Document Name: BGP

Border Gateway Protocol or BGP is a dynamic path-vector routing protocol defined in RFC 4271. Though BGP is described as a routing protocol is not really one, it is an application as it relies heavily on IGPs like OSPF for transport and recursion. It uses TCP port 179 for transport and is unicast. There are two types of BGP, internal BGP and external BGP. This will change its behavior depending on the type.

- **eBGP**: sessions are established with a BGP router that are in a different AS. eBGP has an Administrative distance of 20.
- **iBGP**: Sessions established with an iBGP router that are in the same AS or that participate in the same BGP confederation. iBGP has an Administrative distance of 200.

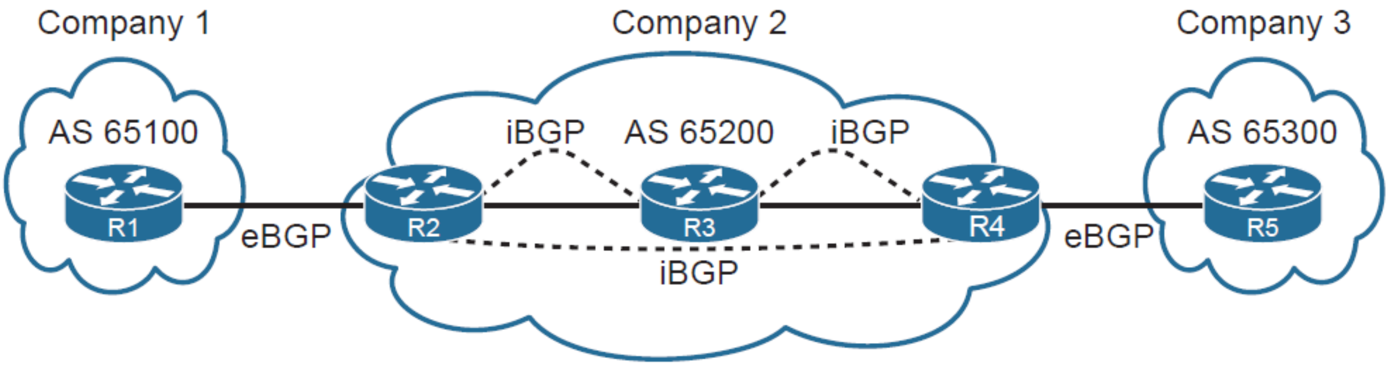
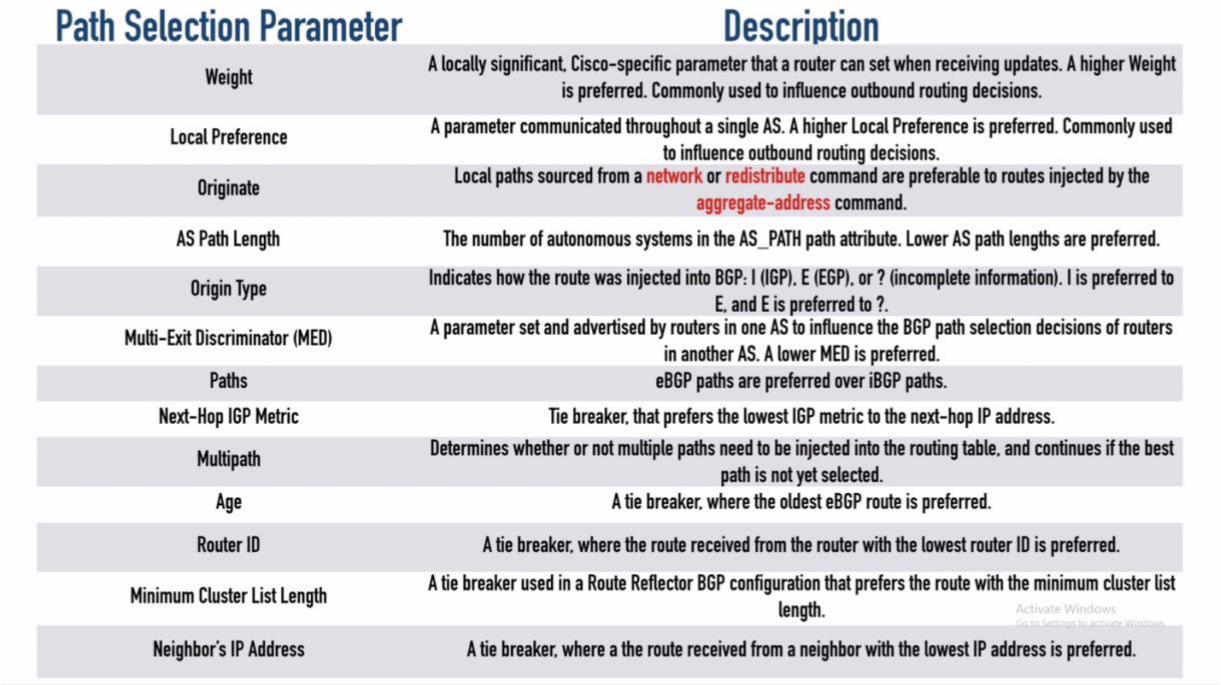


Figure 11-5 *eBGP and iBGP Sessions*

There are no neighbors in BGP there are peers. This is because a neighbor typically means that there is a direct-connection between the two. As with peers there is no need for a direct-connection, as is the case with BGP. BGP exchanges unicast routes but it can also carry lots of other things as well such as MPLS labels. It exchanges Network-Layer Reachability Information or NLRI. BGP updates from external peers can be forwarded to any other peer, but BGP updates from internal peers can ONLY be forwarded to external peers. Certain BGP attributes can only be forwarded to external or internal peers.



Path Selection Parameter	Description
Weight	A locally significant, Cisco-specific parameter that a router can set when receiving updates. A higher Weight is preferred. Commonly used to influence outbound routing decisions.
Local Preference	A parameter communicated throughout a single AS. A higher Local Preference is preferred. Commonly used to influence outbound routing decisions.
Originate	Local paths sourced from a network or redistribute command are preferable to routes injected by the aggregate-address command.
AS Path Length	The number of autonomous systems in the AS_PATH path attribute. Lower AS path lengths are preferred.
Origin Type	Indicates how the route was injected into BGP: I (IGP), E (EGP), or ? (incomplete information). I is preferred to E, and E is preferred to ?.
Multi-Exit Discriminator (MED)	A parameter set and advertised by routers in one AS to influence the BGP path selection decisions of routers in another AS. A lower MED is preferred.
Paths	eBGP paths are preferred over iBGP paths.
Next-Hop IGP Metric	Tie breaker, that prefers the lowest IGP metric to the next-hop IP address.
Multipath	Determines whether or not multiple paths need to be injected into the routing table, and continues if the best path is not yet selected.
Age	A tie breaker, where the oldest eBGP route is preferred.
Router ID	A tie breaker, where the route received from the router with the lowest router ID is preferred.
Minimum Cluster List Length	A tie breaker used in a Route Reflector BGP configuration that prefers the route with the minimum cluster list length.
Neighbor's IP Address	A tie breaker, where a the route received from a neighbor with the lowest IP address is preferred.

Receiving and Viewing Routes

BGP uses three different tables for storing network, prefix, and path attributes.

- **Adj-RIB-In** – Contains NLRIs in their original format (before inbound route policies are processed). The table is purged after all route policies are processed.
- **Loc-RIB** – Contains all the NLRIs that originated locally or were received from other BGP peers. After NLRIs pass the validity and next-hop reachability check, the BGP best-path algorithm selects the best NLRI for the specific prefix. The Loc-RIB table is the table used for presenting routes to the IP Routing table.
- **Adj-RIB-Out** – Contains NLRIs after outbound route policies have been processed.

Configuring BGP

```
router(config)# router bgp [Autonomous System Number]
router(config-router)# neighbor A.B.C.D remote-as [Neighbor's Autonomous System Number]
```

BGP Peering Process.

1. Basic BGP must be configured
2. TCP 3-way handshake must complete
3. BGP peering can start and complete
4. BGP updates are exchanged
5. BGP best path selection process occurs

BGP does not automatically share routes, that has to be configured. To do this you can use the network command or route redistribution. Add the mask if the subnet uses anything else besides the class-full subnet mask.

```
router(config-router)# network A.B.C.D mask A.B.C.D
```

BGP Default Routes

There are two ways to add default routes to BGP. Using a default from an IGP (EIGRP, OSPF, etc) or using the default-originate command. Doing it this way allows you the flexibility of using conditional routes.

This will advertise the default route in your IGP of choice.

```
Router(config-router)# network 0.0.0.0
```

This configuration will allow to use of conditional routes.

```
Router(config-router)# neighbor A.B.C.D default-originate
```

Conditional Routing

Conditional allows you some flexibility when and how you want to advertise certain routes. There are two main conditions that can be applied to a route.

- Advertise this route if this other route is in the routing table
- Advertise this route if this other route is not in the routing table

```
Router(config-router)# neighbor A.B.C.D advertise-map [Name] exist-map [Name]
Router(config-router# neighbor A.B.C.D advertise-map [Name] non-exist-map [Name]
```

eBGP Multi-hop

```
Router(config-router)# neighbor A.B.C.D ebgp-multihop
```

BGP Authentication

This configures a password between two peers.

```
Router(config-router)# neighbor A.B.C.D password [password]
```

Update Source

This configures where BGP packets will be sourced from on a router. This is used if you want to peer from a loop back interface most commonly.

```
Router(config-router)# neighbor A.B.C.D update-source
```

Connected Check

This will set the multi-hop TTL to 1. This allows you to peer to a device that is not directly connected but only one hop away.

```
Router(config-router)# neighbor A.B.C.D disable-connected-check
```

Manual Peering Initiation

This give you control of who starts the BGP peering process.

```
Router(config-router)# neighbor A.B.C.D transport connection-mode [active | passive]
```

TTL Security

When configured all incoming BGP packets must have a TTL that is greater than or equal to ≥ the configured value.

```
Router(config-router)# neighbor A.B.C.D ttl-security hops [1-255]
```

Fall Over

This configuration of fall over works only when an interface changes status (I.e. it goes down in way).

```
Router(config-router)# neighbor A.B.C.D fall-over
```

BFD

Bi-Directional Forwarding Detection. Sub second failover using BFD/UDP "pings" via CEF and UDP (port 3784). The actual configuration for BFD is on the interface. The multiplier specifies the minimum number of consecutive packets that can be missed before the session is considered down.

```
Router(config-if)# bfd interval [50 – 9999mses] min_rx [ 50 - 9999msecs] multiplier [3 - 50]
```

Then the BFD has to be added to the BGP process.

```
Router(config-router)# neighbor A.B.C.D fall-over BFD
```

BFD Show Command.

```
Router# show bfd neighbor detail
```

To configure BFD on a loopback you have to create a BFD map.

```
!
Bfd-template multi-hop [Descriptive name]
 Interval min-tx [50 - 9999] min-rx [] multiplier [3 - 50]
!
Bfd map ipv4 [destination IP address] [source IP address] [Name]
!
!
Neighbor A.B.C.D fall-over multihop
!
```

Address Families

By default, BGP advertises all configured routes to all IPv4 peers. Any other type of routes (IPv6 unicast, IPv4/IPv6 multicast routes) and peer types (IPv6) have to be configured manually. In addition unless configured a certain way BGP will create and maintain separate TCP sessions and routing tables for both IPv4 and IPv6. There is a way to use a single session but you end up either using IPv4 or IPv6 to initiate and maintain the session. All will three configurations are shown below.

Before that the command no bgp default ipv4-unicast prevents IPv4 routes being advertised to IPv6 peers without any whims.

This configuration is using two separate sessions for both IPv4 and IPv6.

```
!
router bgp 1
no bgp default ipv4-unicast
neighbor A.B.C.D remote-as 2
neighbor A:B:C::D remote-as 2
!
address-family ipv4 unicast
neighbor A.B.C.D activate
network D.C.E.E mask 255.255.0.0
!
address-family ipv6 unicast
neighbor A:B:C::D activate
network D:E::/64
!
```

This next example is using a single session with the session being formed via IPv4.

This next example is using a single as well but the session being formed via IPv6.

Peer Groups

Allow you to apply configuration/policy/router filters to multiple BGP neighbors.

First you need to have a prefix list for the routes you want to control see example below:

```
!
ip prefix-list BGP-DEMO
   seq 5 deny 10.0.0.0/8 le 32
   seq 10 deny 172.16.0.0/12 le 32
   seq 15 deny 192.168.0.0/16 le 32
   seq 20 permit 0.0.0.0/0
   seq 25 permit 0.0.0.0/0 ge 8
!
```

```
Router(config)# router bgp 65000
Router(config-router)# neighbor BGP-PG peer-group
Router(config-router)# neighbor BGP-PG prefix-list BGP-DEMO in
Router(config-router)# neighbor 192.0.2.1 peer-group BGP-PG in
```

Route Reflector

When an iBGP route is learned on a device it will not advertise that route to other iBGP peers. Full mesh would fix this it just doesn’t scale in a large environment. A route reflector will advertise learned iBGP routes to its neighbors.

```
Router(config)# router bgp 65000
Router(config-router)# neighbor 172.16.1.1 route-reflector-client
```

Influencing Path Selection

To influence the local preference, you create a route map and apply that to the neighbors whose preference you want to change. Highest local preference wins.

```
Router(config)# route-map LOCALPREF-ISP1
Router(config-route-map)# set local-preference 100

Router(config)# route-map LOCALPREF-ISP2
Router(config-route-map)# set local-preference 200
```

This will change the best route from the .2 neighbor to the .6 neighbor.

```
Router(config)# router bgp 65000
Router(config-router)# neighbor 198.51.100.2 route-map LOCALPREF-ISP1 in
Router(config-router)# neighbor 198.51.100.6 route-map LOCALPREF-ISP2 in
```

This only works on your traffic leaving your network not coming back in. But you can influence that by prepending multiple instances of your AS, it makes it look like more AS hops when it really it isn't.

This can be done as below:

```
Router(config)# route-map ASPATH
Router(config-route-map)# set as-path prepend 65000 65000

Router(config)# router bgp 65000
Router(config-router)# neighbor 198.51.100.2 route-map ASPATH out
```

This will make it appear as though there are two additional AS hops between you and that particular neighbor.

Route Summarization

There are two ways to achieve route summarization in BGP. Statically which is you manually summarizing subnets with the network command. The other is dynamic. This permits the route to automatically summarize what networks that it has configured. See below.

```
router(config-router)# network 172.16.1.0 mask 255.255.255.0
router(config-router)# network 172.16.3.0 mask 255.255.255.0
router(config-router)# network 172.16.3.0 mask 255.255.255.0
```

This will show in the route table as below:

```
    Network          Next Hop            Metric LocPrf Weight Path
*>   172.16.1.0/24    10.23.1.2                          0 65200 65100 i
*>   172.16.2.0/24    10.23.1.2                          0 65200 65100 i
*>   172.16.3.0/24    10.23.1.2                          0 65200 65100 I
```

Each network is advertised into BGP. You enable summarization with the command below.

```
router(config-router-af)# aggregate-address 172.16.0.0 255.255.240.0
```

The routing table will look like this. It will advertise the new summarized route but will leave the individual routes.

```
    Network          Next Hop            Metric LocPrf Weight Path
*>   172.16.0.0/20    10.23.1.2                          0 65200 65100 i
*>   172.16.1.0/24    10.23.1.2                          0 65200 65100 i
*>   172.16.2.0/24    10.23.1.2                          0 65200 65100 i
*>   172.16.3.0/24    10.23.1.2                          0 65200 65100 i
```

To truly summarize the routes you need to use the below command:

```
router(config-router-af)# aggregate-address 172.16.0.0 255.255.240.0 summary-only
```

Now the routing table should look like this:

```
    Network          Next Hop            Metric LocPrf Weight Path
*>   172.16.0.0/20    10.23.1.2                          0 65200 65100 i
```

BGP Communities

BGP communities add the ability to tag routes and for modifying policy both up and downstream. BGP communities are an optional transitive attribute that can traverse from AS to AS. There are two types of communities. Standard and extended. Extended communities are 64 bits in length but are limited in the amount of values that can be used as there are a large amount that are reserved. Standard communities are more common and are 32 bits in length two 16 bit values separated by a colon. It is common to populate the first value with your AS number. It is also important to note that a prefix can belong to multiple communities.

There are some well-known BGP communities. They are listed below.

- **Internet** (0x0) all routes are members of this community.
- **No-export** (0xFFFFFF01) do not advertise to eBGP peers.
- **No-advertise** (0xFFFFFF02) do not advertise to any peer (you may receive this route but not advertise it).
- **Local-as** (0xFFFFFF03) do advertise outside local AS (used with confederations).

Some configuration examples are below.

Transmitting BGP communities.

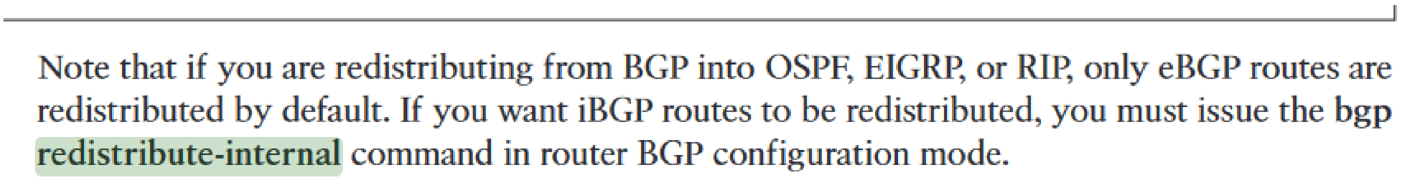
```
!
router bgp 333
neighbor A.B.C.D send-community
neighbor A.B.C.D route-map Community out
!
access-list 1 permit 100.100.0.0 0.0.255.255
!
route-map Community permit 10
match ip address 1
set community 100100
!
router-map Community permit 20
!
```

Matching on BGP Communities.

```
!
router bgp 1112
neighbor 192.168.1.17 route-map INE in
!
ip community-list 1 permit 100100
!
route-map INE permit 10
match community 1
set weight 300
!
router-map INE permit 20
!
```

Redistributing BGP into other protocols

BGP requires you to enable internal redistribution of iBGP into other IGPs.



Note that if you are redistributing from BGP into OSPF, EIGRP, or RIP, only eBGP routes are redistributed by default. If you want iBGP routes to be redistributed, you must issue the **bgp redistribute-internal** command in router BGP configuration mode.

https://community.cisco.com/t5/routing/redistribute-bgp-into-eigrp/td-p/479857

```
!
router bgp 65000
 address-family ipv4 unicast
  bgp redistribute-internal
!
```

NX-OS

```
nexus(config)# feature bgp
nexus(config)# router bgp [AS number}
nexus(config-router)# neighbor [Neighbor IP address]
nexus(config-router)# log-neighbor-changes
nexus(config-router-neighbor)# address-family ipv4 unicast
nexus(config-router-neighbor-afi)# update-source loopback 0
```

Show Commands

```
router# show ip bgp summary
router# show ip bgp neighbor
router# show ip bgp
```