected to other ASs.
31. Make sure that the loopback interfaces on R2, R1, and R3 are being propagated via OSPF. OSPF should not be run on these interfaces, and the routes should not show up as OSPF external routes.

**EBGP/IBGP**

32. Place R3, R2, and R1 in AS 100. R1 will peer with R2 and R3. R1 is the only router in AS100 that should have two IBGP neighbors.

33. Place R4 and R5 in AS 200. R4 should peer with R5 and R3.

34. Place R6 in AS 300 and R7 in AS 400

35. R7 should BGP peer with R6, R5, and R8

36. R5 should BGP peer with R7, R8, R6, and R3.

37. R6 should BGP peer with R7, R8, R5, and R2.

38. R7, R5, and R6 should all see the following networks via BGP from R8:

   - Network 148.1.0.0
   - Network 148.2.0.0
   - Network 148.3.0.0
   - Network 148.4.0.0

39. AS400 should advertise local network 152.1.20.0 via BGP and AS300 should advertise local network 152.1.11.128 via BGP.

40. AS 200 should advertise local network 152.1.13.0 /24 and 152.1.14.0 / 29.

41. AS100 should only accept local routes from AS300. All other routes should be accepted via AS200.

42. AS100 should accept a default route from AS300 in case of a link failure between A200 and AS100. The default should not be advertised outside of AS100.

43. Advertise network 152.1.9.0 via IBGP on R1 and R2. Do not use the redistribute command.

44. AS 100 should advertise all local networks via BGP. To prevent AS 100 from being used as a transit network, AS300 should only accept local networks from AS100.

45. AS100 prefers that network 152.1.9.0 be reachable by the outside world via the link between AS100 and AS300.

46. Under no circumstances can AS100 be used as a transit network for AS200 to reach AS300. To accomplish this, no additional configurations should be added to any router other than R2.
A Case Study in BGP

47. AS200 and AS300 should not pass network 152.1.9.0. No additional configuration should be used to achieve this outside of AS100.

48. R3 should use R5 to reach network 152.1.14.0 and R4 to reach network 152.1.13.0. No additional configuration should be used to achieve this outside of AS200.

49. R7 should originate and advertise network 148.0.0.0, 148.5.0.0, 148.6.0.0, and 148.7.0.0. Add four loopback interfaces to R7: loopback 2 IP address 148.0.0.1/16, loopback 3 IP address 148.5.0.1/16, loopback 4 IP address 148.6.0.1/16, and loopback 5 IP address 148.7.0.1/16.

50. R5 should advertise as few prefixes for 148.x.0.0 as needed.

51. R3 is running an IGP (EIGRP) on the private link between it and AS400. Make sure that the IGP route to network 152.1.20.16 is favored over the EBGP route.

52. The configuration on R3 is growing extremely large and approaching the limitation of the router's NVRAM. Reduce the size of the configuration on R3.

53. Configure R3 so that informational messages are logged to the internal buffer. Make sure the router can store up to 16K of information and the information is time stamped.

54. Configure R3 so that SNMP information can only be read by host 192.1.1.1.

55. Network 152.1.13.0 is suspected of being unstable, causing BGP UPDATE and WITHDRAWN messages to be repeatedly propagated on the network. Configure R3 so that if the network continues to FLAP, it will be withdrawn.

56. Enable R3 so that when its neighbors make a BGP change, the change will take effect without resetting the BGP TCP session. R3 should be able to initiate the change locally; no commands should have to be entered on the neighboring routers.

Answer Guide

1. Cable the routers according to Figure 2-6 and load the configuration provided on the CD to R8. The configuration provided is from a 4500 with an Ethernet interface and a four-port serial network module. If you plan on using a router other than a 4500 or one that has a different way of numbering its interfaces, such as a 3620, the interfaces provided in the configuration will need to be renamed.

2. Erase any existing configuration from the routers and set the hostname on each one. The name will be the corresponding router number; for example, router 1 will be R1.

   The first step is to erase the configuration for all the routers as well as the Catalyst switch. To erase the configuration on the router, perform the following command and then reload the router:

   ```
   R1#write erase or R1#erase startup-config
   ```
This command erases the startup configuration stored in NVRAM on all platforms except the Cisco 7000 series, the Cisco 7200 series, and the Cisco 7500 series. On those models, the router erases or deletes the configuration pointed to by CONFIG_FILE environment variable. If the CONFIG_FILE variable points to NVRAM, then the router erases NVRAM. If the CONFIG_FILE environment variable specifies a flash memory device and a configuration filename, then the command deletes the named file.

Erasing the configuration on the Catalyst 1900 is a bit different. From the main menu, select M for menus.

Enterprise Edition Software
Ethernet Address: 00-D0-C0-69-5A-40
PCA Number: 73-3121-01
PCA Serial Number: FAB03163GAT
Model Number: WS-C1924-A
System Serial Number: FAB0319T0KT
Power Supply S/N: PHI031004WD
PCB Serial Number: FAB03163GAT, 73-3121-01

1 user(s) now active on Management Console.

User Interface Menu

[M] Menus
[K] Command Line
[I] IP Configuration

Enter Selection: M

From the Menu option select S for system configuration.

Catalyst 1900 - Main Menu

[C] Console Settings
[S] System
[N] Network Management
[P] Port Configuration
[A] Port Addressing
[D] Port Statistics Detail
[M] Monitoring
[V] Virtual LAN
[R] Multicast Registration
[F] Firmware
[I] RS-232 Interface
[U] Usage Summaries
[H] Help
[K] Command Line

[X] Exit Management Console

Enter Selection: S
A Case Study in BGP

From the Menu select F to reset to factory defaults. This command will reset the switch with factory defaults. All system parameters will revert to their default factory settings. All static and dynamic addresses will be removed.

----------------------- Settings ---------------------------------------
[N] Name of system        cat
[C] Contact name
[L] Location
[S] Switching mode        FragmentFree
[U] Use of store-and-forward for multicast Disabled
[A] Action upon address violation Suspend
[G] Generate alert on address violation Enabled
[I] Address aging time    300 second(s)
[P] Network port          None
[H] Half duplex back pressure (10-mbps ports) Disabled
[E] Enhanced congestion control (10-mbps ports) Disabled

----------------------- Actions ----------------------------------------
[R] Reset system              [F] Reset to factory defaults
[V] Reset VTP to factory defaults [T] Reset to enable Bridge Groups

----------------------- Related Menus ----------------------------------
[B] Broadcast storm control  [X] Exit to Main Menu

Enter Selection: F

The next step is to set the hostname on each router. To do this, perform the following global configuration command:

router(config)#hostname R5

This is the same for the Catalyst. From the main menu, select K, Command line:

Enterprise Edition Software
Ethernet Address:   00-D0-C0-69-5A-40
PCA Number:         73-3121-01
PCA Serial Number:  FAB03163GAT
Model Number:       WS-C1924-A
System Serial Number: FAB0319T0KT
Power Supply S/N:   PHI031004WD
PCB Serial Number:  FAB03163GAT,73-3121-01

1 user(s) now active on Management Console.

User Interface Menu

[M] Menus
[K] Command Line
[I] IP Configuration

Enter Selection: K
Enter the following global configuration command:

```
cat(config)#hostname cat
```

3. **Set the enable password on each router to be cisco.**

When a router is first configured, no default enable password exists. The enable password is used to keep unauthorized users from changing the configuration. To set the enable password, use the following global configuration command:

```
R5(config)#enable password cisco
```

4. **Configure the terminal server (R7) so that all routers can be accessed by name through it.**

The port number of the terminal server attaches to the corresponding router; that is, port 1 of the terminal server goes to the console port of R1, and the cat switch goes to port 9. Use the Ethernet IP address as the IP address for reverse telneting.

The terminal server provides access to all your test routers via reverse telnet, which is the process of using telnet to make a connection out an asynchronous port.

All the routers in the case study as well as the Catalyst switch will have their console port connected directly to one of the 16 asynchronous interfaces on the Cisco 2511RJ terminal server using a standard Cisco console rolled cable. The routers will be accessed using a reverse telnet connection. To make a reverse telnet connection, you telnet to any active IP address on the box followed by 200x, where x is the port number that you want to access (Telnet 1.1.1.1 2001).

Use the following line commands to configure the terminal server:

```
R7(config)#line 1 16
R7(config-line)#transport input all
R7(config-line)#no exec
```

The command **transport input all** specifies the input transport protocol. By default on IOS 11.1 and later, the transport input is set to none, while prior to 11.1, the default was all. If the transport input is left to none, you will receive an error stating that the connection is refused by the remote host. The command no exec enables only outgoing connections for the line. This prevents the device that you are attached to from sending out unsolicited data. If the port receives unsolicited data, an EXEC process will start that makes the line unavailable.

The next step is to configure hostnames on R7 so you can attach to any router simply by typing in the router name. The Cisco IOS software maintains a table of host names and their corresponding addresses. Similar to a DNS server, you can statically map host names to an IP address. This is very useful and saves a lot of keystrokes when you have multiple devices connected to the terminal server.

The following global configuration commands are used to set the host names:
5. Place Ethernet interface 0 of R5, R6, R7, and R8 in VLAN 1. The routers should be attached in ascending order, starting with port 1 on the Catalyst switch.

This step requires no configuration on the Catalyst because, by default, all ports are in VLAN 1.

6. Configure the router so that if an incorrect command is entered, the router does not try to translate it to an IP address.

By default, the IOS will try to resolve a name to an IP address. If the user mistypes a command, such as typing in “wrete” instead of “write,” the IOS will attempt to resolve the name through a domain name server.

```
R7#wrete
Translating "wrete" . . . domain server (255.255.255.255)
% Unknown command or computer name, or unable to find computer address
```

To disable this function, use the following global configuration command:

```
R7(config)#no ip domain-lookup
```

7. The Media Access Control (MAC) addresses of the Ethernet interfaces attached to the VLAN should have the following format (routernumber.asnumber.0000). For example, R7 would be set to 0007. 0400.0000.

The Cisco IOS enables you to manually set the MAC address on an interface. This is particularly useful for troubleshooting and monitoring. To set the MAC address, use the following interface command:

```
R7(config-if)#mac-address 0007.0300.0000
```

8. Configure the Catalyst so that only the designated router can attach to the specific port.

Catalyst MAC address security enables the Catalyst switch to block input to an Ethernet port when the MAC address of a station attempting to access the port is different from the configured MAC address. When a port receives a frame, the Catalyst compares the source address of that frame to the secure source address learned or configured for the port. By default, when port security is enabled, the Catalyst learns the first device that attaches to
the port; this will be the MAC address that the Catalyst will permit. When a source address change occurs, the port is disabled, and the LED for that port turns orange. When the port is re-enabled, the port LED turns green. If the MAC address is not given, the command turns on the learning mode, so that the first MAC address shown on the port becomes the secure MAC address.

To enable port security on the Catalyst 1900, perform the following commands:

From the main menu, select **M** for menus:

Enterprise Edition Software
Ethernet Address: 00-D0-C0-69-5A-40
PCA Number: 73-3121-01
PCA Serial Number: FAB03163GAT
Model Number: WS-C1924-A
System Serial Number: FAB0319T0KT
Power Supply S/N: PHI031004WD
PCB Serial Number: FAB03163GAT, 73-3121-01

1 user(s) now active on Management Console.

User Interface Menu

[M] Menus
[K] Command Line
[I] IP Configuration

Enter Selection: **M**

Select **P** for port configuration:

[C] Console Settings
[S] System
[N] Network Management
[P] Port Configuration
[A] Port Addressing
[D] Port Statistics Detail
[M] Monitoring
[V] Virtual LAN
[R] Multicast Registration
[F] Firmware
[I] RS-232 Interface
[U] Usage Summaries
[H] Help
[K] Command Line

[X] Exit Management Console

Enter Selection: **P**

Enter the port number:

Identify Port: 1 to 24[1-24], [AUI], [A], [B]:
Select [1 - 24, AUI, A, B]: 3
A Case Study in BGP

Enter A for port addressing:

Catalyst 1900 - Port 1 Configuration

Built-in 10Base-T
802.1d STP State: Forwarding Forward Transitions: 2

------------- Settings -----------------------
[D] Description/name of port
[S] Status of port Enabled
[F] Full duplex Disabled
[I] Port priority (spanning tree) 128 (80 hex)
[C] Path cost (spanning tree) 100
[H] Port fast mode (spanning tree) Enabled

------------- Related Menus -----------------------
[A] Port addressing
[V] View port statistics
[N] Next port
[G] Goto port
[P] Previous port
[X] Exit to Main Menu

Enter Selection: A

Enter S for address security:

------------- Settings -----------------------
[T] Address table size Unrestricted
[S] Addressing security Disabled
[U] Flood unknown unicastst Enabled
[M] Flood unregistered multicasts Enabled

------------- Actions ------------------------
[A] Add a static address
[D] Define restricted static address
[L] List addresses
[E] Erase an address
[R] Remove all addresses
[C] Configure port
[V] View port statistics
[N] Next port
[G] Goto port
[P] Previous port
[X] Exit to Main Menu

Enter Selection: S

Addressing security serves to restrict the usage of the port to a specified, static set of addresses. Network management alerts as well as suspension or disablement of the port may result from address violations.

Enable address security:

Addressing security may be [E]nabled or [D]isabled:

Current setting ===> Disabled

New setting ===> Enabled

Now enter A to add a static address:
Enter Selection:  **A**

Enter the Mac address that you want permitted.

This command adds a static address for the port.

**Enter address:**

Enter address (6 hex octets: hh hh hh hh hh hh):  **00 07 03 00 00 00**

**Enter L to verify the configuration:**

Catalyst 1900 - Port 1 Addressing

Dynamic addresses: 0  Static addresses: 2

Enter Selection:  **L**
9. **Verify step 8 is working by swapping the cables between port 1 and port 2. The port should go into blocking.**

   In order to verify that the port security function is working, swap the cables between port 1 and port 2. If port security is configured correctly, the port should go into blocking mode, which is indicated by the orange LED above the port.

10. **Normalize the port.**

   In order to get the port out of blocking mode, it must be reset. To do this, log into the Catalyst and go into the Port Configuration menu. Below is the output, note that the status of the port is Suspended-violation.

   To reset the port, select S and then Enable. The LED on the port should return to green, indicating a normal state.

   ```
   Catalyst 1900 - Port 1 Configuration
   Built-in 10Base-T
   802.1d STP State:  Blocking  Forward Transitions:  2
   ----------------------- Settings ---------------------------------------
   [D] Description/name of port                [S] Status of port  Suspended-violation
   [F] Full duplex                                 Disabled
   [I] Port priority (spanning tree)               128 (80 hex)
   [C] Path cost (spanning tree)                   100
   [H] Port fast mode (spanning tree)              Enabled
   ----------------------- Related Menus ----------------------------------
   [A] Port addressing           [V] View port statistics
   [P] Previous port             [X] Exit to Main Menu
   Enter Selection:  s
   ```

11. **Assign IP addresses to Ethernet 0 of R5, R6, R7, and R8 from network 152.1.1.0. Do not allocate more addresses than needed, use the first available subnet, and assign the addresses in ascending order, starting with R5.**

   The next step is to assign IP addresses to the Ethernet interface of R5, R6, R7, and R8. The question specifies to use the first available subnet from 152.1.1.0 and to not allocate more address space than needed. Because you need four addresses, you must use a 29-bit mask. This gives you a range of addresses from 152.1.1.0 to 152.1.1.7. Although this provides eight addresses, the first and last ones are reserved:

   152.1.1.0← **Network Address reserved**
   
   152.1.1.1← **R5**
   
   152.1.1.2← **R6**
12. Place Ethernet 0 of R1 and R3 in VLAN 2. Then place Ethernet 1 of R1 and Ethernet 0 of R2 in VLAN 3.

In the previous step, you did not need to make any configuration changes to the Catalyst, because by default all ports are in VLAN 1. You must configure Catalyst ports 5 and 6 so that they are in VLAN 2 and also configure ports 7 and 8 in VLAN 3.

The first step is to add two new VLANs, in this case VLAN 2 and VLAN3. To do this select V from the menu.

Catalyst 1900 - Main Menu

[C] Console Settings
[S] System
[N] Network Management
[P] Port Configuration
[A] Port Addressing
[D] Port Statistics Detail
[M] Monitoring
[V] Virtual LAN
[R] Multicast Registration
[F] Firmware
[I] RS-232 Interface
[U] Usage Summaries
[H] Help
[K] Command Line

[X] Exit Management Console

Enter Selection: V

Enter A to add a VLAN:

Catalyst 1900 - Virtual LAN Configuration

-------------------------------------------------------- Information -----------------------------------------------
VTP version: 1
Configuration revision: 2
Maximum VLANs supported locally: 1005
Number of existing VLANs: 6
Configuration last modified by: 10.10.4.234 at 00-00-0000 00:00:00

------------------------------------------------------- Settings --------------------------------------------------
[N] Domain name
[V] VTP mode control Server
[F] VTP pruning mode Disabled
A Case Study in BGP

[VTP traps] Enabled

----------------------- Actions ----------------------------------------
[L] List VLANs [A] Add VLAN
[M] Modify VLAN [D] Delete VLAN
[E] VLAN Membership [S] VLAN Membership Servers
[T] Trunk Configuration [W] VTP password
[P] VTP Statistics [X] Exit to Main Menu

Enter Selection: A

Select the VLAN type to be added, in this case it is Ethernet:

This command selects the type of VLAN to be added.

The following VLAN types can be added:


Select a VLAN type [1-5]: Ethernet

Select N and then enter the VLAN number.

Catalyst 1900 - Add Ethernet VLAN

----------------------- Settings ---------------------------------------
[N] VLAN Number                  3
[V] VLAN Name                    VLAN0003
[I] 802.10 SAID                   100003
[M] MTU Size                     1500
[L] Translational Bridge 1       0
[J] Translational Bridge 2       0
[T] VLAN State                   Enabled

[S] Save and Exit                [X] Cancel and Exit

Enter Selection: N

This command selects the unique VLAN identifier of a VLAN.

Configuration change only takes effect when the VLAN SAVE command is executed.

Enter VLAN Number [2-1001]:

Current setting ===> 3

New setting ===> 2

Enter S to save the configuration.

Catalyst 1900 - Add Ethernet VLAN

----------------------- Settings ---------------------------------------
[N] VLAN Number                  2
[V] VLAN Name                    VLAN0003
The next step is to place the port in the newly created VLAN. From the Main Menu, select V for VLAN:

Catalyst 1900 - Main Menu
[C] Console Settings
[S] System
[N] Network Management
[P] Port Configuration
[A] Port Addressing
[D] Port Statistics Detail
[M] Monitoring
[V] Virtual LAN
[R] Multicast Registration
[F] Firmware
[I] RS-232 Interface
[U] Usage Summaries
[H] Help
[K] Command Line
[X] Exit Management Console

Enter Selection: V

From the Virtual LAN Configuration menu, select E for VLAN Membership:
A Case Study in BGP

The VLAN membership configuration menu appears, listing all the ports and the VLANs of which they are members. Note that all the ports are in VLAN 1, which is the default setting on the Catalyst.

To change the VLAN membership on a port, select V for VLAN assignment:

Catalyst 1900 - VLAN Membership Configuration

<table>
<thead>
<tr>
<th>Port</th>
<th>VLAN</th>
<th>Membership Type</th>
<th>Port</th>
<th>VLAN</th>
<th>Membership Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Static</td>
<td>13</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Static</td>
<td>14</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Static</td>
<td>15</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Static</td>
<td>16</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Static</td>
<td>17</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>Static</td>
<td>18</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>Static</td>
<td>19</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>Static</td>
<td>20</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>Static</td>
<td>21</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>Static</td>
<td>22</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>Static</td>
<td>23</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>Static</td>
<td>24</td>
<td>1</td>
<td>Static</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>Static</td>
<td></td>
<td>AUI</td>
<td>Static</td>
</tr>
</tbody>
</table>

Enter Selection:  V

The next screen prompts you to enter the ports that you want to place in a particular VLAN. The command enables you to enter the ports one by one or as a group using commas and dashes. In the following example, you enter port 5 and 6, separating them with a dash.

This command assigns or re-assigns ports to a VLAN. If the port is configured with dynamic VLAN membership, then assigning VLAN 0 causes the discovery of VLAN membership to start all over again.

Port numbers should be separated by commas or spaces. A port number range may also be specified. The word ALL indicates all ports. Example: 1, 7-11, AUI, 4, A

Enter port numbers:  5-6

You are then prompted to enter the VLAN in which to place the ports. This example uses VLAN 2. Repeat the same steps to place port 7 in VLAN 3.
13. Connect the rest of the routers serially as per Figure 2-6. Make sure that the correct ports are configured for DTE/DCE (the DCE clock is set to 64K on all routers).

This step is straightforward. The only thing that needs to be done is to set the clock rate on the DCE side of each serial link.

To set the clock rate, use the following interface command:

```
R7(config-if)#clock rate 64000
```

Use the `show controller` interface command to verify that the DCE cable is connected to the correct serial port. The following is a truncated output from the command, note that the cable attached is a V35 DCE cable and the clock rate is set to 64K.

```
R7#show controllers s 0
HD unit 0, idb = 0x970D0, driver structure at 0x9AE28
buffer size 1524  HD unit 0, V.35 DCE cable, clockrate 64000
cpb = 0xE1, eda = 0x4940, cda = 0x4800
RX ring with 16 entries at 0xE14800
```

14. R3 and R2 are connected via Frame Relay to R1. Use physical interfaces on R3 and R2, and a logical interface on R1. Assign addresses to each interface using subnet 152.1.10.0 and do not allocate more addresses than are needed. Use the first subnet available and assign the addresses in ascending order, starting with R1. Configure the interfaces for Frame Relay. Make sure that only DLCIs used by Routers are seen and that all routers can reach each other.

The following is the physical connectivity of R1, R2, and R3 to the Frame Relay switch (R8):

<table>
<thead>
<tr>
<th>Frame Switch (R8) interface S0</th>
<th>R1 Interface S0/0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame Switch (R8) interface S0</td>
<td>R2 Interface S0</td>
</tr>
<tr>
<td>Frame Switch (R8) interface S0</td>
<td>R3 Interface S0/1</td>
</tr>
</tbody>
</table>

This question can be a bit tricky unless you read the case study completely. Question 22 specifies that you should use address 152.1.10.3 /32 for the loopback interface on R3, 152.1.10.2 /32 for the loopback interface on R2, and address 152.1.10.1 /32 on R1.

You need three IP addresses, so you must use a 29-bit mask. This will give you eight total addresses, of which six are usable. The first available subnet is 152.1.10.8 because subnet 152.1.10.0 is used by the loopback interface.

The IP addressing is as follows:

152.1.10.8←Network address reserved
152.1.10.9←R1
152.1.10.10←R2
152.1.10.11←R3
Now that the IP addressing is laid out, you must configure the interfaces for Frame Relay. The question specifies that R3 and R2 use a physical interface and that R1 use a logical or subinterface. To create a logical subinterface on R1, you must first configure the physical interface with the Frame Relay encapsulation information:

```
R1(config)#interface serial 0/0
R1(config-if)#encapsulation frame-relay IETF
R1(config-if)#frame-relay lmi-type ansi
```

Then create the subinterface by specifying the physical interface followed by a dot and the logical interface. If the interface does not exist, it is created. Two types of Frame Relay subinterfaces exist: point-to-point and multipoint:

```
R1(config)#int s0/0.1 multipoint
R1(config-subif)#ip add 152.1.10.9 255.255.255.248
```

The question also states that the router should only see DLCIs that it is using:

```
R1(config-subif)#no frame-relay inverse-arp
```

Because you are disabling inverse arp, you must map the far-end layer-3 address to a layer-2 DLCI. To do this, perform the following:

```
R1(config-subif)#frame-relay map ip 152.1.10.11 200 broadcast
R1(config-subif)#frame-relay map ip 152.1.10.10 100 broadcast
```

Because you are not using logical interfaces on R2 and R3, all configurations are done under the physical interface:

```
R2(config)#interface s0
R2(config-if)#ip address 152.1.10.10 255.255.255.248
R2(config-if)#encapsulation frame-relay IETF
R2(config-if)#frame-relay lmi-type ansi
R2(config-if)#no frame-relay inverse-arp
R2(config-if)#frame-relay map ip 152.1.10.9 100 broadcast
R2(config-if)#frame-relay map ip 152.1.10.11 100 broadcast
```

```
R3(config)#interface serial 0/1
R3(config-if)#ip address 152.1.10.11 255.255.255.248
R3(config-if)#encapsulation frame-relay IETF
R3(config-if)#frame-relay lmi-type ansi
R3(config-if)#no frame-relay inverse-arp
R3(config-if)#frame-relay map ip 152.1.10.10 200 broadcast
R3(config-if)#frame-relay map ip 152.1.10.9 200 broadcast
```
15. **Assign IP addresses to Ethernet 0/0 of R3 and R1 using the next available subnet in 152.1.8.0. The subnet should contain 48 host addresses, R1 should use the first address in the subnet, and R3 should use the last one.**

This question involves simple subnetting. You need to pick a subnet that contains 48 usable IP addresses. A 26-bit mask provides 64 addresses, two of which can’t be used, leaving 62. This is a bit more than you need, but if you choose a 27-bit mask, you would be short. The following is the IP address assignment:

```
152.1.8.0 /26←Network Address
152.1.8.1 /26←R1
... ... ... ... ....
152.1.8.62 /26←R3
152.1.8.63 /26←Broadcast Address
```

The following commands define IP addresses on R3 and R1:

```
R3(config)#interface ethernet 0/0
R3(config-if)#ip address 152.1.8.62 255.255.255.192
R1(config)#interface ethernet 0/0
R1(config-if)#ip address 152.1.8.1 255.255.255.192
```

16. **Assign IP addresses to Ethernet 0/1 of R1 and Ethernet 0 of R2 using the first available subnet in 152.1.9.0. The subnet should contain 128 host addresses, R1 should use the first address in the subnet, and R2 should use the last one.**

This question is also a review of subnetting. You need to pick a subnet that contains 128 usable addresses. At first glance, you might think to use a 25-bit mask that provides 128 addresses, but because two of these addresses cannot be used, you are forced to use a 24-bit mask. The following is the IP address assignment:

```
152.1.9.0 /24←Network Address
152.1.9.1 /24←R1
152.1.9.254 /24←R2
152.1.9.255 /24←Network Broadcast
```

The following commands define IP addresses on R1 and R2:

```
R1(config)#interface ethernet 0/1
R1(config-if)#ip add 152.1.9.1 255.255.255.0
R2(config)#interface ethernet 0/
R2(config-if)#ip address 152.1.9.254 255.255.255.0
```

17. **Assign IP addresses to Serial 0 on R7 and Serial S0/0 on R3 using the first available subnet in 152.1.20.0. The address should be assigned in ascending order, starting with R3.**
A Case Study in BGP

The serial connection between R7 and R3 is point-to-point and requires two addresses. For this use a 30-bit mask. The following is the IP address assignment:

```
152.1.20.0 /30←Network Address
152.1.20.1 /30←R3
152.1.20.2 /30←R7
152.1.20.3 /30←Broadcast Address
```

The following commands define IP addresses on R1 and R2:

```
R3(config)#interface serial 0/0
R3(config-if)#ip address 152.1.20.1 255.255.255.252
R7(config)#interface serial 0
R7(config-if)#ip address 152.1.20.2 255.255.255.252
```

18. **Assign IP addresses to the serial interface on R3 that connects to R4 using the first available subnet in 152.1.12.0. The subnet should contain the minimal hosts needed and the IP addresses should be assigned in ascending order, starting with R3.**

The serial connection between R3 and R4 is a point-to-point connection, so only two addresses are needed. After carefully reading the rest of the case study, determine if you can use the first subnet 152.1.12.0. For this, use a 30-bit mask. The following is the IP address assignment:

```
152.1.12.0 /30←Network Address
152.1.12.1 /30←R3
152.1.12.2 /30←R4
152.1.12.3 /30←Broadcast Address
```

The following commands define IP addresses on R3 and R4:

```
R3(config)#interface serial 1/0
R3(config-if)#ip address 152.1.12.1 255.255.255.252
R4(config)#interface serial 0
R4(config-if)#ip address 152.1.12.2 255.255.255.252
```

19. **Assign IP addresses to the serial interfaces on R3 that connect to R5 using the next available subnet in 152.1.12.0. The subnet should contain the minimal number of hosts needed and the IP addresses should be assigned in ascending order, starting with R3.**

Like the previous questions, the link between R3 and R5 is a point-to-point link, so you only need two addresses. The first subnet is being used by loopback interfaces on R4 and R5, the next available subnet is 152.1.12.8. Use a 30-bit mask because you only need two host IP addresses. The following is the IP address assignment:

```
152.1.12.8 /30←Network Address
152.1.12.9 /30←R3
152.1.12.10/30←R5
152.1.12.11/30←Broadcast Address
```
The following commands define IP addresses on R3 and R5:

```
R3(config)#interface serial 1/1
R3(config-if)#ip address 152.1.12.9 255.255.255.252
R5(config)#interface serial 0
R5(config-if)#ip address 152.1.12.10 255.255.255.252
```

20. Assign an IP address to Serial 1 on R4 and R5 using the next available subnet in 152.1.12.0. The subnet should contain the minimal hosts needed and the IP addresses should be assigned in ascending order, starting with R4.

Once again, because the link between R4 and R5 is point-to-point, you only need two addresses so use a 30-bit mask. The next available subnet is 152.1.12.12. The following is the IP address assignment:

```
152.1.12.12/30 ← Network Address
152.1.12.13/30 ← R3
152.1.12.14/30 ← R5
152.1.12.15/30 ← Broadcast Address
```

The following commands define IP addresses on R3 and R5:

```
R4(config)#interface serial 1
R4(config-if)#ip address 152.1.12.13 255.255.255.252
R5(config)#interface serial 1
R5(config-if)#ip address 152.1.12.14 255.255.255.252
```

21. Assign an IP address to Serial 1 of R2 and Serial 0 of R6 using the first available subnet in 152.1.11.0. The subnet should contain the minimal number of hosts needed and the IP addresses should be assigned in ascending order, starting with R2.

Again, the link between R2 and R6 is point-to-point, so you only need two addresses for this use a 30-bit mask. The next available subnet is 152.1.11.0. The following is the IP address assignment:

```
152.1.11.0 /30 ← Network Address
152.1.11.1 /30 ← R3
152.1.11.2 /30 ← R5
152.1.12.3 /30 ← Broadcast Address
```

The following commands define IP addresses on R3 and R5:

```
R2(config)#interface serial 1
R2(config-if)#ip address 152.1.11.1 255.255.255.252
R6(config)#interface serial 0
R6(config-if)#ip address 152.1.11.2 255.255.255.252
```
A Case Study in BGP

22. Assign the following loopback addresses to the corresponding routers:

- R7 Loopback 0 152.1.20.7 /32
- R7 Loopback 1 152.1.20.17 /28
- R4 Loopback 0 152.1.12.4 /32
- R4 Loopback 0 152.1.13.1 /24
- R5 Loopback 0 152.1.12.5 /32
- R5 Loopback 1 152.1.14.1 /29
- R3 Loopback 0 152.1.10.3 /32
- R2 Loopback 0 152.1.10.2 /32
- R1 Loopback 0 152.1.10.1 /32
- R6 Loopback 0 152.1.11.6 /32
- R6 Loopback 1 152.1.11.129 /26

IGP Protocols

23. Place the serial connections between R3, R2, and R1 in OSPF Area 0.

Enabling OSPF on a router is done in two steps. First, an OSPF process is defined and then an interface is added to the process. The command to start an OSPF process is `Router OSPF [Process #]`. Multiple OSPF processes can be configured on one router.

The following command enables OSPF process 64 on the router and assigns the serial interfaces attached to the Frame Cloud on the routers to area 0:

```
R1(config)#router ospf 64
R1(config-router)#network 152.1.10.8 0.0.0.7 area 0
```

```
R2(config)#router ospf 64
R2(config-router)#network 152.1.10.8 0.0.0.7 area 0
```

```
R3(config)#router ospf 64
R3(config-router)#network 152.1.10.8 0.0.0.7 area 0
```

Use the command `show ip ospf interface` to verify that OSPF is configured on the interface. The following is the output from this command. Notice that on R2, interface S0 is configured for OSPF and is in area 0. The command also shows the network type, timer intervals, and adjacent neighbors.

```
R2#show ip ospf interface
Ethernet0 is up, line protocol is up
    OSPF not enabled on this interface
```
Loopback0 is up, line protocol is up
OSPF not enabled on this interface
Serial0 is up, line protocol is up
Internet Address 152.1.10.10/29 , Area 0
Process ID 64, Router ID 152.1.10.2, Network Type NON_BROADCAST, Cost: 64
Transmit Delay is 1 sec, State DOWN, Priority 1
No designated router on this network
No backup designated router on this network
Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
Serial1 is up, line protocol is up
OSPF not enabled on this interface
Serial0/1 is up, line protocol is up
Internet Address 152.1.10.11/29 , Area 0
Process ID 64, Router ID 152.1.10.3, Network Type NON_BROADCAST, Cost: 64
Transmit Delay is 1 sec, State WAITING, Priority 1
No designated router on this network
No backup designated router on this network
Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
Hello due in 00:00:06
Wait time before Designated router selection 00:01:36
Neighbor Count is 0, Adjacent neighbor count is 0

Because the case study requires that you use a logical interface at the hub (R1) and a physical interface on the spokes (R3 and R4), the interfaces are all non-broadcast. By default, a Frame Relay physical interface and a point-to-point subinterface are non-broadcast, which means a designated router (DR) and backup designated router (BDR) is elected for the network. However, due to the lack of broadcast capabilities, the DR and BDR must have a static list of all routers attached to the Frame Relay cloud. Because the Hub (R1) is the DR, you need to define static neighbors on R1.

The following commands define OSPF neighbors on the hub router:

R1(config)#router ospf 64
R1(config-router)#neighbor 152.1.10.10
R1(config-router)#neighbor 152.1.10.11

When configuring OSPF on NBMA networks such as Frame Relay or ATM, care must be taken on which router becomes the DR and BDR for the network. The DR and BDR require full logical connectivity with all routers on the network.

All multi-access networks with two or more attached routers elect a DR. The DR concept enables a reduction in the number of adjacencies that need to be formed on a network. In order for OSPF-enabled routers to exchange routing information, they must form an adjacency with one another. If a DR is not used, then each router on a multi-access network would need to form an adjacency with every other router (because link state databases are synchronized across adjacencies). This would result in N-1 adjacencies.

Instead, all routers on a multi-access network form adjacencies only with the DR and BDR. Each router sends the DR and BDR routing information, and the DR is responsible for flooding this information to all adjacent routers and originating a network link advertisement on behalf of the network. The BDR is used in case the DR fails.
The reduction in adjacencies reduces the volume of routing protocol traffic as well as the size of the topological database.

The DR is elected using the Hello protocol, which will be described in detail in the IGP case study. The election of the DR is determined by the router priority, which is carried in the Hello packet. The router with the highest priority is elected the DR; if a tie occurs, the router with the highest router ID is selected.

The router ID is the IP address of the loopback interface. If no loopback is configured, the router ID is the highest IP address on the router. The router priority can be configured on the router interface with the `ip ospf priority` command.

When a router first becomes active on a multi-access network, the router checks to see if a DR currently exists for the network. If a DR is present, the router accepts the DR regardless of what its priority is. Once a DR is elected, no other router can become the DR unless the DR fails. If no DR is present on the network, then the routers negotiate the DR based on router priority.

For this question, use the `ip ospf priority` command to assure that R1 becomes the DR for the network. To do this, set the OSPF priority on R1 to 255 and the OSPF priority on R2 and R3 to 0. An OSPF priority of zero means that the router is ineligible in the DR election process.

```
R1(config)#router ospf 64
R1(config-subif)#ip ospf priority 255

R2(config)#router ospf 64
R2(config-if)#ip ospf priority 0

R3(config)#router ospf 64
R3(config-if)#ip ospf priority 0
```

Use the `show ip ospf neighbor` command on R1 to verify that an adjacency with the two spokes has formed. The following code is the output from the command. Notice that R1 has two OSPF neighbors, 152.1.10.3 and 152.1.10.2. These neighbor IDs are the router IDs of each router. OSPF uses the highest loopback interfaces as its router IDs. and if no loopback interface is configured on the router, the highest IP address is used.

```
R1#show ip ospf neighbor
Neighbor ID   Pri  State          Dead Time     Address        Interface
152.1.10.3    0  FULL/DROTHER  00:01:40   152.1.10.11   Serial0/0 .1
152.1.10.2    0  FULL/DROTHER  00:01:43   152.1.10.10   Serial0/0 .1
```

The state of each neighbor is full, meaning that R1 has formed an adjacency with each one of its neighbors. Only the DR and BDR form adjacencies with all the routers on the network. R1 is the DR for the network, and each of the neighbors are DROTHER, which means that they are neither the BDR or DR for the network.
24. Place VLAN 2 in OSPF area 1 and VLAN 3 in OSPF area 2. R1 should be the DR for Area 1 and R2 should be the DR for Area 2.

This question is similar to question 23, the only difference being the network type. The broadcast network type is the default type on LANs, such as Token Ring, Ethernet, or FDDI. The only difference between the two network types is that no neighbors need to be defined in the broadcast model and the Hello, dead, and wait intervals are different.

If either the Hello interval or the dead interval do not match, the router does not form an adjacency with its neighbor.

The following commands enable OSPF process 64 on the router and assign the Ethernet interfaces in VLAN 2 to area 1 and VLAN 3 to area 2:

R1(config)# router ospf 64
R1(config-router)# network 152.1.8.0 0.0.0.63 area 1
R1(config-router)# network 152.1.9.0 0.0.0.255 area 2

R2(config)# router ospf 64
R2(config-router)# network 152.1.9.0 0.0.0.255 area 2
R3(config-router)# network 152.1.8.0 0.0.0.63 area 1

The default priority of an Ethernet interface is 1, so in order to assure that R1 and R2 are the DRs for their respective VLANs, you need to set their priorities higher than 1. To achieve this, you could use a number from 2 to 255. The following commands set the OSPF priority:

R1(config)# interface ethernet 0/0
R1(config-if)# ip ospf priority 2

R2(config)# interface ethernet 0
R2(config-if)# ip ospf priority 2

Because this question requires that the other routers become the BDRs for the network, you cannot set their priorities to 0. This could pose a problem, depending on what order the configuration was done in. For example, if you enable OSPF on VLAN1 and then went in and set the priority on R1, R2 would have already been elected the DR because it has the highest router ID. Remember that, by default, the priority of an Ethernet interface is 1, so when the routers elect a DR/BDR, they first look at the priority. If the priorities are equal, the routers then look at the router ID.

25. R1 should prefer the Frame Relay link over the Ethernet connection. All other connections should prefer the interface with the highest bandwidth.

Cisco’s implementation of OSPF calculates its route based on bandwidth. The higher the bandwidth, the lower the cost and the more preferable the route. The path cost is calculated using the following formula: cost = 100,000,000/bandwidth in bits per second. The cost is inversely proportional to the bandwidth of the link. The higher the bandwidth, the lower the cost.
A Case Study in BGP

This question requires that you set the cost of the path over the Frame Relay circuit so that it is lower or more preferable than the Ethernet link. By default, the OSPF cost of an Ethernet interface is 10 (100M/10M) and the cost of a serial interface is 64 (100M/1.54M). In order to get OSPF to prefer the route over the serial interface, you must manually set the cost lower than that of the Ethernet.

The following command sets the OSPF cost of the serial interface of R1 to 9, which, of course, is one less than the Ethernet interface. The problem is that by doing this R1 also prefers the serial link to reach R2. You must set the cost on the Ethernet interface attached to VLAN2 to a cost of 8:

```
R1(config)# interface serial 0/0 .1
R1(config-subif)# ip ospf cost 9
```

```
R1(config)# interface ethernet 1/0
R1(config-if)# ip ospf cost 8
```

26. **Change the Hello interval on R3 to two minutes.**

This question is tricky if you don’t understand what the Hello interval is used for and the implications of changing it on any router attached to a multi-access network.

The OSPF Hello interval is the length of time in seconds the router waits between sending Hello packets on a particular interface. Once the adjacency is formed, the Hello packet is used to maintain the neighbor relationship. If a Hello packet is not received within the specified dead interval, the neighbor is declared down. The default dead interval is four times the Hello interval.

The following command sets the dead interval on interface S0/1 of R3 to 120 seconds:

```
R3(config)# interface serial 0/1
R3(config-if)# ip ospf hello-interval 120
```

Let’s see what happens when you change the Hello interval on R3 to 120 seconds. Use the command `show ip ospf neighbor` to display the status of the neighbors. The following is the output from the command. Note that R3 has lost its adjacency over its serial link with R1, while the adjacency over the Ethernet still exists.

```
R3# show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
152.1.10.1        2   FULL/DR         00:00:34    152.1.8.1       Ethernet0/0
```

Monitor the OSPF events on R3 with the command `debug ip ospf events`. Notice that R3 is receiving an OSPF packet from R1 and the Hello intervals do not match. If either the Hello interval or the dead interval do not match, then the router does not form an adjacency with its neighbor.

```
OSPF: Mismatched hello parameters from 152.1.10.9
Dead R 120 C 480, Hello R 30 C 120  Mask R 255.255.255.248 C 255.255.255.248
```
In order for this to work, all routers that are attached to the multi-access network must set the Hello interval to 120 seconds.

R1(config)#interface serial 0/0 .1
R1(config-subif)#ip ospf hello-interval 120

R2(config)#interface serial 0
R2(config-if)#ip ospf hello-interval 120

27. Configure EIGRP on the interface connecting R3 to R7. Do not send advertisements out any other interface. Enable EIGRP on all interfaces on R7, but do not send updates out the interface connecting to VLAN1.

Enabling EIGRP on a router consists of two steps. First, an EIGRP routing process is enabled on the router, and then the network command is added, specifying which interfaces will receive and send EIGRP routing updates. It also specifies which networks will be advertised. The following command enables EIGRP on R3:

R3(config)#router eigrp 64
R3(config-router)#network 152.1.0.0

When enabling EIGRP on a router, you specify which classful network the protocol will be run on. Because every interface on the router is part of the classful network 152.1.0.0, EIGRP sends updates out every interface.

To verify which interfaces EIGRP is running on, use the show ip eigrp interface command. The following is the output from the command. Note that EIGRP is running on all interfaces.

R3#show ip eigrp interfaces
IP-EIGRP interfaces for process 64

<table>
<thead>
<tr>
<th>Interface</th>
<th>Peers</th>
<th>Xmit Queue</th>
<th>Mean SRTT</th>
<th>Pacing Time</th>
<th>Multicast Flow Timer</th>
<th>Pending Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Se0/0</td>
<td>1</td>
<td>0/0</td>
<td>269</td>
<td>0/15</td>
<td>1343</td>
<td>0</td>
</tr>
<tr>
<td>Et0/0</td>
<td>0</td>
<td>0/0</td>
<td>0</td>
<td>0/10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Se0/1</td>
<td>0</td>
<td>0/0</td>
<td>0</td>
<td>0/10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Se1/0</td>
<td>0</td>
<td>0/0</td>
<td>0</td>
<td>0/10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Se1/1</td>
<td>0</td>
<td>0/0</td>
<td>0</td>
<td>0/10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lo0</td>
<td>0</td>
<td>0/0</td>
<td>0</td>
<td>0/10</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

This question requires that EIGRP updates only be sent out the serial interface connecting R3 to R7. The passive-interface command enables EIGRP-enabled routers to listen to, but not send, routing updates out a particular interface. This command is typically used when the network router configuration command configures more interfaces than is desirable. To disable the sending of EIGRP updates out the other interfaces, perform the following commands:
A Case Study in BGP

Once again, use the command `show ip eigrp interfaces` to verify which interfaces which EIGRP is sending updates out. The following is the output from the command. Note that EIGRP is only sending updates out of S0/0, which is the interface connected to R7.

```
R3#show ip eigrp interfaces
IP-EIGRP interfaces for process 64

Interface      Peers  Un/Reliable  Mean  SRTT  Un/Reliable  Flow Timer  Routes
Se0/0           1       0/0     269   0/15   1343            0          
```

The same needs to be done on R7, but this time only the interface attached to VLAN1 is passive. The following are the commands:

```
R7(config-router)#network 152.1.0.0
R7(config-router)#passive-interface ethernet 0
```

28. **R7 should be able to see all the networks behind R3. R3 only needs to see network 152.1.20.16/28 via R7.**

R3 can see all the networks attached to R7, but R7 cannot see all the networks behind R3. In order to get the OSPF routes into EIGRP and the EIGRP routes into OSPF, you must perform mutual redistribution, which is when each routing protocol is redistributed into the other.

When redistributing from one protocol to another, you must take the metrics used into account. As discussed earlier, OSPF uses a metric based on the bandwidth of a link. EIGRP uses a metric based up to five variables.

The EIGRP metric is a 32-bit number, which is calculated using bandwidth, delay, reliability, loading, and the Maximum Transmission Unit (MTU). Calculating the metric for a route is a two-step process using the five different characteristics of the link and the K values. The K values are configurable, but this is not recommended. The default K values are K1=1, K2=0, K3=1, K4=0, and K5=0.

Metric = K1*Bandwidth + (K2 * Bandwidth)/(256-load) + K3*Delay

If K5 is not equal to zero, take the metric from step 1 and multiple it by [K5/(reliability + K4)]. If K5 is 0, ignore step 2.

Metric = Metric * [K5/(reliability + K4)]
As shown here, Cisco sets K2, K4, and K5 to 0. This leaves only two variables to compute the EIGRP metric (bandwidth and delay). Because three of the K values are 0, the formula is reduced to

\[ \text{Metric} = \text{Bandwidth} + \text{Delay} \]

The bandwidth is derived by finding the smallest of all bandwidths in the path to the destination and dividing 10,000,000 by that number.

The delay is found by adding all the delays along the paths and dividing that number by 10. The sum of the two numbers is then multiplied by 256. This equation can be written as

\[ \text{Metric} = \left( \frac{10,000,000}{\text{min bandwidth}} \right) + \left( \frac{\text{SUM(interface delay)}}{10} \right) \times 256 \]

In order to redistribute EIGRP into OSPF, you must tell the router what it should use for a metric. The following command is used to redistribute OSPF into EIGRP:

```
R3(config-router)# redistribute ospf 64 metric 1500 100 255 1 1500
```

Show the routing table on R7 to verify that you are receiving all the routes that are in AS100:

```
R7# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
U - per-user static route, o - ODR
Gateway of last resort is not set
152.1.0.0/16 is variably subnetted, 10 subnets, 6 masks
  C 152.1.20.0/32 is directly connected, Loopback0
  C 152.1.20.0/30 is directly connected, Serial0
  D EX 152.1.9.0/24 [170/2244096] via 152.1.20.1, 00:01:37, Serial0
  D 152.1.10.3/32 [90/2297856] via 152.1.20.1, 00:25:57, Serial0
  D 152.1.8.0/26 [90/2195456] via 152.1.20.1, 00:25:57, Serial0
  D 152.1.12.0/30 [90/2681856] via 152.1.20.1, 00:25:57, Serial0
  C 152.1.1.0/29 is directly connected, Ethernet0
  D 152.1.10.8/29 [90/2681856] via 152.1.20.1, 00:25:57, Serial0
  D 152.1.12.8/30 [90/2681856] via 152.1.20.1, 00:25:57, Serial0
  C 152.1.20.16/28 is directly connected, Loopback1

Now you need to redistribute the EIGRP-learned routes into OSPF. The following command is used to redistribute EIGRP into OSPF:

```
R3(config)# redistribute eigrp 64 subnets metric 64
R3(config-router)# redistribution ospf 64
```
A Case Study in BGP

29. **Make sure R7 does not advertise any routes back to R3 that were learned from R3 and vice versa. Do not rely on split horizons!**

Care must be taken when using mutual redistribution in order to prevent routing loops. Metrics and split horizons will help to prevent routing loops, but it is a good idea to configure distribution lists so the router cannot advertise invalid routes. You need to configure a distribute list out on R3 and R7.

On R3, you must configure a distribute list to deny networks 152.1.20.7/32, 152.1.1.0/29, and 152.1.20.16/28. This is a two-step process. First, an access list must be set up to identify and permit or deny a network based on the network address. Then the access list must be applied to routing updates through the use of a distribute list.

- **Define an access list on R3, denying networks 152.1.20.7/32, 152.1.1.0/29, and 152.1.20.16/28:**

  R3(config)#access-list 1 deny 152.1.20.7 0.0.0.0
  R3(config)#access-list 1 deny 152.1.1.0 0.0.0.7
  R3(config)#access-list 1 deny 152.1.20.16 0.0.0.15
  R3(config)#access-list 1 permit any

- **Apply the access list to routing updates through the use of a distribute list:**

  R3(config)#router eigrp 64
  R3(config-router)#distribute-list 1 out eigrp 64

Because you have less routes on R7 to advertise, configure an access list to permit that subset of routes.

- **Define an access list on R7, permitting networks 152.1.20.7/32, 152.1.1.0/29, and 152.1.20.16/28:**

  R7(config)#access-list 1 permit 152.1.20.7 0.0.0.0
  R7(config)#access-list 1 permit 152.1.1.0 0.0.0.7
  R7(config)#access-list 1 permit 152.1.20.16 0.0.0.15
  R7(config)#access-list 1 deny any

- **Apply the access list to routing updates through the use of a distribute list:**

  R7(config)#router eigrp 64
  R7(config-router)#distribute-list 1 out eigrp 64

30. **Enable EIGRP using process 56 on R4 and R5. Don’t run EIGRP on the interfaces that are connected to other ASs.**

Enabling EIGRP on a router is performed in two steps. First, an EIGRP-routing process is enabled on the router, and then the network command specifies which interfaces will receive and send EIGRP-routing updates. It also specifies which networks will be advertised. The following command enables EIGRP on R4 and R5:
When enabling EIGRP on a router, you must specify which classful network the protocol will be run on. Because every interface on the router is part of the classful network 152.1.0.0, EIGRP sends updates out every interface.

The question states that EIGRP should not be run on any interface connecting to another AS. To prevent this, use the passive interface command on S0 of R4 and R5 and the Ethernet interface on R5:

```
R4(config)#router eigrp 56
R4(config-router)#passive-interface serial 0

R5(config)#router eigrp 56
R5(config-router)#passive-interface serial 0
R5(config-router)#passive-interface ethernet 0
```

To verify which interfaces EIGRP is running on, use the `show ip eigrp interfaces` command. The following is the output from the command. Note that EIGRP is not running on the Serial interface 0 or the Ethernet interface of R5:

```
R5#show ip eigrp interfaces
IP-EIGRP interfaces for process 56
    Xmit Queue   Mean   Pacing Time   Multicast    Pending
    Interface    Peers  Un/Reliable  SRTT   Un/Reliable   Flow Timer   Routes
Lo0            0       0/0          0       0/10            0           0
Lo1            0       0/0          0       0/10            0           0
Se1            0       0/0          0       0/15            0           0
```

31. Make sure that the loopback interfaces on R2, R1, and R3 are being propagated via OSPF. OSPF should not be run on these interfaces, and the routes should not show up as OSPF external routes.

In order to get any network advertised in OSPF, it must be part of the OSPF process or be redistributed into the process. The instructions state that the routes should not show up as OSPF external. This means that you cannot redistribute connected subnets into OSPF. You must add the network to the OSPF process, but the problem with this is that once the network is added to the OSPF process, OSPF will be enabled on that interface. To get around this, make the interface passive.

The passive interface command disables the sending and receiving of OSPF router information. The specified interface address appears as a stub network in the OSPF domain:

```
R3(config)#router ospf 64
R3(config-router)#network 152.1.10.3 0.0.0.0 area 0
```
A Case Study in BGP

R3(config-router)#passive-interface loopback 0
R1(config)#router ospf 64
R1(config-router)#network 152.1.10.1 0.0.0.0 area 0
R1(config-router)#passive-interface loopback 0
R2(config)#router ospf 64
R2(config-router)#network 152.1.10.2 0.0.0.0 area 0
R2(config-router)#passive-interface loopback 0

EBGP/IBGP

32. Place R3, R2, and R1 in AS 100. R1 will IBGP peer with R2 and R3. R1 is the only router in AS100 that should have two IBGP neighbors.

Enabling BGP is a two-step process. First, the BGP routing process is enabled on the router; then neighbors are defined. If the neighbors are in the same AS, it is IBGP. If the neighbors are in a different AS, it is EBGP.

R1(config)#router bgp 100
R2(config)#router bgp 100
R3(config)#router bgp 100

To prevent routing loops within an AS, BGP does not advertise to internal BGP peers’ routes that it has learned about via other internal BGP peers. In Figure 2-7, R3 advertises all the routes it has learned via EBGP to R1, and these routes are not advertised to R2. This is because R1 does not pass IBGP routes between R3 and R2. In order for R2 to learn about these routes, an IBGP connection between R1 and R2 is needed.

Figure 2-7
IBGP full-mesh requirement
The full mesh requirement of IBGP creates the need for neighbor statements to be defined for each IBGP router. In an AS of 100 routers, this would require 100 neighbor statements to be defined. As you can see, this does not scale well.

To get around this, the concept of a route reflector has been defined. A route reflector acts as a concentration router or focal point for all IBGP sessions. Routers that peer with the route reflector are called route reflector clients. The clients peer with the route reflector and exchange routing information. The route reflector then exchanges or “reflects” this information to all clients, thereby eliminating the need for a fully meshed environment.

The instructions specify that R3 should not neighbor with R2, so you need to make R1 a route reflector for AS 100. R3 and R2 then only need to neighbor with R1. To do this, first define the neighbors:

```
R1(config)#router bgp 100
R1(config-router)#neighbor 152.1.10.10 remote-as 100
R1(config-router)#neighbor 152.1.10.11 remote-as 100
R1(config-router)#no synchronization
```

```
R2(config)#router bgp 100
R2(config-router)#neighbor 152.1.10.9 remote-as 100
R2(config-router)#neighbor 152.1.11.2 remote-as 300
R2(config-router)#no synchronization
```

```
R3(config)#router bgp 100
R3(config-router)#neighbor 152.1.10.9 remote-as 100
R3(config-router)#neighbor 152.1.12.2 remote-as 200
R3(config-router)#neighbor 152.1.12.10 remote-as 200
R3(config-router)#no synchronization
```

**NOTE:** In order for a router to advertise a network to another BGP speaker, the network to be advertised must be present in the IP routing table. Because you don’t want to redistribute BGP into OSPF in AS100, disable synchronization. If this is not done, the routers will not load the BGP routes into the IP routing table and R1 will not reflect the routes to R2.

Then specify R1 as a BGP route reflector and configure the specified neighbor as the route-reflector-client:

```
R1(config)#router bgp 100
R1(config-router)#neighbor 152.1.10.11 route-reflector-client
R1(config-router)#neighbor 152.1.10.10 route-reflector-client
```

To verify that BGP is working properly, display the BGP neighbors on R1 with the command `show ip bgp neighbors`. The following is the output from this command, which has been truncated. Notice that R1 has two IBGP neighbors: R2 (152.1.10.10) and R3
To identify itself to its neighbors, the BGP process uses a RouterID, which is an IP address. This address is either the loopback address or the highest IP address of an active interface on the router. The router ID is calculated at boot time and will remain constant until the BGP process is removed or the router is reloaded.

One of the most important things shown by the command `show ip bgp neighbors` is the line `BGP state =`. This shows the status of the BGP connection, which has six possible states.

Before BGP speakers can exchange network layer reachability information (networks being advertised), a BGP session must be established. Figure 2-8 illustrates the states that BGP neighbor negotiation goes through before a connection becomes fully established.
Initially, BGP is in an idle state until an operator initiates a start event, which is usually caused by establishing or restarting a BGP session.

In this state, BGP is waiting for the transport protocol connection to be completed. If the transport protocol connection succeeds, an Open message is sent to the peer router and the BGP state changes to OpenSent. If the connection fails, the local system changes to an active state and continues to listen for connections.

In this state, BGP is trying to acquire a peer by initiating a transport protocol connection. If the connection is successful, an OPEN message is sent to the peer router. If the connection retry timer expires, the BGP state changes to connect and continues to listen for connections that may be initiated by the remote BGP peer.
A Case Study in BGP

OpenSent State  In this state, BGP is waiting for an OPEN message from its peer. When an OPEN message is received, all fields are checked for correctness. If an error is detected, the local system sends a NOTIFICATION message and changes its state to Idle. If no errors occur, BGP starts sending keepalive messages to its peer.

OpenConfirm  In this state, BGP waits for a keepalive or a notification message. If the local system receives a keepalive message, it changes its state to established. If the Hold timer expires before a keepalive message is received, the local system sends a notification message and changes its state to Idle.

Established  This is the final stage of the neighbor negotiation. In the established state, BGP peers can exchange update, notifications, and keepalive messages.

33. Place R4 and R5 in AS 200. R4 should peer with R5 and R3.

This question requires that you place R4 and R5 in AS 100. To do this, define the BGP process with the following command:

\[
\begin{align*}
\text{R4(config)} & \# \text{router bgp 200} \\
\text{R5(config)} & \# \text{router bgp 200}
\end{align*}
\]

The question also requires that R4 peer with R5 and R3. To do this, define neighbors under the BGP process:

\[
\begin{align*}
\text{R4(config)} & \# \text{router bgp 200} \\
\text{R4(config-router)} & \# \text{neighbor 152.1.12.1 remote-as 100} \\
\text{R4(config-router)} & \# \text{neighbor 152.1.12.14 remote-as 200}
\end{align*}
\]

34. Place R6 in AS 300 and R7 in AS 400.

This question requires that you place R6 in AS 300 and R7 in AS 400. This is the same process that you used in the last question. The only difference is that the AS number has changed.

\[
\begin{align*}
\text{R6(config)} & \# \text{router bgp 300} \\
\text{R7(config)} & \# \text{router bgp 400}
\end{align*}
\]

35. R7 should BGP peer with R6, R5, and R8.

This question requires that R7 peer with R6, R5, and R8. To do this, simply define neighbor statements to each router with the appropriate remote AS:

\[
\begin{align*}
\text{R7(config)} & \# \text{router bgp 400} \\
\text{R7(config-router)} & \# \text{neighbor 152.1.1.1 remote-as 200} \\
\text{R7(config-router)} & \# \text{neighbor 152.1.1.2 remote-as 300} \\
\text{R7(config-router)} & \# \text{neighbor 152.1.1.4 remote-as 500}
\end{align*}
\]

36. R5 should BGP peer with R7, R8, R6, and R3.
This question requires that R5 peer with R7, R8, R6, and R3. To do this, simply define neighbor statements to each router with the appropriate remote AS:

```
R5(config)#router bgp 200
R5(config-router)#neighbor 152.1.1.2 remote-as 300
R5(config-router)# neighbor 152.1.1.3 remote-as 400
R5(config-router)# neighbor 152.1.1.4 remote-as 500
R5(config-router)# neighbor 152.1.12.9 remote-as 100
R5(config-router)# neighbor 152.1.12.13 remote-as 200
```

37. R6 should peer with R7, R8, R5, and R2.

This question requires that R6 peer with R7, R8, R5, and R2. To do this, simply define neighbor statements to each router with the appropriate remote AS:

```
R6(config)#router bgp 300
R6(config-router)# neighbor 152.1.1.1 remote-as 200
R6(config-router)# neighbor 152.1.1.3 remote-as 400
R6(config-router)# neighbor 152.1.1.4 remote-as 500
R6(config-router)# neighbor 152.1.11.1 remote-as 100
```

38. R7, R5, and R6 should all see the following networks via BGP from R8.

```
network 148.1.0.0
network 148.2.0.0
network 148.3.0.0
network 148.4.0.0
```

Use the `show ip bgp` command to verify that R5, R7, and R6 all see the networks that are being advertised by R8. The following is the output from the command on R5. Notice that the routes are being learned from AS 400 (R7), AS 300 (R6), and AS 500 (R8).

```
R5#show ip bgp
BGP table version is 6, local router ID is 152.1.14.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
*> 148.1.0.0      152.1.1.4            0     0    500 i
  152.1.1.4          0 300 500 i
  152.1.1.4          0 400 500 i
*> 148.2.0.0      152.1.1.4            0     0    500 i
  152.1.1.4          0 400 500 i
  152.1.1.4          0 300 500 i
*> 148.3.0.0      152.1.1.4            0     0    500 i
  152.1.1.4          0 300 500 i
  152.1.1.4          0 400 500 i
*> 148.4.0.0      152.1.1.4            0     0    500 i
  152.1.1.4          0 300 500 i
  152.1.1.4          0 400 500 i
```
A Case Study in BGP

39. **AS400 should advertise local network 152.1.20.0 via BGP, and AS300 should advertise local network 152.1.11.128 via BGP.**

Now from R7, you will advertise network 152.1.20.16 /30 via BGP and from R6 network 152.1.11.128 via BGP. In order for a router to advertise a network to another BGP speaker, two conditions must be met:

- **a.** The BGP process must be aware of the route, either through the use of the network command or by redistribution.

- **b.** The network to be advertised must be present in the IP routing table.

For the purposes of this case study, you will be using the network command under the BGP process. This takes care of the first rule, making the BGP process aware of the route. The network command gives you better control of what is being redistributed from the IGP into BGP, allowing the user to individually list the prefixes that need to be advertised via BGP. The maximum number of network statements that can be configured on a Cisco router is 200. If you have more than 200 networks to advertise, dynamic redistribution must be used.

The second rule is met because networks 152.1.20.0 and 152.1.11.128 are directly connected networks. Therefore, they are in the IP routing table.

To advertise network 152.1.20.16 /30 via BGP, add the following command under the BGP process. The mask must be added because the network does not fall on a major network boundary (255.0.0.0, 255.255.0.0, or 255.255.255.0). For example, the statement 152.1.0.0 is sufficient to send the prefix 152.1.0.0 /16:

```plaintext
R7(config-router)# network 152.1.20.16 255.255.255.240
R6(config-router)# network 152.1.11.128 mask 255.255.255.192
```

To verify that the network is being advertised, use the command `show ip bgp`. The following is the output from the command on R6. Notice that network 152.1.11.128 /26 is now in the BGP table.

```plaintext
R6# show ip bgp
BGP table version is 7, local router ID is 152.1.11.129
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
*> 148.1.0.0        152.1.1.4           0             0 500 i
  * 152.1.1.4                                              0 200 500 i
  * 152.1.1.4                                              0 400 500 i
*> 148.2.0.0        152.1.1.4           0             0 500 i
  * 152.1.1.4                                              0 400 500 i
  * 152.1.1.4                                              0 200 500 i
*> 148.3.0.0        152.1.1.4           0             0 500 i
```
40. AS 200 should advertise local network 152.1.13.0 /24 and 152.1.14.0 /29.

Now from R4, you will advertise network 152.1.13.0 /24 and from R5 network 152.1.14.0 /29 via BGP:

R4(config-router)#network 152.1.13.0 mask 255.255.255.0
R5(config-router)#network 152.1.14.0 mask 255.255.255.248

41. AS100 should only accept local routes from AS300. All other routes should be accepted via AS200.

AS100 should only accept local routes from AS300. In order to achieve this, you must apply an inbound route filter on R2, denying any route that originated outside of AS300.

In order to filter the routes, you must first identify them. To do this, use a regular expression to identify routes based on AS path information. A regular expression is a pattern to match against an input string. When a regular expression is created it specifies the pattern that a string must match. The following is a list of keyboard characters that have special meaning when used in regular expressions.

<table>
<thead>
<tr>
<th>Character</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period</td>
<td>.</td>
<td>Match any character, including white space.</td>
</tr>
<tr>
<td>Asterisk</td>
<td>*</td>
<td>Match zero or more sequences of the pattern.</td>
</tr>
<tr>
<td>Plus sign</td>
<td>+</td>
<td>Match one or more sequences of the pattern.</td>
</tr>
<tr>
<td>Question Mark</td>
<td>?</td>
<td>Match zero or one occurrence of the pattern.</td>
</tr>
<tr>
<td>Caret</td>
<td>^</td>
<td>Begins with.</td>
</tr>
<tr>
<td>Dollar Sign</td>
<td>$</td>
<td>Ends with.</td>
</tr>
<tr>
<td>Underscore</td>
<td>_</td>
<td>Match the following.</td>
</tr>
<tr>
<td>Brackets</td>
<td>[]</td>
<td>Match a single value in range.</td>
</tr>
<tr>
<td>Hyphen</td>
<td>-</td>
<td>Separates the end points of a range.</td>
</tr>
</tbody>
</table>
A Case Study in BGP

R2 prevents any network that does not originate from AS 300 from being accepted via BGP from R6. Filtering routes based on AS path information can be very useful when all routes from a particular AS need to be filtered. If filtering based on an AS path is not used, the administrator would have to list each route one by one or potentially filter on a prefix. AS path filtering provides an efficient alternative to this.

In order to filter routes based on AS path information, you must identify the AS path based on the defined regular expression and apply this to a BGP neighbor through a filter list:

**a.** Define the regular expression to deny any route that does not originate from AS300:

```
R2# configure terminal
R2(config)# ip as-path access-list 1 permit ^300$ ← Permit any route that originates from AS 300
```

Use the command `show ip bgp regexp` to see which routes the regular expression matches. The following is the output from the command. Note that network 152.1.11.128/26 is the only route that matches the regular expression (`^300$`). This command is very useful in verifying that the regular expression covers the routes that you intend it to.

```
R2# show ip bgp regexp ^300$
BGP table version is 9, local router ID is 152.1.10.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric LocPrf Weight Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>*&gt; 152.1.11.128/26</td>
<td>152.1.11.2</td>
<td>0</td>
</tr>
</tbody>
</table>
```

**b.** Define a route map named `Accept_Local` that accepts partial routes from AS300. The route map will try to match any path defined by the as-path access-list 1:

```
R2(config)#route-map Accept_Local permit 10
R2(config-route-map)#match as-path 1
```

**c.** Apply the route map to inbound traffic from neighbor (R6) 152.1.11.2:

```
R2(config-router)#neighbor 152.1.11.2 route-map Accept_Local in
```

In order for the changes to take effect, the BGP neighbor must be reset. To do this, use the command `clear ip bgp *`. This causes the TCP session between neighbors to be reset, the neighbor negotiations to be restarted, and the cache to be invalidated.

Verify that this is working properly by displaying the BGP table on R2. The following is the output from the command. Note that the only route that R2 is receiving from AS 300 is 152.1.11.128:
R2#show ip bgp
BGP table version is 25, local router ID is 152.1.10.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
*> 148.1.0.0        152.1.12.10                          0 200 500 1
*> 148.2.0.0        152.1.12.10                          0 200 500 1
*> 148.3.0.0        152.1.12.10                          0 200 500 1
*> 148.4.0.0        152.1.12.10                          0 200 500 1
* 152.1.11.128/26  152.1.12.10                          0 200 300 1
*> 152.1.11.2        152.1.12.10                          0 200 400 1
*> 152.1.13.0/24    152.1.12.2                           0   0 200 1
*> 152.1.14.0/29    152.1.12.2                           0   0 200 1
*> 152.1.20.16/28   152.1.12.10                          0 200 400 1

Verify that R3 is seeing network 152.1.11.128/26 via R2 with the show ip bgp command. The following code is the output from the command. Note that R3 is not receiving network 152.1.11.128/26 from R2; the route is being learned from R5.

R3#show ip bgp
BGP table version is 13, local router ID is 152.1.10.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
*> 148.1.0.0        152.1.12.10                          0 200 500 1
*> 148.2.0.0        152.1.12.10                          0 200 500 1
*> 148.3.0.0        152.1.12.10                          0 200 500 1
*> 148.4.0.0        152.1.12.10                          0 200 500 1
* 152.1.11.128/26  152.1.12.10                          0 200 300 1
*> 152.1.13.0/24    152.1.12.2                           0   0 200 1
*> 152.1.14.0/29    152.1.12.2                           0   0 200 1
*> 152.1.20.16/28   152.1.12.10                          0 200 400 1

The reason that the route to network 152.1.11.128/26 is not in the BGP table is because R3 does not know how to reach network 152.1.11.2 (that is, the next hop address). When a route is learned via BGP, the next hop address that is advertised is the IP address of the neighbor that announced the route, which in this case is R6 (152.1.11.2). When routes are injected into the AS via EBGP, the next hop learned from EBGP is carried unaltered into IBGP.

To fix this problem, you either have to advertise network 152.1.1.0 in our IGP (which is OSPF) or use the next hop self command. This command forces the router to advertise itself, rather than the external peer, as the next hop to reach the route. The following command on R2 forces the router to advertise all routes to neighbor 152.1.10.9 (R1) using its own IP address:

R2(config-router)#neighbor 152.1.10.9 next-hop-self

In order for the changes to take effect, the BGP neighbor must be reset. To do this, use the command clear ip bgp * . This causes the TCP session between neighbors to be
A Case Study in BGP

reset, the neighbor negotiations to be restarted from scratch, and the cache to be invalidated.

Verify that this is working properly by displaying the BGP table on R3. The following is the output from the command. Note that R3 now sees the network 152.1.11.128/26 via both R5 and R2:

![BGP Table Output](https://example.com/bgp_table.png)

42. AS100 should accept a default route from AS300 in case of a link failure between A200 and AS100. The default should not be advertised outside of AS100.

Cisco provides a way to target a default route to a specific peer. This prevents R6 from advertising the default to all peers. The following command on R6 only advertises a default route to neighbor (R2) 152.1.11.1:

```
R6(config) router bgp 300
R6(config-router) neighbor 152.1.11.1 default-originate
```

In order for the changes to take effect, the BGP neighbor must be reset. To do this, use the command `clear ip bgp *`. This causes the TCP session between neighbors to be reset, the neighbor negotiations to be restarted from scratch, and the cache to be invalidated.

You must also prevent R3 from advertising the route to AS200, so apply a filter list on R3. In order to filter routes based on a network address, you must identify network addresses through the use of an access list and apply that list to a BGP neighbor using a distribute list:

a. Define the access list on R3 to deny network 0.0.0.0:

```
R3(config)#access-list 2 deny 0.0.0.0
R3(config)#access-list 2 permit any
```

b. Apply the distribution list to both BGP neighbors:
In order for the changes to take effect, the BGP neighbor's must be reset. To do this, use the command **clear ip bgp**`. This causes the TCP session between neighbors to be reset, the neighbor negotiations to be restarted from scratch, and the cache to be invalidated.

RouterA#**clear ip bgp**

Display the BGP table on R5 and R4 with the command **show ip bgp**. The following is the output from the command on R5. Notice that the route to network 0.0.0.0 is no longer in the BGP table.

```
R5#**show ip bgp**
BGP table version is 15, local router ID is 152.1.14.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>Weight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 148.1.0.0</td>
<td>152.1.1.4</td>
<td>0</td>
<td>400</td>
<td>500</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 148.1.0.0</td>
<td>152.1.1.4</td>
<td>0</td>
<td>500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* 148.2.0.0</td>
<td>152.1.1.4</td>
<td>0</td>
<td>400</td>
<td>500</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 148.2.0.0</td>
<td>152.1.1.4</td>
<td>0</td>
<td>500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* 148.3.0.0</td>
<td>152.1.1.4</td>
<td>0</td>
<td>500</td>
<td>500</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 148.3.0.0</td>
<td>152.1.1.4</td>
<td>0</td>
<td>500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* 148.4.0.0</td>
<td>152.1.1.4</td>
<td>0</td>
<td>500</td>
<td>500</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 148.4.0.0</td>
<td>152.1.1.4</td>
<td>0</td>
<td>500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* 152.1.11.128/26</td>
<td>152.1.1.2</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 152.1.11.128/26</td>
<td>152.1.1.2</td>
<td>0</td>
<td>300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* 152.1.12.9</td>
<td>152.1.1.2</td>
<td>0</td>
<td>400</td>
<td>300</td>
<td>i</td>
</tr>
<tr>
<td>* 152.1.12.9</td>
<td>152.1.1.2</td>
<td>0</td>
<td>400</td>
<td>300</td>
<td>i</td>
</tr>
<tr>
<td>* 152.1.12.9</td>
<td>152.1.1.2</td>
<td>0</td>
<td>500</td>
<td>300</td>
<td>i</td>
</tr>
</tbody>
</table>

43. **Advertise network 152.1.9.0 via IBGP on R1 and R2. Do not use the redistribute command.**

Routes can be injected dynamically or statically into BGP. Because the question states that the redistribute command cannot be used, you will need to statically inject routes into BGP. To do this, use the network command, which is a way to individually list prefixes that need to be sent in BGP.

To advertise network 152.1.9.0 /24 via BGP, add the following command under the BGP process. The mask must be added because the network does not fall on a major network
A Case Study in BGP

boundary (255.0.0.0, 255.255.0.0, or 255.255.255.0). For example, the statement 152.1.0.0 would be sufficient to send the prefix 152.1.0.0 /16:

R1(config)#router bgp 100
R1(config-router)#network 152.1.0.0 mask 255.255.255.0

44. To prevent AS 100 from being used as a transit network, AS300 should only accept local networks from AS100. To accomplish this, don’t add any configurations to any router in AS 100.

In order to prevent AS100 from being used as a transit network, either block the sending or receiving of routes that do not originate in AS100. Because the question specifies that you can only add configuration commands to AS300, you must block the receiving of updates from R2 that do not originate in AS100. To do this, apply an inbound filter on BGP updates received from R2.

R6 prevents any network that does not originate on AS 100 from being received from AS100. Filtering routes based on AS path information can be very useful when all routes from a particular AS need to be filtered. If filtering based on an AS path is not used, the administrator would have to list each route one by one or potentially filter on a prefix. AS path filtering provides an efficient alternative to this.

In order to filter routes based on AS path information, you must identify the AS path based on the defined regular expression and apply this to a BGP neighbor through a filter list.

a. Define the regular expression to permit any route that originates from AS100:

R6(config)#ip as-path access-list 2 permit ^100$

Use the show ip bgp regexp command to see which routes the regular expression matches. The following is the output from the command. Note that network 152.1.9.0/24 is the only route that matches the regular expression (^100$). This command is very useful in verifying that the regular expression covers the routes that you intend it to cover.

R6#show ip bgp regexp ^100$
BGP table version is 16, local router ID is 152.1.11.129
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network Next Hop Metric LocPrf Weight Path
*> 152.1.9.0/24 152.1.11.1 0 100 i

b. Define a route map named Receive_Local that permits only local networks routes from AS100. The route map will try to match any path defined by the as-path access-list 2:
R6(config)#route-map Receive_Local permit 10
R6(config-route-map)#match as-path 2

c. Apply the route map to inbound traffic from neighbor (R2) 152.1.11.1:

R2(config-router)#neighbor 152.1.11.1 route-map Receive_Local in

In order for the changes to take effect, the BGP neighbor must be reset. To do this, use the command clear ip bgp *.
This causes the TCP session between neighbors to be reset. This also restarts the neighbor negotiations from scratch and invalidates the cache.

To quickly determine which routes are being learned via AS100, use the command show ip bgp regexp 100.
This command shows any network that has AS100 in its path. The following is the output from the command. Note that only network 152.1.9.0 /24 is being advertised via AS100.

R6#show ip bgp regexp 100
BGP table version is 17, local router ID is 152.1.11.129
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric LocPrf Weight Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* 152.1.9.0/24</td>
<td>152.1.1.1</td>
<td>0 200 100 100 100 i</td>
</tr>
<tr>
<td>*</td>
<td>152.1.1.1</td>
<td>0 500 200 100 100 100 i</td>
</tr>
<tr>
<td>*</td>
<td>152.1.11.1</td>
<td>0 100 i</td>
</tr>
<tr>
<td>*</td>
<td>152.1.1.1</td>
<td>0 400 200 100 100 100 i</td>
</tr>
</tbody>
</table>

45. AS100 prefers that network 152.1.9.0 be reachable by the outside world via the link between AS100 and AS300.

For this question, R3 must be configured so that when it advertises network 152.1.9.0 /24 to AS200 it will not be as preferable as the route through R2. In order to do this, you must first understand how BGP selects its path.

BGP uses a set of parameters (attributes) that describe the characteristics of a route. The attributes are sent in the BGP update packets with each route. The router then uses these attributes to select the best route to the destination.

It is important to understand the BGP decision process in order to be able to correctly manipulate path selection. The following is the order of the BGP decision process used by the router in path selection:

1. If the next hop is unreachable, do not consider it.
2. Prefer the path that has the largest weight.
3. If the routes have the same weight, use the route with the highest local preference.
4. If the routes have the same local preference, prefer the route that was originated by BGP on this router.
5. If no route was originated, prefer the route with the shortest AS path.
A Case Study in BGP

6. If all paths are of the same AS length, prefer the route with lowest origin code (IGP < EGP < INCOMPLETE).
7. If the origin codes are the same, prefer the path with the lowest Multi-Exit Discriminator (MED).
8. If the MEDs are the same, prefer external paths over internal paths.
9. If still the same, prefer the path through the closest IGP neighbor.
10. If still the same, prefer the path with the lowest BGP Router ID

For this question, you will prepend an additional AS number to network 152.1.9.0 /24 before R3 advertises it to AS 200. This makes the AS_path length for this network shorter via AS300 (rule number 5)
In order to manipulate the AS path information, you need to identify which routes will be manipulated through the use of an access list, define a policy that will be applied to those routes through a route map, and then assign the route map to a BGP neighbor:

a. Add access-list 3 to R3, permitting network 152.1.9.0 /24:

   R3#configure terminal
   R3(config)#access-list 3 permit 152.1.9.0 0.0.0.255

b. Define a route map named Add_Path that prepends two additional AS path numbers (AS100 and AS100) to the route if it matches access list 3:

   R3#configure terminal
   R3(config)#route-map Add_Path permit 10
   R3(config-route-map)#match ip address 3
   R3(config-route-map)#set as-path prepend 100 100
   R3(config)#route-map Add_Path permit 20
   R3(config-route-map)#set as-path prepend

   NOTE: Instance 20 of the route map is added so that the other routes will be distributed. When an update does not meet the criteria of a route map instance, BGP applies the next instance. If the update does not meet any criteria, the update is not redistributed or controlled. So if instance 20 is left out, only network 152.1.9.0 would be advertised.

   Apply the route map to outbound routing updates to BGP neighbor 152.1.12.2 (R4) and neighbor 152.1.12.10 (R5):

   R3#configure terminal
   R3(config)#router bgp 100
   R3(config-router)#neighbor 152.1.12.10 route-map Add_Path out
   R3(config-router)#neighbor 152.1.12.2 route-map Add_Path out
In order for the changes to take effect, the BGP neighbors must be reset. To do this, use the command `clear ip bgp *`. This causes the TCP session between neighbors to be reset, the neighbor negotiations to be restarted from scratch, and the cache to be invalidated:

R3# clear ip bgp *

Display the BGP table entry for network 152.1.9.0 on R4 with the command `show ip bgp 152.1.9.0`. The following is the output from the command. Notice that the best route to network 152.1.9.0 is now via AS300.

R5# show ip bgp 152.1.9.0
BGP routing table entry for 152.1.9.0/24, version 37
Paths: (4 available, best #2, advertised over IBGP, EBGP)
500 300 100
  152.1.1.2 from 152.1.1.4 (148.4.0.1)
    Origin IGP, valid, external
300 100
  152.1.1.2 from 152.1.1.2 (152.1.11.129)
    Origin IGP, valid, external, best
400 300 100
  152.1.1.2 from 152.1.1.3 (152.1.20.17)
    Origin IGP, valid, external
100 100 100
  152.1.12.9 from 152.1.12.9 (152.1.10.3)
    Origin IGP, valid, external

46. Under no circumstances can AS100 be used as a transit network for AS200 to reach AS300. To accomplish this, no additional configurations should be made to any router except R2.

The tricky part of this question is how do you prevent routes from being advertised to AS 200 without additional configurations. Because you already have an outbound filter in place, you simply need to remove instance 20 of route map Add_Path. It is added is to permit the rest of the networks to be advertised. By default, when an update does not meet the criteria of a route map instance, BGP applies the next instance. If the update does not meet any criteria, the update is not redistributed or controlled. So if instance 20 is left out, only network 152.1.9.0 would be advertised, which is exactly what you want. So simply remove instance 20 of the route map (no additional configuration changes are made):

R3(config)# no route-map Add_Path permit 20

In order for the changes to take effect, the BGP neighbor must be reset. To do this, use the command `clear ip bgp *`. This causes the TCP session between neighbors to be reset, the neighbor negotiations to be restarted, and the cache to be invalidated.
To quickly determine which routes are being learned via AS100, use the command `show ip bgp regexp 100` on R5. This command shows any network that has AS100 in its path. The following is the output from the command. Note that only network 152.1.9.0 /24 is being advertised via AS100.

```
R5#show ip bgp regexp 100
BGP table version is 26, local router ID is 152.1.14.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
* i 152.1.9.0/24      152.1.12.1                  100      0 100 100 100 i
*>                   152.1.12.9                           0 100 100 100 i
```

47. **Network 152.1.9.0 should not be passed by AS200 or AS300. No additional configuration should be used to achieve this outside of AS100.**

In the past examples, you used filter lists and route maps to prevent updates from being advertised. Because this question requires that no changes be made outside of AS100, you must use community attributes.

To prevent network 152.1.9.0 /24 from being advertised outside AS200 and AS300, you need to use the no-export community value. A community is a group of destinations that shares some common property. They are used to simplify routing policies by identifying routes based on a logical property, rather than an IP prefix or AS number.

The community attribute (type code 8) is an optional transitive attribute that varies in length and consists of a set of four-byte values. Communities in the range of 0x00000000 through 0x0000FFFF and 0xFFFF0000 through 0xFFFFFFFF are reserved. These are well-known communities, which means they have global significance. The no-export community is an example of a well-known community. When a route is carrying this community, it is not advertised to peers outside the confederation. If only one AS is in the confederation, then it is not advertised outside the AS. By default, each AS is a confederation.

In order to prevent routes from being advertised outside of an AS using the no-export community, you first need to identify the prefix using an access list, define the community that is assigned to that prefix with a route map, and apply the route map to a neighbor.

**a.** Define an access list to permit prefix 152.1.9.0 /24. This is already done on R3, so you only need to do this on R2:

```
R2(config)#access-list 3 permit 152.1.9.0 0.0.0.255
```

**b.** A route map named Set_Community is defined to match on prefix 152.1.9.0/24 and set its community attribute to no-export:
R3(config)#route-map Set_Community permit 10
R3(config-route-map)#match ip address 3
R3(config-route-map)#set community no-export

R2(config)#route-map Set_Community permit 10
R2(config-route-map)#match ip address 3
R2(config-route-map)#set community no-export

c. The last step is to apply the route map to a neighbor. The send-community keyword must be assigned to a neighbor session in order to enable the community attribute to be sent to a specified neighbor.

R2(config-router)#neighbor 152.1.11.2 route-map Set_Community out
R2(config-router)#neighbor 152.1.11.2 send-community

R3(config-router)#neighbor 152.1.12.10 route-map Set_Community out
R3(config-router)#neighbor 152.1.12.10 send-community
R3(config-router)#neighbor 152.1.12.2 route-map Set_Community out
R3(config-router)#neighbor 152.1.12.2 send-community

Use the show ip bgp community no-export on R5 to verify that network 152.1.9.0/24 has the community attribute set. The following is the output from the command:

R5#show ip bgp community no-export
BGP table version is 92, local router ID is 152.1.14.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

Network          Next Hop          Metric LocPrf Weight Path
*> 152.1.9.0/24      152.1.12.9                           0 100 i

48. R3 should use R5 to reach network 152.1.14.0 and R4 to reach network 152.1.13.0. No additional configuration should be used to achieve this outside of AS200.

Display the BGP table entry for network 152.1.14.0 /29 on R3. The following is the output from the command. Note the best path is via R4 (152.1.12.2).

R3#show ip bgp 152.1.14.0
BGP routing table entry for 152.1.14.0/29, version 41
Paths: (2 available, best a1, advertised over I BGP)
200 152.1.12.2 from 152.1.12.2 (152.1.13.1)
  Origin IGP, valid, external, best
200 152.1.12.10 from 152.1.12.10 (152.1.14.1)
  Origin IGP, metric 0, valid, external

Display the BGP table entry for network 152.1.13.0 on R3. The following is the output from the command. Note the best path is via R4 (152.1.12.2).

R3#show ip bgp 152.1.13.0
BGP routing table entry for 152.1.13.0/24, version 12
Paths: (2 available, best a1, advertised over I BGP)
A Case Study in BGP

NOTE: The reason that R4 is preferred over R5 is because the router ID is lower, and all other factors are the same. The router uses the route learned from the router with the lowest router ID, which in this case is R4.

Set the MED for prefix 152.1.14.0 advertised from R4 to 50 and prefix 152.1.13.0 to 100. On R5, set the MED for prefix 152.1.14.0 to 100 and prefix 152.1.13.0 to 50. The lower the MED, the more preferred the route.

The MED attribute is the external metric of a route. Unlike the local preference attribute, the MED is exchanged between ASs, but the MED that comes into an AS does not leave.

In order to manipulate the MED, you must identify which networks will be manipulated through the use of an access list, define a policy that will be applied to those routes through a route map, and then assign the route map to a BGP neighbor:

a. Add access-list 1 to R4 and R5, permitting network 152.1.14.0 /29:

    R4(config)#access-list 1 permit 152.1.14.0 0.0.0.7
    R5(config)#access-list 1 permit 152.1.14.0 0.0.0.7

Also add access-list 2 to R4 and R5, permitting network 152.1.13.0 /24:

    R4(config)#access-list 2 permit 152.1.13.0 0.0.0.255
    R5(config)#access-list 2 permit 152.1.13.0 0.0.0.255

b. Define a route map on R4, named Set_Med. The route map sets the MED attribute for network 152.1.14.0 to 50 and sets the MED attribute of network 152.1.13.0 to 100.

    Define a route map on R5, named Set_Med. The first route map sets the MED attribute for network 152.1.13.0 to 50 and the MED attribute of network 152.1.12.0 to 100.
Apply the route map `Set_Med` on outbound routing updates to R3:

R4(config-router)#neighbor 152.1.12.9 route-map Set_Med out
R4(config-router)#neighbor 152.1.12.1 route-map Set_Med out

In order for the changes to take effect, the BGP neighbor's must be reset. To do this, use the command `clear ip bgp *`. This causes the TCP session between neighbors to be reset, the neighbor negotiations to be restarted from scratch, and the cache to be invalidated.

49. R7 should originate and advertise network 148.0.0.0, 148.5.0.0, 148.6.0.0, and 148.7.0.0. Add four loopback interfaces to R7: Loopback 2 IP address 148.0.0.1/16, Loopback 3 IP address 148.5.0.1/16, Loopback 4 IP address 148.6.0.1/16, and Loopback 5 IP address 148.7.0.1/16.

The first part of the question requires the user to create four loopback interfaces and assign IP addresses:

R7(config)#int loopback 2
R7(config-if)#ip address 148.0.0.1 255.255.0.0
R7(config)#int loopback 3
R7(config-if)#ip address 148.5.0.1 255.255.0.0
R7(config)#int loopback 4
R7(config-if)#ip address 148.6.0.1 255.255.0.0
R7(config)#int loopback 5
R7(config-if)#ip address 148.7.0.1 255.255.0.0

In order to advertise a route via BGP, two conditions must be met:

- The BGP process must be aware of the route, either through the use of the network command or by redistribution.
- The network to be advertised must be present in the IP routing table.

The second condition is met because the routes are directly connected. To meet the first condition, use the network command under BGP:

R7(config)#router bgp 400
R7(config-router)#network 148.0.0.0
R7(config-router)#network 148.5.0.0
R7(config-router)#network 148.6.0.0
R7(config-router)#network 148.7.0.0
A Case Study in BGP

50. *R5 should advertise as few prefixes for 148.x.0.0 as needed.*

All the 148.x.0.0 prefixes can be advertised as an aggregate using the aggregate address command. This router configuration command creates an aggregate entry in the BGP routing table if any more specific BGP routes are available that fall in the aggregate range. In this example, the router had routes to networks 148.0.0.0 /16, 148.1.0.0 /16, 148.2.0.0 /16, 148.3.0.0 /16, 148.4.0.0 /16, 148.5.0.0 /16, 148.6.0.0 /16, and 148.7.0.0 /16. All these networks could be advertised in one aggregate route 148.0.0.0/13.

Figure 2-10 shows the concept of route aggregation, also referred to as supernetting. Eight Class B networks, each using a standard 16-bit mask, are shown.

- 148.0.0.0 /16
- 148.1.0.0 /16
- 148.2.0.0 /16
- 148.3.0.0 /16
- 148.4.0.0 /16
- 148.5.0.0 /16
- 148.6.0.0 /16
- 148.7.0.0 /16

Without supernetting, a router advertising these eight networks would need to send a separate route update for each one. Supernetting enables a router to advertise more than one network with a single advertisement. In the case of your eight networks, a router can advertise the 148.0.0.0 network with a 13-bit mask. Notice that the default 16-bit mask has been shortened to 13 bits. Supernetting works by reducing the number of subnet bits in a routing advertisement. This has the effect of matching several networks with a single routing update.

In Figure 2-9, all eight networks have an exact match for their first 13 bits. The remaining three bits of the original 16-bit mask are now part of the supernet. Notice that the .0, .1, .2, .3, .4, .5, and .7 networks use all eight combinations of the three remaining bits. Thus, a 13-bit subnet mask can be used to advertise eight Class B networks with a single advertisement of 148.0.0.0/13.

If the aggregate command is used with no additional arguments, the aggregate is advertised as originating from the aggregating router and has the atomic aggregate attribute set to show that information might be missing. The more specific routes will also be advertised along with the aggregate. This command has several optional arguments:

**as-set**  The as-set keyword creates an aggregate entry with a path that consists of all the elements contained in all the paths that are being summarized.
summary-only  The summary-only keyword suppresses advertisements of more specific routes to all neighbors.

suppress-map  The suppress-map keyword creates the aggregate route but suppresses advertisements of specified routes. A route map can be used to selectively suppress some more specific routes of the aggregate and allow others.

For this question, you want to send only the summary. The following command aggregates the addresses:

```
R5(config-router)# aggregate-address 148.0.0.0 255.248.0.0 summary-only
```

51.  R3 is running an IGP (EIGRP) on the private link between it and AS400. Make sure that only the IGP route to network 152.1.20.16 is favored over the EBGP route.

Network 152.1.20.16 is learned via both EIGRP and BGP. Because multiple routing protocols are present on most networks, there needs to be a way to select which route to use if a network is announced via multiple sources. Cisco uses a parameter called the administrative distance of a protocol to determine which one is used to route a packet.

The following is a list of routing protocols supported on a Cisco router and the default administrative distance of each one. The lower the distance, the more preferred the routing source.

The table above indicates that a directly connected route is generally preferred over a static route, which is preferred over an EBGP route, and so on. EBGP has a distance of 20, which is preferred over all IGP routes.
Cisco provides a way to force IGP routes to take precedence over EBGP routes. The concept is called backdoor links. EBGP routes can be tagged as backdoor routes, which set the distances of these routes to the same as BGP local or 200. Because the distance is higher than the IGP route, the backdoor IGP route is preferred.

For this question, you will tag network 152.1.20.16 as a backdoor route, in which case R3 will prefer the route learned via EIGRP.

Display the routing table on R3. The following code is the output, and note that network 152.1.20.16 is in the routing table as an EBGP-learned route.

```
R3#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route
Gateway of last resort is not set

152.1.0.0/16 is variably subnetted, 14 subnets, 6 masks
B 152.1.11.128/26 [20/0] via 152.1.12.10, 00:21:27
D 152.1.20.7/32 [90/2297856] via 152.1.20.2, 02:00:30, Serial0/0
C 152.1.20.0/30 is directly connected, Serial0/0
O IA 152.1.9.0/24 [110/18] via 152.1.8.1, 00:16:29, Ethernet0/0
```

A Case Study in BGP

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Administrative Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directly connected</td>
<td>0</td>
</tr>
<tr>
<td>Static</td>
<td>1</td>
</tr>
<tr>
<td>EBGP</td>
<td>20</td>
</tr>
<tr>
<td>EIGRP</td>
<td>90</td>
</tr>
<tr>
<td>IGRP</td>
<td>100</td>
</tr>
<tr>
<td>OSPF</td>
<td>110</td>
</tr>
<tr>
<td>ISIS</td>
<td>115</td>
</tr>
<tr>
<td>RIP</td>
<td>120</td>
</tr>
<tr>
<td>EGP</td>
<td>140</td>
</tr>
<tr>
<td>EIGRP (External)</td>
<td>170</td>
</tr>
<tr>
<td>IBGP</td>
<td>200</td>
</tr>
<tr>
<td>BGP Local</td>
<td>200</td>
</tr>
<tr>
<td>Unknown</td>
<td>255</td>
</tr>
</tbody>
</table>

Cisco provides a way to force IGP routes to take precedence over EBGP routes. The concept is called backdoor links. EBGP routes can be tagged as backdoor routes, which set the distances of these routes to the same as BGP local or 200. Because the distance is higher than the IGP route, the backdoor IGP route is preferred.

For this question, you will tag network 152.1.20.16 as a backdoor route, in which case R3 will prefer the route learned via EIGRP.

Display the routing table on R3. The following code is the output, and note that network 152.1.20.16 is in the routing table as an EBGP-learned route.
To tag the network prefix as a backdoor route, perform the following on R3:

```
R3(config)#router bgp 100
R3(config-router)#network 152.1.20.16 mask 255.255.255.240 backdoor
```

In order for the changes to take effect, the BGP neighbor must be reset. To do this, use the command `clear ip bgp *`. This causes the TCP session between neighbors to be reset, the neighbor negotiations to be restarted from scratch, and the cache to be invalidated.

Display the routing table on R3. The following code is the output. Note that network 152.1.12.0/24 is now in the routing table as an EIGRP learned route.

```
R3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route
Gateway of last resort is not set

152.1.0.0/16 is variably subnetted, 14 subnets, 6 masks
B 152.1.11.128/26 [20/0] via 152.1.12.10, 00:01:59
D 152.1.12.7/32 [90/2297856] via 152.1.20.2, 02:07:46, Serial1/0
C 152.1.12.0/30 is directly connected, Serial1/0
O IA 152.1.9.0/24 [110/18] via 152.1.8.1, 00:23:44, Ethernet0/0
C 152.1.10.3/32 is directly connected, Loopback0
C 152.1.8.0/26 is directly connected, Ethernet0/0
O IA 152.1.10.1/32 [110/11] via 152.1.8.1, 00:16:29, Ethernet0/0
D 152.1.13.0/24 [20/50] via 152.1.12.10, 01:59:39
C 152.1.12.0/30 is directly connected, Serial1/0
B 152.1.14.0/29 [20/50] via 152.1.12.10, 01:59:43
D 152.1.1.0/29 [90/2195456] via 152.1.20.2, 02:00:31, Serial0/0
C 152.1.10.8/30 is directly connected, Serial1/1
B 148.0.0.0/13 [20/0] via 152.1.12.10, 01:57:12
```

52. The configuration on R3 is growing extremely large and approaching the limitation of NVRAM. Reduce the size of the configuration on R3.

When the NVRAM is not large enough to store the router configuration, Cisco provides an option that enables the configuration to be compressed. This option should only be used if
A Case Study in BGP

the configuration file is bigger than the memory available in NVRAM. To compress the file, perform the following command:

R3(config)#service compress-config

53. Configure R3 so that informational messages are logged to the internal buffer. Make sure the router can store up to 16K of information and the information is time-stamped.
Message logging to the console port is enabled by default. In order to send messages to any other destination, you must configure the router.
To disable message logging, use the no logging on command. Enabling the logging process can slow down the router because a process must wait until the messages are written to the console before continuing.
To enable message logging to the buffer, perform the following task in global configuration mode:

R3(config)#logging buffered 16000

The logging buffered command copies logging messages to an internal buffer. The buffer is circular, so newer messages overwrite older messages after the buffer is full. The IOS enables the user to configure the buffer size, but the buffer size should not be too large because the router could run out of memory for other tasks.

NOTE: The free processor memory can be viewed on the router with the show memory EXEC command. This number represents the maximum memory available and should not be changed.

In order to correlate events, it is helpful to time-stamp each message with the date and time that it occurred. By default, log messages are not time-stamped. You can enable time-stamping of log messages by performing the following task in global configuration mode:

R3(config)#service timestamps log datetime localtime

54. Configure R3 so that SNMP information can only be read by host 192.1.1.1.
SNMP is made up of three components: an SNMP manager, an SNMP agent, and a Management Information Base (MIB). SNMP provides a message format for sending and receiving information between the manager and the agent. The manager is typically part of a NMS such as HP Open view or Netview 6K. The agent and MIB reside on the router or device that will be managed.
The agent contains the MIB variables, which the SNMP manager can request or change. The manager can request a value from an agent (Read\Get) or change a value on an agent (Write\Set).
Chapter 2

The question specifies that you only want to permit server 192.1.1.1 SNMP read-access. The first step is to specify a community string and permit read-only access. The community string is used as a password to permit access to the agent on the router. Optionally, the MIB view can be set to define a subset of MIBs that the given community can access. By default, the community will have access to all MIBs.

To enable SNMP read-only access to community cisco on R3, perform the following command:

R3(config)#snmp-server community cisco ro

Because you only want to permit access from server 192.1.1.1, you need to define an access list and apply it to the SNMP community string. Define access list 99 to permit IP address 192.1.1.1:

R3(config)#access-list 99 permit 192.1.1.1 0.0.0.0

Apply the access list to the community string:

R3(config)#snmp-server community cisco ro 99

Using the snmp-server community command, you specify a string and, optionally, a MIB view and an access list. The string is used as a password. The MIB view defines the subset of all MIB objects that the given community may access. The access list identifies the IP addresses of systems with SNMPv1 managers that might use the community string to gain access to the SNMPv1 agent.

55. Network 152.1.13.0 is suspected of being unstable, causing BGP UPDATE and WITHDRAWN messages to be repeatedly propagated on the network. Configure R3 so that if the network continues to FLAP it will be withdrawn.

Cisco provides a mechanism to control route instability or flapping, with a feature called route dampening. A route that is unstable, frequently going up and down, causes BGP UPDATE and WITHDRAWN messages to be repeatedly propagated on the network. This routing traffic can quickly use up link bandwidth and central processing unit (CPU) cycles on the router.

The router monitors a route and categorizes it as either well-behaved or ill-behaved. The recent history of the route is used to estimate future stability. That is, a route that has gone up and down two times in the last three minutes is ill-behaved and will be penalized in proportion to the expected future instability. Each time the route flaps, it is given a penalty. When the penalty reaches a predefined threshold, the route is suppressed. The more frequently a route flaps in a period of time, the faster it will be suppressed.

Once the route is suppressed, the router continues to monitor its stability. An algorithm is put in place to reduce (decay) the penalty exponentially. Once the route is deemed stable, it is then advertised.
The following set of terms and parameters applies to Cisco’s implementation of route dampening:

**Penalty**  An incremented numeric number that is assigned to the route each time it flaps.

**Half-lifetime**  A configurable numeric value that describes the amount of time that must pass to reduce the penalty by half.

**Suppress limit**  A numeric value that is compared with the penalty. If the penalty is greater than the suppress limit, the route is suppressed.

**Reuse limit**  A numeric value that is compared with the penalty. If the penalty is less than the reuse limit, a suppressed route will no longer be suppressed.

**Suppressed route**  A route that is not advertised even though it is up. If the penalty is greater than the suppress limit, the route is suppressed.

**History entry**  An entry used to store flap information.

To apply route dampening to a specific network, use a route map to identify the match criteria and then apply the dampening using the route map.

The first step is to define an access list to identify the network address 152.1.13.0:

```
R3(config)# access-list 5 permit 152.1.13.0 0.0.0.255
```

Next, define a route-map named dampen_152_1_13_0 and apply the dampening parameters:

```
R3(config)# route-map dampen_152_1_13_0 permit 10
R3(config-route-map)# set dampening 20 950 2500 80
```

This first parameter sets the time that must pass before the penalty of a suppressed route is reduced by half (half-life) to 20 minutes. The next number sets the limit, after which a suppressed route will be readvertised (reuse limit) to 950. The next number sets the limit at which a route will be suppressed to 2500. If the penalty exceeds 2500, the route will be suppressed. The last value is the maximum time in minutes that the route can be suppressed for (max-suppress-time), which in this case is 80 minutes.

The final step is to apply the route map to the BGP process:

```
R3(config)# router bgp 100
R3(config-router)# bgp dampening route-map dampen_152_1_13_0
```

In order for the changes to take effect, the BGP neighbor must be reset. To do this, use the command `clear ip bgp *`. This causes the TCP session between neighbors to be...
reset, the neighbor negotiations to be restarted from scratch, and the cache to be invalidated.

Let’s take a look and see what happens when the network 152.1.13.0 flaps. Shut down interface loopback 1 on R4, which will cause the route flap for network 152.1.13.0.

The following output shows how R3 treats the flapping route 152.1.13.0. Note that the route has flapped twice, because the router counts a flap as any time the path information changes for a route. Because the route went up and down, this is treated as two flaps.

```
R3#show ip bgp 152.1.13.0
BGP routing table entry for 152.1.13.0/24 , version 8
Paths: (1 available, best a1)
   200
     152.1.12.2 from 152.1.12.2 (152.1.13.1)
       Origin IGP, metric 50, valid, external, best
       Dampinfo: penalty 1472, flapped 2 times in 00:00:49
```

You can cause the network to flap again by shutting down the loopback interface on R4. The following output shows the route after four flaps. The penalty is now 2652, which is higher than the 2500 limit. The route has been suppressed and will not be passed on. The route will be unusable for 29 minutes and 40 seconds, at which time, if no other flaps occur, the penalty will be decayed to the reuse limit of 950.

```
R3#show ip bgp 152.1.13.0
BGP routing table entry for 152.1.13.0/24 , version 9
Paths: (1 available, no best path)
   200, (suppressed due to dampening)
     152.1.12.2 from 152.1.12.2 (152.1.13.1)
       Origin IGP, metric 50, valid, external
       Dampinfo: penalty 2652, flapped 4 times in 00:07:32, reuse in 00:29:40
```

Enable R3 so that when its neighbors make a BGP change, the change will take effect without resetting the BGP TCP session. R3 should be able to initiate the change locally; no commands should have to be entered on the neighboring routers.

One requirement of BGP is resetting the neighbors’ TCP connection in order for a new policy to take effect. BGP soft configuration enables policies to be configured and activated without resetting the BGP and TCP sessions. This enables the new policy to take effect without significantly affecting the network. Without BGP soft configuration, BGP is required to reset the neighbor TCP connection in order for the new changes to take effect. This is accomplished using the clear ip bgp * command, which is used throughout this chapter.

Two types of BGP soft reconfiguration exist: outbound reconfiguration, which will make the new local outbound policy take effect without resetting the BGP session, and inbound soft reconfiguration, which enables the new inbound policy to take effect.
A Case Study in BGP

This question requires that you use inbound soft reconfiguration on R1. To accomplish this, an additional router command needs to be added before a soft reconfiguration can be issued. The reason is that this command tells the router to start storing the received updates.

Add the following configurations to the BGP process on R1:

R1(config)#router bgp 100
R1(config-router)#neighbor 152.1.10.10 soft-reconfiguration inbound
R1(config-router)#neighbor 152.1.10.11 soft-reconfiguration inbound

Inbound soft reconfiguration can then be triggered with the command:

clear ip bgp [*|address | peer-group ] [soft in]

The problem with inbound reconfiguration is that in order to generate new inbound updates without resetting the BGP session, all inbound updates (whether accepted or rejected) need to be stored by the router. This is memory-intensive and wherever possible it should be avoided.

To avoid the memory overhead needed for inbound soft reconfiguration, the same outcome could be achieved by doing an outbound soft reconfiguration at the other end of the connection.

An outbound soft reconfiguration can be triggered with the command:

clear ip bgp [*|address | peer-group ] [soft out]

---

Complete Network Diagram

Table 2-2 shows the IP address worksheet for this Case Study. Figure 2-9 shows the complete network diagram with IP addressing.
### Table 2-2

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Mask</th>
<th>Subnet Address</th>
<th>Broadcast Address</th>
<th>Router</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>148.0.0.1</td>
<td>255.255.0.0</td>
<td>148.0.0.0</td>
<td></td>
<td>R7</td>
<td>L2</td>
</tr>
<tr>
<td>148.1.0.1</td>
<td>255.255.0.0</td>
<td>148.1.0.0</td>
<td></td>
<td>R8</td>
<td>L0</td>
</tr>
<tr>
<td>148.2.0.1</td>
<td>255.255.0.0</td>
<td>148.2.0.0</td>
<td></td>
<td>R8</td>
<td>L1</td>
</tr>
<tr>
<td>148.3.0.1</td>
<td>255.255.0.0</td>
<td>148.3.0.0</td>
<td></td>
<td>R8</td>
<td>L3</td>
</tr>
<tr>
<td>148.4.0.1</td>
<td>255.255.0.0</td>
<td>148.4.0.0</td>
<td></td>
<td>R8</td>
<td>L4</td>
</tr>
<tr>
<td>148.5.0.1</td>
<td>255.255.0.0</td>
<td>148.5.0.0</td>
<td></td>
<td>R7</td>
<td>L3</td>
</tr>
<tr>
<td>148.6.0.1</td>
<td>255.255.0.0</td>
<td>148.6.0.0</td>
<td></td>
<td>R7</td>
<td>L4</td>
</tr>
<tr>
<td>148.7.0.1</td>
<td>255.255.0.0</td>
<td>148.7.0.0</td>
<td></td>
<td>R7</td>
<td>L5</td>
</tr>
<tr>
<td>152.1.1.1</td>
<td>255.255.255.248</td>
<td>152.1.1.0</td>
<td>152.1.1.7</td>
<td>R5</td>
<td>E0</td>
</tr>
<tr>
<td>152.1.1.2</td>
<td>255.255.255.248</td>
<td>152.1.1.0</td>
<td>152.1.1.7</td>
<td>R6</td>
<td>E0</td>
</tr>
<tr>
<td>152.1.1.3</td>
<td>255.255.255.248</td>
<td>152.1.1.0</td>
<td>152.1.1.7</td>
<td>R7</td>
<td>E0</td>
</tr>
<tr>
<td>152.1.1.4</td>
<td>255.255.255.248</td>
<td>152.1.1.0</td>
<td>152.1.1.7</td>
<td>R8</td>
<td>E0</td>
</tr>
<tr>
<td>152.1.1.8</td>
<td>255.255.255.192</td>
<td>152.1.8.0</td>
<td>152.1.8.63</td>
<td>R1</td>
<td>E0/0</td>
</tr>
<tr>
<td>152.1.1.8</td>
<td>255.255.255.192</td>
<td>152.1.8.0</td>
<td>152.1.8.63</td>
<td>R3</td>
<td>E0/0</td>
</tr>
<tr>
<td>152.1.1.9</td>
<td>255.255.255.0</td>
<td>152.1.9.0</td>
<td>152.1.9.255</td>
<td>R1</td>
<td>E0/1</td>
</tr>
<tr>
<td>152.1.9.254</td>
<td>255.255.255.0</td>
<td>152.1.9.0</td>
<td>152.1.9.255</td>
<td>R2</td>
<td>E0</td>
</tr>
<tr>
<td>152.1.10.1</td>
<td>255.255.255.255</td>
<td>152.1.10.1</td>
<td></td>
<td>R1</td>
<td>L0</td>
</tr>
<tr>
<td>152.1.10.2</td>
<td>255.255.255.255</td>
<td>152.1.10.2</td>
<td></td>
<td>R2</td>
<td>L0</td>
</tr>
<tr>
<td>152.1.10.3</td>
<td>255.255.255.255</td>
<td>152.1.10.3</td>
<td></td>
<td>R3</td>
<td>L0</td>
</tr>
<tr>
<td>152.1.10.9</td>
<td>255.255.255.248</td>
<td>152.1.10.8</td>
<td>152.1.10.15</td>
<td>R1</td>
<td>S0/0.1</td>
</tr>
<tr>
<td>152.1.10.10</td>
<td>255.255.255.248</td>
<td>152.1.10.8</td>
<td>152.1.10.15</td>
<td>R2</td>
<td>S0</td>
</tr>
<tr>
<td>152.1.10.11</td>
<td>255.255.255.248</td>
<td>152.1.10.8</td>
<td>152.1.10.15</td>
<td>R3</td>
<td>S0/1</td>
</tr>
<tr>
<td>152.1.11.1</td>
<td>255.255.255.252</td>
<td>152.1.11.0</td>
<td>152.1.11.3</td>
<td>R2</td>
<td>S1</td>
</tr>
<tr>
<td>152.1.11.2</td>
<td>255.255.255.252</td>
<td>152.1.11.0</td>
<td>152.1.11.3</td>
<td>R6</td>
<td>S0</td>
</tr>
<tr>
<td>152.1.11.6</td>
<td>255.255.255.255</td>
<td>152.1.11.6</td>
<td></td>
<td>R6</td>
<td>L0</td>
</tr>
<tr>
<td>152.1.11.129</td>
<td>255.255.255.192</td>
<td>152.1.11.128</td>
<td>152.1.11.191</td>
<td>R6</td>
<td>L1</td>
</tr>
<tr>
<td>152.1.12.1</td>
<td>255.255.255.252</td>
<td>152.1.12.0</td>
<td>152.1.12.3</td>
<td>R3</td>
<td>S1/0</td>
</tr>
<tr>
<td>152.1.12.2</td>
<td>255.255.255.252</td>
<td>152.1.12.0</td>
<td>152.1.12.3</td>
<td>R4</td>
<td>S0</td>
</tr>
<tr>
<td>152.1.12.4</td>
<td>255.255.255.255</td>
<td>152.1.12.4</td>
<td></td>
<td>R4</td>
<td>L0</td>
</tr>
<tr>
<td>152.1.12.5</td>
<td>255.255.255.255</td>
<td>152.1.12.5</td>
<td></td>
<td>R5</td>
<td>L0</td>
</tr>
<tr>
<td>152.1.12.9</td>
<td>255.255.255.252</td>
<td>152.1.12.8</td>
<td>152.1.12.11</td>
<td>R3</td>
<td>S1/1</td>
</tr>
<tr>
<td>152.1.12.10</td>
<td>255.255.255.252</td>
<td>152.1.12.8</td>
<td>152.1.12.11</td>
<td>R5</td>
<td>S0</td>
</tr>
<tr>
<td>152.1.12.13</td>
<td>255.255.255.252</td>
<td>152.1.12.12</td>
<td>152.1.12.15</td>
<td>R4</td>
<td>S1</td>
</tr>
<tr>
<td>152.1.12.14</td>
<td>255.255.255.252</td>
<td>152.1.12.12</td>
<td>152.1.12.15</td>
<td>R5</td>
<td>S1</td>
</tr>
<tr>
<td>152.1.13.1</td>
<td>255.255.255.255</td>
<td>152.1.13.0</td>
<td>152.1.13.255</td>
<td>R4</td>
<td>L0</td>
</tr>
<tr>
<td>152.1.14.1</td>
<td>255.255.255.248</td>
<td>152.1.14.0</td>
<td>152.1.14.8</td>
<td>R5</td>
<td>L0</td>
</tr>
<tr>
<td>152.1.20.1</td>
<td>255.255.255.252</td>
<td>152.1.20.0</td>
<td>152.1.20.3</td>
<td>R3</td>
<td>S0/0</td>
</tr>
<tr>
<td>152.1.20.2</td>
<td>255.255.255.252</td>
<td>152.1.20.0</td>
<td>152.1.20.3</td>
<td>R7</td>
<td>S0</td>
</tr>
<tr>
<td>152.1.20.7</td>
<td>255.255.255.255</td>
<td>152.1.20.7</td>
<td></td>
<td>R7</td>
<td>L0</td>
</tr>
<tr>
<td>152.1.20.17</td>
<td>255.255.255.240</td>
<td>152.1.20.16</td>
<td>152.1.20.31</td>
<td>R7</td>
<td>L1</td>
</tr>
</tbody>
</table>
A Case Study in BGP

Figure 2-9
Network diagram with IP addressing
Router Configurations

R1

! version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname R1
!
!
interface Loopback0
   ip address 152.1.10.1 255.255.255.255
!
interface Ethernet0/0
   mac-address 0001.0100.0002
   ip address 152.1.8.1 255.255.255.192
   ip ospf priority 2
!
interface Serial0/0
   no ip address
   encapsulation frame-relay IETF
   no fair-queue
   frame-relay lmi-type ansi
!
interface Serial0/0 .1 multipoint
   ip address 152.1.10.9 255.255.255.248
   ip ospf cost 9
   ip ospf hello-interval 120
   ip ospf priority 255
   frame-relay map ip 152.1.10.10 100 broadcast
   frame-relay map ip 152.1.10.11 200 broadcast
   no frame-relay inverse-arp
!
interface Ethernet1/0
   ip address 152.1.9.1 255.255.255.0
   ip ospf cost 8
!
router ospf 64
   passive-interface Loopback0
   network 152.1.10.8 0.0.0.7 area 0
   network 152.1.8.0 0.0.0.63 area 1
   network 152.1.9.0 0.0.0.255 area 2
   network 152.1.10.1 0.0.0.0 area 0
   neighbor 152.1.10.11
   neighbor 152.1.10.10
!
router bgp 100
   no synchronization
   network 152.1.9.0 mask 255.255.255.0
   neighbor 152.1.10.10 remote-as 100
   neighbor 152.1.10.10 route-reflector-client
A Case Study in BGP

neighbor 152.1.10.10 soft-reconfiguration inbound
neighbor 152.1.10.11 remote-as 100
neighbor 152.1.10.11 route-reflector-client
neighbor 152.1.10.11 soft-reconfiguration inbound

! no ip classless

!
!
line con 0
line aux 0
line vty 0 4
login

end

R2

!
version 11.2
no service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname R2
!
interface Loopback0
ip address 152.1.10.2 255.255.255.255
!
interface Ethernet0
ip address 152.1.9.254 255.255.255.0
ip ospf priority 2
!
interface Serial0
ip address 152.1.10.10 255.255.255.248
encapsulation frame-relay IETF
ip ospf hello-interval 120
ip ospf priority 0
frame-relay map ip 152.1.10.9 100 broadcast
frame-relay map ip 152.1.10.11 100 broadcast
no frame-relay inverse-arp
frame-relay lmi-type ansi
!
interface Serial1
ip address 152.1.11.1 255.255.255.252
!
routerr ospf 64
passive-interface Loopback0
network 152.1.10.8 0.0.0.7 area 0
network 152.1.9.0 0.0.0.255 area 2
network 152.1.10.2 0.0.0.0 area 0
!
routerr bgp 100
no synchronization
neighbor 152.1.10.9 remote-as 100
neighbor 152.1.10.9 next-hop-self
neighbor 152.1.11.2 remote-as 300
neighbor 152.1.11.2 send-community
neighbor 152.1.11.2 route-map Accept_Local in
neighbor 152.1.11.2 route-map Set_Community out
!
no ip classless
ip as-path access-list 1 permit ^300$
ip as-path access-list 2 permit ^$
access-list 3 permit 152.1.9.0 0.0.0.255
route-map Send_Local permit 10
  match as-path 2
  route-map Accept_Local permit 10
    match as-path 1
  route-map Set_Community permit 10
  match ip address 3
  set community no-export
!
!
line con 0
line 1 16
line aux 0
line vty 0 4
  login
!
end

R3

!
version 11.1
service timestamps log datetime localtime
service compress-config
service udp-small-servers
service tcp-small-servers
!
hostname R3
!
interface Loopback0
  ip address 152.1.10.3 255.255.255.255
!
!
interface Ethernet0/0
  ip address 152.1.8.62 255.255.255.192
!
interface Serial0/0
  ip address 152.1.20.1 255.255.255.252
!
interface Serial0/1
  ip address 152.1.10.11 255.255.255.248
  encapsulation frame-relay IETF
  ip ospf hello-interval 120
  ip ospf priority 0
  no frame-relay inverse-arp
A Case Study in BGP

frame-relay lmi-type ansi
frame-relay map ip 152.1.10.9 200 broadcast
frame-relay map ip 152.1.10.10 200 broadcast
!
interface Serial1/0
ip address 152.1.12.1 255.255.255.252
!
interface Serial1/1
ip address 152.1.12.9 255.255.255.252
!
router eigrp 64
redistribute ospf 64 metric 1500 100 255 1 1500
!
interface Serial1/0
ip address 152.1.12.1 255.255.255.252
!
interface Serial1/1
ip address 152.1.12.9 255.255.255.252
!
router ospf 64
redistribute eigrp 64 metric 64 subnets
!
router bgp 100
no synchronization
bgp dampening route-map dampen_152_1_13_0
network 152.1.10.9 0.0.0.0 remote-as 100
neighbor 152.1.10.9 remote-as 100
distribute-list 1 out eigrp 64
!
router ospf 64
redistribute eigrp 64 metric 64 subnets
!
router ospf 64
redistribute eigrp 64 metric 64 subnets
!
router bgp 100
no synchronization
bgp dampening route-map dampen_152_1_13_0
network 152.1.10.9 0.0.0.0 remote-as 100
distribute-list 1 out eigrp 64
!
router ospf 64
redistribute eigrp 64 metric 64 subnets
!
router ospf 64
redistribute eigrp 64 metric 64 subnets
!
router bgp 100
no synchronization
bgp dampening route-map dampen_152_1_13_0
network 152.1.10.9 0.0.0.0 remote-as 100
neighbor 152.1.10.9 remote-as 100
distribute-list 1 out eigrp 64
!
route-map Set_Community permit 10
   match ip address 3
   set community no-export
!
snmp-server community cisco RO 99
!
line con 0
line aux 0
line vty 0 4
   login
!
end

R4
!
version 11.2
service udp-small-servers
service tcp-small-servers
!
hostname R4
!
!
interface Loopback0
   ip address 152.1.12.4 255.255.255.255
!
interface Loopback1
   ip address 152.1.13.1 255.255.255.0
!
interface Serial0
   ip address 152.1.12.2 255.255.255.252
   clockrate 1000000
!
interface Serial1
   ip address 152.1.12.13 255.255.255.252
!
router eigrp 56
   passive-interface Serial0
   network 152.1.0.0
!
router bgp 200
   no synchronization
   network 152.1.13.0 mask 255.255.255.0
   neighbor 152.1.12.1 remote-as 100
   neighbor 152.1.12.1 route-map Set_Med out
   neighbor 152.1.12.14 remote-as 200
!
no ip classless
access-list 1 permit 152.1.14.0 0.0.0.7
access-list 2 permit 152.1.13.0 0.0.0.255
A Case Study in BGP

route-map Set_Med permit 10
 match ip address 1
  set metric 100
!
route-map Set_Med permit 20
 match ip address 2
  set metric 50
!
route-map Set_Med permit 30
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
no synchronization
network 152.1.14.0 mask 255.255.255.248
aggregate-address 148.0.0.0 255.248.0.0 summary-only
neighbor 152.1.1.2 remote-as 300
neighbor 152.1.1.3 remote-as 400
neighbor 152.1.1.4 remote-as 500
neighbor 152.1.12.9 remote-as 100
neighbor 152.1.12.9 route-map Set_Med out
neighbor 152.1.12.13 remote-as 200
!
access-list 1 permit 152.1.14.0 0.0.0.7
access-list 2 permit 152.1.13.0 0.0.0.255
route-map Set_Med permit 10
  match ip address 1
  set metric 50
!
route-map Set_Med permit 20
  match ip address 2
  set metric 100
!
route-map Set_Med permit 30
!
line con 0
  ip netmask-format bit-count
line 1 16
  transport input all
line aux 0
  transport input all
line vty 0 4
  login
!
end

R6
!
version 11.0
service udp-small-servers
service tcp-small-servers
!
hostname R6
!
enable password cisco
!
interface Loopback0
  ip address 152.1.11.6 255.255.255.255
!
interface Loopback1
  ip address 152.1.11.129 255.255.255.192
!
interface Ethernet0
  mac-address 0006.0300.0000
A Case Study in BGP

ip address 152.1.1.2 255.255.255.248
!
interface Serial0
  ip address 152.1.1.2 255.255.255.252
  clockrate 64000
!
interface Serial1
  no ip address
  shutdown
!
router bgp 300
  network 152.1.1.128 mask 255.255.255.192
  neighbor 152.1.1.1 remote-as 200
  neighbor 152.1.1.3 remote-as 400
  neighbor 152.1.1.4 remote-as 500
  neighbor 152.1.1.11 remote-as 100
  neighbor 152.1.1.11 default-originate
  neighbor 152.1.1.11 route-map Receive_Local in
!
ip as-path access-list 2 permit ^100$
route-map Receive_Local permit 10
  match as-path 2
!
!
line con 0
line aux 0
  transport input all
line vty 0 4
  ip netmask-format bit-count
  exec-timeout 30 0
  password cisco
  login
!
end

R7

! 
version 11.2
no service udp-small-servers
no service tcp-small-servers
!
hostname R7
!
enable password cisco
!
ip host r1 2001 152.1.1.3
ip host r2 2002 152.1.1.3
ip host r4 2004 152.1.1.3
ip host r5 2005 152.1.1.3
ip host r6 2006 152.1.1.3
ip host r7 2007 152.1.1.3
ip host r8 2008 152.1.1.3
ip host cat 2009 152.1.1.3
interface Loopback0
 ip address 152.1.20.7 255.255.255.255
!
interface Loopback1
 ip address 152.1.20.17 255.255.255.240
!
interface Loopback2
 ip address 148.0.0.1 255.255.0.0
!
interface Loopback3
 ip address 148.5.0.1 255.255.0.0
!
interface Loopback4
 ip address 148.6.0.1 255.255.0.0
!
interface Loopback5
 ip address 148.7.0.1 255.255.0.0
!
interface Ethernet0
 mac-address 0007.0300.0000
 ip address 152.1.1.3 255.255.255.248
!
interface Serial0
 ip address 152.1.20.2 255.255.255.252
 no fair-queue
clockrate 64000
!
router eigrp 64
 passive-interface Ethernet0
 network 152.1.0.0
distribute-list 1 out eigrp 64
!
router bgp 400
 network 152.1.20.16 mask 255.255.255.240
 network 148.0.0.0
 network 148.5.0.0
 network 148.7.0.0
 network 148.6.0.0
 neighbor 152.1.1.1 remote-as 200
 neighbor 152.1.1.2 remote-as 300
 neighbor 152.1.1.4 remote-as 500
!
no ip classless
access-list 1 permit 152.1.20.7
access-list 1 permit 152.1.1.0 0.0.0.7
access-list 1 permit 152.1.20.16 0.0.0.15
access-list 1 deny any
!
line con 0
line 1 16
 no exec
transport input all
line aux 0
line vty 0 4
 password cisco
 login
!
end
Frame Switch

! version 11.1
service slave-log
service udp-small-servers
service tcp-small-servers
!
hostname R8
!
frame-relay switching
!
interface Loopback0
  ip address 148.1.0.1 255.255.0.0
!
interface Loopback1
  ip address 148.2.0.1 255.255.0.0
!
interface Loopback3
  ip address 148.3.0.1 255.255.0.0
!
interface Loopback4
  ip address 148.4.0.1 255.255.0.0
!
interface Ethernet0
  mac-address 0008.0500.0000
  ip address 152.1.1.4 255.255.255.248
  media-type 10BaseT
!
interface Serial0
  description Frame Relay connection to R1 s0
  no ip address
  encapsulation frame-relay IETF
  no fair-queue
  clockrate 64000
  frame-relay lmi-type ansi
  frame-relay intf-type dce
  frame-relay route 100 interface Serial1 100
  frame-relay route 200 interface Serial2 200
!
interface Serial1
  description Frame Relay connection to R2 s0
  no ip address
  encapsulation frame-relay IETF
  clockrate 64000
  frame-relay lmi-type ansi
  frame-relay intf-type dce
  frame-relay route 100 interface Serial0 100
!
interface Serial2
  description Frame Relay connection to R3 s 1
  no ip address
  encapsulation frame-relay IETF
  clockrate 64000
  frame-relay lmi-type ansi
  frame-relay intf-type dce
frame-relay route 200 interface Serial0 200

! router bgp 500
  network 148.2.0.0
  network 148.3.0.0
  network 148.4.0.0
  network 148.1.0.0
  neighbor 152.1.1.1 remote-as 200
  neighbor 152.1.1.2 remote-as 300
  neighbor 152.1.1.3 remote-as 400

  no ip classless

! line con 0
! line aux 0
! line vty 0 4
  login

end

BGP Technology Appendix

BGP Message Format

All BGP messages use the same fixed size header, which is shown here.

<table>
<thead>
<tr>
<th>0</th>
<th>15</th>
<th>23</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MARKER</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MARKER</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MARKER</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MARKER</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LENGTH</td>
<td>TYPE</td>
<td></td>
</tr>
</tbody>
</table>

**Marker**  This is a 16-byte field that is used to either authenticate incoming BGP messages or to detect a loss of synchronization between BGP peers. If the type of message is OPEN or if the OPEN message carries no authentication information, then the marker is all ones. Otherwise, the marker field is computed based on the authentication being used.
A Case Study in BGP

**Length** The length field is a two-byte field that indicates the total length of the message. The smallest permitted length is 19 bytes and the largest is 4,096.

**Type:** The type field is a one-byte field that indicates the type of BGP message. Four BGP message types are available:

- OPEN
- UPDATE
- KEEPALIVE
- NOTIFICATION

**OPEN Message Format**

BGP can not exchange routing information until the neighbor negotiation is complete. After the transport connection is established, the first message is an OPEN message. This message contains information on the BGP version number, the AS number, the hold time, the BGP identifier, and other optional parameters. If the peers disagree on any of the parameters, a notification error is sent and the peer connection does not get established.

If the OPEN message is acceptable, meaning the peer router agrees on the parameters, then a KEEPALIVE message is sent back to confirm the OPEN message.

In addition to the fixed-size BGP header, the OPEN message contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>The version field is a one-byte field that indicates the version of the BGP protocol. During the neighbor negotiation, peer routers agree on the BGP version number to be used. The highest version that both routers support is usually used.</td>
</tr>
</tbody>
</table>
My Autonomous System  This field is two bytes long and indicates the AS number of the sending router.

Hold Time  This field is two bytes long and indicates the maximum time in seconds that may elapse between the receipt of a successive KEEPALIVE and/or UPDATE message. If the hold time is exceeded, the neighbor is considered dead.

The hold time is negotiated between neighbors and is set to the lowest value. The router that receives the open message must calculate the hold time by using the smallest of its configured hold times and the hold time received in the open message.

BGP Identifier  This field is four bytes long and indicates the BGP identifier of the sending router. This is the router ID, which is the highest loopback address or the highest IP address on the router at BGP session startup.

Optional Parameter Length  This field is one byte long and indicates the total length in bytes of the optional parameter field. If no optional parameters are present, the field is set to 0.

Optional Parameters  This is a variable-length field that indicates the list of optional parameters used in BGP neighbor session negotiation.

UPDATE Message Format

The UPDATE message is the primary message used to communicate information between BGP peers. When a BGP speaker advertises or withdraws a route from a peer router, an UPDATE message is used. The UPDATE message always includes the fixed-length BGP header and can optionally include the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unfeasible Routes Length</td>
<td>Two bytes</td>
</tr>
<tr>
<td>Withdrawn Routes</td>
<td>Variable</td>
</tr>
<tr>
<td>Total Path Attribute Length</td>
<td>Two bytes</td>
</tr>
<tr>
<td>Path Attributes</td>
<td>Variable</td>
</tr>
<tr>
<td>Network Layer Reachability Information</td>
<td>Variable</td>
</tr>
</tbody>
</table>

Unfeasible Routes Length  This two-byte field indicates the total length in octets of the Withdrawn Routes field. If the value is 0, it indicates that no routes are being withdrawn from service.
A Case Study in BGP

Withdrawn Routes  This variable-length field contains a list of IP address prefixes of the routes that are being withdrawn from service.

Total Path Attribute Length  This two-byte field indicates the total length in octets of the Path Attributes field.

Path Attributes  This variable-length field contains a list of BGP attributes associated with the prefixes in the Network Layer Reachability field. The path attributes give information on the prefixes that are being advertised, such as the degree of preference or the origin of the prefix. This information is used for filtering and in the route decision process. The path attributes fall into four categories:

- **Well-Known Mandatory**: An attribute that has to exist in the BGP update and must be recognized by all BGP vendor implementations. ORIGIN, AS_PATH, and NEXT_HOP are three examples of well-know mandatory attributes.

  - **ORIGIN**: The ORIGIN attribute is an example of a well-known mandatory attribute that indicates the origin of the routing update with respect to the AS that originated it. This attribute tells how the original route was put into the BGP table. A route could be learned via an IGP such as OSPF, which was redistributed into BGP, learned through an external routing protocol (such as EGP), or learned through something other than an IGP or EGP, such as a static route.

    Three origins are possible: IGP, EGP, or INCOMPLETE. The router uses this information in its decision process when choosing between multiple routes. The router prefers the path with the lowest ORIGIN type, with an IGP lower than EGP, and with an EGP that is lower than INCOMPLETE.

  - **AS_PATH**: AS_PATH is a well-known mandatory attribute that indicates the ASs through which routing information contained in this UPDATE message has passed. In Figure 2-11, prefix 1.0.0.0 /8 is advertised to AS 300 and AS 200. When AS 300 passes the prefix to AS 200, it appends its AS number to the AS_PATH. When AS 200 receives the UPDATE from AS300, it knows that 1.0.0.0 originated in AS100 and then passed through AS300.

  - **NEXT_HOP**: This attribute defines the IP address of the border router that should be used as the next hop to the destinations listed in the UPDATE message. For example, in Figure 2-12, RouterB learns route 1.0.0.0 /8 via 192.1.1.1, which is the IP address of its EBGP peer. When RouterB advertises this route to RouterC, it includes the next hop information unaltered. RouterC then receives an IBGP update for network 1.0.0.0 /8 with a next hop of 192.1.1.1.

- **Well-know discretionary**: An attribute that must be recognized by all BGP implementations but may or may not be sent in a BGP update.

  - **LOCAL_PREF**: The local preference attribute is a degree of preference given to a route to compare it with other routes to the same destination. The highest local preference is the
most preferred route. The local preference is not included in update messages that are sent to BGP neighbors outside of the AS. If the attribute is contained in an update received from a BGP neighbor in a different AS, it is ignored.

In Figure 2-13, you see where the local preference attribute would be used. RouterB and RouterC are both advertising network 1.0.0.0 /8 into AS 400, but because the link between RouterB and RouterD is a high-speed link, you would like traffic destined for network 1.0.0.0 /8 to use this route. The local preference attribute will be used to manipulate the flow of traffic within AS 400. RouterD gives routes coming from RouterB a local preference of 150, and RouterE gives routes coming from RouterC a local preference of 100. Because RouterD and RouterE are exchanging routes via IBGP, they both will use the route with the highest local preference. In this case, all traffic
destined for network 1.0.0.0 /8 will be routed over the high-speed link between RouterB and RouterD.

- **ATOMIC_AGGREGATE**: The atomic aggregate attribute indicates that information has been lost. When routes are aggregated, this causes a loss of information because the aggregate is coming from different sources that have different attributes. If a router sends an aggregate that causes a loss of information, it is required to attach the atomic aggregate attribute to the route.

- **Optional Transitive**: An optional transitive attribute is not required to be supported by all BGP implementations. However, if it is not recognized by the BGP process, it will look at the transitive flag. If the flag is set, the BGP implementation should accept the attribute and pass it along to other BGP speakers.

- **AGGREGATOR**: This attribute identifies the BGP speaker (IP address) and AS number that performed the route aggregation.

- **Optional nontransitive**: An optional nontransitive attribute is not required to be supported by all BGP implementations. However, if the attribute is not recognized by the BGP process, it will look at the transitive flag. If the flag is not set, the attribute should be quietly ignored and not passed along to other BGP peers.
**MULTI_EXIT_DISC (MED):** MED is used by the BGP speakers to discriminate between multiple exit points to a neighboring AS. A lower MED is preferred over a higher one. The MED attribute is exchanged between ASs, but a MED attribute that comes into an AS does not leave the AS (nontransitive). This differs from the local preference attribute. External routers can influence the path selection of another AS, whereas with local preference you can only influence the route selection within your own AS.

In Figure 2-14, the MED attribute is used to influence the path that RouterA uses to reach network 1.0.0.0 /8. Both RouterB and RouterC are advertising network 1.0.0.0 /8 to RouterA. However, because RouterC is closer to this network, you would like all traffic destined to network 1.0.0.0 /8 from RouterA to be routed through RouterC. To achieve this, set the MED on route 1.0.0.0 /8 advertised from RouterC to 100 and the MED on route 1.0.0.0 /8 advertised from RouterB to 200. Because the MED coming from RouterC is lower, RouterA will prefer this route.

**Network Layer Reachability:** This variable-length field contains a list of IP address prefixes that are reachable via the sender. All path attributes contained in a given update message apply to the destinations carried in the Network Layer Reachablity Information field of the UPDATE message.
A Case Study in BGP

**KEEPALIVE Message Format**

KEEPALIVE messages are exchanged periodically between peers to determine whether peers are reachable. These messages are exchanged between peers often enough to prevent the hold timer from expiring.

**NOTIFICATION Message Format**

A NOTIFICATION message is sent whenever an error condition is detected. The BGP connection is closed immediately after sending it. In addition to the fixed-size BGP header, the NOTIFICATION message contains the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error code</td>
<td>One byte</td>
</tr>
<tr>
<td>Error Subcode</td>
<td>One byte</td>
</tr>
<tr>
<td>Data</td>
<td>Variable</td>
</tr>
</tbody>
</table>

**Error Code**  This one-byte field indicates the type of notification. The following are the possible types:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Symbolic Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Message Header Error</td>
</tr>
<tr>
<td>2</td>
<td>OPEN Message Error</td>
</tr>
<tr>
<td>3</td>
<td>UPDATE Message Error</td>
</tr>
<tr>
<td>4</td>
<td>Hold Timer Expired</td>
</tr>
<tr>
<td>5</td>
<td>Finite State Machine Error</td>
</tr>
<tr>
<td>6</td>
<td>Cease</td>
</tr>
</tbody>
</table>

**Error Subcode**  This one-byte field provides more specific information about the nature of the reported error. Each error code may have one or more error subcodes associated with it. The following is a list of error subcodes:

**Data**  This variable length field is used to diagnose the reason for the notification. The contents of the Data field depend upon the error code and error subcode.
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Subcode</th>
</tr>
</thead>
</table>
| 1- Message Header Error | — Connection Not Synchronized  
— Bad Message Length  
— Bad Message Type |
| 2- OPEN Message Error Subcodes | — Unsupported Version Number  
— Bad Peer AS  
— Bad BGP Identifier  
— Unsupported Optional Parameter  
— Authentication Failure  
— Unacceptable Hold Time |
| 3- UPDATE Message Error Subcodes | — Malformed Attribute List  
— Unrecognized Well-Known Attribute  
— Missing Well-Known Attribute  
— Attribute Flags Error  
— Attribute Length Error  
— Invalid ORIGIN Attribute  
— AS Routing Loop  
— Invalid NEXT_HOP Attribute  
— Optional Attribute Error  
— Invalid Network Field  
— Malformed AS_Path |
| 4- Hold Timer Expired | Not applicable |
| 5- Finite State Machine Error | Not applicable |
| 6- Cease | Not applicable |