Module 6 Implementing BGP

Lesson 1 Explaining BGP Concepts and Terminology

BGP – Border Gateway Protocol

Using BGP to Connect to the Internet
- If only one ISP, do not need BGP. If multiple ISPs, use BGP, because it allows manipulation of path attributes so that the optimal path can be selected.
- IGP: a routing protocol that exchanges routing information within an AS. RIP, IGRP, OSPF, IS-IS, and EIGRP are examples of IGPs.
- EGP: a routing protocol that exchanges routing information between different autonomous systems. BGP is an example of EGP.
- BGP is an interdomain routing protocol (IDRP), also known as an EGP.
- EBGP: BGP is running between routers in different autonomous systems
- IBGP: BGP is running between routers in the same autonomous systems

Multihoming
- Connecting to two or more ISPs to increase the following:
  - Reliability: If one ISP or connection fails, there is still Internet access.
  - Performance: Better path selection to common Internet destinations.
- Perform multihoming with BGP, three common ways to do:
  - Each ISP passes only a default route to the AS: The default route is passed to the internal routers.
  - Each ISP passes only a default route and provider-owned specific routes to the AS: These routes may be passed to internal routers, or all internal routers in the transit path can run BGP and pass these routes between them.
  - Each ISP passes all routes to the AS: All internal routers in the transit path run BGP and pass these routes between them.

Example: default Routes from All Providers
Default Routes from All Providers and Partial Table

Example: Full Routes from All Providers

BGP Autonomous Systems
- An AS is a collection of networks under a single technical administration.
- IGPs operate within an AS.
- BGP is used between autonomous systems.
- Exchange of loop-free routing information is guaranteed.
- AS numbers: 16-bit numbers from 1 to 65535. private: 64512 - 65535

BGP Path-Vector Routing
- IGPs announce networks and describe the metric to reach those networks.
- BGP announces paths and the networks that are reachable at the end of the path.
BGP describes the path by using attributes, which are similar to metrics.

- BGP allows administrators to define policies or rules for how data will flow through the autonomous systems.

**BGP Routing Policies**

- BGP can support any policy conforming to the hop-by-hop (AS-by-AS) routing paradigm.

**BGP Characteristics**

- BGP is most appropriate when at least one of the following conditions exists:
  - An AS allows packets to transit through it to reach other autonomous systems (for example, it is a service provider).
  - An AS has multiple connections to other autonomous systems.
  - Routing policy and route selection for traffic entering and leaving the AS must be manipulated.
- BGP is not always appropriate. Do not have to use BGP if you have one of the following conditions:
  - Limited understanding of route filtering and BGP path selection process
  - A single connection to the Internet or another AS
  - Lack of memory or processor power to handle constant updates on BGP routers
- BGP is a path-vector protocol with the following enhancements over distance vector protocols:
  - Reliable updates: BGP runs on top of TCP (port 179)
  - Incremental, triggered updates only
  - Periodic keepalive messages to verify TCP connectivity (every 60 seconds)
  - Rich metrics (called path vectors or attributes)
  - Designed to scale to huge internetworks (for example, the Internet)
- NOTE: BGP use TCP as its transport layer. OSPF, IGRP, EIGRP reside directly above the IP layer and RIP use UDP

**BGP Databases**

- Neighbor table
  - List of BGP neighbors
- BGP table (forwarding database)
  - List of all networks learned from each neighbor
  - Can contain multiple paths to destination networks
  - Contains BGP attributes for each path
- IP routing table
  - List of best paths to destination networks
- AD: EBGP 20; IBGP: 200

**BGP Message Types**

- Open
  - Includes holdtime and BGP router ID
• Keepalive
• Update
  - Information for one path only (could be to multiple networks)
  - Includes path attributes and networks
• Notification
  - When error is detected
  - BGP connection is closed after being sent
Lesson 2 Explaining EBGP and IBGP

Peers = Neighbors

- A “BGP peer,” also known as a “BGP neighbor,” is a specific term that is used for BGP speakers that have established a neighbor relationship.
- Any two routers that have formed a TCP connection to exchange BGP routing information are called BGP peers or BGP neighbors.
- BGP peer must be configured with a BGP `neighbor` command.

External BGP

- When BGP is running between neighbors that belong to different autonomous systems, it is called EBGP.
- EBGP neighbors, by default, need to be directly connected.

Internal BGP

- When BGP is running between neighbors within the same AS, it is called IBGP.
- The neighbors do not have to be directly connected.
IBGP in a Transit AS (ISP)

Redistributing BGP into an IGP (OSPF in this example) is not recommended. Instead, run IBGP on all routers.

IBGP in a NonTransit AS

By default, routes learned via IBGP are never propagated to other IBGP peers, so they need full-mesh IBGP.

Routing Issues if BGP Is Not on in All Routers in Transit Path

Router C will drop the packet to network 10.0.0.0. Router C is not running IBGP; therefore, it has not learned about the route to network 10.0.0.0 from router B. In this example, router B and router E are not redistributing BGP into OSPF.
Lesson 3 Configuring Basic IBGP Operations

BGP Commands

- \textbf{R(config)#router bgp autonomous-system}
  - This command enters router configuration mode only; subcommands must be entered to activate BGP.
  - Only one instance of BGP can be configured on the router at a single time.
  - The autonomous system number identifies the autonomous system to which the router belongs. (the local AS number)
  - The autonomous system number in this command is compared to the autonomous system numbers listed in neighbor statements to determine if the neighbor is an internal or external neighbor.

BGP \textit{neighbor remote-as} Commands

- \textbf{R(config-router)#neighbor \{ip-address | peer-group-name\} remote-as autonomous-system}
  - The \textit{neighbor} command activates a BGP session with this neighbor.
  - The IP address that is specified is the destination address of BGP packets going to this neighbor.
  - This router must have an IP path to reach this neighbor before it can set up a BGP relationship.
  - The \textit{remote-as} option shows what AS this neighbor is in. This AS number is used to determine if the neighbor is internal or external.
  - This command is used for both external and internal neighbors.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{ip-address}</td>
<td>Identifies the peer router</td>
</tr>
<tr>
<td>\textit{peer-group-name}</td>
<td>Identifies the name of a BGP peer group</td>
</tr>
<tr>
<td>\textit{autonomous-system}</td>
<td>Identifies the AS of the peer router</td>
</tr>
</tbody>
</table>

Example: BGP \textit{neighbor} Commands

![Diagram of network topology with router bgp 65102 neighbor 192.168.1.2 remote-as 65101, router bgp 65101 neighbor 192.168.1.1 remote-as 65102, and router bgp 65101 neighbor 10.2.2.2 remote-as 65101]
BGP neighbor shutdown Commands

- **R(config-router)#neighbor {ip-address | peer-group-name} shutdown**
  - Administratively brings down a BGP neighbor
  - Used for maintenance and policy changes to prevent route flapping
- **R(config-router)#no neighbor {ip-address | peer-group-name} shutdown**
  - Re-enables a BGP neighbor that has been administratively shut down

BGP Issues with Source IP Address

- When creating a BGP packet, the neighbor statement defines the destination IP address and the outbound interface defines the source IP address.
- When a BGP packet is received for a new BGP session, the source address of the packet is compared to the list of neighbor statements:
  - If a match is found, a relationship is established.
  - If no match is found, the packet is ignored.
- Make sure that the source IP address matches the address that the other router has in its neighbor statement.

Example: IBGP Peering Issue

To establish the IBGP session between router A and router D, which neighbor addresses should be used?

<table>
<thead>
<tr>
<th>What IP address should router A use for peering with router D?</th>
<th>What IP address should router D use for peering with router A?</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.4.4.4</td>
<td>10.1.1.1</td>
</tr>
<tr>
<td>10.2.2.4</td>
<td>10.3.3.1</td>
</tr>
<tr>
<td>4.4.4.4</td>
<td>1.1.1.1</td>
</tr>
</tbody>
</table>

- If router D uses `neighbor 10.3.3.1 remote-as 65102`, but router A is sending the BGP packets to router D via router B, the source IP address will be 10.1.1.1. Therefore, the IBGP session between router A and router D cannot be established.
- A solution to this problem is to establish the IBGP session using a loopback interface when there are multiple paths between the IBGP neighbors.
BGP neighbor update-source Command

- The `neighbor update-source` command allows the BGP process to use the IP address of a specified interface as the source IP address of all BGP updates to that neighbor.
- A loopback interface is usually used, because it will be available as long as the router is operational.
- The IP address used in the `neighbor` command on the other router will be the destination IP address of all BGP updates and should be the loopback interface of this router.
- The `neighbor update-source` command is normally used only with IBGP neighbors.
- The address of an EBGP neighbor must be directly connected by default; the loopback of an EBGP neighbor is not directly connected.

Example: BGP Using Loopback Address

```
```

- The `neighbor` command should be used on both routers.

BGP neighbor ebgp-multihop Command

- The `neighbor ebgp-multihop` command allows the router to accept and attempt BGP connections to external peers residing on networks that are not directly connected.
- This command increases the default of one hop for EBGP peers by changing the default Time to Live (TTL) value of 1. (default: TTL is set to 255)
- It allows routes to the EBGP loopback address (which will have a hop count greater than 1).
BGP is not designed to perform load balancing; paths are chosen because of policy, not based on bandwidth. BGP will choose only a single best path. Using the loopback addresses and the `neighbor ebgp-multihop` command as shown in this example allows load balancing, as well as redundancy, across the two paths between the autonomous systems.

**Next-Hop Behavior**
- BGP is an AS-by-AS routing protocol, not a router-by-router routing protocol.
- In BGP, the next hop does not mean the next router; it means the IP address to reach the next AS.
- For EBGP, the default next hop is the IP address of the neighbor router that sent the update.
- For IBGP, the BGP protocol states that the next hop advertised by EBGP should be carried into IBGP.

**Example: Next-Hop Behavior**
- Router A advertises network 172.16.0.0 to router B in EBGP, with a next hop of 10.10.10.3.
- Router B advertises 172.16.0.0 in IBGP to router C, keeping 10.10.10.3 as the next-hop address.
BGP neighbor next-hop-self Command

- **R(config-router)#neighbor {ip-address | peer-group-name} next-hop-self**
  - Forces all updates for this neighbor to be advertised with this router as the next hop.
  - The IP address used for the **next-hop-self** option will be the same as the source IP address of the BGP packet.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip-address</td>
<td>Identifies the peer router to which advertisements are sent with this router identified as the next hop</td>
</tr>
<tr>
<td>peer-group-name</td>
<td>Identifies the name of a BGP peer group</td>
</tr>
</tbody>
</table>

Example: next-hop-self Configuration

```
router bgp 65101
neighbor 172.16.1.1 remote-as 65100
neighbor 3.3.3.3 remote-as 65101
neighbor 3.3.3.3 update-source Loopback0
neighbor 3.3.3.3 next-hop-self

! router eigrp 1
network 10.0.0.0
network 2.0.0.0
```

Example: Next Hop on a Multiaccess Network
- Router B advertises network 172.30.0.0 to router A in EBGP with a next hop of 10.10.10.2, not 10.10.10.1. This avoids an unnecessary hop.
- BGP is being efficient by informing AS 64520 of the best entry point into AS 65000 for network 172.30.0.0.
- Router B in AS 65000 also advertises to AS 64520 that the best entry point for each network in AS 64600 is the next hop of router C because that is the best path to move through AS 65000 to AS 64600.

Using a Peer Group
- `R(config-router)#neighbor peer-group-name peer-group`
  - This command creates a peer group.
- `R(config-router)#neighbor ip-address peer-group peer-group-name`
  - This command defines a template with parameters set for a group of neighbors instead of individually.
  - This command is useful when many neighbors have the same outbound policies.
  - Members can have a different inbound policy.
  - Updates are generated once per peer group.
  - Configuration is simplified.
  - Must enter the `neighbor peer-group-name peer-group` command before the router will accept this command.

Example: Using a Peer Group

```
router bgp 65100
neighbor 192.168.24.1 remote-as 65100
neighbor 192.168.24.1 update-source Loopback 0
neighbor 192.168.24.1 next-hop-self
neighbor 192.168.24.1 distribute-list 20 out
neighbor 192.168.25.1 remote-as 65100
neighbor 192.168.25.1 update-source Loopback 0
neighbor 192.168.25.1 next-hop-self
neighbor 192.168.25.1 distribute-list 20 out
neighbor 192.168.26.1 remote-as 65100
neighbor 192.168.26.1 update-source Loopback 0
neighbor 192.168.26.1 next-hop-self
neighbor 192.168.26.1 distribute-list 20 out
```

```
router bgp 65100
neighbor internal peer-group
neighbor internal remote-as 65100
neighbor internal update-source Loopback 0
neighbor internal next-hop-self
neighbor internal distribute-list 20 out
neighbor 192.168.24.1 peer-group internal
neighbor 192.168.25.1 peer-group internal
neighbor 192.168.26.1 peer-group internal
```
BGP network Command

- R(config-router)#network network-number [mask network-mask] [route-map map-tag]
  - This command tells BGP what network to advertise.
  - The command does not activate the protocol on an interface.
  - Without a mask option, the command advertises classful networks. If a subnet of the classful network exists in a routing table, the classful address is announced.
  - With the mask option, BGP looks for an exact match in the local routing table before announcing the route.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network-number</td>
<td>Identifies an IP network to be advertised by BGP.</td>
</tr>
<tr>
<td>network-mask</td>
<td>(Optional) Identifies the subnet mask to be advertised by BGP.</td>
</tr>
<tr>
<td>map-tag</td>
<td>(Optional) Identifier of a configured route map. The route map is examined to filter the networks to be advertised. If not specified, all networks are advertised. If the route-map keyword is specified, but no route map tags is listed, no networks will be advertised.</td>
</tr>
</tbody>
</table>

- The neighbor command tells BGP where to advertise, and the network command tells BGP what to advertise.

Example: BGP network Command

- R(config-router)#network 192.168.1.1 mask 255.255.255.0
  - The router looks for exactly 192.168.1.1/24 in the routing table, but cannot find it, so it will not announce anything.
- R(config-router)#network 192.168.0.0 mask 255.255.0.0
  - The router looks for exactly 192.168.0.0/16 in the routing table.
  - If the exact route is not in the table, you can add a static route to null0 so that the route can be announced.

BGP Synchronization

- Synchronization rule: Do not use or advertise to an external neighbor a route learned by IBGP until a matching route has been learned from an IGP. In other words, BGP and the IGP must be synchronized before the networks learned from an IBGP neighbor can be used.
  - Ensures consistency of information throughout the AS
  - Safe to have it off only if all routers in the transit path in the AS are running full-mesh IBGP; off by default in Cisco IOS software release 12.2(8)T and later
- R(config-router)#no synchronization
  - Disables BGP synchronization so that a router will advertise routes in BGP without learning them in an IGP
- R(config-router)#synchronization
  - Enables BGP synchronization so that a router will not advertise routes in BGP until it learns them in an IGP
- BGP synchronization is unnecessary in some situations. It is safe to have BGP
synchronization off only if all routers in the transit path in the AS are running full-mesh IBGP.

- Use synchronization if there are routers in the BGP transit path in the AS that are not running BGP (therefore, the routers do not have full-mesh IBGP within the AS).
- In the past, the best practice was to redistribute BGP into the IGP running in an AS, so that IBGP was not needed in every router in the transit path. In this case, synchronization was needed to make sure that packets did not get lost, so synchronization was on by default. As the Internet grew, the number of routes in the BGP table became too much for the IGPs to handle, so the best practice changed to not redistributing BGP into the IGP, but instead using IBGP on all routers in the transit path. In this case, synchronization is not needed; it is now off by default.

**Example: BGP Synchronization**

- If synchronization is on, then:
  - Routers A, C, and D would not use or advertise the route to 172.16.0.0 until they receive the matching route via an IGP.
  - Router E would not hear about 172.16.0.0.
- If synchronization is off (the default), then:
  - Routers A, C, and D would use and advertise the route that they receive via IBGP; router E would hear about 172.16.0.0.
  - If router E sends traffic for 172.16.0.0, routers A, C, and D would route the packets correctly to router B.
- In modern autonomous systems, because the size of the Internet routing table is large, redistributing from BGP into an IGP is not scalable; therefore, most modern autonomous systems run full-mesh IBGP and do not require synchronization. Advanced BGP configuration methods, for example, using route reflectors and confederations, reduce the full-mesh requirements.
Example: BGP Configuration

1. RouterB(config)# router bgp 65000
2. RouterB(config-router)# neighbor 10.1.1.2 remote-as 64520
3. RouterB(config-router)# neighbor 192.168.2.2 remote-as 65000
4. RouterB(config-router)# neighbor 192.168.2.2 update-source Loopback 0
5. RouterB(config-router)# neighbor 192.168.2.2 next-hop-self
6. RouterB(config-router)# network 172.16.10.0 mask 255.255.255.0
7. RouterB(config-router)# network 192.168.1.0
8. RouterB(config-router)# network 192.168.3.0
9. RouterB(config-router)# no synchronization
BGP States
When establishing a BGP session, BGP goes through the following states:
1. **Idle**: Router is searching routing table to see whether a route exists to reach the neighbor.
2. **Connect**: Router found a route to the neighbor and has completed the three-way TCP handshake.
3. **Open sent**: Open message sent, with the parameters for the BGP session.
4. **Open confirm**: Router received agreement on the parameters for establishing session.
   - Alternatively, router goes into **active** state if no response to open message
5. **Established**: Peering is established; routing begins.

BGP Established and Idle States
- **Idle**: The router in this state cannot find the address of the neighbor in the routing table. Check for an IGP problem. Is the neighbor announcing the route?
  - The router is idle because of one of the following scenarios:
    - It is waiting for a static route to that IP address or network to be configured.
    - It is waiting for the local routing protocol (IGP) to learn about this network through an advertisement from another router.
  - Check these two conditions first to correct this problem:
    - Ensure that the neighbor announces the route in its local routing protocol (IGP).
    - Verify that you have not entered an incorrect IP address in the neighbor statement.
- **Established**: The established state is the proper state for BGP operations. In the output of the `show ip bgp summary` command, if the state column has a number, then the route is in the established state. The number is how many routes have been learned from this neighbor.

Example: **show ip bgp neighbors Command**

<table>
<thead>
<tr>
<th>RouterA#sh ip bgp neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP neighbor is 172.31.1.3, remote AS 64998, external link</td>
</tr>
<tr>
<td>BGP version 4, remote router ID 172.31.2.3</td>
</tr>
<tr>
<td>BGP state = Established, up for 00:19:10</td>
</tr>
<tr>
<td>Last read 00:00:10, last write 00:00:10, hold time is 180, keepalive interval is 60 seconds</td>
</tr>
<tr>
<td>Neighbor capabilities:</td>
</tr>
<tr>
<td>Route refresh: advertised and received(old &amp; new)</td>
</tr>
<tr>
<td>Address family IPv4 Unicast: advertised and received</td>
</tr>
<tr>
<td>Message statistics:</td>
</tr>
<tr>
<td>InQ depth is 0</td>
</tr>
<tr>
<td>OutQ depth is 0</td>
</tr>
<tr>
<td>Sent</td>
</tr>
<tr>
<td>Opens:</td>
</tr>
<tr>
<td>Notifications:</td>
</tr>
<tr>
<td>Updates:</td>
</tr>
<tr>
<td>&lt;output omitted&gt;</td>
</tr>
</tbody>
</table>
BGP Active State Troubleshooting

- **Active**: The router has sent an open packet and is waiting for a response. The state may cycle between active and idle. The neighbor may not know how to get back to this router because of the following reasons:
  - Neighbor does not have a route to the source IP address of the BGP open packet generated by this router.
  - Neighbor is peering with the wrong address.
  - Neighbor does not have a neighbor statement for this router.
  - AS number is misconfiguration. – the most common problem

**Example: BGP Active State Troubleshooting**

- At the router with the wrong remote AS number:
  
  ```
  %BGP-3-NOTIFICATION: sent to neighbor 172.31.1.3 2/2 (peer in wrong AS) 2 bytes FDE6 FFF FFF FFF FFF FFF FFF FFF 002D 0104 FDE6 00B4 AC1F 0203 1002 0601 0400 0100 0102 0280 0002 0202 00
  ```

- At the remote router:
  
  ```
  %BGP-3-NOTIFICATION: received from neighbor 172.31.1.1 2/2 (peer in wrong AS) 2 bytes FDE6
  ```

**Example: BGP Peering**

```
RouterA# show ip bgp summary
BGP router identifier 10.1.1.1, local AS number 65001
BGP table version is 124, main routing table version 124
9 network entries using 1053 bytes of memory
22 path entries using 1144 bytes of memory
12/5 BGP path/bestpath attribute entries using 1488 bytes of memory
6 BGP AS-PATH entries using 144 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3829 total bytes of memory
BGP activity 58/49 prefixes, 72/50 paths, scan interval 60 secs

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>V</th>
<th>AS</th>
<th>MsgRcvd</th>
<th>MsgSent</th>
<th>TblVer</th>
<th>InQ</th>
<th>OutQ</th>
<th>Up/Down</th>
<th>State/PfxRcd</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.0.2</td>
<td>4</td>
<td>65001</td>
<td>11</td>
<td>11</td>
<td>124</td>
<td>0</td>
<td>0</td>
<td>00:02:28</td>
<td>8</td>
</tr>
<tr>
<td>172.31.1.3</td>
<td>4</td>
<td>64998</td>
<td>21</td>
<td>18</td>
<td>124</td>
<td>0</td>
<td>0</td>
<td>00:01:13</td>
<td>6</td>
</tr>
<tr>
<td>172.31.11.4</td>
<td>4</td>
<td>64999</td>
<td>11</td>
<td>10</td>
<td>124</td>
<td>0</td>
<td>0</td>
<td>00:01:11</td>
<td>6</td>
</tr>
</tbody>
</table>
```

- The `show ip bgp summary` command is one way to verify the neighbor relationship.
BGP Neighbor Authentication

- **R(config-router)#neighbor** { *ip-address* | *peer-group-name* } **password** *string*
  - BGP authentication uses MD5.
  - Configure a key (password); router generates a message digest, or hash, of the key and the message.
  - Message digest is sent; key is not sent.
  - Router generates and checks the MD5 digest of every segment sent on the TCP connection. Router authenticates the source of each routing update packet that it receives.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>ip-address</em></td>
<td>IP address of the BGP-speaking neighbor.</td>
</tr>
<tr>
<td><em>peer-group-name</em></td>
<td>Name of a BGP peer group.</td>
</tr>
<tr>
<td><em>string</em></td>
<td>Case-sensitive password of up to 25 characters. The first character cannot be a number. The string can contain any alphanumeric characters, including spaces. You cannot specify a password in the format number-space-anything. The space after the number can cause authentication to fail.</td>
</tr>
</tbody>
</table>

- If a router has a password configured for a neighbor, but the neighbor router does not:
  %TCP-6-BADAUTH: No MD5 digest from 10.1.0.2(179) to 10.1.0.1(20236)
- If the two routers have different passwords configured:
  %TCP-6-BADAUTH: Invalid MD5 digest from 10.1.0.1(12293) to 10.1.0.2(179)

Example: BGP Neighbor Authentication
Example: `show ip bgp` Command

```
RouterA# show ip bgp
BGP table version is 14, local router ID is 172.31.11.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

<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>*&gt; 10.1.0.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>* i</td>
<td>10.1.0.2</td>
<td>0</td>
<td>100</td>
<td>0 i</td>
<td></td>
</tr>
<tr>
<td>*&gt; 10.1.1.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>*&gt; i10.1.2.0/24</td>
<td>10.1.0.2</td>
<td>0</td>
<td>100</td>
<td>0 i</td>
<td></td>
</tr>
<tr>
<td>*&gt; 10.97.97.0/24</td>
<td>172.31.1.3</td>
<td>0</td>
<td>64998</td>
<td>64997</td>
<td>i</td>
</tr>
<tr>
<td>*</td>
<td>172.31.11.4</td>
<td>0</td>
<td>64999</td>
<td>64997</td>
<td>i</td>
</tr>
<tr>
<td>* i</td>
<td>172.31.11.4</td>
<td>0</td>
<td>64999</td>
<td>64997</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 10.254.0.0/24</td>
<td>172.31.1.3</td>
<td>0</td>
<td>64998</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>172.31.11.4</td>
<td>0</td>
<td>64999</td>
<td>64998</td>
<td>i</td>
</tr>
<tr>
<td>r</td>
<td>172.31.11.4</td>
<td>0</td>
<td>64999</td>
<td>64998</td>
<td>i</td>
</tr>
<tr>
<td>r i</td>
<td>172.31.1.3</td>
<td>0</td>
<td>64998</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>*&gt; 172.31.2.0/24</td>
<td>172.31.1.3</td>
<td>0</td>
<td>64998</td>
<td>i</td>
<td></td>
</tr>
</tbody>
</table>
```

Example: `show ip bgp rib-failure` Command

```
RouterA# show ip bgp rib-failure

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>RIB-failure</th>
<th>RIB-NH Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.31.1.0/24</td>
<td>172.31.1.3</td>
<td>Higher admin</td>
<td>n/a</td>
</tr>
<tr>
<td>172.31.11.0/24</td>
<td>172.31.11.4</td>
<td>Higher admin</td>
<td>n/a</td>
</tr>
</tbody>
</table>
```

- Displays networks that are not installed in the RIB and the reason that they were not installed.

**Clearing the BGP Session**

- When policies such as access lists or attributes are changed, the change takes effect immediately, and the next time that a prefix or path is advertised or received, the new policy is used. It can take a long time for the policy to be applied to all networks.
- You must trigger an update to ensure that the policy is immediately applied to all affected prefixes and paths.
- Ways to trigger an update:
  - Hard reset
  - Soft reset
  - Route refresh
Hard Reset of BGP Sessions

- **R#show ip bgp ***
  - Resets all BGP connections with this router.
  - Entire BGP forwarding table is discarded.
  - BGP session makes the transition from established to idle; everything must be relearned.

- **R#clear ip bgp [neighbor-address]**
  - Resets only a single neighbor.
  - BGP session makes the transition from established to idle; everything from this neighbor must be relearned.
  - Less severe than clear ip bgp *.

Soft Reset Outbound

- **R#clear ip bgp {*|neighbor-address} [soft out]**
  - Routes learned from this neighbor are not lost.
  - This router resends all BGP information to the neighbor without resetting the connection.
  - The connection remains established.
  - This option is highly recommended when you are changing outbound policy.
  - The **soft out** option does not help if you are changing inbound policy.

Inbound Soft Reset

- **R(config-router)#neighbor [ip-address] soft-reconfiguration inbound**
  - This router stores all updates from this neighbor in case the inbound policy is changed.
  - The command is memory-intensive.

- **R#clear ip bgp {*|neighbor-address} soft in**
  - Uses the stored information to generate new inbound updates

Route Refresh: Dynamic Inbound Soft Reset

- **R#clear ip bgp {*|neighbor-address} [soft in | in]**
  - Routes advertised to this neighbor are not withdrawn.
  - Does not store update information locally.
  - The connection remains established.
  - Introduced in Cisco IOS software release 12.0(2)S and 12.0(6)T.

NOTE: The **clear ip bgp soft** command performs a soft reconfiguration of both inbound and outbound updates.
debug ip bgp updates Command

RouterA# debug ip bgp updates
Mobile router debugging is on for address family: IPv4 Unicast
RouterA# clear ip bgp 10.1.0.2
<output omitted>
*Feb 24 11:06:41.309: %BGP-5-ADJCHANGE: neighbor 10.1.0.2 Up
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (format) 10.1.1.0/24, next 10.1.0.1, metric 0, path Local
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (prepend, chgflags: 0x0) 10.1.0.0/24, next 10.1.0.1, metric 0, path Local
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (format) 10.1.0.0/24, next 10.1.0.1, metric 0, path Local
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 NEXT_HOP part 1 net 10.97.97.0/24, next 172.31.11.4
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (format) 10.97.97.0/24, next 172.31.11.4, metric 0, path 64999 64997
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 NEXT_HOP part 1 net 172.31.22.0/24, next 172.31.11.4
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (format) 172.31.22.0/24, next 172.31.11.4, metric 0, path 64999
<output omitted>
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd UPDATE w/ attr: nexthop 10.1.0.2, origin i, localpref 100, metric 0
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd 10.1.2.0/24
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd 10.1.0.0/24
Lesson 4 Selecting a BGP Path

BGP Path Attributes

- BGP metrics are called path attributes.
- Characteristics of path attributes include:
  - Well-known versus optional
  - Mandatory versus discretionary
  - Transitive versus nontransitive
  - Partial
- Path attributes:
  - Well-known mandatory
  - Well-known discretionary
  - Optional transitive
  - Optional nontransitive
  - Only optional transitive attributes can be marked as partial.

Well-Known Attributes

- Well-known attributes
  - Must be recognized by all compliant BGP implementations
  - Are propagated to other neighbors
- Well-known mandatory attributes
  - Must be present in all update messages
- Well-known discretionary attributes
  - May be present in update messages

Optional Attributes

- Optional attributes
  - They are recognized by some implementations (could be private); but expected not to be recognized by all BGP routers.
  - Recognized optional attributes are propagated to other neighbors based on their meaning.
- Optional transitive attributes
  - If not recognized, marked as partial and propagated to other neighbors
- Optional nontransitive attributes
  - Discarded if not recognized

BGP Attributes

- Well-known mandatory attribute
  - AS path
  - Next-hop
  - Origin
- Well-known discretionary attribute
  - Local preference
  - Atomic aggregate
- Optional transitive attribute
  - Aggregator
- Optional nontransitive attribute
  - Multi-exit discriminator (MED)

**AS Path Attribute**
- A list of autonomous systems that a route has traversed:
  - For example, on router B, the path to 192.168.1.0 is the AS sequence (65500, 64520).
  - A to B: (65500, 65000)
  - C to A: (64520)
  - C to B: (65000)
- The AS path attribute is well-known, mandatory.

**Next-Hop Attribute**
- The IP address of the next AS to reach a given network:
  - Router A advertises network 172.16.0.0 to router B in EBGP, with a next hop of 10.10.10.3.
  - Router B advertises 172.16.0.0 in IBGP to router C, keeping 10.10.10.3 as the next-hop address.
- The next-hop attribute is well-known, mandatory.

**Origin Attribute**
- IGP (i): `network` command
- EGP (e): Redistributed from EGP
- Incomplete (?): Redistributed from IGP or static
- The origin attribute informs all autonomous systems in the internetwork how the prefixes were introduced into BGP.
- The origin attribute is well-known, mandatory.
Example: Origin Attribute

RouterA# show ip bgp
BGP table version is 14, local router ID is 172.31.11.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

<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>*&gt; 10.1.0.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td></td>
<td>i</td>
</tr>
<tr>
<td>* i</td>
<td>10.1.0.2</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 10.1.1.0/24</td>
<td>0.0.0.0</td>
<td>0</td>
<td>32768</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>* i</td>
<td>10.1.0.2</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>i</td>
</tr>
<tr>
<td>*&gt; 10.97.97.0/24</td>
<td>172.31.1.3</td>
<td>0</td>
<td>64998</td>
<td>64997</td>
<td>i</td>
</tr>
<tr>
<td>*</td>
<td>172.31.11.4</td>
<td>0</td>
<td>64999</td>
<td>64997</td>
<td>i</td>
</tr>
<tr>
<td>* i</td>
<td>172.31.11.4</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64999</td>
</tr>
<tr>
<td>*&gt; 10.254.0.0/24</td>
<td>172.31.1.3</td>
<td>0</td>
<td>64998</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>172.31.11.4</td>
<td>0</td>
<td>64999</td>
<td>64998</td>
<td>i</td>
</tr>
<tr>
<td>* i</td>
<td>172.31.1.3</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64998</td>
</tr>
<tr>
<td>r&gt; 172.31.1.0/24</td>
<td>172.31.1.3</td>
<td>0</td>
<td>64998</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>r</td>
<td>172.31.11.4</td>
<td>0</td>
<td>64999</td>
<td>64998</td>
<td>i</td>
</tr>
<tr>
<td>r i</td>
<td>172.31.1.3</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>64998</td>
</tr>
<tr>
<td>*&gt; 172.31.2.0/24</td>
<td>172.31.1.3</td>
<td>0</td>
<td>64998</td>
<td>i</td>
<td></td>
</tr>
</tbody>
</table>

Local Preference Attribute

- Paths with highest local preference value are preferred:
  - Local preference is used to advertise to IBGP neighbors about how to leave their AS.
  - The local preference is sent to IBGP neighbors only (that is, within the AS only).
  - The local preference attribute is well-known and discretionary.
  - Default value is 100.
**MED Attribute**
- The paths with the lowest MED (also called the metric) value are the most desirable:
  - MED is used to advertise to EBGP neighbors how to exit their AS to reach networks owned by this AS.
  - By default, a router compares the MED attribute only for paths from neighbors in the same AS.
- The MED attribute is optional and nontransitive.
- MED influences inbound traffic to an AS, and local preference influences outbound traffic from an AS.

*NOTE:* The MED attribute means that BGP is the only protocol that can affect how routes are sent into an AS.

**Weight Attribute (Cisco Only)**
- Paths with the highest weight value are preferred
- Weight not sent to any BGP neighbors; local to this router only
- The weight can have a value from 0 to 65535. Paths that the router originates have a weight of 32768 by default, and other paths have a weight of 0 by default.

**BGP Path Selection**
- The BGP forwarding table usually has multiple paths from which to choose for each network.
- BGP is not designed to perform load balancing:
  - Paths are chosen because of policy.
  - Paths are not chosen based on bandwidth.
- The BGP selection process eliminates any multiple paths through attrition until a single best path is left.
- That best path is submitted to the routing table manager process and evaluated against the methods of other routing protocols for reaching that network (using administrative distance).
The route from the source with the lowest administrative distance is installed in the routing table.

Route Selection Decision Process

- Consider only (synchronized) routes with no AS loops and a valid next hop, and then:
  1. Prefer highest weight (local to router).
  2. Prefer highest local preference (global within AS).
  3. Prefer route originated by the local router (next hop = 0.0.0.0).
  4. Prefer shortest AS path.
  5. Prefer lowest origin code (IGP < EGP < incomplete).
  6. Prefer lowest MED (exchanged between autonomous systems).
  7. Prefer EBGP path over IBGP path.
  8. Prefer the path through the closest IGP neighbor.
 10. Prefer the path with the lowest neighbor BGP router ID.
11. Prefer the path with the lowest neighbor IP address.

- The MED comparison is made only if the neighboring AS is the same for all routes considered, unless the bgp always-compare-med command is enabled.
Lesson 5 Using Route Maps to Manipulate Basic BGP Paths

BGP Is Designed to Implement Policy Routing

- BGP is designed for manipulating routing paths.

Changing BGP Local Preference For All Routes

- Local preference is used in these ways:
  - Within an AS between IBGP speakers
  - To determine the best path to exit the AS to reach an outside network
  - Set to 100 by default; higher values preferred

- `R(config-router)#bgp default local-preference value`
  - This command changes the default local preference value.
  - All routes advertised to an IBGP neighbor have the local preference set to the value specified.
  - `value`: Local preference value from 0 to 4294967295. A higher value is more preferred.

- If an EBGP neighbor receives a local preference value, EBGP neighbor ignores it.

Local Preference Case Study

- The best path for router C to 65003:
  - Steps 1 and 2 look at weight and local preference and use the default settings of weight equaling 0 and local preference equaling 100 for all routes that are learned from the IBGP neighbors of A and B.
  - Step 3 does not help decide the best path because the three AS routes are not owned or originated by AS 65001.
- Step 4 prefers the shortest AS path; the options are two autonomous systems (65002, 65003) through router A or three autonomous systems through IBGP neighbor router B (65005, 65004, 65003). Thus, the shortest AS path from router C to AS 65003 is through router A.
  - The best path for router C to 65005:
    - The best path from router C to networks in AS 65005 is also selected by Step 4, the shortest AS path. The shortest path from router C to AS 65005 is through router B because it consists of one AS (65005) compared to four autonomous systems (65002, 65003, 65004, 65005) through router A.
  - The best path for router C to 65004:
    - The best path from router C to networks in AS 65004 is also selected by Step 4, the shortest AS path. The shortest path from router C to AS 65004 is through router B because it consists of two autonomous systems (65005, 65004) compared to three autonomous systems (65002, 65003, 65004) through router A.

**Router C BGP Table with Default Settings**

```
RouterC# show ip bgp
BGP table version is 7, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale
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>* i172.16.0.0</td>
<td>172.20.50.1</td>
<td>100</td>
<td>0</td>
<td>65005 65004 65003 i</td>
<td></td>
</tr>
<tr>
<td>*&gt;i</td>
<td>192.168.28.1</td>
<td>100</td>
<td>0</td>
<td>65002 65003 i</td>
<td></td>
</tr>
<tr>
<td>*&gt;i172.24.0.0</td>
<td>172.20.50.1</td>
<td>100</td>
<td>0</td>
<td>65005 i</td>
<td></td>
</tr>
<tr>
<td>* i</td>
<td>192.168.28.1</td>
<td>100</td>
<td>0</td>
<td>65002 65003 65004 65005 i</td>
<td></td>
</tr>
<tr>
<td>*&gt;i172.30.0.0</td>
<td>172.20.50.1</td>
<td>100</td>
<td>0</td>
<td>65005 65004 i</td>
<td></td>
</tr>
<tr>
<td>* i</td>
<td>192.168.28.1</td>
<td>100</td>
<td>0</td>
<td>65002 65003 65004i</td>
<td></td>
</tr>
</tbody>
</table>
```

- By default, BGP selects the shortest AS path as the best (>) path.
- In AS 65001, the percentage of traffic going to 172.24.0.0 is 30%, 172.30.0.0 is 20%, and 172.16.0.0 is 10%.
- 50% of all traffic will go to the next hop of 172.20.50.1 (AS 65005), and 10% of all traffic will go to the next hop of 192.168.28.1 (AS 65002).
- Make traffic to 172.30.0.0 select the next hop of 192.168.28.1 to achieve load sharing where both external links get approximately 30% of the load.
This figure demonstrates the use of a route map on router A to alter the network 172.30.0.0 BGP update from router X (192.168.28.1) to have a high local preference value of 400 so that it will be more preferred.

Router C BGP Table with Local Preference Learned

Best (> ) paths for networks 172.16.0.0/16 and 172.24.0.0/16 have not changed.
Best (> ) path for network 172.30.0.0 has changed to a new next hop of 192.168.28.1 because the next hop of 192.168.28.1 has a higher local preference, 400.
In AS 65001, the percentage of traffic going to 172.24.0.0 is 30%, 172.30.0.0 is 20%, and 172.16.0.0 is 10%.
30% of all traffic will go to the next hop of 172.20.50.1 (AS 65005), and 30% of all traffic will go to the next hop of 192.168.28.1 (AS 65002).
Changing BGP MED for All Routes
- MED is used when multiple paths exist between two autonomous systems.
- A lower MED value is preferred.
- The default setting for Cisco is MED = 0.
- The metric is an optional, nontransitive attribute.
- Usually, MED is shared only between two autonomous systems that have multiple EBGP connections with each other.
- `R(config-router)#default-metric number`
  - `number`: (Optional) The value of the metric, which for BGP is the MED
- MED is considered the metric of BGP.
- All routes that are advertised to an EBGP neighbor are set to the value specified using this command.

BGP Using Route Maps and the MED

Route Map for Router A

```
Router A’s Configuration:
router bgp 65001
neighbor 2.2.2.2 remote-as 65001
neighbor 3.3.3.3 remote-as 65001
neighbor 2.2.2.2 update-source loopback0
neighbor 3.3.3.3 update-source loopback0
neighbor 192.168.28.1 remote-as 65004
neighbor 192.168.28.1 route-map med_65004 out !
access-list 66 permit 192.168.25.0.0 0.0.0.255
access-list 66 permit 192.168.26.0.0 0.0.0.255
!
route-map med_65004 permit 10
match ip address 66
set metric 100
!
route-map med_65004 permit 100
set metric 200
```
Route Map for Router B

Router B’s Configuration:
router bgp 65001
neighbor 1.1.1.1 remote-as 65001
neighbor 3.3.3.3 remote-as 65001
neighbor 1.1.1.1 update-source loopback0
neighbor 3.3.3.3 update-source loopback0
neighbor 172.20.50.1 remote-as 65004
neighbor 172.20.50.1 route-map med_65004 out
  !
  access-list 66 permit 192.168.24.0.0 0.0.0.255
  !
  route-map med_65004 permit 10
  match ip address 66
  set metric 100
  !
  route-map med_65004 permit 100
  set metric 200

Router B’s Configuration:

MED Learned by Router Z

RouterZ# show ip bgp
BGP table version is 7, local router ID is 122.30.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
  internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

<table>
<thead>
<tr>
<th>Network</th>
<th>Next Hop</th>
<th>Metric</th>
<th>LocPrf</th>
<th>We ight</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>* &gt;/i 192.168.24.0</td>
<td>172.20.50.2</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>65001 i</td>
</tr>
<tr>
<td>* i 192.168.25.0</td>
<td>172.20.50.2</td>
<td>200</td>
<td>100</td>
<td>0</td>
<td>65001 i</td>
</tr>
<tr>
<td>* &gt;/i 192.168.26.0</td>
<td>172.20.50.2</td>
<td>200</td>
<td>100</td>
<td>0</td>
<td>65001 i</td>
</tr>
</tbody>
</table>

- Examine the networks that have been learned from AS 65001 on Router Z in AS 65004.
- For all networks: Weight is equal (0); local preference is equal (100); routes are not originated in this AS; AS path is equal (65001); origin code is equal (i).
- 192.168.24.0 has a lower metric (MED) through 172.20.50.2 (100) than 192.168.28.2 (200).
- 192.168.25.0 has a lower metric (MED) through 192.168.28.2 (100) than 172.20.50.2 (200).
- 192.168.26.0 has a lower metric (MED) through 192.168.28.2 (100) than 172.20.50.2 (200).
BGP in an Enterprise

![Diagram of BGP in an Enterprise]