Chapter 6:

Implementing a Border Gateway Protocol Solution for ISP Connectivity

- CCNP-RS ROUTE

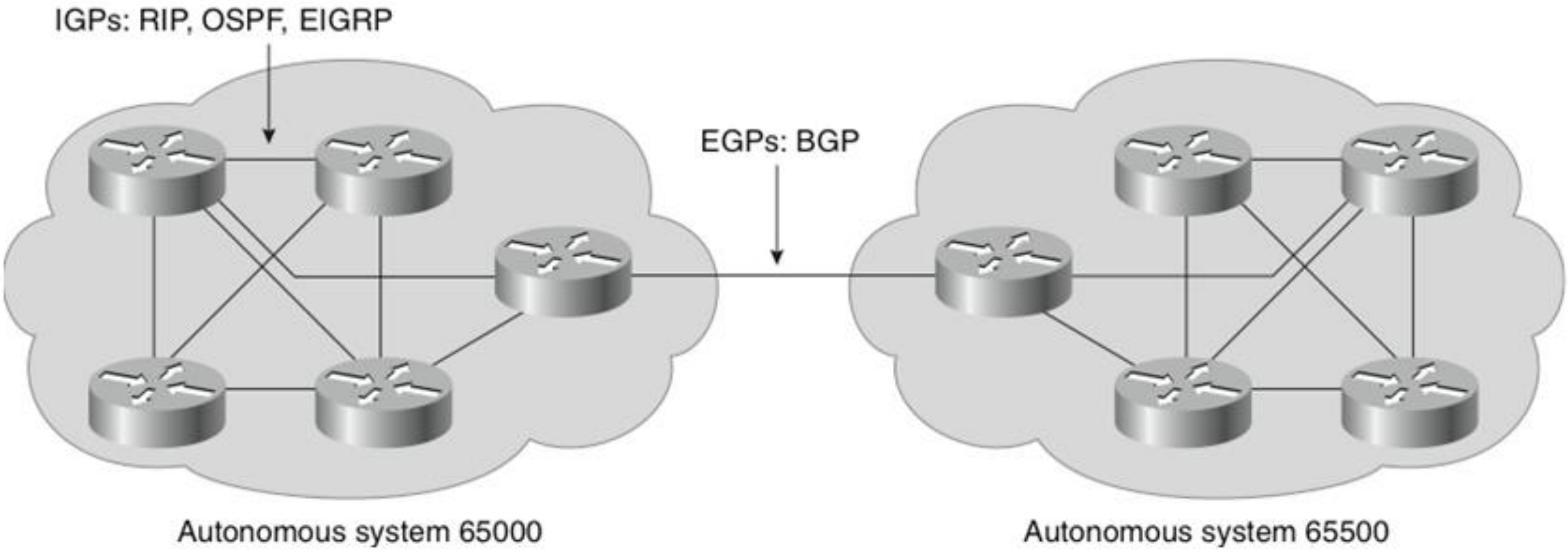
Chapter 6 Objectives

- Describe basic BGP terminology and operation, including EBGP and IBGP.
- Configure basic BGP.
- Verify and troubleshoot basic BGP.
- Describe and configure various methods for manipulating path selection.
- Describe and configure various methods of filtering BGP rotuing updates.

BGP Terminology, Concepts, and Operation

IGP versus EGP

- **Interior gateway protocol (IGP)**
  - A routing protocol operating within an Autonomous System (AS).
  - RIP, OSPF, and EIGRP are IGPs.
- **Exterior gateway protocol (EGP)**
  - A routing protocol operating between different AS.
  - BGP is an interdomain routing protocol (IDRP) and is an EGP.

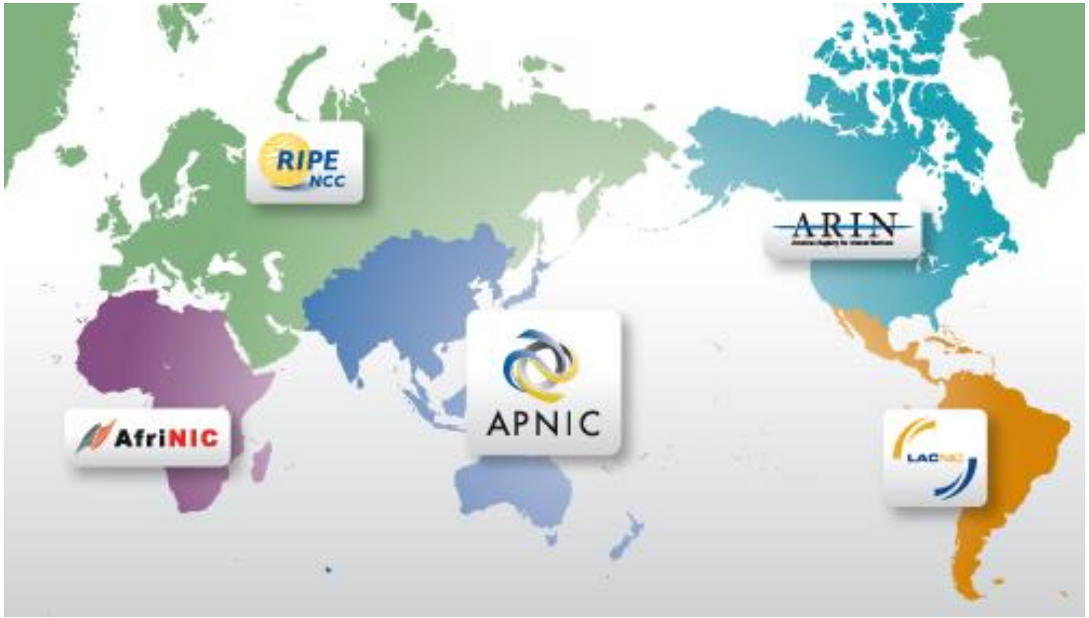


Autonomous Systems (AS)

- An AS is a group of routers that share similar routing policies and operate within a single administrative domain.
- An AS typically belongs to one organization.
  - A single or multiple interior gateway protocols (IGP) may be used within the AS.
  - In either case, the outside world views the entire AS as a single entity.
- If an AS connects to the public Internet using an exterior gateway protocol such as BGP, then it must be assigned a unique AS number which is managed by the Internet Assigned Numbers Authority (IANA).

IANA

- The IANA is responsible for allocating AS numbers through five Regional Internet Registries (RIRs).
  - RIRs are nonprofit corporations established for the purpose of administration and registration of IP address space and AS numbers in key geographic locations.



Regional Internet Registries (RIRs)

RIR Name	Geographic Coverage	Link
AfriNIC	Continent of Africa	www.afrinic.net
APNIC (Asia Pacific Network Information Centre)	Asia Pacific region	www.apnic.net
ARIN (American Registry for Internet Numbers)	Canada, the United States, and several islands in the Caribbean Sea and North Atlantic Ocean	www.arin.net
LACNIC (Latin America and Caribbean Internet Addresses Registry)	Central and South America and portions of the Caribbean	www.lacnic.net
RIPE (Réseaux IP Européens)	Europe, the Middle East, and Central Asia	www.ripe.net

AS Numbers

- AS numbers can be between **1** to **65,535**.
  - RIRs manage the AS numbers between **1** and **64,512**.
  - The **64,512 - 65,535** numbers are reserved for private use (similar to IP Private addresses).
  - The IANA is enforcing a policy whereby organizations that connect to a single provider use an AS number from the private pool.
- **Note**:
  - The current AS pool of addresses is predicted to run out by 2012.
  - For this reason, the IETF has released RFC 4893 and RFC 5398.
  - These RFCs describe BGP extensions to increase the AS number from the two-octet (16-bit) field to a four-octet (32-bits) field, increasing the pool size from **65,536** to **4,294,967,296** values.

BGP Basics

- The Internet is a collection of autonomous systems that are interconnected to allow communication among them.
  - BGP provides the routing between these autonomous systems.
- BGP is a path vector protocol.
- It is the only routing protocol to use TCP.
  - OSPF and EIGRP operate directly over IP. IS-IS is at the network layer.
  - RIP uses the User Datagram Protocol (UDP) for its transport layer.

BGP Basics

- BGP version 4 (BGP-4) is the latest version of BGP.
  - Defined in RFC 4271.
  - Supports supernetting, CIDR and VLSM .
- BGP4 and CIDR prevent the Internet routing table from becoming too large.
  - Without CIDR, the Internet would have 2,000,000 + entries.
  - With CIDR, Internet core routers manage around 300,000 entries.
  - http://bgp.potaroo.net/

# of Current BGP Routes

As of November 6, 2015, there were 589,614 routes in the routing tables of the Internet core routers.

http://bgpupdates.potaroo.net/instability/bgpupd.html

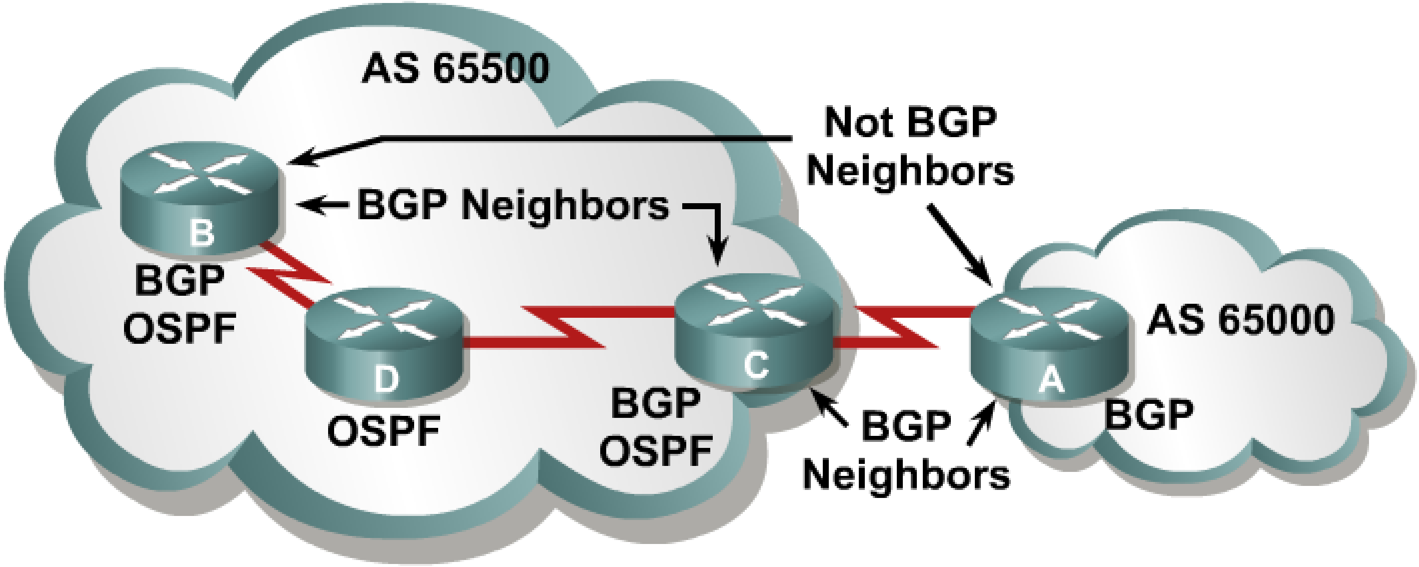
```
7 Day BGP Profile: 6-November-2015 00:00 - 12-November-2015 23:59 (UTC+1000)

Number of BGP Update Messages:        3424321
Number of Prefix Updates:      5855085
Number of Prefix Withdrawals:  232624
Average Prefixes per BGP Update:      1.78
Average BGP Update Messages per second:       4.95
Average Prefix Updates per second:    8.81
Peak BGP Update Message Rate per second:      4472     (13:38:12 Thu, 5-Nov-2015)
Peak Prefix Update Rate per second:   1178      (16:37:55 Thu, 5-Nov-2015)
Peak Prefix Withdraw Rate per second: 14027     (13:38:12 Thu, 5-Nov-2015)
Prefix Count:   589614
Updated Prefix Count:   384179
Stable Prefix Count:    205435
Origin AS Count:        52273
Updated Origin AS Count:       41380
Stable Origin AS Count:        10893
Unique Path Count:      410355
Updated Path Count:     261439
Stable Path Count:      148916
```

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.

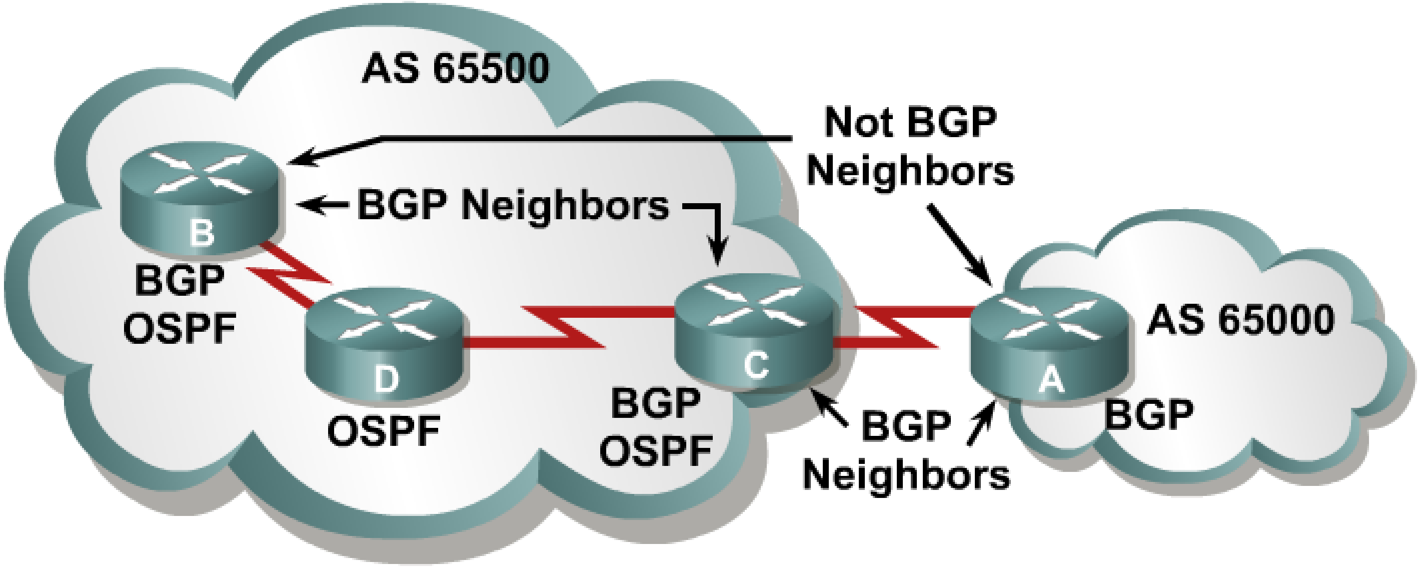
BGP Operational Overview

- When two routers establish a TCP enabled BGP connection, they are called **neighbors** or **peers**.
  - Peer routers exchange multiple connection messages.
- Each router running BGP is called a **BGP speaker**.



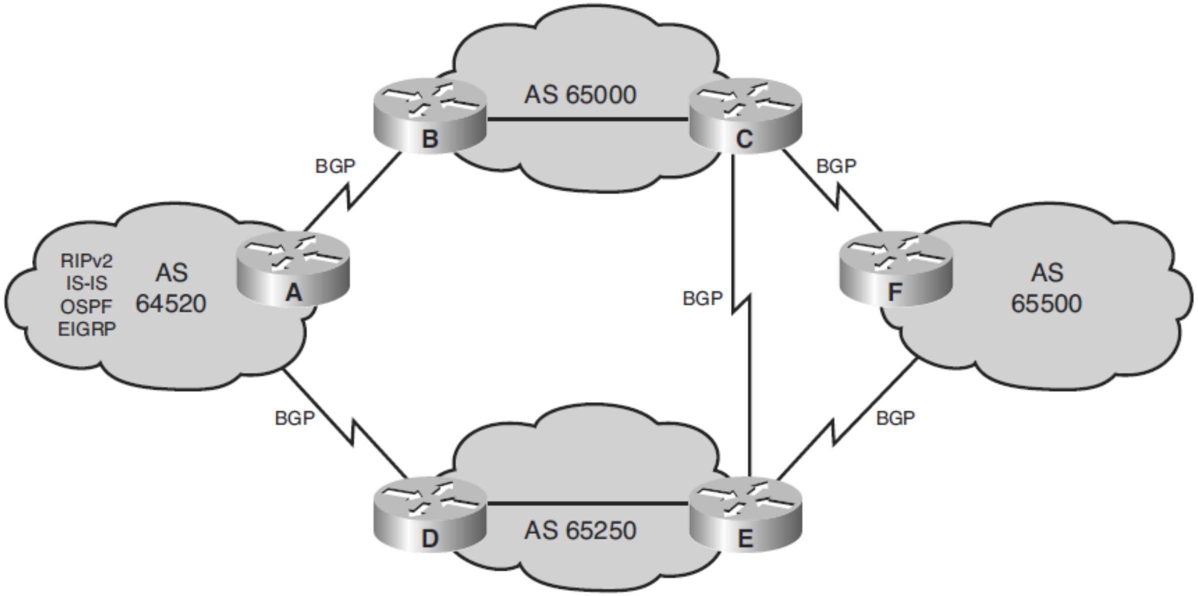
BGP Operational Overview

- 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.



BGP Use Between AS

- BGP provides an interdomain routing system that guarantees the loop-free exchange of routing information between autonomous systems.



Comparison BGP with IGPs

- BGP works differently than IGPs because it does not make routing decisions based on best path metrics.
  - Instead, BGP is a policy-based routing protocol that allows an AS to control traffic flow using multiple BGP attributes.
- Routers running BGP exchange network attributes including a list of the full path of BGP AS numbers that a router should take to reach a destination network.
- BGP allows an organization to fully use all of its bandwidth by manipulating these path attributes.

Comparing IGPs with BGP

Protocol	Interior or Exterior	Type	Hierarchy Required?	Metric
RIP	Interior	Distance vector	No	Hop count
OSPF	Interior	Link state	Yes	Cost
IS-IS	Interior	Link state	Yes	Metric
EIGRP	Interior	Advanced distance vector	No	Composite
BGP	**Exterior**	**Path vector**	**No**	**Path vectors (attributes)**

Connecting Enterprise Networks to an ISP

- Modern corporate IP networks connect to the global Internet.
- Requirements that must be determined for connecting an enterprise to an ISP include the following:
  - Public IP address space
  - Enterprise-to-ISP connection link type and bandwidth
  - Connection redundancy
  - Routing protocol

Public IP Address Space

- Public IP addresses are used:
  - By internal enterprise clients to access the Internet using NAT.
  - To make enterprise servers accessible from the Internet using static NAT.
- Public IP addresses are available from ISPs and RIRs.
  - Most enterprises acquire their IP addresses and AS number from ISPs.
  - Large enterprises may want to acquire IP addresses and AS number from a RIR.

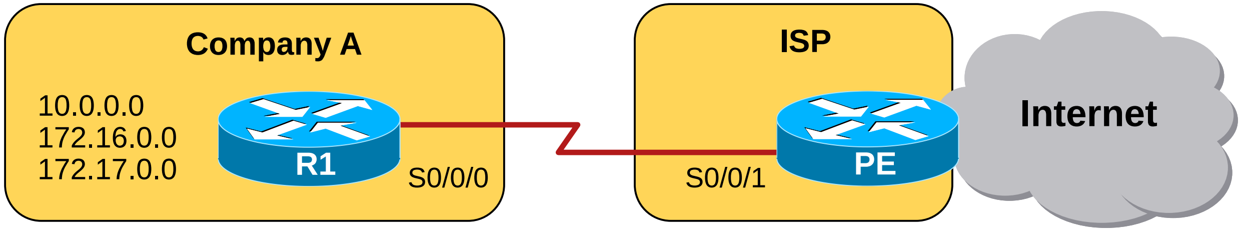
Connection and Routing Questions

- Which connection options does the ISP offer?
- Which routing options does the ISP offer?
- Will the enterprise network be connected to multiple ISPs?
- Does the routing need to support one link to an ISP or multiple links, to one or multiple ISPs?
- Is traffic load balancing over multiple links required?
- How much routing information needs to be exchanged with the ISP?
- Does the routing need to respond to the changes in the network topology, such as when a link goes down?

Using Static Routes Example

- Static routes are the simplest way to implement routing with an ISP.
  - Typically a customer has a single connection to an ISP and the customer uses a default route toward the ISP while the ISP deploys static routes toward the customer.

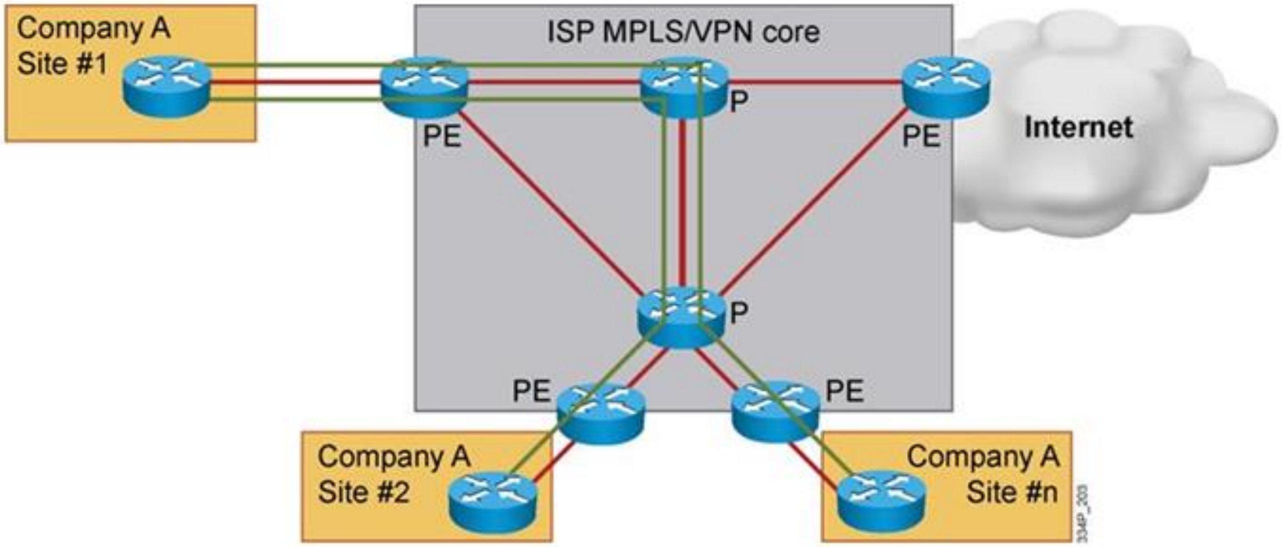
```
R1(config)# router eigrp 110
R1(config-router)# network 10.0.0.0
R1(config-router)# exit
R1(config)# ip default-network 0.0.0.0
R1(config)# ip route 0.0.0.0 0.0.0.0 serial 0/0/0
```



```
PE(config)# ip route 10.0.0.0 255.0.0.0 serial 0/0/1
PE(config)# ip route 172.16.0.0 255.255.0.0 serial 0/0/1
PE(config)# ip route 172.17.0.0 255.255.0.0 serial 0/0/1
```

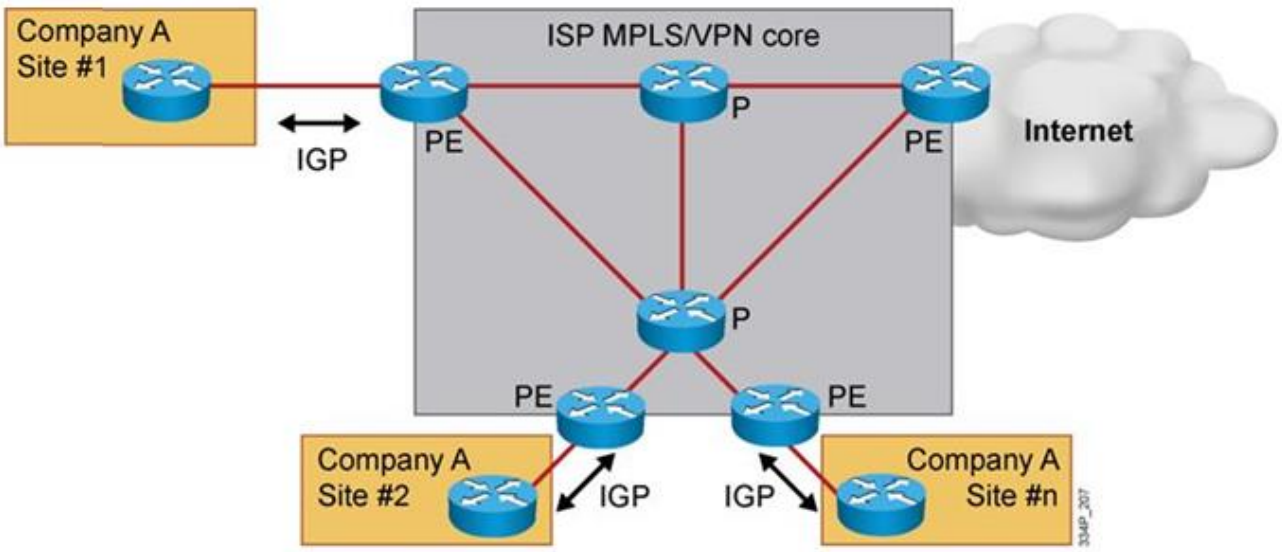
Using Layer 2 Circuit Emulation Example

- Service providers may offer Layer 2 MPLS VPN to connect Company A’s sites.
  - The VPN provides a Layer 2 service across the backbone and Company A’s edge routers are connected together on the same IP subnet.
  - There is no routing exchange between the ISP and Company A.



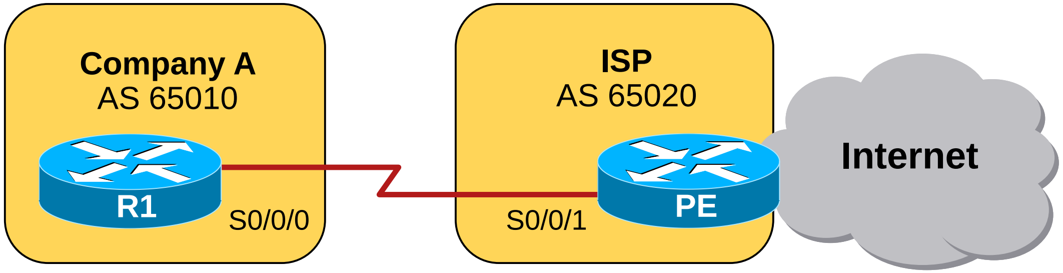
Using Layer 3 MPLS VPN Example

- Service providers may offer Layer 3 MPLS VPN.
  - The VPN provides a Layer 3 service across the backbone and Company A's edge routers are connected to ISP edge routers using different IP subnets.
  - Routing between the customer and ISP is required.



Using BGP

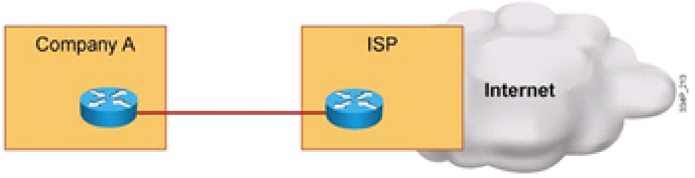
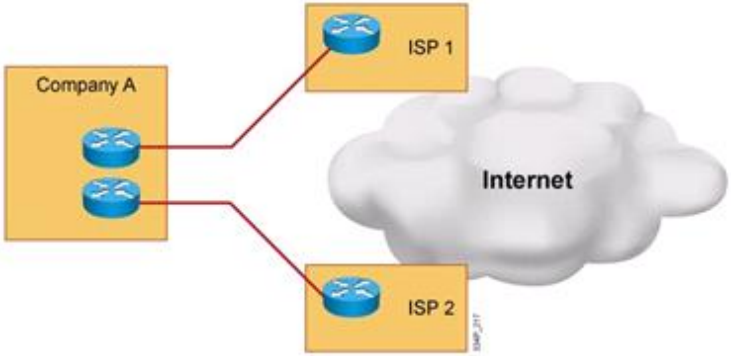
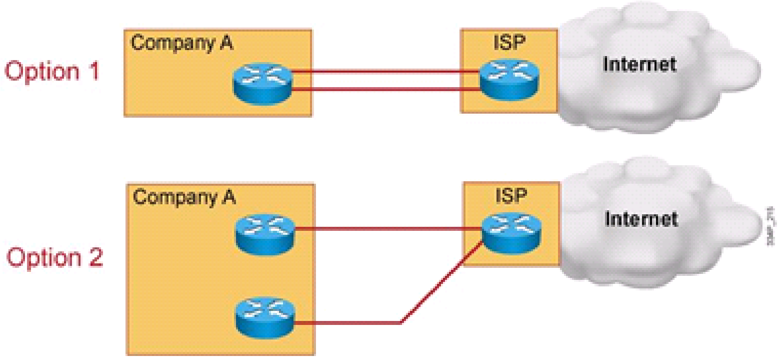
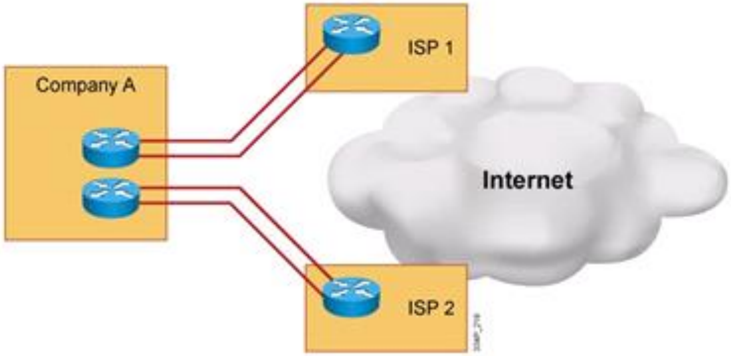
- BGP can be used to dynamically exchange routing information.
- BGP can also be configured to react to topology changes beyond a customer-to-ISP link.



Connection Redundancy

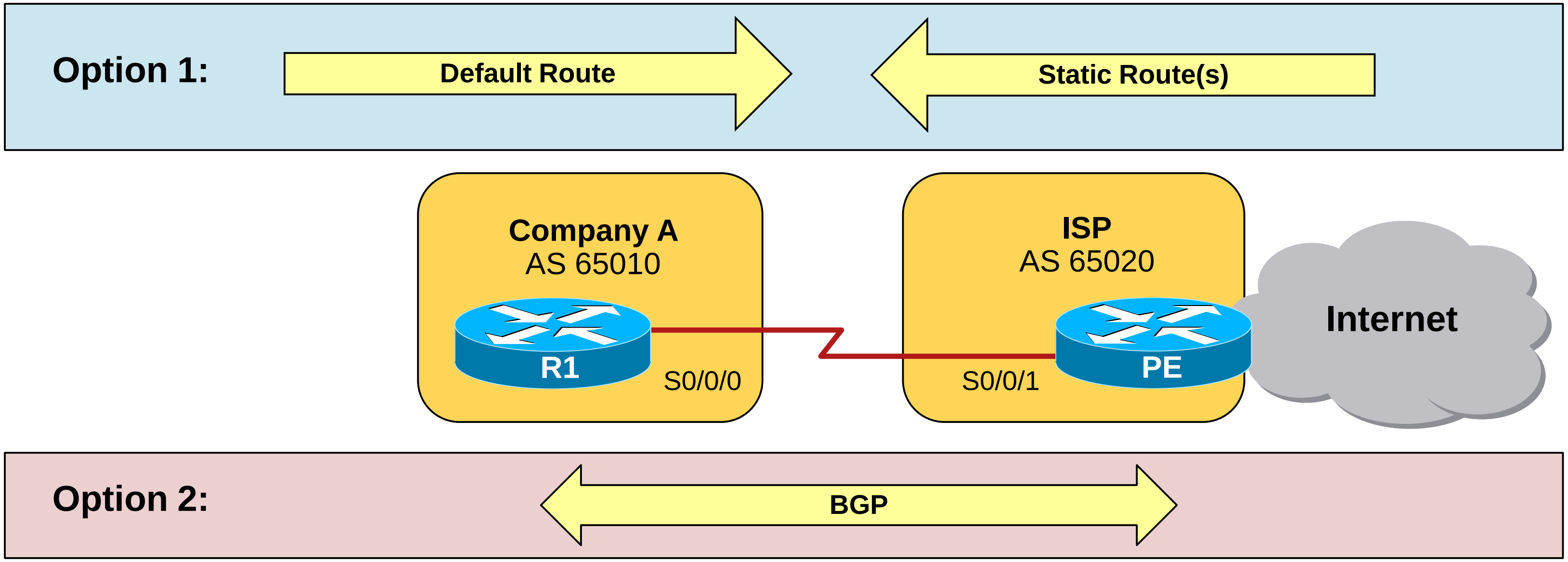
- Redundancy can be achieved by deploying redundant links, deploying redundant devices, and using redundant components within a router.
  - The ISP connection can also be made redundant.
- When a customer is connected to a single ISP the connection is referred to as *single-homed or dual-homed.*
- When a customer is connected to multiple ISPs the connection is referred to as *multihomed or dual-multihomed.*

Connection Redundancy

Connecting to One ISP	Connecting to Two or more ISPs
Single-homed  	**Multihomed** 
Dual-homed  Company A ISP Internet Option 1 Company A ISP Internet Option 2	**Dual-multihomed** 

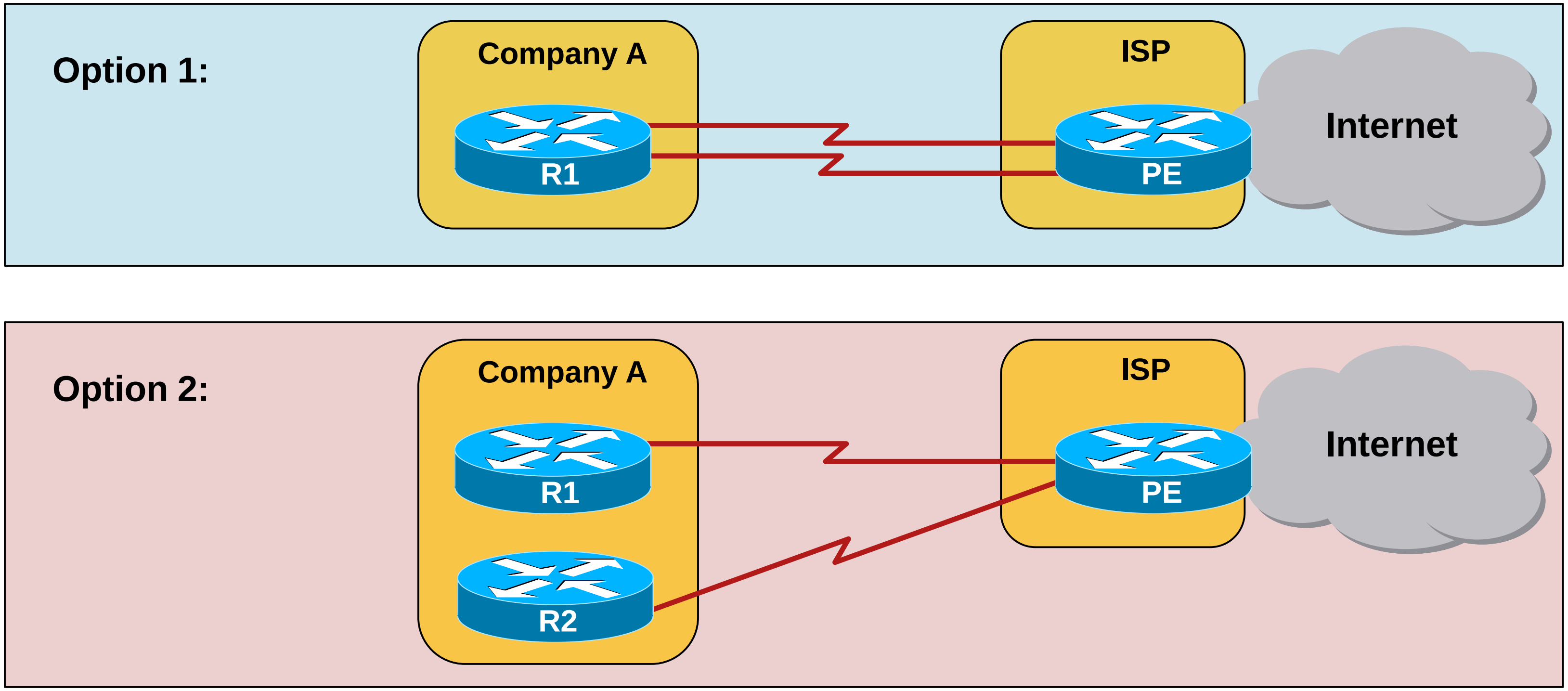
Connecting to One ISP: Single-Homed

- The connection type depends on the ISP offering (e.g., leased line, xDSL, Ethernet) and link failure results in a no Internet connectivity.
- The figure displays two options:
  - **Option 1**: Static routes are typically used with a static default route from the customer to the ISP, and static routes from the ISP toward customer networks.
  - **Option 2**: When BGP is used, the customer dynamically advertises its public networks and the ISP propagates a default route to the customer.



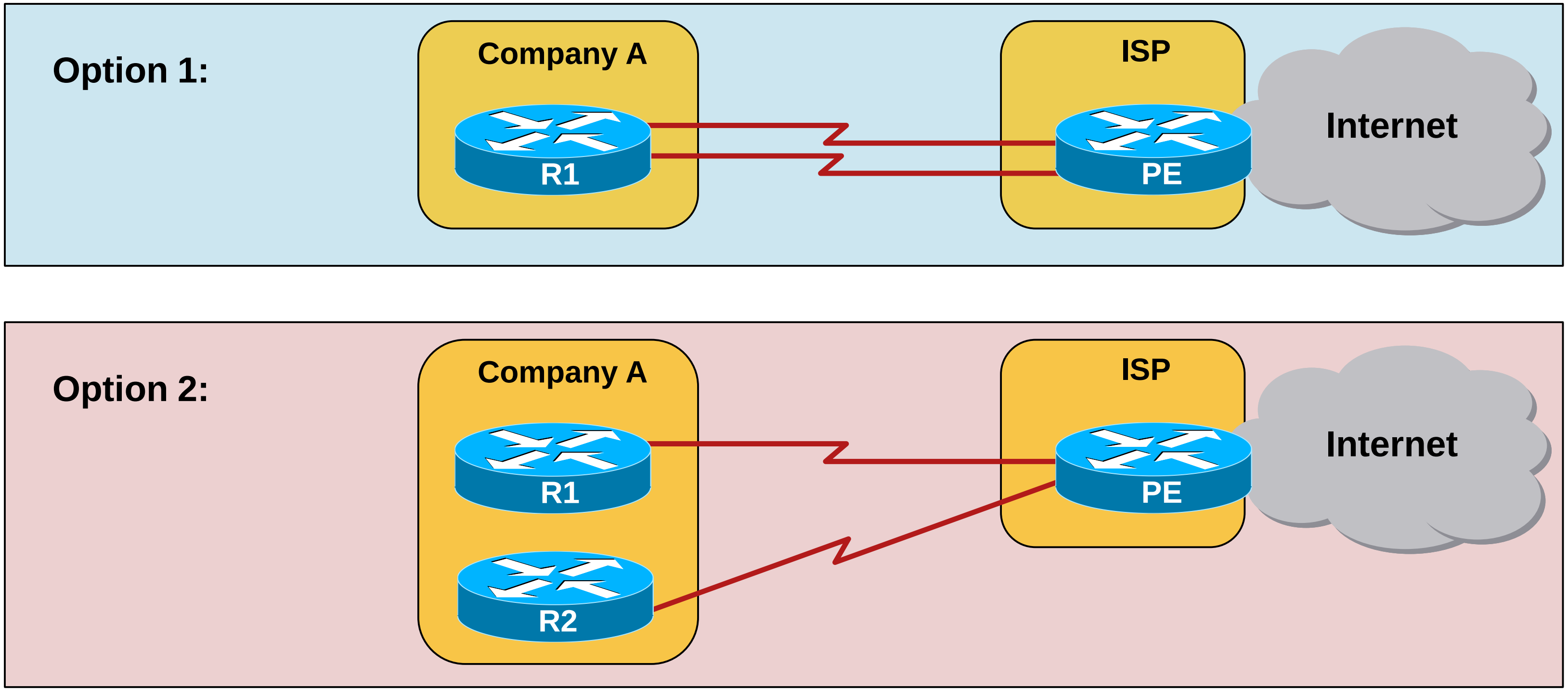
Connecting to One ISP: Dual-Homed

- The figure displays two dual-homed options:
  - **Option 1**: Both links can be connected to one customer router.
  - **Option 2**: To enhance resiliency, the two links can terminate at separate routers in the customer's network.



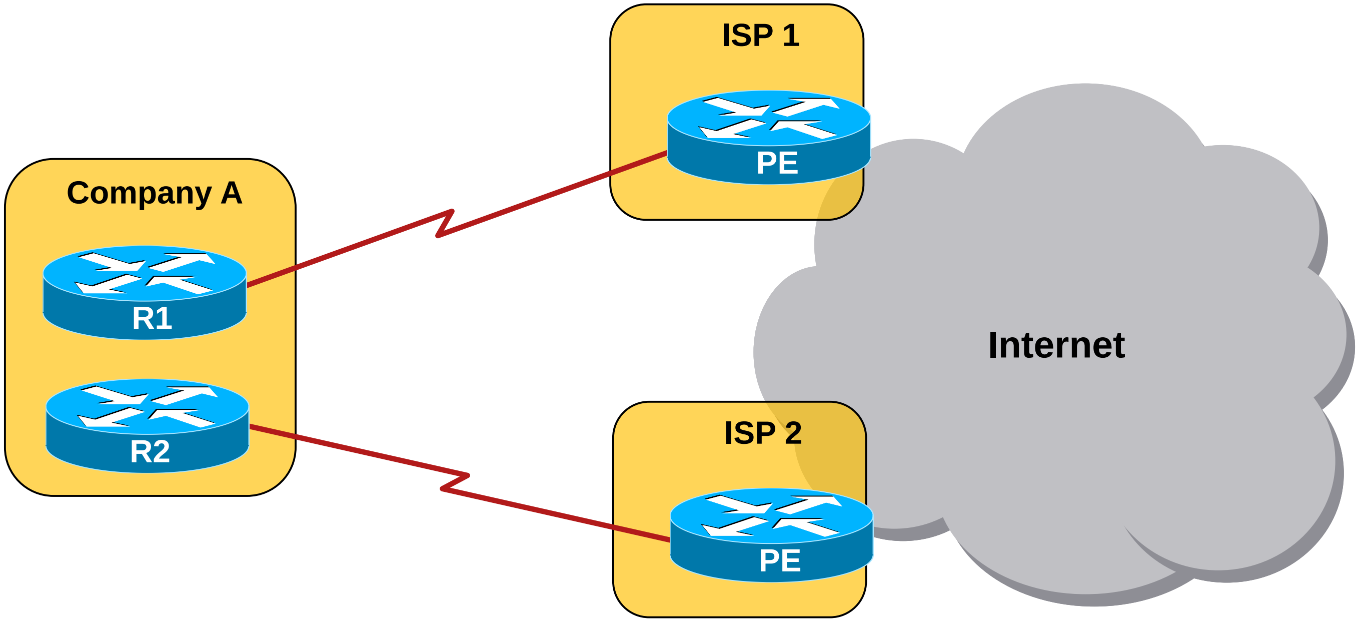
Connecting to One ISP: Dual-Homed

- Routing deployment options include:
  - Primary and backup link functionality in case the primary link fails.
  - Load sharing using Cisco Express Forwarding (CEF).
- Regardless, routing can be either static or dynamic (BGP).



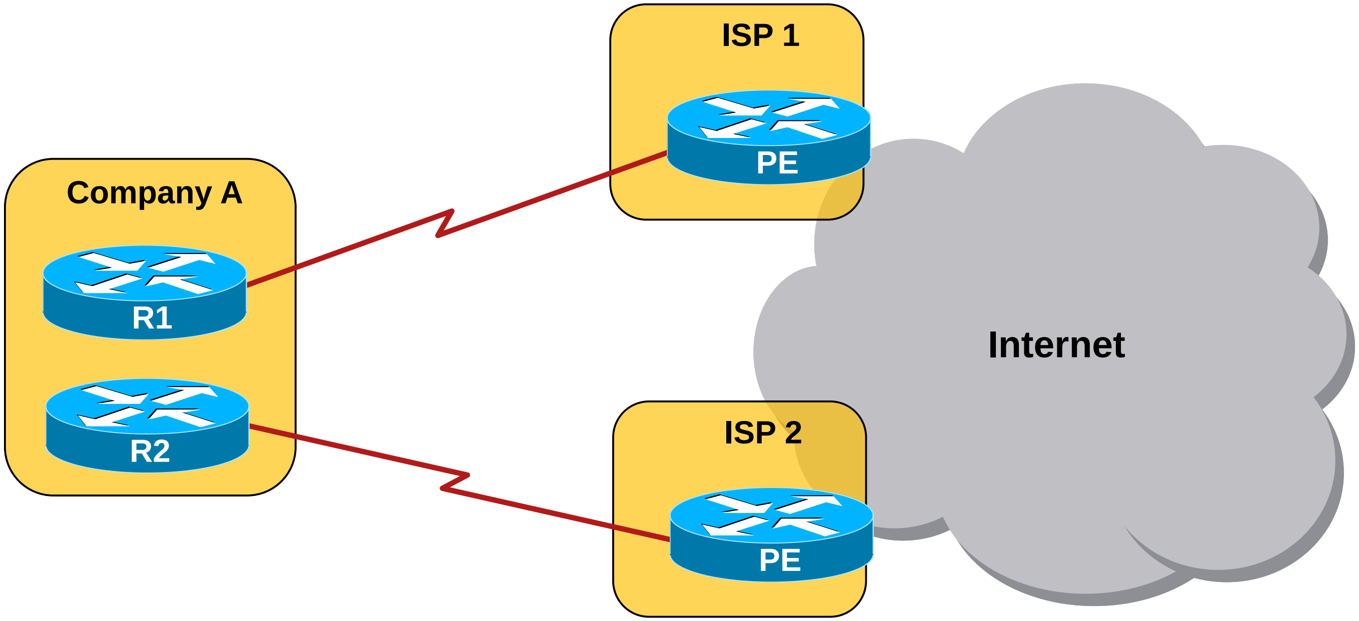
Connecting to Multiple ISPs: Multihomed

- Connections from different ISPs can terminate on the same router, or on different routers to further enhance the resiliency.
- Routing must be capable of reacting to dynamic changes therefore BGP is typically used.



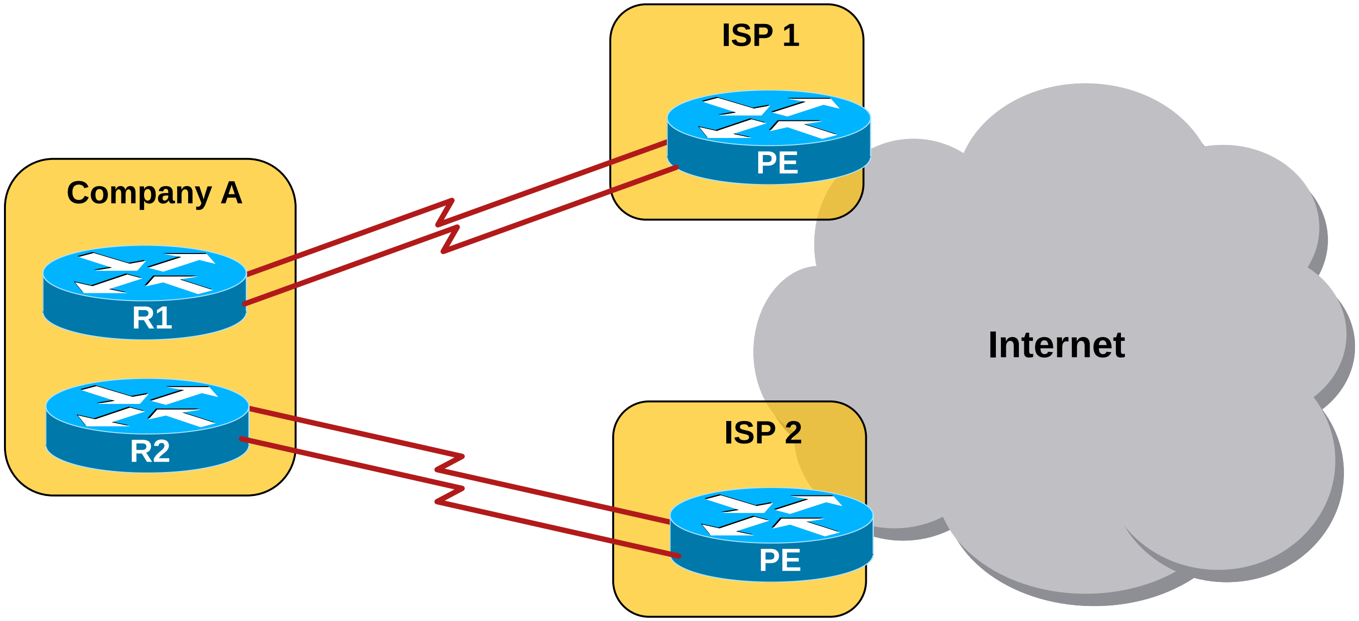
Connecting to Multiple ISPs: Multihomed

- Multihomed benefits include:
  - Achieving an ISP-independent solution.
  - Scalability of the solution, beyond two ISPs.
  - Resistance to a failure to a single ISP.
  - Load sharing for different destination networks between ISPs.



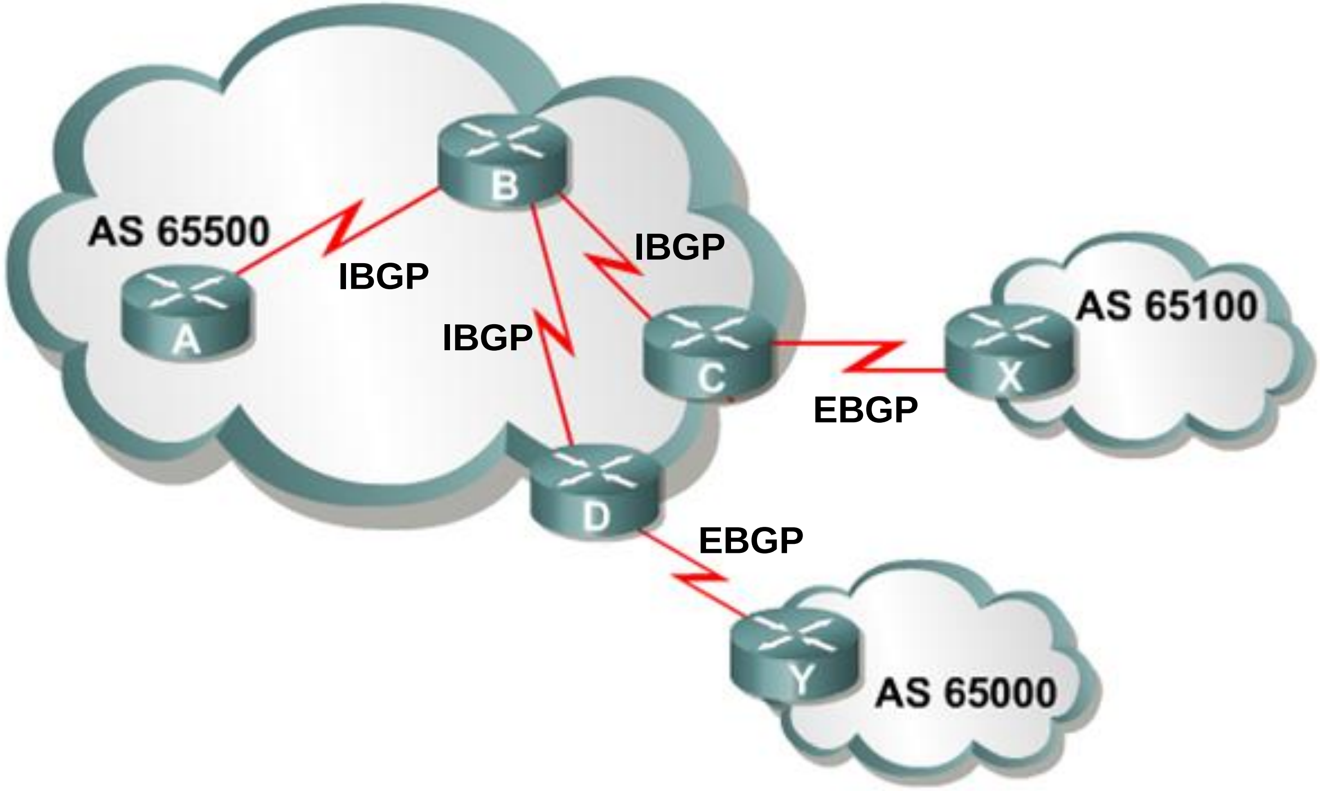
Connecting Multiple ISPs: Dual-Multihomed

- Dual multihomed includes all the benefits of multihomed connectivity, with enhanced resiliency.
- The configuration typically has multiple edge routers, one per ISP, and uses BGP.



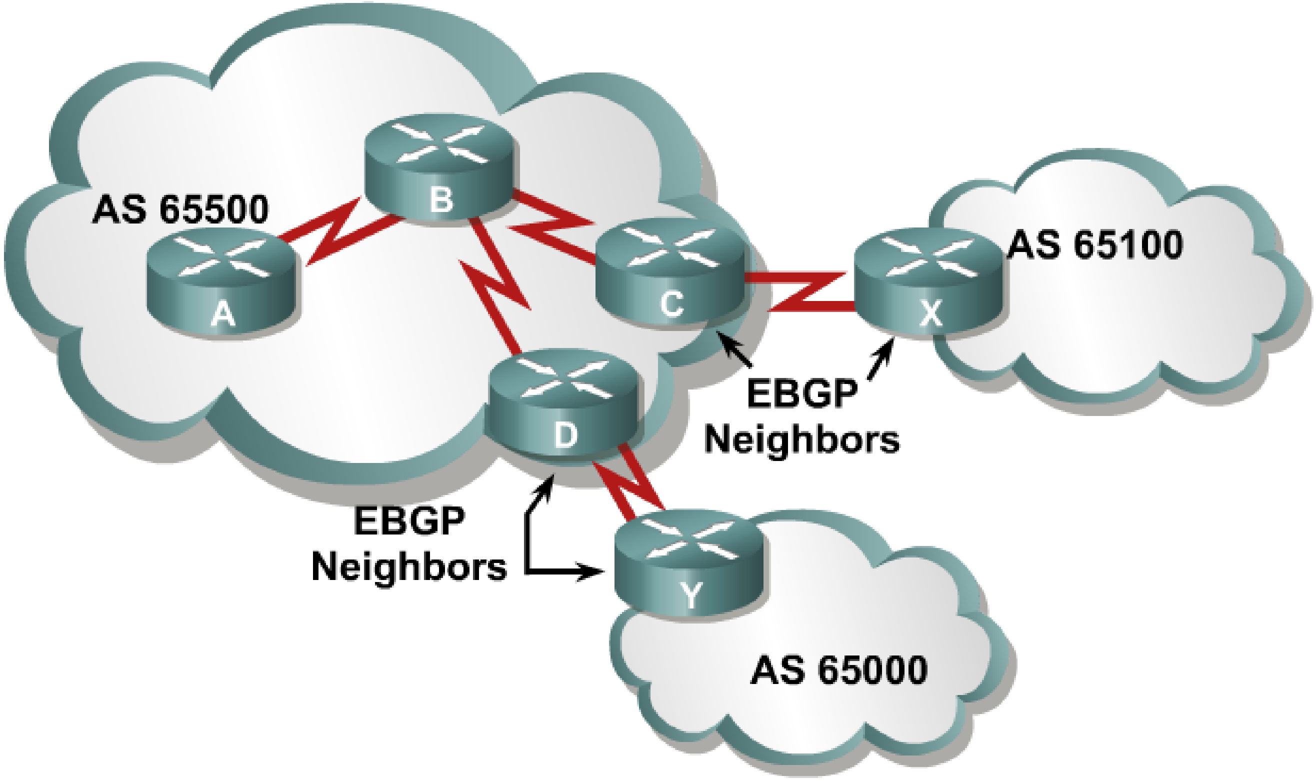
Using BGP in an Enterprise Network

- When BGP is running between routers in different AS, it is called **External BGP (EBGP)**.
- When BGP is running between routers in the same AS, it is called **Internal BGP (IBGP)**.



External BGP

- EBGP neighbors are in different autonomous systems.
  - EBGP neighbors need to be directly connected.

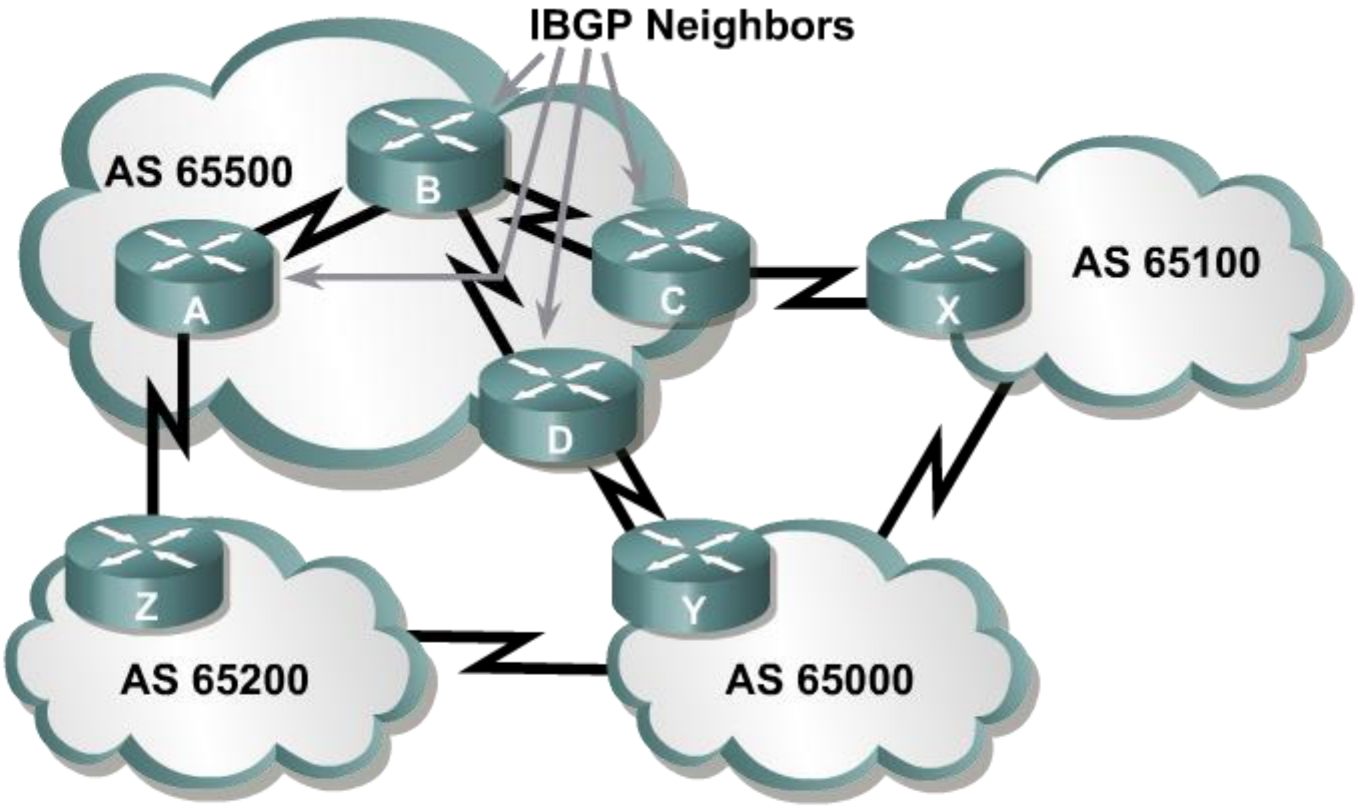


EBGP Neighbor Relationship Requirements

- Define neighbors:
  - A TCP session (three-way handshake) must be established before starting BGP routing update exchanges.
- Reachability:
  - EBGP neighbors are usually directly connected.
- Different AS number:
  - EBGP neighbors must have different AS numbers.

Internal BGP

- IBGP neighbors are in the same autonomous systems.
  - IBGP neighbors do not need to be directly connected.

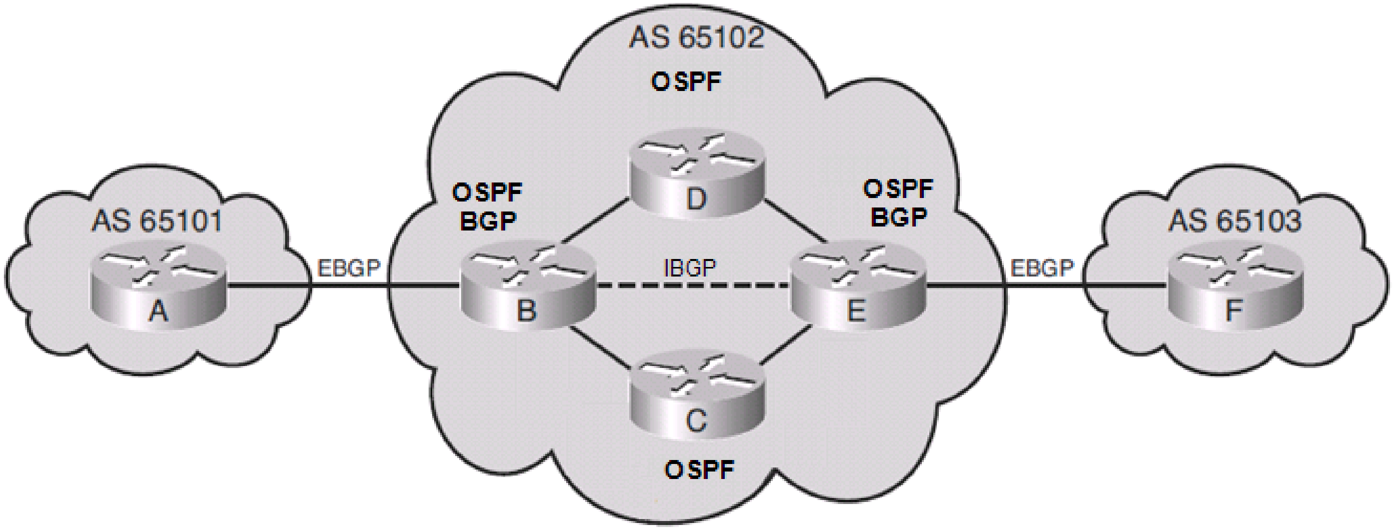


IBGP Neighbor Relationship Requirements

- **Define neighbors:**
  - A TCP session (three-way handshake) must be established before starting BGP routing update exchanges.
- **Reachability:**
  - IBGP neighbors must be reachable usually by using an IGP.
  - Loopback IP addresses are typically used to identify IBGP neighbors.
- **Same AS number:**
  - IBGP neighbors must have the same AS number.

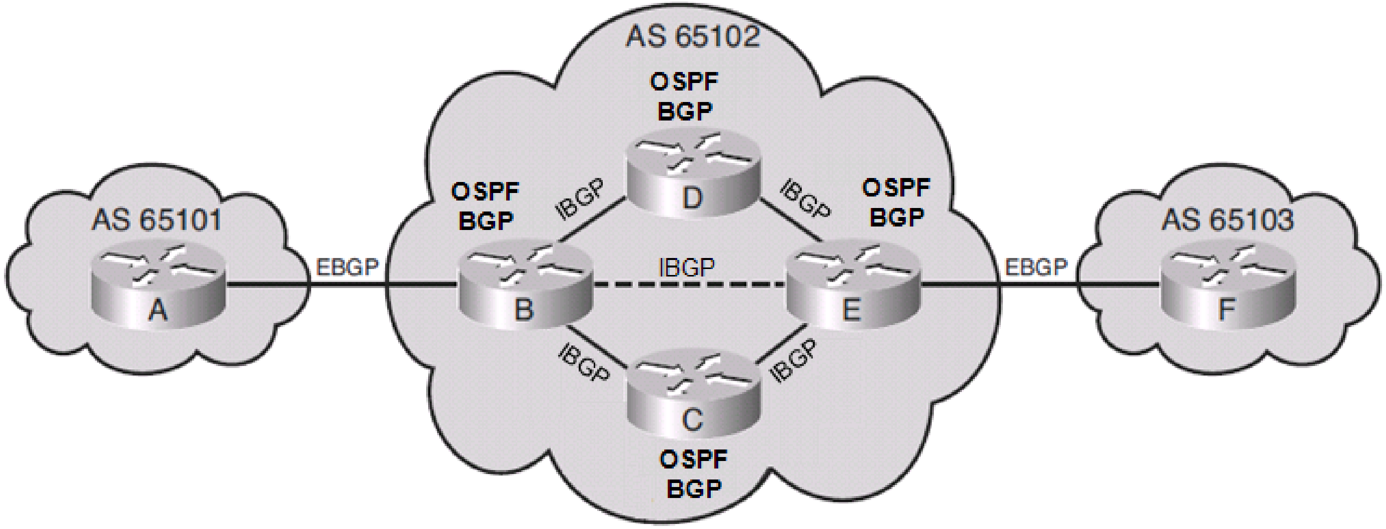
IBGP in a Transit AS

- A transit AS is an AS that routes traffic from one external AS to another external AS.
- In this example, AS 65102 is a service provider network.
  - Only the two edge routers (router B and E) are running BGP and have established an IBGP neighbor relationship using OSPF.
  - Although the EBGP routes could be redistributed into OSPF, the potential number of BGP routes may overwhelm OSPF and is therefore not recommended.



IBGP in a Transit AS

- A better solution for a provider network would be to have a fully meshed BGP internetwork.
  - BGP runs on all internal routers and all routers establish IBGP sessions.
  - IBGP routers have complete knowledge of external routes.

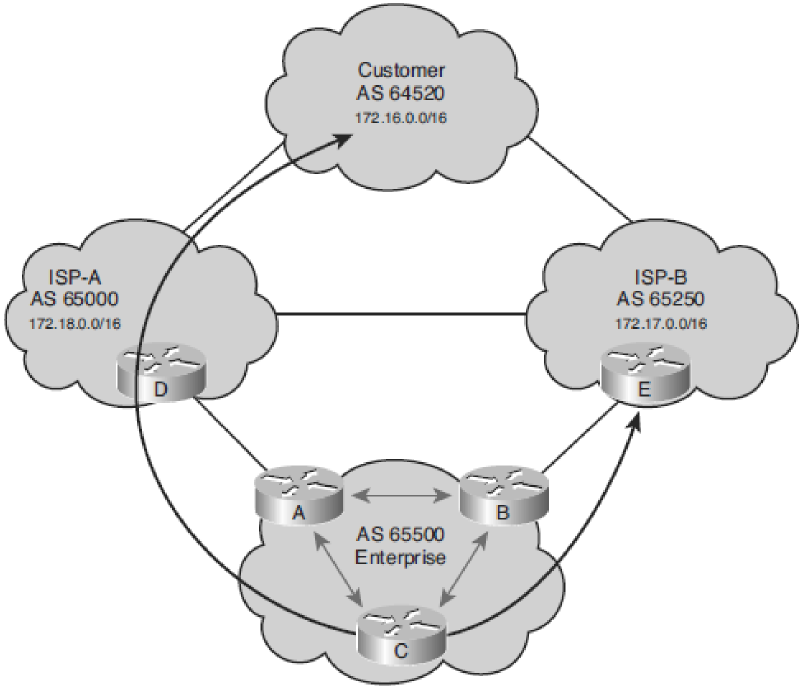


IBGP in a Nontransit AS

- A nontransit AS is an AS that does not route traffic from one external AS to another external AS.
  - Nontransit AS networks are typically enterprise networks.
- All routers in a nontransit AS must still have complete knowledge of external routes.
- To avoid routing loops within an AS, BGP specifies that routes learned through IBGP are never propagated to other IBGP peers.
  - It is assumed that the sending IBGP neighbor is fully meshed with all other IBGP speakers and has sent each IBGP neighbor the update.

BGP in an Enterprise Example

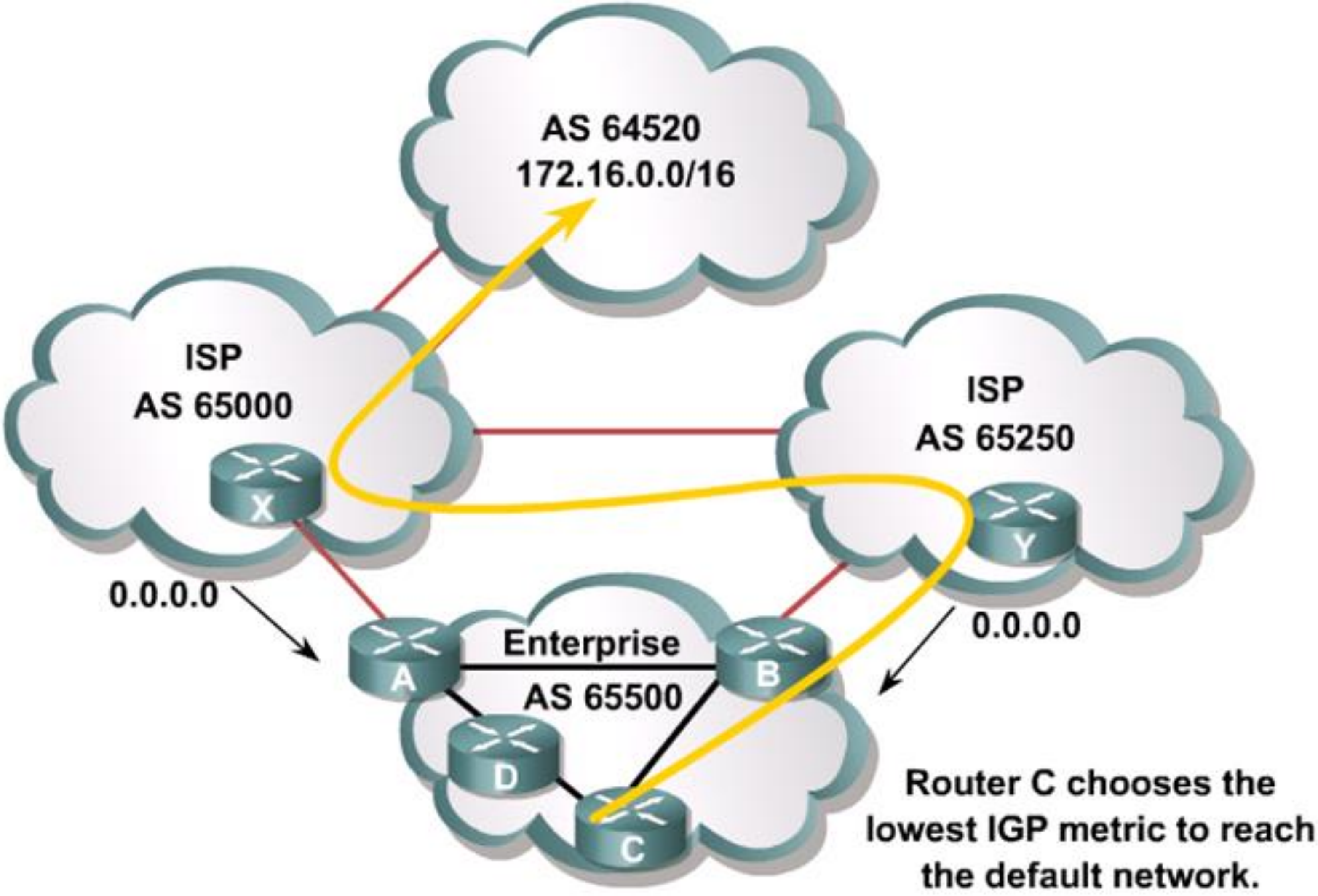
- Enterprise AS 65500 is learning routes from both ISP-A and ISP-B via EBGP and is also running IBGP on all of its routers.
  - If one of the connections to the ISPs goes down, traffic will be sent through the other ISP.
- An undesirable situation could occur if the enterprise AS is configured as a transit AS.
  - For example, AS 65500 learns the 172.18.0.0/16 route from ISP-A.
  - If router B advertises that route to ISP-B, then ISP-B may decide to use it.
  - This undesirable configuration could be avoided through careful BGP configuration.



Three Multihoming Connection Options

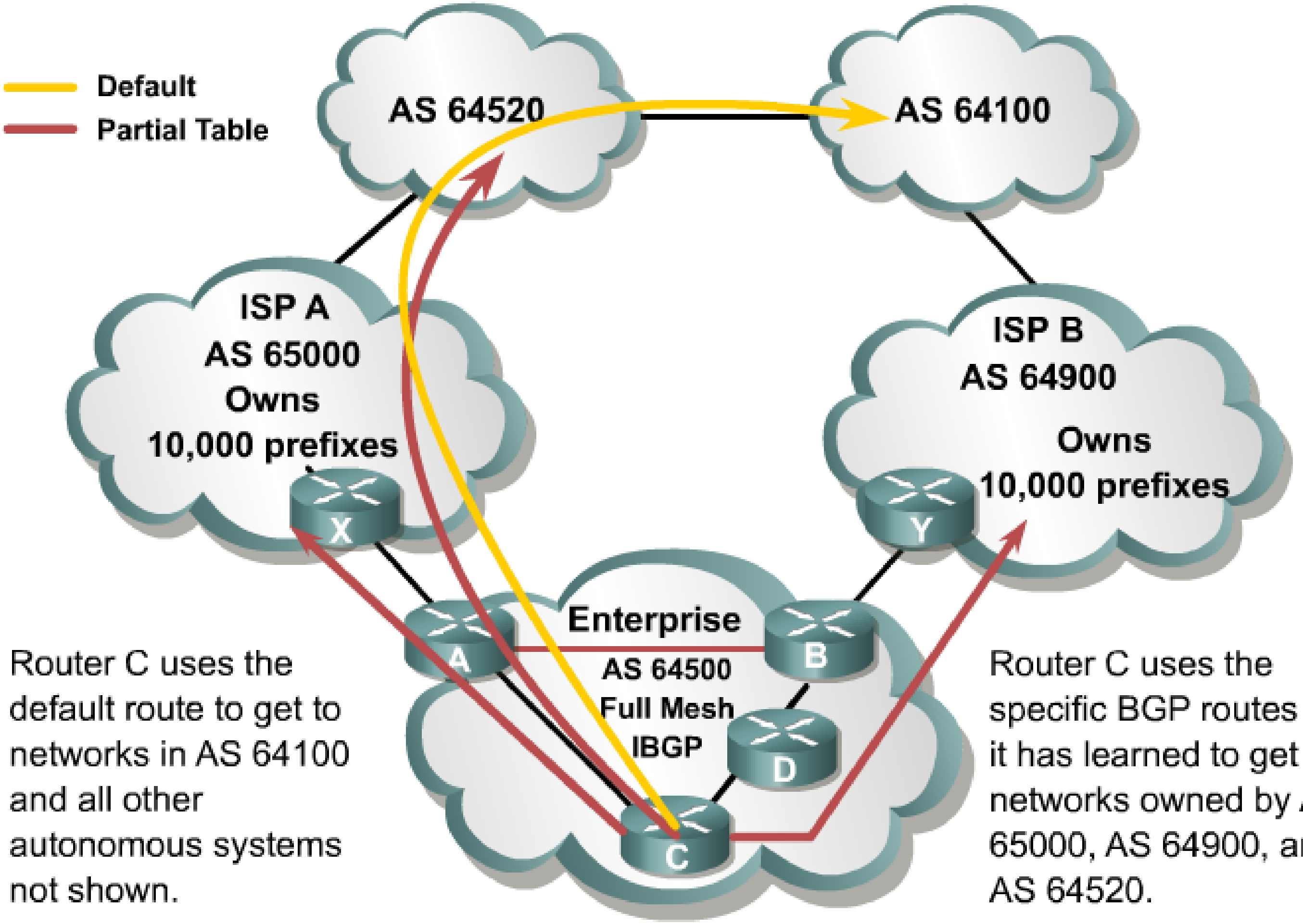
1. Each ISP passes only a default route to the AS.
   - The default route is passed on to internal routers.
2. Each ISP passes only a default route and provider-owned specific routes to the AS.
   - These routes may be propagated to internal routers, or all internal routers in the transit path can run BGP to exchange these routes.
3. Each ISP passes all routes to the AS.
   - All internal routers in the transit path run BGP to exchange these routes.

Default Routes from All Providers



Router C chooses the lowest IGP metric to reach the default network.

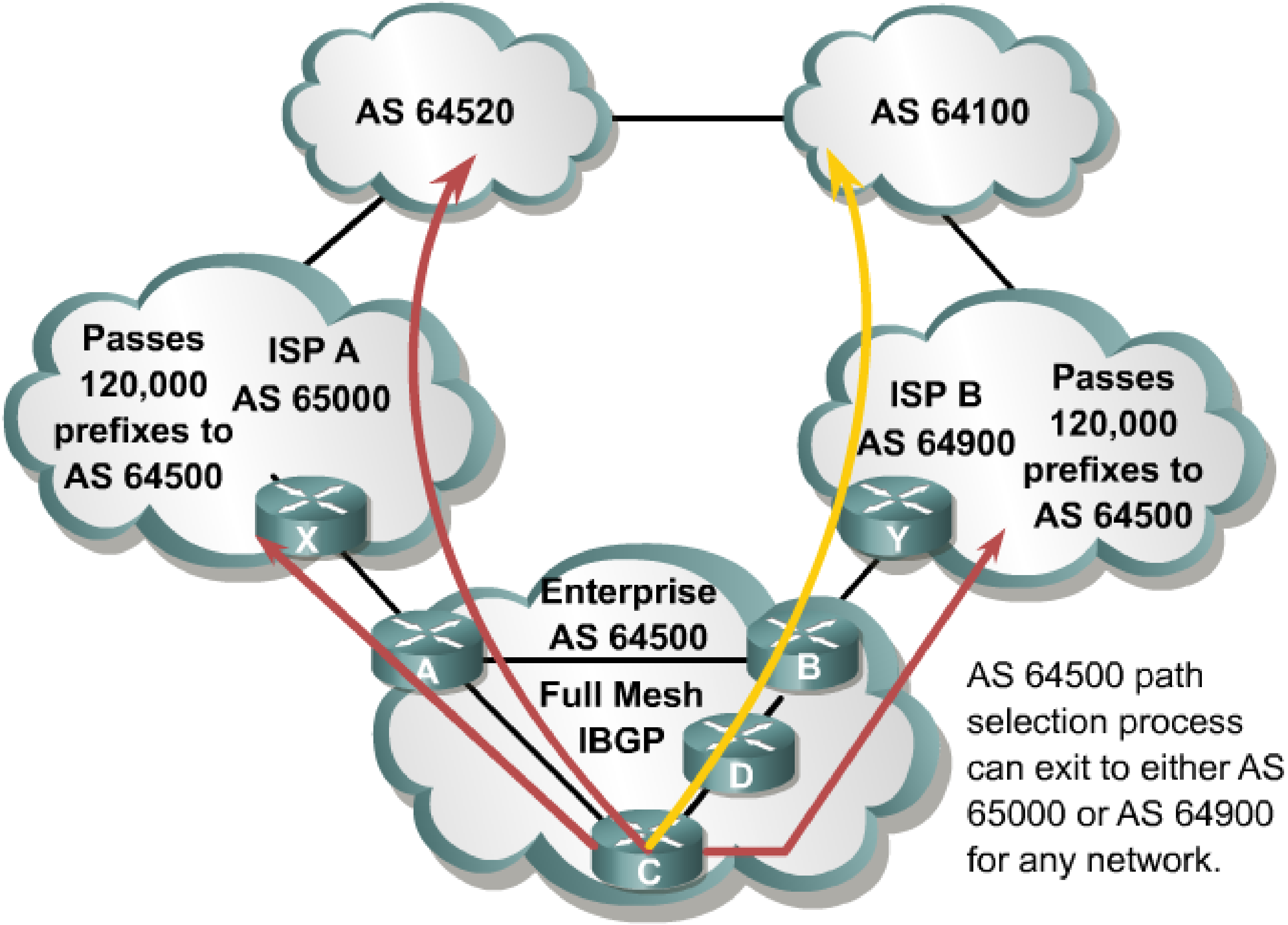
Default Routes and Partial Updates



Router C uses the default route to get to networks in AS 64100 and all other autonomous systems not shown.

Router C uses the specific BGP routes that it has learned to get to networks owned by AS 65000, AS 64900, and AS 64520.

Full Routes from All Providers



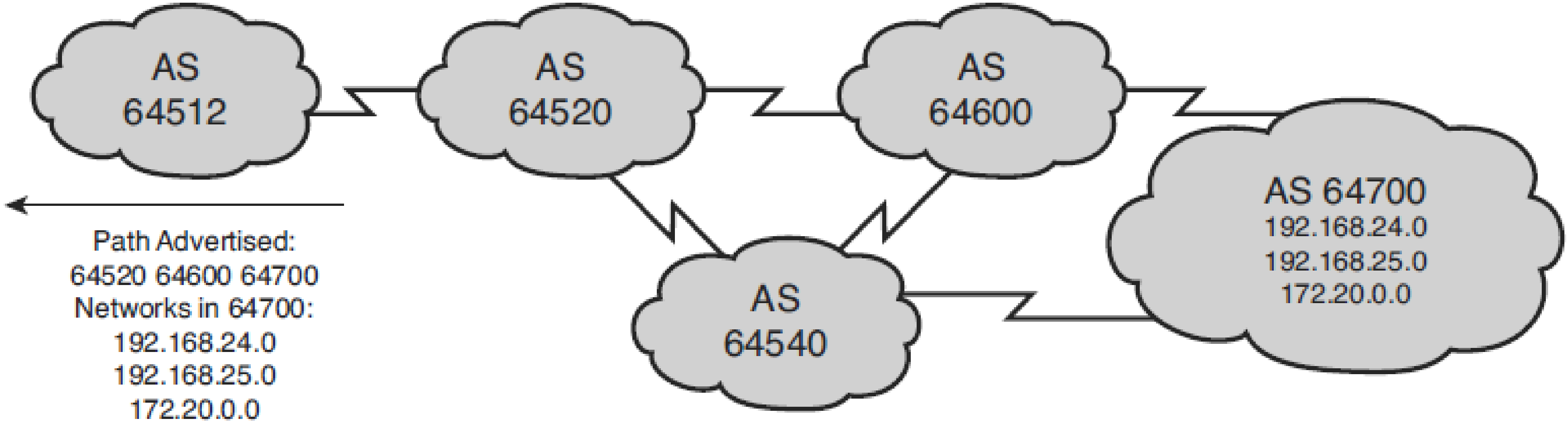
AS 64500 path selection process can exit to either AS 65000 or AS 64900 for any network.

BGP Path Vector Characteristics

- Internal routing protocols announce a list of networks and the metrics to get to each network.
- In contrast, BGP routers exchange network reachability information, called **path vectors**, made up of **path attributes**.

BGP Path Vector Characteristics

- The path vector information includes:
  - A list of the full path of BGP AS numbers (hop by hop) necessary to reach a destination network.
  - Other attributes including the IP address to get to the next AS (the next-hop attribute) and how the networks at the end of the path were introduced into BGP (the origin code attribute).



When to Use BGP

- Most appropriate when the effects of BGP are well-understood and at least one of the following conditions exists:
  - The AS has multiple connections to other autonomous systems.
  - The AS allows packets to transit through it to reach other autonomous systems (eg, it is a service provider).
  - Routing policy and route selection for traffic entering and leaving the AS must be manipulated.

When Not to Use BGP

- Do not use BGP if one or more of the following conditions exist:
  - A single connection to the Internet or another AS.
  - Lack of memory or processor power on edge routers to handle constant BGP updates.
  - You have a limited understanding of route filtering and the BGP path-selection process.
- In these cases, use static or default routes instead.

BGP Synchronization

- The BGP synchronization rule states that:
  - “*A BGP router should not use, or advertise to an external neighbor, a route learned by IBGP, unless that route is local or is learned from the IGP.*”
  - If synchronization is enabled, a router learning a route via IBGP waits until the IGP has propagated the route within the autonomous system and then advertises it to external peers.
  - With the default of synchronization disabled, BGP can use and advertise to external BGP neighbors routes learned from an IBGP neighbor that are not present in the local routing table.
- BGP synchronization is disabled by default in Cisco IOS Software Release 12.2(8)T and later.
  - It was on by default in earlier Cisco IOS Software releases.

BGP Table

- BGP keeps its own table for storing BGP information received from and sent to BGP neighbors.
  - This table is also known as the BGP table, BGP topology table, BGP topology database, BGP routing table, and the BGP forwarding database.
- The router offers the best routes from the BGP table to the IP routing table.

BGP Tables

- 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

- There are four different BGP message types:

Open Message

Octets	16	2	1	1	2	2	4	1	7
	Marker	Length	Type	Version	AS	Hold Time	BGP ID	Optional Length	Optional

Update Message

Octets	16	2	1	2	Variable	2	Variable	Variable
	Marker	Length	Type	Unfeasible Routes length	Withdrawn Routes	Attribute Length	Attributes	NLRI

Notification Message

Octets	16	2	1	1	1	Variable
	Marker	Length	Type	Error Code	Error Sub-code	Diagnostic Data

Keepalive Message

Octets	16	2	1
	Marker	Length	Type

BGP Message Header

- All messages begin with the same 3 field headers

Open Message

Octets	16	2	1	1	2	2	4	1	7
	Marker	**Length**	**Type**	Version	AS	Hold Time	BGP ID	Optional Length	Optional

Update Message

Octets	16	2	1	2	Variable	2	Variable	Variable
	Marker	**Length**	**Type**	Unfeasible Routes length	Withdrawn Routes	Attribute Length	Attributes	NLRI

Notification Message

Octets	16	2	1	1	1	Variable
	Marker	**Length**	**Type**	Error Code	Error Sub-code	Diagnostic Data

Keepalive Message

Octets	16	2	1
	Marker	**Length**	**Type**

Open Message

- Once a TCP connection has been established, the Open message is sent and includes a set of parameters that have to be agreed upon before a full BGP adjacency can be established.
- Once both BGP peers have agreed upon mutual capabilities, they can start exchanging routing information by means of BGP Update messages.

Open Message

Octets	16	2	1	1	2	2	4	1	7
	Marker	Length	Type	Version	AS	Hold Time	BGP ID	Optional Length	Optional

Update Message

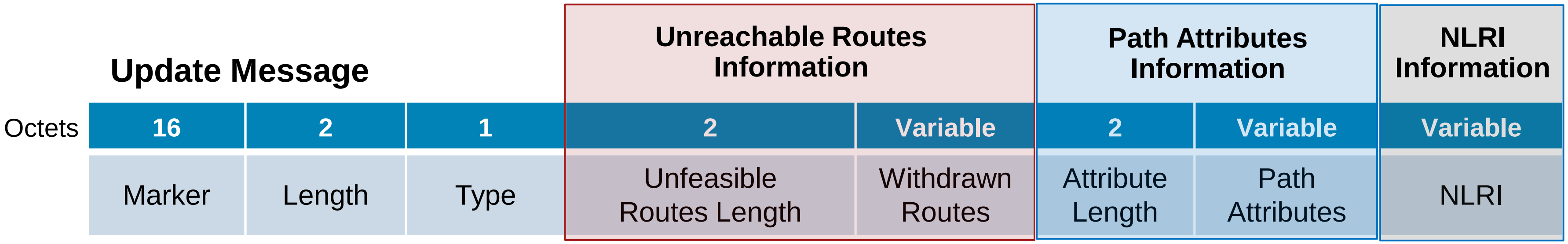
- Update messages contain all the information BGP uses to construct a loop-free picture of the internetwork.
- A BGP update message has information on one path only; multiple paths require multiple update messages.
  - All the attributes in the update message refer to that path, and the networks are those that can be reached through it.

Update Message

Octets	16	2	1	2	Variable	2	Variable	Variable
	Marker	Length	Type	Unfeasible Routes Length	Withdrawn Routes	Attribute Length	Path Attributes	NLRI

Update Message

- An update message includes the following information:
  - Unreachable routes information
  - Path attribute information
  - Network-layer reachability information (NLRI)
    - This field contains a list of IP address prefixes that are reachable by this path.



Update Message

				Unreachable Routes Information		Path Attributes Information		NLRI Information
Octets	16	2	1	2	Variable	2	Variable	Variable
	Marker	Length	Type	Unfeasible Routes Length	Withdrawn Routes	Attribute Length	Path Attributes	NLRI

NLRI format

- The NLRI is a list of <**length**, **prefix**> tuples.
  - One tuple for each reachable destination.
  - The **prefix** represents the reachable destination
  - The prefix **length** represents the # of bits set in the subnet mask.

IP Address Subnet Mask	NLRI
10.1.1.0 255.255.255.0	**24**, **10.1.1.0**
192.24.160.0 255.255.224.0	**19**, **192.24.160.0**

Notification Message

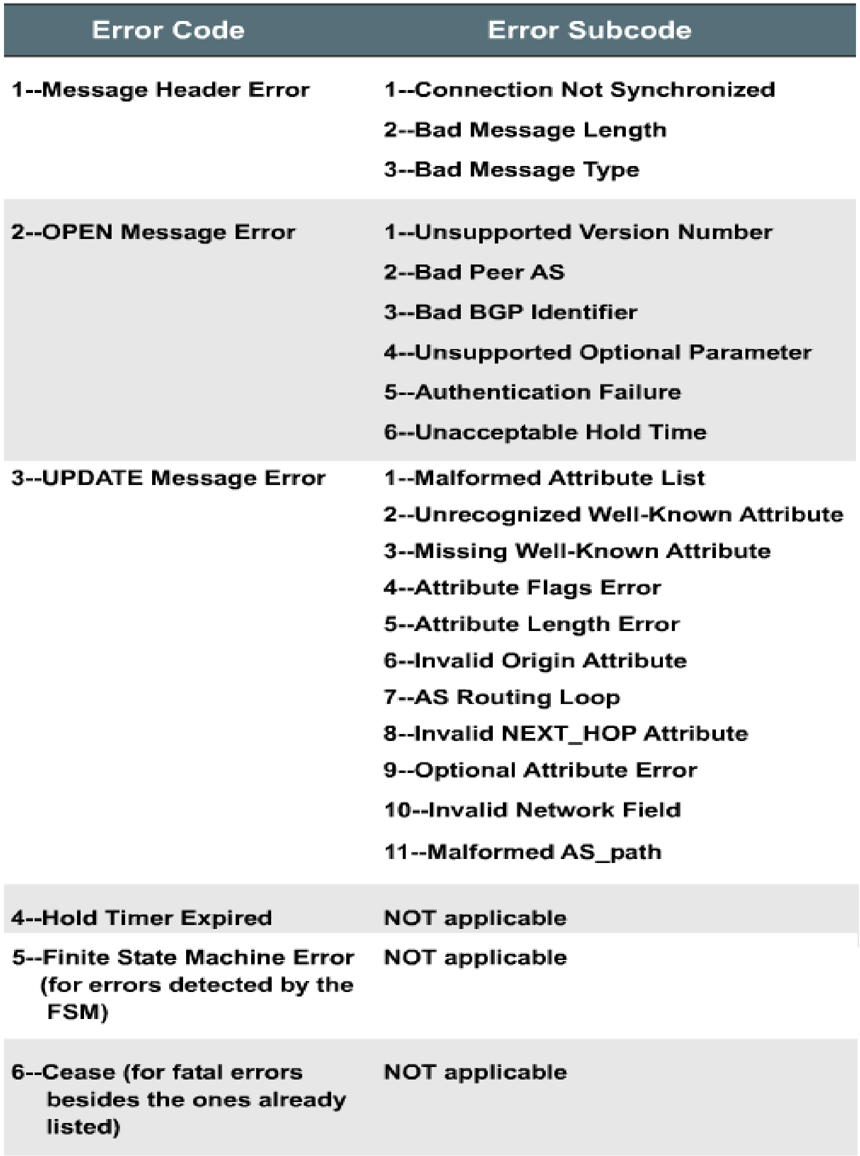
- A BGP notification message is sent when an error condition is detected.
  - The BGP connection is closed immediately after this is sent.
- Notification messages include an error code, an error subcode, and data related to the error.

Notification Message

Octets	16	2	1	1	1	Variable
	Marker	Length	Type	Error Code	Error Sub-code	Diagnostic Data

Notification Message

- Sample error codes and their associated subcodes.



Error Code	Error Subcode
1--Message Header Error	1--Connection Not Synchronized 2--Bad Message Length 3--Bad Message Type
2--OPEN Message Error	1--Unsupported Version Number 2--Bad Peer AS 3--Bad BGP Identifier 4--Unsupported Optional Parameter 5--Authentication Failure 6--Unacceptable Hold Time
3--UPDATE Message Error	1--Malformed Attribute List 2--Unrecognized Well-Known Attribute 3--Missing Well-Known Attribute 4--Attribute Flags Error 5--Attribute Length Error 6--Invalid Origin Attribute 7--AS Routing Loop 8--Invalid NEXT_HOP Attribute 9--Optional Attribute Error 10--Invalid Network Field 11--Malformed AS_path
4--Hold Timer Expired	NOT applicable
5--Finite State Machine Error (for errors detected by the FSM)	NOT applicable
6--Cease (for fatal errors besides the ones already listed)	NOT applicable

Keepalive Message Type

- Keepalive messages are sent between peers every 60 seconds (by default) to maintain connections.
- The message consists of only a message header (19 bytes).
    - Hold time is three times the KEEPALIVE timer of 60 seconds.
    - If the periodic timer = 0, no keepalives are sent.
    - Recommended keepalive interval is one-third of the hold time interval.

Keepalive Message

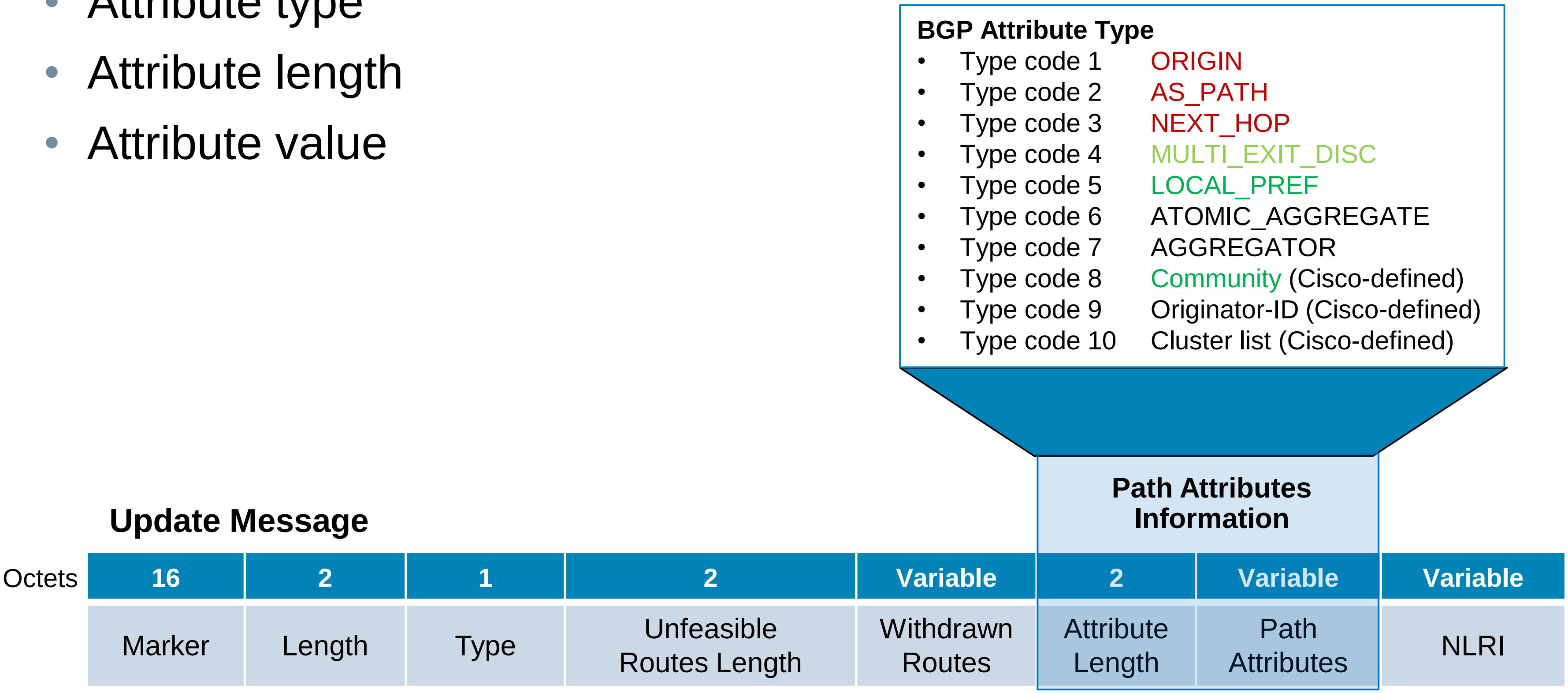
Octets	16	2	1
	Marker	**Length**	**Type**

Path Attributes

- Path attributes are a set of BGP metrics describing the path to a network (route).
  - BGP uses the path attributes to determine the best path to the networks.
  - Some attributes are mandatory and automatically included in update messages while others are manually configurable.
- BGP attributes can be used to enforce a routing policy.
- Configuring BGP attributes provides administrators with many more path control options.
  - E.g., filter routing information, prefer certain paths, customize BGP's behavior.

Path Attributes

- A BGP update message includes a variable-length sequence of path attributes describing the route.
- A path attribute consists of three fields:
  - Attribute type
  - Attribute length
  - Attribute value



BGP Attribute Type

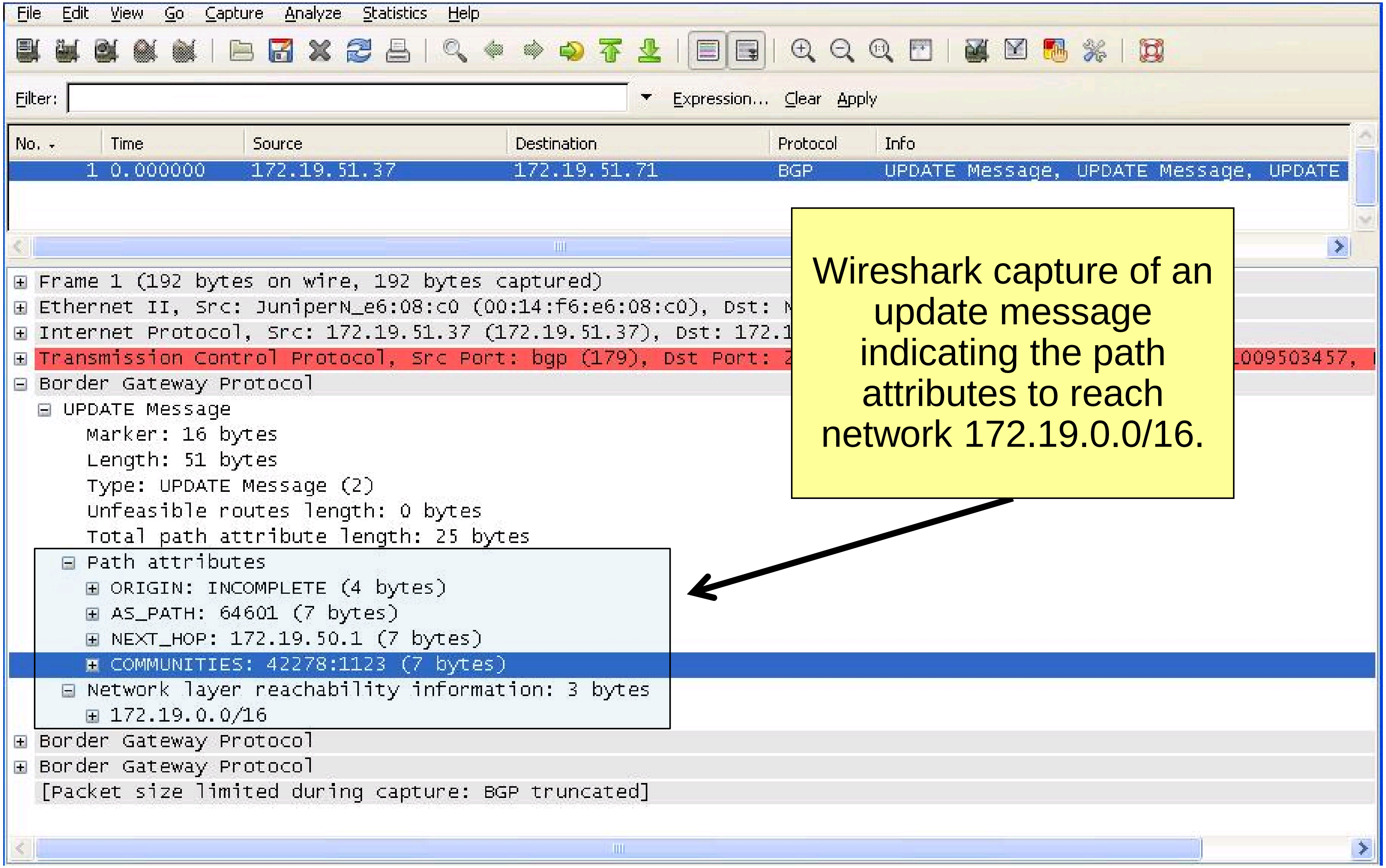
Type code 1	ORIGIN
Type code 2	AS_PATH
Type code 3	NEXT_HOP
Type code 4	MULTI_EXIT_DISC
Type code 5	LOCAL_PREF
Type code 6	ATOMIC_AGGREGATE
Type code 7	AGGREGATOR
Type code 8	Community (Cisco-defined)
Type code 9	Originator-ID (Cisco-defined)
Type code 10	Cluster list (Cisco-defined)

Path Attributes Information

Update Message

Octets	16	2	1	2	Variable	2	Variable	Variable
	Marker	Length	Type	Unfeasible Routes Length	Withdrawn Routes	Attribute Length	Path Attributes	NLRI

Path Attributes Within Update Message



File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No.   Time       Source         Destination    Protocol  Info
    1 0.000000   172.19.51.37   172.19.51.71   BGP       UPDATE Message, UPDATE Message, UPDATE
```

```
⊞ Frame 1 (192 bytes on wire, 192 bytes captured)
⊞ Ethernet II, Src: JuniperN_e6:08:c0 (00:14:f6:e6:08:c0), Dst: N
⊞ Internet Protocol, Src: 172.19.51.37 (172.19.51.37), Dst: 172.1
⊞ Transmission Control Protocol, Src Port: bgp (179), Dst Port: 2                 009503457,
⊟ Border Gateway Protocol
  ⊟ UPDATE Message
      Marker: 16 bytes
      Length: 51 bytes
      Type: UPDATE Message (2)
      Unfeasible routes length: 0 bytes
      Total path attribute length: 25 bytes
    ⊟ Path attributes
      ⊞ ORIGIN: INCOMPLETE (4 bytes)
      ⊞ AS_PATH: 64601 (7 bytes)
      ⊞ NEXT_HOP: 172.19.50.1 (7 bytes)
      ⊞ COMMUNITIES: 42278:1123 (7 bytes)
    ⊟ Network layer reachability information: 3 bytes
      ⊞ 172.19.0.0/16
⊞ Border Gateway Protocol
⊞ Border Gateway Protocol
  [Packet size limited during capture: BGP truncated]
```

Wireshark capture of an update message indicating the path attributes to reach network 172.19.0.0/16.

Attributes

- Some attributes are mandatory and automatically included in update messages while others are manually configurable.

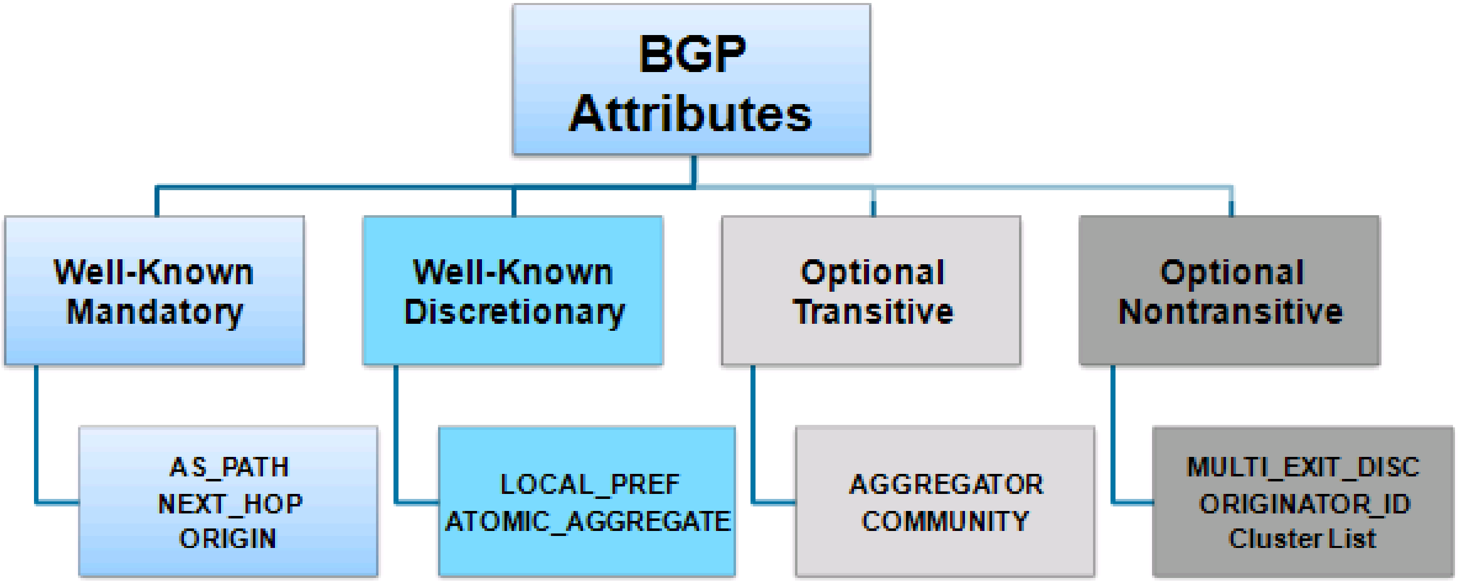
Attribute	EBGP	IBGP
AS_PATH	Well-known Mandatory	Well-known Mandatory
NEXT_HOP	Well-known Mandatory	Well-known Mandatory
ORIGIN	Well-known Mandatory	Well-known Mandatory
LOCAL_PREF	Not allowed	Well-known Discretionary
ATOMIC_AGGREGATE	Well-known Discretionary	Well-known Discretionary
AGGREGATOR	Optional Transitive	Optional Transitive
COMMUNITY	Optional Transitive	Optional Transitive
MULTI_EXIT_DISC	Optional Nontransitive	Optional Nontransitive

Automatically included in update message

Can be configured to help provide path control.

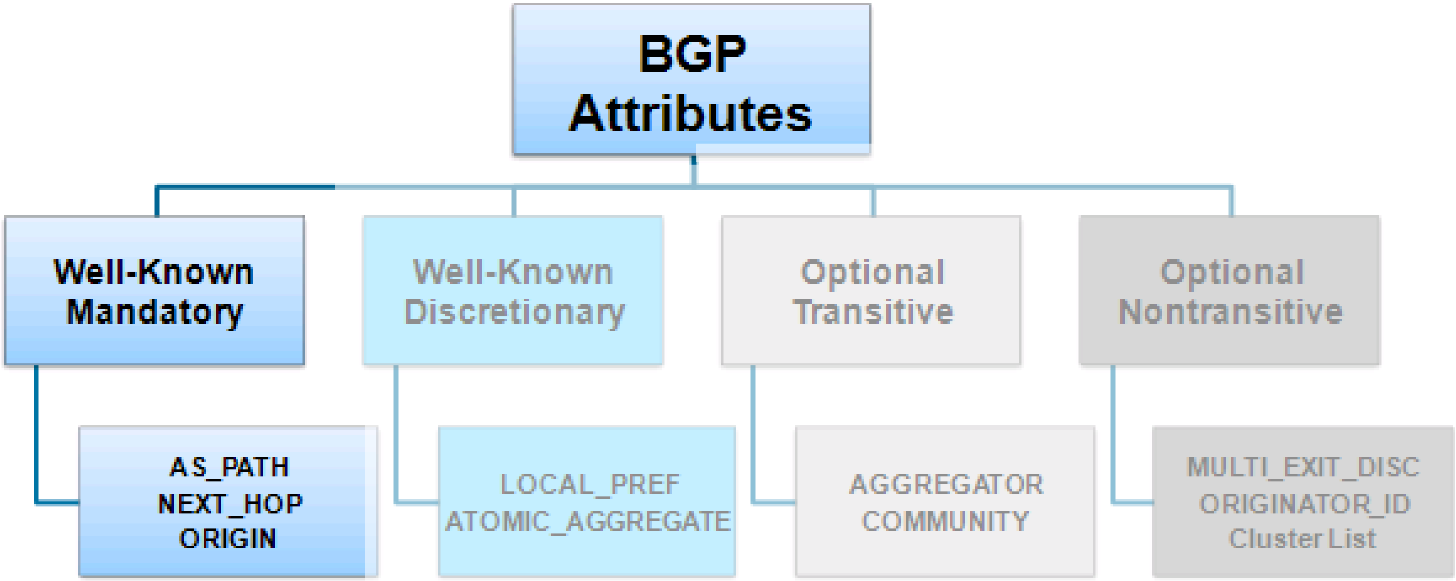
Path Attributes

- There are four different attribute types.
  - Not all vendors recognize the same BGP attributes.



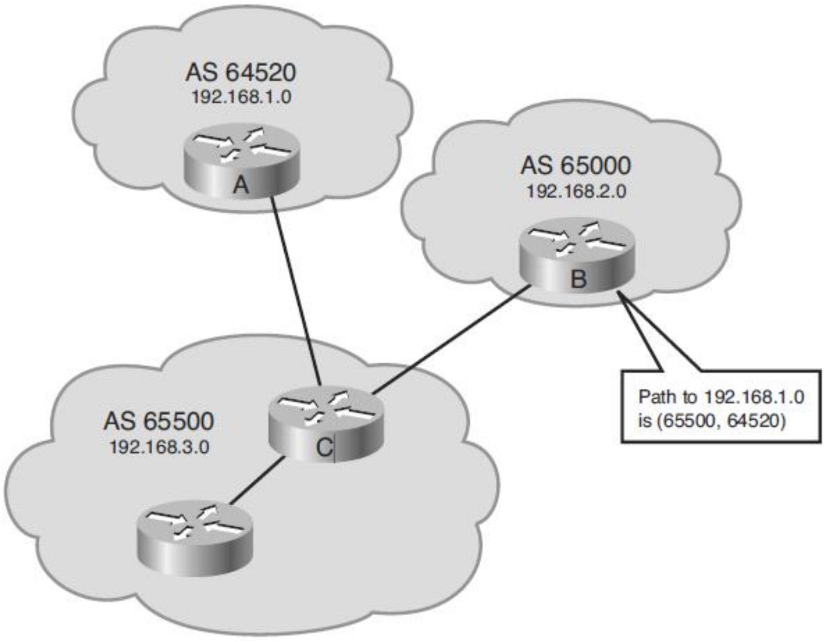
Well-Known Mandatory

- Attribute is recognized by all implementations of BGP and must appear in a BGP update message.
  - If missing, a notification error will be generated.
- Well-known mandatory attributes ensures that all BGP implementations agree on a standard set of attributes.



Well-Known Mandatory: AS_PATH

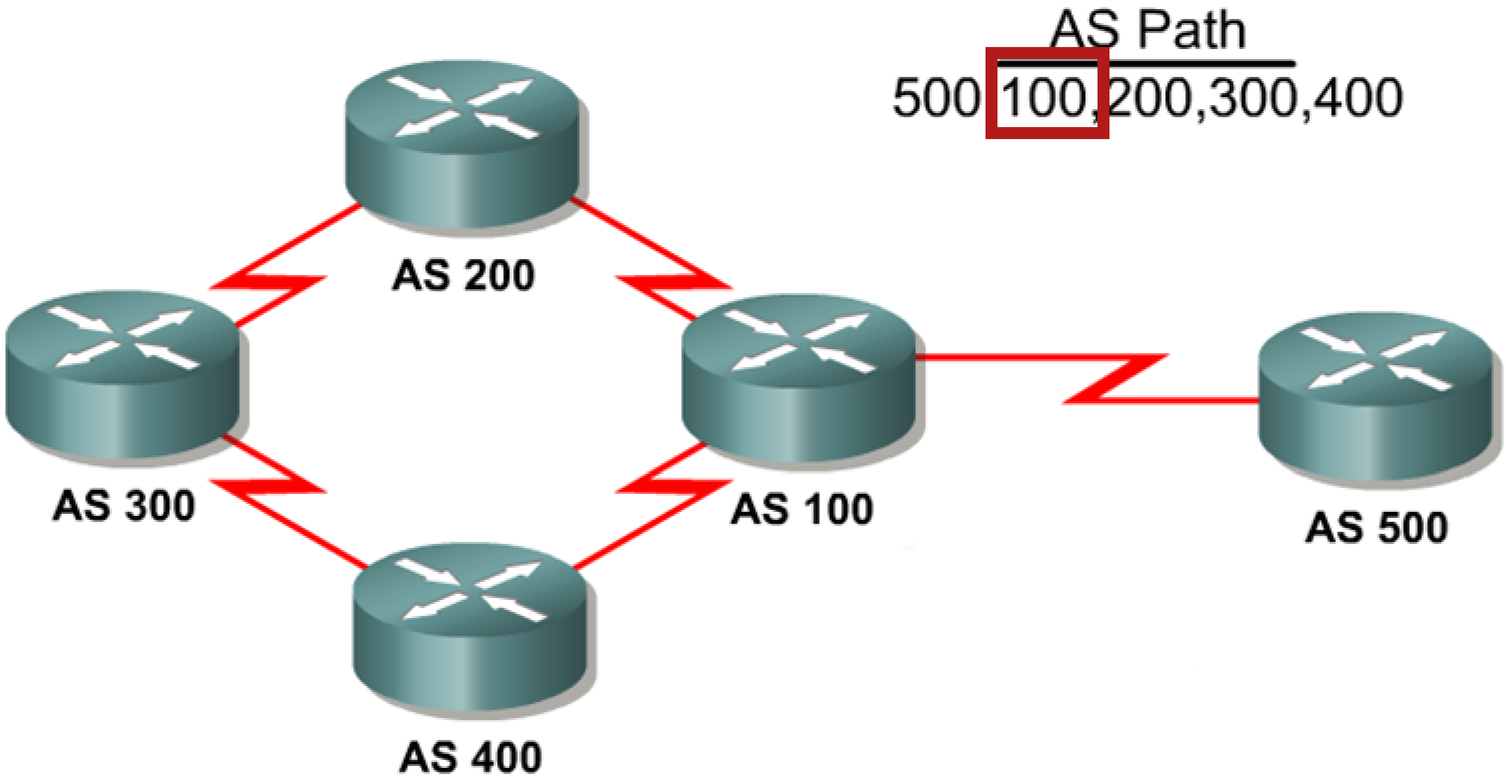
- The AS_PATH attribute contains a list of AS numbers to reach a route.
- Whenever a route update passes through an AS, the AS number is added to the beginning of the AS_PATH attribute before it is advertised to the next EBGP neighbor.



Path to 192.168.1.0 is (65500, 64520)

Well-Known Mandatory: AS_PATH

- BGP always includes the AS_PATH attribute in its update.



Well-Known Mandatory: NEXT_HOP

- The NEXT_HOP attribute indicates the IP address that is to be used to reach a destination.
- The IP address is the entry point of the next AS along the path to that destination network.
  - Therefore, for EBGP, the next-hop address is the IP address of the neighbor that sent the update.

Well-Known Mandatory: ORIGIN

- The ORIGIN attribute defines the origin of the path which could be:
  - **IGP**:
    - The route is interior to the originating AS and normally occurs when a **`network`** command is used to advertise the route via BGP.
    - An origin of IGP is indicated with an “**i**” in the BGP table.
  - **EGP**:
    - (Obsolete) The route is learned via EGP which is considered a historic routing protocol and is not supported on the Internet.
    - An origin of EGP is indicated with an “**e**” in the BGP table.
  - **Incomplete**:
    - The route’s origin is unknown or is learned via some other means and usually occurs when a route is redistributed into BGP.
    - An incomplete origin is indicated with a “**?**” in the BGP table.

Well-Known Mandatory: ORIGIN

```
R1# show ip bgp
BGP table version is 24, local router ID is 172.16.1.2
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
*> 192.208.10.0     192.208.10.5             0               0  300 i
*> 172.16.1.0       0.0.0.0                  0           32768  i
<output omitted>
```

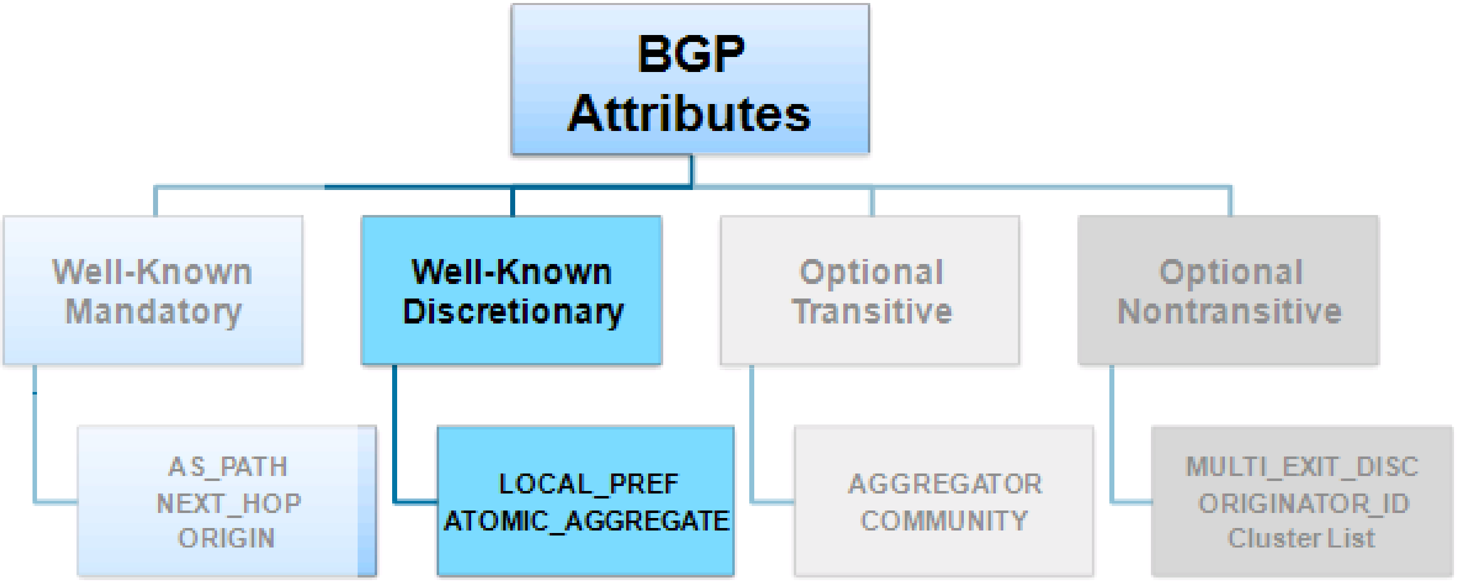
i = Route generated by the **network** command.

```
R1# show ip bgp
<output omitted>
   Network          Next Hop                Metric  LocPrf  Weight  Path
*> 10.1.1.0/24       0.0.0.0                     0           32768  ?
*> 192.168.1.0/24   10.1.1.2                    84           32768  ?
*> 192.168.2.0/24   10.1.1.2                    74           32768  ?
<output omitted>
```

? = Route generated by unknown method (usually redistributed).

Well-Known Discretionary

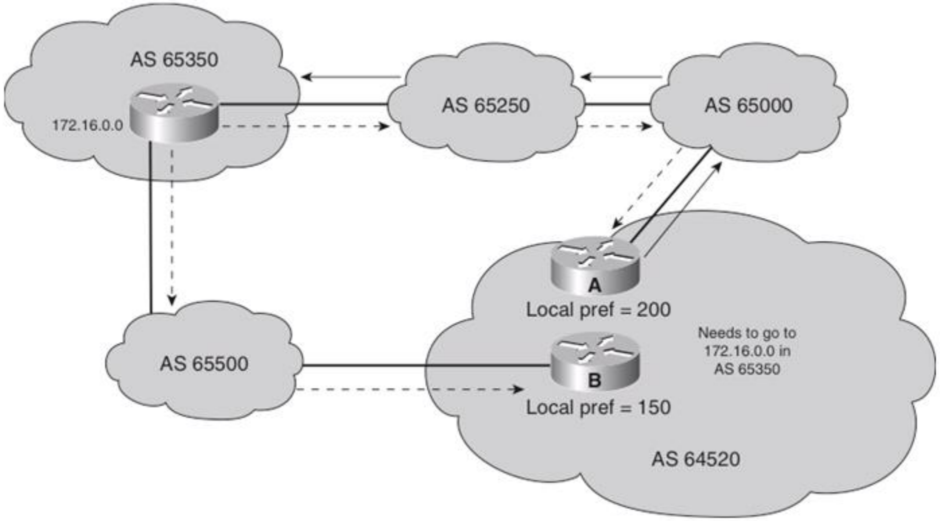
- Attribute is recognized by all implementations of BGP but may not be sent in the BGP update message.



Well-Known Discretionary: LOCAL_PREF

- The Local Preference attribute provides an indication to the “local” routers in the AS about which path is preferred to exit the AS.
  - A path with a higher local preference is preferred.
  - The default value for local preference on a Cisco router is 100.
- It is configured on a router and exchanged between IBGP routers.
  - It is not passed to EBGP peers.

Well-Known Discretionary: LOCAL_PREF



- Routers A and B are IBGP neighbors in AS 64520 and both receive updates about network 172.16.0.0 from different directions.
  - The local preference on router A is set to 200.
  - The local preference on router B is set to 150.
- Because the local preference for router A is higher, it is selected as the preferred exit point from AS 64520.

Configuring the Default Local Preference

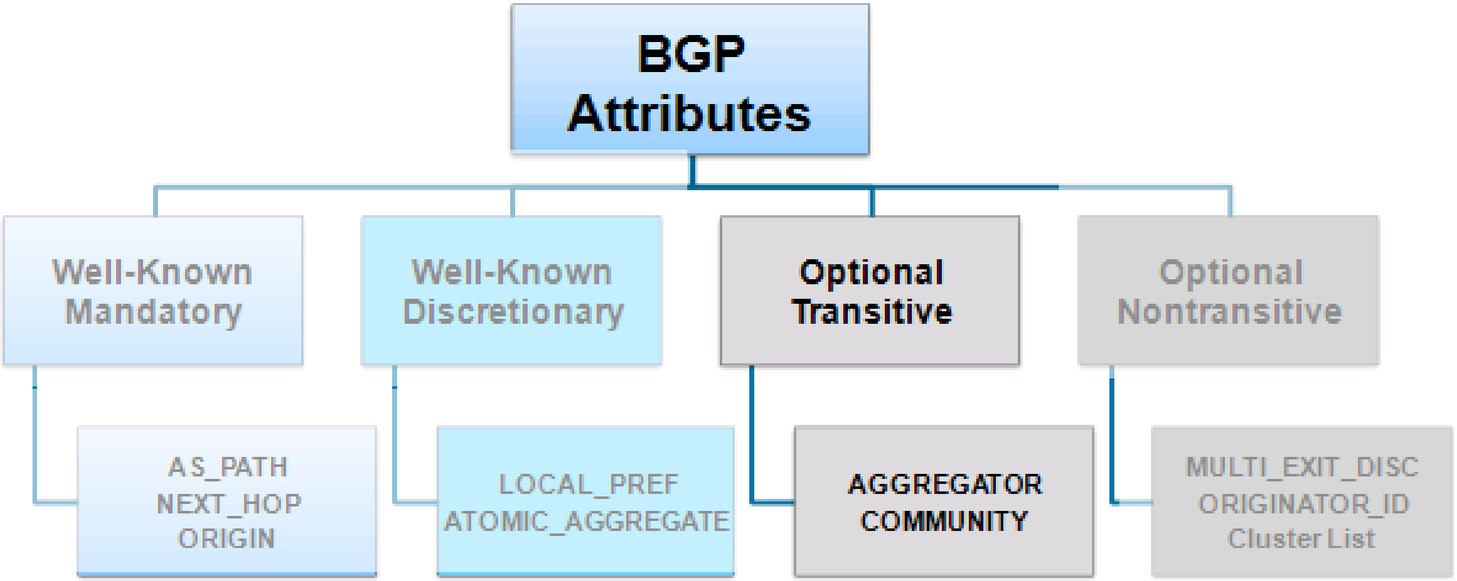
- The **`bgp default local-preference`** command changes the default local preference value.
  - With this command, all IBGP routes that are advertised have the local preference set to the value specified.
  - If an EBGP neighbor receives a local preference value, the EBGP neighbor ignores it.

Well-Known Discretionary: ATOMIC_AGGREGATE

- The Atomic Aggregate attribute is used to indicate that routes have been summarized.
  - Attribute warns that the received information may not necessarily be the most complete route information available.
- Attribute is set to either True or False with “true” alerting other BGP routers that multiple destinations have been grouped into a single update.
  - Router update includes its router ID and AS number along with the supernet route enabling administrators to determine which BGP router is responsible for a particular instance of aggregation.
  - Tracing a supernet to its original "aggregator" may be necessary for troubleshooting purposes.

Optional Transitive

- Attribute may or may not be recognized by all BGP implementations.
- Because the attribute is transitive, BGP accepts and advertises the attribute even if it is not recognized.

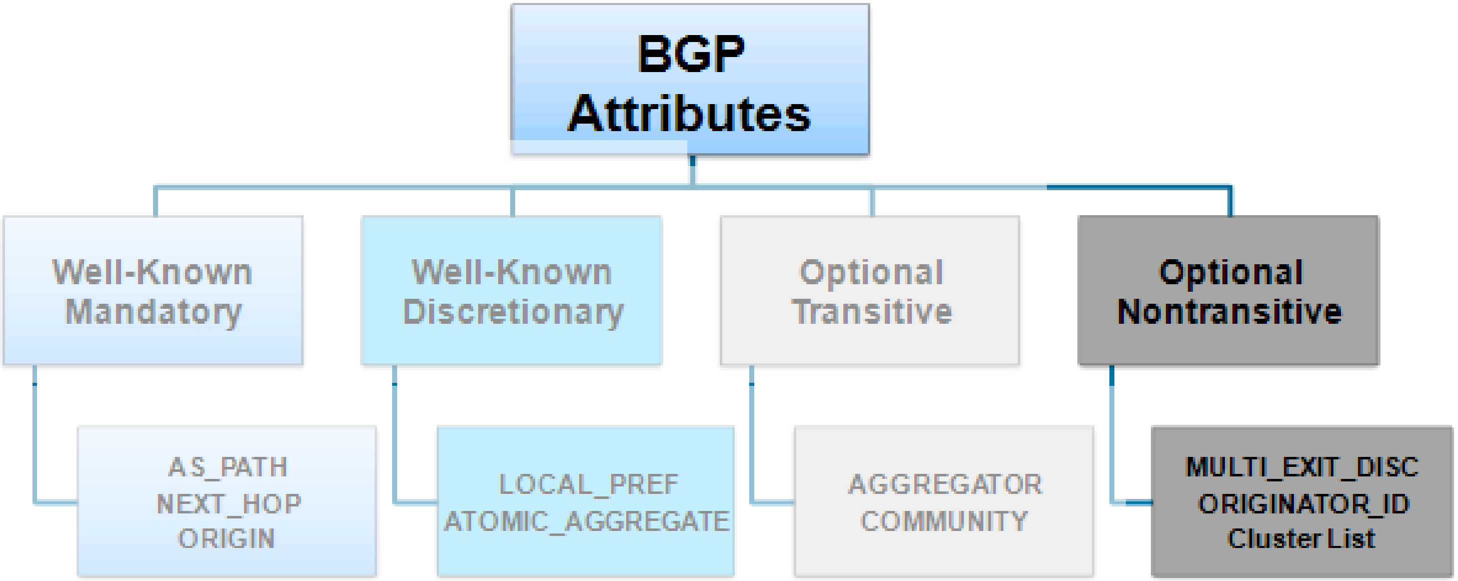


Optional Transitive: Community

- The BGP community attribute can be used to filter incoming or outgoing routes.
  - BGP routers can tag routes with an indicator (the community) and allow other routers to make decisions based on that tag.
- If a router does not understand the concept of communities, it defers to the next router.
  - However, if the router does understand the concept, it must be configured to propagate the community; otherwise, communities are dropped by default.
- Communities are not restricted to one network or one AS, and they have no physical boundaries.

Optional Nontransitive

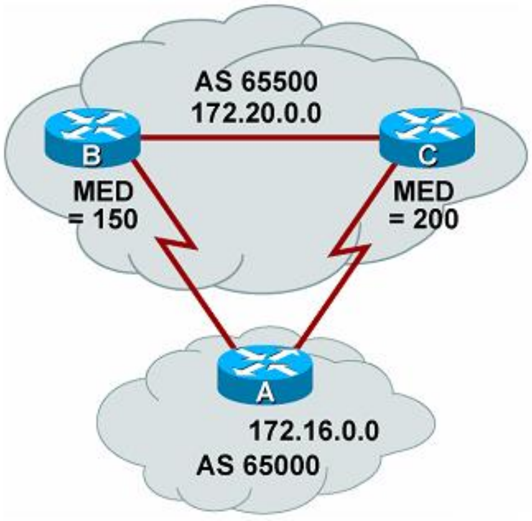
- Attribute that may or may not be recognized by all BGP implementations.
- Whether or not the receiving BGP router recognizes the attribute, it is nontransitive and is not passed along to other BGP peers.



Optional Nontransitive: MED

- The Multiple Exit Discriminator (MED) attribute, also called the *metric,* provides a hint to external neighbors about the preferred path into an AS that has multiple entry points.
  - Lower MED is preferred over a higher MED!
- The MED is sent to EBGP peers and those routers propagate the MED within their AS.
  - The routers within the AS use the MED, but do not pass it on to the next AS.
  - When the same update is passed on to another AS, the metric will be set back to the default of 0.
- By using the MED attribute, BGP is the only protocol that can affect how routes are sent into an AS.

Optional Nontransitive: MED

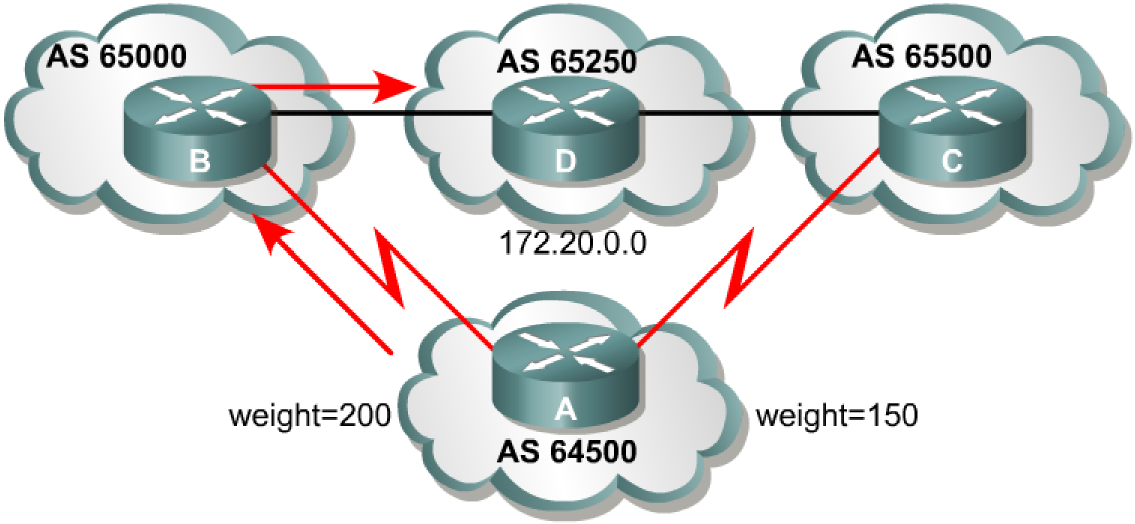


- Routers B and C include a MED attribute in the updates to router A.
    - Router B MED attribute is set to 150.
    - Router C MED attribute is set to 200.
- When A receives updates from B and C, it picks router B as the best next hop because of the lower MED.

Cisco Weight Attribute

- The Weight attribute is a Cisco proprietary attribute.
- Similar in function to the local preference, the weight attribute applies when 1 router has multiple exit points.
  - Local preference is used when 2+ routers provide multiple exit points.
- It is configured locally on a router and is not propagated to any other routers.
  - Routes with a higher weight are preferred when multiple routes exist to the same destination.
- 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.

Cisco Weight Attribute

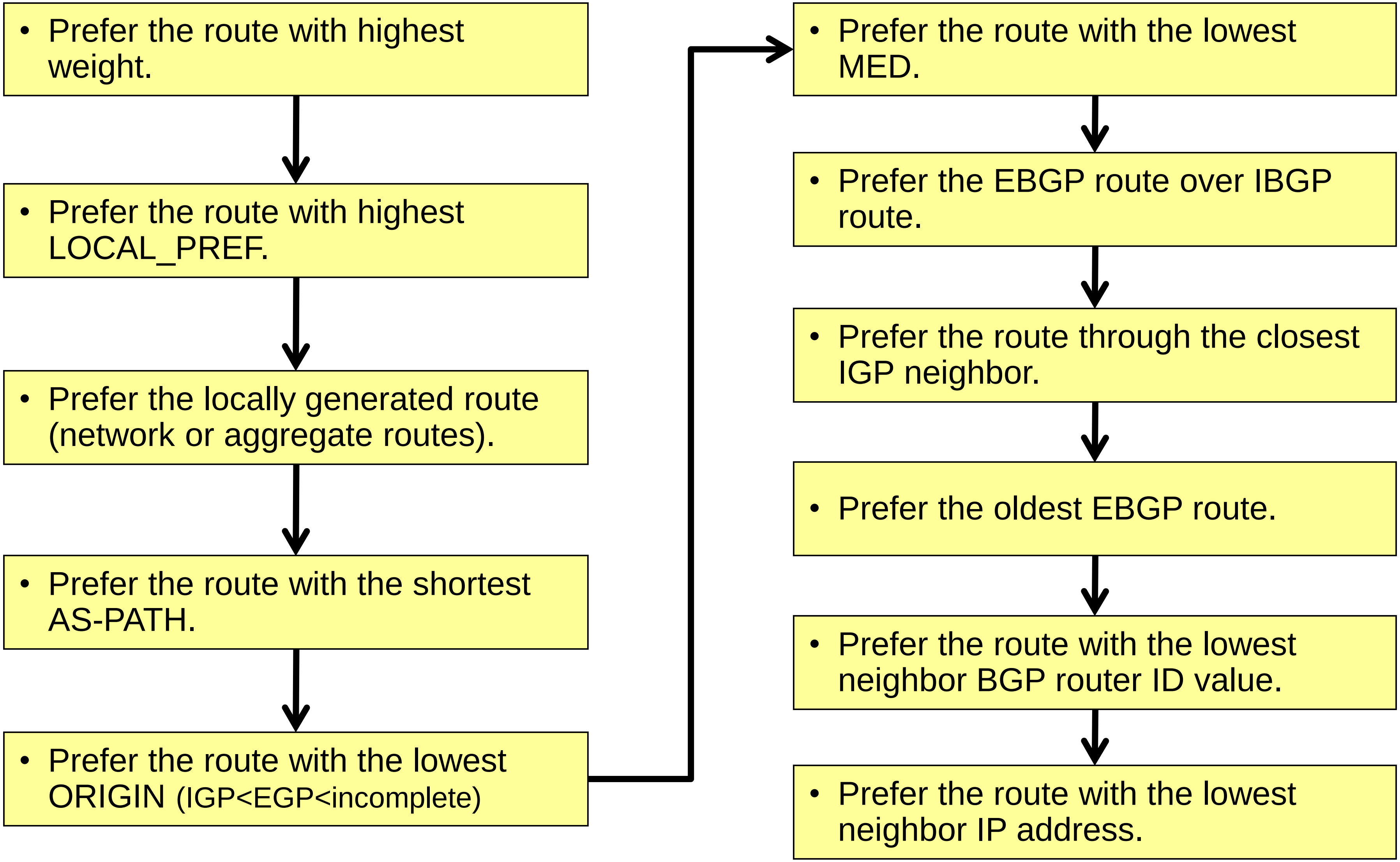


- Routers B and C learn about network 172.20.0.0 from AS 65250 and propagate the update to router A.
  - Therefore Router A has two ways to reach 172.20.0.0.
- Router A sets the weight of updates as follows:
  - Updates coming from router B are set to 200
  - Updates coming from router C are set to 150.
- Router A uses router B because of the higher weight.

BGP Route Selection Process

- The BGP best path decision is based on the value of attributes that the update contains and other BGP-configurable factors.
- BGP considers only synchronized routes with no AS loops and a valid next-hop address.

BGP Route Selection Process



- Prefer the route with highest weight.
- Prefer the route with highest LOCAL_PREF.
- Prefer the locally generated route (network or aggregate routes).
- Prefer the route with the shortest AS-PATH.
- Prefer the route with the lowest ORIGIN (IGP<EGP<incomplete)
- Prefer the route with the lowest MED.
- Prefer the EBGP route over IBGP route.
- Prefer the route through the closest IGP neighbor.
- Prefer the oldest EBGP route.
- Prefer the route with the lowest neighbor BGP router ID value.
- Prefer the route with the lowest neighbor IP address.

Configuring BGP

Planning to Deploy BGP

- Prior to deploying a BGP routing solution, the following should be considered:
  - IP addressing plan
  - Network topology
  - BGP relationship with service provider(s)
- Once the requirements have been assessed, the implementation plan can be created.

Implementing Basic BGP

- The information necessary to implement BGP routing includes the following:
  - The AS numbers of enterprise and service provider.
  - The IP addresses of all the neighbors (peers) involved.
  - The networks that are to be advertised into BGP
- In the implementation plan, basic BGP tasks include the following:
  - Define the BGP process
  - Establish the neighbor relationships
  - Advertise the networks into BGP

Verifying BGP

- After implementing BGP, verification should confirm proper deployment on each router.
- Verification tasks include verifying:
  - That the appropriate BGP neighbor relationships and adjacencies are established.
  - That the BGP table is populated with the necessary information.
  - That IP routing table is populated with the necessary information.
  - That there is connectivity in the network between routers and to other devices.
  - That BGP behaves as expected in a case of a topology change, by testing link failure and router failure events.

Documenting

- After a successful BGP deployment, the solution and verification process and results should be documented for future reference.
- Documentation should include:
  - A topology map
  - The IP addressing plan
  - The autonomous system hierarchy
  - The networks and interfaces included in BGP on each router
  - The default and any special metrics configured
  - The verification results.

Enable BGP Routing

- Define BGP as the IP routing protocol.

```
Router(config)#
```

```
router bgp autonomous-system
```

- The *autonomous-system* value is either an internally generated number (if not connecting to a provider network) or obtained from an ISP or RIR.
  - It is a required parameter.
  - It can be any positive integer in the range from 1 to 65535.
- Only one instance of BGP can be configured on the router at a single time.

4-byte AS numbers

- On Cisco routers there are two formats used to configure a 4-byte AS number:
  - **asplain**: The Cisco implementation.
  - **asdot**: The RFC 5396 implementation.
    - Use the `bgp asnotation dot` command to configure.
    - AS numbers must be written using the asdot format, or the regular expression match will fail.
- **Note:** The 4-byte AS number will not be used in this chapter; therefore, all examples use the 2-byte AS numbering format.
- **Example:**
  - AS 65546 is represented as 1.10
  - AS 65536 is represented as 1.0
  - AS 4294967295 is represented as 65535.65535

http://labs.spritelink.net/ascalc

Defining BGP Neighbors

- Identify peer router with which to establish a BGP session.

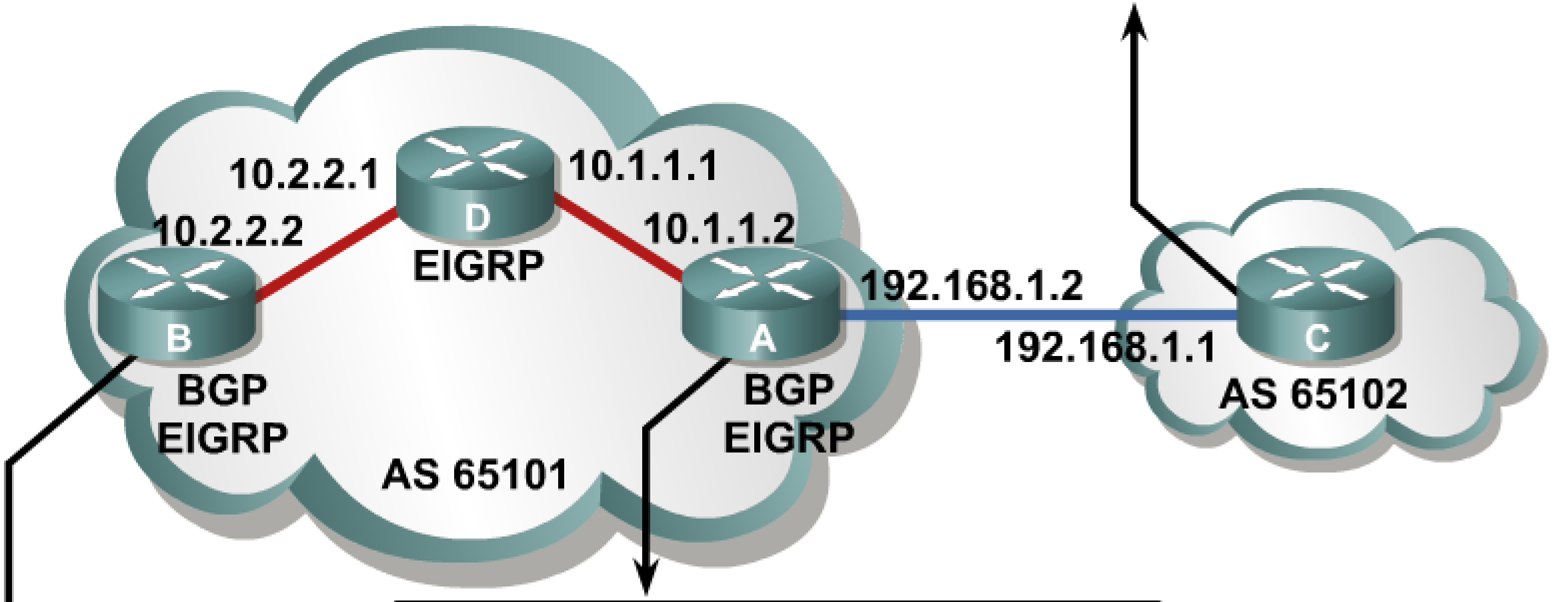
```
Router(config-router)#
```

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

- The *ip-address* is the destination address of the BGP peer.
  - The address must be reachable before attempting to establish the BGP relationship.
- The *autonomous-system* value is used to identify if the session is with internal BGP (IBGP) peers or with external BGP (EBGP) peers.
  - If the value is the same as the router’s AS, then an IBGP session is attempted.
  - If the value is not the same as the router’s AS, then an EBGP session is attempted.

Example: BGP `neighbor` Command

```
router bgp 65102
 neighbor 192.168.1.2 remote-as 65101
```



```
router bgp 65101
 neighbor 192.168.1.1 remote-as 65102
 neighbor 10.2.2.2 remote-as 65101
```

```
router bgp 65101
 neighbor 10.1.1.2 remote-as 65101
```

BGP Peer Groups

- In BGP, neighbors are often configured with the same update policies.
- To simplify configuration and make updating more efficient, neighbors with the same update policies can be grouped into **peer groups**.
  - Recommended approach when there are many BGP peers.
- Instead of separately defining the same policies for each neighbor, a peer group can be defined with these policies assigned to the peer group.
  - Individual neighbors are then made members of the peer group.
  - Members of the peer group inherit all the peer group’s configuration options.
  - Only options that affect the inbound updates can be overridden.

Defining a BGP Peer Group

- Create a peer group on the local router.

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

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

- The *peer-group-name* is the name of the BGP peer group to be created.
- The name is local to the router on which it is configured and is not passed to any other router.

Assign Neighbors to the Peer Group

- Assign neighbors as part of the peer group.

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

```
neighbor ip-address peer-group peer-group-name
```

- The *`ip-address`* is the IP address of the neighbor that is to be assigned as a member of the peer group.
- The *`peer-group-name`* must already exist.
  - Note: The **`clear ip bgp peer-group`** *`peer-group-name`* EXEC command can be used to reset the BGP connections for all members of a peer group.

Shut Down a BGP Neighbor

- To disable an existing BGP neighbor or peer group relationship.

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

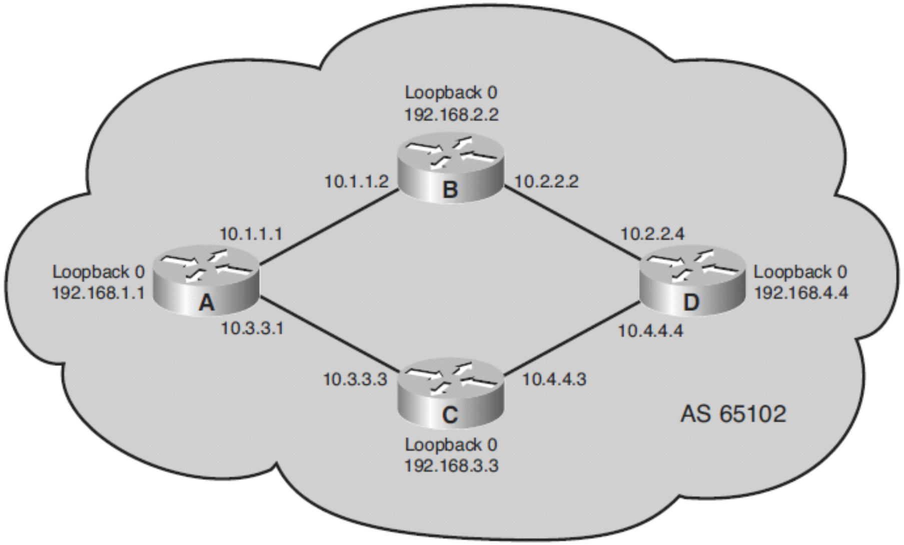
```
neighbor {ip-address | peer-group-name} shutdown
```

- Useful when making major policy changes to a neighboring router.
- The command not only terminates the session, but also removes all associated routing information.
- To re-enable the neighbor prepend the **`no`** keyword to the command.

IBGP Source IP Address Problem

- BGP does not accept unsolicited updates.
  - It must be aware of every neighboring router and have a **`neighbor`** statement for it.
- For example, when a router creates and forwards a packet, the IP address of the outbound interface is used as that packet’s source address by default.
  - For BGP packets, this source IP address must match the address in the corresponding **`neighbor`** statement on the other router or the routers will not establish the BGP session.
  - This is not a problem for EBGP neighbors as they are typically directly connected.

IBGP Source IP Address Problem



- When multiple paths exist between IBGP neighbors, the BGP source address can cause problems:
  - Router D uses the **`neighbor 10.3.3.1 remote-as 65102`** command to establish a relationship with A.
  - However, router A is sending BGP packets to D via B therefore the source IP address of the packets is 10.1.1.1.
  - The IBGP session between A and D cannot be established because D does not recognize 10.1.1.1 as a BGP neighbor.

IBGP Source IP Address Solution

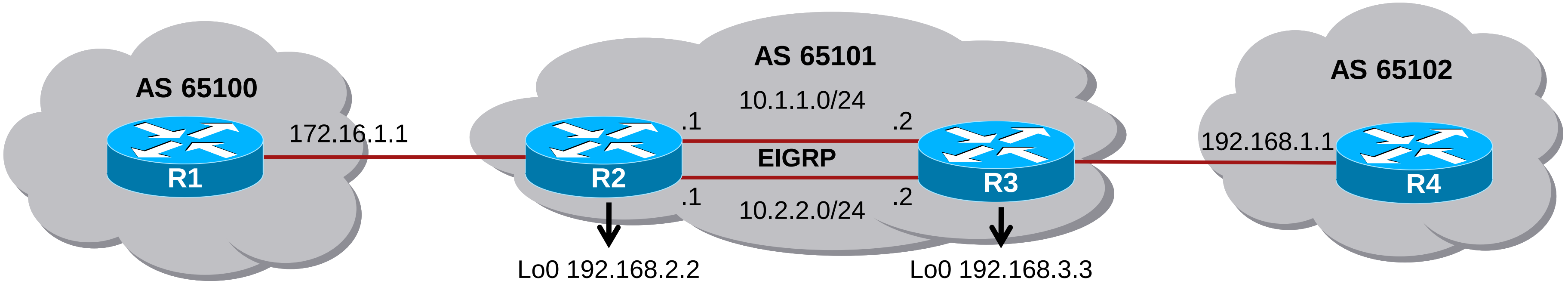
- Establish the IBGP session using a loopback interface.

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

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

- Informs the router to use a loopback interface address for all BGP packets.
- Overrides the default source IP address for BGP packets.
- Typically only used with IBGP sessions.
- As an added bonus, physical interfaces can go down for any number of reasons but loopbacks never fail.

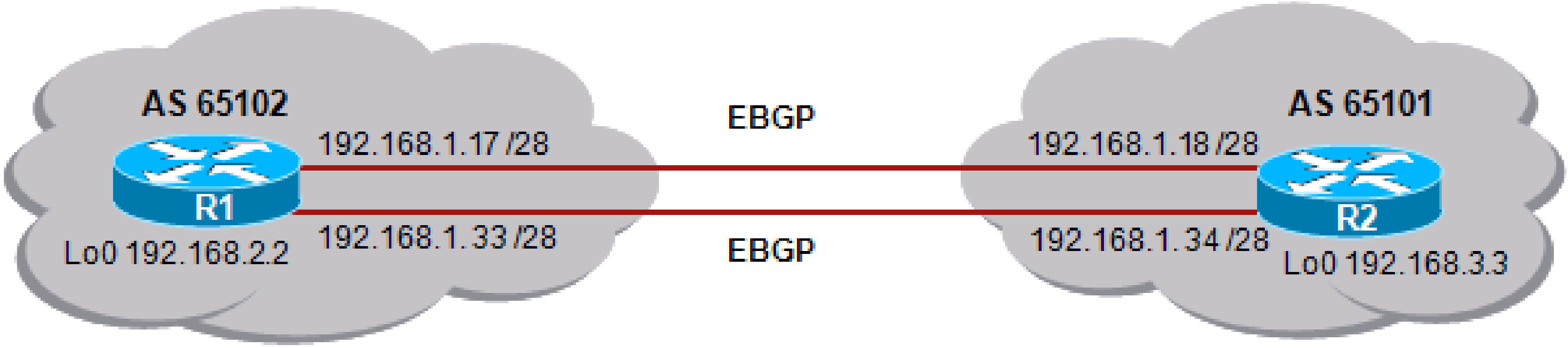
IBGP Source IP Address Example



```
R2(config)# router bgp 65101
R2(config-router)# neighbor 172.16.1.1 remote-as 65100
R2(config-router)# neighbor 192.168.3.3 remote-as 65101
R2(config-router)# neighbor 192.168.3.3 update-source loopback0
R2(config-router)# exit
R2(config)# router eigrp 1
R2(config-router)# network 10.0.0.0
R2(config-router)# network 192.168.2.0
R2(config-router)#
```

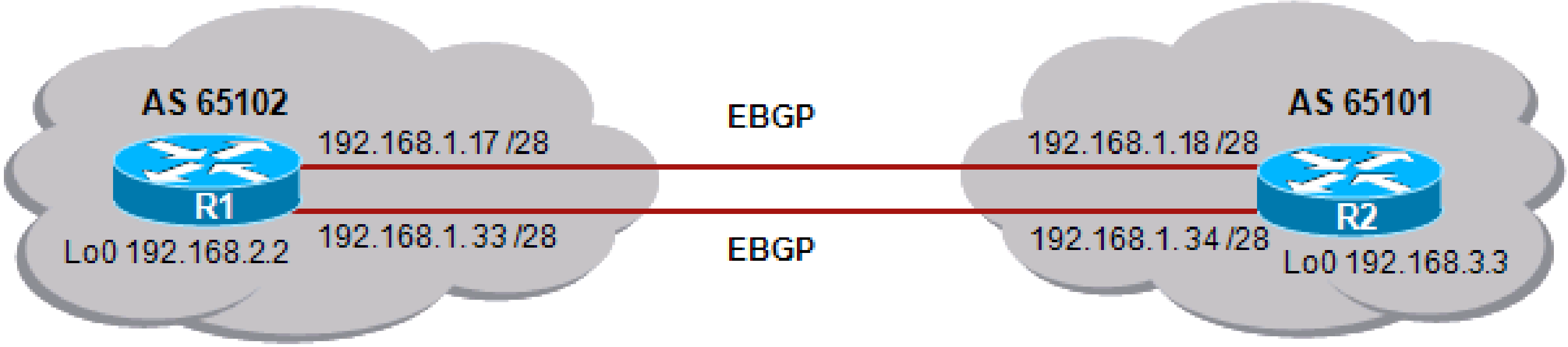
```
R3(config)# router bgp 65101
R3(config-router)# neighbor 192.168.1.1 remote-as 65102
R3(config-router)# neighbor 192.168.2.2 remote-as 65101
R3(config-router)# neighbor 192.168.2.2 update-source loopback0
R3(config-router)# exit
R3(config)# router eigrp 1
R3(config-router)# network 10.0.0.0
R3(config-router)# network 192.168.3.0
R3(config-router)#
```

EBGP Dual-Homed Problem



- R1 in AS 65102 is dual-homed with R2 in AS 65101.
- A problem can occur if R1 only uses a single **`neighbor`** statement pointing to 192.168.1.18 on R2 .
  - If that link fails, the BGP session between these AS is lost, and no packets pass from one autonomous system to the next, even though another link exists.
- A solution is configuring two **`neighbor`** statements on R1 pointing to 192.168.1.18 and 192.168.1.34.
  - However, this doubles the BGP updates from R1 to R2.

EBGP Dual-Homed Solution



- The ideal solution is to:
  - Use loopback addresses.
  - Configure static routes to reach the loopback address of the other router.
  - Configure the **`neighbor ebgp-multihop`** command to inform the BGP process that this neighbor is more than one hop away.

Enable Multihop EBGP

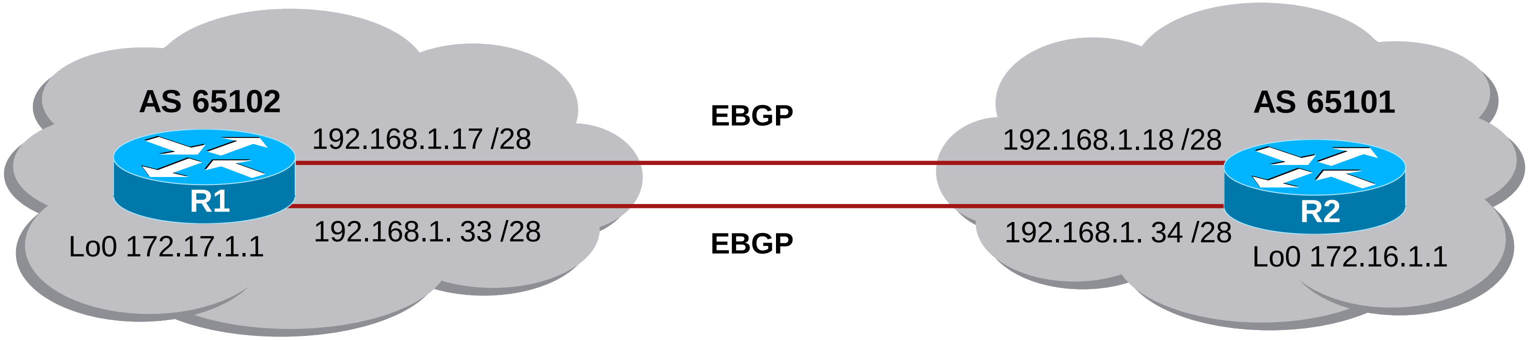
- Increase the time-to-live (TTL) for EBGP connections.

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

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

- This command is of value when redundant paths exist between EBGP neighbors.
- The default *ttl* is 1, therefore BGP peers must be directly connected.
  - The range is from 1 to 255 hops.
- Increasing the *ttl* enables BGP to establish EBGP connections beyond one hop and also enables BGP to perform load balancing.

Multihop EBGP Example



```
R1(config)# router bgp 65102
R1(config-router)# neighbor 172.16.1.1 remote-as 65101
R1(config-router)# neighbor 172.16.1.1 update-source loopback0
R1(config-router)# neighbor 172.16.1.1 ebgp-multihop 2
R1(config-router)#  exit
R1(config)# ip route 172.16.1.1 255.255.255.255 192.168.1.18
R1(config)# ip route 172.16.1.1 255.255.255.255 192.168.1.34
R1(config)#
```

```
R2(config)# router bgp 65101
R2(config-router)# neighbor 172.17.1.1 remote-as 65102
R2(config-router)# neighbor 172.17.1.1 update-source loopback0
R2(config-router)# neighbor 172.17.1.1 ebgp-multihop 2
R2(config-router)#  exit
R2(config)# ip route 172.17.1.1 255.255.255.255 192.168.1.17
R2(config)# ip route 172.17.1.1 255.255.255.255 192.168.1.33
R2(config)#
```

Advertising EBGP Routes to IBGP Peers

- When an EBGP router receives an update from an EBGP neighbor and forwards the update to its IBGP peers, the source IP address will still be that of the EBGP router.
  - IBGP neighbors will have to be configured to reach that external IP address.
- Another solution is to override a router’s default behavior and force it to advertise itself as the next-hop address for routes sent to a neighbor.
  - To do so, use the **`neighbor next-hop-self`** router configuration command

neighbor next-hop-self Command

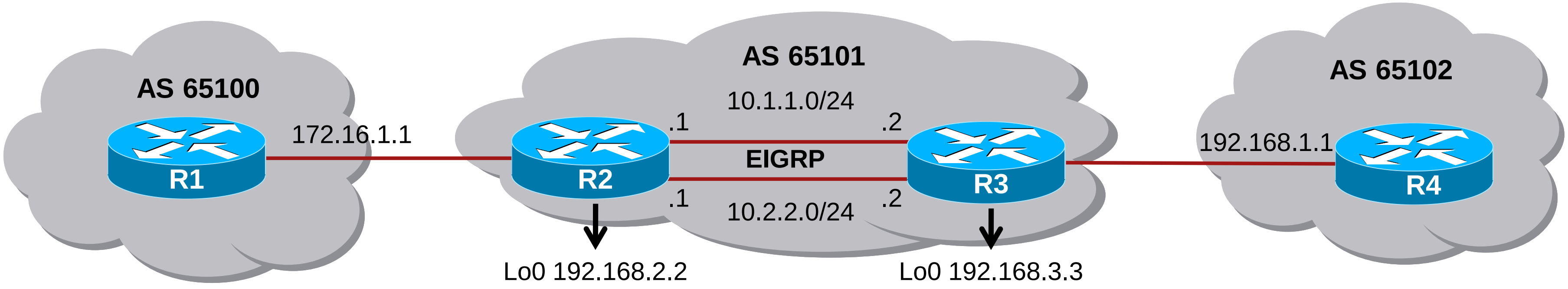
- Configure the router as the next hop for a BGP-speaking peer.

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

```
neighbor {ip-address | peer-group-name} next-hop-self
```

- The command forces BGP to advertise itself as the source of the routes.
- The *ip-address* identifies the peer router to which advertisements will be sent, with this router identified as the next hop.
- This command is useful in unmeshed networks (such as Frame Relay) where BGP neighbors may not have direct access to all other neighbors on the same IP subnet.

Next Hop Self Example



```
R2(config)# router bgp 65101
R2(config-router)# neighbor 172.16.1.1 remote-as 65100
R2(config-router)# neighbor 192.168.3.3 remote-as 65101
R2(config-router)# neighbor 192.168.3.3 update-source loopback0
R2(config-router)# neighbor 192.168.3.3 next-hop-self
R2(config-router)# exit
R2(config)# router eigrp 1
R2(config-router)# network 10.0.0.0
R2(config-router)# network 192.168.2.0
R2(config-router)#
```

BGP Synchronization

- Recall that the BGP synchronization rule states that:
  - “*A BGP router should not use, or advertise a route learned by IBGP, unless that route is local or is learned from the IGP*.”
- By default synchronization is disabled, therefore BGP can use and advertise to an external BGP neighbor routes learned from an IBGP neighbor that are not present in the local routing table.
  - Use the **`synchronization`** router configuration command to enable BGP synchronization so that a router will not advertise routes in BGP until it learns them in an IGP.
  - The **`no synchronization`** router configuration command disables synchronization.

Defining Networks That BGP Advertises

- Two options are available to advertise networks into BGP:
  - The **`network`** command.
  - Redistributing IGP routes into BGP.
- Note: Redistributing is not recommended because it could result in unstable BGP tables.

Identify BGP Networks

- Enable BGP to advertise a network if it is present.

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

```
network network-number [mask network-mask] [route-map
  map-tag]
```

- The BGP **network** command determines which networks this router advertises.
  - Unlike IGPs, the command does not start BGP on specific interfaces.
- The **mask** parameter indicates that BGP-4 supports subnetting and supernetting.
  - If the mask is not specified, this command announces only the classful network
- It is also important to note that the prefix must exactly match (address and mask) an entry in the IP routing table.

BGP Route Must Be in IP Routing Table

- It is important to understand that any network (both address and mask) must exist in the routing table for the network to be advertised in BGP.
- For example, to summarize many networks and advertise a CIDR block 192.168.0.0/16, configure:

```
network 192.168.0.0 mask 255.255.0.0
ip route 192.168.0.0 255.255.0.0 null0
```

- Now BGP can find an exact match in the routing table and announce the 192.168.0.0/16 network to its neighbors.
  - The advertised static route would never actually be used since BGP would contain longer prefix matching routes in its routing table.

BGP Authentication

- BGP supports message digest 5 (MD5) neighbor authentication.
  - MD5 sends a “message digest” (also called a “hash”), which is created using the key and a message.
  - The message digest is then sent instead of the key.
  - The key itself is not sent, preventing it from being read by someone eavesdropping on the line while it is being transmitted.

- To enable MD5 authentication on a TCP connection between two BGP peers, use the router configuration command:

```
neighbor {ip-address | peer-group-name} password string
```

Enable MD5 authentication

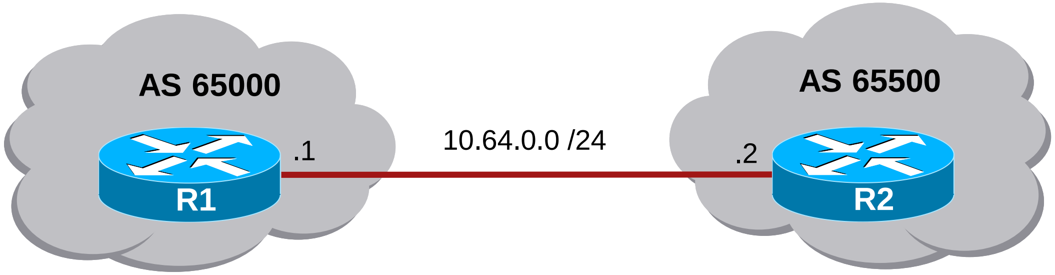
- Enable MD5 authentication between two BGP peers.

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

```
neighbor {ip-address | peer-group-name} password string
```

- This is the only command required to enable MD5 authentication.
- The *string* value is:
  - 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.

Configuring MD5 Authentication



```
R1(config)# router bgp 65000
R1(config-router)# neighbor 10.64.0.2 remote-as 65500
R1(config-router)# neighbor 10.64.0.2 password BGP-Pa55w0rd
R1(config-router)#
```

```
R2(config)# router bgp 65500
R2(config-router)# neighbor 10.64.0.1 remote-as 65000
R2(config-router)# neighbor 10.64.0.1 password BGP-Pa55w0rd
R2(config-router)#
```

MD5 Configuration Problems

- If a router has a password configured for a neighbor, but the neighbor router does not have a password configured, the following message will appear on the console screen:

```
%TCP-6-BADAUTH: No MD5 digest from 10.1.0.2(179) to
10.1.0.1(20236)
```

- Similarly, if the two routers have different passwords configured, the following will appear:

```
%TCP-6-BADAUTH: Invalid MD5 digest from
10.1.0.1(12293) to 10.1.0.2(179)
```

Clearing the BGP Session

- When policies such as access lists or attributes are changed, the Cisco IOS applies changes on only those updates received or sent *after* and not existing routes in the BGP and routing tables*.*
  - It can take a long time for the policy to be applied to all networks.
- There are three ways to ensure that the policy change is immediately applied to all affected prefixes and paths.
  - Hard reset
  - Soft reset (outbound and inbound)
  - Route refresh

Hard Reset of BGP Sessions

- Reset all BGP connections with this router.

```
Router#
```

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

- Entire BGP forwarding table is discarded.
- BGP session makes the transition from established to idle; everything must be relearned.
- When the *`neighbor-address`* value is used, it resets only a single neighbor and BGP session. Everything from this neighbor must be relearned.
  - It is less severe than **`clear ip bgp *`**.

Soft Reset Outbound

- Resets all BGP connections without loss of routes.

```
Router#
```

```
clear ip bgp {* | neighbor-address} [soft out]
```

- The connection remains established and the command does not reset the BGP session.
  - Instead the router creates a new update and sends the whole table to the specified neighbors.
- This update includes withdrawal commands for networks that the neighbor will not see anymore based on the new outbound policy.
- This option is highly recommended when you are changing outbound policy.

Soft Reset Inbound: Method #1

- Two commands are required.

```
Router(config-router)#
neighbor {ip-address} soft-reconfiguration inbound
```

- Use this command when changes need to be made without forcing the other side to resend everything.
- It causes the BGP router to retain an unfiltered table of what a neighbor had sent but can be memory intensive.

```
Router#
clear ip bgp {* | neighbor-address} [soft in]
```

- Causes the router to use the stored unfiltered table to generate new inbound updates and the new results are placed in the BGP forwarding database.

Soft Reset Inbound: Method #2

- Also called route refresh.

```
Router#
```

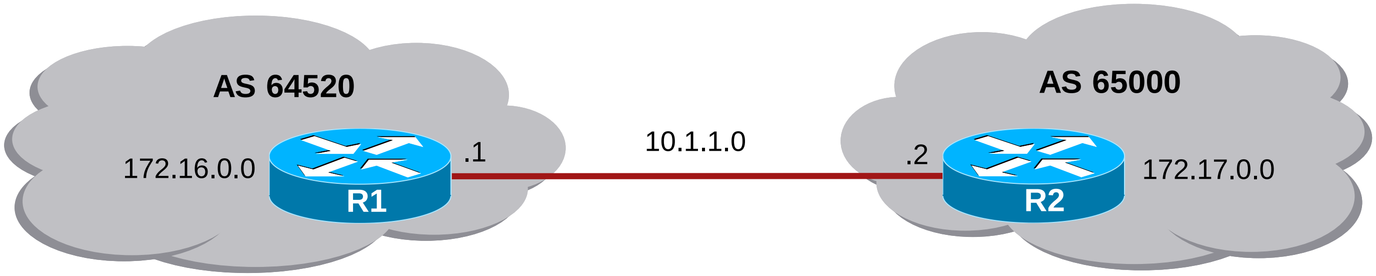
```
clear ip bgp {* | neighbor-address} [soft in | in]
```

- This dynamically soft resets inbound updates.
- Unlike method #1, this method requires no preconfiguration and requires significantly less memory.

Monitoring Received BGP Routes

Command	Description
show ip bgp neighbors {*address*} **received-routes**	Displays all received routes (both accepted and rejected) from the specified neighbor.
show ip bgp neighbors {*address*} **routes**	Displays all routes that are received and accepted from the specified neighbor. This output is a subset of the output displayed by the **received-routes** keyword.
show ip bgp	Displays entries in the BGP table.
show ip bgp summary	Displays neighbors state.
show ip bgp neighbors {*address*} **advertised-routes**	Displays all BGP routes that have been advertised to neighbors.

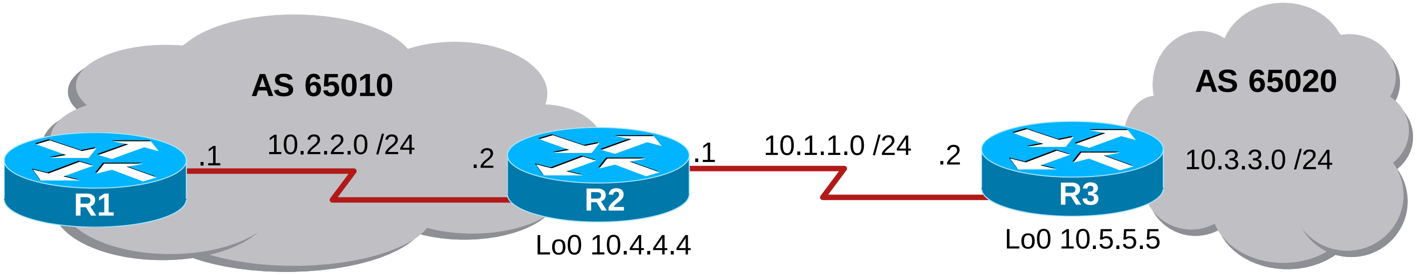
BGP Configuration Example #1



```
R1(config)# router bgp 64520
R1(config-router)# neighbor 10.1.1.2 remote-as 65000
R1(config-router)# network 172.16.0.0
R1(config-router)#
```

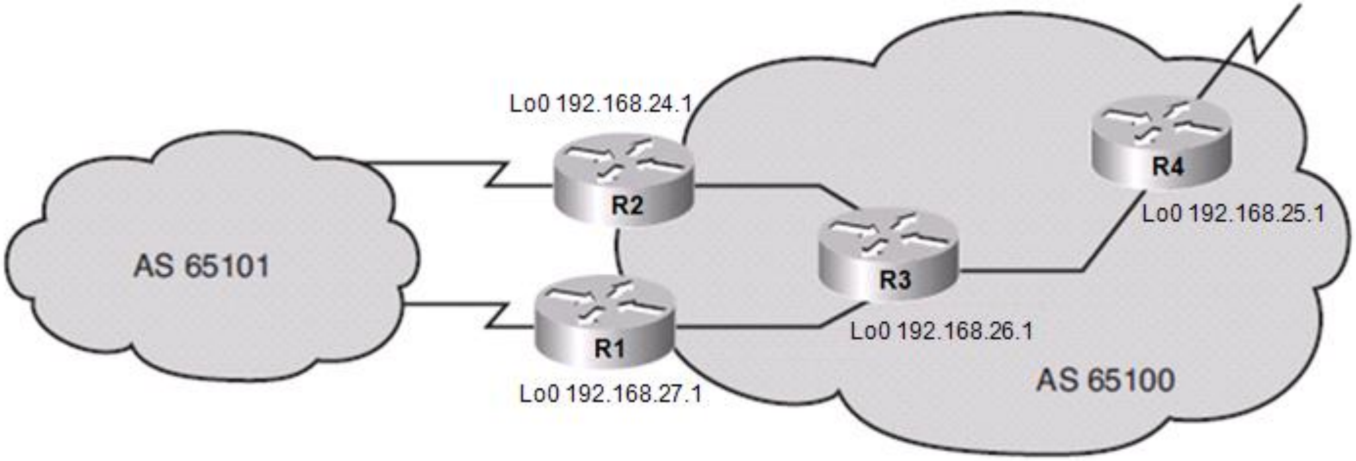
```
R2(config)# router bgp 65000
R2(config-router)# neighbor 10.1.1.1 remote-as 64520
R2(config-router)# network 172.17.0.0
R2(config-router)#
```

BGP Configuration Example #2



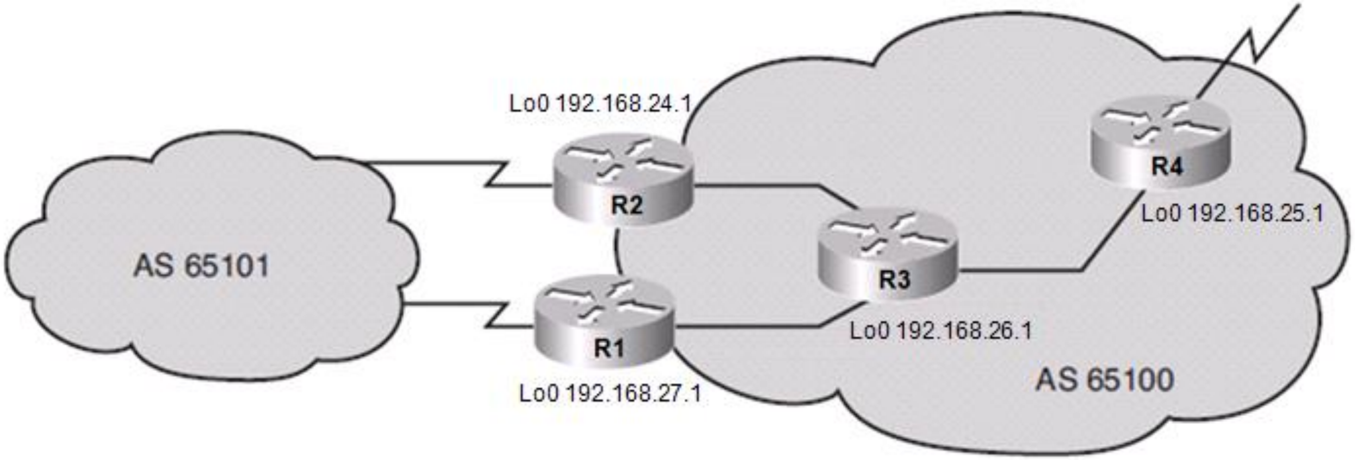
```
R2(config)# router bgp 65010
R2(config-router)# neighbor 10.1.1.2 remote-as 65020
R2(config-router)# network 10.2.2.0 mask 255.255.255.0
R2(config-router)# network 10.4.4.0 mask 255.255.255.0
R2(config-router)#
```

BGP Without Peer Group Example



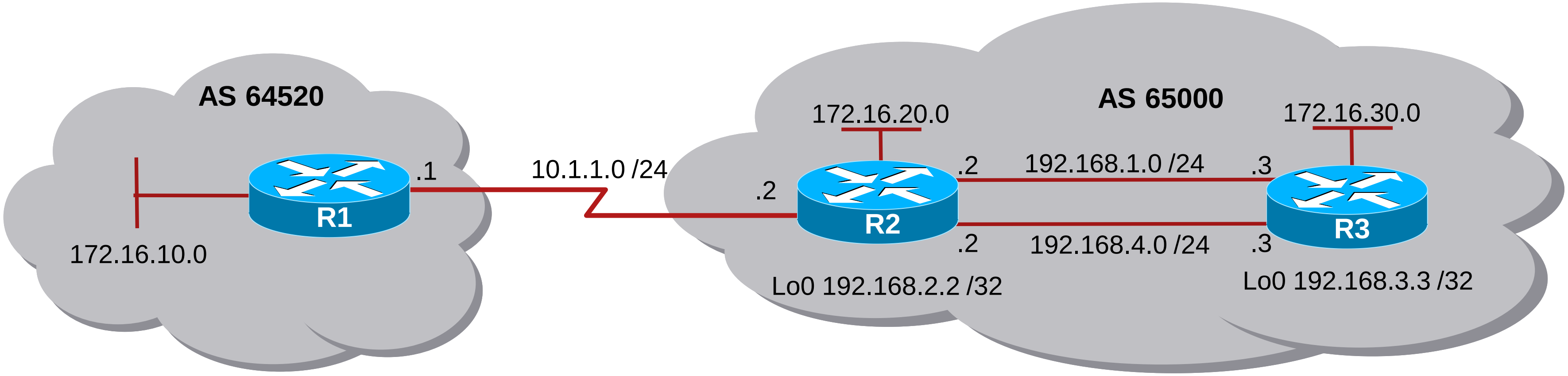
```
R1(config)# router bgp 65100
R1(config-router)# neighbor 192.168.24.1 remote-as 65100
R1(config-router)# neighbor 192.168.24.1 update-source loopback 0
R1(config-router)# neighbor 192.168.24.1 next-hop-self
R1(config-router)# neighbor 192.168.24.1 distribute-list 20 out
R1(config-router)#
R1(config-router)# neighbor 192.168.25.1 remote-as 65100
R1(config-router)# neighbor 192.168.25.1 update-source loopback 0
R1(config-router)# neighbor 192.168.25.1 next-hop-self
R1(config-router)# neighbor 192.168.25.1 distribute-list 20 out
R1(config-router)#
R1(config-router)# neighbor 192.168.26.1 remote-as 65100
R1(config-router)# neighbor 192.168.26.1 update-source loopback 0
R1(config-router)# neighbor 192.168.26.1 next-hop-self
R1(config-router)# neighbor 192.168.26.1 distribute-list 20 out
R1(config-router)#
```

BGP With Peer Group Example



```
R1(config)# router bgp 65100
R1(config-router)# neighbor INTERNAL peer-group
R1(config-router)# neighbor INTERNAL remote-as 65100
R1(config-router)# neighbor INTERNAL update-source loopback 0
R1(config-router)# neighbor INTERNAL next-hop-self
R1(config-router)# neighbor INTERNAL distribute-list 20 out
R1(config-router)# neighbor 192.168.24.1 peer-group INTERNAL
R1(config-router)# neighbor 192.168.25.1 peer-group INTERNAL
R1(config-router)# neighbor 192.168.26.1 peer-group INTERNAL
R1(config-router)#
```

IBGP and EBGP Example



```
R2(config)# router bgp 65000
R2(config-router)# neighbor 10.1.1.1 remote-as 64520
R2(config-router)# neighbor 192.168.3.3 remote-as 65000
R2(config-router)# neighbor 192.168.3.3 update-source loopback 0
R2(config-router)# neighbor 192.168.3.3 next-hop-self
R2(config-router)# network 172.16.20.0 mask 255.255.255.0
R2(config-router)# network 192.168.1.0
R2(config-router)# network 192.168.3.0
R2(config-router)#
```

Verifying and Troubleshooting BGP

Verifying and Troubleshooting BGP

Command	Description
show ip bgp	Displays entries in the BGP table. Specify a network number to get more specific information about a particular network.
show ip bgp neighbors	Displays detailed information about the TCP and BGP connections to neighbors.
show ip bgp summary	Displays the status of all BGP connections.
show ip bgp neighbors {*address*} advertised-routes	Displays all BGP routes that have been advertised to neighbors.
show ip bgp rib-failure	Displays BGP routes that were not installed in the routing information base (RIB), and the reason that they were not installed.
debug ip bgp [dampening | events | keepalives | updates]	

Verifying BGP: `show ip bgp`

Display the BGP topology database (the BGP table).

The status codes are shown in the first column of each line of output.

- ***** means that the next-hop address (in the fifth column) is valid.

- **r** means a RIB failure and the route was not installed in the RIB.

A **>** in the second column indicates the best path for a route selected by BGP.

This route is offered to the IP routing table.

The third column is either blank or has an "**i**" in it.

- If it has an **i**, an IBGP neighbor advertised this route to this router.

- If it is blank, BGP learned that route from an external peer.

```
R1# 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 64998 64997 i
*                   172.31.11.4                            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 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 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
```

This section lists three BGP path attributes: metric (MED), local preference, and weight.

The Path section lists the AS path. The last AS # is the originating AS.

If blank the route is from the current autonomous system.

The last column displays the ORIGIN attribute).

- **i** means the original router probably used a **`network`** command to introduce this network into BGP.

- **?** means the route was probably redistributed from an IGP into the BGP process.

Verifying BGP: `show ip bgp rib-failure`

- Displays BGP routes that were not installed in the RIB and the reason that they were not installed.
- In this example, the displayed routes were not installed because a route or routes with a better administrative distance already existed in the RIB.

```
R1# show ip bgp rib-failure
Network Next Hop RIB-failure RIB-NH Matches
172.31.1.0/24 172.31.1.3 Higher admin distance n/a
172.31.11.0/24 172.31.11.4 Higher admin distance n/a
```

Verifying BGP: `show ip bgp summary`

Verify the BGP neighbor relationship.

```
R1# 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
```

Verifying BGP: debug ip bgp updates

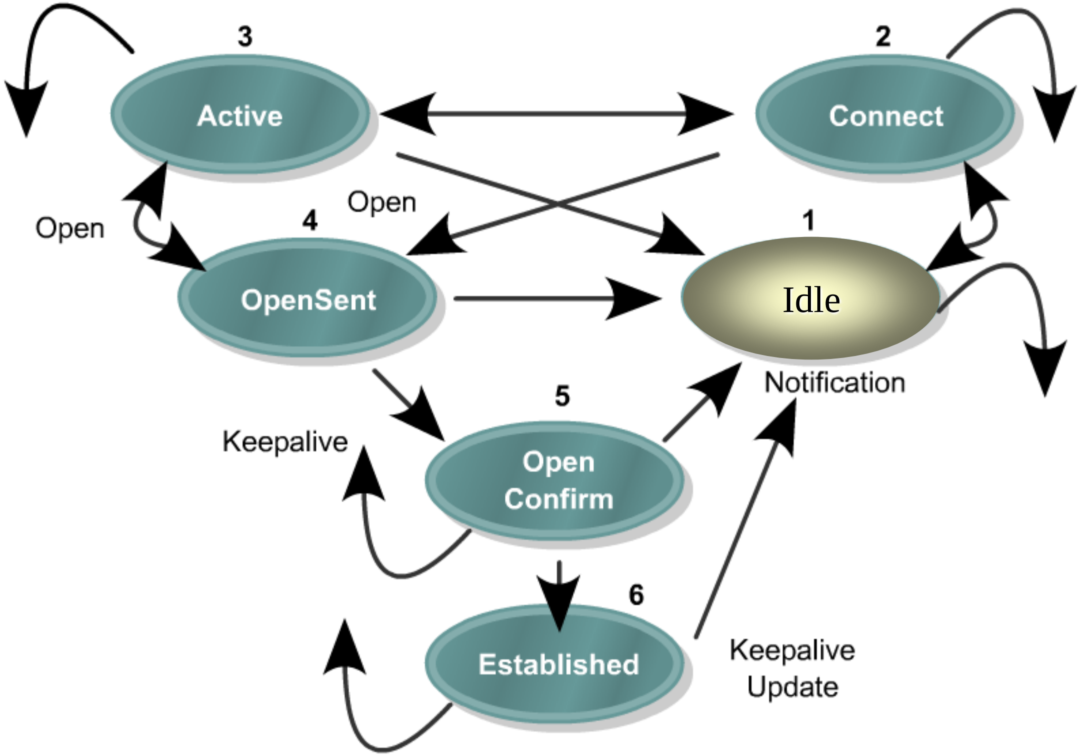
Verify the BGP neighbor relationship.

```
R1# debug ip bgp updates
Mobile router debugging is on for address family: IPv4 Unicast
R1# clear ip bgp 10.1.0.2
<output omitted>
*May 24 11:06:41.309: %BGP-5-ADJCHANGE: neighbor 10.1.0.2 Up
*May 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
*May 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
*May 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
*May 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
*May 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
*May 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>
*May 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
*May 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd 10.1.2.0/24
*May 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd 10.1.0.0/24
```

BGP States

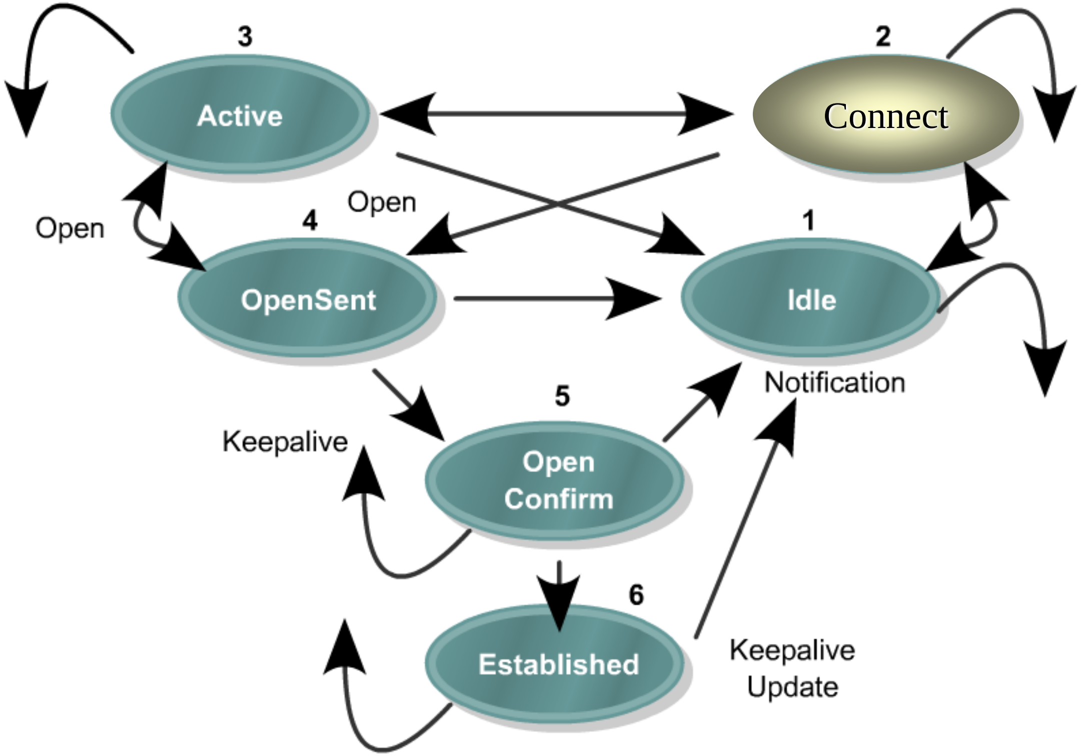
- BGP is a state machine that takes a router through the following states with its neighbors:
  - **Idle**
  - **Connect**
  - **Open sent**
  - **Open confirm**
  - **Established**
- The **Idle** state begins once the **`neighbor`** command is configured.

Idle State



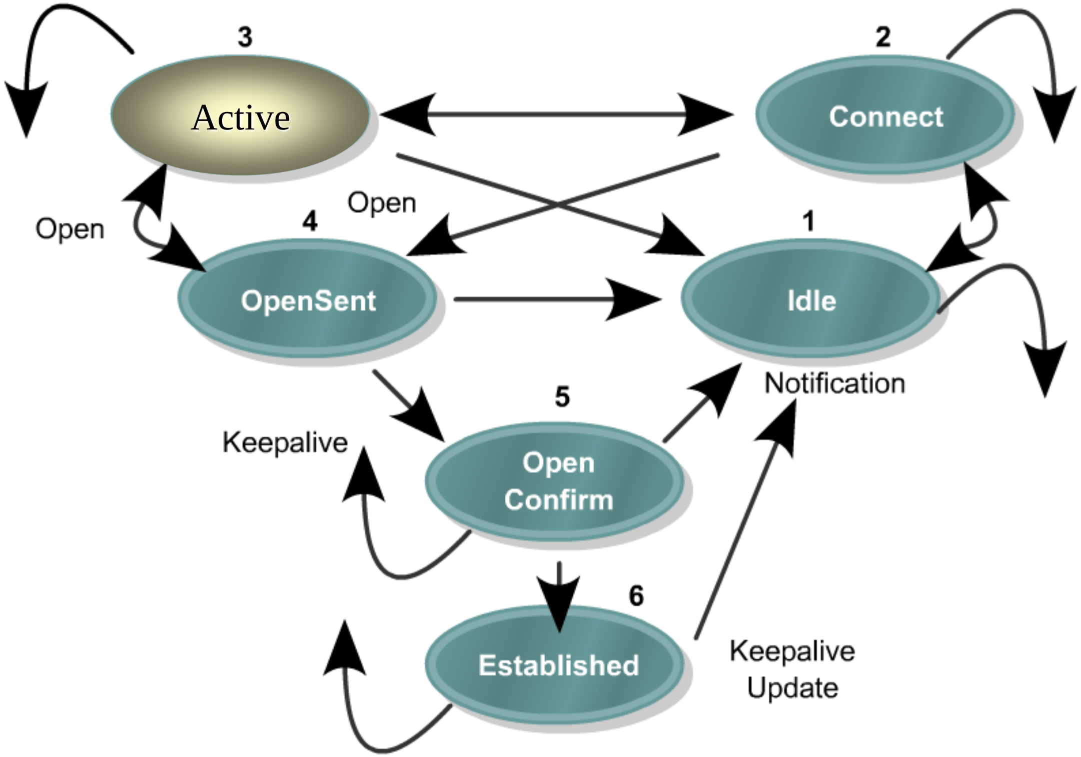
- The router is searching the routing table to see whether a route exists to reach the neighbor.
- If a router remains in this state then the router is:
  - Waiting for a static route to that IP address or network to be configured.
  - Waiting for the IGP to learn about this network from another router.

Connect State



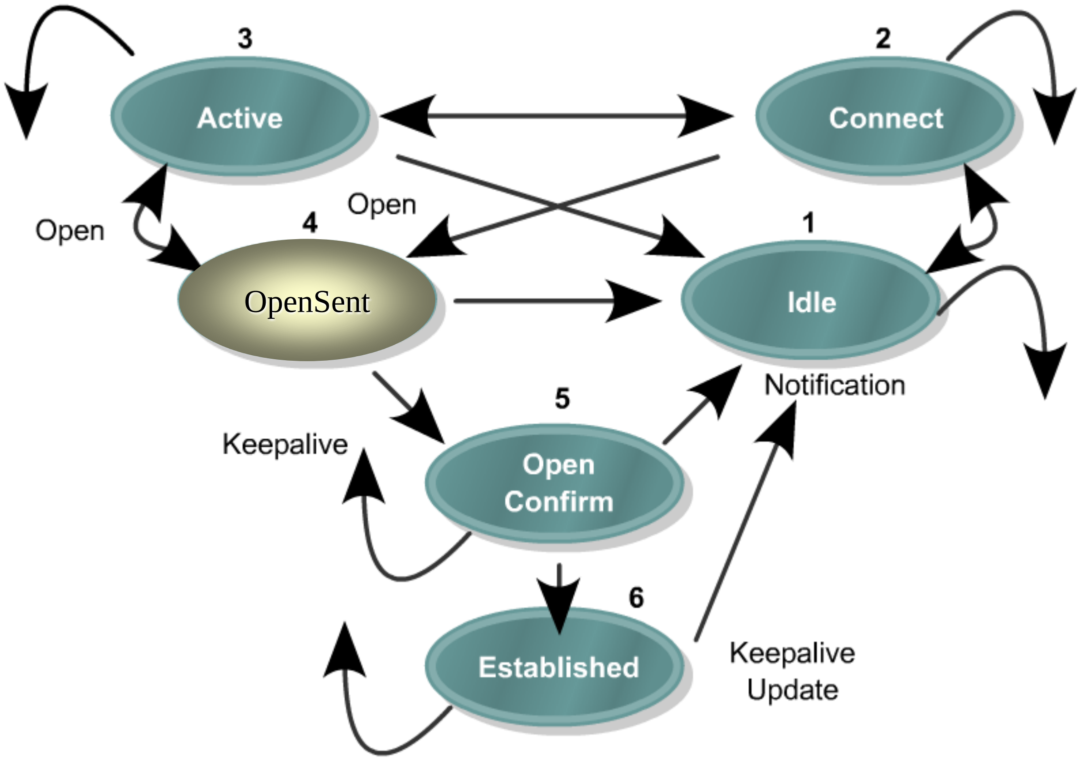
- The router found a route to the neighbor and has completed the three-way TCP handshake.

Active State



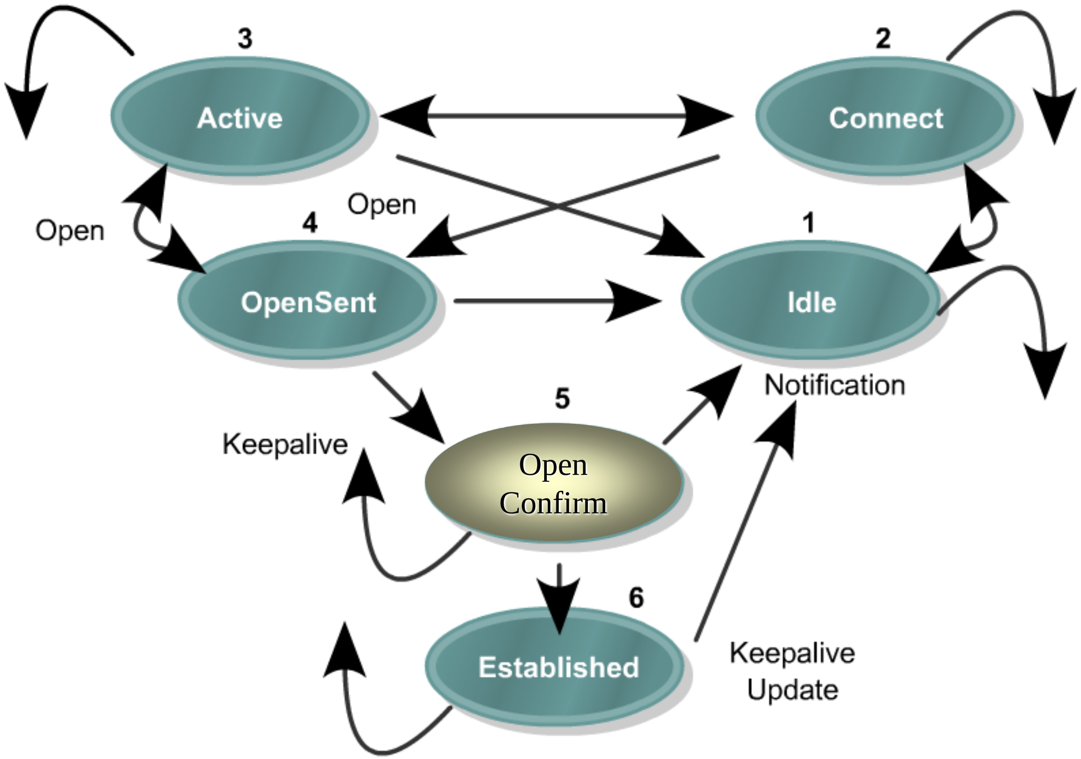
- BGP is trying to acquire a peer by initiating a TCP connection.
- If it is successful, it transitions to OpenSent otherwise the state returns to Idle.
- If the router remains is this state it means that the router has not received a response (open confirm packet) back from the neighbor.
    - Reasons for this include missing neighbor statement or incorrect AS number.

Open Sent State



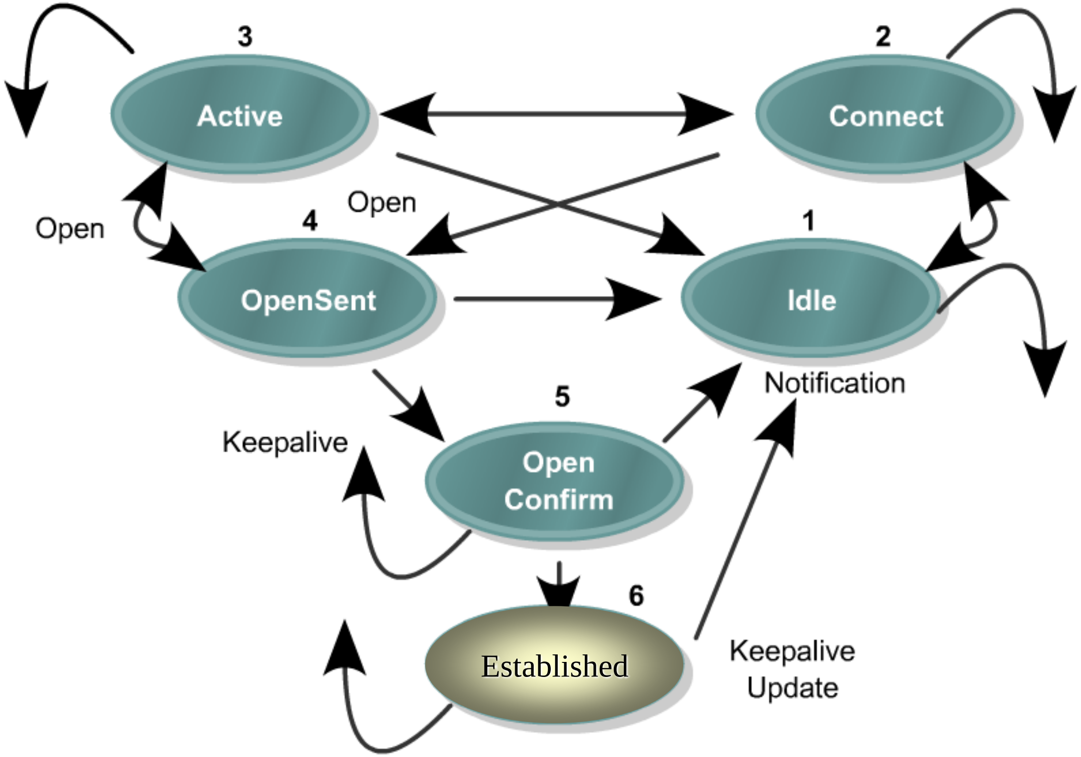
- An open message was sent, with the parameters for the BGP session.

Open Confirm



- The router received agreement on the parameters for establishing a session.

Established State



- This is the desired state for a neighbor relationship.
- It means peering is established and routing begins.

Verifying BGP: `show ip bgp neighbors`

Verify the BGP neighbor relationship.

```
R1# show 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>
```

Basic BGP Path Manipulation Using Route Maps

Route Maps and BGP

- In Chapter 4, Policy Based Routing (PBR) was used for redistribution.
  - Route maps are implemented using the **`redistribute`** command.
- In Chapter 5, route maps were used to define a routing policy other than basic destination-based routing using the routing table.
  - Route maps are implemented using the **`ip policy route-map`** command.
- In this chapter, route maps will be used with BGP to assign or alter BGP attributes.
  - Route maps are implemented using the **`neighbor route-map`** command.

Configuring Route Maps in BGP

Sample implementation plan:

- Define and name the route map with the **`route-map`** command.
  - Define the conditions to match (the **`match`** statements).
  - Define the action to be taken when there is a match (the **`set`** statements).
- Define which attribute to alter using the **`neighbor route-map`** router configuration command.
  - Filters incoming or outgoing BGP routes.
- Verify results.

Implementing Route Maps in BGP

```
Router(config)#
route-map map-tag [permit | deny] [sequence-number]
```

- Defines the route map conditions.

```
Router(config-route-map)#
match {criteria}
```

- Defines the criteria to match.

```
Router(config-route-map)#
set {actions}
```

- Defines the action to be taken on a match.

```
Router(config-router)#
neighbor {ip-address | peer-group-name} route-map map-name
{in | out}
```

- Applies the route-map to filter incoming or outgoing BGP routes to a neighbor.

`match` Commands Used in BGP

Command	Description
`match as-path`	Matches the AS_PATH attribute
`match ip address`	Matches any routes that have a destination network number address that is permitted by a standard or extended ACL
`match metric`	Matches routes with the metric specified
`match community`	Matches a BGP community
`match interface`	Matches any routes that have the next hop out of one of the interfaces specified
`match ip next-hop`	Matches any routes that have a next-hop router address that is passed by one of the ACLs specified
`match ip route-source`	Matches routes that have been advertised by routers and access servers at the address that is specified by the ACLs
`match route-type`	Matches routes of the specified type
`match tag`	Matches tag of a route

** Partial list*

match as-path Command

- Match a BGP autonomous system path access list.

```
Router(config-route-map)#
```

```
match as-path path-list-number
```

- The *path-list-number* is the AS path access list.
  - It can be an integer from 1 to 199.
- The value set by this command overrides global values.

match ip-address Command

- Specify criteria to be matched using ACLs or prefix lists.

```
Router(config-route-map)#
```

```
match ip address {access-list-number | name} [...access-
  list-number | name] | prefix-list prefix-list-name
  [..prefix-list-name]
```

Parameter	Description
access-list-number | name	The number or name of a standard or extended access list to be used to test incoming packets. If multiple access lists are specified, matching any one results in a match.
prefix-list *prefix-list-name*	Specifies the name of a prefix list to be used to test packets. If multiple prefix lists are specified, matching any one results in a match.

set Commands Used in BGP

Command	Description
`set weight`	Sets the BGP weight value
`set local-preference`	Sets the LOCAL-PREF attribute value
`set as-path`	Modifes an AS path for BGP routes
`set origin`	Sets the ORIGIN attribute value
`set metric`	Sets the Multi-Exit_Disc (MED) value
`set community`	Sets the BGP communities attribute
`set automatic-tag`	Computes automatically the tag value
`set ip next-hop`	Indicates which IP address to output packets
`set interface`	Indicates which interface to output packets
`set ip default next-hop`	Indicates which default IP address to use to output packets
`set default interface`	Indicates which default interface to use to output packets

** Partial list*

set weight Command

- Specify the BGP weight for the routing table.

```
Router(config-route-map)#
```

```
set weight number
```

- The *number* is the weight value.
  - It can be an integer ranging from 0 to 65535.
- The implemented weight is based on the first matched AS path.
- Weights assigned with this command override the weights assigned using the **neighbor weight** command.

`set local-preference` Command

- Specify a preference value for the AS path.

```
Router(config-route-map)#
```

```
set local-preference number-value
```

- The *number-value* is the preference value.
  - An integer from 0 to 4294967295.
  - Default 100.

set as-path Command

- Modify an AS path for BGP routes.

```
Router(config-route-map)#
```

```
set as-path {tag | prepend as-path-string}
```

Parameter	Description
tag	Converts the tag of a route into an autonomous system path. Applies only when redistributing routes into BGP.
prepend	Appends the string following the keyword **prepend** to the AS path of the route that is matched by the route map. Applies to inbound and outbound BGP route maps.
as-path-string	AS number to prepend to the AS_PATH attribute. The range of values for this argument is 1 to 65535. Up to 10 AS numbers can be entered.

set metric Command

- Specify a preference value for the AS path.

```
Router(config-route-map)#
```

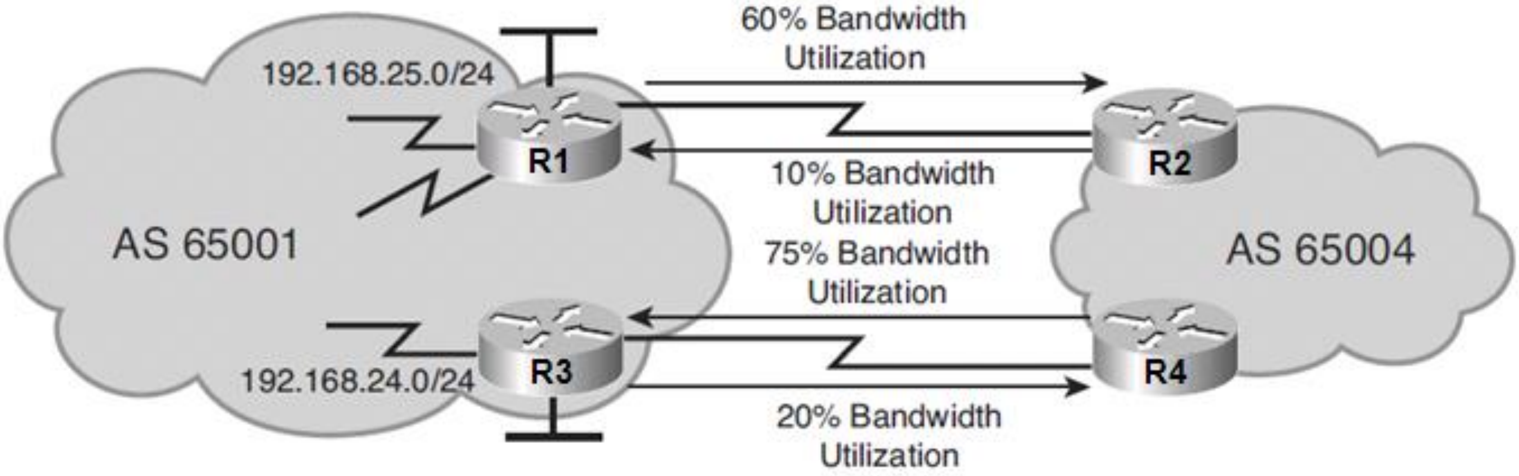
```
set metric metric-value
```

- The *metric-value* is use to set the MED attribute.
  - An integer from 0 to 294967295.

BGP Path Manipulation

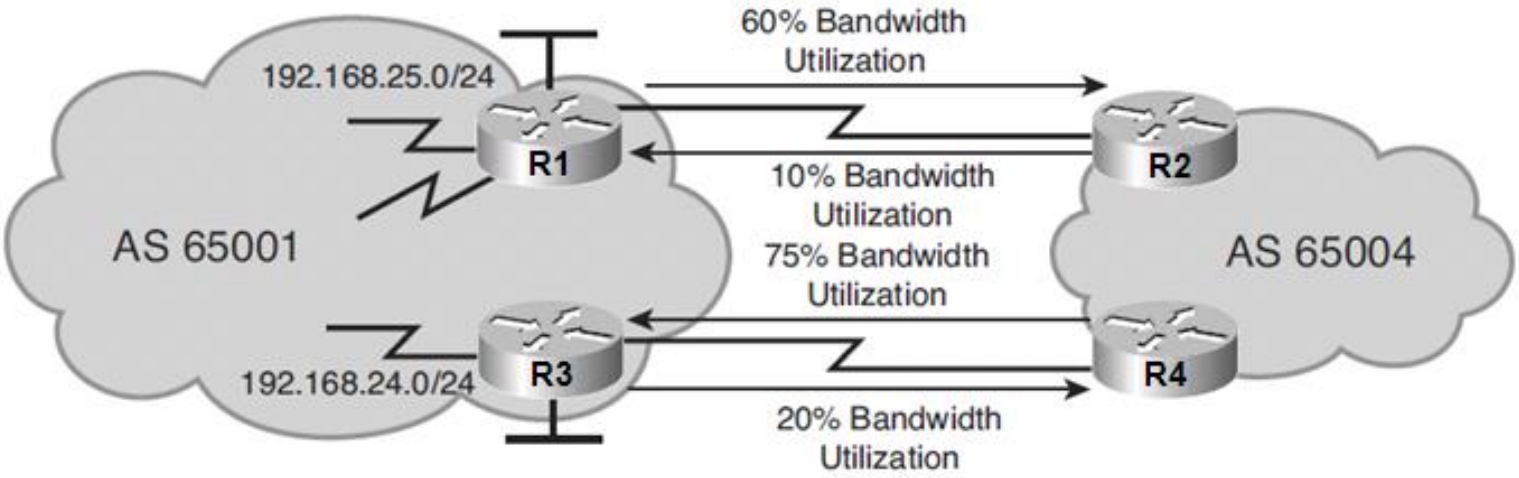
- Unlike IGPs, BGP was never designed to choose the quickest path.
- BGP was designed to manipulate traffic flow to maximize or minimize bandwidth use.

BGP Without Routing Policy Example #1


60% Bandwidth Utilization
192.168.25.0/24
R1
R2
10% Bandwidth Utilization
AS 65001
75% Bandwidth Utilization
AS 65004
R3
R4
192.168.24.0/24
20% Bandwidth Utilization

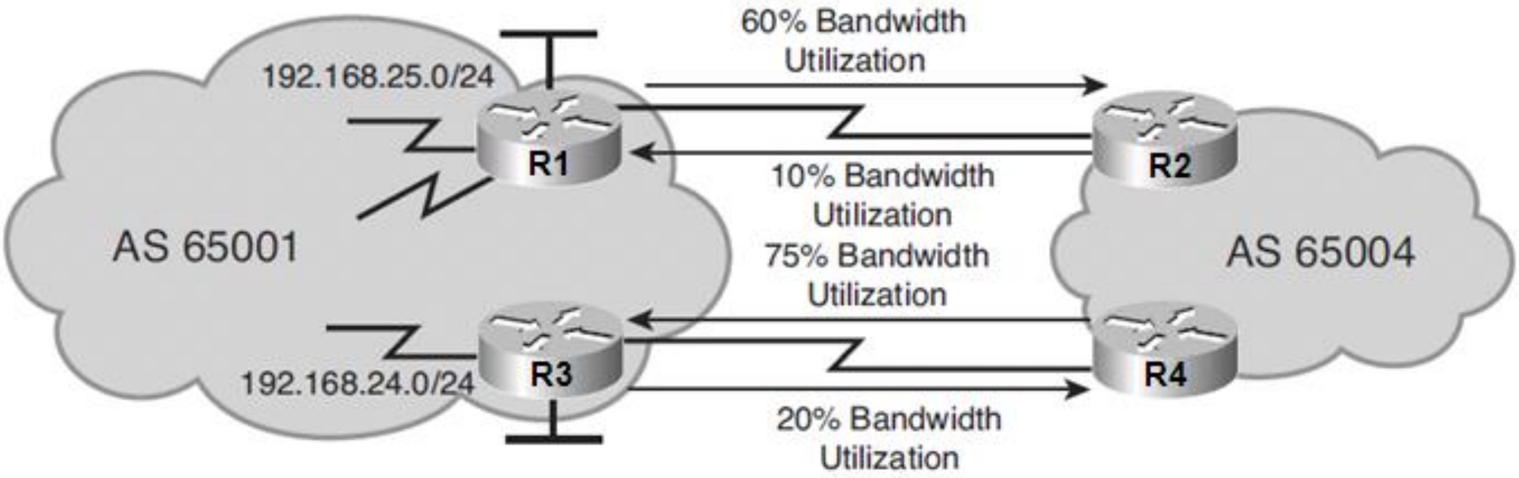
- In this example consider that:
  - R1 is using 60% of its outbound bandwidth to AS 65004.
  - R3 is using 20% of its outbound bandwidth to AS 65004.
  - R2 is using 10% of its outbound bandwidth to AS 65001.
  - R4 is using 75% of its outbound bandwidth to AS 65001.
- Traffic should be diverted using the local preference attribute.
  - The weight attribute could not be used in this scenario since there are two edge routers.

Which traffic should be re-routed?


60% Bandwidth Utilization
192.168.25.0/24
R1
R2
10% Bandwidth Utilization
AS 65001
75% Bandwidth Utilization
AS 65004
R3
R4
192.168.24.0/24
20% Bandwidth Utilization

- To determine which path to manipulate, perform a traffic analysis on Internet-bound traffic by examining the most heavily visited addresses, web pages, or domain names.
  - Examine network management records or accounting information.
- If a heavily accessed traffic pattern is identified, a route map could be used to divert that traffic over the lesser used links

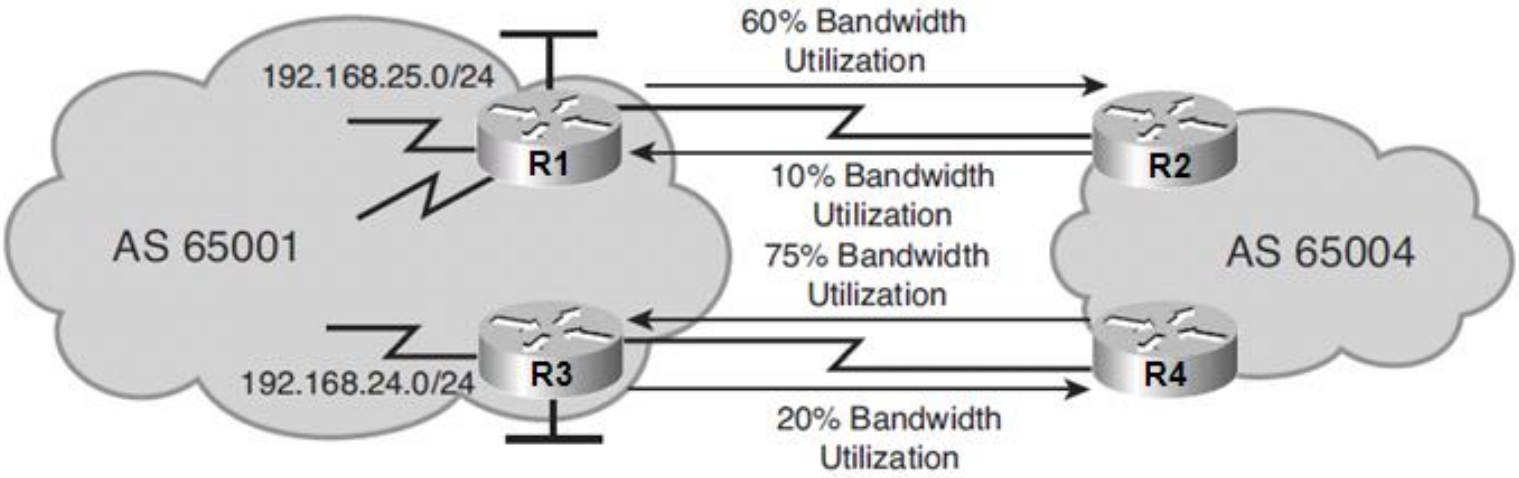
BGP With Routing Policy Example #1



60% Bandwidth Utilization
192.168.25.0/24
R1
R2
10% Bandwidth Utilization
AS 65001
75% Bandwidth Utilization
AS 65004
R3
R4
192.168.24.0/24
20% Bandwidth Utilization

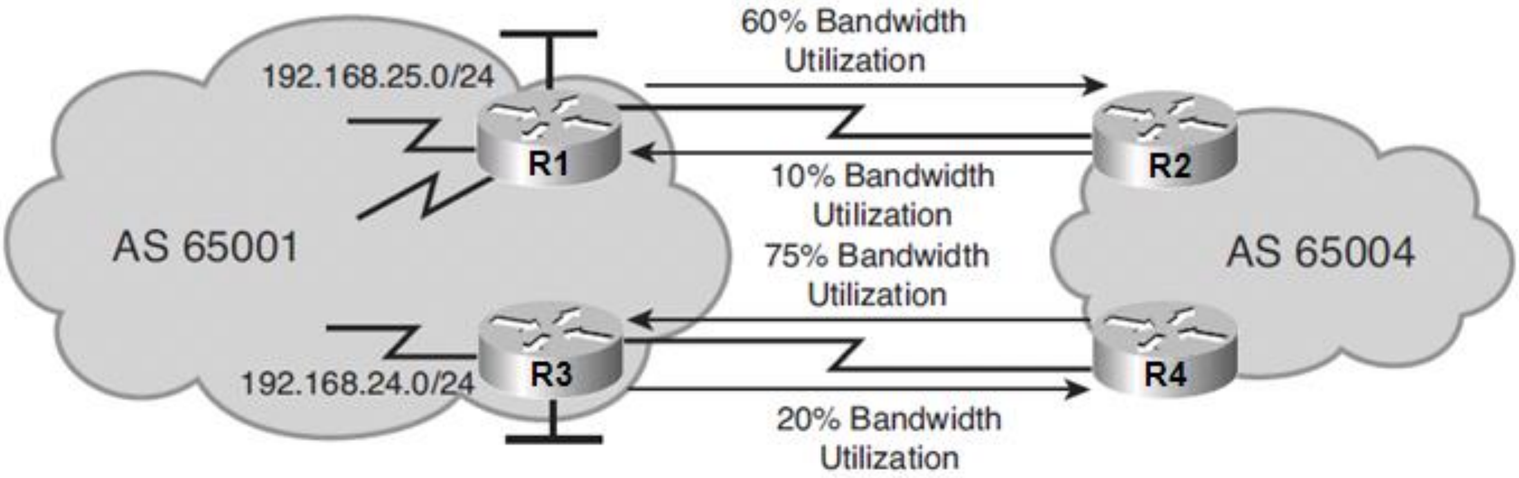
- For example, assume that 35% of all traffic from AS 65001 has been going to http://www.cisco.com.
  - The administrator does a reverse DNS lookup and obtains the Cisco IP address and AS number.
- A route map can be used to change the local preference to manipulate packets destined to Cisco’s network over the less used links.

BGP Routing Policy Example #2


60% Bandwidth Utilization
192.168.25.0/24
R1
R2
10% Bandwidth Utilization
75% Bandwidth Utilization
AS 65001
AS 65004
R3
R4
192.168.24.0/24
20% Bandwidth Utilization

- Notice that the inbound load to R3 (75%) is much higher in bandwidth utilization than the inbound load to R1 (10%).
- The BGP MED attribute can be used to manipulate how traffic enters autonomous system 65001.
- For example, R1 in AS 65001 can announce a lower MED for routes to network 192.168.25.0/24 to AS 65004 than R3 announces.

BGP Routing Policy Example #2



60% Bandwidth Utilization
192.168.25.0/24
R1
R2
10% Bandwidth Utilization
75% Bandwidth Utilization
AS 65001
AS 65004
R3
R4
192.168.24.0/24
20% Bandwidth Utilization

- Keep in mind that the MED is considered a *recommendation* because the receiving autonomous system can override it by manipulating another variable that is considered before the MED is evaluated.
- For example, R2 and R4 in AS 65004 could be configured with their own local preference policy which would override the MED recommendation from AS 65001.

BGP Route Selection Process

1. Prefer highest Weight
2. Prefer highest LOCAL_PREF
3. Prefer locally generated routes
4. Prefer shortest AS_PATH
5. Prefer lowest ORIGIN (IGP < EGP < incomplete)
6. Prefer lowest MED
7. Prefer EBGP over IBGP
8. Prefer routes through closest IGP neighbor
9. Prefer routes with lowest BGP router ID
10. Prefer routes with lowest neighbor IP address

Change the Weight

- The weight attribute is used only when one router is multihomed and determines the best path to leave the AS.
  - Only the local router is influenced.
  - Higher weight routes are preferred.
- There are two ways to alter the route weight:
  - To change the weight for all updates from a neighbor use the neighbor weight router configuration command.
  - To change the weight of specific routes / as path, use route maps.

BGP Route Selection Process

1. Prefer highest Weight
2. Prefer highest LOCAL_PREF
3. Prefer locally generated routes
4. Prefer shortest AS_PATH
5. Prefer lowest ORIGIN (IGP < EGP < incomplete)
6. Prefer lowest MED
7. Prefer EBGP over IBGP
8. Prefer routes through closest IGP neighbor
9. Prefer routes with lowest BGP router ID
10. Prefer routes with lowest neighbor IP address

Changing the Default Weight Example

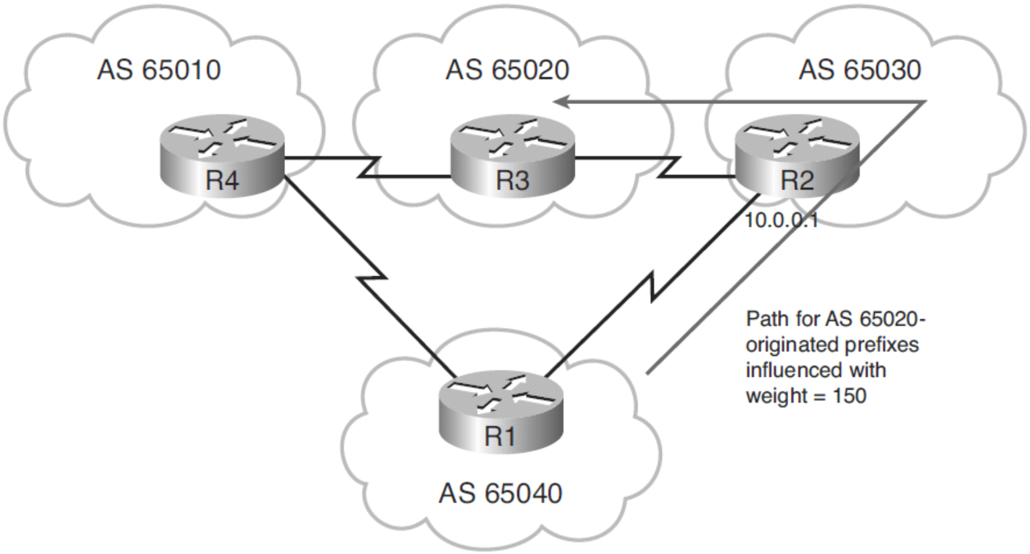
- Assign a default weight to all routes from a peer.

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

```
neighbor {ip-address | peer-group-name} weight number
```

- Routes learned through another BGP peer have a default weight of 0 and routes sourced by the local router have a default weight of 32768.
- The *number* is the weight to assign.
  - Acceptable values are from 0 to 65535.
- The route with the highest weight will be chosen as the preferred route when multiple routes are available to a particular network.
- **Note**: The weights assigned with the **set weight route-map** command override the weights assigned using the **neighbor weight** command.

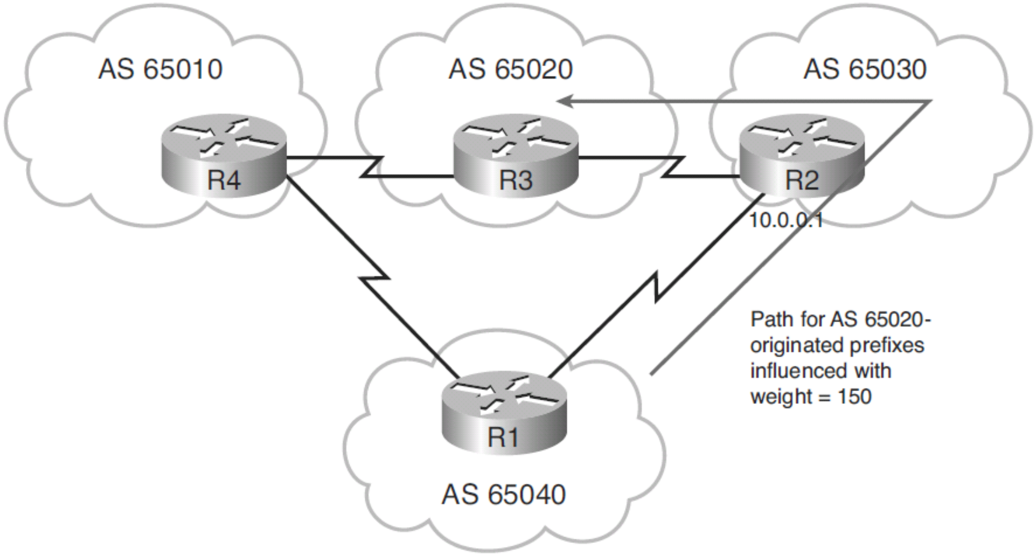
Changing Weight with Route Map Example



Path for AS 65020-originated prefixes influenced with weight = 150

- In this example consider that:
  - The routing policy dictates that for any network originated by AS 65020, use the path to AS 65030 as the primary way out of AS 65040.
  - If R1 needs to access routes connected to R3, then it goes through R2.
- This can be achieved by placing a higher weight (150) on all incoming announcements from AS 65030 (10.0.0.1), which carry the information about the network originated in AS 65020.

Changing Weight with Route Map Example



Path for AS 65020-originated prefixes influenced with weight = 150

```
R1(config)# route-map SET-WEIGHT permit 10
R1(config-route-map)# match as-path 10
R1(config-route-map)# set weight 150
R1(config-route-map)#
R1(config-route-map)# route-map SET-WEIGHT permit 20
R1(config-route-map)# set weight 100
R1(config-route-map)# exit
R1(config)# ip as-path access-list 10 permit _65020$
R1(config)#
R1(config)# router bgp 65040
R1(config-router)# neighbor 10.0.0.1 remote-as 65030
R1(config-router)# neighbor 10.0.0.1 route-map SET-WEIGHT in
```

Configure an Autonomous System ACL

- Configure an autonomous system path filter.

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

```
ip as-path access-list acl-number {permit | deny} regexp
```

- Similar to an IP ACL, this command is used to configure an AS path filter using a regular expression .
- The `acl-number` is a value from 1 to 500 that specifies the AS_PATH access list number.
- The `regexp` regular expression defines the AS-path filter.

Regular Expression Syntax

- **Atom**: A single character.
  - `.` matches any single character.
  - `^` matches the start of the input string.
  - `$` matches the end of the input string.
  - `\` matches the character.
- **Piece**: one of these symbols
  - `*` matches 0 or more sequences of the atom.
  - `+` matches 1 or more sequences of the atom.
  - `?` matches the atom or the null string.
- **Branch**: 1 or more concatenated pieces.
- **Range**: A sequence of characters within square brackets.
  - Example is [abcd].

Regular Expression Examples

Regular Expression	Resulting Expression
`a*`	Expression indicates any occurrence of the letter "a", which includes none
`a+`	indicates that at least one occurrence of the letter "a" must be present
`ab?a`	Expression matches "aa" or "aba".
`_100_`	Expression means via AS100.
`_100$`	Expression indicates an origin of AS100.
`^100 .*`	Expression indicates transmission from AS100
`^$`	Expression indicates origination from this AS
`^([0-9]+)(_\1)*$`	Expression indicates AS-PATH prepend

Change the Local Preference

- The local preference is used only within an AS (between IBGP speakers) to determine the best path to leave the AS.
  - Higher values are preferred.
  - The local preference is set to 100 by default.
- There are two ways to alter the local preference:
  - To change the default local-preference for all routes advertised by the router use the **`bgp default local-preference`** *`value`* router configuration command.
  - To change the local-preference of specific routes / as path, use route maps.

BGP Route Selection Process

1. Prefer highest Weight
2. Prefer highest LOCAL_PREF
3. Prefer locally generated routes
4. Prefer shortest AS_PATH
5. Prefer lowest ORIGIN (IGP < EGP < incomplete)
6. Prefer lowest MED
7. Prefer EBGP over IBGP
8. Prefer routes through closest IGP neighbor
9. Prefer routes with lowest BGP router ID
10. Prefer routes with lowest neighbor IP address

Setting Default Local Preference Example

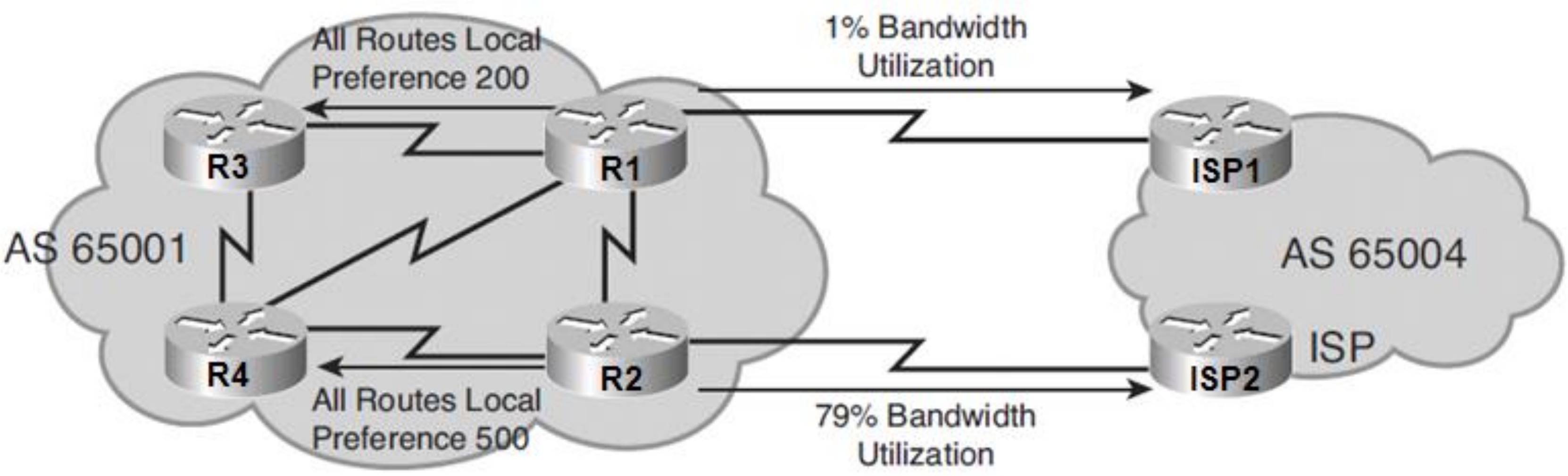
- Change the default local preference for outgoing routes.

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

```
bgp default local-preference number
```

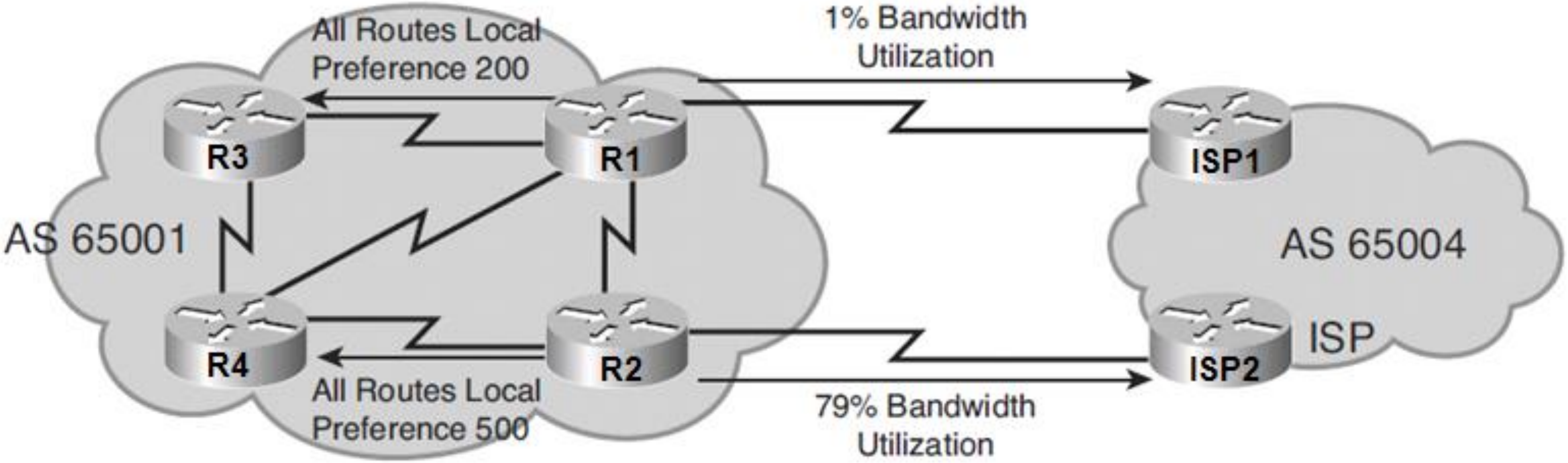
- The local preference attribute applies a degree of preference to a route during the BGP best path selection process.
  - The attribute is exchanged only between iBGP peers.
  - The route with the highest local preference is preferred.
- The *number* is the local preference value from 0 to 4294967295.
  - Cisco IOS software applies a local preference value of 100.
- Note: The local preference assigned with the **set local-preference route-map** command override the weights assigned using this command.

Setting Default Local Preference Example


All Routes Local Preference 200
1% Bandwidth Utilization
R3
R1
ISP1
AS 65001
AS 65004
R4
R2
ISP2
ISP
All Routes Local Preference 500
79% Bandwidth Utilization

- The BGP routing policy in this example dictates that:
  - The default local preference for all routes on R1 should be set to 200.
  - The default local preference for all routes on R2 should be set to 500.

Setting Default Local Preference Example



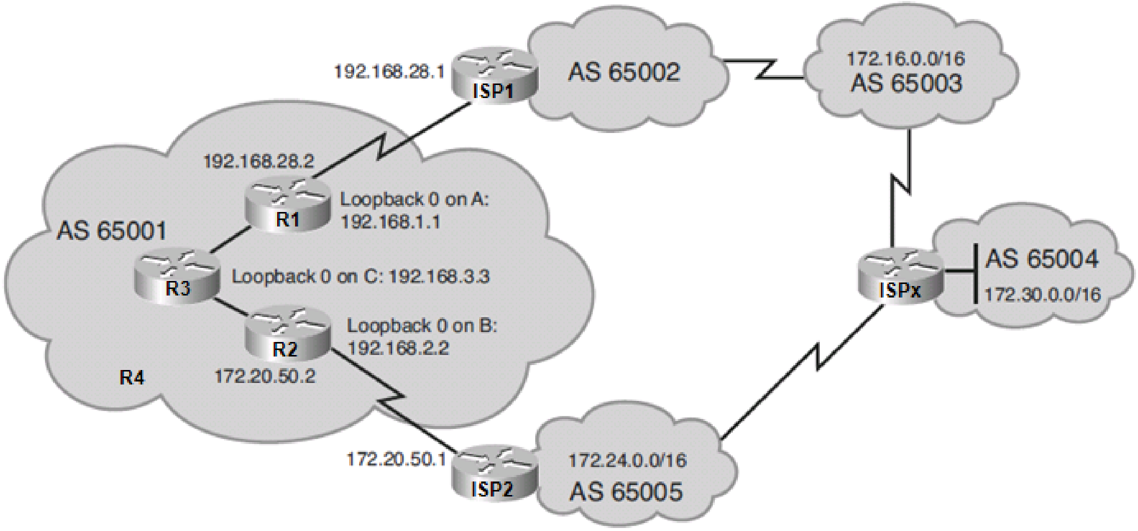
All Routes Local Preference 200
1% Bandwidth Utilization
R3
R1
ISP1
AS 65001
AS 65004
R4
R2
ISP2
ISP
All Routes Local Preference 500
79% Bandwidth Utilization

```
R1(config)# router bgp 65001
R1(config-router)# bgp default local-preference 200
R1(config-router)#
```

```
R2(config)# router bgp 65001
R2(config-router)# bgp default local-preference 500
R2(config-router)#
```

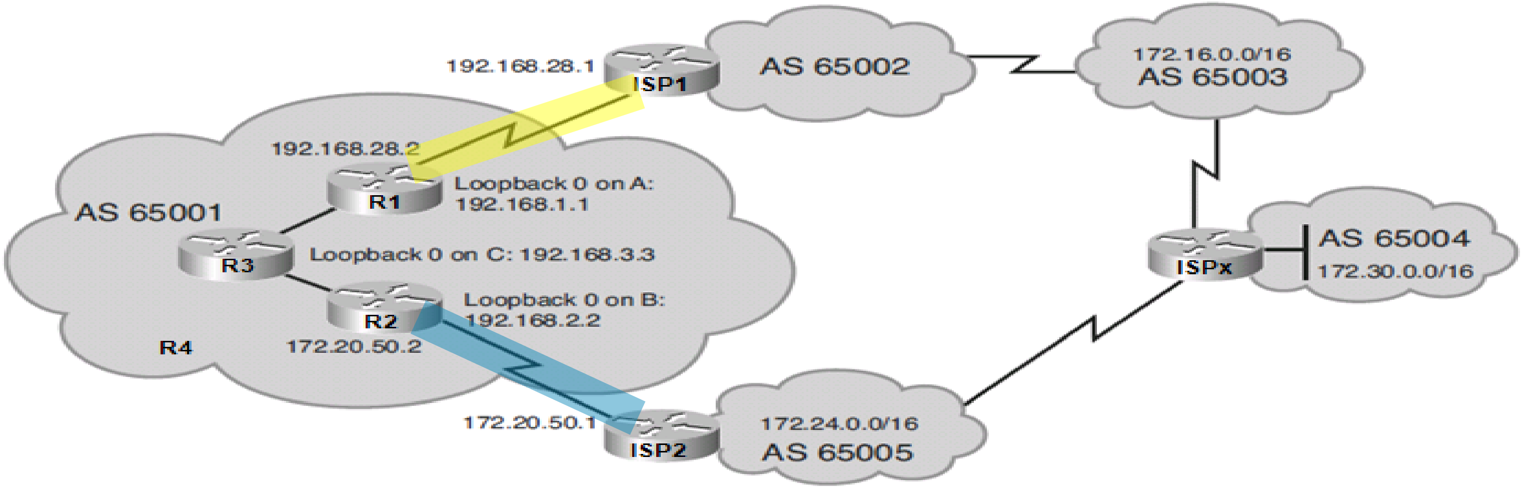
- The resulting configuration makes the IBGP routers in AS 65001 send all Internet bound traffic to R2, but the R1 to ISP1 link is underutilized.
  - Route maps could be configured to select specific routes to have a higher local preference.

Local Preference and Route Map Example



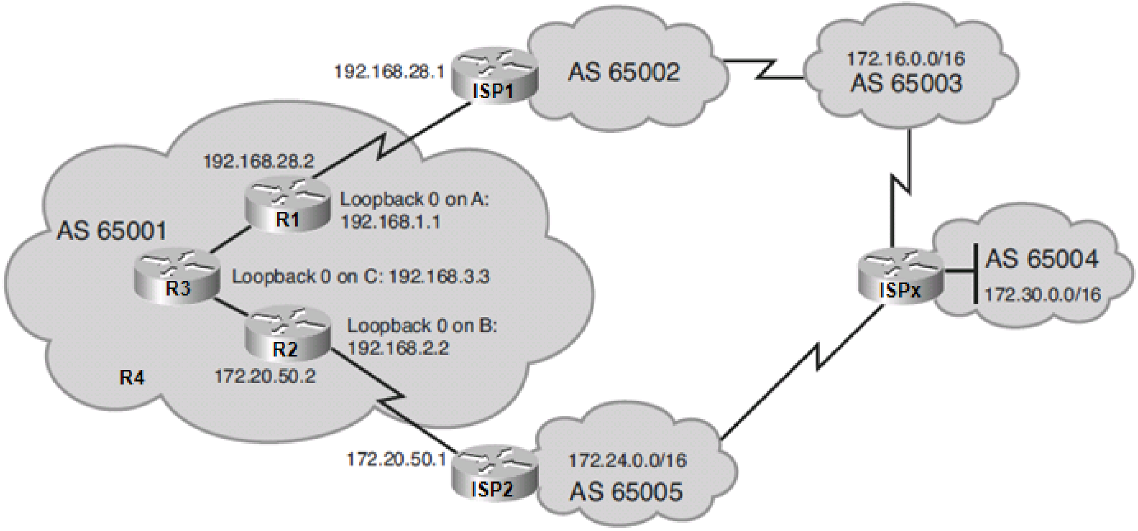
- The BGP routing policy results in the following:
  - All routes have a weight of 0 and a default local preference of 100.
  - BGP uses the shortest AS-path to select the best routes as follows:
    - For network 172.16.0.0, the shortest AS-path is through ISP1.
    - For network 172.24.0.0, the shortest AS-path is through ISP2.
    - For network 172.30.0.0, the shortest AS-path is through ISP2.

Local Preference and Route Map Example



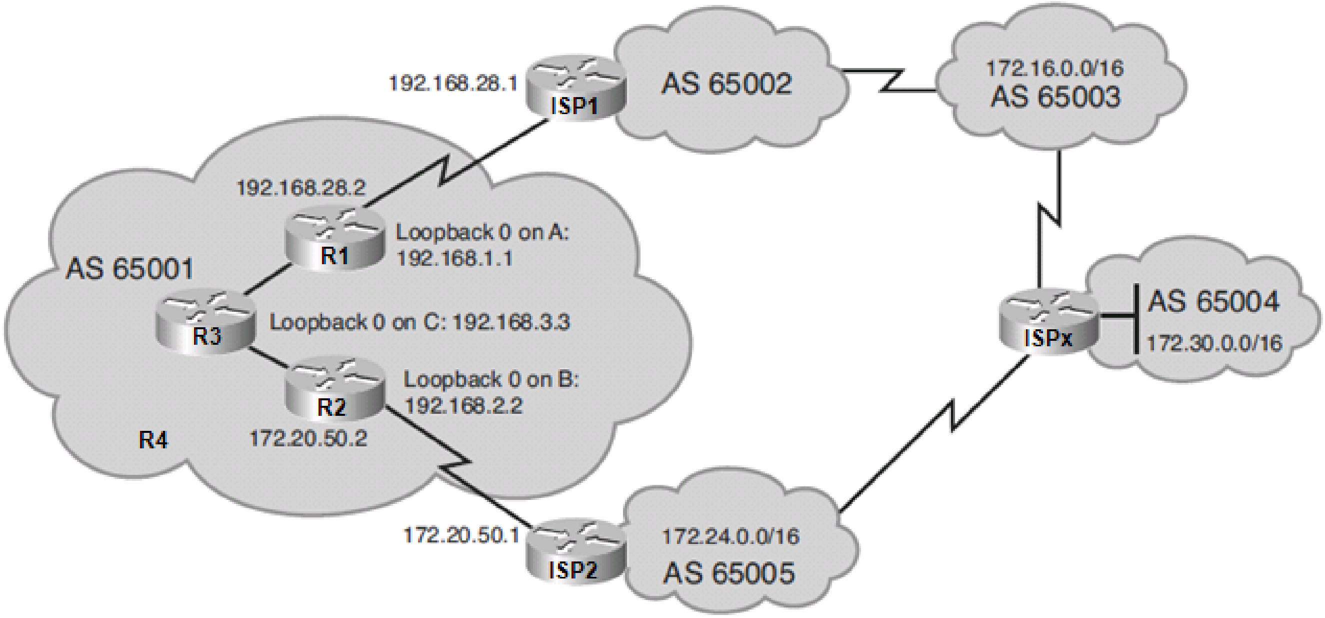
```
R3# show ip bgp
BGP table version is 7, local router ID is 192.168.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  r
   RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
Network             Next Hop      Metric LocPrf  Weight Path
* i172.16.0.0       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
```

Local Preference and Route Map Example



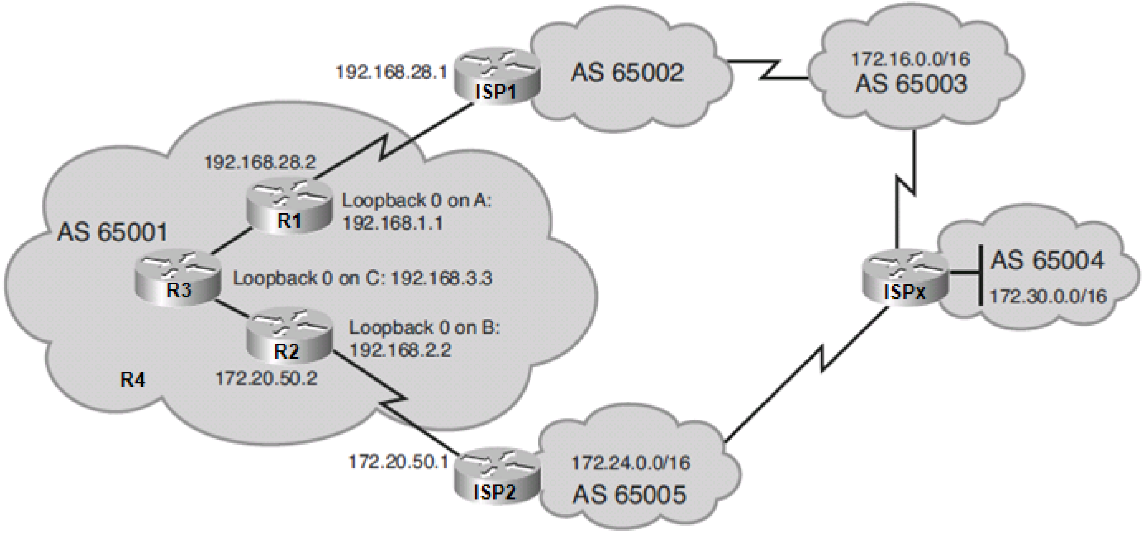
- A traffic analysis reveals the following traffic patterns:
  - 10% of traffic flows from R1 to ISP1 to network 172.16.0.0.
  - 50% of Internet traffic flow from R2 to ISP2 to networks network 172.24.0.0 and network 172.30.0.0.
  - The remaining 40 percent is going to other destinations.
- A solution is to use route maps to divert traffic to 172.30.0.0 through R1.

Local Preference and Route Map Example



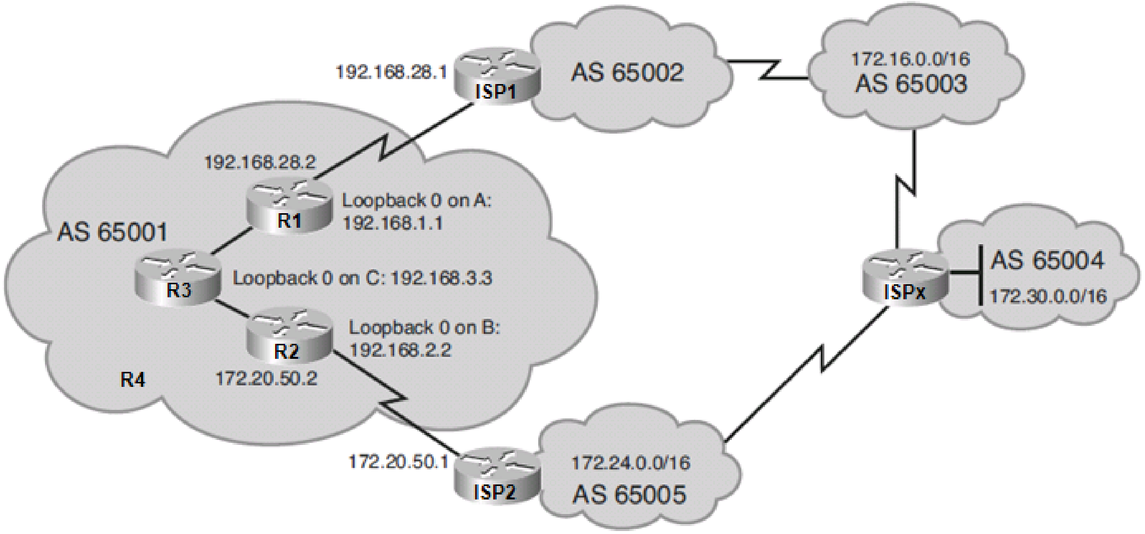
```
R1(config)# access-list 65 permit 172.30.0.0 0.0.255.255
R1(config)#
R1(config)# route-map LOCAL_PREF permit 10
R1(config-route-map)# match ip address 65
R1(config-route-map)# set local-preference 400
R1(config-route-map)#
R1(config-route-map)# route-map LOCAL_PREF permit 20
R1(config-route-map)# exit
R1(config)#
```

Local Preference and Route Map Example



```
R1(config)# router bgp 65001
R1(config-router)# neighbor 192.168.2.2 remote-as 65001
R1(config-router)# neighbor 192.168.2.2 update-source loopback0
R1(config-router)# neighbor 192.168.3.3 remote-as 65001
R1(config-router)# neighbor 192.168.3.3 update-source loopback0
R1(config-router)# neighbor 192.168.28.1 remote-as 65002
R1(config-router)# neighbor 192.168.28.1 route-map LOCAL_PREF in
R1(config-router)# exit
R1(config)#
```

Local Preference and Route Map Example



```
R3# show ip bgp
BGP table version is 7, local router ID is 192.168.3.3
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
```

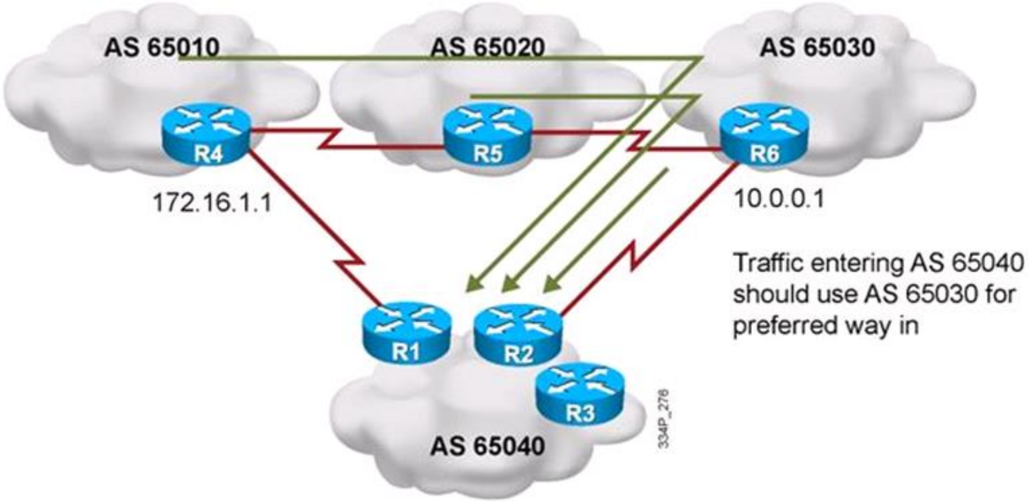
Modifying the AS Path

- By default, if no BGP path selection tools are configured to influence traffic flow (i.e. weight, local-preference), BGP uses the shortest AS path, regardless of available bandwidth.
- To influence the path selection based on the AS_PATH, configure AS-path prepending.
  - The AS path is extended with multiple copies of the AS number of the sender making it appear longer.

BGP Route Selection Process

1. Prefer highest Weight
2. Prefer highest LOCAL_PREF
3. Prefer locally generated routes
4. Prefer shortest AS_PATH
5. Prefer lowest ORIGIN (IGP < EGP < incomplete)
6. Prefer lowest MED
7. Prefer EBGP over IBGP
8. Prefer routes through closest IGP neighbor
9. Prefer routes with lowest BGP router ID
10. Prefer routes with lowest neighbor IP address

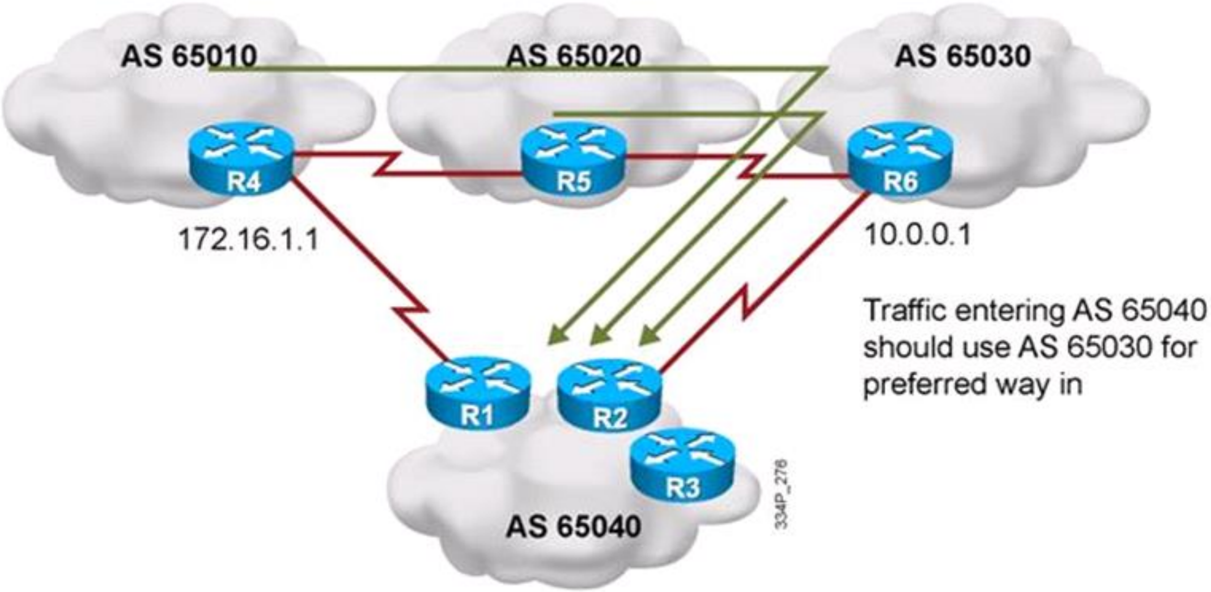
Modifying the AS Path Example



Traffic entering AS 65040 should use AS 65030 for preferred way in

- The BGP routing policy in this example dictates that:
  - Traffic entering AS 65040 should be through R6 in AS 65030 and not R4 in AS 65010.
- One way to do this is make R1 advertise the AS 65040 networks with a less desirable AS path by configuring AS-path prepending.

Modifying the AS Path Example



Traffic entering AS 65040 should use AS 65030 for preferred way in

```
R1(config)# route-map SET-AS-PATH permit 10
R1(config-route-map)# set as-path prepend 65040 65040 65040
R1(config-route-map)# exit
R1(config)# router bgp 65040
R1(config-router)# neighbor 172.16.1.1 remote-as 65010
R1(config-router)# neighbor 172.16.1.1 route-map SET-AS-PATH out
R1(config-router)# exit
R1(config)#
```

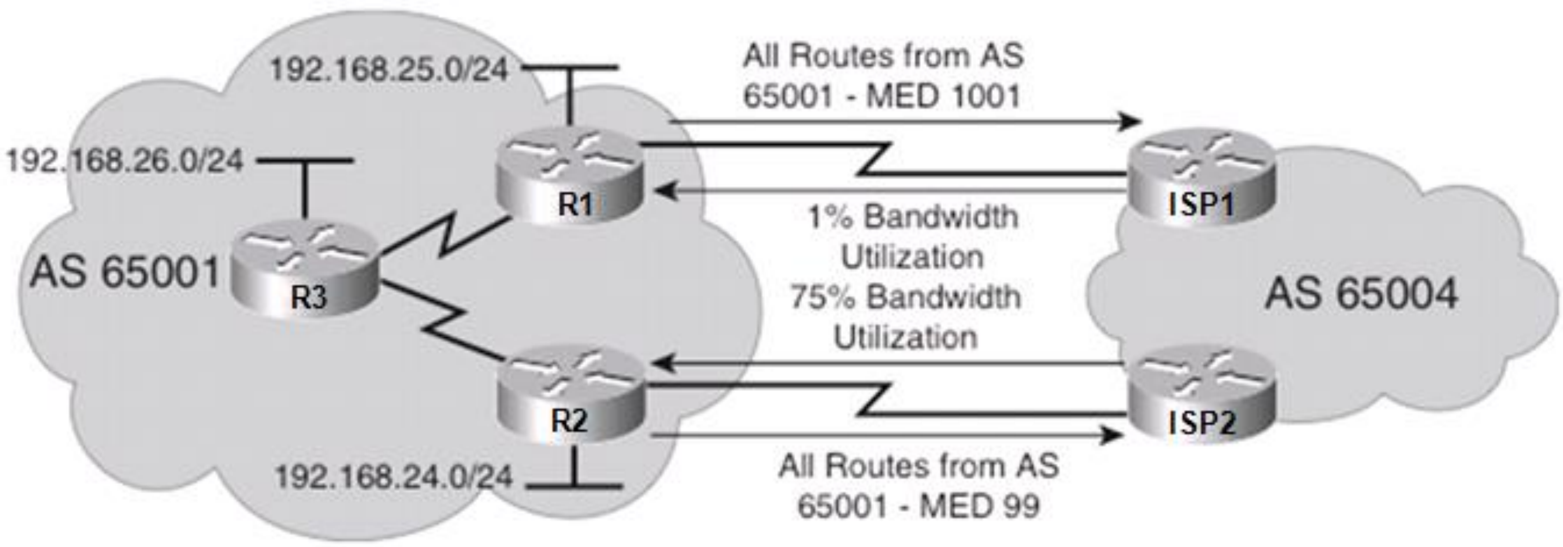
Setting the MED

- MED is used to decide how to enter an AS when multiple paths exist.
  - When comparing MED values for the same destination network in the BGP path-selection process, the lowest MED value is preferred.
  - Default is 0.
- However, because MED is evaluated late in the BGP path-selection process, it usually has no influence.
- There are two ways to alter the MED:
  - To change the MED for all routes use the **`default-metric`** router configuration command.
  - To change the MED of specific routes / as path, use route maps.

BGP Route Selection Process

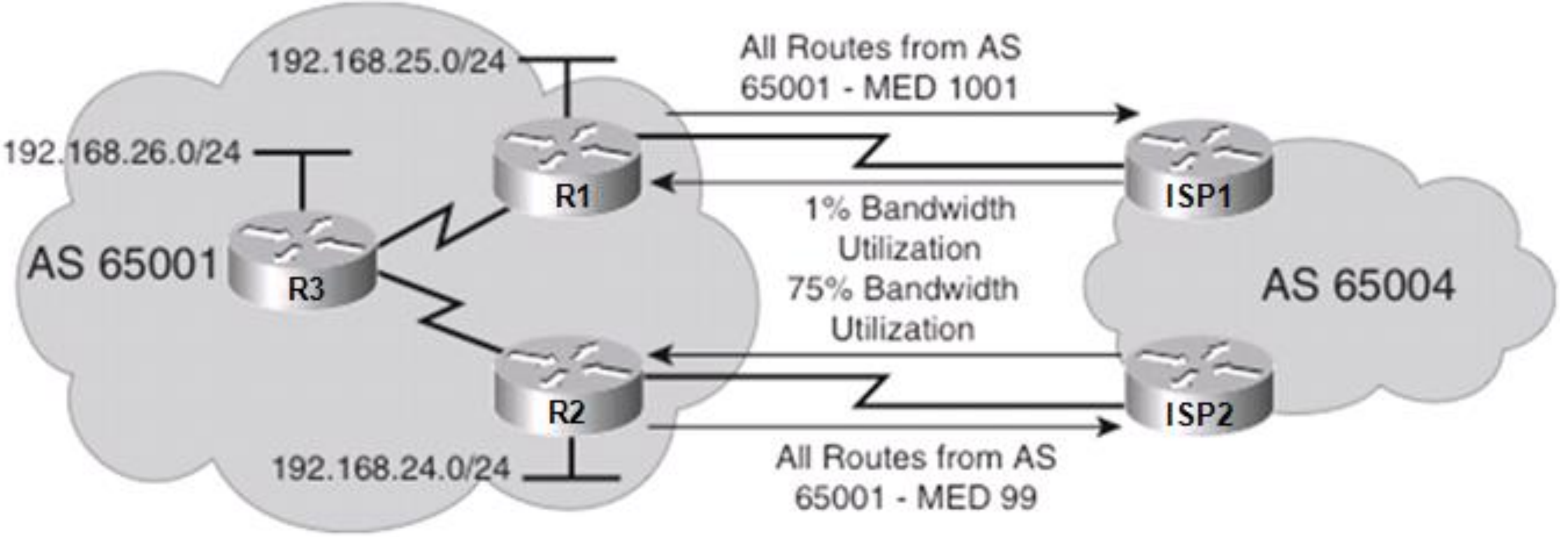
1. Prefer highest Weight
2. Prefer highest LOCAL_PREF
3. Prefer locally generated routes
4. Prefer shortest AS_PATH
5. Prefer lowest ORIGIN (IGP < EGP < incomplete)
6. Prefer lowest MED
7. Prefer EBGP over IBGP
8. Prefer routes through closest IGP neighbor
9. Prefer routes with lowest BGP router ID
10. Prefer routes with lowest neighbor IP address

Setting the Default MED Example



- The BGP routing policy in this example dictates that:
  - The default MED of R1 should be changed to 1001.
  - The default MED of R2 should be changed to 99.

Setting the Default MED Example

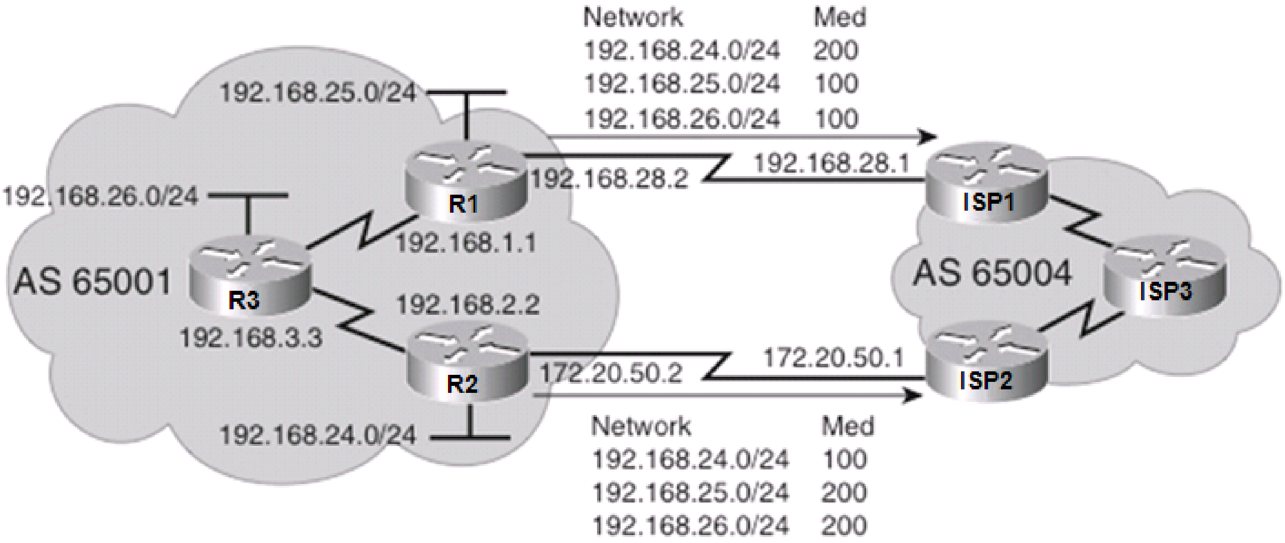


```
R1(config)# router bgp 65001
R1(config-router)# default metric 1001
R1(config-router)#
```

```
R2(config)# router bgp 65001
R2(config-router)# default metric 99
R2(config-router)#
```

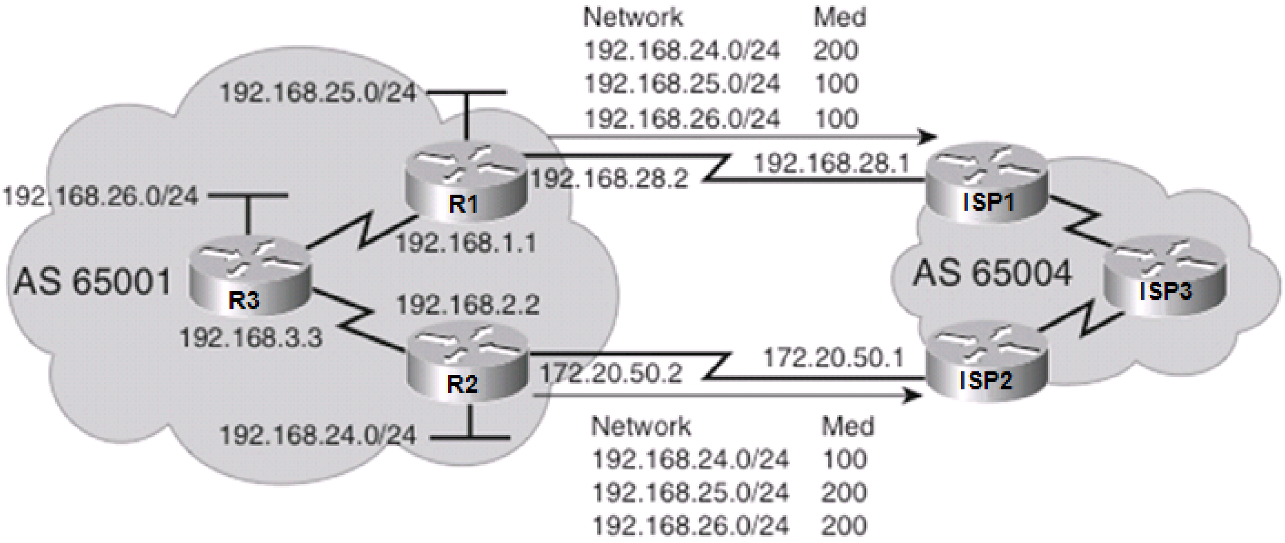
- The results are that the inbound bandwidth utilization on:
  - R1 to ISP1 link has decreased to almost nothing except for BGP routing updates.
  - R2 to ISP2 link has increased due to all returning packets from AS 65004.
  - A better solution is to have route maps configured that will make some networks have a lower MED through R1 and other networks to have a lower MED through R2.

Setting the MED with Route Maps Example



```
R1(config)# access-list 66 permit 192.168.25.0 0.0.0.255
R1(config)# access-list 66 permit 192.168.26.0 0.0.0.255
R1(config)#
R1(config)# route-map MED-65004 permit 10
R1(config-route-map)# match ip address 66
R1(config-route-map)# set metric 100
R1(config-route-map)#
R1(config-route-map)# route-map MED-65004 permit 100
R1(config-route-map)# set metric 200
R1(config-route-map)# exit
R1(config)#
```

Setting the MED with Route Maps Example

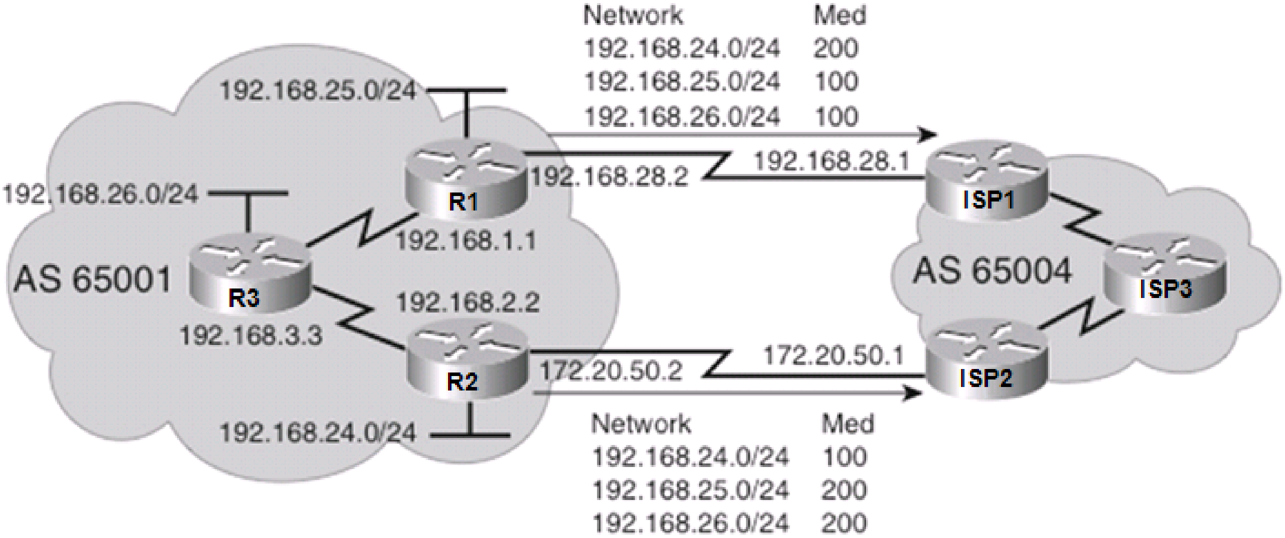


Network	Med
192.168.24.0/24	200
192.168.25.0/24	100
192.168.26.0/24	100

Network	Med
192.168.24.0/24	100
192.168.25.0/24	200
192.168.26.0/24	200

```
R1(config)# router bgp 65001
R1(config-router)# neighbor 192.168.2.2 remote-as 65001
R1(config-router)# neighbor 192.168.2.2 update-source loopback0
R1(config-router)# neighbor 192.168.3.3 remote-as 65001
R1(config-router)# neighbor 192.168.3.3 update-source loopback0
R1(config-router)# neighbor 192.168.28.1 remote-as 65004
R1(config-router)# neighbor 192.168.28.1 route-map MED-65004 out
R1(config-router)#exit
```

Setting the MED with Route Maps Example

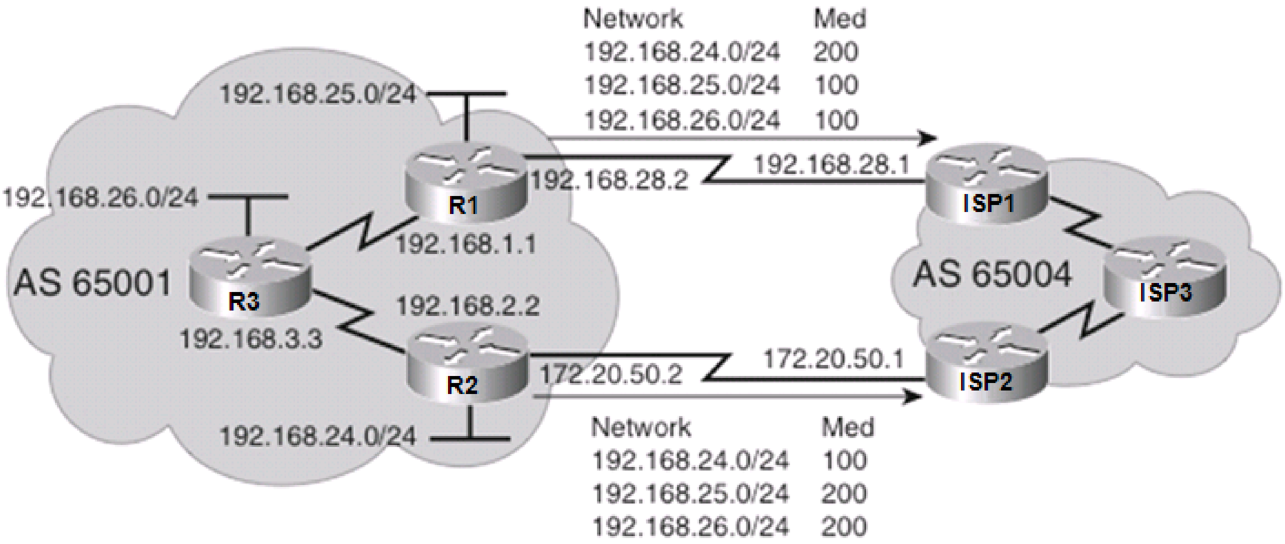


Network	Med
192.168.24.0/24	200
192.168.25.0/24	100
192.168.26.0/24	100

Network	Med
192.168.24.0/24	100
192.168.25.0/24	200
192.168.26.0/24	200

```
R2(config)# access-list 66 permit 192.168.24.0 0.0.0.255
R2(config)#
R2(config)# route-map MED-65004 permit 10
R2(config-route-map)# match ip address 66
R2(config-route-map)# set metric 100
R2(config-route-map)#
R2(config-route-map)# route-map MED-65004 permit 100
R2(config-route-map)# set metric 200
R2(config-route-map)# exit
R2(config)#
```

Setting the MED with Route Maps Example

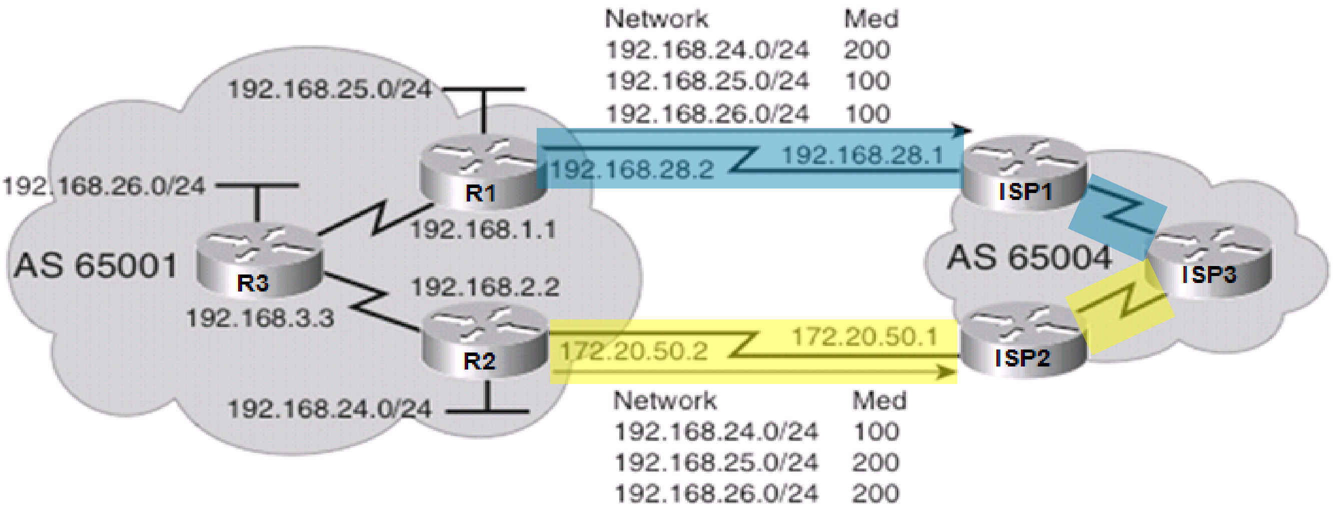


Network	Med
192.168.24.0/24	200
192.168.25.0/24	100
192.168.26.0/24	100

Network	Med
192.168.24.0/24	100
192.168.25.0/24	200
192.168.26.0/24	200

```
R2(config)# router bgp 65001
R2(config-router)# neighbor 192.168.1.1 remote-as 65001
R2(config-router)# neighbor 192.168.1.1 update-source loopback0
R2(config-router)# neighbor 192.168.3.3 remote-as 65001
R2(config-router)# neighbor 192.168.3.3 update-source loopback0
R2(config-router)# neighbor 172.20.50.1 remote-as 65004
R2(config-router)# neighbor 172.20.50.1 route-map MED-65004 out
R2(config-router)# exit
R2(config)#
```

Setting the MED with Route Maps Example



Network	Med
192.168.24.0/24	200
192.168.25.0/24	100
192.168.26.0/24	100

Network	Med
192.168.24.0/24	100
192.168.25.0/24	200
192.168.26.0/24	200

```
ISP3# show ip bgp
BGP table version is 7, local router ID is 192.168.1.1
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
```

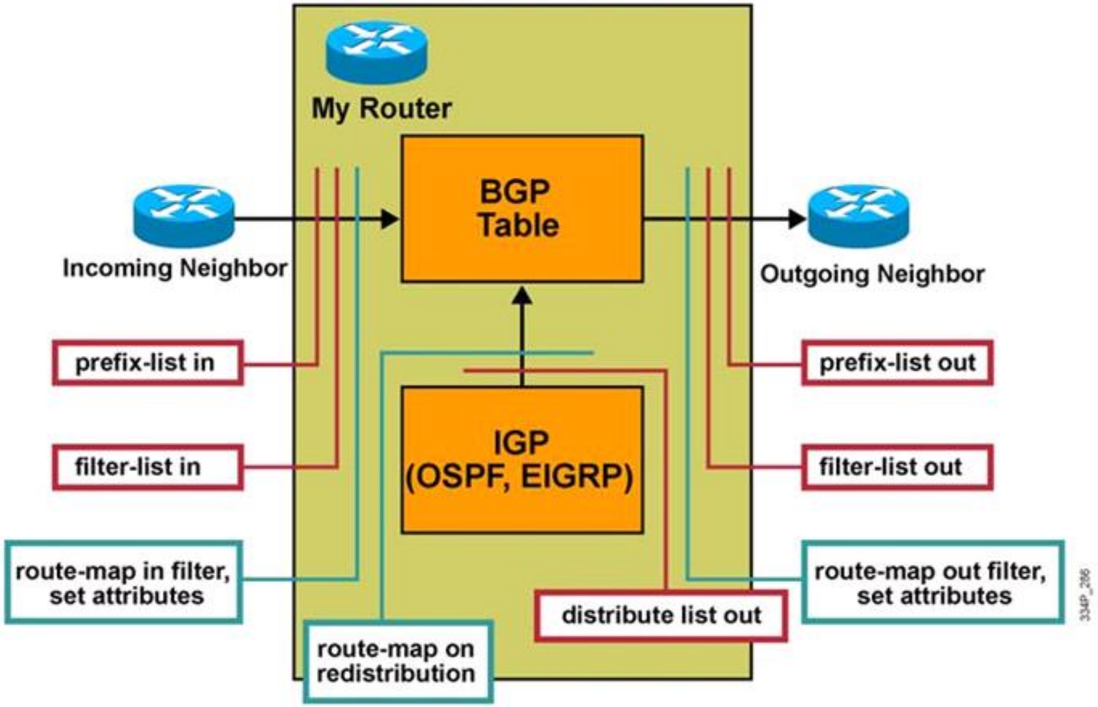
Filtering BGP Routing Updates

Filtering BGP Routing Updates

- BGP can potentially receive a high number of routing updates.
  - To optimize BGP configuration, route filtering may be applied.
- Filtering includes:
  - Filter lists
  - Prefix lists
  - Route maps

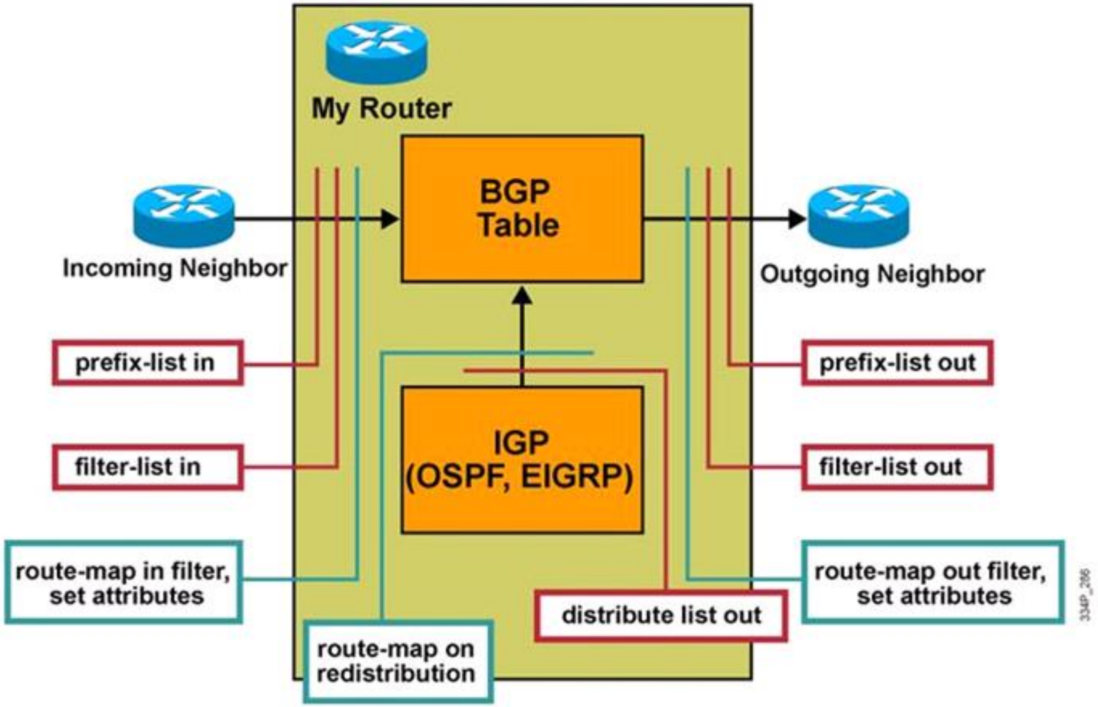
Filtering BGP Routing Updates

- Incoming routes are subject to prefix lists, filter-lists, and route maps before they will be accepted into the BGP table.
  - Similarly, outgoing routes must pass the outgoing route-maps, filter list, and prefix list before they will be transmitted to the neighbor.



Filtering BGP Routing Updates

- If redistributing from an IGP into BGP, the routes must successfully pass any prefix list or route map applied to the redistribution process before the route is injected into the BGP table.



Apply a BGP Filter To Routes

- Apply a filter list to routes from or to a neighbor.

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

```
neighbor {ip-address | peer-group-name} filter-list
  access-list-number {in | out}
```

Parameter	Description
ip-address	IP address of the BGP neighbor.
peer-group-name	Name of a BGP peer group.
access-list-number	Number of an AS-path access list.
in	Access list is applied to incoming routes.
out	Access list is applied to outgoing routes.

Planning BGP Filtering Using Prefix Lists

- When planning BGP filter configuration using prefix lists, the following steps should be documented:
  - Define the traffic filtering requirements, including the following:
    - Filtering updates
    - Controlling redistribution
  - Configure the **`ip prefix-list`** statements.
  - Apply the prefix list to filter inbound or outbound updates using the **`neighbor prefix-list`** router configuration command.

Configure a Prefix List

- Define a prefix list.

```
Router(config)#
```

```
ip prefix-list {list-name | list-number} [seq seq-value] {deny |
  permit} network/length [ge ge-value] [le le-value]
```

Parameter	Description
list-name	The name of the prefix list that will be created (it is case sensitive).
list-number	The number of the prefix list that will be created.
seq *seq-value*	A 32-bit sequence number of the **prefix-list** statement. Default sequence numbers are in increments of 5 (5, 10, 15, and so on).
deny | **permit**	The action taken when a match is found.
network / *length*	The prefix to be matched and the length of the prefix. The network is a 32-bit address; the length is a decimal number.
ge *ge-value*	(Optional) The range of the prefix length to be matched. The range is assumed to be from *ge-value* to 32 if only the **ge** attribute is specified.
le *le-value*	(Optional) The range of the prefix length to be matched. The range is assumed to be from length to *le-value* if only the **le** attribute is specified.

Apply a Prefix List

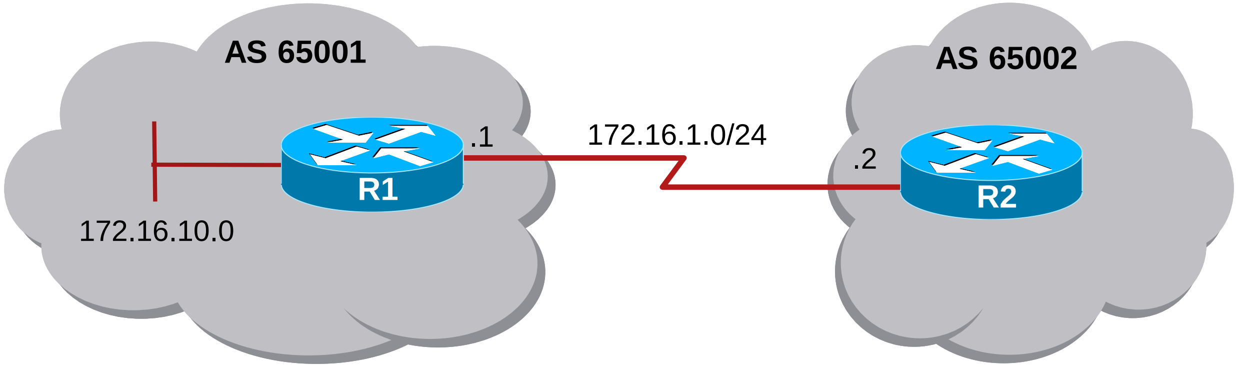
- Apply a prefix list to routes from or to a neighbor.

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

```
neighbor {ip-address | peer-group-name} prefix-list prefix-list-name
   {in | out}
```

Parameter	Description
ip-address	IP address of the BGP neighbor.
peer-group-name	Name of a BGP peer group.
prefix-list-name	Name of a prefix list.
in	Prefix list is applied to incoming advertisements.
out	Prefix list is applied to outgoing advertisements.

BGP Filtering Using Prefix Lists Example

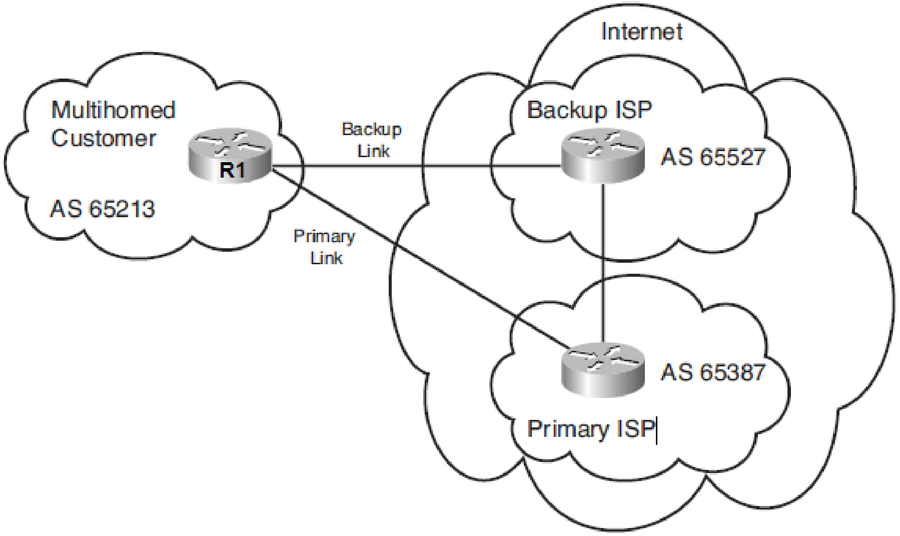


```
R1(config)# ip prefix-list ANY-8to24-NET permit 0.0.0.0/0 ge 8 le 24
R1(config)# router bgp 65001
R1(config-router)# neighbor 172.16.1.2 remote-as 65002
R1(config-router)# neighbor 172.16.1.2 prefix-list ANY-8to24-NET in
R1(config-router)# end
R1#
R1# show ip prefix-list detail ANY-8to24-NET
ip prefix-list ANY-8to24-NET:
Description: test-list
count: 1, range entries: 1, sequences: 10 - 10, refcount: 3
seq 10 permit 0.0.0.0/0 ge 8 le 24 (hit count: 0, refcount: 1)
```

Planning BGP Filtering Using Route Maps

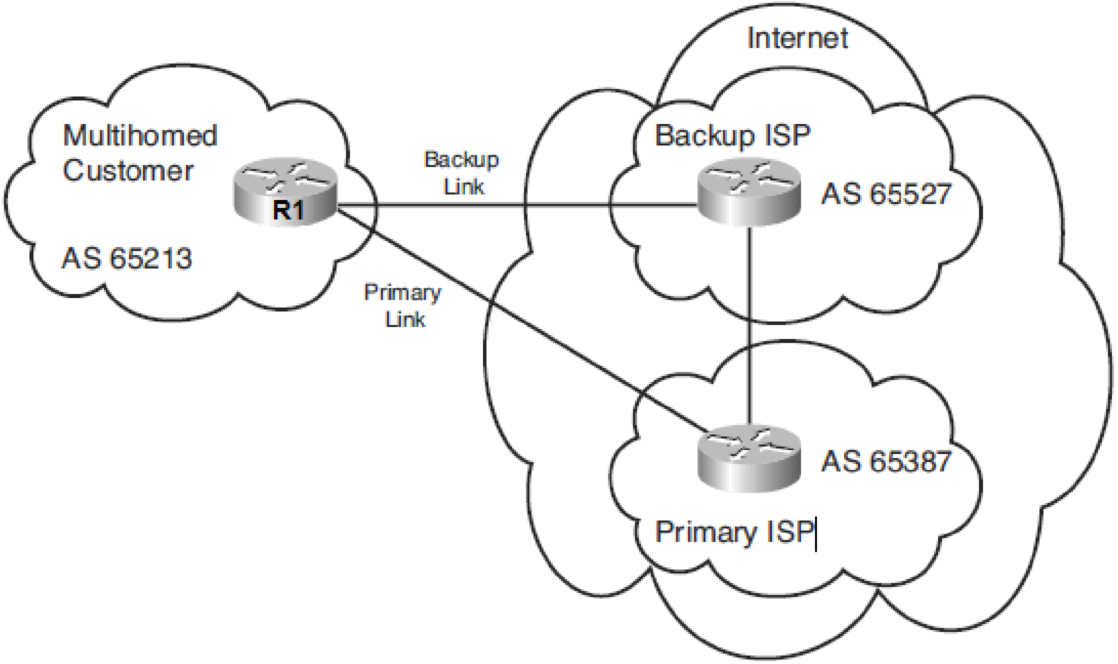
- When planning BGP filter configuration using route maps, the following steps should be documented:
  - Define the route map, including:
    - The **`match`** statements
    - The **`set`** statements
  - Configure route filtering using the route map.

BGP Filtering Using Route Maps



```
R1(config)# ip as-path access-list 10 permit _65387$
R1(config)# ip prefix-list DEF-ONLY seq 10 permit 0.0.0.0/0
R1(config)#
R1(config)# route-map FILTER permit 10
R1(config-route-map)# match ip address prefix-list DEF-ONLY
R1(config-route-map)# match as-path 10
R1(config-route-map)# set weight 150
R1(config-route-map)#
R1(config-route-map)# route-map FILTER permit 20
R1(config-route-map)# match ip address prefix-list DEF-ONLY
R1(config-route-map)# set weight 100
R1(config-route-map)# exit
```

BGP Filtering Using Route Maps



```
R1(config)# router bgp 65213
R1(config-router)# neighbor 10.2.3.4 remote-as 65527
R1(config-router)# neighbor 10.2.3.4 route-map FILTER in
R1(config-router)# neighbor 10.4.5.6 remote-as 65387
R1(config-router)# neighbor 10.4.5.6 route-map FILTER in
R1(config-router)#
```

Chapter 6 Summary

The chapter focused on the following topics:

- BGP terminology and concepts, including:
  - BGP’s use between autonomous systems.
  - The range of private AS numbers: 64512 to 65535.
  - Requirements for Enterprise connection to an ISP including public IP address space, link type and bandwidth, routing protocol, and connectivity redundancy.
- The four connection link type options: circuit emulation, MPLS VPNs, static routes, and BGP.
- The four connection redundancy types: Single-homed, Dual-homed, Multihomed, Dual-multihomed.
- BGP neighbor (peer) relationships:
  - IBGP is when BGP runs between routers in the same AS
  - EBGP is when BGP runs between routers that are in different autonomous systems; EBGP neighbors are typically directly connected

Chapter 6 Summary

- Multihoming options:
  - Each ISP passes only a default route to the AS.
  - Each ISP passes only a default route and provider-owned specific routes to the AS.
  - Each ISP passes all routes to the AS.
- BGP's loop free guarantee, because it does not accept a routing update that already includes its AS number in the path list.
- When to use BGP and when not to use BGP.
- BGP’s classification as a path vector protocol and its use of TCP protocol 179.
- The use of full-mesh IBGP on all routers in the transit path within the AS.
- The BGP synchronization rule.
- The three tables used by BGP: the BGP table, IP routing table, and BGP neighbor table.
- The four BGP message types: open, keepalive, update, and notification..

Chapter 6 Summary

- BGP attributes: well-known or optional, mandatory or discretionary, and transitive or nontransitive.
- The BGP Well-known attributes including: AS-path, next-hop and origin.
- The BGP Well-known discretionary attributes including: local-preference, atomic aggregate.
- The BGP optional transitive attributes including: aggregator and community.
- The BGP optional nontransitive attributes including the MED.
- The Cisco specific weight attribute was also discussed.
- The 11-step BGP route selection decision process.
- BGP configuration commands.
- BGP verification commands.
- BGP path manipulation commands.

Chapter 6 Labs

- BGP-LAB-1.1
- BGP-LAB-1.2
- Full Scale IPv4

Q&A