



Configuring Basic BGP



BSCI Module 6

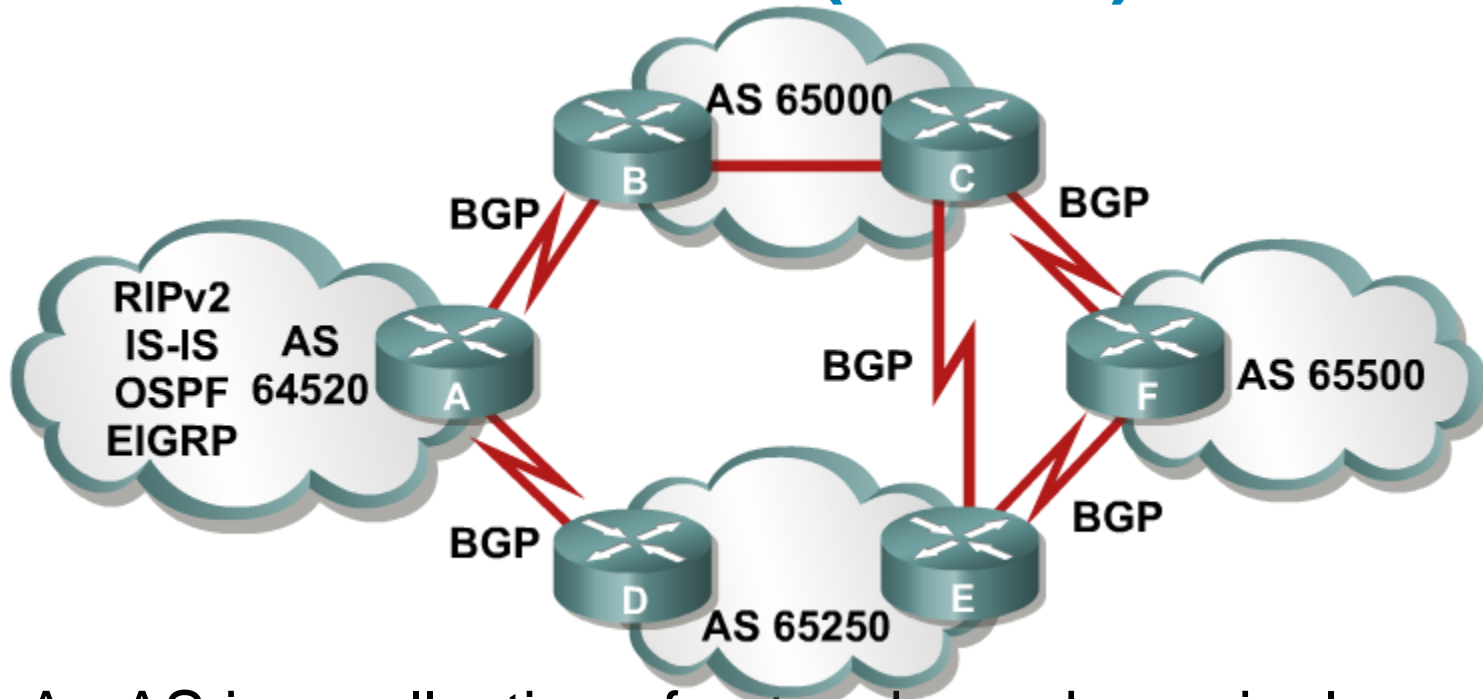
Objectives

- Describe various databases and messages used in BGP.
- Describe how to configure a BGP session for external and internal neighboring routers.
- Describe how to administratively shutdown a BGP neighbor.
- Describe EBGP peering.
- Describe BGP Established and Idle states.
- Identify problems associated with a router remaining in Active states.
- Configure BGP peer groups.
- Configure BGP authentication.
- Troubleshoot BGP sessions
- Describe how to use the debug `ip debug` command.
- Describe how to use the BGP Local Preference attribute.
- Describe how to configure route maps using the BGP MED attribute.

Review



BSCI Module 6 BGP (review)



- An AS is a collection of networks under a single technical administration.
- IGP's operate within an AS.
- BGP is used between autonomous systems.
- Exchange of loop-free routing information is guaranteed.

BSCI Module 6 BGP (review)

- An AS is a group of routers that share similar routing policies and operate within a single administrative domain.
- An AS can be a collection of routers running a single IGP, or it can be a collection of routers running different protocols all belonging to one organization.
- In either case, the outside world views the entire Autonomous System as a single entity.

Review from CCNP 1 Advanced Routing 3.1

BSCI Module 6 BGP (review)

- Internet Assigned Numbers Authority (IANA) is enforcing a policy whereby organizations that connect to a single provider and share the provider's routing policies use an AS number from the private pool, 64,512 to 65,535.
- These private AS numbers appear only within the provider's network and are replaced by the provider's registered number upon exiting the network.

Review from CCNP 1 Advanced Routing 3.1

BSCI Module 6 BGP (review)

- When two routers establish a TCP enabled BGP connection, they are called neighbors or peers.
- Each router running BGP is called a BGP speaker.
- Peer routers exchange multiple messages to open and confirm the connection parameters, such as the version of BGP to be used.
- If there are any disagreements between the peers, notification errors are sent and the connection fails.

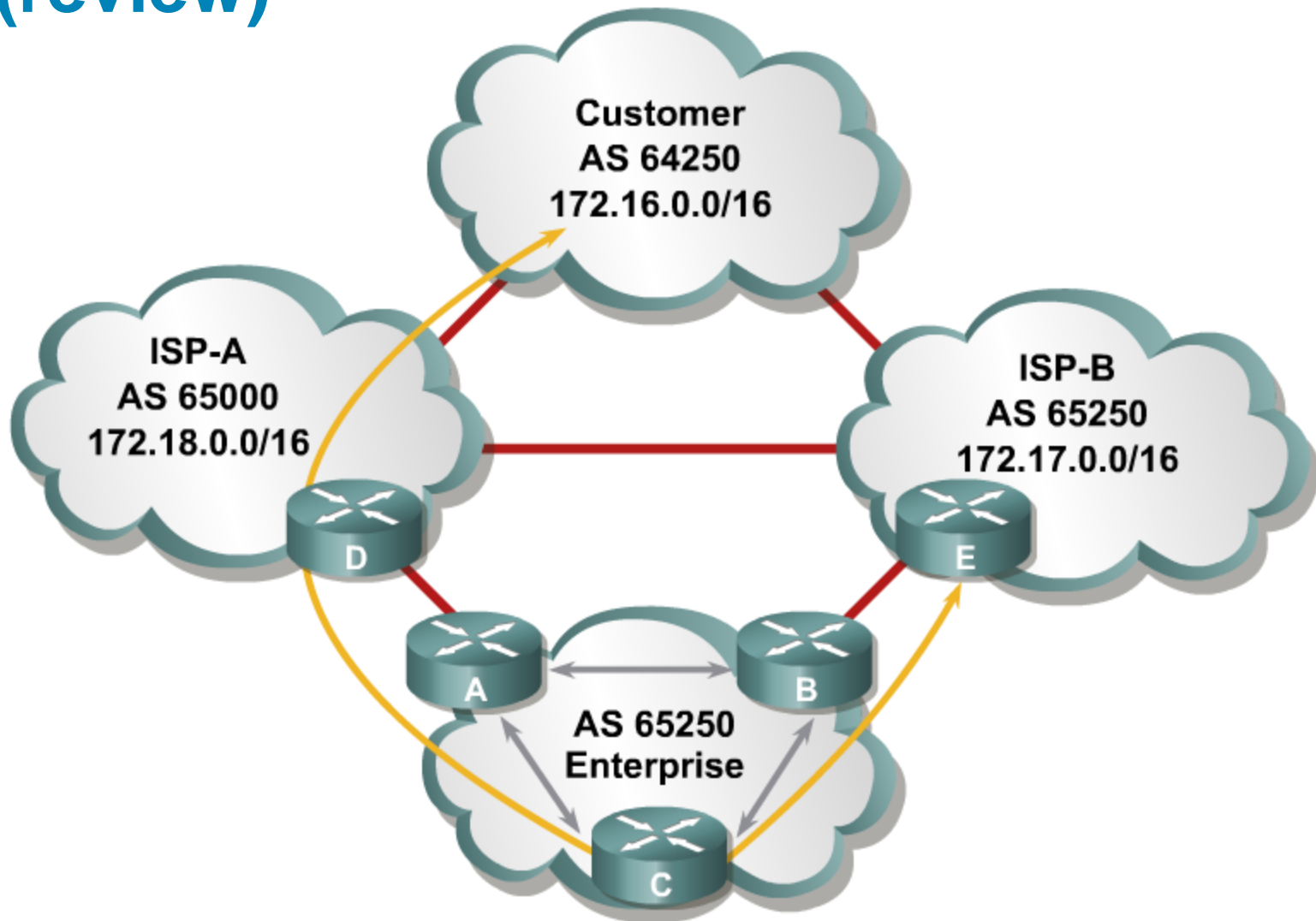
Review from CCNP 1 Advanced Routing 3.1

BSCI Module 6 BGP (review)

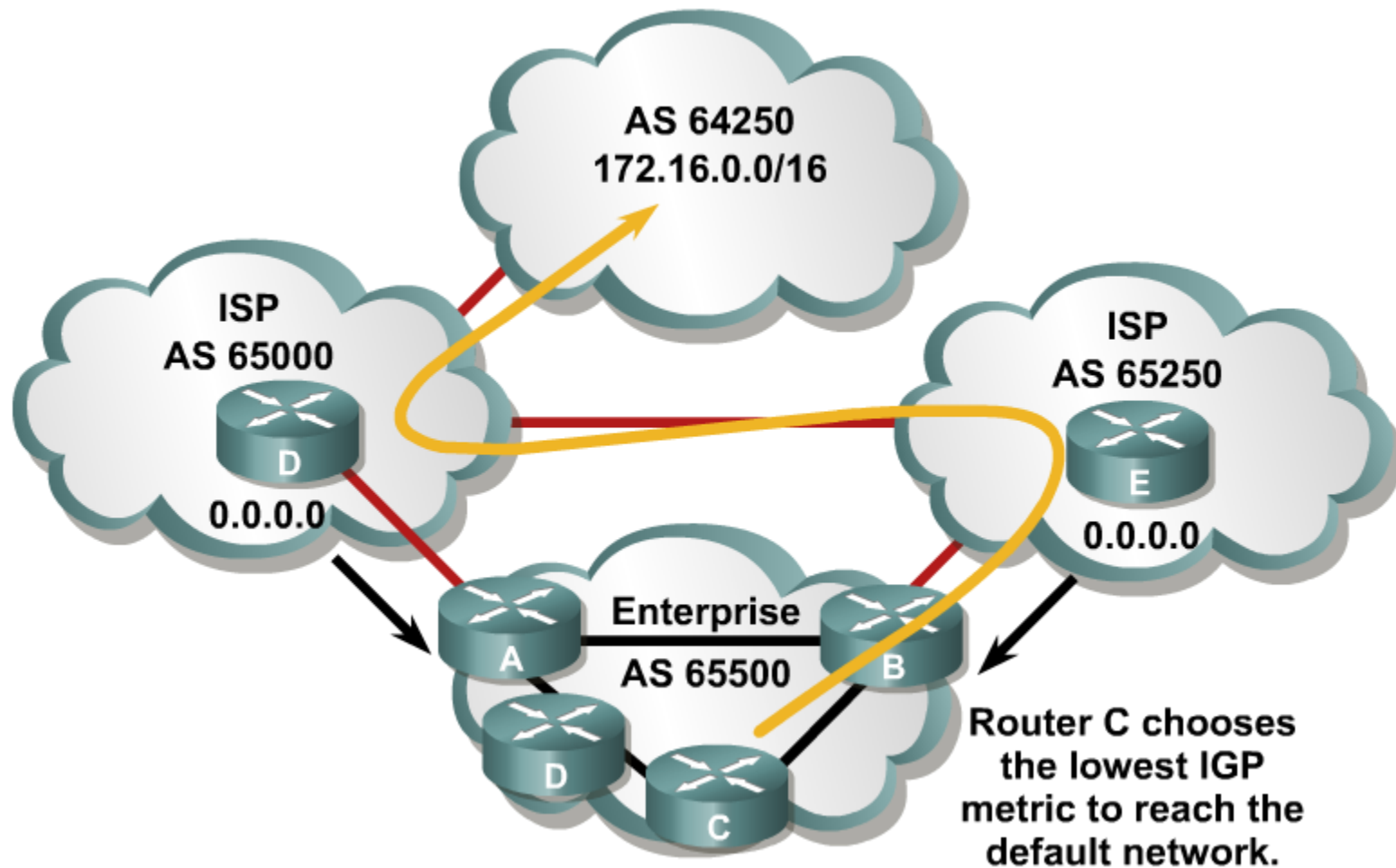
- When BGP neighbors first establish a connection, they exchange all candidate BGP routes.
- After this initial exchange, incremental updates are sent as network information changes.
- Incremental updates are more efficient than complete table updates.
- This is especially true with BGP routers, which may contain the complete Internet routing table.

Review from CCNP 1 Advanced Routing 3.1

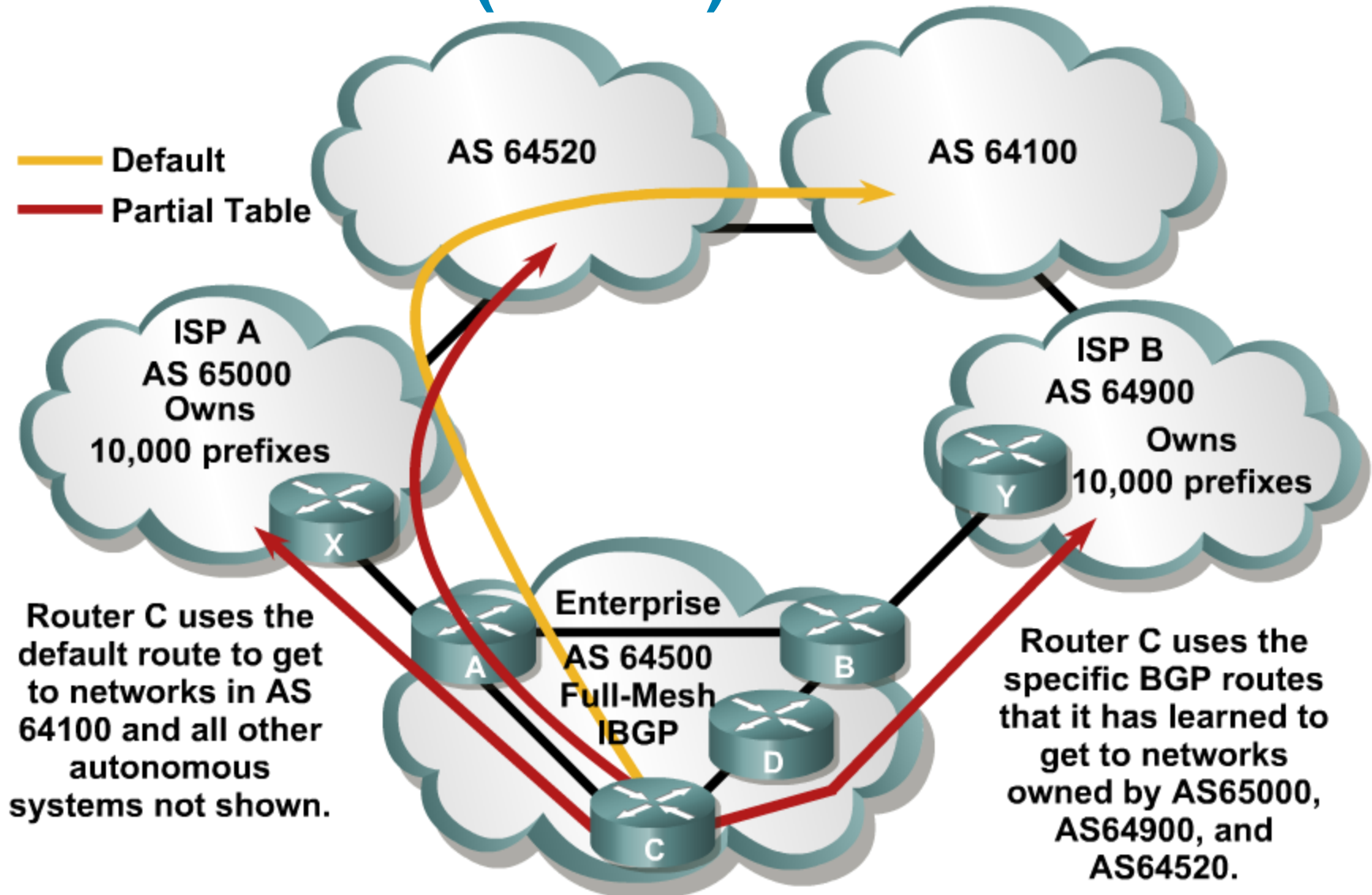
Using BGP to Connect to the Internet (review)



Example: Default Routes from All Providers (review)



Default Routes from All Providers and Partial Table (review)



BGP Terms



BGP Databases

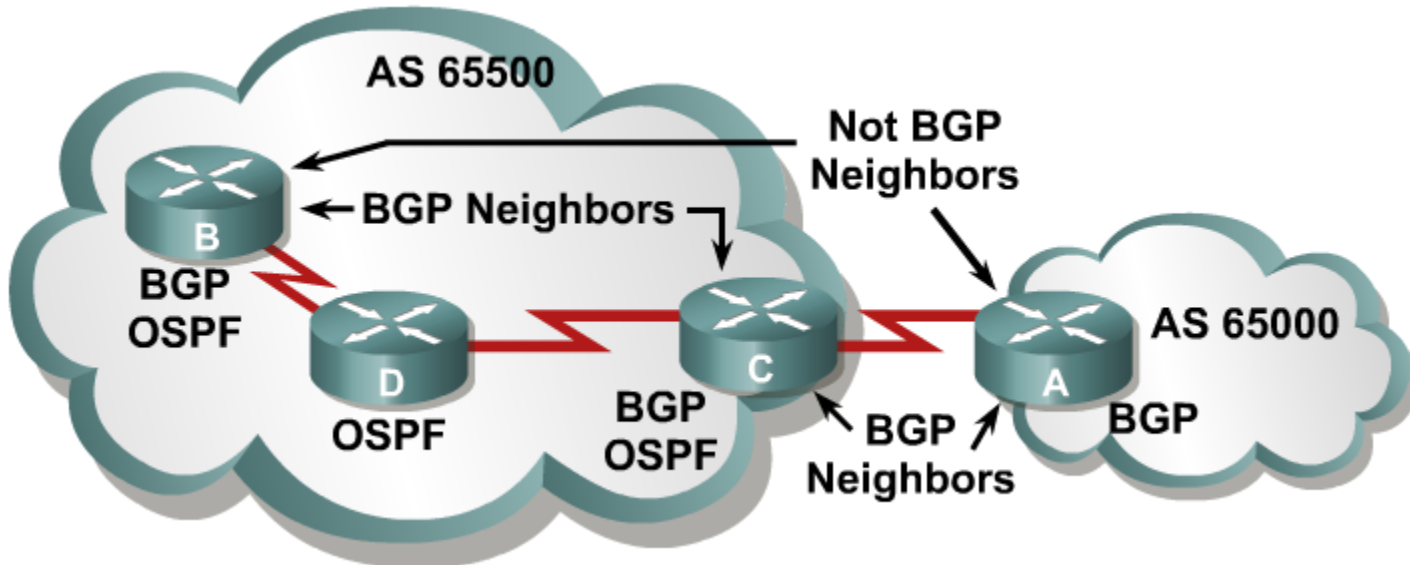
- Neighbor table
 - List of BGP neighbors
- BGP table (forwarding database)
 - List of all networks learned from each neighbor
 - Can contain multiple paths to destination networks
 - Contains BGP attributes for each path
- IP routing table
 - List of best paths to destination networks

BGP Message Types

BGP defines the following message types:

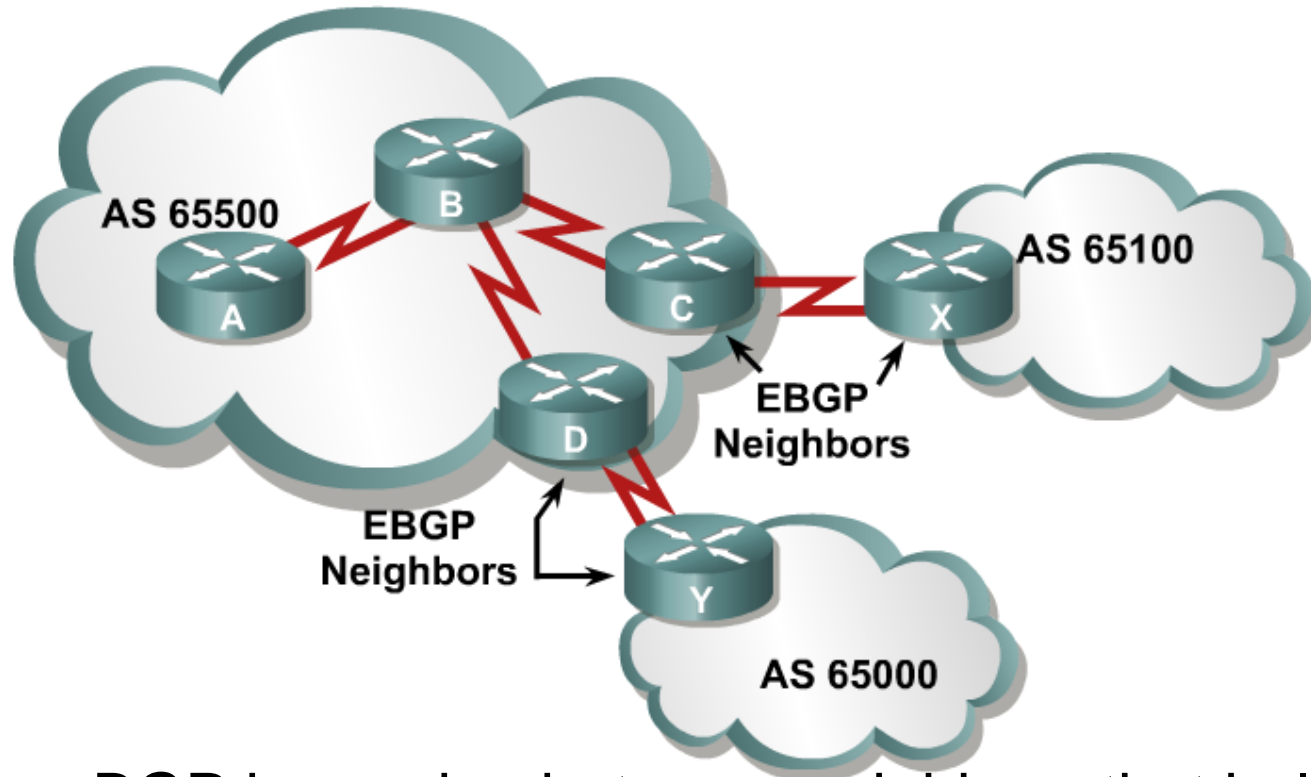
- Open
 - Includes holdtime and BGP router ID
- Keepalive
- Update
 - Information for one path only (could be to multiple networks)
 - Includes path attributes and networks
- Notification
 - When error is detected
 - BGP connection is closed after being sent

Peers = Neighbors



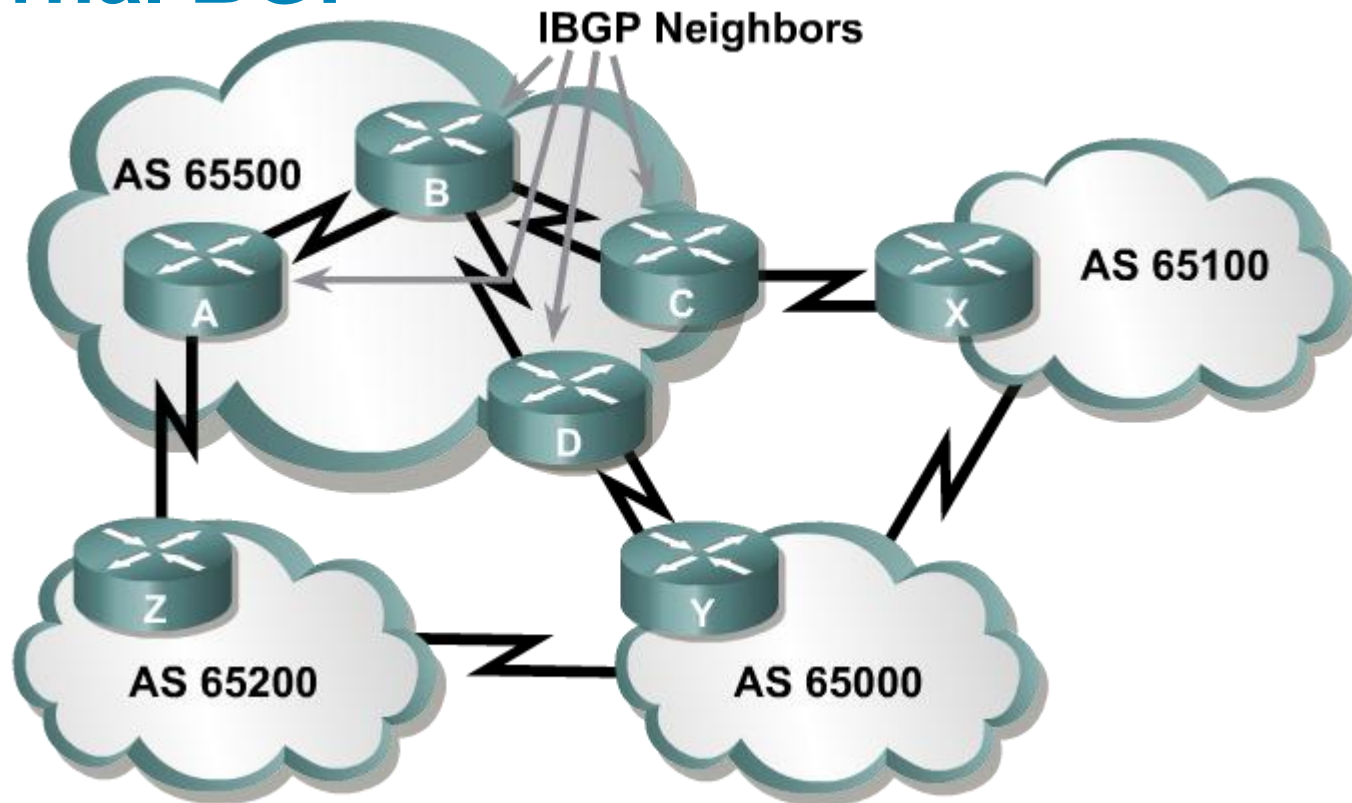
- A “BGP peer,” also known as a “BGP neighbor,” is a specific term that is used for BGP speakers that have established a neighbor relationship.
- Any two routers that have formed a TCP connection to exchange BGP routing information are called BGP peers or BGP neighbors.

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.

BGP Commands



BGP Commands

Router (config) #

```
router bgp autonomous-system
```

- This command just enters router configuration mode; subcommands must be entered in order 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 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.

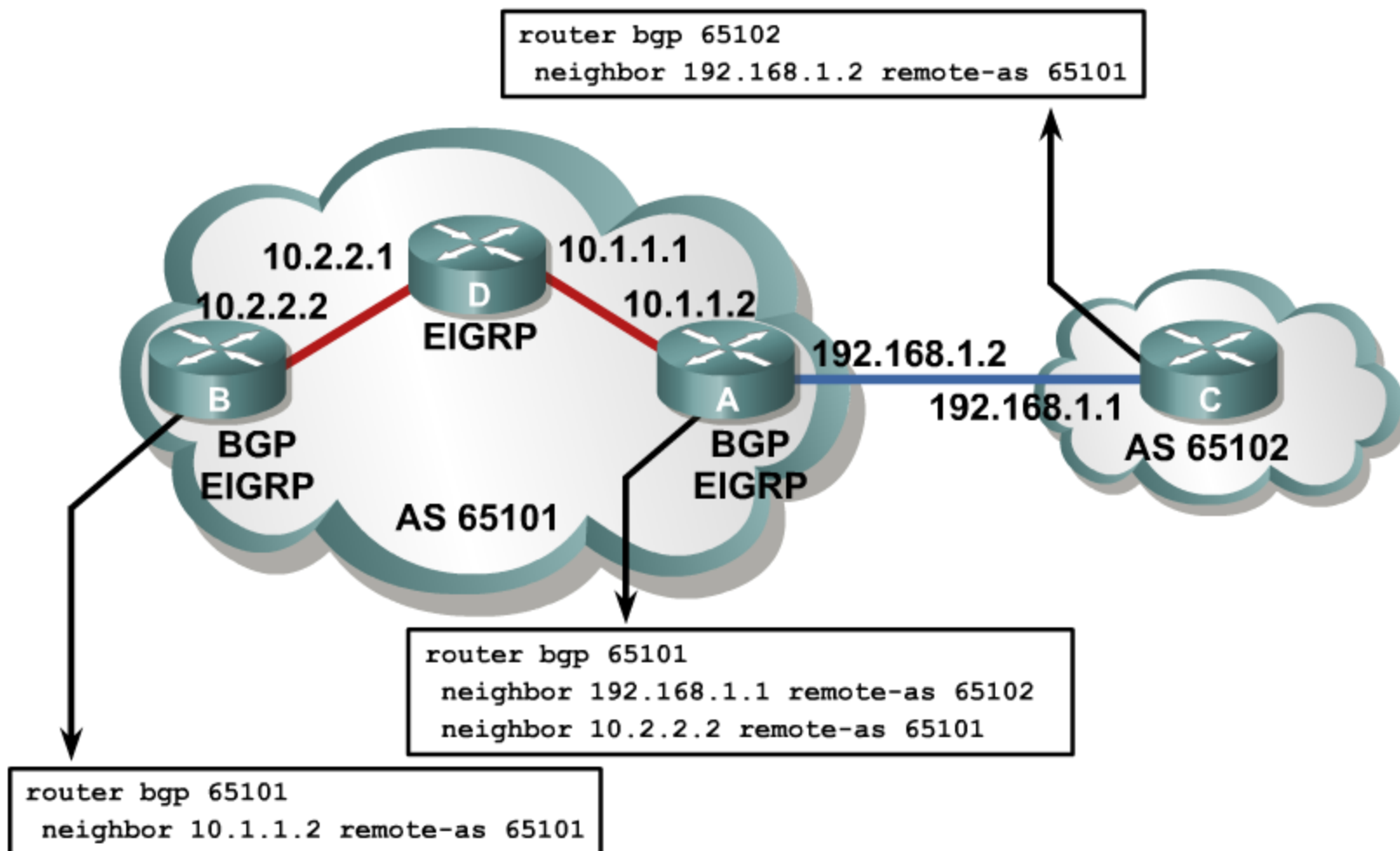
BGP `neighbor remote-as` Command

Router (config-router) #

```
neighbor {ip-address | peer-group-name}  
remote-as autonomous-system
```

- The **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 **remote-as** 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.

Example: BGP neighbor Command



BGP neighbor shutdown Command

Router(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

Router(config-router)#

```
no neighbor {ip-address | peer-group-name} shutdown
```

- Re-enables a BGP neighbor that has been administratively shut down

BGP neighbor update-source Command

Router (config-router) #

```
neighbor {ip-address | peer-group-name} update-source  
interface-type interface-number
```

- This 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 Addresses



```
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
!
router eigrp 1
 network 10.0.0.0
 network 2.0.0.0
```

```
router bgp 65101
 neighbor 192.168.1.1 remote-as 65102
 neighbor 2.2.2.2 remote-as 65101
 neighbor 2.2.2.2 update-source Loopback0
!
router eigrp 1
 network 10.0.0.0
 network 3.0.0.0
```


BGP neighbor ebgp-multihop Command

Router (config-router) #

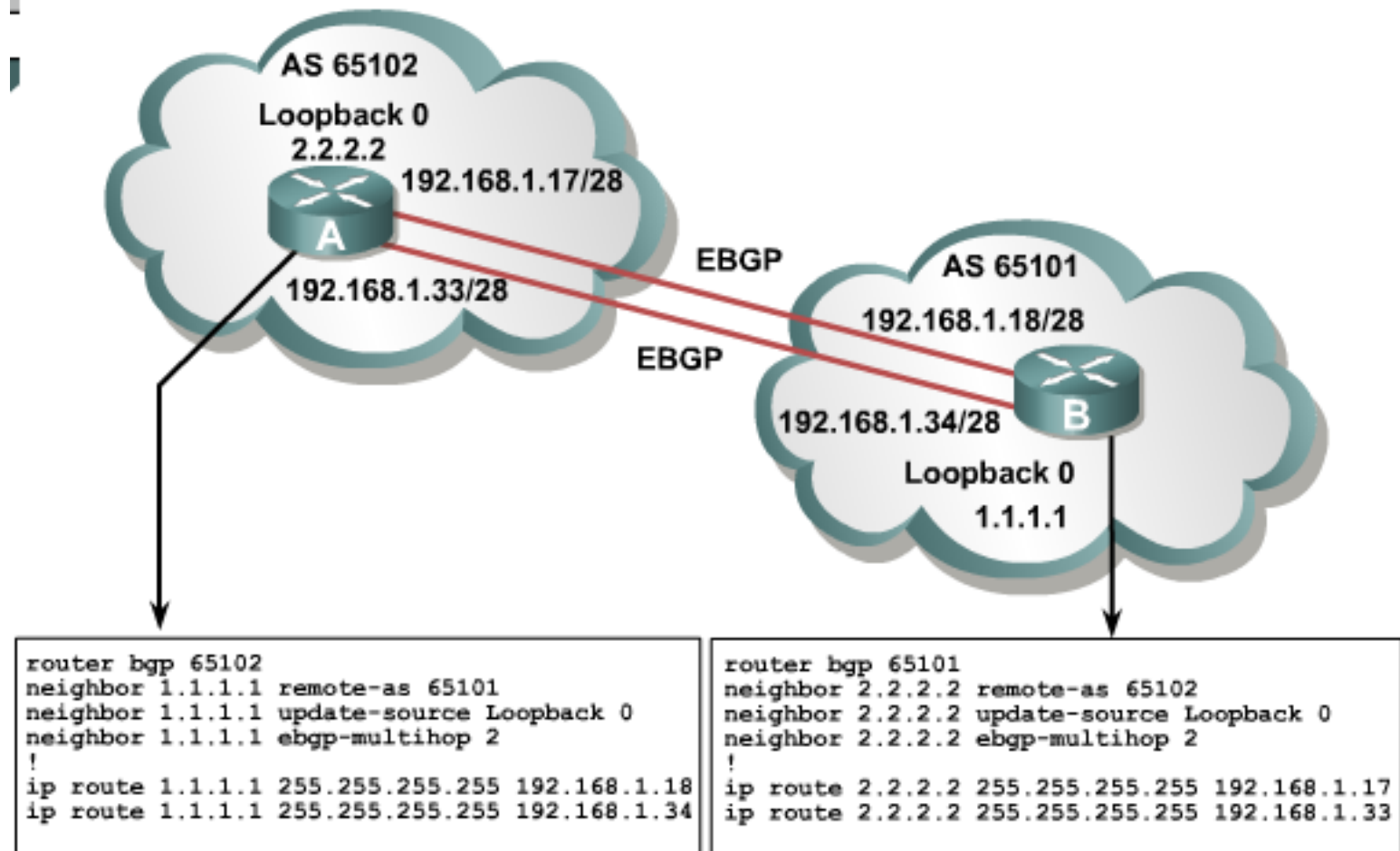
```
neighbor {ip-address | peer-group-name} ebgp-multihop [ttl]
```

- This command increases the default of one hop for EBGP peers.
- It allows routes to the EBGP loopback address (which will have a hop count greater than 1).

The neighbor ebgp multihop Command Parameters

- *ip-address* is the IP address of the BGP-speaking neighbor.
- *peer-group-name* is the Name of a BGP peer group.
- *ttl* (Optional) TTL in the range from 1 to 255 hops

Example: ebgp-multihop Command



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
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
10.1.0.2	4	65001	11	11	124	0	0	00:02:28	8
172.31.1.3	4	64998	21	18	124	0	0	00:01:13	6
172.31.11.4	4	64999	11	10	124	0	0	00:01:11	6

BGP States



BGP States

When establishing a BGP session, BGP goes through the following steps:

- **Idle**: Router is searching routing table to see if a route exists to reach the neighbor.
- **Connect**: Router found a route to the neighbor and has completed the three-way TCP handshake.
- **Open sent**: Open message sent, with the parameters for the BGP session.
- **Open confirm**: Router received agreement on the parameters for establishing session.

Alternatively, router goes into **Active** state if no response to open message

- **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?
- **Established:** The established state is the proper state for BGP operations.
- In 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

```
RouterA#sh ip bgp neighbors
BGP neighbor is 172.31.1.3, remote AS 64998, external link
  BGP version 4, remote router ID 172.31.2.3
  BGP state = Established, up for 00:19:10
  Last read 00:00:10, last write 00:00:10, hold time is 180, keepalive
  interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

```

	Sent	Rcvd
Opens:	7	7
Notifications:	0	0
Updates:	13	38

```
<output omitted>
```

BGP Active State Troubleshooting

- **Active:** The router has sent out 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:
 1. Neighbor does not have a route to the source IP address of the BGP open packet generated by this router
 2. Neighbor peering with the wrong address
 3. Neighbor does not have a **neighbor** statement for this router
 4. AS number misconfiguration

Example: BGP Active State Troubleshooting

AS number misconfiguration:

- 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
```

```
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 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
```

BGP Peer Groups & Neighbors



Using a Peer Group

Router (config-router) #

```
neighbor peer-group-name peer-group
```

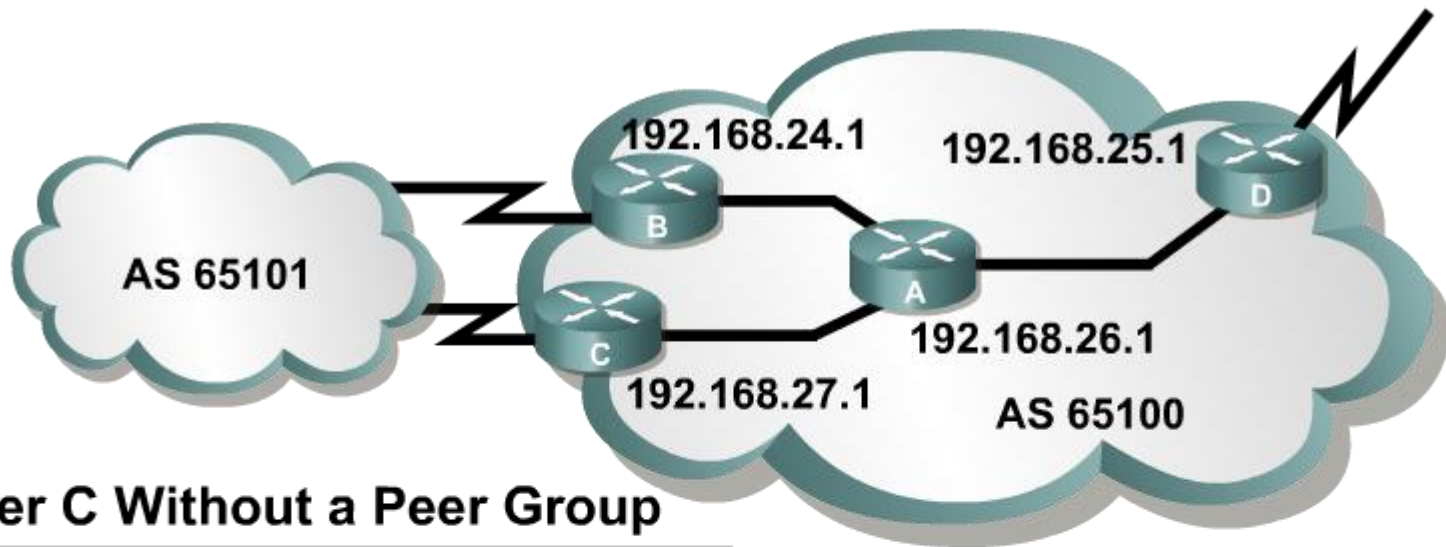
- This command creates a peer group.

Router (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.

Example: Using a Peer Group



Router C Without 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 C Using a Peer Group

```
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
```

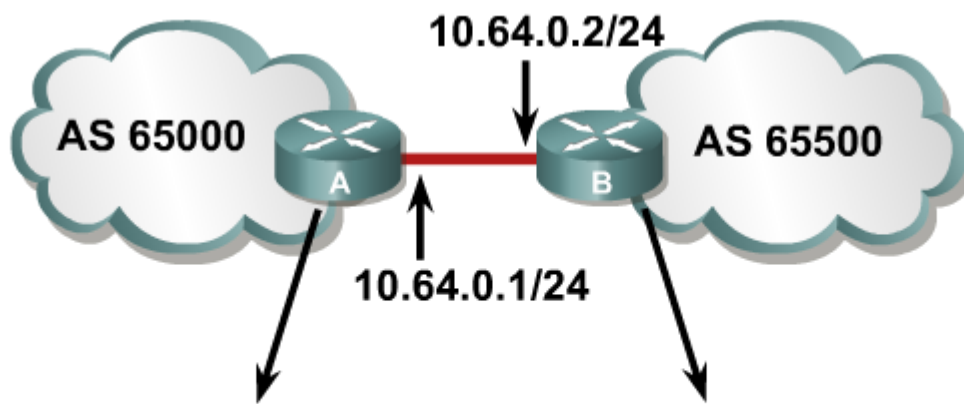
BGP Neighbor Authentication

Router (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

Example: BGP Neighbor Authentication



```
router bgp 65500
 neighbor 10.64.0.2 remote-as 65500
 neighbor 10.64.0.2 password v6lne0qkel133&
```

```
router bgp 65500
 neighbor 10.64.0.1 remote-as 65000
 neighbor 10.64.0.1 password v6lne0qkel133&
```

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
   Network                Next Hop                Metric LocPrf Weight Path
*> 10.1.0.0/24            0.0.0.0                  0           32768 i
* i                       10.1.0.2                  0          100           0 i
*> 10.1.1.0/24            0.0.0.0                  0           32768 i
*>i10.1.2.0/24           10.1.0.2                  0          100           0 i
*> 10.97.97.0/24         172.31.1.3                0           0 64998 64997 i
*                          172.31.11.4                0           0 64999 64997 i
* i                       172.31.11.4                0          100           0 64999 64997 i
*> 10.254.0.0/24         172.31.1.3                0           0 64998 i
*                          172.31.11.4                0           0 64999 64998 i
* i                       172.31.1.3                0          100           0 64998 i
r> 172.31.1.0/24         172.31.1.3                0           0 64998 i
r                          172.31.11.4                0           0 64999 64998 i
r i                       172.31.1.3                0          100           0 64998 i
*> 172.31.2.0/24         172.31.1.3                0           0 64998 i
<output omitted>
```

Displays networks from lowest to highest.

Resetting BGP Sessions



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 will be 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

Router#

```
clear 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.

Router#

```
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

Router#

```
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

Router (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.

Router#

```
clear ip bgp {*|neighbor-address} soft in
```

- Uses the stored information to generate new inbound updates.

Route Refresh: Dynamic Inbound Soft Reset

Router#

```
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 IOS 12.0(2)S and 12.0(6)T

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 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
```

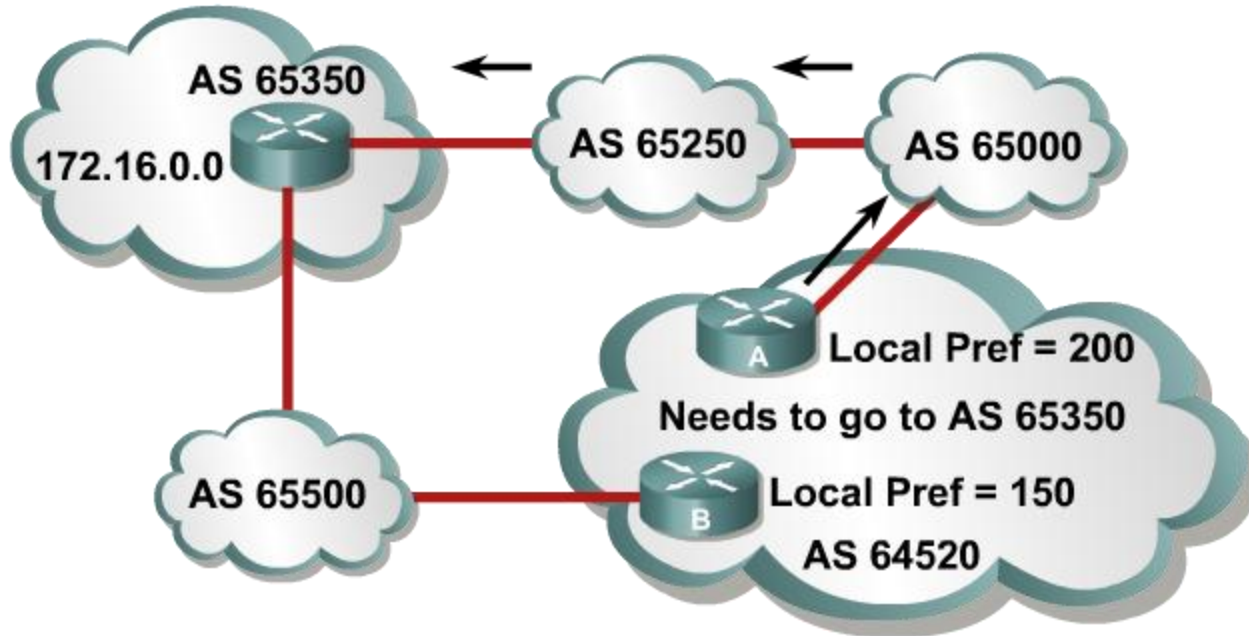
```
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd 10.1.2.0/24
```

```
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd 10.1.0.0/24
```

BGP Local Preference Case Study



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 AS only).
- The local preference attribute is well-known and discretionary.
- Default value = 100

Changing BGP Local Preference For All Routes

Local preference is used in these ways:

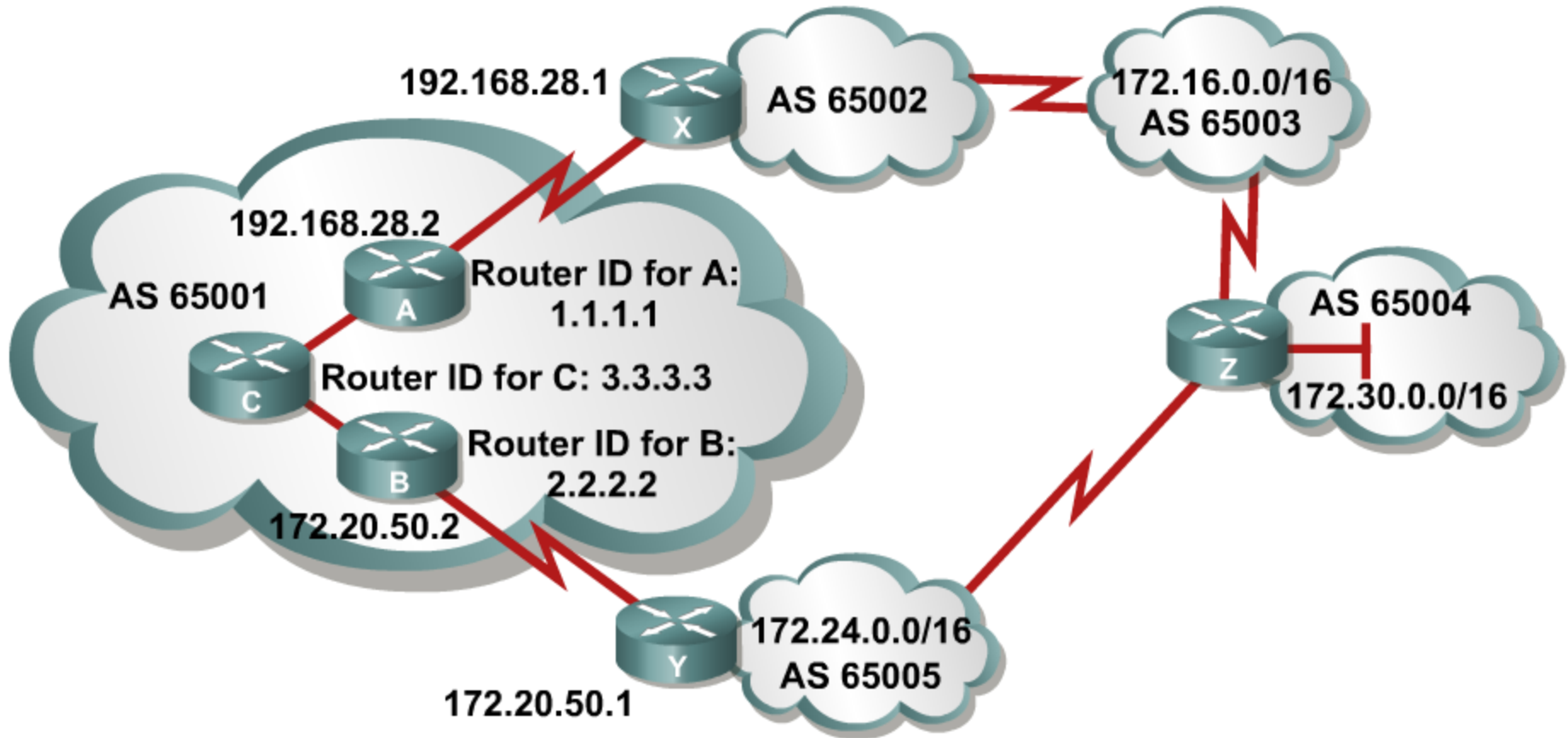
- Within an AS between IBGP speakers.
- To determine the best path to exit the AS to reach an outside network.
- Set to 100 by default; higher values are preferred.

`Router(config-router) #`

```
bgp default local-preference value
```

- Changes the default local preference value.
- All routes advertised to an IBGP neighbor have the local preference set to the value specified.

Local Preference Case Study



What is the best path for router C to 65003, 65004, and 65005?

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
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i172.16.0.0	172.20.50.1	100	0	65005	65004 65003 i
*>i	192.168.28.1	100	0	65002	65003 i
*>i172.24.0.0	172.20.50.1	100	0	65005	i
* i	192.168.28.1	100	0	65002	65003 65004 65005 i
*>i172.30.0.0	172.20.50.1	100	0	65005	65004 i
* i	192.168.28.1	100	0	65002	65003 65004i

By default, BGP selects the shortest AS path as the best (>) path.

In AS 65001, the percent 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.

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 remote-as 65001 update-source loopback0
neighbor 3.3.3.3 remote-as 65001 update-source loopback0
neighbor 192.168.28.1 remote-as 65002
neighbor 192.168.28.1 route-map local_pref in
!
access-list 65 permit 172.30.0.0 0.0.255.255
!
route-map local_pref permit 10
match ip address 65
set local-preference 400
!
route-map local_pref permit 20
```

Router C BGP Table with Local Preference Learned

```
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
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i172.16.0.0	172.20.50.1		100	0	65005 65004 65003 i
*>i	192.168.28.1		100	0	65002 65003 i
*>i172.24.0.0	172.20.50.1		100	0	65005 i
* i	192.168.28.1		100	0	65002 65003 65004 65005 i
* i172.30.0.0	172.20.50.1		100	0	65005 65004 i
*>i	192.168.28.1		400	0	65002 65003 65004i

- 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 due to the next hop of 192.168.28.1 having 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).

BGP MED

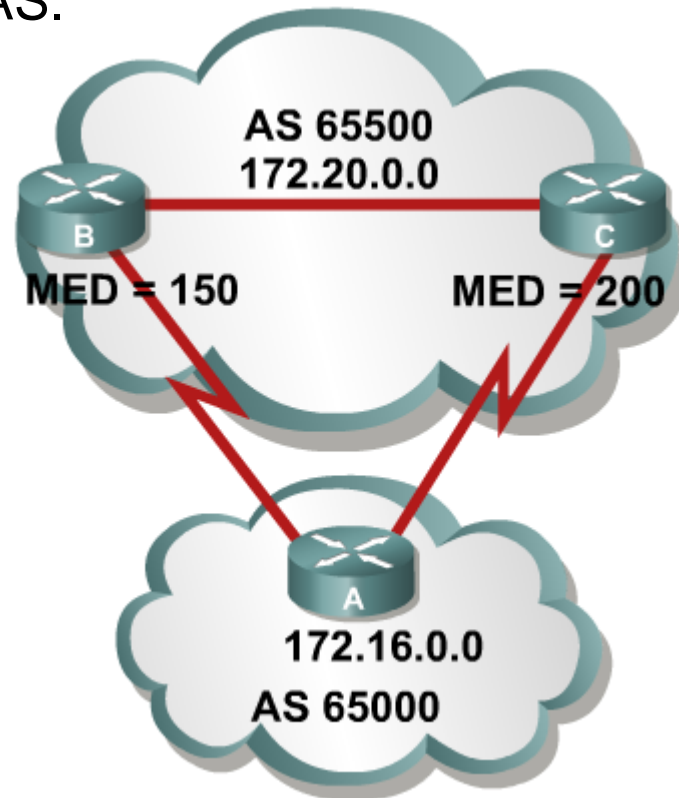


MED Attribute

The paths with the lowest MED (also called the metric) value are the most desirable:

- MED is used to advertise to EBGPs neighbors how to exit their AS to reach networks owned by this AS.

The MED attribute is optional and nontransitive.



Changing BGP MED For All Routes

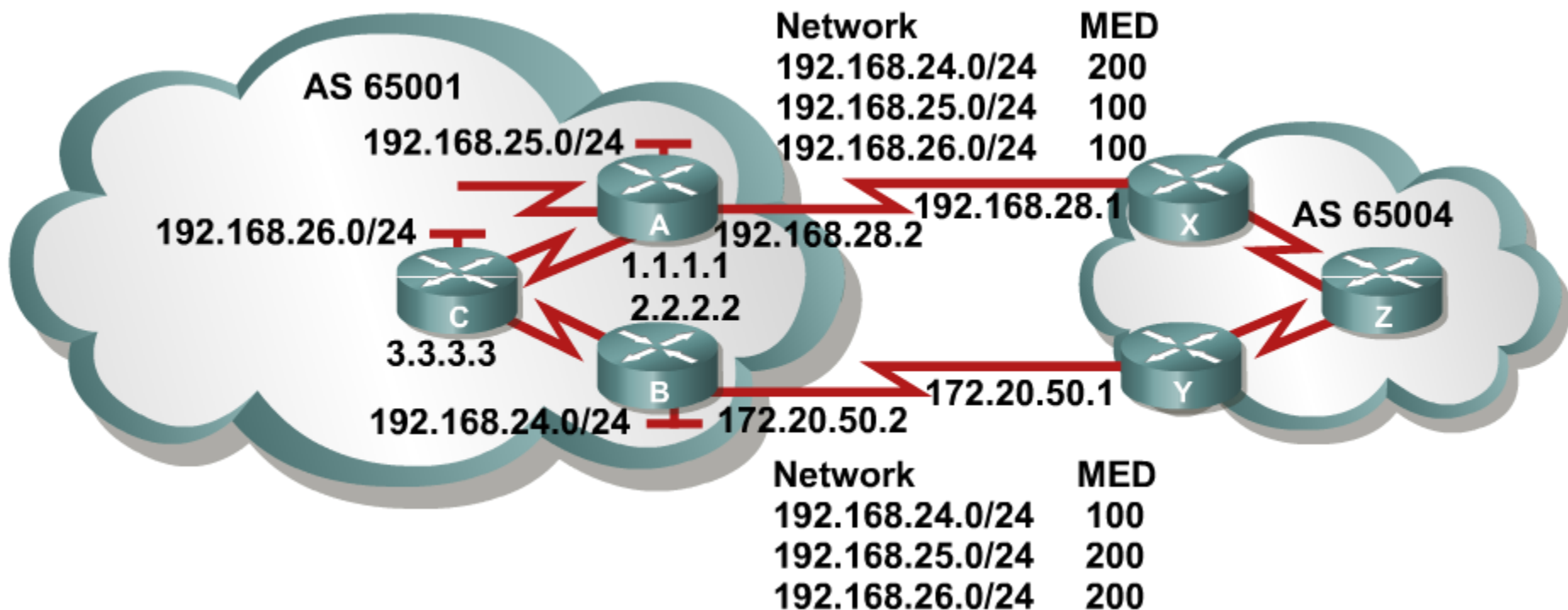
- MED is used when multiple paths exist between two autonomous systems.
- A lower MED value is preferred.
- The default setting for Cisco is MED = 0.
- The metric is optional, nontransitive attribute.
- Usually, MED is shared only between two autonomous systems that have multiple EBGP connections with each other.

Router (config-router) #

```
default-metric number
```

- 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
```

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
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*>i192.168.24.0	172.20.50.2	100	100	0	65001 i
* i	192.168.28.2	200	100	0	65001 i
* i192.168.25.0	172.20.50.2	200	100	0	65001 i
*>i	192.168.28.2	100	100	0	65001 i
* i192.168.26.0	172.20.50.2	200	100	0	65001 i
*>i	192.168.28.2	100	100	0	65001 i

- 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).

Route Selection Decision Process

Consider only (synchronized) routes with no AS loops and a valid next hop, and then:

- Prefer highest weight (local to router).
- Prefer highest local preference (global within AS).
- Prefer route originated by the local router (next hop = 0.0.0.0).
- Prefer shortest AS path.
- Prefer lowest origin code (IGP < EGP < incomplete).
- Prefer lowest MED (exchanged between autonomous systems).
- Prefer EBGP path over IBGP path.
- Prefer the path through the closest IGP neighbor.
- Prefer oldest route for EBGP paths.
- Prefer the path with the lowest neighbor BGP router ID.
- Prefer the path with the lowest neighbor IP address.

Summary

- BGP is a path-vector routing protocol that allows routing policy decisions at the AS level to be enforced.
- BGP forms EBGP relationships with external neighbors and IBGP with internal neighbors. All routers in the transit path within an AS must run fully-meshed IBGP.
- When BGP is properly configured, it will: establish a neighbor relationship, set the next-hop address, set the source IP address of a BGP update, and announce the networks to other BGP routers.
- BGP performs a multi-step process when selecting the best path to reach a destination.
- BGP can manipulate path selection to affect inbound and outbound traffic policies of an AS. Route maps can be configured in order to manipulate the local preference and MED BGP attributes.

