UNIVERSITY OF ALBERTA

Project name: In depth analysis of BGP protocol, its security vulnerabilities and solutions

Supervisor: Professor Gurpreet Nanda
Designation : Senior Solution Architect at Fujitsu

Student name: Harsimran Singh

Table of Contents

Abstract

BGP is a de facto standard as an inter-domain routing protocol. The whole internet relies on BGP protocol to exchange routing and reachability information between different Autonomous system numbers. As such it is important to understand the concepts in detail of the protocol including the reasons as to why it is used, what are the parameters it has, how they can be tuned, how filters can be applied to filter or allow some data and what are the security issues associated with the protocol and what are the best practises to limit those issues, which in turn benefits the learners and also the organizations can use this to understand the concepts and issues related to this protocol. Many wrong configurations and flaws have already led to a lot of financial loss to different organizations which either led to some information into the black hole or led to stealing of the data as per attacker's will. Additionally this project discusses the various models developed to resolve the security issues with BGP protocol and also discusses its drawbacks as to why they were never implemented. Also ideas regarding some new solutions are being discussed which try to overcome the drawbacks of the previous solutions and is more in favour of the organizations and vendors to implement to better secure the protocol.

The report has three sections with the first one discussing the concepts and configuration details of BGP. The second part discusses the security issues with the protocol and different vulnerabilities in the protocol which can impact the performance of a network , which in turn can impact different organizations across the world and can bring internet to a standstill. The third part discusses different models and previous researches done to mitigate these security issues, the drawbacks associated with these issues and some new ideas to provide solution to the security problem in BGP.

Hardware and software used

- Cisco 3745 routers with C3745-ADVENTERPRISEK9_IVS-M), Version 12.4(25d), RELEASE SOFTWARE (fc1)
- GNS 3 simulator tool
- Wireshark for packet capturing
- Ubuntu machine
- Hping3 utility
- Python programming language
- Packet tracer Cisco tool
- SQL database

Section 1 : BGP concepts and Configurations

1.1 History

BGP was created in 1989 with the publishing of RFC 1105 defining the key fundamental concepts of protocol along with the various message types, formats and how routes are propagated with BGP. The key reason for the creation of BGP was the rapid increase in the size of internet which EGP(Exterior Gateway Protocol) was not able to handle as it was not designed to handle very large networks and several weaknesses. EGP had limited capabilities which were not enough for the internet to function. EGP was not capable of efficiently routing in such an environment because of its inability to detect loops, it's very slow convergence time, and its lack of tools to support routing policies. Lack of features and policies in EGP led to its modification but it ultimately gave birth to an entirely new protocol BGP which is the core of the Internet today. The whole world communicates its routes using BGP today. It is the core of internet backbone.

The initial version of BGP BGP-1 used the concept of directional topology with certain routers being up, down or horizontal relative to each other which led to the refinement of protocol to BGP version 2 which defined the concept of path attributes and made BGP better suited to arbitrary AS topology. Further BGP version 3 and BGP version 4 were introduced.

BGP version 4 is the latest version of BGP in use today to exchange ipv4 routes and supports CIDR routing. When internet grew fast and the IP addresses began to ran out the concept of CIDR routing was introduced which allowed to distribute classless networks to the organizations rather than classful networks which saved a lot of IP addresses and prevented the rapid vanishing of IPV4 address space. his also led to a steep increase in he size of routing tables as with the classless networks a single classful network was further subnetted into many networks with each network present in the routing table, thus increasing the increase in size of routing table. BGP was modified to allow prefixes to be specified that represent a set of aggregated networks. This led to a decrease in the size of routing tables as a similar set of IPV4 address space was allocated to a particular region instead of distributing it to different regions and hence these routes are summarized when they are injected into the other region,. Hence the other region will just see a single route for this whole region saving the routing table space.

1.2 Concept of Autonomous system(AS)

An autonomous system is a network or a collection of networks that are all managed and supervised by a single entity or organization. AS has many networks and subnets and has a common set of routing policies and routing logic. It was introduced to regulate organizations such as Internet service providers (ISP), educational institutions and government bodies. These systems are made up of many different networks but are operated by a single entity for easy management and are grouped under a single AS. Border Gateway Protocol (BGP) is the protocol that addresses the routing of packets among different autonomous systems to connect them. BGP uses the ASN to uniquely identify each system. This is particularly important when routing and managing routing tables for external networks or autonomous systems around their borders. Within an AS any IGP can be used for route sharing and policies, while BGP is used to connect to two different AS together. The important

characteristic of an autonomous system from the BGP point of view is that the autonomous system appears to other autonomous systems to have a single coherent interior routing plan, and it presents a consistent picture of which destinations can be reached through it. **IANA** is the main organization responsible for allocating Autonomous system numbers. **Regional Internet registries (RIRs)** are non profit corporations established for the purpose of administration and registration of IP address space and autonomous system numbers. There are five RIRs, as follows:
1 African Network Information Centre (AfriNIC) is responsible for the continent of Africa.
2 Asia Pacific Network Information Centre (APNIC) administers the numbers for the Asia Pacific region.
3 American Registry for Internet Numbers (ARIN) has jurisdiction over assigning numbers for Canada, the United States, and several islands in the Caribbean Sea and North Atlantic Ocean.
4 Latin American and Caribbean IP Address Regional Registry (LACNIC) is responsible for allocation in Latin America and portions of the Caribbean.
5 Reséaux IP Européens Network Coordination Centre (RIPE NCC) administers the numbers for Europe, the Middle East, and Central Asia.

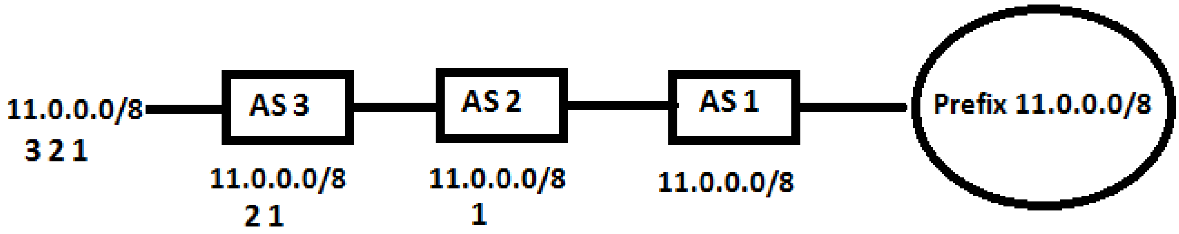
There are two types of AS Numbers:

1)**Public AS Numbers :** is required only when an AS is exchanging routing information with other Autonomous Systems on the public Internet. That is, all routes originating from an AS is visible on the Internet. For 16 bit AS numbers range is 1 to 64495 and we have extended 32 bit AS numbers also.

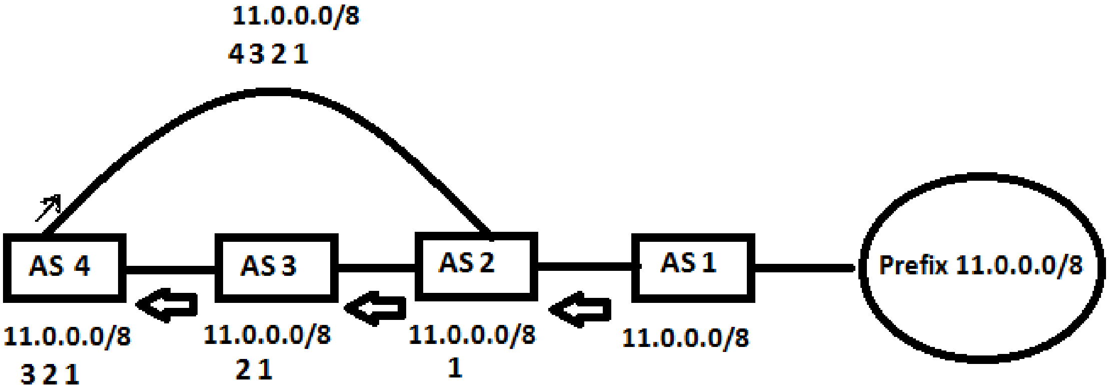
2)**Private AS Numbers** : A Private AS Number should be used if an AS is only required to communicate via Border Gateway Protocol with a single provider. As the routing policy between the AS and the provider will not be visible in the Internet, a Private AS Number can be used for this purpose. IANA has reserved, for Private Use, a contiguous block of 1023 Autonomous System numbers from the 16-bit Autonomous System Numbers registry, 64512 – 65534. IANA has also reserved, for Private Use, a contiguous block of 94,967,295 Autonomous System numbers from the 32-bit Autonomous System Numbers registry, 4200000000 – 4294967294.

1.3 Introduction to BGP and BGP basics

BGP is an inter domain routing protocol used to exchange routing information between two different autonomous. It advertises, learns and using its different Path attributes chooses the best paths inside the global internet. These path attributes provides a wide range of options to choose the best paths instead of just the metrics as in IGP which are based on a few factors such as cost of link, delay, reliability, load etc. BGP advertises Network layer reachablity Information(NLRI) which is a combination of a prefix, prefix length and various path attributes associated with it. Before exchanging any routing information BGP establishes a neighbour relationship with the router with which routing information is to be exchanged. Unlike IGP'S BGP neighbour's don't have to be on same subnet. They can be either directly connected or can be separated by many subnets and can be anywhere in the world with the only condition that the ip addresses used to establish neighbour relationship shall be reachable. To increase the reliability of the peer connection, BGP uses TCP port 179 as its underlying delivery mechanism. The update mechanisms of BGP are handled by the TCP layer and various duties as acknowledgment, retransmission, and sequencing are performed by TCP. Because BGP rides on TCP, a separate point-to-point connection to each peer must be established. BGP is a path vector protocol and contains a list of all the Autonomous system a packet will traverse before reaching its destination.

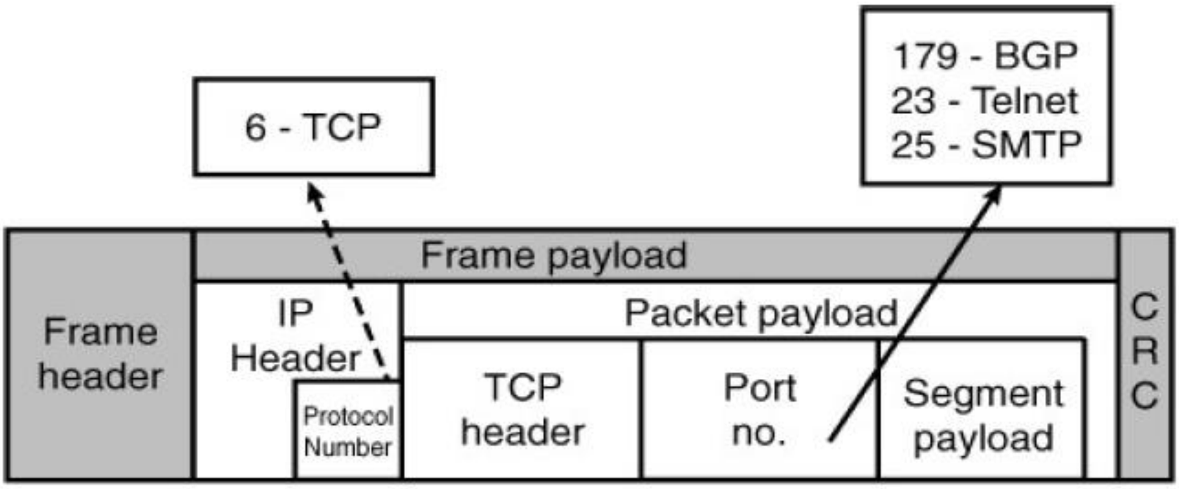


As can be seen in the figure prefix 11.0.0.0/8 is exchanged by AS 1 to AS 2. As the prefix is exchanged and the AS1 router send an EBGP update to AS2 it adds its AS in the AS path attribute and as can be seen as the update reaches AS 2 it lists the Autonomous system number 1 in AS path attribute. Now when the AS 2 sends an outbound update for this prefix to AS 3 it adds its own AS number in AS path attribute along with the previous AS number. Hence now the AS 3 BGP table has 2 autonomous systems in AS path length attribute with the previous autonomous system on the left side and originating autonomous system on the right most side. The AS path length attribute is also used to avoid routing loops.



As can be seen in the diagram if there is a loop in the system such that AS 4 is connected to AS 2, then AS path is used to avoid that loop. The concept behind that is simple, if an AS sees itself listed in the update received from the neighbouring AS it knows that the prefix has already traversed that Autonomous system and hence ignores that update as reintroducing that prefix in its own AS can cause a routing loop to occur.

By default if no other path attribute is configured this path attribute acts as a default tie breaker in selecting the best path to be installed in BGP table.



BGP using TCP for communication (Image reference CCNP ROUTE 642-902)

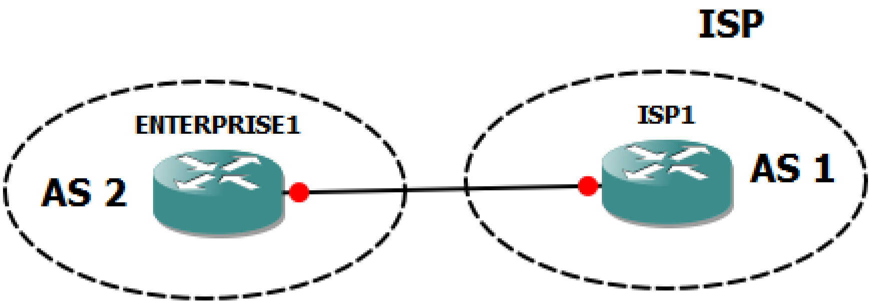
1.4 Where and Where not to use BGP when using it for outbound Routing

Depending on the number of ISP's and number of links to each ISP an enterprise can consider using either default routes or BGP for the best path choice.

It makes more sense to consider BGP for outbound routing if either at least two ISP's exist or two links to a single ISP exist. Then BGP can be used to choose one path over the other for particular destinations on the internet. There are different designs which are used to explain in which situations to use BGP for outbound routing.

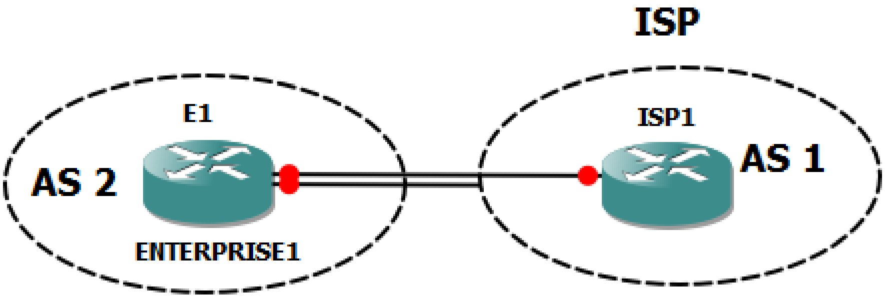
1.4.1 Single homed design

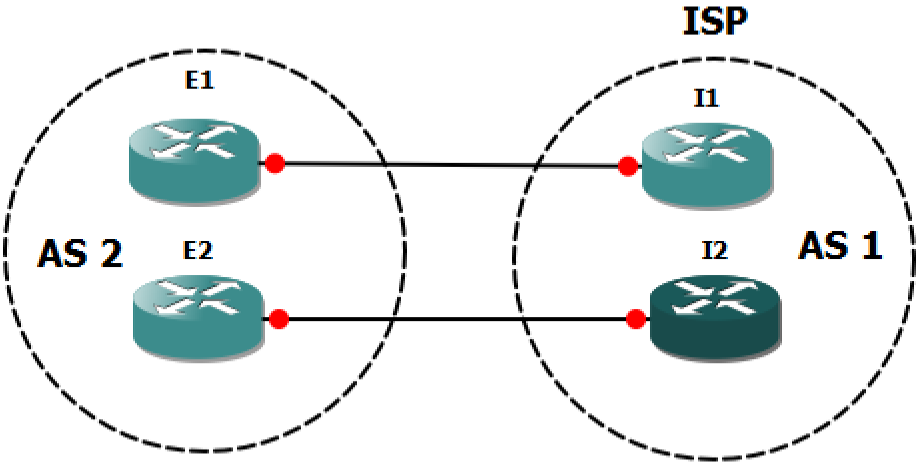
A single homed design involves a single link and a single ISP. Since only a single ISP is involved and there is only one exit door there is no requirement to use BGP in this case. Instead default route can be used from enterprise to ISP and a static route can be used at ISP to point towards Enterprise. Or alternatively a default route can be used to share via BGP and no other routes shall be shared.



1.4.2 Dual homed design

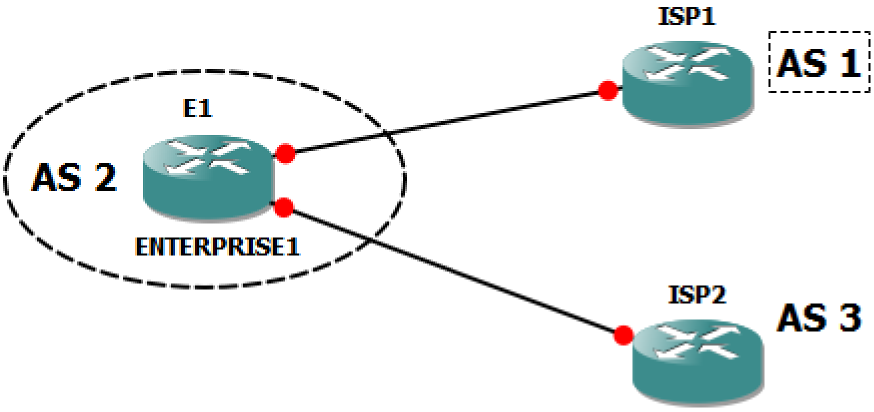
It has two or more links to internet but it involves only a single ISP. It can be configured in such a way that one path can always be used and other can be kept standby or both paths can be load shared or if there are two enterprise routers then one path can be preferred over other or the traffic can be load shared between those two links or the priority traffic can be sent through one link while the normal traffic can be sent through other.

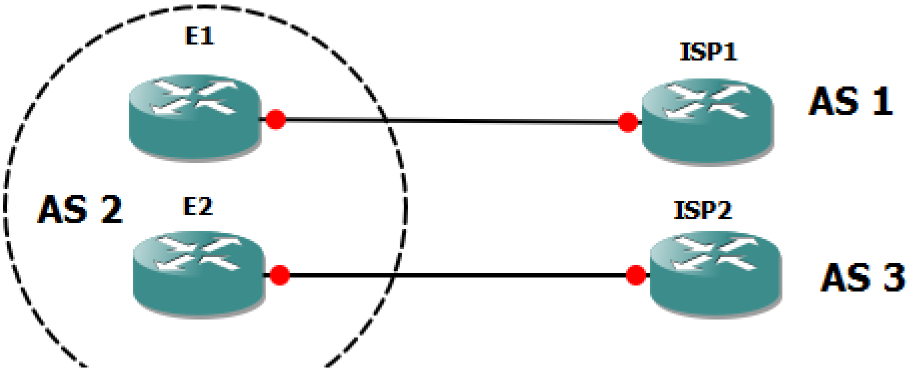




1.4.3 Single multi homed design

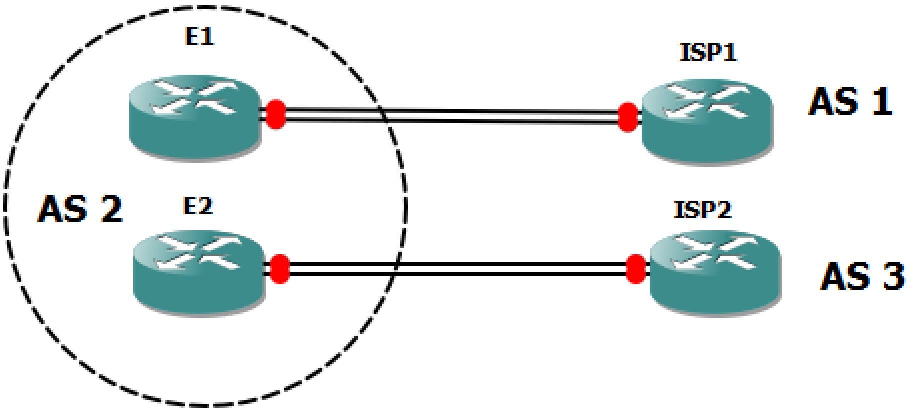
It has a single connection to two or more multiple ISP's. Hence there is a choice to influence the outbound routes using BGP. Either full BGP updates can be obtained from both ISPs or partial BGP updates can be obtained from one ISP and full from other. Also BGP can be modified to prefer one link over other using its path attributes. The enterprise router E1 can be a single router or two different routers to two ISP'S.

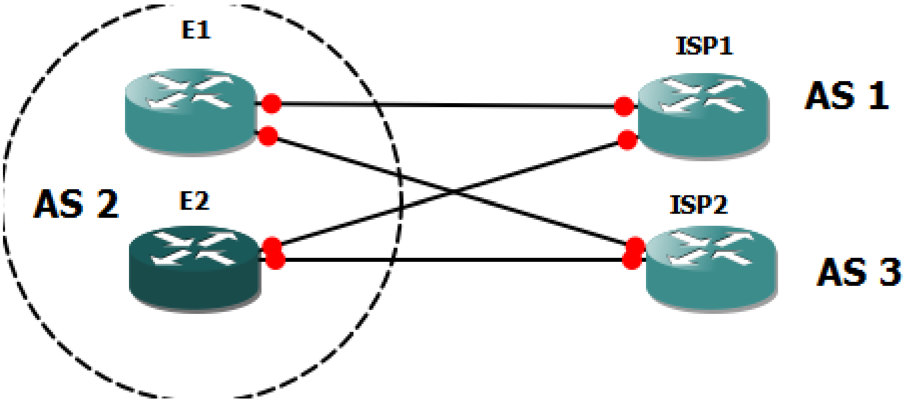




1.4.4 Dual multi homed design

It involves dual connections with dual ISP'S or a single connection to each ISP from two different enterprise routers E1 ,E2 as shown in figure. As such the redundancy increases and also BGP can be used to use efficiently these links and load balance the traffic as required.



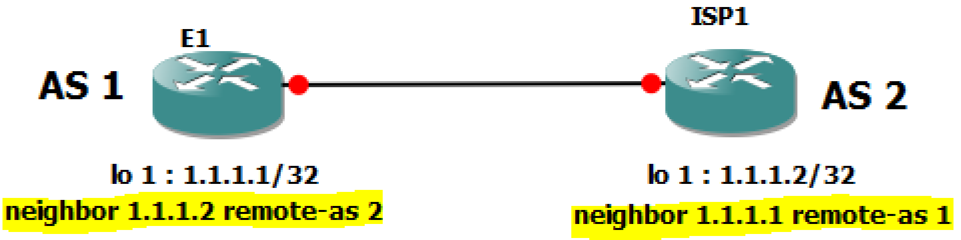


1.5 Bgp neighbor relationships

Any router running BGP is called a BGP speaker. A BGP router forms a direct neighbor relationship over port TCP 179 with the other BGP router(peer) whose address is mentioned in the neighbor command configured on that router. Each neighbor is configured explicitly in the command. If a router receives a BGP neighbor request from an IP address which is not mentioned in the neighbor commad on this router the router ignores the neighbor request. After a TCP connection is established BGP starts exchanging open messages to negotiate different parameters and finally settles down in established state after which updates are being exchanged.

Various parameters are checked before two routers become neighbors. These parameters are listed below:

1) The router must receive a connection request from the source ip address which is configured in the neighbor command of its bgp protocol. As can be seen in the figure down ISP1 router shall receive a bgp request from ip address 1.1.1.1 which is configured in its neighbor statement. If it receives a request from any other IP address the request is rejected.



2) The autonomous system number listed in the neighbor command shall be same as configured on the neighboring router. Refering to above figure ISP1 router states remote-as of 1 in neighbor command. Hence E1 router should have been configured for AS1 else neighborship will fail.

3) Router IDS must be unique for both routers

4) Neighbor authentication check must pass if configured.

1.5.1 BGP messages and neighbor states

BGP uses 4 message types:

1) **Open Message**

After the TCP session is established, both neighbors send Open messages. Each neighbor uses this message to identify itself and to specify its BGP operational parameters. The Open message includes the following information:

A) BGP version number

This specifies the version (2, 3, or 4) of BGP that the originator is running. Unless a router is set to run an earlier version with the **neighbor version** command, it defaults to BGP-4. If a neighbor is running an earlier version of BGP, it rejects the Open message specifying version 4; the BGP-4 router then changes to BGP-3 and sends another Open message specifying this version. This negotiation continues until both neighbors agree on the same version.

B) Autonomous system number

This is the AS number of the originating router. It determines whether the BGP session is EBGP (if the AS numbers of the neighbors differ) or IBGP (if the AS numbers are the same).

C) Hold time

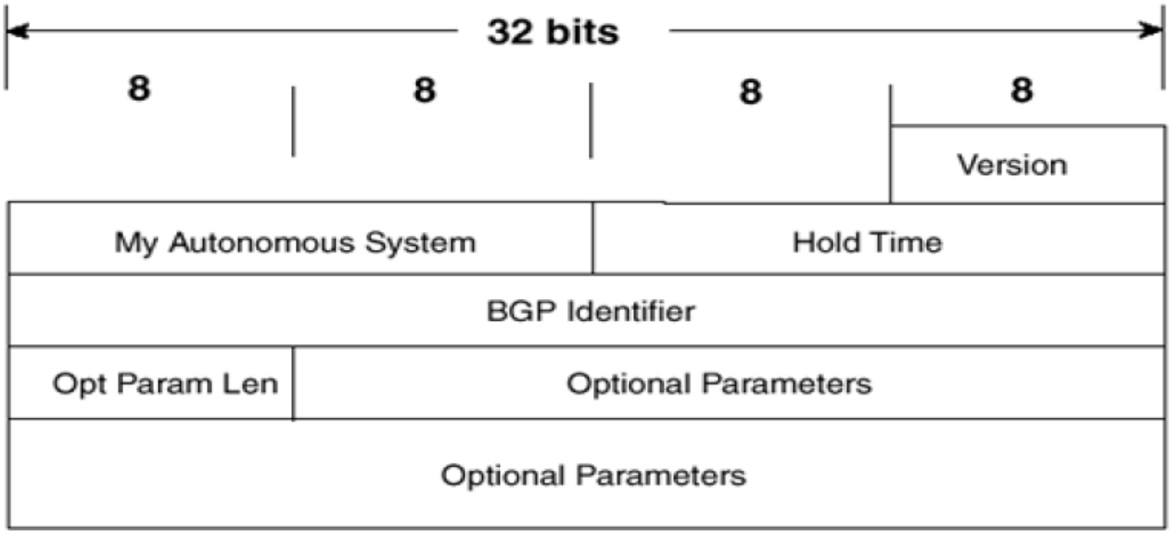
This is the maximum number of seconds that can elapse before the router must receive either a Keepalive or an Update message. The hold time must be either 0 seconds (in which case, Keepalives must not be sent) or at least 3 seconds; the default Cisco hold time is 180 seconds. If the neighbors' hold times differ, the smaller of the two times becomes the accepted hold time.

D) BGP identifier

This is an IP address that identifies the neighbor. The Cisco IOS determines the BGP Identifier in exactly the same way as it determines the OSPF router ID: The numerically highest loopback address is used; if no loopback interface is configured with an IP address, the numerically highest IP address on a physical interface is selected.

E)Optional parameters

This field is used to advertise support for such optional capabilities as authentication, multiprotocol support, and route refresh.



2) Keepalive Message

If a router accepts the parameters specified in its neighbor's Open message, it responds with a keepalive. Subsequent keepalives are sent every 60 seconds by Cisco default, or a period equal to one-third the agreed-upon hold time. If the keepalives are not received during the received holddown timer value the neighbor is considered dead.

3) Update Message

The Update message advertises feasible routes, withdrawn routes, or both. The Update message includes the following information:

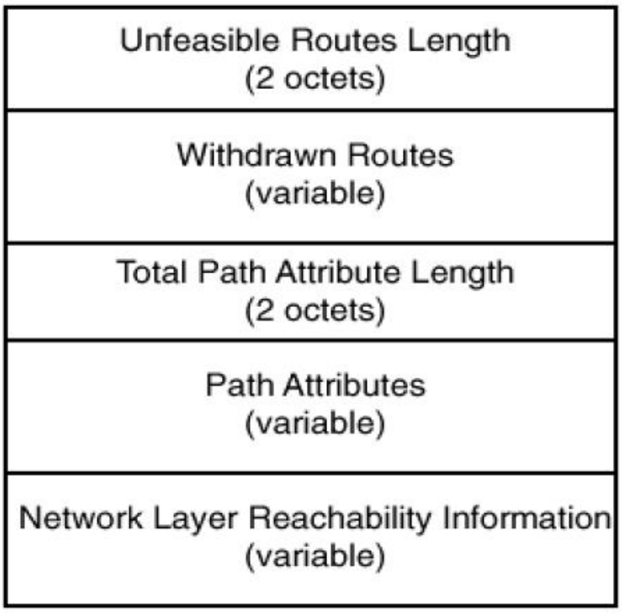
1) **Network Layer Reachability Information (NLRI)—** This is one or more (Length, Prefix)

tuples that advertise IP address prefixes and their lengths. If 206.193.160.0/19 were being advertised, for example, the Length portion would specify the /19 and the Prefix portion would specify 206.193.160.

2) **Path Attributes—** The path attributes, described in a later section of the same name, are

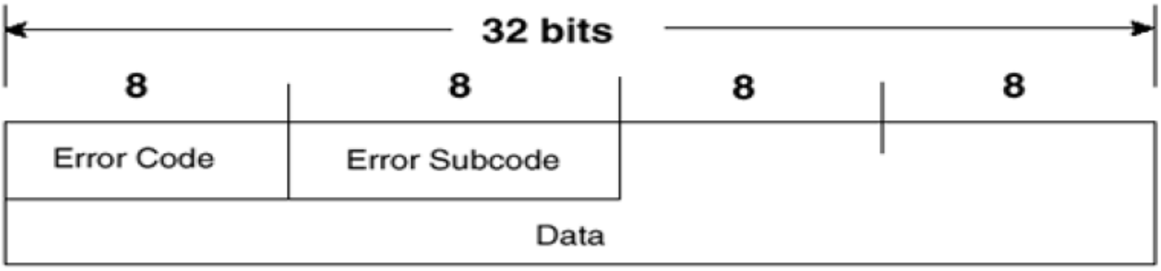
characteristics of the advertised NLRI. The attributes provide the information that allows BGP to choose a shortest path, detect routing loops, and determine routing policy.

3) **Withdrawn Routes**— These are (Length, Prefix) tuples describing destinations that have become unreachable and are being withdrawn from service.



4) **Notification message :**

The Notification message is sent whenever an error is detected and always causes the BGP connection to close. Example if BGP versions mismatch between two neighbors a notification message is issued.



Error code provides information about the error and error subcode provides more specific information.

BGP neighbor states

1) Idle State :

BGP always begins in the Idle state, in which it refuses all incoming connections. The BGP process initializes all BGP resources, starts the Connect Retry timer, initializes a TCP connection to the neighbor, listens for a TCP initialization from the neighbor, and changes its state to Connect.

2) Connect State :

In this state, the BGP process is waiting for the TCP connection to be completed. If the TCP connection is successful, the BGP process clears the Connect Retry timer, completes initialization, sends an Open message to the neighbor, and transitions to the Open Sent state. If the TCP connection is unsuccessful, the BGP process continues to listen for a connection to be initiated by the neighbor, resets the Connect Retry timer, and transitions to the Active state.

3) Active State:
In this state, the BGP process is trying to initiate a TCP connection with the neighbor. If the TCP connection is successful, the BGP process clears the Connect Retry timer, completes initialization, sends an Open message to the neighbor, and transitions to Open sent. The Hold timer is set to 4 minutes.
If the Connect Retry timer expires while BGP is in the Active state, the process transitions back to the Connect state and resets the Connect Retry timer.

4) OpenSent State:
In this state, an Open message has been sent, and BGP is waiting to hear an Open from its neighbor. When an Open message is received, all its fields are checked. If errors exist, a Notification message is sent and the state transitions to Idle.
If no errors exist in the received Open message, a keepalive message is sent and the keepalive timer is set. The Hold time is negotiated, and the smaller value is agreed upon. The peer connection is determined to be either internal or external, based on the peer's AS number, and the state is changed to Open confirm.

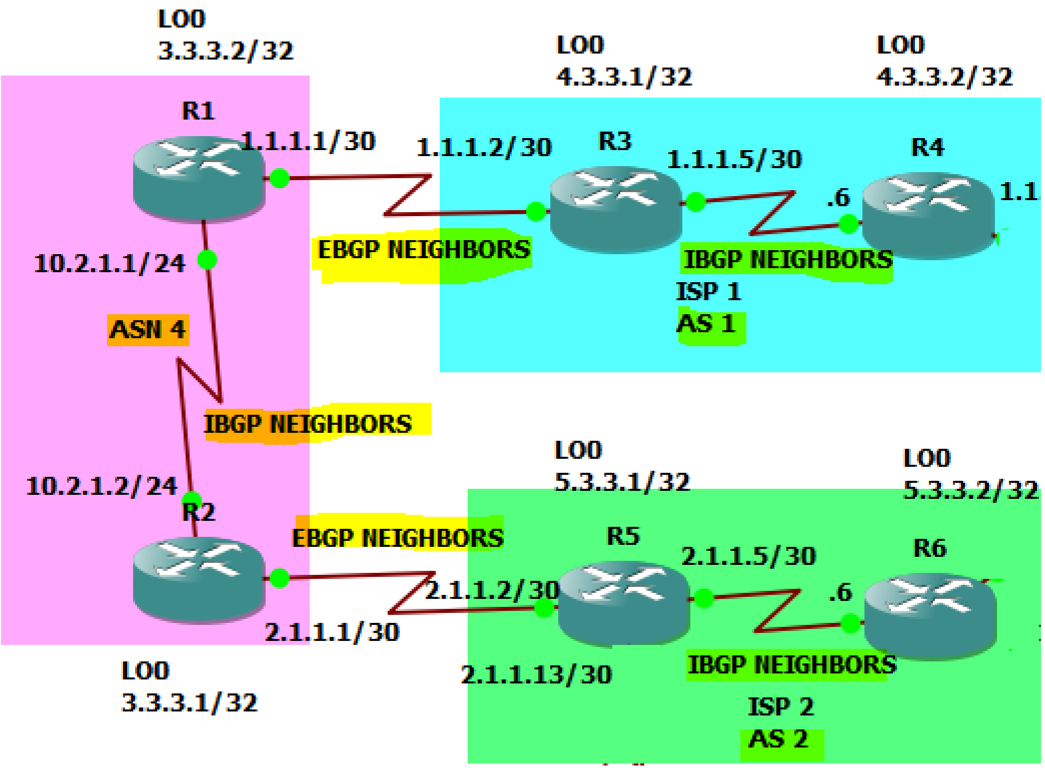
5) OpenConfirm State:
In this state, the BGP process waits for a keepalive or Notification message. If a keepalive is received, the state transitions to Established. If a Notification is received, or a TCP disconnect is received, the state transitions to Idle.
If the Hold timer expires, an error is detected, or a Stop event occurs, a Notification is sent to the neighbor and the BGP connection is closed, changing the state to Idle.

6)Established State
In this state, the BGP peer connection is fully established and the peers can exchange Update, keepalive, and Notification messages. If an Update or keepalive message is received, the Hold timer is restarted (if the negotiated hold time is nonzero). If a Notification message is received, the state transitions to Idle. Any other event (again, except for the Start event, which is ignored) causes a Notification to be sent and the state to transition to Idle.

1.5.2 BGP neighbor Types

1) Internal BGP neighbors
2)External BGP neighbors

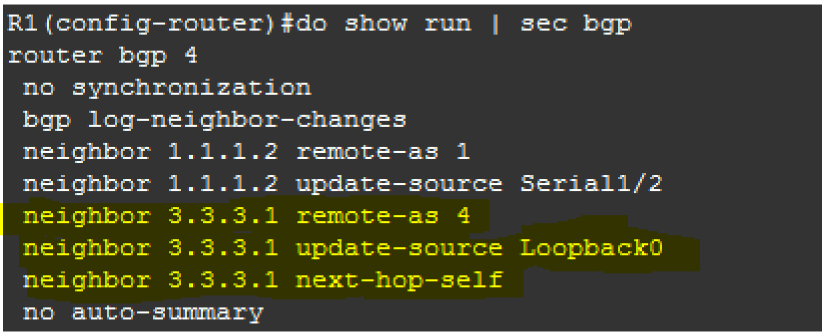
1) Internal BGP neighbors :When BGP neigbors are in the same Autonomous system, the neighbor relationship between them is IBGP. IBGP is run within an autonomous system to exchange BGP information so that all internal BGP speakers have the same routing information about outside autonomous systems.
Requirements for IBGP neigbors :
1)Same autonomous system
2)Define neighbors in neighbor command
3) IBGP neighbor ip address must be reachable. Usually IGP protocol is used to share the routing information within an AS so that the ip addresses specified in the neighbor command is reachable. As an example in the below network R1 and R2 are IBGP neighbors.



Network diagram for this Part

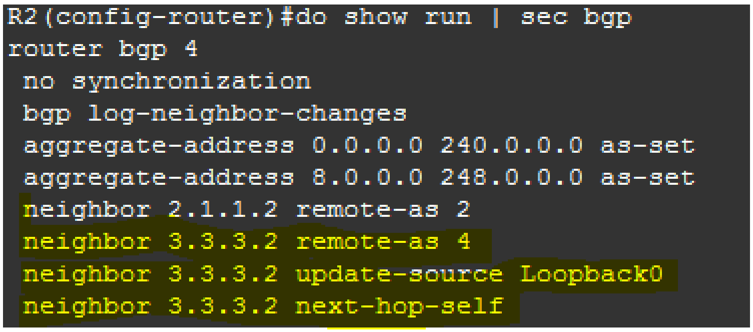
Configurations on R1

BGP configuration commands
Useful commands are highlighted below. As can be seen



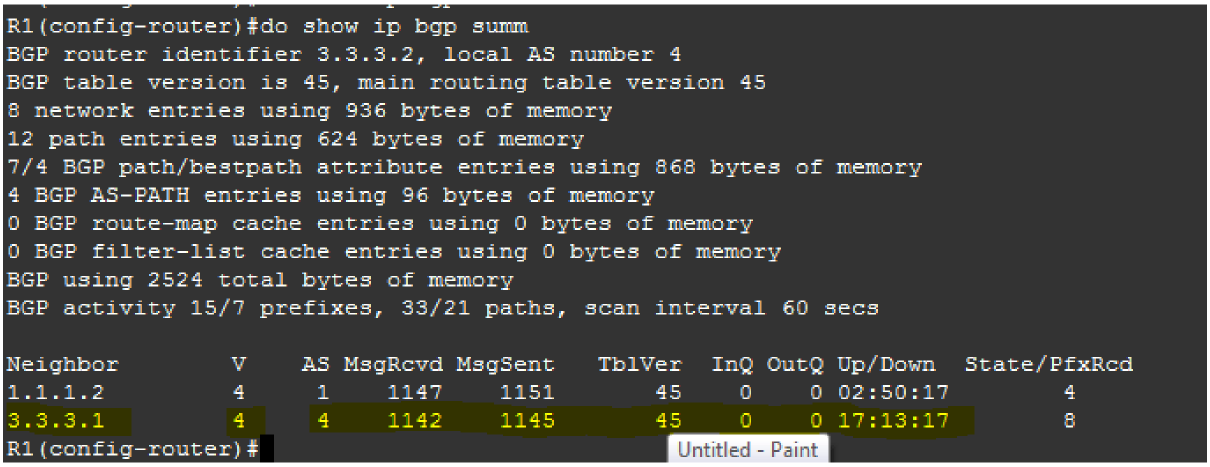
```
R1(config-router)#do show run | sec bgp
router bgp 4
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.2 remote-as 1
 neighbor 1.1.1.2 update-source Serial1/2
 neighbor 3.3.3.1 remote-as 4
 neighbor 3.3.3.1 update-source Loopback0
 neighbor 3.3.3.1 next-hop-self
 no auto-summary
```

Configuration command on R2



```
R2(config-router)#do show run | sec bgp
router bgp 4
 no synchronization
 bgp log-neighbor-changes
 aggregate-address 0.0.0.0 240.0.0.0 as-set
 aggregate-address 8.0.0.0 248.0.0.0 as-set
 neighbor 2.1.1.2 remote-as 2
 neighbor 3.3.3.2 remote-as 4
 neighbor 3.3.3.2 update-source Loopback0
 neighbor 3.3.3.2 next-hop-self
```

Show ip BGP neighbors on R1



```
R1(config-router)#do show ip bgp summ
BGP router identifier 3.3.3.2, local AS number 4
BGP table version is 45, main routing table version 45
8 network entries using 936 bytes of memory
12 path entries using 624 bytes of memory
7/4 BGP path/bestpath attribute entries using 868 bytes of memory
4 BGP AS-PATH entries using 96 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 2524 total bytes of memory
BGP activity 15/7 prefixes, 33/21 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.1.1.2         4     1    1147    1151       45    0    0 02:50:17        4
3.3.3.1         4     4    1142    1145       45    0    0 17:13:17        8
R1(config-router)#
```

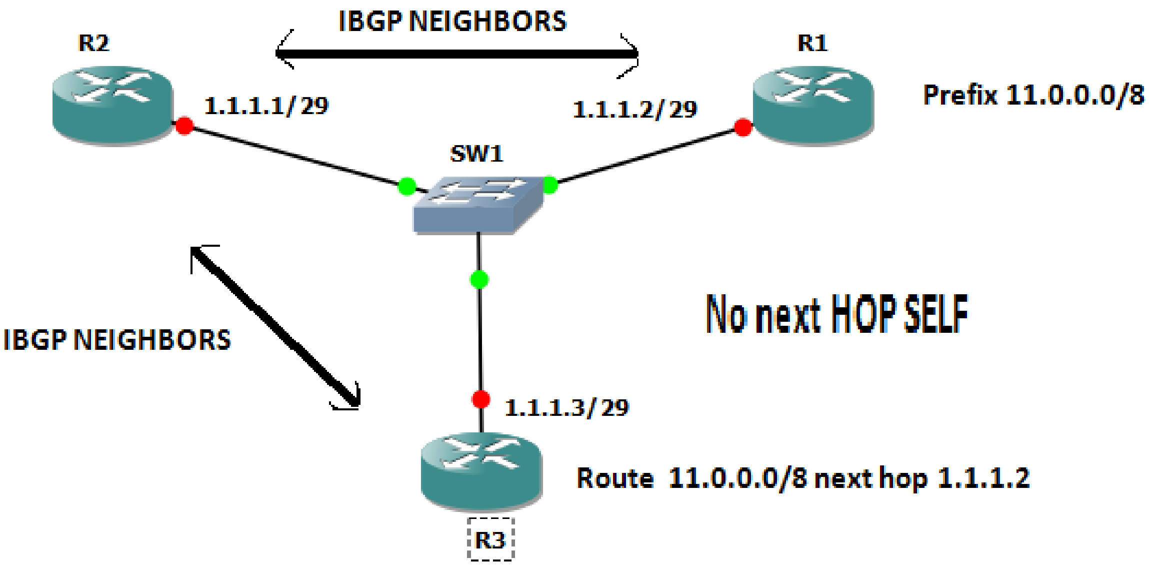
As can be seen in above outputs basically two commands are required for BGP neighbors to come up(Cisco router c3745).

1) Neighbor [ip address] remote-as[as no.] : This command states the ip address and AS number of the remote neighbor. If the ip address of neighbor is not reachable or is incorrect then the neighborship won't come up and a notification message will be issued telling that AS number at the remote end is different than configured on this router.

2 **Neighbor update source** : This command is required when the neighborship is formed on interface using loopback address. By this command the updates sent to the neighbor have a source address as mentioned in this command and apart from this the next hop neighbor listed in the neighboring router is this ip address. This command is used to take advantage of redundant links or paths that are used to reach between neighbors. As an example an organization running IBGP would have multiple routes for the same destination. Suppose if an interface ip address is mentioned in the neighbor command then if that interface goes down then the BGP neighbor will turn down. But if the neighborship is established on loopback interface then that interface or subnet will still be reachable using other alterfnate routers as two BGP neigbhbors don't need to be directly connected for neighborship. In addition if redundant links are present between two router then if two links are there between neighbors two neighbor statements have to be configured on each router which doubles the updates sent to the same neighbor. So, if loopback interface is used this can be avoided as using the source address of loopback interface only one update packet set would be generated which will improve the throughput and efficiency of the system.

1.5.2.1 Neighbor Next-hop self concept

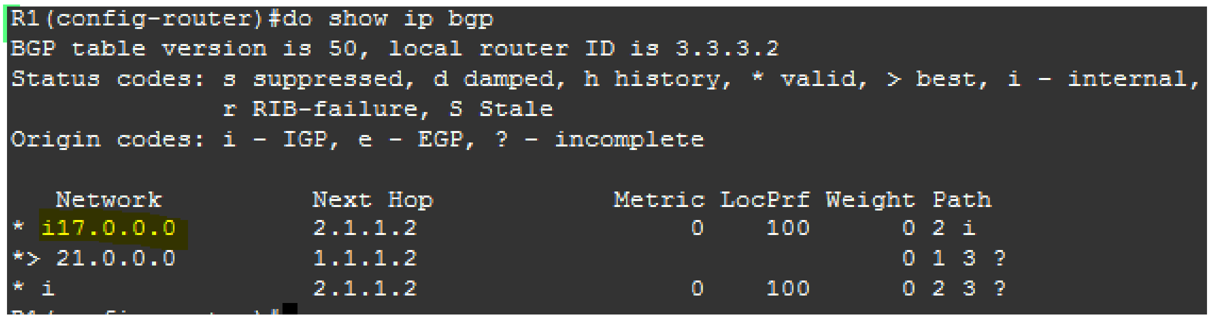
The default behaviour of IBGP is that it sends the update to its neighbor without changing the next hop. This process is advantageous in a broadcast environment as shown in the below diagram. R1 and R3 are IBGP neighbors and R2 and R1 are IBGP neighbors and all are connected in a broadcast environment. As can be seen when R2 send the update to R3 for the prefix 11.0.0.0/8 to R3 it doesn't send the update by listing itself as the next hop.



Behaviour with Next hop self command not configured

Instead it keeps the next hop same as it received from R1. Now this is of great advantage in this context as the packet can directly reach R1 instead of going through R2 thus improving the delay and avoiding packet processing at R2.
But it has its own disadvantages. As shown in the network diagram for this section suppose R2 forwards an EBGP update received from R5 to R1. In this case if the next hop remains unchanged then R1 shall have a route in its BGP table with the next hop IP of either R5 interface towards R2 or loopback interface of R5(Depends on what ip address neighbor relationship is established). This way R1 has to have learn the route towards R5 either using IGP or by using static route. Hence R1 will perform a recursive lookup to reach the next hop ip address in the routing table which increased the processing time and delay. Neighbor next hop self command solves this problem by listing R2 as the next hop.

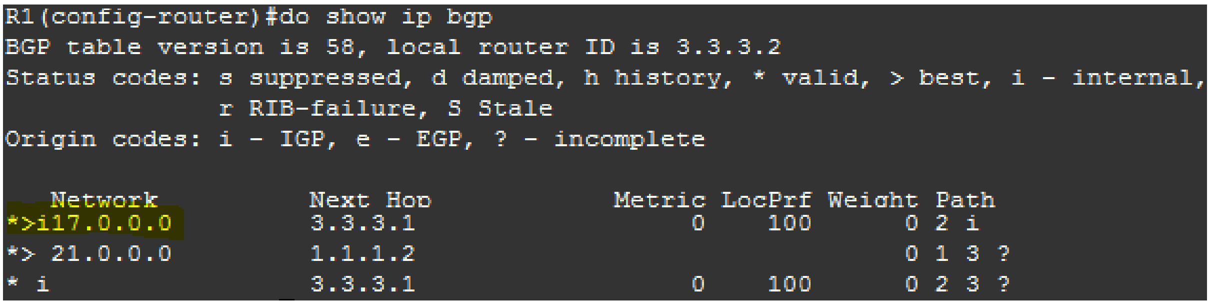
Show ip bgp command impact on next hop with the R2 configured and not configured for the next hop self command.



```
R1(config-router)#do show ip bgp
BGP table version is 50, local router ID is 3.3.3.2
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
* i17.0.0.0         2.1.1.2                  0    100      0 2 i
*> 21.0.0.0         1.1.1.2                                0 1 3 ?
* i                 2.1.1.2                  0    100      0 2 3 ?
```

With no next hop self command on R2

As can be seen the network 17.0.0.0/8 which is learned from router R5 EBGP has next hop ip of R5 listed. Since there is no route in R1 for this ip address the route in BGP table is unusable.

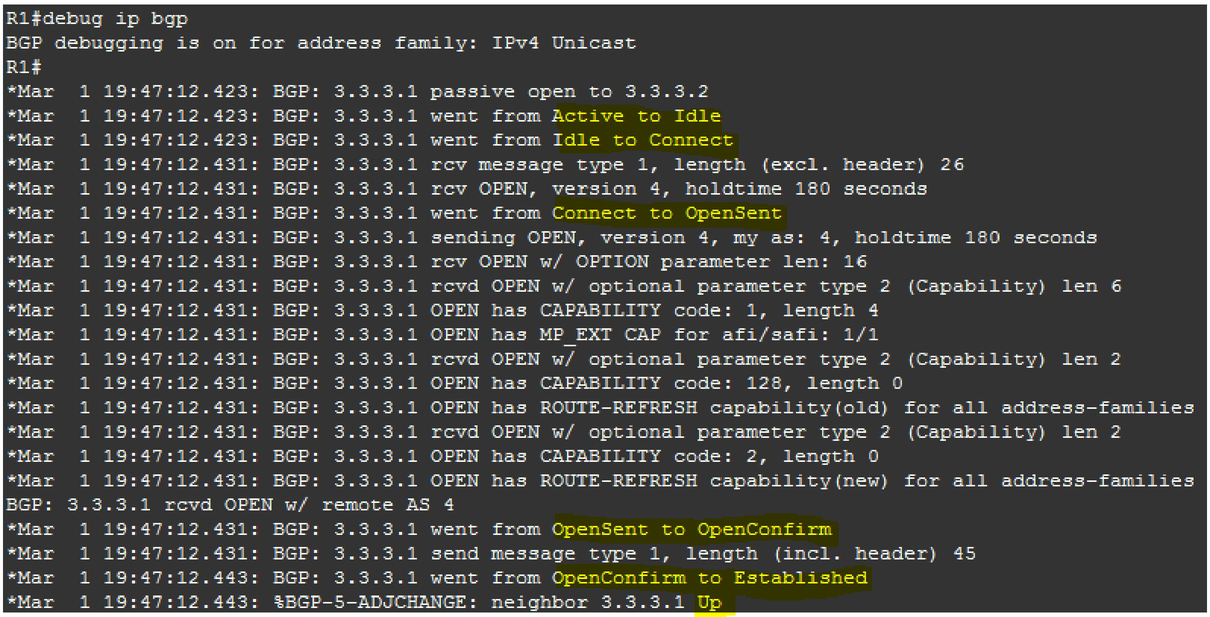


```
R1(config-router)#do show ip bgp
BGP table version is 58, local router ID is 3.3.3.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i17.0.0.0         3.3.3.1                  0    100      0 2 i
*> 21.0.0.0         1.1.1.2                                0 1 3 ?
* i                 3.3.3.1                  0    100      0 2 3 ?
```

With next hop self command on R2

As can be seen now the network 17.0.0.0/8 lists a next hop ip address of R2 which is reachable via the IGP protocol used in this AS. Hence the route is valid and usable.

Messages exchanged during neigborship process. Important states are highlighted.



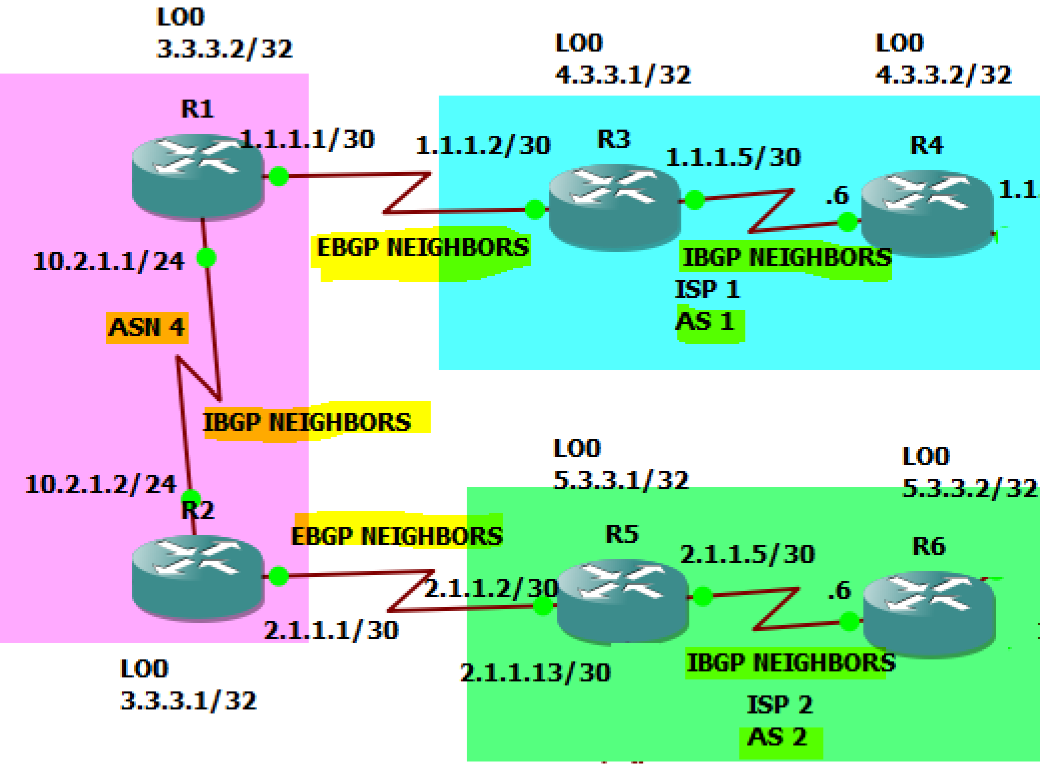
```
R1#debug ip bgp
BGP debugging is on for address family: IPv4 Unicast
R1#
*Mar  1 19:47:12.423: BGP: 3.3.3.1 passive open to 3.3.3.2
*Mar  1 19:47:12.423: BGP: 3.3.3.1 went from Active to Idle
*Mar  1 19:47:12.423: BGP: 3.3.3.1 went from Idle to Connect
*Mar  1 19:47:12.431: BGP: 3.3.3.1 rcv message type 1, length (excl. header) 26
*Mar  1 19:47:12.431: BGP: 3.3.3.1 rcv OPEN, version 4, holdtime 180 seconds
*Mar  1 19:47:12.431: BGP: 3.3.3.1 went from Connect to OpenSent
*Mar  1 19:47:12.431: BGP: 3.3.3.1 sending OPEN, version 4, my as: 4, holdtime 180 seconds
*Mar  1 19:47:12.431: BGP: 3.3.3.1 rcv OPEN w/ OPTION parameter len: 16
*Mar  1 19:47:12.431: BGP: 3.3.3.1 rcvd OPEN w/ optional parameter type 2 (Capability) len 6
*Mar  1 19:47:12.431: BGP: 3.3.3.1 OPEN has CAPABILITY code: 1, length 4
*Mar  1 19:47:12.431: BGP: 3.3.3.1 OPEN has MP_EXT CAP for afi/safi: 1/1
*Mar  1 19:47:12.431: BGP: 3.3.3.1 rcvd OPEN w/ optional parameter type 2 (Capability) len 2
*Mar  1 19:47:12.431: BGP: 3.3.3.1 OPEN has CAPABILITY code: 128, length 0
*Mar  1 19:47:12.431: BGP: 3.3.3.1 OPEN has ROUTE-REFRESH capability(old) for all address-families
*Mar  1 19:47:12.431: BGP: 3.3.3.1 rcvd OPEN w/ optional parameter type 2 (Capability) len 2
*Mar  1 19:47:12.431: BGP: 3.3.3.1 OPEN has CAPABILITY code: 2, length 0
*Mar  1 19:47:12.431: BGP: 3.3.3.1 OPEN has ROUTE-REFRESH capability(new) for all address-families
BGP: 3.3.3.1 rcvd OPEN w/ remote AS 4
*Mar  1 19:47:12.431: BGP: 3.3.3.1 went from OpenSent to OpenConfirm
*Mar  1 19:47:12.431: BGP: 3.3.3.1 send message type 1, length (incl. header) 45
*Mar  1 19:47:12.443: BGP: 3.3.3.1 went from OpenConfirm to Established
*Mar  1 19:47:12.443: %BGP-5-ADJCHANGE: neighbor 3.3.3.1 Up
```

Note: IBGP neigbor relationships are usually formed on loopback interfaces to take the full advantage of alternate routes and links.

<u>External BGP neighbors</u>

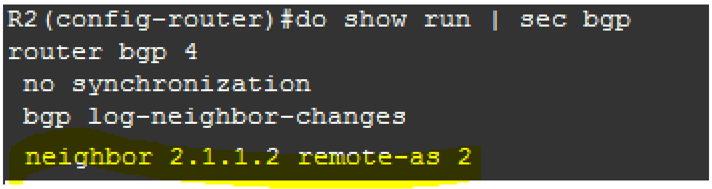
When the neighboring BGP routers are in different Autonomous systems the resulting neighborship is an EBGP connection. Usually EBGP connection is between different organizations and as such only one link connects the two routers. So if there is just one link connecting the two routers it's good to establish neighborship on the interface directly rather than loopback ip address, so that if interface fails BGP connection fails.

In the below network diagram R2 and R5 are EBGP neighbors and R1 and R3 are EBGP neighbors.

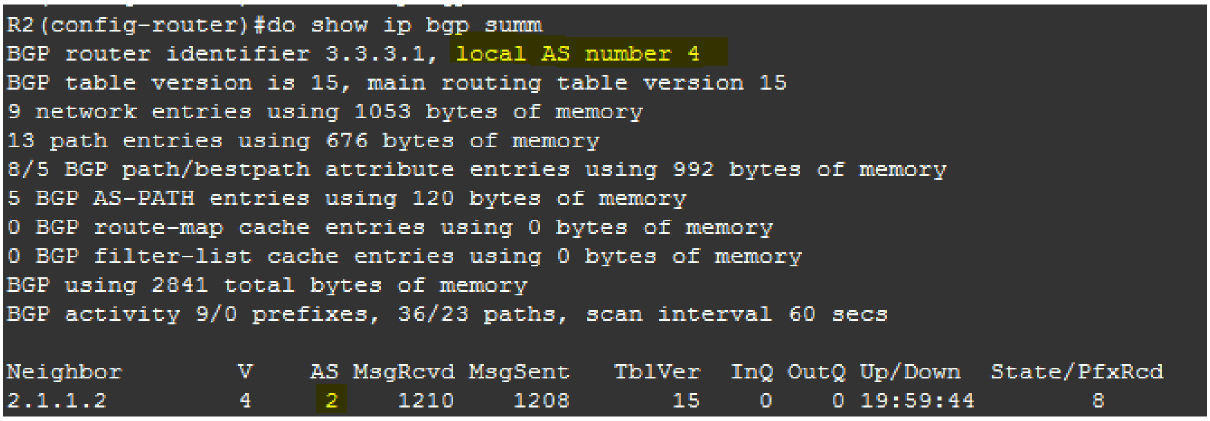


Network Diagram For this section

Configuration command on R2 and R5

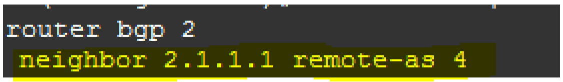


```
R2(config-router)#do show run | sec bgp
router bgp 4
 no synchronization
 bgp log-neighbor-changes
 neighbor 2.1.1.2 remote-as 2
```

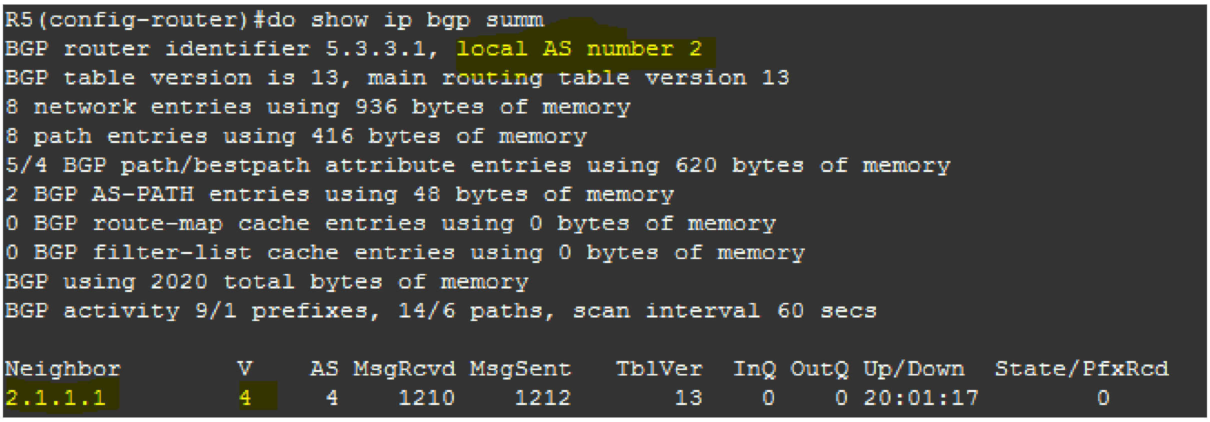


```
R2(config-router)#do show ip bgp summ
BGP router identifier 3.3.3.1, local AS number 4
BGP table version is 15, main routing table version 15
9 network entries using 1053 bytes of memory
13 path entries using 676 bytes of memory
8/5 BGP path/bestpath attribute entries using 992 bytes of memory
5 BGP AS-PATH entries using 120 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 2841 total bytes of memory
BGP activity 9/0 prefixes, 36/23 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.1.1.2         4     2    1210    1208       15    0    0 19:59:44        8
```

R5



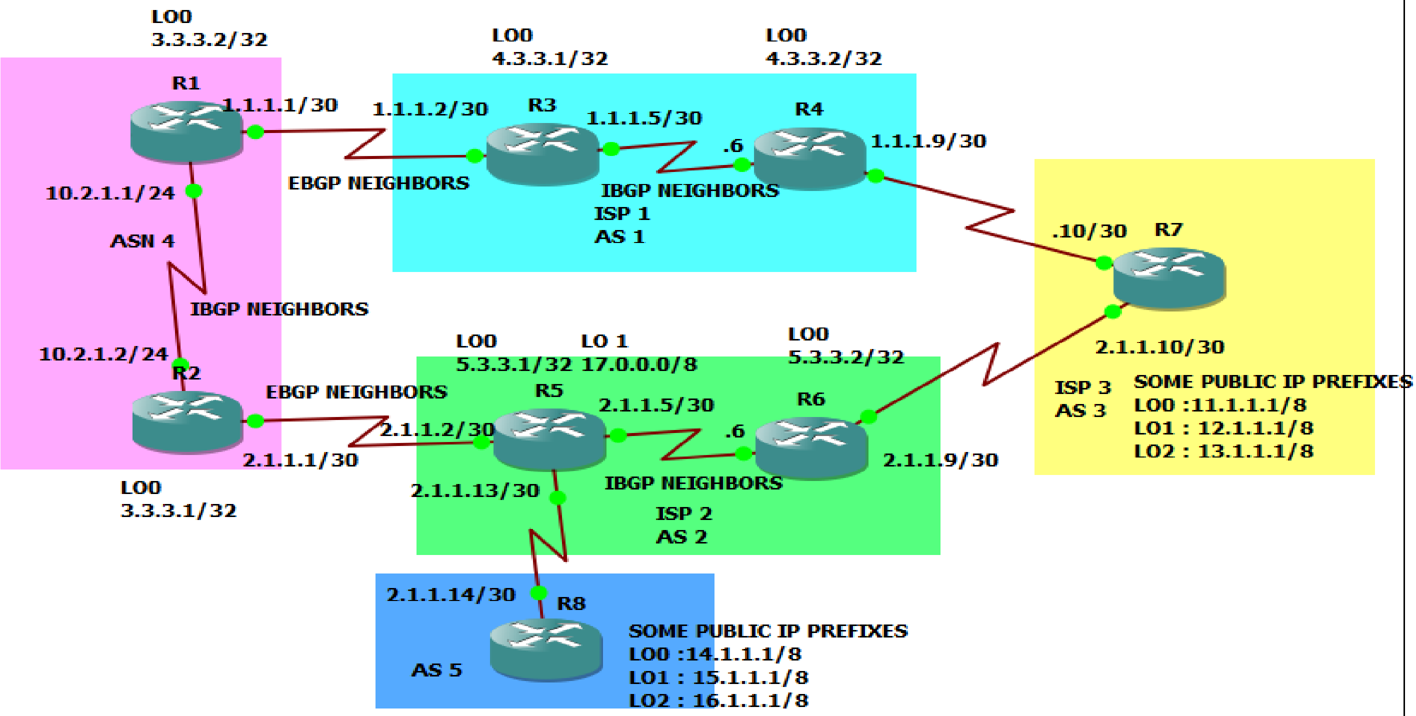
```
router bgp 2
 neighbor 2.1.1.1 remote-as 4
```



```
R5(config-router)#do show ip bgp summ
BGP router identifier 5.3.3.1, local AS number 2
BGP table version is 13, main routing table version 13
8 network entries using 936 bytes of memory
8 path entries using 416 bytes of memory
5/4 BGP path/bestpath attribute entries using 620 bytes of memory
2 BGP AS-PATH entries using 48 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 2020 total bytes of memory
BGP activity 9/1 prefixes, 14/6 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.1.1.1         4     4    1210    1212       13    0    0 20:01:17        0
```

1.6 BGP Tables and different ways of injecting BGP routes

BGP table is separate table containing the routes learned via BGP neighbors. After establishing an adjacency, the neighbors exchange their best BGP routes. Each router collects these routes from each neighbor with which it successfully established an adjacency and places them in its BGP forwarding database. The routes are learnt in the BGP table and the best routes are installed in the IP routing table. It's not important that every best BGP route is placed in the ip routing table. If a route is learned via another routing table having a better a better administrative distance then that route is installed in the routing table and the BGP table states such routes with a message called RIB Failure.
A BGP router adds entries to its local BGP table by using the same methods of IGP such as using network command, By redistribution or by learning from a neighboring BGP router.



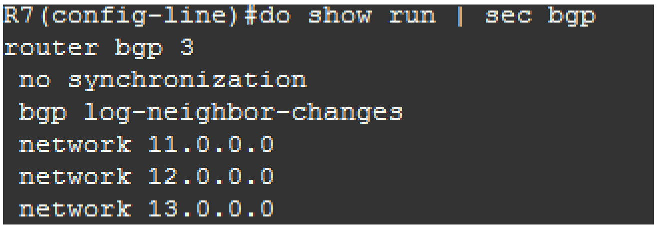
Network diagram for this section

Injecting routes using BGP network command

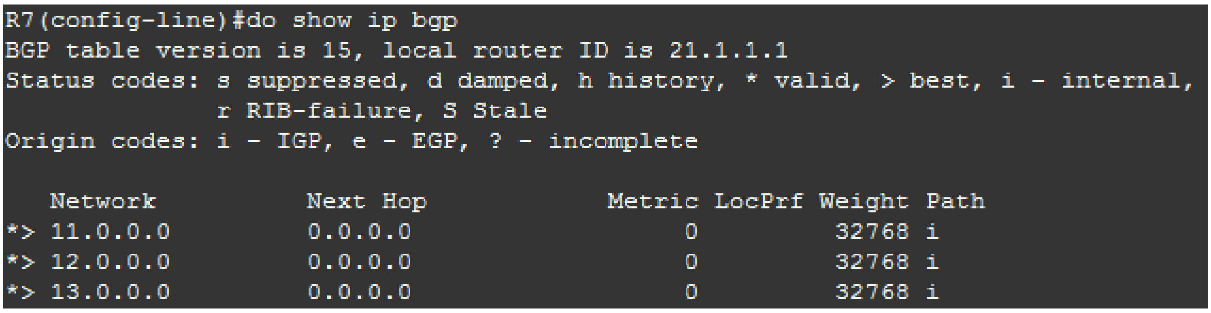
The BGP network command instructs the router to look for network as defined in the network command in the ip routing table. If the network is listed in the ip routing table the network is injected into BGP table.

Command Format network [subnet] [mask] route-map

As an example in the above diagram the prefixes listed on router R7 are injected using the network command in the BGP table.



```
R7(config-line)#do show run | sec bgp
router bgp 3
 no synchronization
 bgp log-neighbor-changes
 network 11.0.0.0
 network 12.0.0.0
 network 13.0.0.0
```



```
R7(config-line)#do show ip bgp
BGP table version is 15, local router ID is 21.1.1.1
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
*> 11.0.0.0         0.0.0.0                  0         32768 i
*> 12.0.0.0         0.0.0.0                  0         32768 i
*> 13.0.0.0         0.0.0.0                  0         32768 i
```

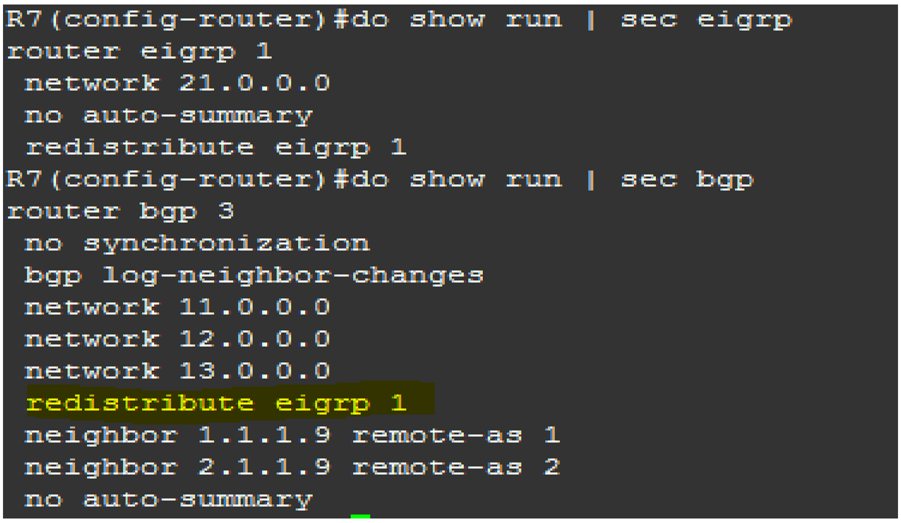
In above command the network command uses the default subnet mask of classful network. The next hop address is listed as 0.0.0.0 indicating that routes were injected on this router. Auto summary is turned off by default. But if the command is configured with auto summary enabled the results can be different.

Example if in the above figure if the 11.0.0.0/8 prefix is changed to 11.1.1.0/24 and auto summary is enabled on the router and if the network command list the network as 11.0.0.0 (classful) then since a matching subnet of this rangeid=s present in the ip routing table this network 11.0.0.0 is inserted into the bgp table.

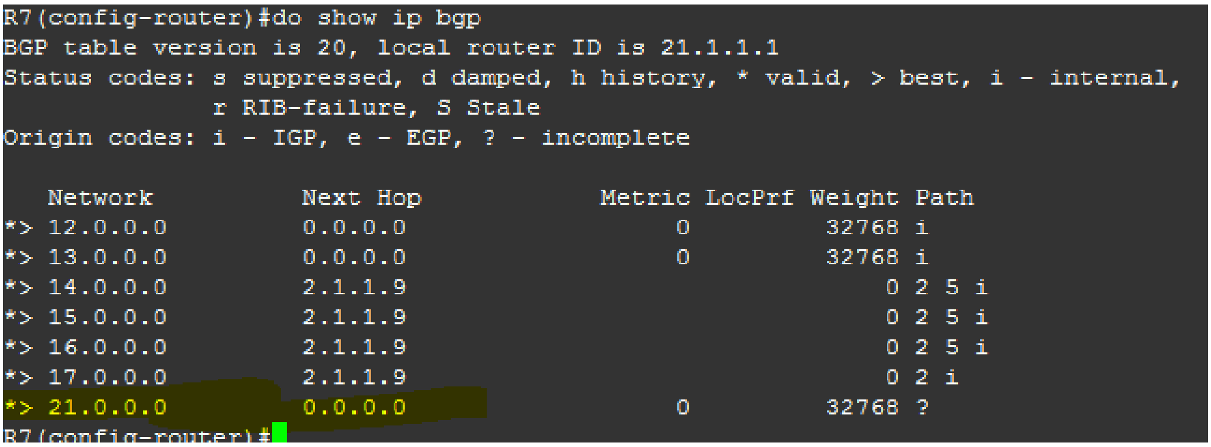
Inserting Routes using Redistribute command

Routes can be inserted into the BGP table from IGP protocols, static routes and default routes using redistribute command.BGP does not use the concept of metrics like IGP so redistributing does not require considering setting the metrics during redistribution process. Instead BGp uses path attributes to select the best route. to control the redistribution of routes bgp can use various filters such as distribute lists, prefix list, as path filters etc to filter the routes during redistribution and while sending the updates.

As an example in the above network diagram router R7 installs a route into its bgp table by redistributing route from the EIGRP protocol running on the router. As can be seen bgp uses a redistribute command and the network is injected into the BGP table as seen below in the show ip bgp table command. The impact of auto summary command is different on the redistribute process. If any matching prefix is found in the ip routing table while redistributing, then that prefix is placed in the BGP table along with the same network summarized to its classful network boundry.



```
R7(config-router)#do show run | sec eigrp
router eigrp 1
 network 21.0.0.0
 no auto-summary
 redistribute eigrp 1
R7(config-router)#do show run | sec bgp
router bgp 3
 no synchronization
 bgp log-neighbor-changes
 network 11.0.0.0
 network 12.0.0.0
 network 13.0.0.0
 redistribute eigrp 1
 neighbor 1.1.1.9 remote-as 1
 neighbor 2.1.1.9 remote-as 2
 no auto-summary
```

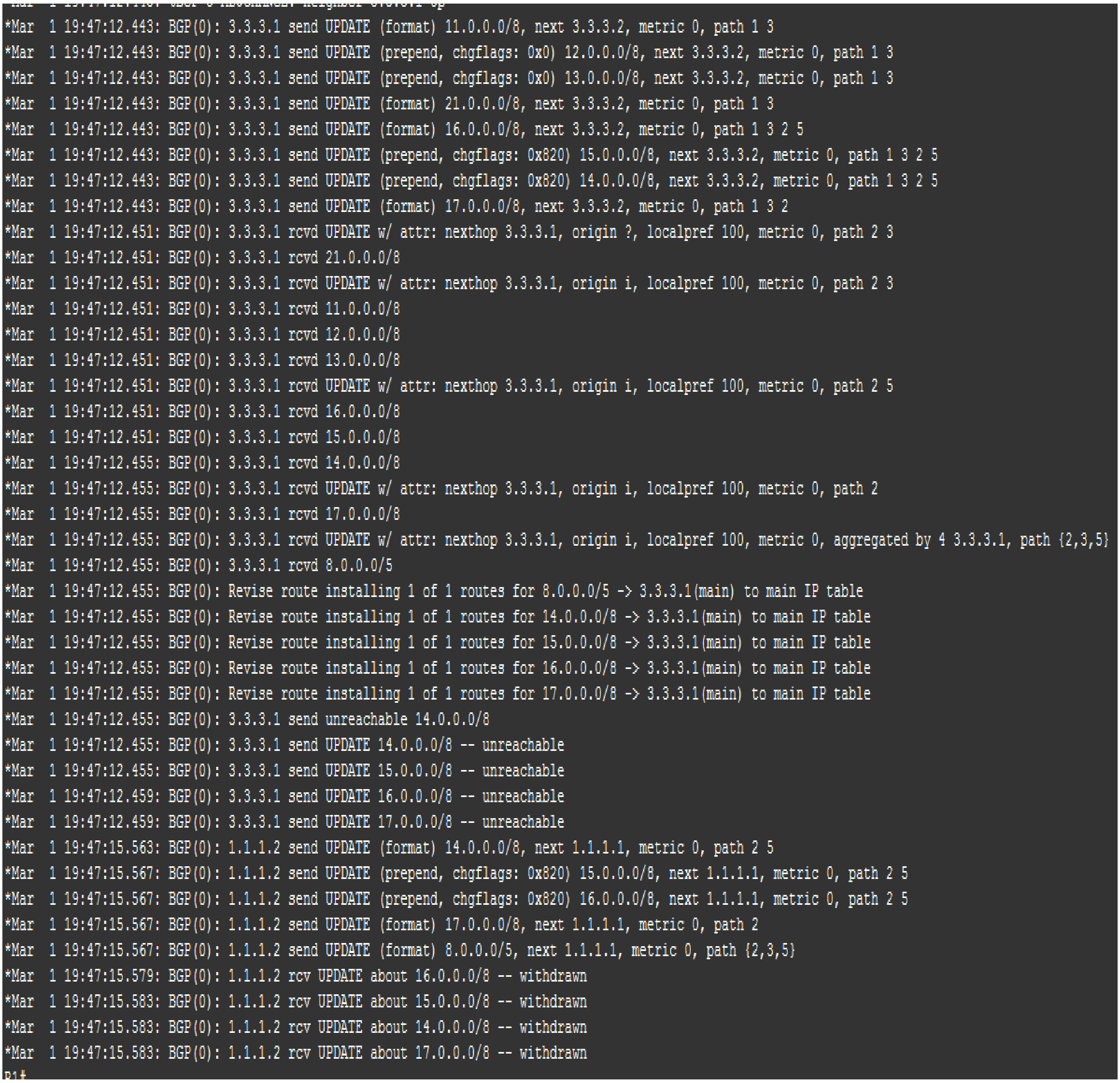


```
R7(config-router)#do show ip bgp
BGP table version is 20, local router ID is 21.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 12.0.0.0         0.0.0.0                  0         32768 i
*> 13.0.0.0         0.0.0.0                  0         32768 i
*> 14.0.0.0         2.1.1.9                                0 2 5 i
*> 15.0.0.0         2.1.1.9                                0 2 5 i
*> 16.0.0.0         2.1.1.9                                0 2 5 i
*> 17.0.0.0         2.1.1.9                                0 2 i
*> 21.0.0.0         0.0.0.0                  0         32768 ?
R7(config-router)#
```

<u>Receiving and sending bgp routes using bgp update messages</u>

BGP router takes routes from the local bgp table and advertsises the best routes in the BGP table to the neighboring BGP routers, where the neighbors decide using the the BGP decison process which routes to use for forwarding and installing in the IP routing table.

Below snapshot indicates the update message received by router R1 from Router R2 with respect to the network diagram mentioned in the previous section. As can be seen the update message contains list of prefixes, path attributes used by those prefixes and a list of withdrawn routes. The format of BGP update message has already been shared in the update message section discussed previously.



```
*Mar  1 19:47:12.443: BGP(0): 3.3.3.1 send UPDATE (format) 11.0.0.0/8, next 3.3.3.2, metric 0, path 1 3
*Mar  1 19:47:12.443: BGP(0): 3.3.3.1 send UPDATE (prepend, chgflags: 0x0) 12.0.0.0/8, next 3.3.3.2, metric 0, path 1 3
*Mar  1 19:47:12.443: BGP(0): 3.3.3.1 send UPDATE (prepend, chgflags: 0x0) 13.0.0.0/8, next 3.3.3.2, metric 0, path 1 3
*Mar  1 19:47:12.443: BGP(0): 3.3.3.1 send UPDATE (format) 21.0.0.0/8, next 3.3.3.2, metric 0, path 1 3
*Mar  1 19:47:12.443: BGP(0): 3.3.3.1 send UPDATE (format) 16.0.0.0/8, next 3.3.3.2, metric 0, path 1 3 2 5
*Mar  1 19:47:12.443: BGP(0): 3.3.3.1 send UPDATE (prepend, chgflags: 0x820) 15.0.0.0/8, next 3.3.3.2, metric 0, path 1 3 2 5
*Mar  1 19:47:12.443: BGP(0): 3.3.3.1 send UPDATE (prepend, chgflags: 0x820) 14.0.0.0/8, next 3.3.3.2, metric 0, path 1 3 2 5
*Mar  1 19:47:12.443: BGP(0): 3.3.3.1 send UPDATE (format) 17.0.0.0/8, next 3.3.3.2, metric 0, path 1 3 2
*Mar  1 19:47:12.451: BGP(0): 3.3.3.1 rcvd UPDATE w/ attr: nexthop 3.3.3.1, origin ?, localpref 100, metric 0, path 2 3
*Mar  1 19:47:12.451: BGP(0): 3.3.3.1 rcvd 21.0.0.0/8
*Mar  1 19:47:12.451: BGP(0): 3.3.3.1 rcvd UPDATE w/ attr: nexthop 3.3.3.1, origin i, localpref 100, metric 0, path 2 3
*Mar  1 19:47:12.451: BGP(0): 3.3.3.1 rcvd 11.0.0.0/8
*Mar  1 19:47:12.451: BGP(0): 3.3.3.1 rcvd 12.0.0.0/8
*Mar  1 19:47:12.451: BGP(0): 3.3.3.1 rcvd 13.0.0.0/8
*Mar  1 19:47:12.451: BGP(0): 3.3.3.1 rcvd UPDATE w/ attr: nexthop 3.3.3.1, origin i, localpref 100, metric 0, path 2 5
*Mar  1 19:47:12.451: BGP(0): 3.3.3.1 rcvd 16.0.0.0/8
*Mar  1 19:47:12.451: BGP(0): 3.3.3.1 rcvd 15.0.0.0/8
*Mar  1 19:47:12.455: BGP(0): 3.3.3.1 rcvd 14.0.0.0/8
*Mar  1 19:47:12.455: BGP(0): 3.3.3.1 rcvd UPDATE w/ attr: nexthop 3.3.3.1, origin i, localpref 100, metric 0, path 2
*Mar  1 19:47:12.455: BGP(0): 3.3.3.1 rcvd 17.0.0.0/8
*Mar  1 19:47:12.455: BGP(0): 3.3.3.1 rcvd UPDATE w/ attr: nexthop 3.3.3.1, origin i, localpref 100, metric 0, aggregated by 4 3.3.3.1, path {2,3,5}
*Mar  1 19:47:12.455: BGP(0): 3.3.3.1 rcvd 8.0.0.0/5
*Mar  1 19:47:12.455: BGP(0): Revise route installing 1 of 1 routes for 8.0.0.0/5 -> 3.3.3.1(main) to main IP table
*Mar  1 19:47:12.455: BGP(0): Revise route installing 1 of 1 routes for 14.0.0.0/8 -> 3.3.3.1(main) to main IP table
*Mar  1 19:47:12.455: BGP(0): Revise route installing 1 of 1 routes for 15.0.0.0/8 -> 3.3.3.1(main) to main IP table
*Mar  1 19:47:12.455: BGP(0): Revise route installing 1 of 1 routes for 16.0.0.0/8 -> 3.3.3.1(main) to main IP table
*Mar  1 19:47:12.455: BGP(0): Revise route installing 1 of 1 routes for 17.0.0.0/8 -> 3.3.3.1(main) to main IP table
*Mar  1 19:47:12.455: BGP(0): 3.3.3.1 send unreachable 14.0.0.0/8
*Mar  1 19:47:12.455: BGP(0): 3.3.3.1 send UPDATE 14.0.0.0/8 -- unreachable
*Mar  1 19:47:12.455: BGP(0): 3.3.3.1 send UPDATE 15.0.0.0/8 -- unreachable
*Mar  1 19:47:12.459: BGP(0): 3.3.3.1 send UPDATE 16.0.0.0/8 -- unreachable
*Mar  1 19:47:12.459: BGP(0): 3.3.3.1 send UPDATE 17.0.0.0/8 -- unreachable
*Mar  1 19:47:15.563: BGP(0): 1.1.1.2 send UPDATE (format) 14.0.0.0/8, next 1.1.1.1, metric 0, path 2 5
*Mar  1 19:47:15.567: BGP(0): 1.1.1.2 send UPDATE (prepend, chgflags: 0x820) 15.0.0.0/8, next 1.1.1.1, metric 0, path 2 5
*Mar  1 19:47:15.567: BGP(0): 1.1.1.2 send UPDATE (prepend, chgflags: 0x820) 16.0.0.0/8, next 1.1.1.1, metric 0, path 2 5
*Mar  1 19:47:15.567: BGP(0): 1.1.1.2 send UPDATE (format) 17.0.0.0/8, next 1.1.1.1, metric 0, path 2
*Mar  1 19:47:15.567: BGP(0): 1.1.1.2 send UPDATE (format) 8.0.0.0/5, next 1.1.1.1, metric 0, path {2,3,5}
*Mar  1 19:47:15.579: BGP(0): 1.1.1.2 rcv UPDATE about 16.0.0.0/8 -- withdrawn
*Mar  1 19:47:15.583: BGP(0): 1.1.1.2 rcv UPDATE about 15.0.0.0/8 -- withdrawn
*Mar  1 19:47:15.583: BGP(0): 1.1.1.2 rcv UPDATE about 14.0.0.0/8 -- withdrawn
*Mar  1 19:47:15.583: BGP(0): 1.1.1.2 rcv UPDATE about 17.0.0.0/8 -- withdrawn
R1#
```

1.7 BGP attributes

BGP routers send BGP update messages about destination networks to other BGP routers. Update messages can contain network layer reachability information, which is a list of one or more networks (IP address prefixes and their prefix lengths), and path attributes, which are a set of BGP metrics describing the path to these networks. BGP uses the path attributes to determine the best path to the networks.
Path attributes fall into four separate categories :

1 Well-known mandatory : Must appear in all BGP update messages.
A)AS PATH
B) NEXT HOP

C) ORIGIN

2) Well-known discretionary does not have to be present in all BGP update messages but it is well recognized by BGP
A) Local preference
B) Atomic aggregate

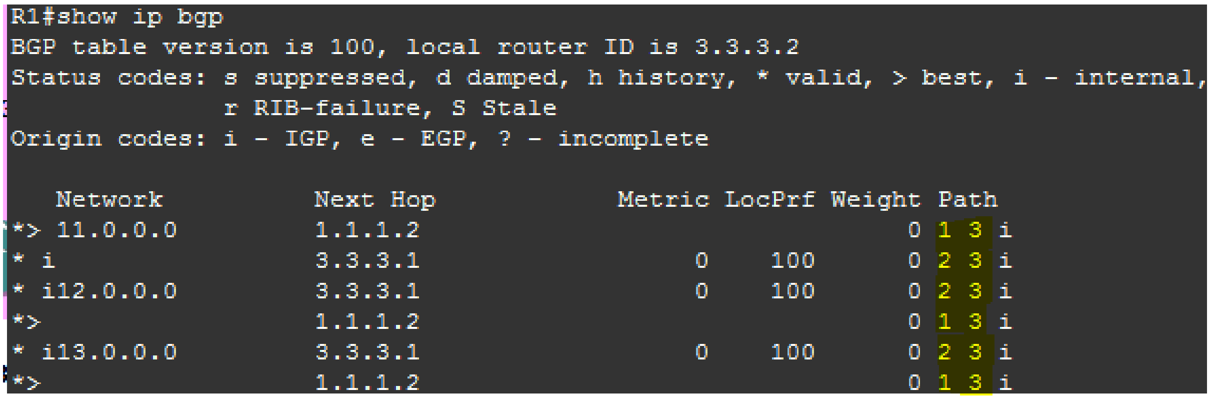
3)Optional transitive : BGP routers that do not implement an optional transitive attribute should pass it to other BGP routers untouched and mark the attribute as partial.
A) aggregator
B) community

4)Optional non transitive : BGP routers that do not implement an optional nontransitive attribute must delete the attribute and must not pass it to other BGP routers.
A) Multi exit descriminator(MED)

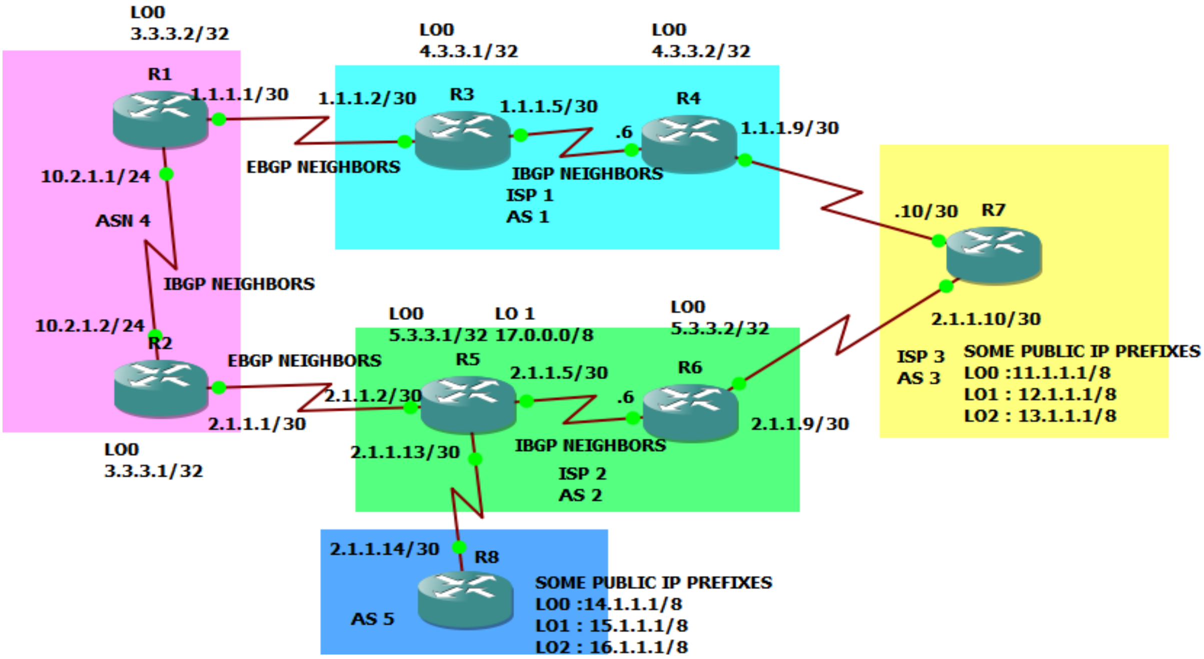
A path attribute is of variable length and consists of three fields:
1) Attribute type, which consists of a 1-byte attribute flags field and a 1-byte attribute-type code field
2) Attribute length
3) Attribute value
The first bit of the attribute flags field indicates whether the attribute is optional or well known. The second bit indicates whether an optional attribute is transitive or nontransitive. The third bit indicates whether a transitive attribute is partial or complete. The fourth bit indicates whether the attribute length field is 1 or 2 bytes. The rest of the flag bits are unused and are set to 0.

1.7.1 AS path attribute

It is a list of autonomous system numbers that a route has traversed to reach a destination, with the number of Autonomous system that originated the route at the end of the list.
When a route passes through an Autonomous system and its is advertised to the neighboring autonomous system in an EBGP update the AS number of the sending AS is prepended to the AS path list. As can be seen in the example network shown below when the routes advertised by BGP on router R7 are received at router R1 all transit AS paths are prepended in the AS list. As can be seen in the snapshot when the routes from router R7 are advertised the AS numbers are prepended to it from right to left with the most recent on the left side. So to reach the networks connected to router R7 the packet traverses first through AS1 then AS 3 and then it reaches its destination. This attribute also helps in mitigating loops. If the AS who advertised the route sees its own AS in the update it simply ignores the update as it knows that the route has already traversed that AS.



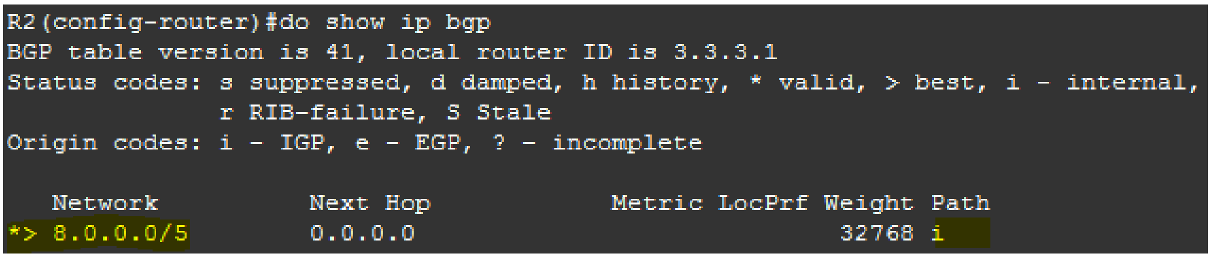
```
R1#show ip bgp
BGP table version is 100, local router ID is 3.3.3.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.0.0.0         1.1.1.2                                0 1 3 i
* i                 3.3.3.1                  0    100      0 2 3 i
* i12.0.0.0         3.3.3.1                  0    100      0 2 3 i
*>                  1.1.1.2                                0 1 3 i
* i13.0.0.0         3.3.3.1                  0    100      0 2 3 i
*>                  1.1.1.2                                0 1 3 i
```



Network diagram for this section

The AS path attribute consists of upto 4 different components called segments as follows:
1) AS_SEQ : AS_SEQ is a path attribute covered above which contains a prepended lists of AS numbers for routes.

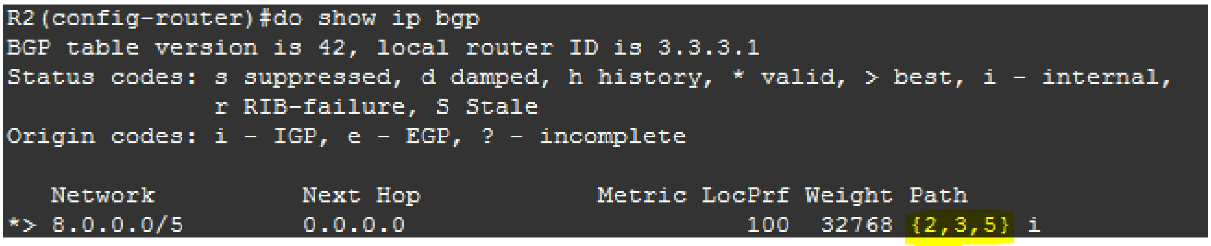
2) AS_SET : This is used when a manual summary is created using an aggregate-address command. Now suppose there are different routes in the BGP table having different list of AS_SEQ numbers. Ad such what AS numbers shall be included along with the summarized route and in what order. When the AS numbers listed in any one route or even a single AS number in any route is different in such a way that routes which are summarized don't have the same AS numbers, then the aggregate-address command creates a null AS i.e no As number is listed in the AS path list. As an example in the above network diagram route aggregation is configured on router R2 for routes betwwen 11.0.0.0/8 to 15.0.0.0/8 using 8.0.0.0/5 summary route and since they have originated in different AS and the listed AS numbers are different a null AS path is created as can be seen in the BGP table of R2 router.



```
R2(config-router)#do show ip bgp
BGP table version is 41, local router ID is 3.3.3.1
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
*> 8.0.0.0/5        0.0.0.0                            32768 i
```

As such routing loops can occur if this summary route is advertised to the routers through which the prefixes belonging to this summary route have already traversed.
To avoid this AS SET feature is used in the aggregate address command. When as-set is used all the as numbers in the routes belonging to the summary are sent in AS path attribute list of the summary route. The list is a random list and is used to give any idea to the other routers that through which ASes the route has traversed. Snapshot after using the as-set command. Aggregate-address command can be configured with the various options such as :

1)summary-only only summary route is sent
2)normal - summary routes and the individual routes belonging to that summary are sent.
3)suppress-map - certain routes belonging to the summary are advertised along with summary.



```
R2(config-router)#do show ip bgp
BGP table version is 42, local router ID is 3.3.3.1
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
*> 8.0.0.0/5        0.0.0.0                       100  32768 {2,3,5} i
```

3) AS_CONFED_SEQ : It is a path attribute component which contains a prepended lists of AS numbers for routes in a confederation. Confederation concepts are covered later.

4)AS_CONFED_SET : It is a path attribute component which is same as AS_SET component but it is used in confederations.

1.7.2 Next HOP path attribute

The BGP next-hop attribute is a well-known mandatory attribute that indicates the next-hop IP address that is to be used to reach a destination. BGP like IGP is a hop by hop routing protocol with each hop referring to an AS. The next-hop address for a network from another autonomous system is an IP address of the entry point of the next autonomous system along the path to that destination network. For EBGP, the next-hop address is the IP address of the neighbor that sent the update. For IBGP the concepts are already discussed in the IBGP neighbor section.

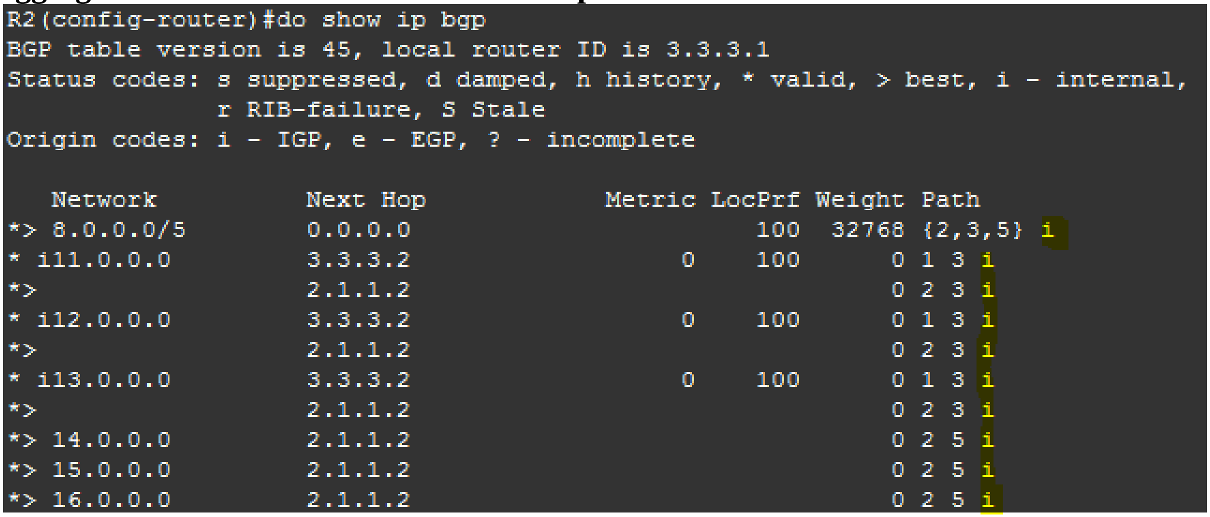
Origin Path attribute

The origin is a well-known mandatory attribute that defines the origin of the path information. The origin PA provides a general information as to how a particular NLRI was injected into the BGP table. It can be IGP, EGP OR INCOMPLETE(indicated by a ?)

IGP(i): is the origin code when the route is injected by the network command, aggregate-address command(some cases) or neighbor default-originate command
EGP(E) : is the origin code when the route is injected by Exterior gateway Protocol
Incomplete : It is the code when the route is injected using redistribute command or default information originate command or aggregate-address command in some cases.

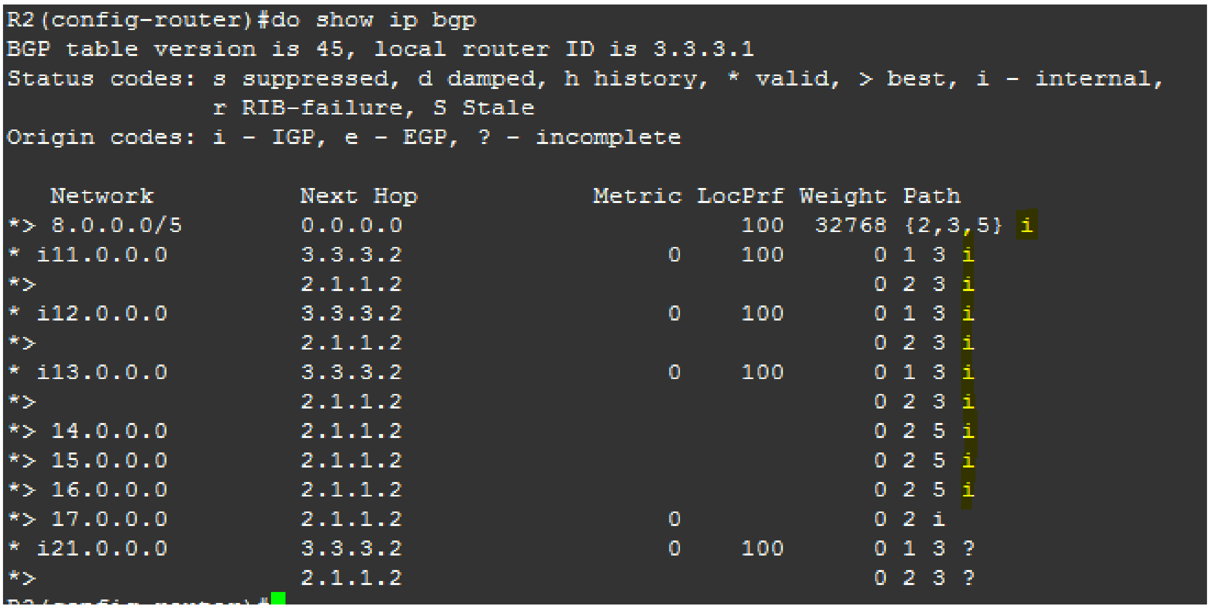
For aggregate-address command the following are the special cases for origin code:
1) If as-set option is not used then the origin code for aggregate route is i.
2)If as-set option is used and all the component routes have origin code of i then the origin code is i.
3)If as-set option is used an even a single subset route of summary has an incomplete origin then the origin becomes incomplete for the summarized route.

As an example lets analyse the above network diagram and consider router R2. For the aggregate- address command if no as-set option is used.



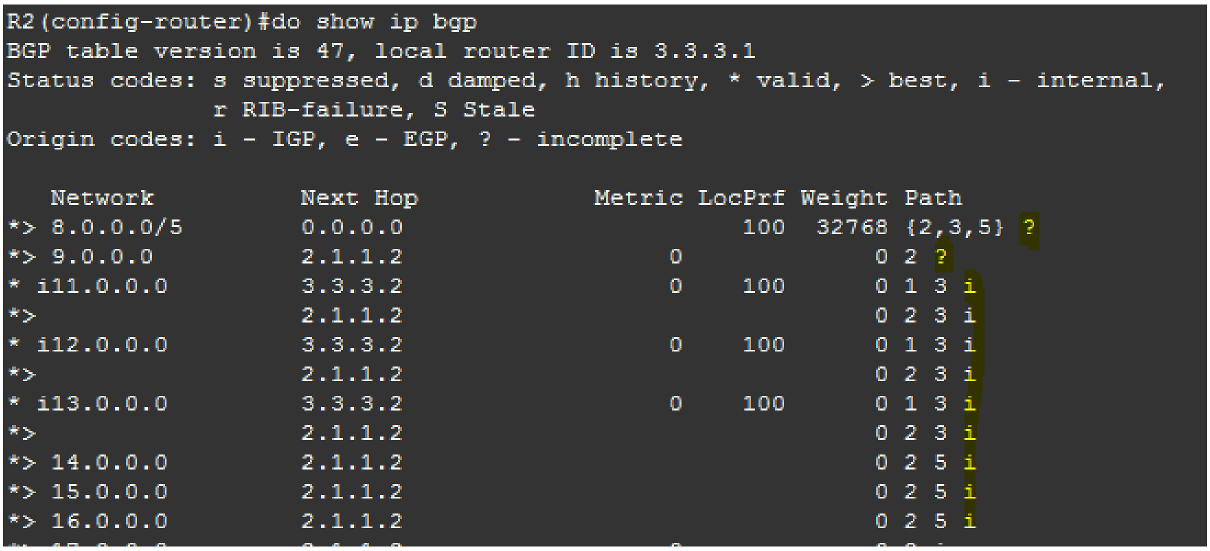
```
R2(config-router)#do show ip bgp
BGP table version is 45, local router ID is 3.3.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 8.0.0.0/5        0.0.0.0                       100  32768 {2,3,5} i
* i11.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
* i12.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
* i13.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
*> 14.0.0.0         2.1.1.2                                0 2 5 i
*> 15.0.0.0         2.1.1.2                                0 2 5 i
*> 16.0.0.0         2.1.1.2                                0 2 5 i
```

If as-set option is used and all routes have a origin code of i.



```
R2(config-router)#do show ip bgp
BGP table version is 45, local router ID is 3.3.3.1
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
*> 8.0.0.0/5        0.0.0.0                       100  32768 {2,3,5} i
* i11.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
* i12.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
* i13.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
*> 14.0.0.0         2.1.1.2                                0 2 5 i
*> 15.0.0.0         2.1.1.2                                0 2 5 i
*> 16.0.0.0         2.1.1.2                                0 2 5 i
*> 17.0.0.0         2.1.1.2                  0             0 2 i
* i21.0.0.0         3.3.3.2                  0    100      0 1 3 ?
*>                  2.1.1.2                                0 2 3 ?
R2(config-router)#
```

If as-set option is used and atleast one route has an origin code of ?. 9.0.0.0 prefix has an origin code of ? so the summarized route origin code also changed to ?



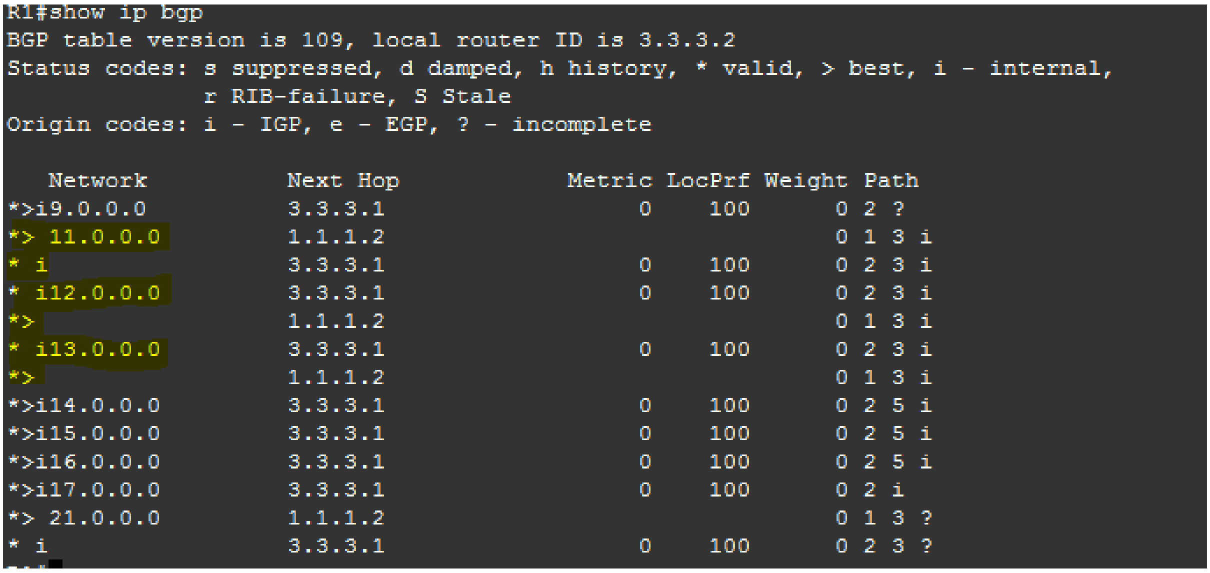
```
R2(config-router)#do show ip bgp
BGP table version is 47, local router ID is 3.3.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 8.0.0.0/5        0.0.0.0                       100  32768 {2,3,5} ?
*> 9.0.0.0          2.1.1.2                  0             0 2 ?
* i11.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
* i12.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
* i13.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
*> 14.0.0.0         2.1.1.2                                0 2 5 i
*> 15.0.0.0         2.1.1.2                                0 2 5 i
*> 16.0.0.0         2.1.1.2                                0 2 5 i
```

1.7.4 Local Preference Path Attribute

It is a well known discretionary attribute that indicates to the routers in an AS which path to take to exit an autonomous system The higher the local preference the more preferred the path is. By default local preference is 100 in cisco routers but can be modified using set local preference command in route maps. as an example considering the above same network lets check the bgp table of R1 and R2 and modify the exit paths takes by these routers.

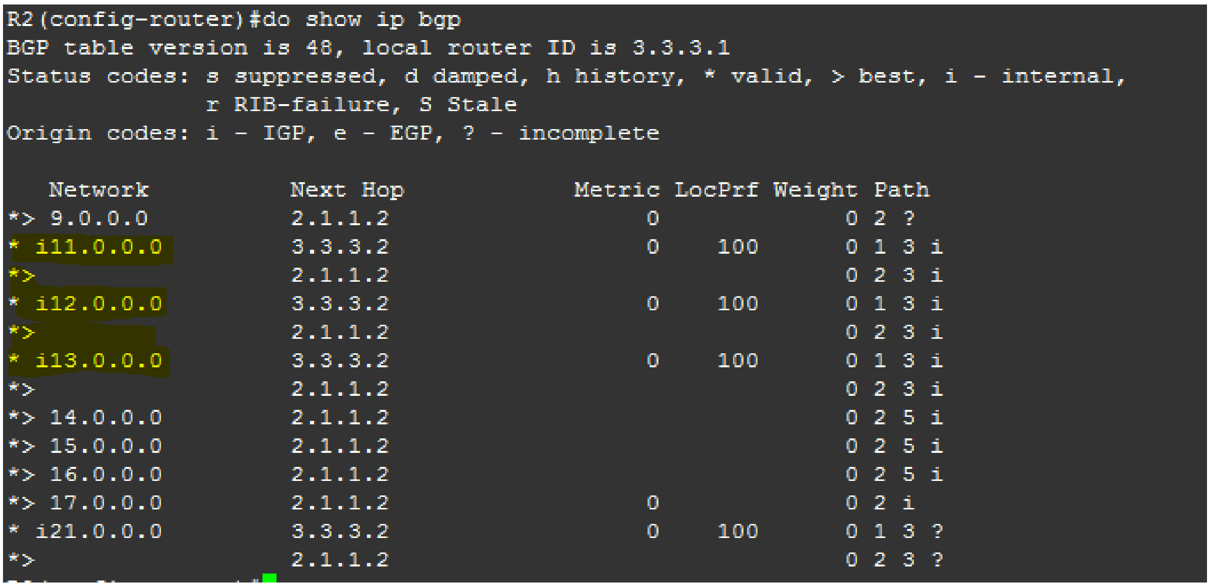
Show ip bgp

R1



```
R1#show ip bgp
BGP table version is 109, local router ID is 3.3.3.2
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
*>i9.0.0.0          3.3.3.1                  0    100      0 2 ?
*> 11.0.0.0         1.1.1.2                                0 1 3 i
* i                 3.3.3.1                  0    100      0 2 3 i
* i12.0.0.0         3.3.3.1                  0    100      0 2 3 i
*>                  1.1.1.2                                0 1 3 i
* i13.0.0.0         3.3.3.1                  0    100      0 2 3 i
*>                  1.1.1.2                                0 1 3 i
*>i14.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i15.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i16.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i17.0.0.0         3.3.3.1                  0    100      0 2 i
*> 21.0.0.0         1.1.1.2                                0 1 3 ?
* i                 3.3.3.1                  0    100      0 2 3 ?
```

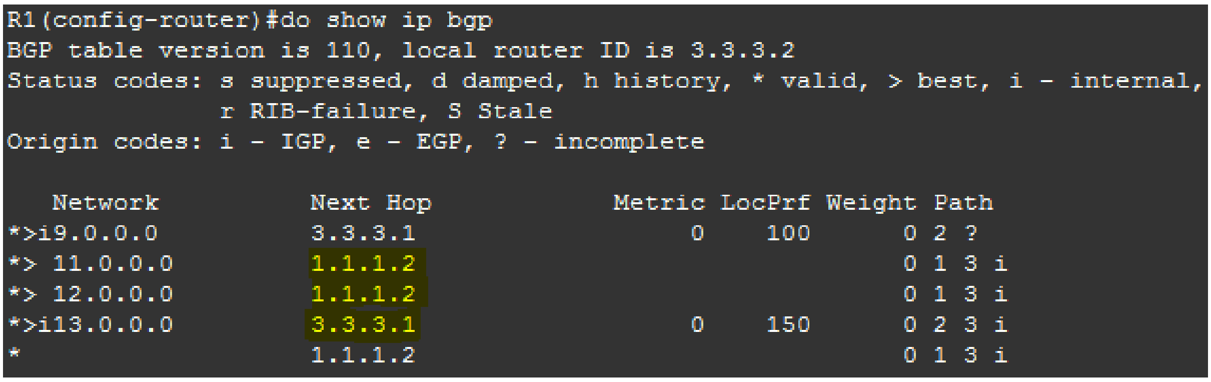
R2



```
R2(config-router)#do show ip bgp
BGP table version is 48, local router ID is 3.3.3.1
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
*> 9.0.0.0          2.1.1.2                  0             0 2 ?
* i11.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
* i12.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
* i13.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
*> 14.0.0.0         2.1.1.2                                0 2 5 i
*> 15.0.0.0         2.1.1.2                                0 2 5 i
*> 16.0.0.0         2.1.1.2                                0 2 5 i
*> 17.0.0.0         2.1.1.2                  0             0 2 i
* i21.0.0.0         3.3.3.2                  0    100      0 1 3 ?
*>                  2.1.1.2                                0 2 3 ?
```

Now as can be seen R1 has the best path for prefixes 11.0.0.0,12.0.0.0,13.0.0.0 through next hop 1.1.1.2 which goes through AS 1. While R2 has the same routes with next hop of 2.1.1.2 autonomous system 2. This happened because the local preference value for both routers for these routes is same due to which they prefer EBGP path over IBGP learned path. This matrix as which PA acts as tie breaker would be discussed in a later section. For now lets modify the local preference such that for route 11.0.0.0,12.0.0.0 the next hop path is through AS 1 and for 13.0.0.0 the exit point is AS 5.
For this we will modify the local preference value to 150 on R1 for 11.0.0.0,12.0.0.0 routes and on R2 to 150 for 13.0.0.0 route.

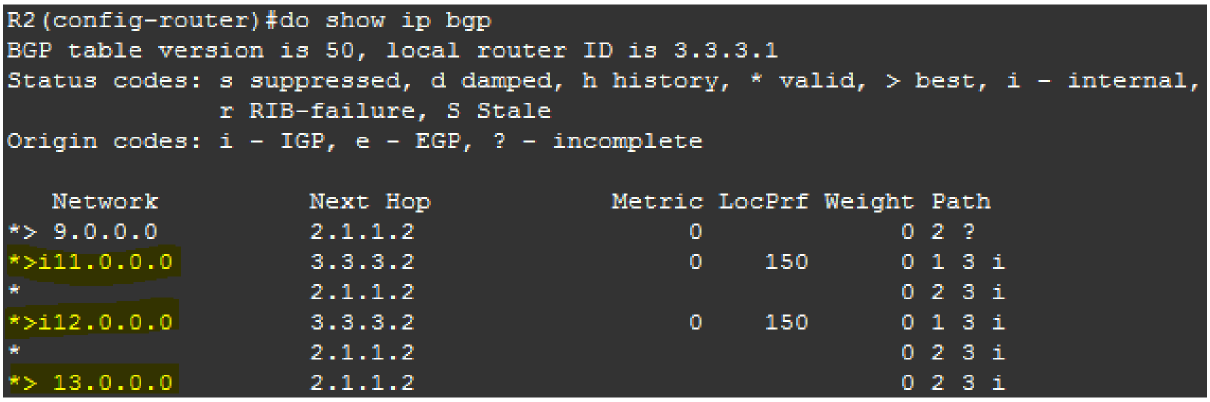
R1 output



```
R1(config-router)#do show ip bgp
BGP table version is 110, local router ID is 3.3.3.2
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
*>i9.0.0.0          3.3.3.1                  0    100      0 2 ?
*> 11.0.0.0         1.1.1.2                                0 1 3 i
*> 12.0.0.0         1.1.1.2                                0 1 3 i
*>i13.0.0.0         3.3.3.1                  0    150      0 2 3 i
*                   1.1.1.2                                0 1 3 i
```

As can be seen the next hop for 13.0.0.0 route point towards R2 and it also lists the value of 150 for local preference making it the best exit path for 13.0.0.0 route. Also the adition next hops although they were not the best which were listed for 11.0.0.0 and 12.0.0.0 are no longer visible giving a proof that only best routes are advertised from the bgp table and since the R2 router has installed the best routes for 11.0.0.0 and 12.0.0.0 through R1 thise are no longer visible as R2 is no longer sending its alternate paths which are through AS5 to R1.

R2 output



```
R2(config-router)#do show ip bgp
BGP table version is 50, local router ID is 3.3.3.1
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
*> 9.0.0.0          2.1.1.2                  0             0 2 ?
*>i11.0.0.0         3.3.3.2                  0    150      0 1 3 i
*                   2.1.1.2                                0 2 3 i
*>i12.0.0.0         3.3.3.2                  0    150      0 1 3 i
*                   2.1.1.2                                0 2 3 i
*> 13.0.0.0         2.1.1.2                                0 2 3 i
```

Configuration commands on R1 for above results

```
access-list 1 permit 11.0.0.0 0.255.255.255
access-list 1 permit 12.0.0.0 0.255.255.255
route-map local_pref permit 5
 match ip address 1
 set local-preference 150
!
route-map local_pref permit 10
!
router bgp 4
neighbor 3.3.3.1 remote-as 4
 neighbor 3.3.3.1 update-source Loopback0
 neighbor 3.3.3.1 next-hop-self
 neighbor 3.3.3.1 route-map local_pref out
 no auto-summary
```

Configuration commands on R2 for above results

```
access-list 1 permit 13.0.0.0 0.255.255.255
route-map local_pref permit 5
 match ip address 1
 set local-preference 150
!
route-map local_pref permit 10
!
router bgp 4
neighbor 3.3.3.2 remote-as 4
 neighbor 3.3.3.2 update-source Loopback0
 neighbor 3.3.3.2 next-hop-self
 neighbor 3.3.3.2 route-map local_pref out
 no auto-summary
```

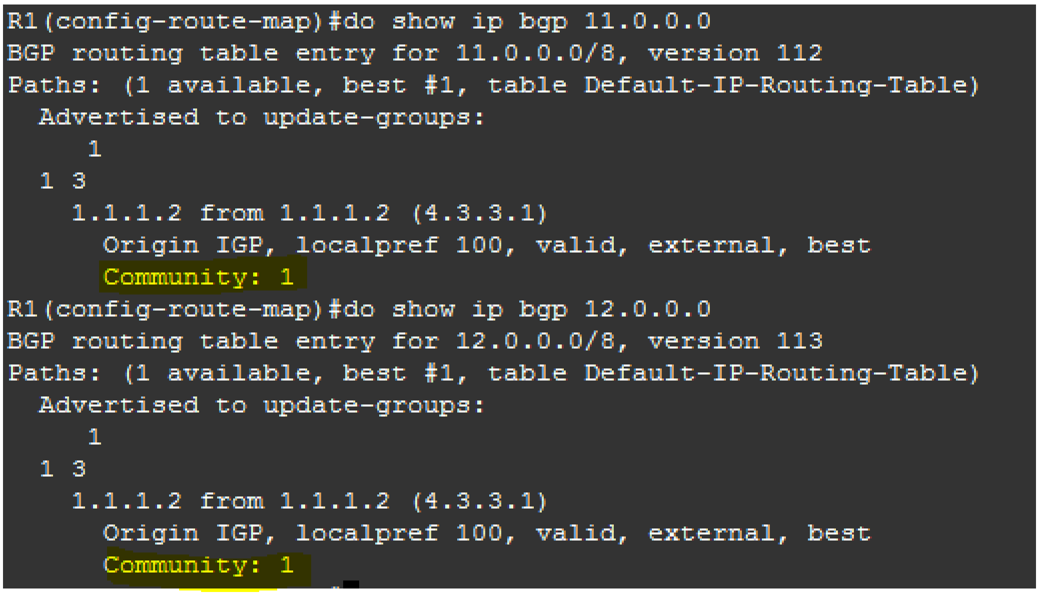
Neighbor [ip address] route-map[in|out] command is used to apply the policies in the rquired direction. As in the above case he route-map is applied in the outbound directions and a clear ip bgp * out command is issued so that the update is resent to the neighbor with the changes.

1.7.5 Community Attribute

BGP communities can be used to filter incoming or outgoing routing routes. BGP communities allow routers to tag routes with a community value and this community vale can be matched on the remote router and can be used to discard the route or modify the PA from those routes. bgp communities allow routers in one AS to communicate policy information to routers in other Autonomous system. It is an optional transitive attribute and can even pass through AS which don't even understand community PA.

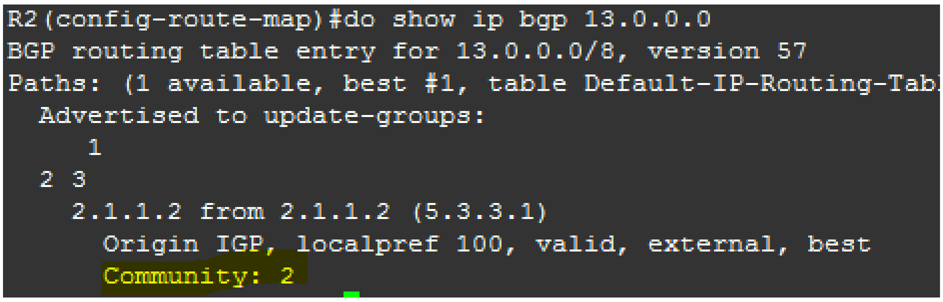
In the network diagram, we have done the modification of local preference by matching the access-list in the route map. Now we will modify the local preference by matching the community list. For this we will set the community value to 1 for routes 11.0.0.0 and 12.0.0.0 on router R3 in autonomous system 1 and then propagate that community to autonomous system 4.
For 13.0.0.0 route we will set the community value to 2 on router R5 in AS2.
We have to enable send community option on both routers R3 and R5 to end the community value in updates. Then will match the community value on R1 and R2 using community list and set the local preference

As can be seen after configuration community value of 1 is seen for the routes 11.0.0.0 and 12.0.0.0 on R1.



```
R1(config-route-map)#do show ip bgp 11.0.0.0
BGP routing table entry for 11.0.0.0/8, version 112
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     1
  1 3
    1.1.1.2 from 1.1.1.2 (4.3.3.1)
      Origin IGP, localpref 100, valid, external, best
      Community: 1
R1(config-route-map)#do show ip bgp 12.0.0.0
BGP routing table entry for 12.0.0.0/8, version 113
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     1
  1 3
    1.1.1.2 from 1.1.1.2 (4.3.3.1)
      Origin IGP, localpref 100, valid, external, best
      Community: 1
```

Similarly on R2 community value of 2 is seen



```
R2(config-route-map)#do show ip bgp 13.0.0.0
BGP routing table entry for 13.0.0.0/8, version 57
Paths: (1 available, best #1, table Default-IP-Routing-Tab
  Advertised to update-groups:
     1
  2 3
    2.1.1.2 from 2.1.1.2 (5.3.3.1)
      Origin IGP, localpref 100, valid, external, best
      Community: 2
```

Configuration on R3 and R5 to create and send community

R3

```
access-list 1 permit 11.0.0.0 0.255.255.255
access-list 1 permit 12.0.0.0 0.255.255.255
route-map set_community permit 5
 match ip address 1
 set community 1
!
route-map set_community permit 10
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.1 remote-as 4
 neighbor 1.1.1.1 send-community both
 neighbor 1.1.1.1 route-map set_community out
```

R5

```
access-list 1 permit 13.0.0.0 0.255.255.255
route-map set_community permit 5
 match ip address 1
 set community 2
!
route-map set_community permit 10
!
router bgp 2
 neighbor 2.1.1.1 remote-as 4
 neighbor 2.1.1.1 send-community both
 neighbor 2.1.1.1 route-map set_community out
```

R1 and R2 configuration match community an set local preference

R1

```
ip community-list 1 permit 1
!
route-map local_pref permit 5
 match community 1
 set local-preference 150
!
route-map local_pref permit 10
```

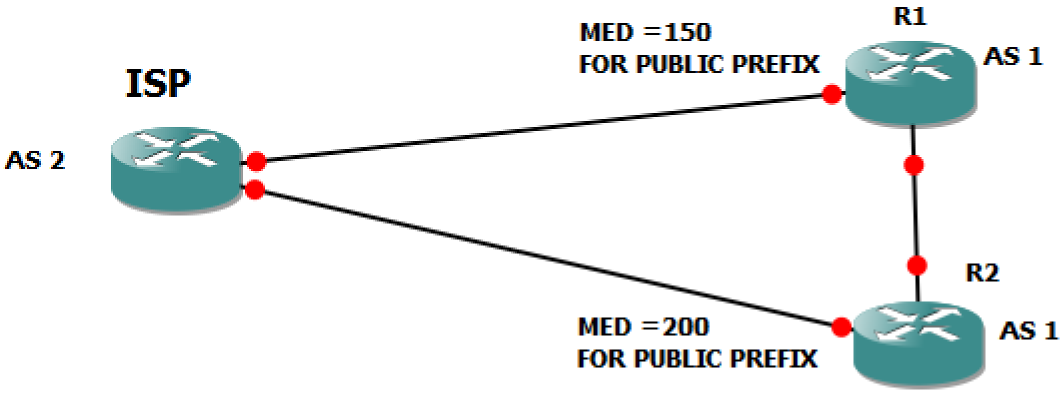
R2

```
ip community-list 1 permit 2
!
route-map local_pref permit 5
 match community 2
 set local-preference 150
!
```

route-map local_pref permit 10

1.7.6 MED Attribute

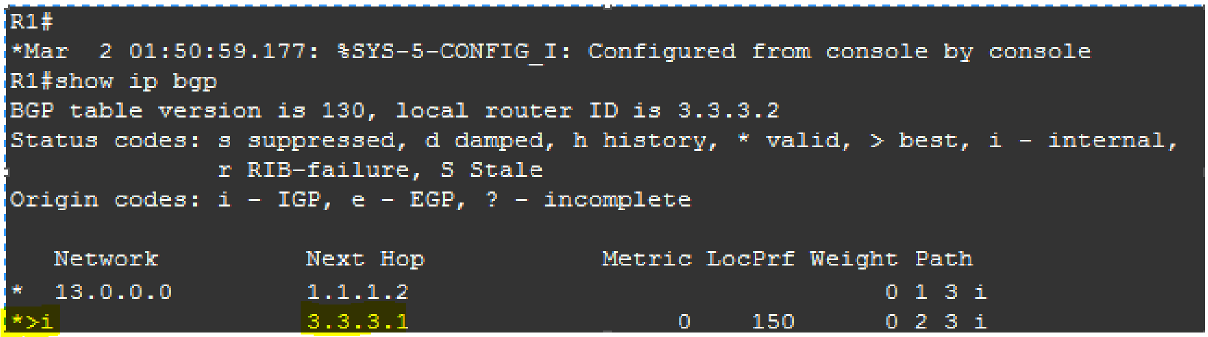
Multiexit discriminator indicates to external neighbors the preferred path into an autonomous system. This is a dynamic way for an autonomous system to try to influence another autonomous system as to which way it should choose to reach a certain route if there are multiple entry points into the autonomous system. The MED is sent to EBGP peers; those routers propagate the MED within their autonomous system, and the routers within the autonomous system use the MED, but do not pass it on to the next autonomous system. When the same update is passed on to another autonomous system, the metric will be set back to the default of 0.BGP is the only protocol that can affect the path used to send traffic into an autonomous system using MED. As can be seen in the below figure the inbound paths can be influenced. For the public prefix used by AS1 R1 sets MED Value to 150 while R2 sets the MED value to 200 and send the update to ISP. This influences the route choice of ISP who installs the router with least MED which is router R1 and hence all the inbound traffic is directed towards router R1.



1.7.7 Weight attribute (Cisco standard)

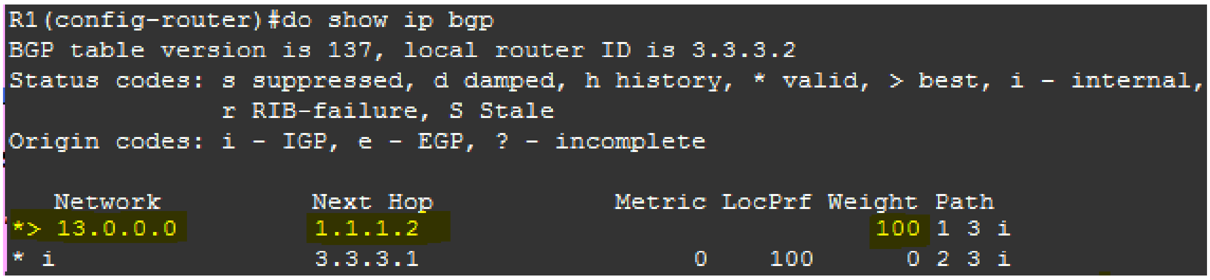
The weight attribute is a Cisco-defined attribute used for the path-selection process. The weight attribute is configured locally and provides local routing policy only; it is not propagated to any BGP neighbors. Routes with a higher weight are preferred when multiple routes to the same destination exist. 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. The weight attribute applies when using one router with multiple exit points out of an autonomous system, as compared to the local preference attribute, which is used when two or more routers provide multiple exit points.
As an example taking the network diagram into consideration we know that for 13.0.0.0 prefix the preferred path is through R2. we can set the weight using weight command in route map or directly in the neighbor command and choose router R3 or AS1 as an exit point for prefix 13.0.0.0.

Show ip BGP before weight command is used on R1
As can be seen 13.0.0.0 route is installed through router R2(IBGP neighbor) and the weight value is 0 for this route since it didn't originated on this router. Now we will increase the weight to 100 for this route and see the results.



```
R1#
*Mar  2 01:50:59.177: %SYS-5-CONFIG_I: Configured from console by console
R1#show ip bgp
BGP table version is 130, local router ID is 3.3.3.2
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
*  13.0.0.0         1.1.1.2                                0 1 3 i
*>i                 3.3.3.1                  0    150      0 2 3 i
```

After weight attribute is applied to the inbound updates coming from router R3 in AS 1



```
R1(config-router)#do show ip bgp
BGP table version is 137, local router ID is 3.3.3.2
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
*> 13.0.0.0         1.1.1.2                              100 1 3 i
* i                 3.3.3.1                  0    100      0 2 3 i
```

Note that the next hop has changed to 1.1.1.2 for the best route.

1.8 BGP route selection criteria

BGP uses a list of above mentioned path attributes and other tie breakers to determine the best path to the destination. When multiple routes to reach a destination are learnt a step wise approach taking into consideration the bgp path attributes is applied to determine the best path. The path attributes also provide additional list of tools in manipulating the routes taken by bgp. The steps in determining the best path are:

Step 0 : Next hop reachability : Although this step is not a part of standard approach of bgp to determine the best route but if the next hop is not reachable bgp never chooses the route and prefers the other route in place of that. BGP does this by performing a recursive lookup for next hop in the the ip routing table and if it does not find any route in the routing table for the next hop it ignores that route as the packets sent to that wont ever make to the destination.

Step 1 : Highest Weight (Cisco proprietary): This is a Cisco proprietary feature and is valid for locally manipulating the exit interface for routes received on the router. administartaive weight can be assigned a value between 0 to 65535. By default for the route locally generated on the router weight is 32768. Weight attribute can't be propagated to the neighboring routers.

Step 2 : Highest Local preference : This path attribute is highly useful in manipulating the exit path for all routers within an AS. Higher the local preference more preferred is the path.

Step 3 : Locally injected routes : Routes injected on a router using network command o redistributed on a router are always preferred. They always have next hop value of 0.0.0.0 meaning locally injected on this router.

Step 4 : Shorter AS path Length: Routes who travel less number of autonomous systems are always preferred over other routes if the bgp decision process recahes step 4. This means that less AS numbers are listed in the list making the path more preferred over others. AS_SET and AS_CONFED_SEQ and AS_CONFED_SET are treated as single AS rather than a list of individual autonomous systems present in the braces.

Step 5 : Origin Path Attribute : The routes of certain origin are preferred over other routes. Like the routes with the origin code IGP (i) are preferred over EGP (e) routes which in turn are preferred over routes with incomplete(?) origin.

Step 6 : Multi Exit Discriminator : All the above path attributes are designed to influence outbound paths, but this path attribute can be set to influence inbound paths to a certain source , that is it can be used to tell the return packet which path to use when retuning from a destination back to source.

Step 7 : Neighbor Type : Prefer ebgp routes over ibgp routes.

Step 8 : IGP metrics : Route with lower IGP metric to a destination is preferred.

These steps are used to determine the single best path to reach the destination. Even if the router is configured to install multiple paths to a destination in the ip routing table only one path is taken over if the best path is determined by the above mentioned 9 steps. So, if the router is still not able to determine the single best path to the destination additional tie breaker steps are used and in this case more than one routes can be installed in the ip routing table depending on the configuration. Even if multiple paths are installed in the ip routing table bgp still prefers one single best path in its ip routing table. Additional tie breaking steps are :

Step 9 : Keep the oldest Ebgp route : If the routes compared are Ebgp routes then prefer the oldest ebgp routes. This helps in avoiding route flaps which can effect a whole lot autonomous systems and can be very bad as it can make the routing tables unstable which can take a long time to converge sometimes disrupting the traffic sent for the prefix in the mean time.

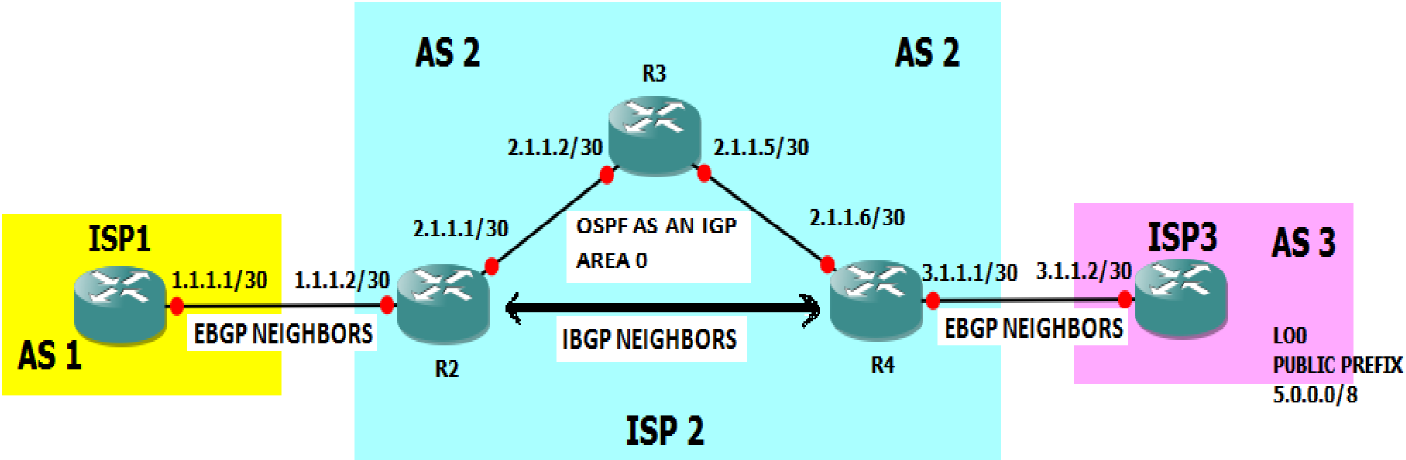
Step 10 : Smallest neighbor RID : This step requires additional configuration on the router and can be utilized only if the router is configured to do so. If configured the next hop address of the router with lowest router id is preferred.

Step 11 : Smallest neighbor RID : This step can come into picture when there are two links between two routers connected in such a way that there are two routes through these two links. Although this type of design should be avoided as it can lead to excessive updates by configuring the neighborship over loopback interfaces, but if configured the path with the lowest neighbor id that is the lowest ip address configured in the neighbor command would be preferred.

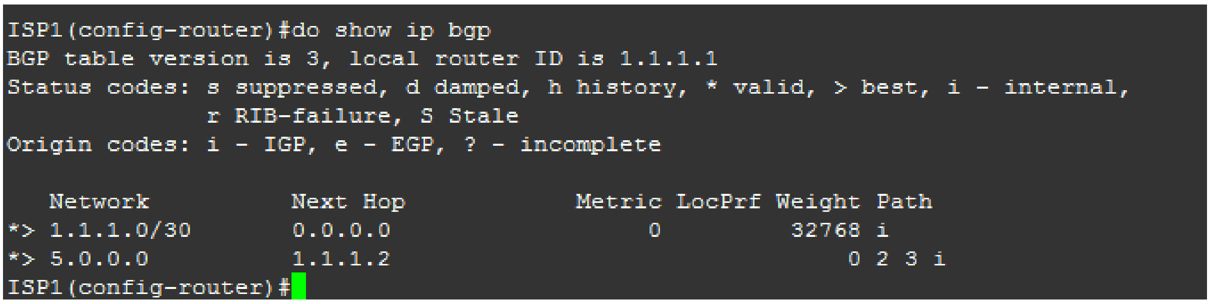
1.9 IBGP full mesh connectivity and alternate solutions to full mesh

IBGP default behaviour doesn't allow routes learnt from one ibgp neighbor to be propagated to the other ibgp neighbor. This is done in order to ensure that no routing loops occur within an AS for EBGP connections routing loops can be avoided using AS path attribute but since the AS path attribute won't change within an AS routing loops can occur. So IBGP connectivity requires the IBGP neighbors to be in full mesh connectivity or implement a method such that the connectivity appears to be full and routes are propogated to every ibgp neighbor. As an example we will discuss the problem of black hole routing in which due to the lack of full mesh connectivity or not enabling BGP within an AS on routers can lead to routing problems sending traffic to black hole.

Case 1 BGP not running on all routers within an AS

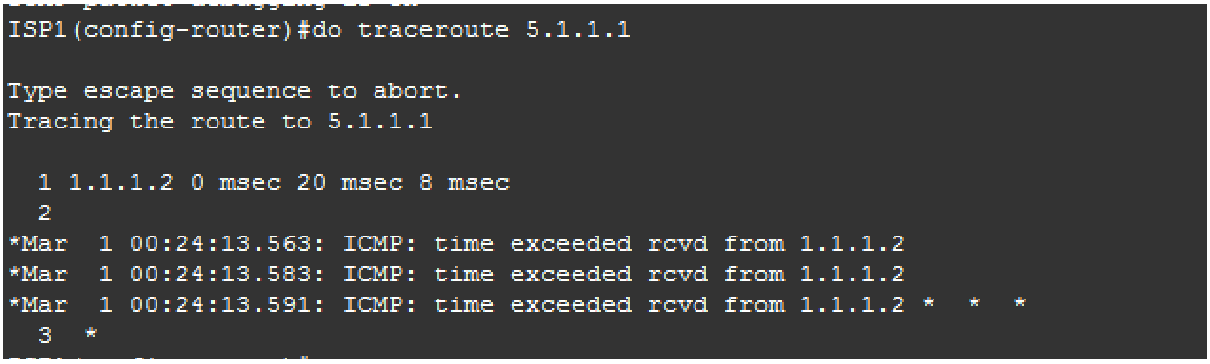


As we can see in the network diagram above this is the case when BGP is not enabled in all the routers in the AS 2. R2 and R4 are IBGP neighbors but ibgp is not enabled on router R3. ospf is enabled as an IGP in AS 2. When AS 3 sends an EBGP update for prefix 5.0.0.0/8 it is received by R4 in As 2 and r4 sends an ibgp update for the same prefix to R2 and R2 then sends the EBGP update to ISP1. Now the problem in this scenario is that router R3 is unaware of the route for the network 5.0.0.0/8. When R3 receives a packet destined for prefix 5.0.0.0/8 which would be sent by either r2 or r4 as r3 is a transit router, r3 looks into the routing table and as it notices no route for that prefix in the routing table it discards the packet. This is called black hole routing as the route exists and is valid, but due to lack of knowledge to the transit routers, the packets never reached the destination and are dropped in between.



```
ISP1(config-router)#do show ip bgp
BGP table version is 3, local router ID is 1.1.1.1
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
*> 1.1.1.0/30       0.0.0.0                  0         32768 i
*> 5.0.0.0          1.1.1.2                                0 2 3 i
ISP1(config-router)#
```

ISP 1 router has route for 5.0.0.0/8 prefix



```
ISP1(config-router)#do traceroute 5.1.1.1

Type escape sequence to abort.
Tracing the route to 5.1.1.1

  1 1.1.1.2 0 msec 20 msec 8 msec
  2
*Mar  1 00:24:13.563: ICMP: time exceeded rcvd from 1.1.1.2
*Mar  1 00:24:13.583: ICMP: time exceeded rcvd from 1.1.1.2
*Mar  1 00:24:13.591: ICMP: time exceeded rcvd from 1.1.1.2 *  *  *
  3  *
```

But the traceroute never reaches 5.1.1.1 as the packet is dropped cause r3 has no route for 5.0.0.0 prefix and icmp time exceeded message is sent

ISP 1 Configuration

```
ISP1#show run
!
hostname ISP1
!
interface FastEthernet0/0
 ip address 1.1.1.1 255.255.255.252
!
router bgp 1
 no synchronization
 bgp log-neighbor-changes
 network 1.1.1.0
 network 1.1.1.0 mask 255.255.255.252
 neighbor 1.1.1.2 remote-as 2
 no auto-summary
!
end
```

R2 Configuration

```
R2(config-router)#do show run

hostname R2
!

interface FastEthernet0/0
 ip address 1.1.1.2 255.255.255.252

!
interface FastEthernet0/1
 ip address 2.1.1.1 255.255.255.252
!
router ospf 1
 network 2.1.1.0 0.0.0.3 area 0
!
router bgp 2
 no synchronization
 neighbor 1.1.1.1 remote-as 1
 neighbor 2.1.1.6 remote-as 2
 neighbor 2.1.1.6 next-hop-self
 no auto-summary
!
end
```

R3 Configuration

```
R3(config-router)#do show run
hostname R3
!
interface FastEthernet0/0
 ip address 2.1.1.5 255.255.255.252
!
interface FastEthernet0/1
 ip address 2.1.1.2 255.255.255.252
!
router ospf 1
 log-adjacency-changes
 network 2.1.1.0 0.0.0.3 area 0
 network 2.1.1.4 0.0.0.3 area 0
!
end
```

R4 configuartion

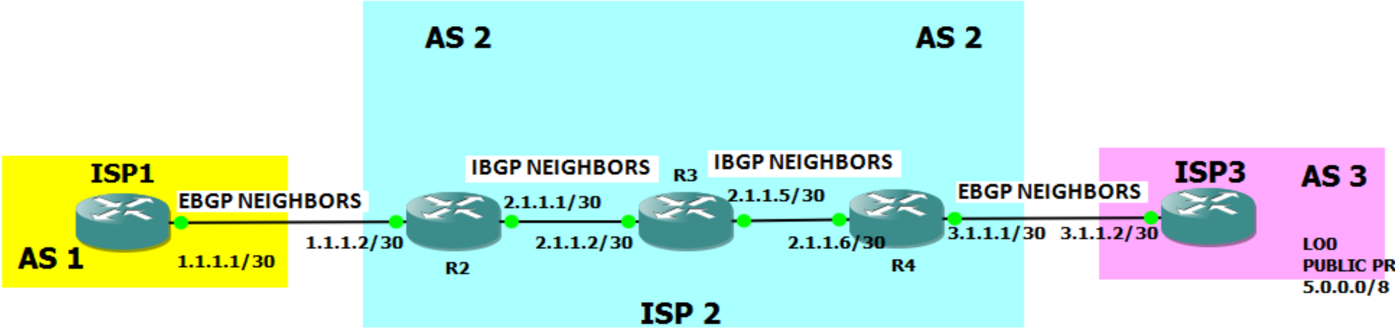
```
R4(config-router)#do show run
Building configuration...
hostname R4
!
interface FastEthernet0/0
 ip address 2.1.1.6 255.255.255.252
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 3.1.1.1 255.255.255.252
 duplex auto
 speed auto
!

!
router ospf 1
 log-adjacency-changes
 network 2.1.1.4 0.0.0.3 area 0
!
router bgp 2
 no synchronization
 bgp log-neighbor-changes
 neighbor 2.1.1.1 remote-as 2
 neighbor 2.1.1.1 next-hop-self
 neighbor 3.1.1.2 remote-as 3
 no auto-summary
!
```

```
end
```

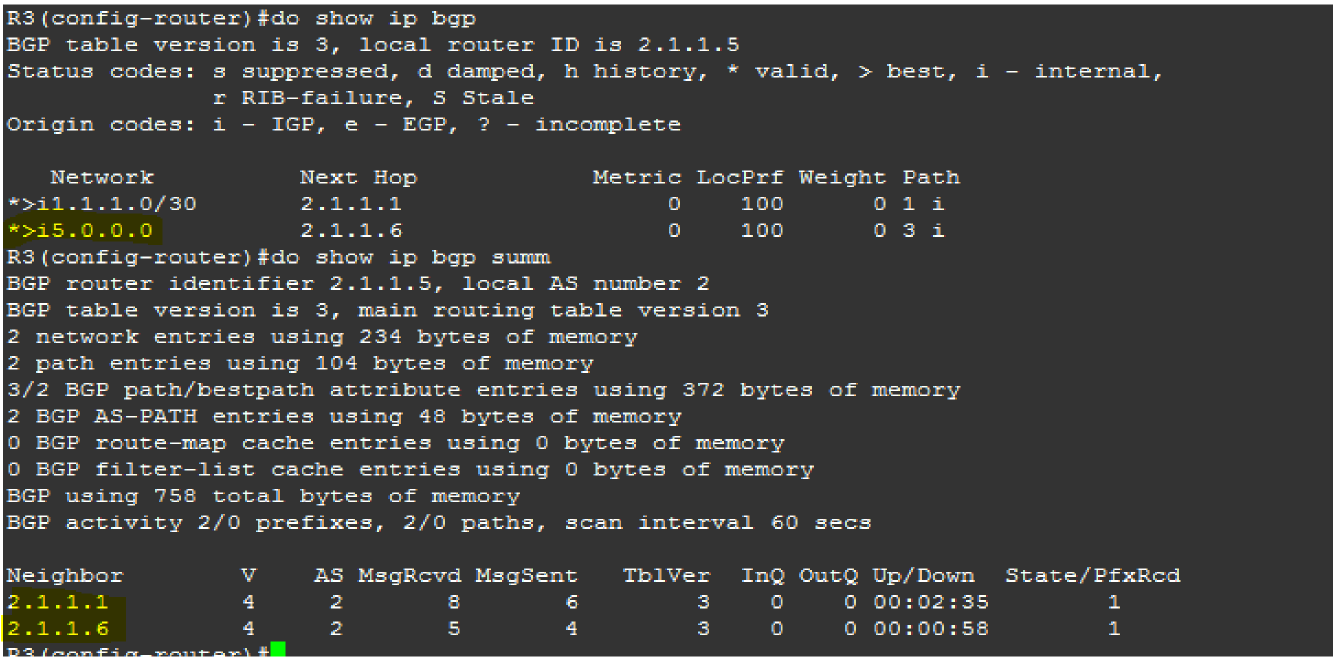
ISP 3 configuration

```
ISP3(config-router)#do show run
hostname ISP3
!
interface Loopback0
 ip address 5.1.1.1 255.0.0.0
!
interface FastEthernet0/1
 ip address 3.1.1.2 255.255.255.252
!
router bgp 3
 no synchronization
 bgp log-neighbor-changes
 network 5.0.0.0
 neighbor 3.1.1.1 remote-as 2
 no auto-summary
!
end
```

Case 2 Partial mesh IBGP connectivity

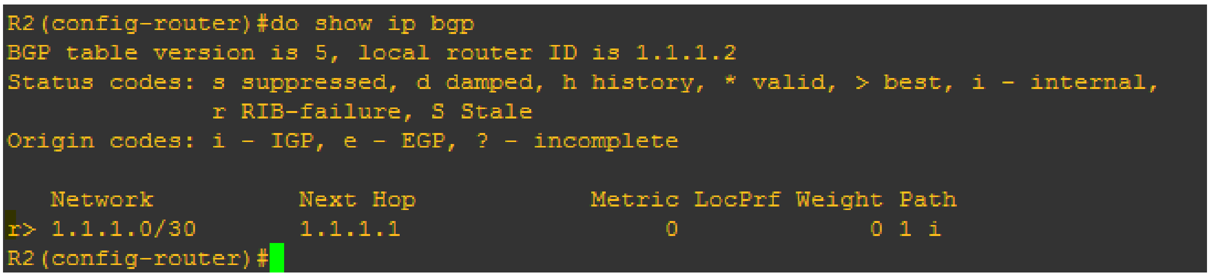


In this case R2 and R3 are IBGP neighbors and r3 and r4 are IBGP neighbors but there is no BGP relationship between R2 and R4. so when r4 learns a EBGP route from ISP 3 for prefix 5.0.0.0/8 it send an IBGP update to R3 but R3 never sends the update for 5.0.0.0/8 prefix to R2 because R3 thinks that R2 and R4 have direct connectivity with each other due to default consideration of full mesh BGP connectivity. So, R3 never sends the BGP update to R2 and hence ISP 1 never knows about 5.0.0.0 prefix.



```
R3(config-router)#do show ip bgp
BGP table version is 3, local router ID is 2.1.1.5
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
*>i1.1.1.0/30       2.1.1.1                  0    100      0 1 i
*>i5.0.0.0          2.1.1.6                  0    100      0 3 i
R3(config-router)#do show ip bgp summ
BGP router identifier 2.1.1.5, local AS number 2
BGP table version is 3, main routing table version 3
2 network entries using 234 bytes of memory
2 path entries using 104 bytes of memory
3/2 BGP path/bestpath attribute entries using 372 bytes of memory
2 BGP AS-PATH entries using 48 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 758 total bytes of memory
BGP activity 2/0 prefixes, 2/0 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.1.1.1         4     2       8       6        3    0    0 00:02:35        1
2.1.1.6         4     2       5       4        3    0    0 00:00:58        1
R3(config-router)#
```

As can be seen R3 received 5.0.0.0/8 route from R4 and it has IBGP relationship with both R2 and R4.



```
R2(config-router)#do show ip bgp
BGP table version is 5, local router ID is 1.1.1.2
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
r> 1.1.1.0/30       1.1.1.1                  0             0 1 i
R2(config-router)#
```

But R2 never receives any IBGP route advertised by R4 because R3 never advertised those routes to R2.
The only route visible here is the connected route which is shown as a rib failure in BGP table because it is a directly connected route so the BGP route is not installed in the routing table and the rib failure is indicated in the BGP table stating that this route was not injected into routing table due to BGP.

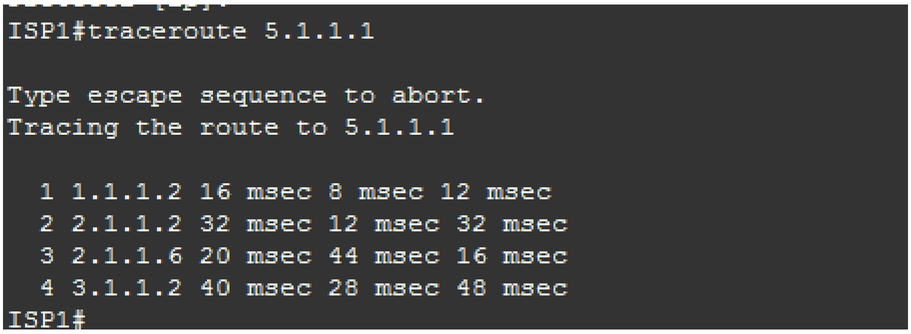
These problems of IBGP full mesh connectivity can be solved with the alternate solutions in which case IBGP can even run with partial mesh or even if it not enabled on all routers. The various methods are listed below :

1) BGP synchronization and redistributing routes: solves the problem in case 1
2) Route reflectors : solves case 2
3) BGP confederations : solves case 2

1.9.1 BGP synchronization

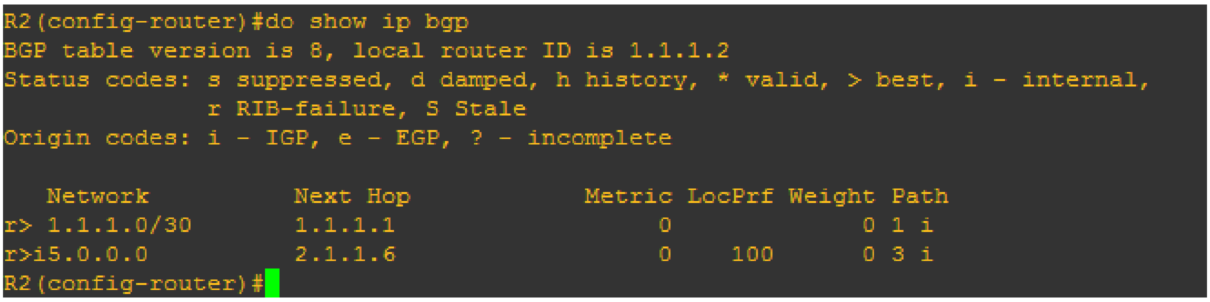
BGP synchronization solves the problem of advertising a black hole route to other AS and redistribution solves the problem of black hole routing. Although it is not a preferred method to redistribute the bgp routes into the igp as too many routes redistributed from bgp into igp can lead to the crash of igp routing processes as they are not designed to handle that much routes becuase they take too much memory storing routes which will ultimately result in overutilization of router resources resulting in a crash. Synchronization logic works like if the route learnt via BGP is not learnt via IGP and not present in the routing table do not install the route in BGP table. This solves the above problem as the route won't be present in the IGP because the BGP was never redistributed into IGP and so the route was never learnt via IGP into the routing table and due to synchronization it would never be installed in the BGP table.

Synchronization is just enabled with the command synchronization in BGP. Let's see the chnge in results with synchronization enabled.



```
ISP1#traceroute 5.1.1.1

Type escape sequence to abort.
Tracing the route to 5.1.1.1

  1 1.1.1.2 16 msec 8 msec 12 msec
  2 2.1.1.2 32 msec 12 msec 32 msec
  3 2.1.1.6 20 msec 44 msec 16 msec
  4 3.1.1.2 40 msec 28 msec 48 msec
ISP1#
```

As we can see that the end to end connectivity is up.

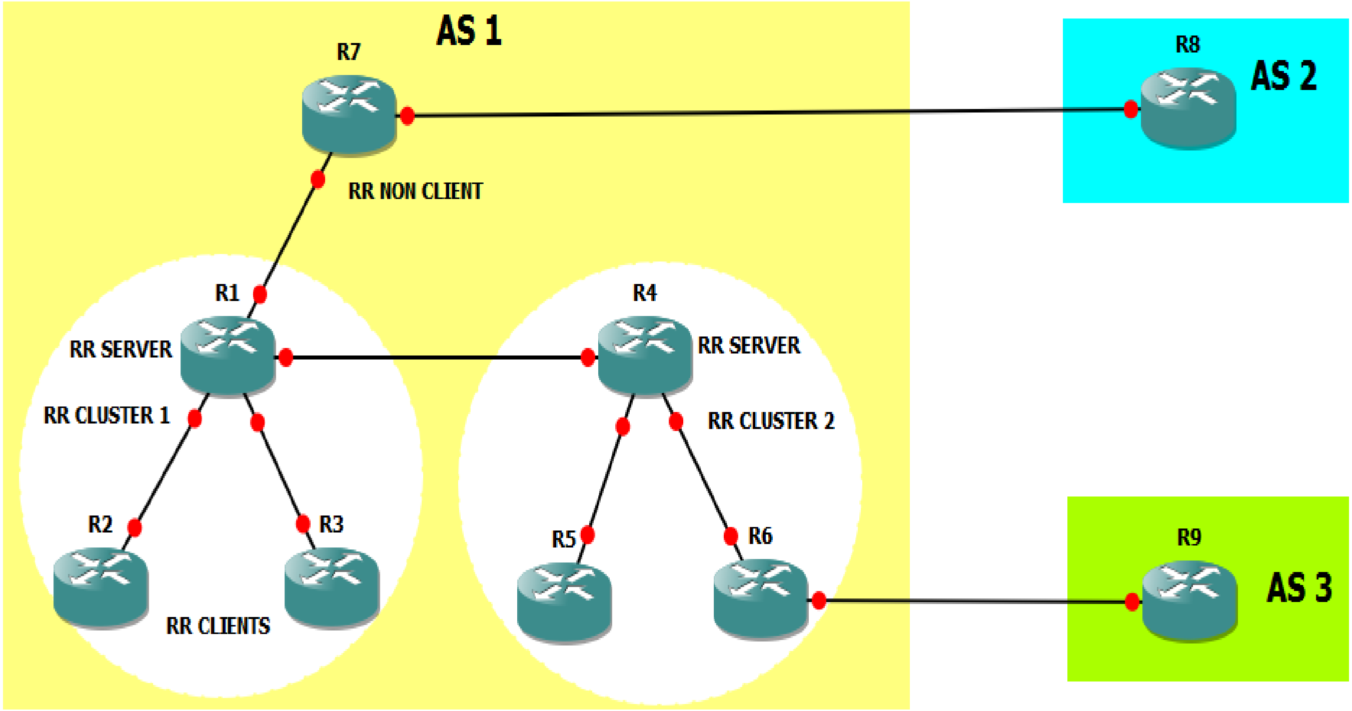


```
R2(config-router)#do show ip bgp
BGP table version is 8, local router ID is 1.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
r> 1.1.1.0/30       1.1.1.1                  0             0 1 i
r>i5.0.0.0          2.1.1.6                  0    100      0 3 i
R2(config-router)#
```

Routes on R2 in BGP table. since 5.0.0.0/8 states rib failure in BGP table it proves that route was learnt via IGP and is in the routing table.

The only change in configuration done for this part is that synchronization is enabled on router R2 and routes from BGP are redistributed into OSPF on router R4

<u>2) BGP route Reflectors</u>

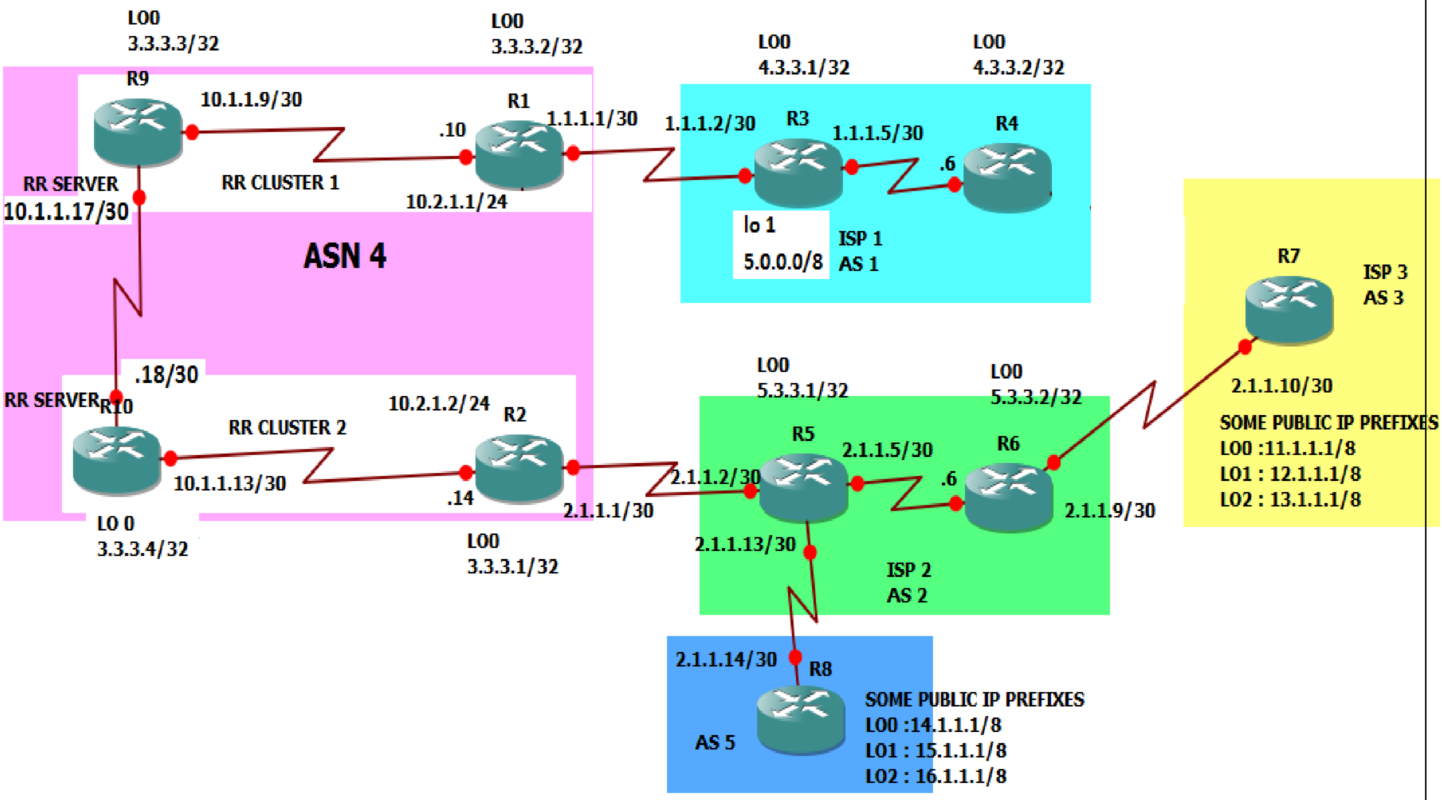


Route reflectors provide an alternative solution to full mesh connectivity in Ibgp and allow ibgp learned routes to other ibgp routers thus removing the need for full mesh connectivity and providing a loop free solution for Ibgp partial mesh connectivity. It involves the concept of route reflector Server, client and non client. Whenever a route is received by a client or non client it is sent to the RR server connected to them which in turn reflects those route to all the clients. A group of clients and a RR server is referred to as a cluster. Multiple RR clusters can exist in an AS. The condition for RR clusters is that every RR server should be fully meshed with RR servers of other clusters, else the topology database won't be uniform in the cluster. Route are reflected by clients to RR server and non client to RR servers and clients to clients but routes are not reflected between non clients. To avoid loops Route Reflectors uses various path attributes which are Cisco proprietary such as:

1)**Cluster_list:** Every RR cluster has a cluster ID associated with it which is used to avoid loops. If a cluster receives the route update listing its cluster id it discards that route because it knows that that route has already been advertised in that cluster.

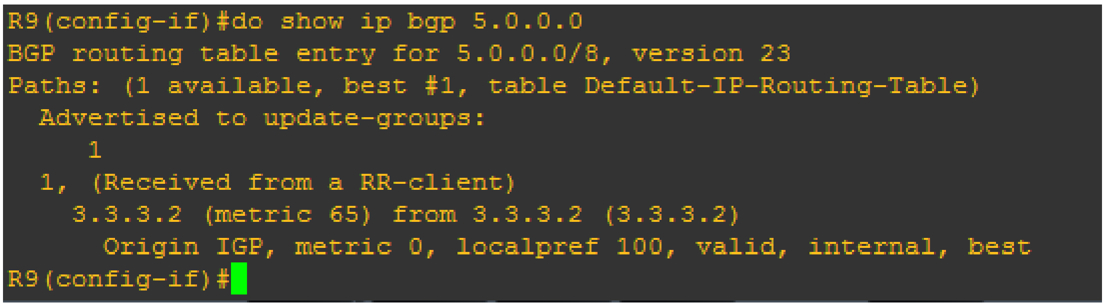
2)**Originator_ID :** This lists the RID of the first router that advertised the route into the AS. Hence if a router has its own RID listed as originator ID it does not use that route.

As an example we will take into consideration the below mentioned network diagram in which two RR clusters with one client each are configured. We will verify our configuration by checking whether 5.0.0.0/8 prefix injected AS 1 is reflected by RR cluster 1 to RR cluster 2 by router R9 which acts as RR server for cluster 1. Similarly we will check the output for 11.0.0.0/8,12.0.0.0/8,13.0.0.0/8 on router R9 whether these routes are reflected by R10 to R9.



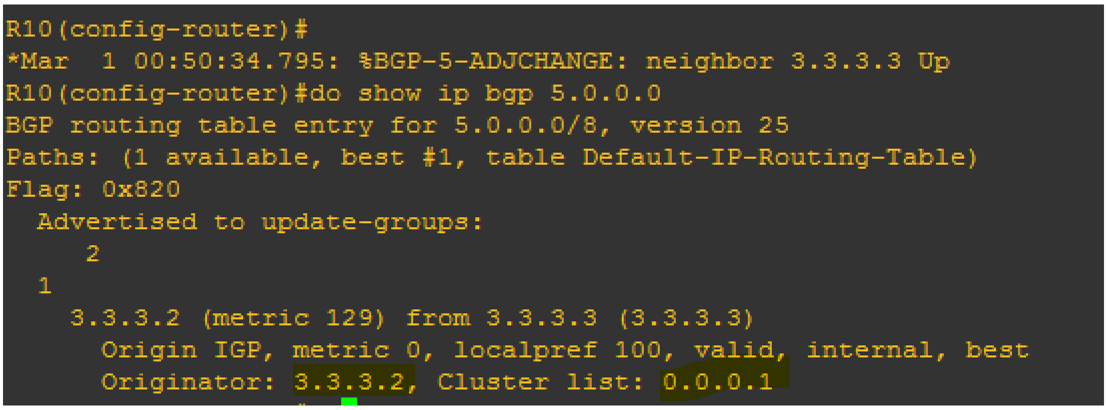
Network Diagram for this part

Show ip bgp 5.0.0.0/8 output on R9



```
R9(config-if)#do show ip bgp 5.0.0.0
BGP routing table entry for 5.0.0.0/8, version 23
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     1
  1, (Received from a RR-client)
    3.3.3.2 (metric 65) from 3.3.3.2 (3.3.3.2)
      Origin IGP, metric 0, localpref 100, valid, internal, best
R9(config-if)#
```

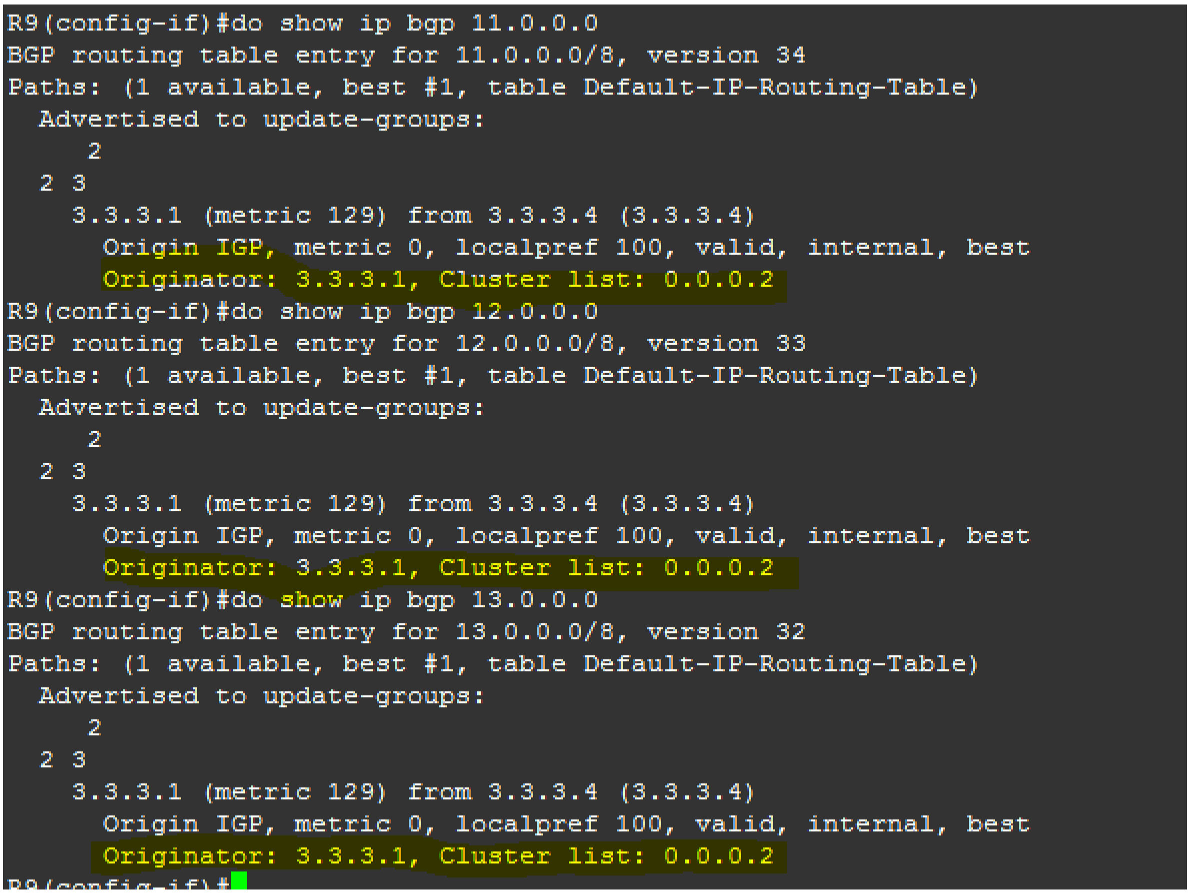
R10



```
R10(config-router)#
*Mar  1 00:50:34.795: %BGP-5-ADJCHANGE: neighbor 3.3.3.3 Up
R10(config-router)#do show ip bgp 5.0.0.0
BGP routing table entry for 5.0.0.0/8, version 25
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Flag: 0x820
  Advertised to update-groups:
     2
  1
    3.3.3.2 (metric 129) from 3.3.3.3 (3.3.3.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 3.3.3.2, Cluster list: 0.0.0.1
```

We can clearly see the reflected routes on R10 by R9 by checking the cluster id and the originator id.

For 11.0.0.0/8, 12.0.0.0/8 and 13.0.0.0/8 prefixes on R9



```
R9(config-if)#do show ip bgp 11.0.0.0
BGP routing table entry for 11.0.0.0/8, version 34
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     2
  2 3
    3.3.3.1 (metric 129) from 3.3.3.4 (3.3.3.4)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 3.3.3.1, Cluster list: 0.0.0.2
R9(config-if)#do show ip bgp 12.0.0.0
BGP routing table entry for 12.0.0.0/8, version 33
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     2
  2 3
    3.3.3.1 (metric 129) from 3.3.3.4 (3.3.3.4)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 3.3.3.1, Cluster list: 0.0.0.2
R9(config-if)#do show ip bgp 13.0.0.0
BGP routing table entry for 13.0.0.0/8, version 32
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
     2
  2 3
    3.3.3.1 (metric 129) from 3.3.3.4 (3.3.3.4)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 3.3.3.1, Cluster list: 0.0.0.2
R9(config-if)#
```

We can clearly see that R9 has the routes received from cluster 2 by seeing the originator id and cluster id.
Hence with the use RR full mesh connectivity is no more required.

R9 and R10 configuration is shared below. On other routers no additional configuration is required.

R9

hostname R9

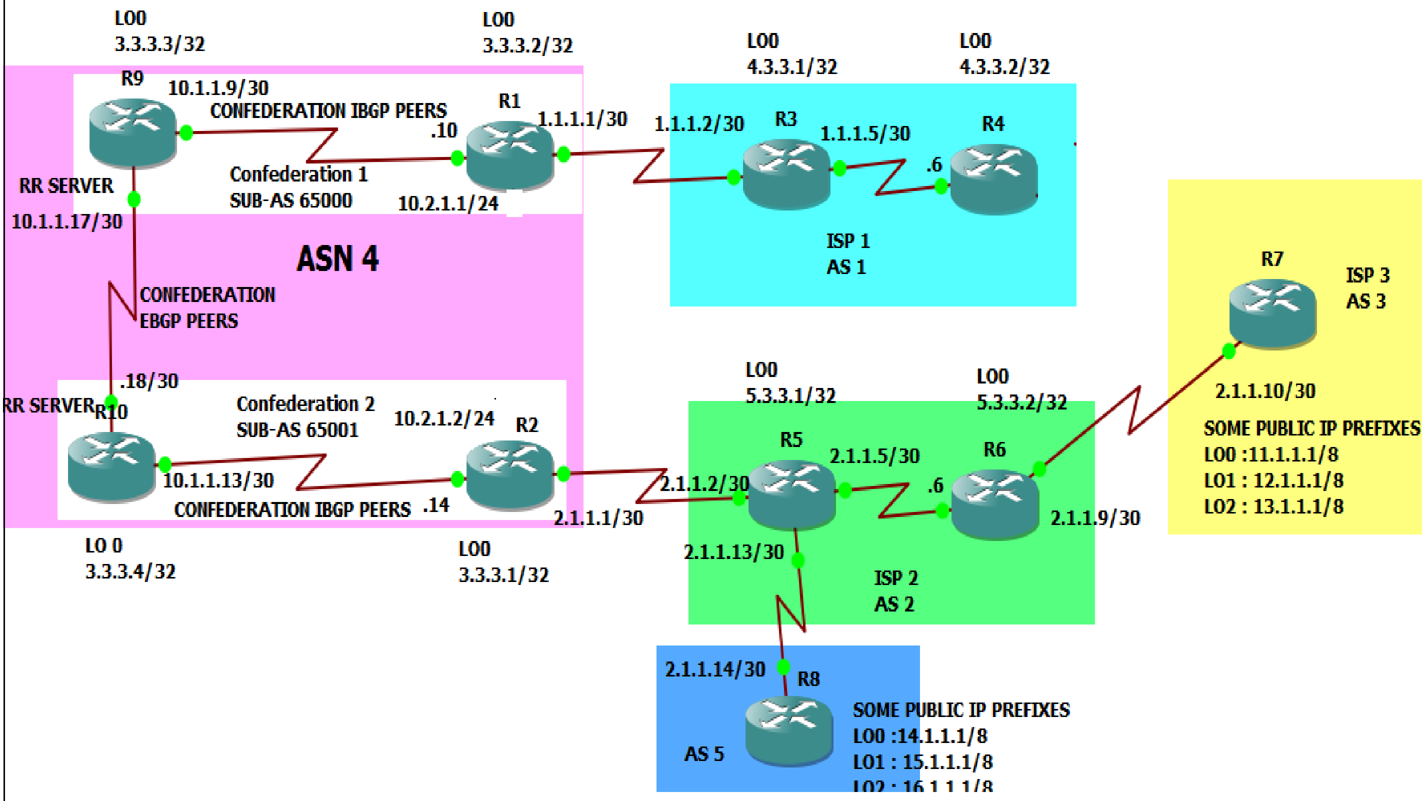
```
!
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
!
interface Serial1/0
 ip address 10.1.1.9 255.255.255.252
!
router ospf 1
 log-adjacency-changes
 network 3.3.3.3 0.0.0.0 area 0
 network 10.1.1.8 0.0.0.3 area 0
 network 10.1.1.16 0.0.0.3 area 0
!
router bgp 4
 no synchronization
 bgp cluster-id 1
 bgp log-neighbor-changes
 neighbor 3.3.3.2 remote-as 4
 neighbor 3.3.3.2 update-source Loopback0
 neighbor 3.3.3.2 route-reflector-client
 neighbor 3.3.3.2 next-hop-self
 neighbor 3.3.3.4 remote-as 4
 neighbor 3.3.3.4 update-source Loopback0
 neighbor 3.3.3.4 next-hop-self
 no auto-summary
```

R10

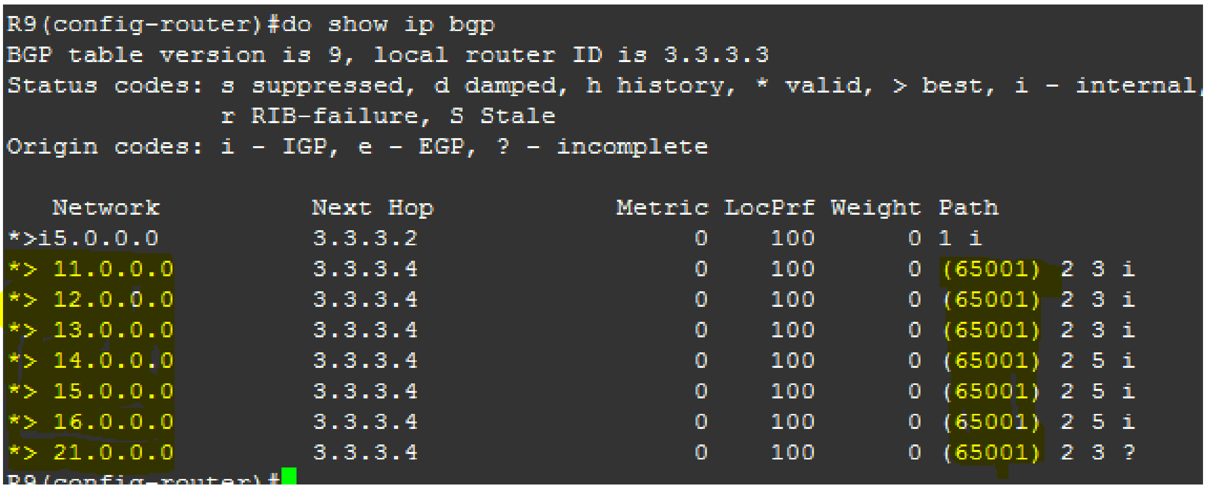
```
hostname R10
!
interface Loopback0
 ip address 3.3.3.4 255.255.255.255
!
interface Serial1/2
 ip address 10.1.1.18 255.255.255.252
!
interface Serial1/3
 ip address 10.1.1.13 255.255.255.252

router ospf 1
 log-adjacency-changes
 network 3.3.3.4 0.0.0.0 area 0
 network 10.1.1.12 0.0.0.3 area 0
 network 10.1.1.16 0.0.0.3 area 0
!
router bgp 4
 no synchronization
 bgp cluster-id 2
 bgp log-neighbor-changes
 neighbor 3.3.3.1 remote-as 4
 neighbor 3.3.3.1 update-source Loopback0
 neighbor 3.3.3.1 route-reflector-client
 neighbor 3.3.3.1 next-hop-self
 neighbor 3.3.3.3 remote-as 4
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 3.3.3.3 next-hop-self
 no auto-summary
!
end
```

3) BGP Confederations

It involves the process of sub-autonomous system inside the main autonomous system. Peers inside the same sub autonomous system are considered to be ibgp peers while peers inside separate sub autonomous system are considered to be ebgp peers. This way full mesh of connectivity is avoided as ebgp peers do not require full mesh of connectivity. Although within a sub-AS full mesh of connectivity is required. AS_CONFED_SEQ is used as an as path attribute inside a confederation and the sub autonomous systems are added in sequence and are placed in braces. For aggregated routes AS_CONFEF_SET path attribute is used.

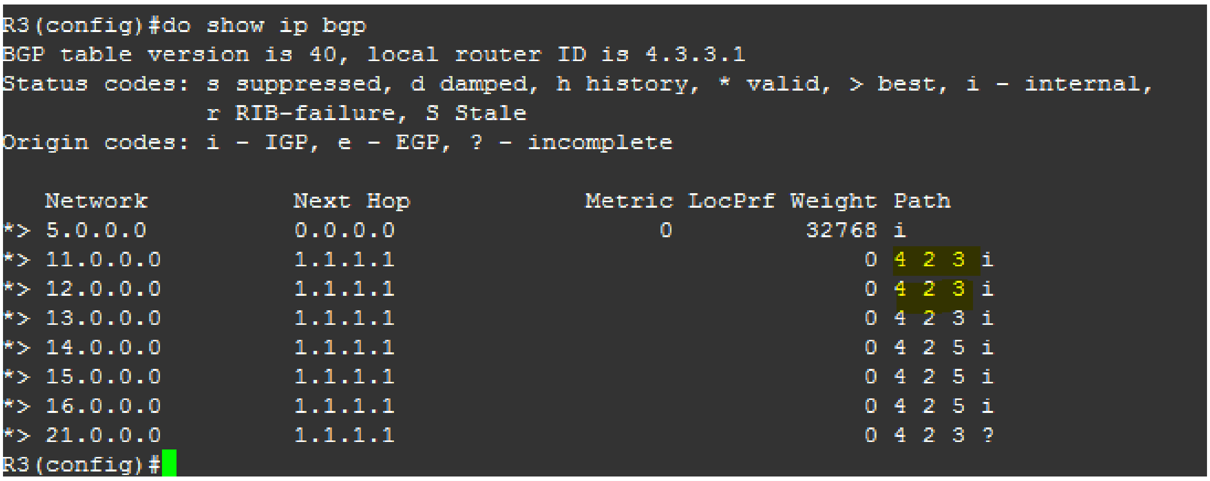


As an example we will use the above network to demonstrate the concept of BGP confederation. As can be seen the AS4 is divided u=into two confederations 65000 and 65001. private ASNS are used within an AS as these ASNS won't be advertised outside the AS. R9 and R10 act as confederation ebgp peers. As an example when 11.0.0.0/8,12.0.0.0/8 and 13.0.0.0/8 routes are advertised by R5 to R2, R2 follows the igbp rules within a confederation and send the update to r10 with ibgp rules. R10 router send the same update to R9 following ebgp rules except that next hop for the neighbor remains unchanged to the R2 interface.



```
R9(config-router)#do show ip bgp
BGP table version is 9, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i5.0.0.0          3.3.3.2                  0    100      0 1 i
*> 11.0.0.0         3.3.3.4                  0    100      0 (65001) 2 3 i
*> 12.0.0.0         3.3.3.4                  0    100      0 (65001) 2 3 i
*> 13.0.0.0         3.3.3.4                  0    100      0 (65001) 2 3 i
*> 14.0.0.0         3.3.3.4                  0    100      0 (65001) 2 5 i
*> 15.0.0.0         3.3.3.4                  0    100      0 (65001) 2 5 i
*> 16.0.0.0         3.3.3.4                  0    100      0 (65001) 2 5 i
*> 21.0.0.0         3.3.3.4                  0    100      0 (65001) 2 3 ?
R9(config-router)#
```

As can be seen in the output of show ip BGP on r9 router the routes learnt from r10 as highlighted are clearly listing a confederation AS of 65001 in braces along with other ASes. I have configured next-hop self command on the neighbors due to which we can see r 10 as next hop on r9 router. Now lets check the output on R3 router for ip bgp table.



```
R3(config)#do show ip bgp
BGP table version is 40, local router ID is 4.3.3.1
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
*> 5.0.0.0          0.0.0.0                  0         32768 i
*> 11.0.0.0         1.1.1.1                                0 4 2 3 i
*> 12.0.0.0         1.1.1.1                                0 4 2 3 i
*> 13.0.0.0         1.1.1.1                                0 4 2 3 i
*> 14.0.0.0         1.1.1.1                                0 4 2 5 i
*> 15.0.0.0         1.1.1.1                                0 4 2 5 i
*> 16.0.0.0         1.1.1.1                                0 4 2 5 i
*> 21.0.0.0         1.1.1.1                                0 4 2 3 ?
R3(config)#
```

As we can clearly see Sub autonomous system is not propagated outside the AS and the only AS listed is AS 4.

Router configuration

R1

```
hostname R1
!
interface Loopback0
 ip address 3.3.3.2 255.255.255.255
interface Serial1/0
 ip address 10.1.1.10 255.255.255.252
```

```
!
router ospf 1
 log-adjacency-changes
 network 3.3.3.2 0.0.0.0 area 0
 network 10.1.1.8 0.0.0.3 area 0
 network 10.2.1.0 0.0.0.255 area 0
!
router bgp 65000
 bgp confederation identifier 4
 neighbor 1.1.1.2 remote-as 1
 neighbor 1.1.1.2 update-source Serial1/2
 neighbor 3.3.3.3 remote-as 65000
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 3.3.3.3 next-hop-self
```

R2

```
hostname R2
!
interface Loopback0
 ip address 3.3.3.1 255.255.255.255
!
interface Serial1/1
 ip address 10.2.1.2 255.255.255.0
!
interface Serial1/2
 ip address 2.1.1.1 255.255.255.252
interface Serial1/3
 ip address 10.1.1.14 255.255.255.252
!
router ospf 1
 log-adjacency-changes
 network 3.3.3.1 0.0.0.0 area 0
 network 10.1.1.12 0.0.0.3 area 0
 network 10.2.1.0 0.0.0.255 area 0
!
router bgp 65001
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 4
 neighbor 2.1.1.2 remote-as 2
 neighbor 3.3.3.4 remote-as 65001
 neighbor 3.3.3.4 update-source Loopback0
 neighbor 3.3.3.4 next-hop-self
```

R9

```
hostname R9
!
interface Loopback0
 ip address 3.3.3.3 255.255.255.255
!
```

```
interface Serial1/0
 ip address 10.1.1.9 255.255.255.252
!
interface Serial1/2
 ip address 10.1.1.17 255.255.255.252

!
router ospf 1
 log-adjacency-changes
 network 3.3.3.3 0.0.0.0 area 0
 network 10.1.1.8 0.0.0.3 area 0
 network 10.1.1.16 0.0.0.3 area 0
!
router bgp 65000
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 4
 bgp confederation peers 65001
 neighbor 3.3.3.2 remote-as 65000
 neighbor 3.3.3.2 update-source Loopback0
 neighbor 3.3.3.2 next-hop-self
 neighbor 3.3.3.4 remote-as 65001
 neighbor 3.3.3.4 ebgp-multihop 2
 neighbor 3.3.3.4 update-source Loopback0
```

R10

```
hostname R10
!
interface Loopback0
 ip address 3.3.3.4 255.255.255.255
!
interface Serial1/2
 ip address 10.1.1.18 255.255.255.252
 serial restart-delay 0
!
interface Serial1/3
 ip address 10.1.1.13 255.255.255.252
!
router ospf 1
 log-adjacency-changes
 network 3.3.3.4 0.0.0.0 area 0
 network 10.1.1.12 0.0.0.3 area 0
 network 10.1.1.16 0.0.0.3 area 0
!
router bgp 65001
 no synchronization
 bgp log-neighbor-changes
 bgp confederation identifier 4
 bgp confederation peers 65000
 neighbor 3.3.3.1 remote-as 65001
 neighbor 3.3.3.1 update-source Loopback0
 neighbor 3.3.3.1 next-hop-self
 neighbor 3.3.3.3 remote-as 65000
 neighbor 3.3.3.3 ebgp-multihop 2
 neighbor 3.3.3.3 update-source Loopback0
 neighbor 3.3.3.3 next-hop-self
```

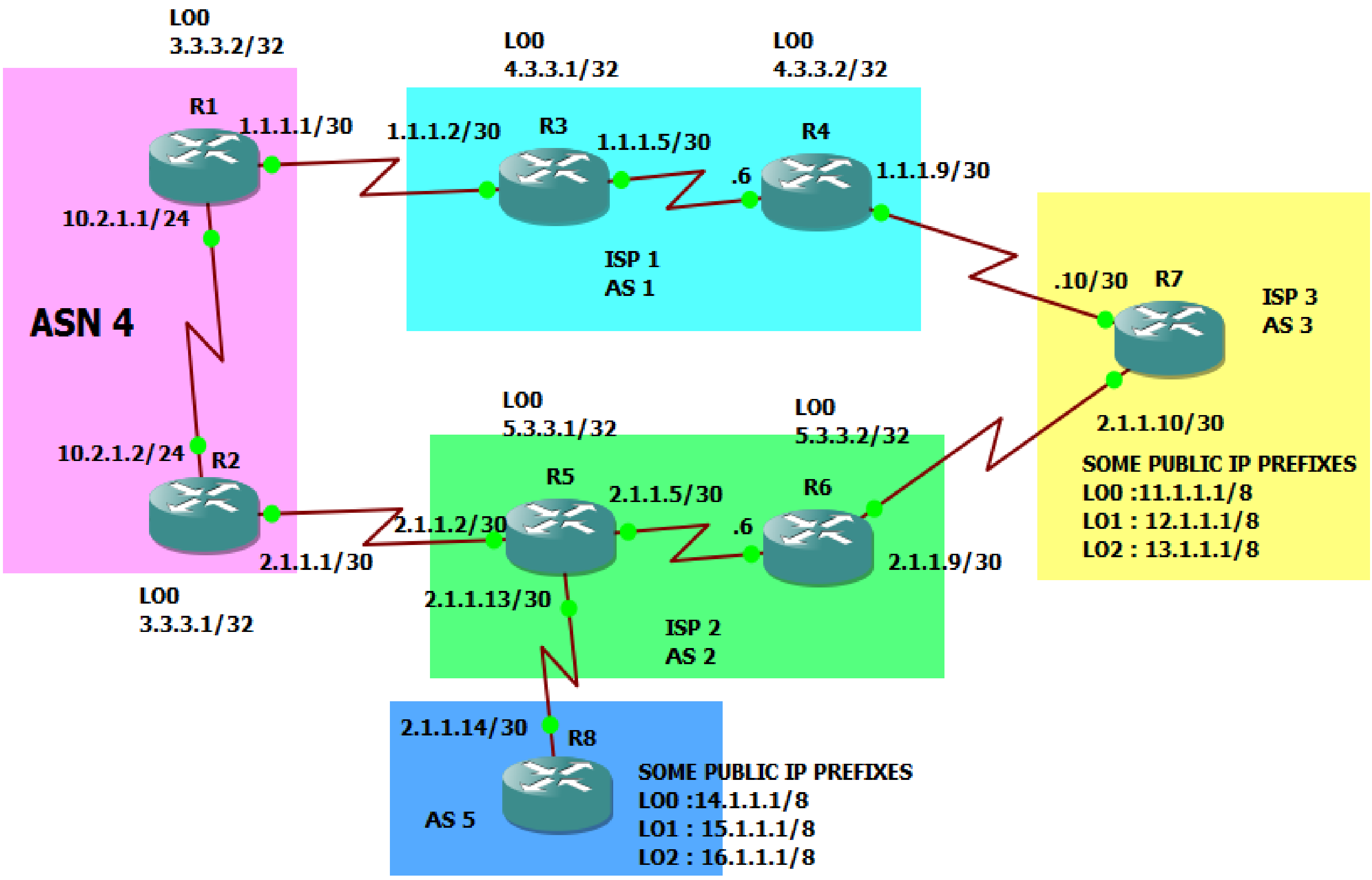
As we can see in the configuration there is a new concept of ebgp multihop used above. This is used in case of ebgp neighborships over loopback interfaces because by default the router sets the ttl value of 1 when sending requests for ebgp neighborships. Although it is one hop away but when the receiving interface receives the packet and tries handing over it to loopback interface it decrements the ttl value by 1 resulting in dropping of the packet. So by setting the value of ebgp multihop to 2 the packet reaches the loopback interface.

1.10 Backdoor routes

Sometimes a leased line connection can exist between two sites along with the ebgp connection between them through the internet or some other media. In this case by default the route used should go through the leased line but because the administrative distance of ebgp is 20 it is always the least administrative distance among other igp and is always the preferred route. In this case for the routes which need to be reachable through leased line a backdoor command should be configured in bgp for that network which sets the ebgp administrative distance value equal to ibgp value making IGP routes better than bgp routes , thus helping in utilizing the leased line connection. This concept is called the concept of backdoor routes in BGP.

1.11 Filtering routes in BGP

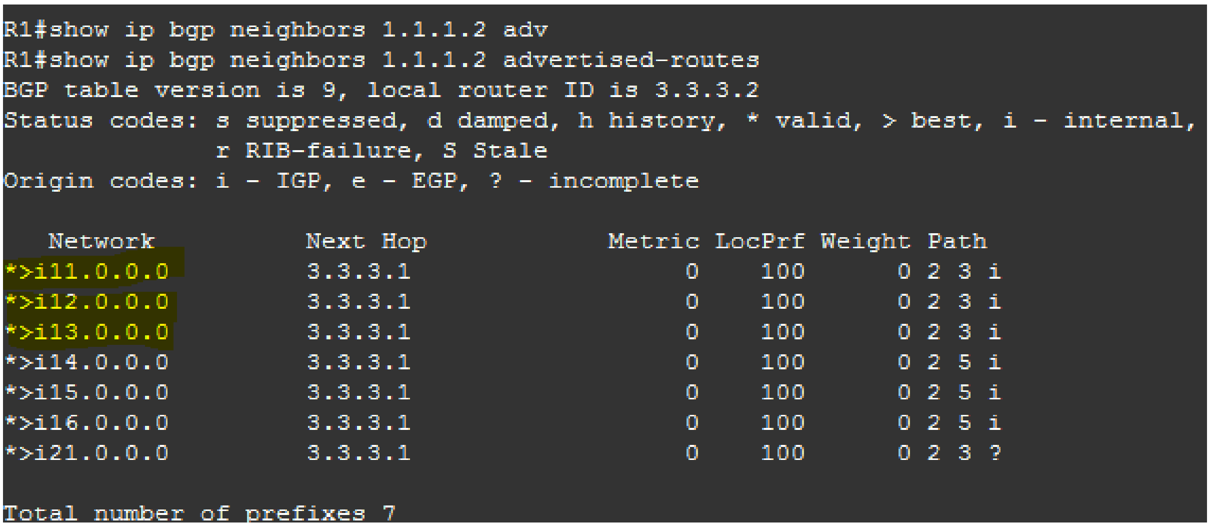
BGP routes require filtering at some point in time so that only interested routes are being propagated in an AS. Like for an AS it is sometimes important to filter routes received from one AS to be filtered from going into another AS, thus becoming a transit AS and consuming the bandwidth of that AS. There are many methods to filter BGP routes. Each of them is discussed below:



1.11.1 Distribute lists

These lists can be applied in both inbound and outbound directions and can use standard and extended access-lists for filtering. As an example we will filter the subnets 11.0.0.0, 12.0.0.0 and 13.0.0.0 from reaching AS1 through AS4. So we will apply outbound filter for these networks on R1 interface towards AS1. As an assumption for this part the link between R4 and R7 is shutdown i.e. there is no direct connection between AS1 and AS3.

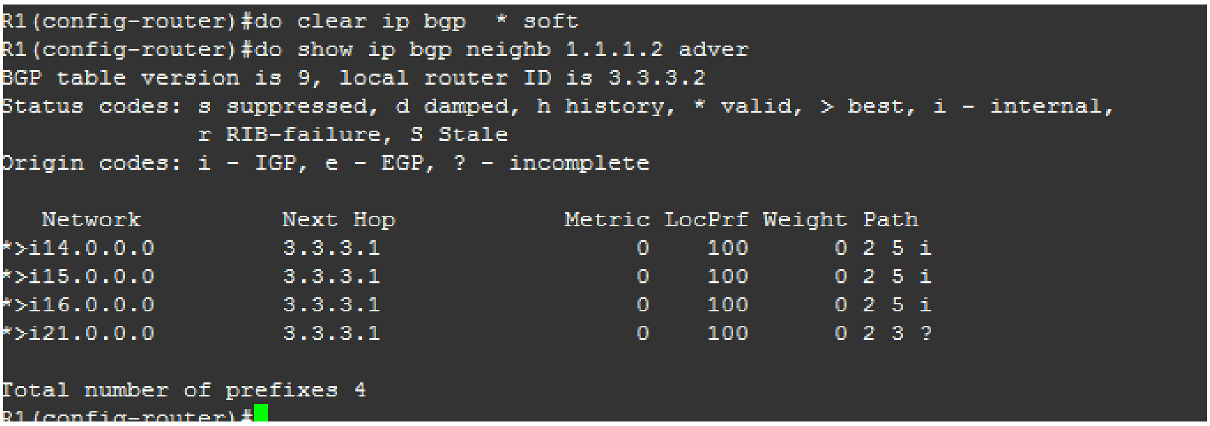
1) For this we will configure an access-list matching these routes with a deny statement and permitting all other routes.
2) Then we will apply a distribute list in the outbound direction referring to this access-list.



```
R1#show ip bgp neighbors 1.1.1.2 adv
R1#show ip bgp neighbors 1.1.1.2 advertised-routes
BGP table version is 9, local router ID is 3.3.3.2
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
*>i11.0.0.0         3.3.3.1                  0    100      0 2 3 i
*>i12.0.0.0         3.3.3.1                  0    100      0 2 3 i
*>i13.0.0.0         3.3.3.1                  0    100      0 2 3 i
*>i14.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i15.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i16.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i21.0.0.0         3.3.3.1                  0    100      0 2 3 ?

Total number of prefixes 7
```

As we can see routes 11.0.0.0, 12.0.0.0 and 13.0.0.0 are advertised to the neighbor 1.1.1.2(R3) in AS 1.

Next we will apply the distribute list and see the results.



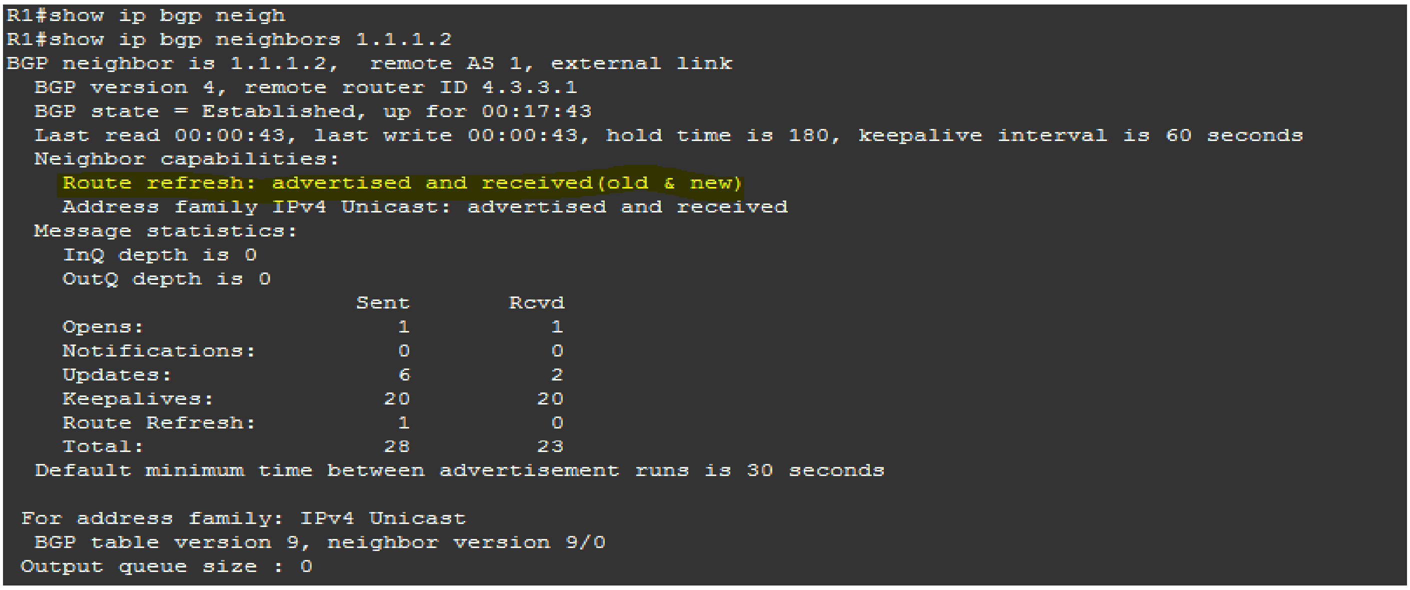
```
R1(config-router)#do clear ip bgp  * soft
R1(config-router)#do show ip bgp neighb 1.1.1.2 adver
BGP table version is 9, local router ID is 3.3.3.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i14.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i15.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i16.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i21.0.0.0         3.3.3.1                  0    100      0 2 3 ?

Total number of prefixes 4
R1(config-router)#
```

As we see the above mentioned routes are no longer advertised to the neighbor thereby filtering those routes.
We have used clear ip bgp * soft to do a soft reset of sessions between the bgp neighbors. If we don't do this the outbound policy is not applied and all the routes are sent as if no outbound filter existed. By using this command, outbound filter if any is applied and the update is sent accordingly. if the policy is applied in inbound direction then either update soft configuration inbound feature can be used which allows the router to store the update received from the neighboring router and later reprocess the update by passing through inbound filter or the router is enabled with the route refresh capability which allows the router to request the neighboring router to resend the update.

R1 config for filtering

```
access-list 1 deny   11.0.0.0 0.255.255.255
access-list 1 deny   12.0.0.0 0.255.255.255
access-list 1 deny   13.0.0.0 0.255.255.255
access-list 1 permit any
router bgp 4
 neighbor 1.1.1.2 remote-as 1
 neighbor 1.1.1.2 update-source Serial1/2
```

neighbor 1.1.1.2 distribute-list 1 out



```
R1#show ip bgp neigh
R1#show ip bgp neighbors 1.1.1.2
BGP neighbor is 1.1.1.2,  remote AS 1, external link
  BGP version 4, remote router ID 4.3.3.1
  BGP state = Established, up for 00:17:43
  Last read 00:00:43, last write 00:00:43, 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:                  1          1
    Notifications:          0          0
    Updates:                6          2
    Keepalives:            20         20
    Route Refresh:          1          0
    Total:                 28         23
  Default minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 9, neighbor version 9/0
 Output queue size : 0
```

Output indicating the router has route refresh capability

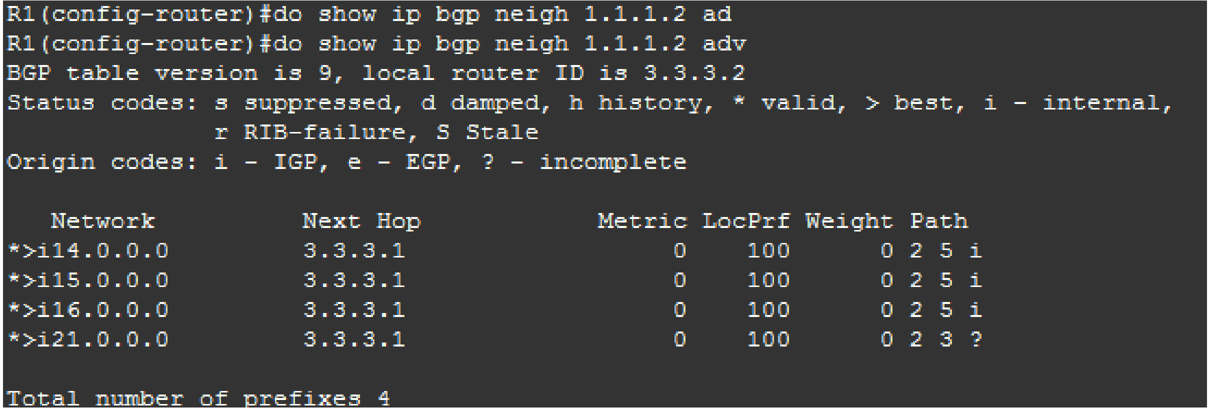
1.11.2 Filtering using Prefix list

Filtering involving prefix list is same as the filtering using distribute-list except that prefix lists match the prefixes to be filtered using prefix-lists and then the policy is applied in outbound or inbound direction. We will filter the same above mentioned prefixes using prefix-list this time.

R1 configuration

ip prefix-list 1 seq 5 deny 11.0.0.0/8
ip prefix-list 1 seq 10 deny 12.0.0.0/8
ip prefix-list 1 seq 15 deny 13.0.0.0/8
ip prefix-list 1 seq 20 permit 0.0.0.0/0 le 32
router bgp 4
neighbor 1.1.1.2 remote-as 1
 neighbor 1.1.1.2 update-source Serial1/2
 neighbor 1.1.1.2 prefix-list 1 out

As we can see the configuration is very much same except that prefix list is applied instead of access-list for matching prefixes.



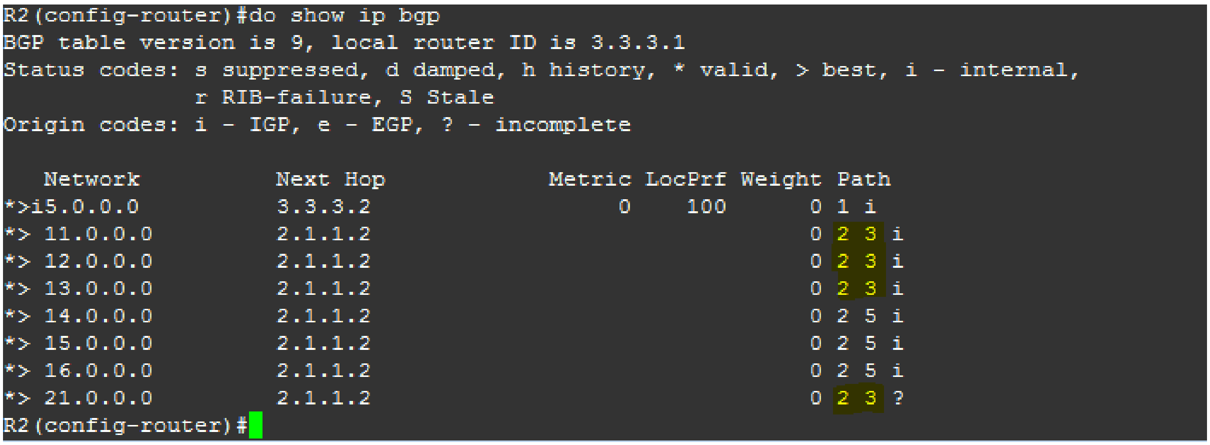
```
R1(config-router)#do show ip bgp neigh 1.1.1.2 ad
R1(config-router)#do show ip bgp neigh 1.1.1.2 adv
BGP table version is 9, local router ID is 3.3.3.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i14.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i15.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i16.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i21.0.0.0         3.3.3.1                  0    100      0 2 3 ?

Total number of prefixes 4
```

As we can see the filtered routes are not advertised anymore.

1.11.3 IP AS_path filter lists

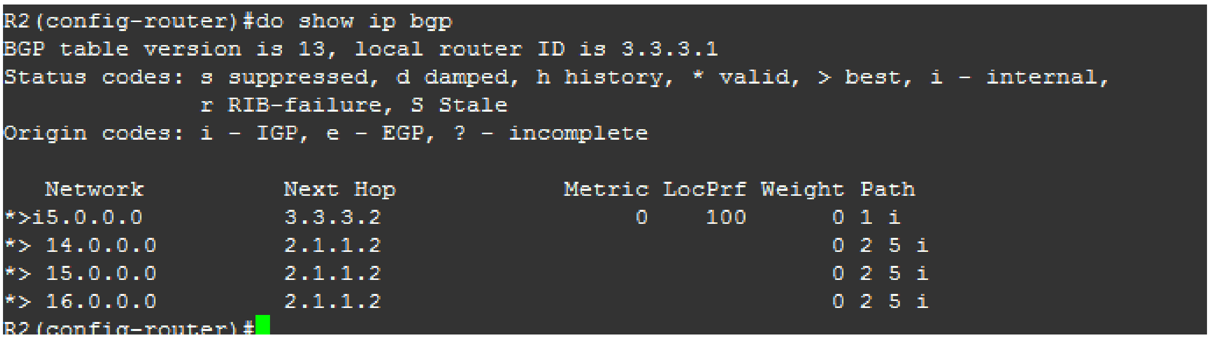
These filter lists can be used to match the routes based on the AS path sequence. The matched AS can be filtered out and the other As numbers can be permitted or vice versa. Reg expressions can be used to match the AS paths and differing logics can be applied to filter the routes starting or ending with a particular AS or matching some other AS logic. AS an example taking the above network diagram into consideration we will filter all the routes coming from AS3 on router r2 in inbound direction. For that we fill first create an AS path access-list with a deny statement to deny the routes coming from AS 3 and permitting all other routes and then applying the as path access-list on R2 in inbound direction.

Before applying filtering let's check the BGP routes on router R2.



```
R2(config-router)#do show ip bgp
BGP table version is 9, local router ID is 3.3.3.1
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
*>i5.0.0.0          3.3.3.2                  0    100      0 1 i
*> 11.0.0.0         2.1.1.2                                0 2 3 i
*> 12.0.0.0         2.1.1.2                                0 2 3 i
*> 13.0.0.0         2.1.1.2                                0 2 3 i
*> 14.0.0.0         2.1.1.2                                0 2 5 i
*> 15.0.0.0         2.1.1.2                                0 2 5 i
*> 16.0.0.0         2.1.1.2                                0 2 5 i
*> 21.0.0.0         2.1.1.2                                0 2 3 ?
R2(config-router)#
```

As we can see the routes originating in AS 3 have their first as number listed as 3 on extreme right in the path column in the path list. Now let's apply the filter and see the results.



```
R2(config-router)#do show ip bgp
BGP table version is 13, local router ID is 3.3.3.1
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
*>i5.0.0.0          3.3.3.2                  0    100      0 1 i
*> 14.0.0.0         2.1.1.2                                0 2 5 i
*> 15.0.0.0         2.1.1.2                                0 2 5 i
*> 16.0.0.0         2.1.1.2                                0 2 5 i
R2(config-router)#
```

As we can see after applying the filter all routes matching AS 3 as their origin are being filtered out.

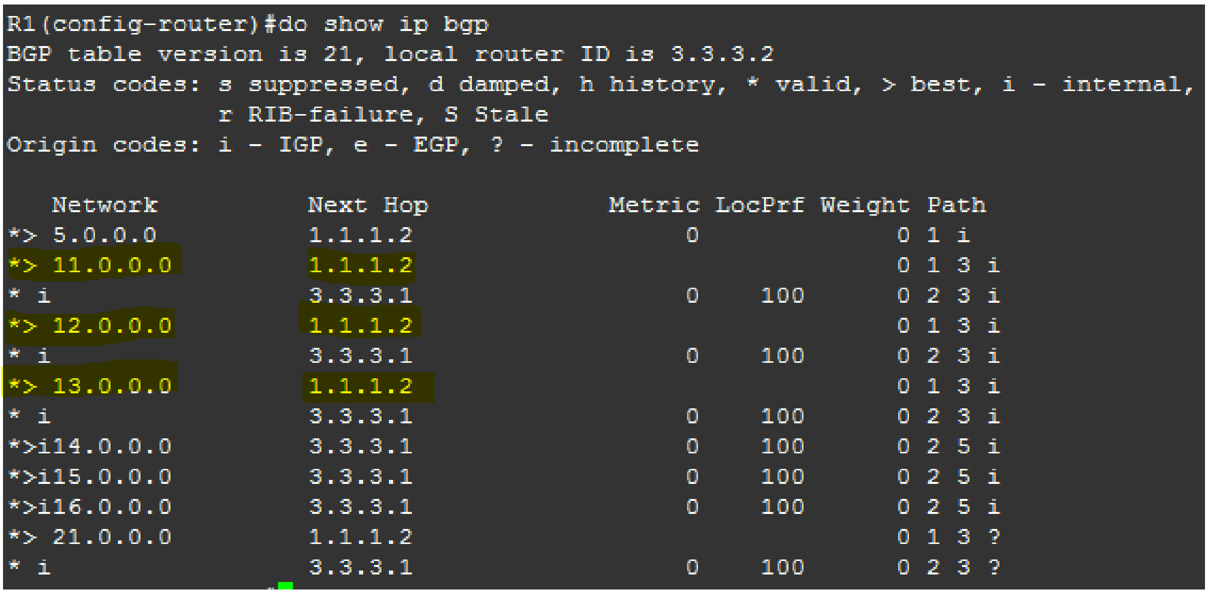
<u>R2 configuration</u>

ip as-path access-list 1 deny _3$ ($ is a metacharacter indicating end of an expression)
ip as-path access-list 1 permit .* (wildcard to match all as paths)

```
router bgp 4
 bgp log-neighbor-changes
 neighbor 2.1.1.2 remote-as 2
 neighbor 2.1.1.2 filter-list 1 in
```

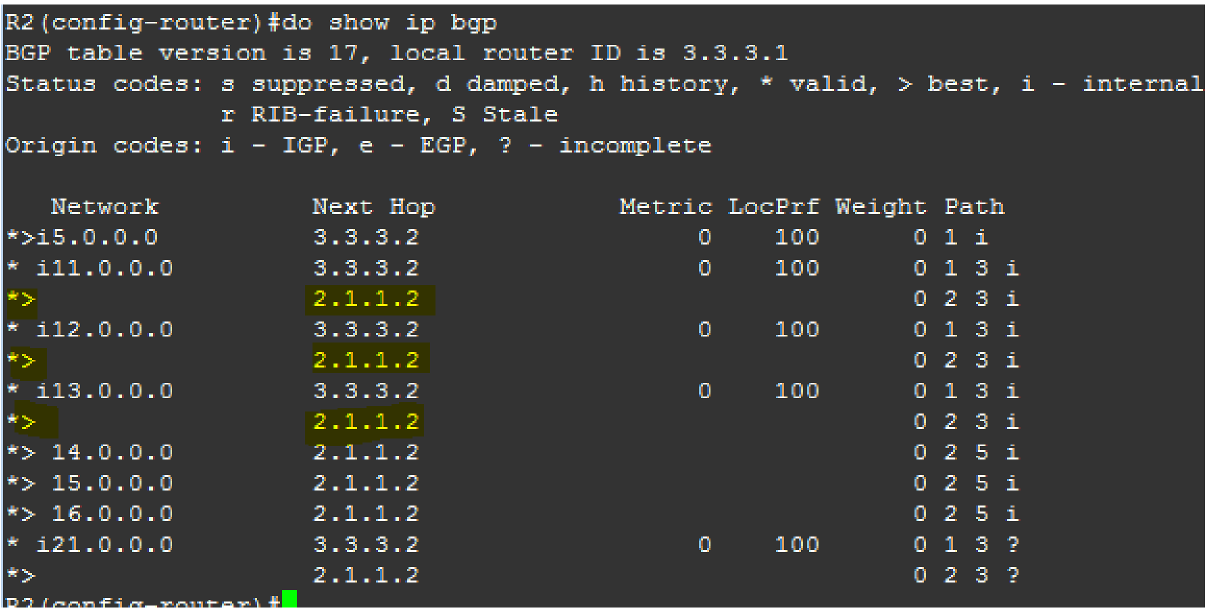
Another method to make one path prefer over another path is by varying the length of the AS path. More AS numbers can be prepended to a particular AS sequence so that their length is increased as compared to the length received through other AS and that route is preferred.

As an example we will use the above mentioned network and we will enable the link between the router R4 and R7 this time. so that R1 router has a route to AS 3 through AS 1 and R2 has a route to AS 3 through AS 2, as ebgp routes are preferred over ibgp routes because rest other parameters in bgp decission process are same. Let's see the output. for 11.0.0.0,12.0.0.0,13.0.0.0 routes on R1 and R2.



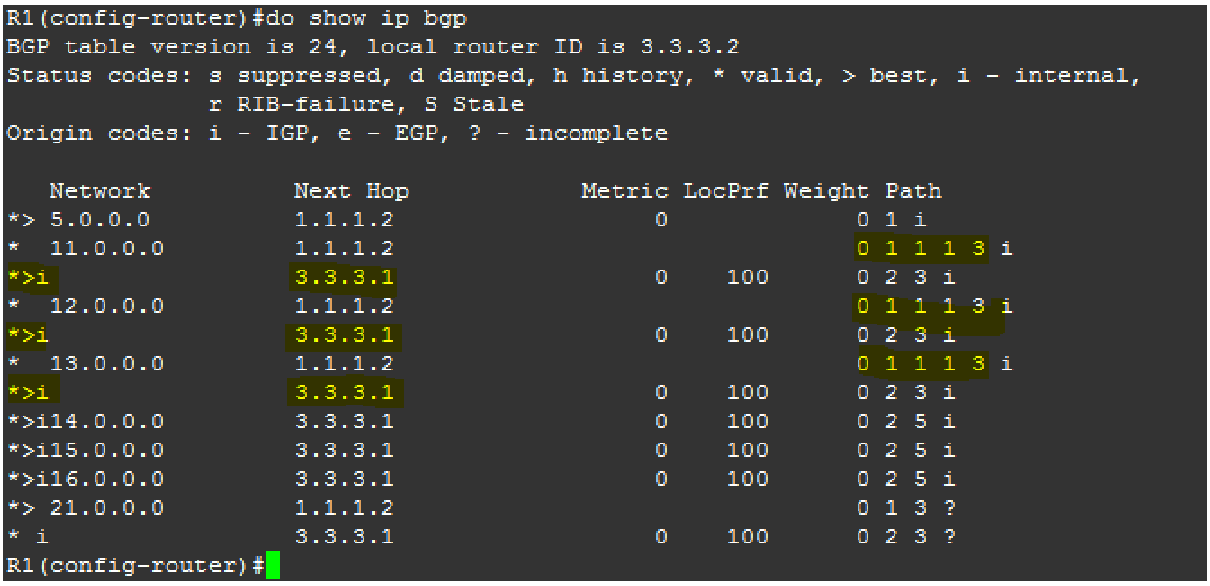
```
R1(config-router)#do show ip bgp
BGP table version is 21, local router ID is 3.3.3.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 5.0.0.0          1.1.1.2                  0             0 1 i
*> 11.0.0.0         1.1.1.2                                0 1 3 i
* i                 3.3.3.1                  0    100      0 2 3 i
*> 12.0.0.0         1.1.1.2                                0 1 3 i
* i                 3.3.3.1                  0    100      0 2 3 i
*> 13.0.0.0         1.1.1.2                                0 1 3 i
* i                 3.3.3.1                  0    100      0 2 3 i
*>i14.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i15.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i16.0.0.0         3.3.3.1                  0    100      0 2 5 i
*> 21.0.0.0         1.1.1.2                                0 1 3 ?
* i                 3.3.3.1                  0    100      0 2 3 ?
```

As we can see for the NLRI 11.0.0.0,12.0.0.0,13.0.0.0 there are two routes available but the best route is the ebgp route which is through AS 1 with next hop of 1.1.1.2.



```
R2(config-router)#do show ip bgp
BGP table version is 17, local router ID is 3.3.3.1
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
*>i5.0.0.0          3.3.3.2                  0    100      0 1 i
* i11.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
* i12.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
* i13.0.0.0         3.3.3.2                  0    100      0 1 3 i
*>                  2.1.1.2                                0 2 3 i
*> 14.0.0.0         2.1.1.2                                0 2 5 i
*> 15.0.0.0         2.1.1.2                                0 2 5 i
*> 16.0.0.0         2.1.1.2                                0 2 5 i
* i21.0.0.0         3.3.3.2                  0    100      0 1 3 ?
*>                  2.1.1.2                                0 2 3 ?
R2(config-router)#
```

As we can see for the NLRI 11.0.0.0,12.0.0.0,13.0.0.0 there are two routes available but the best route is the ebgp route which is through AS 3 with next hop of 2.1.1.2.
Now we will prepend the AS path for these routes on ROUTER R1 and increase the path length such that it becomes greater than the path length received from router R2. Its recommended to prepend the last AS always to avoid any routing loops. For this we will use route maps for prepending the as and matching the interesting routes with an ACL.



```
R1(config-router)#do show ip bgp
BGP table version is 24, local router ID is 3.3.3.2
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
*> 5.0.0.0          1.1.1.2                  0             0 1 i
*  11.0.0.0         1.1.1.2                                0 1 1 1 3 i
*>i                 3.3.3.1                  0    100      0 2 3 i
*  12.0.0.0         1.1.1.2                                0 1 1 1 3 i
*>i                 3.3.3.1                  0    100      0 2 3 i
*  13.0.0.0         1.1.1.2                                0 1 1 1 3 i
*>i                 3.3.3.1                  0    100      0 2 3 i
*>i14.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i15.0.0.0         3.3.3.1                  0    100      0 2 5 i
*>i16.0.0.0         3.3.3.1                  0    100      0 2 5 i
*> 21.0.0.0         1.1.1.2                                0 1 3 ?
* i                 3.3.3.1                  0    100      0 2 3 ?
R1(config-router)#
```

As we can see now the AS path length of routes 11.0.0.0, 12.0.0.0, 13.00.0.0 is increaed and number 1 AS is prepended two times. Hence router R1 chooses the path through Ibgp neighbor in this case as highlighted.

R1 configuration

access-list 2 permit 11.0.0.0 0.255.255.255
access-list 2 permit 12.0.0.0 0.255.255.255
access-list 2 permit 13.0.0.0 0.255.255.255
no cdp log mismatch duplex

```
!
route-map prepend_As_path permit 5
 match ip address 2
 set as-path prepend 1 1
!
route-map prepend_As_path permit 10
router bgp 4
 no synchronization
 bgp log-neighbor-changes
 neighbor 1.1.1.2 remote-as 1
 neighbor 1.1.1.2 update-source Serial1/2
 neighbor 1.1.1.2 route-map prepend_As_path in
```

1.11.4 Filtering using route maps

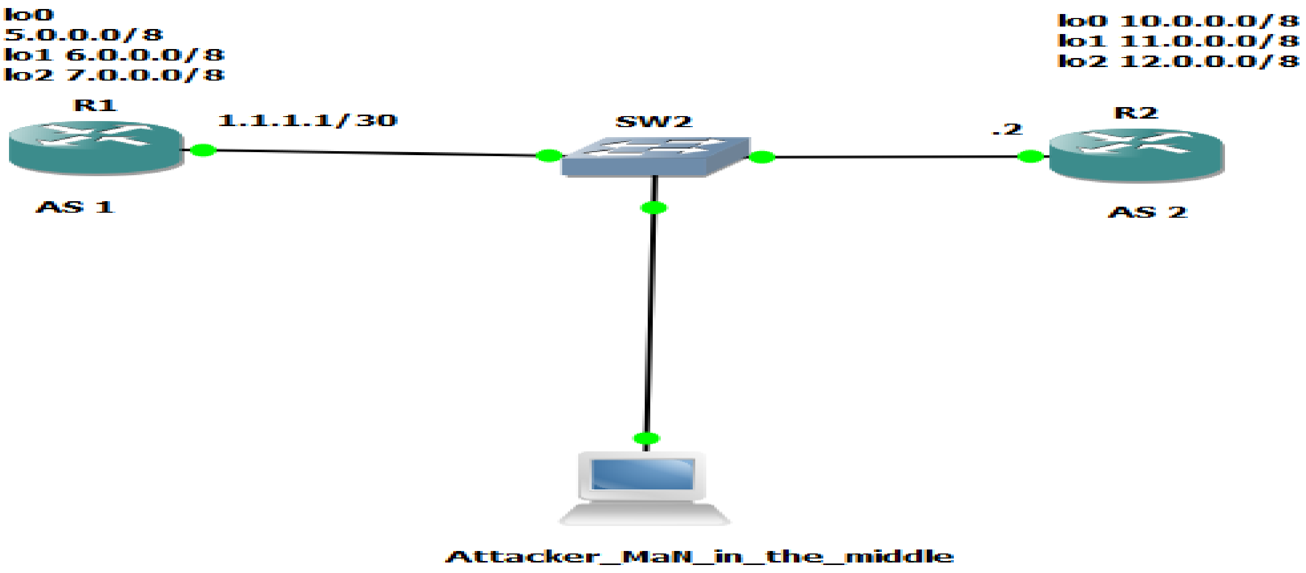
Route map is a powerful tool to filter routes and set various parameters by matching the bgp path attributes and by referring access-lists, prefix lists for matching logic and then setting the various parameters. It has a permit and deny action and a default statement of deny all at the end of route map. As we can see in the above example of as path prepending we have used a route map which matches a particular prefix by referencing to an access-list and then sets the required parameters with the set command. Then this route-map is referenced in the neighbor command under bgp in either inbound or outbound direction.

Section 2 : Security issues in BGP and methods to compromise BGP

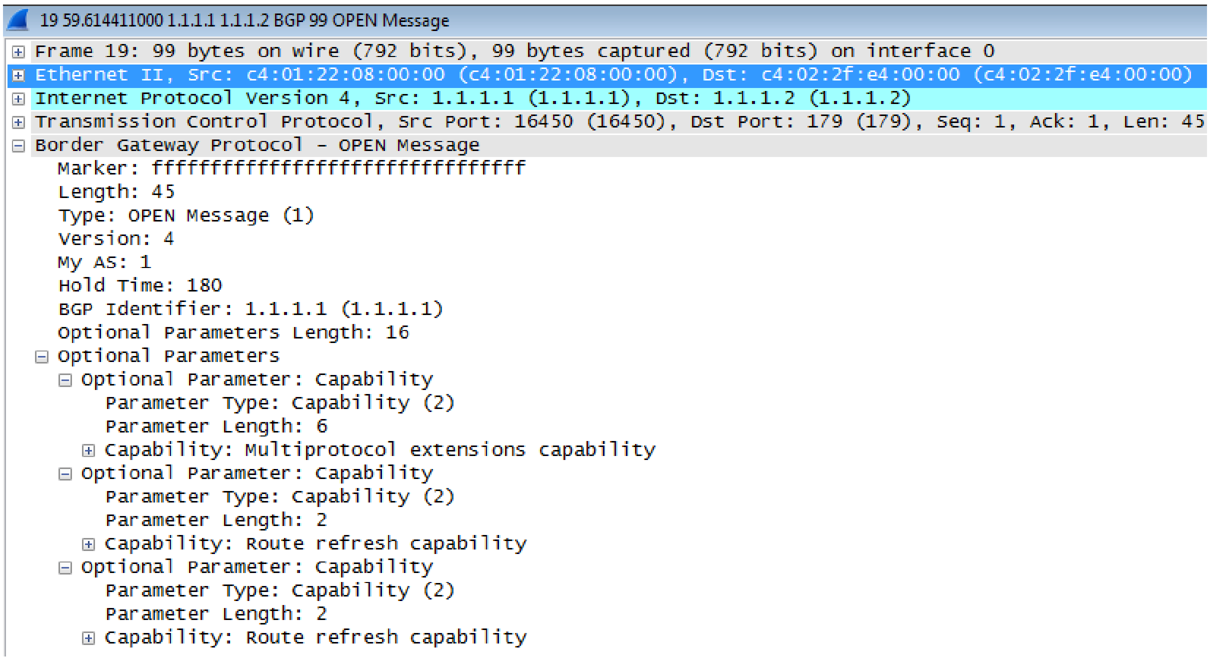
BGP was not designed with security in mind and is hence vulnerable to a variety of attacks. The attacks vary from stealing information by eaves dropping, resetting the bgp sessions, injecting false bgp routes, session hijacking. We will discuss some of the secuirty flaws in bgp below:

1) Eaves dropping : BGP is very vulnerable to eaves dropping as the bgp session between two speakers is not encrypted. Hence the data in transit between the bgp speakers is not secure at all. As an example we will take the below network into consideration and on the attacker pc record the wireshark outputs for the bgp sessions. The attacker is a man in the middle who can launch reconnaissance tools and study the underneath network details which he can use to later exploit the network. As soon as the communication starts between the the two router s i.e the bgp session comes up the attacker starts watching the packets and checks the various parameters starting from the AS numbers, updates, routes, sequence numbers in TCP packets, Ack numbers etc.

Let's see the snapshots.



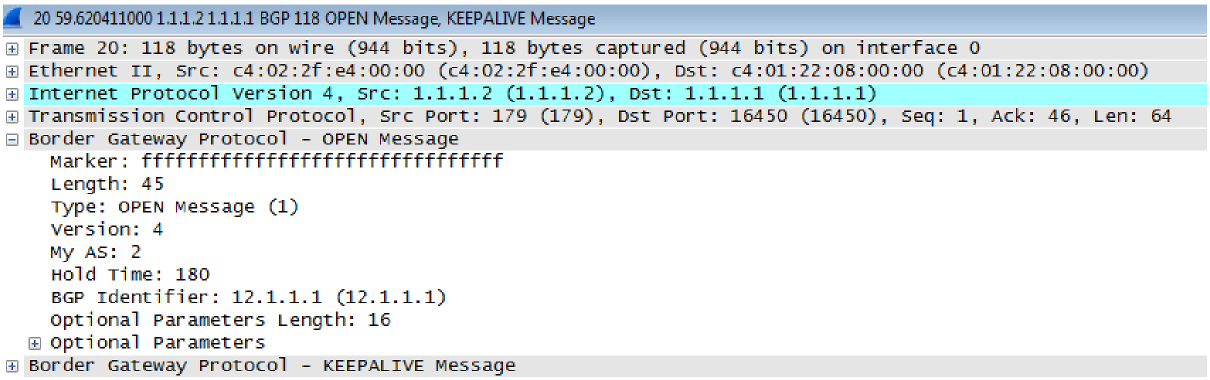
<u>BGP open message</u> (Wireshark capture)



```
19 59.614411000 1.1.1.1 1.1.1.2 BGP 99 OPEN Message
⊞ Frame 19: 99 bytes on wire (792 bits), 99 bytes captured (792 bits) on interface 0
⊞ Ethernet II, Src: c4:01:22:08:00:00 (c4:01:22:08:00:00), Dst: c4:02:2f:e4:00:00 (c4:02:2f:e4:00:00)
⊞ Internet Protocol Version 4, Src: 1.1.1.1 (1.1.1.1), Dst: 1.1.1.2 (1.1.1.2)
⊞ Transmission Control Protocol, Src Port: 16450 (16450), Dst Port: 179 (179), Seq: 1, Ack: 1, Len: 45
⊟ Border Gateway Protocol - OPEN Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 45
    Type: OPEN Message (1)
    Version: 4
    My AS: 1
    Hold Time: 180
    BGP Identifier: 1.1.1.1 (1.1.1.1)
    Optional Parameters Length: 16
  ⊟ Optional Parameters
    ⊟ Optional Parameter: Capability
        Parameter Type: Capability (2)
        Parameter Length: 6
      ⊞ Capability: Multiprotocol extensions capability
    ⊟ Optional Parameter: Capability
        Parameter Type: Capability (2)
        Parameter Length: 2
      ⊞ Capability: Route refresh capability
    ⊟ Optional Parameter: Capability
        Parameter Type: Capability (2)
        Parameter Length: 2
      ⊞ Capability: Route refresh capability
```

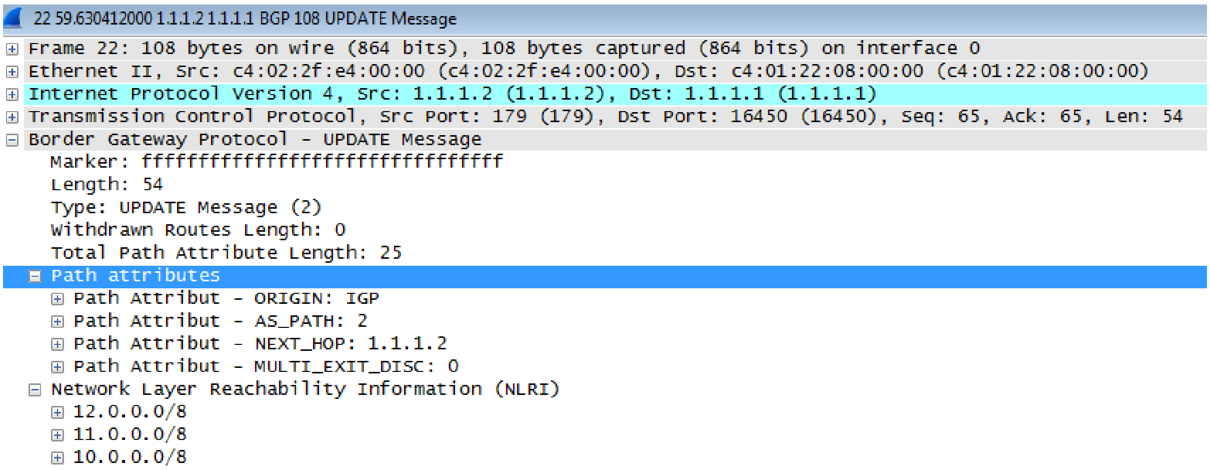
AS we can see the capture shows the BGP open message parameters which includes the ports , ip addresses, involved in the communication, AS numbers and sequence numbers and the optional parameters which states the capabilities.

Open confirm message received from the remote end



```
20 59.620411000 1.1.1.2 1.1.1.1 BGP 118 OPEN Message, KEEPALIVE Message
⊞ Frame 20: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
⊞ Ethernet II, Src: c4:02:2f:e4:00:00 (c4:02:2f:e4:00:00), Dst: c4:01:22:08:00:00 (c4:01:22:08:00:00)
⊞ Internet Protocol Version 4, Src: 1.1.1.2 (1.1.1.2), Dst: 1.1.1.1 (1.1.1.1)
⊞ Transmission Control Protocol, Src Port: 179 (179), Dst Port: 16450 (16450), Seq: 1, Ack: 46, Len: 64
⊟ Border Gateway Protocol - OPEN Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 45
    Type: OPEN Message (1)
    Version: 4
    My AS: 2
    Hold Time: 180
    BGP Identifier: 12.1.1.1 (12.1.1.1)
    Optional Parameters Length: 16
  ⊞ Optional Parameters
⊞ Border Gateway Protocol - KEEPALIVE Message
```

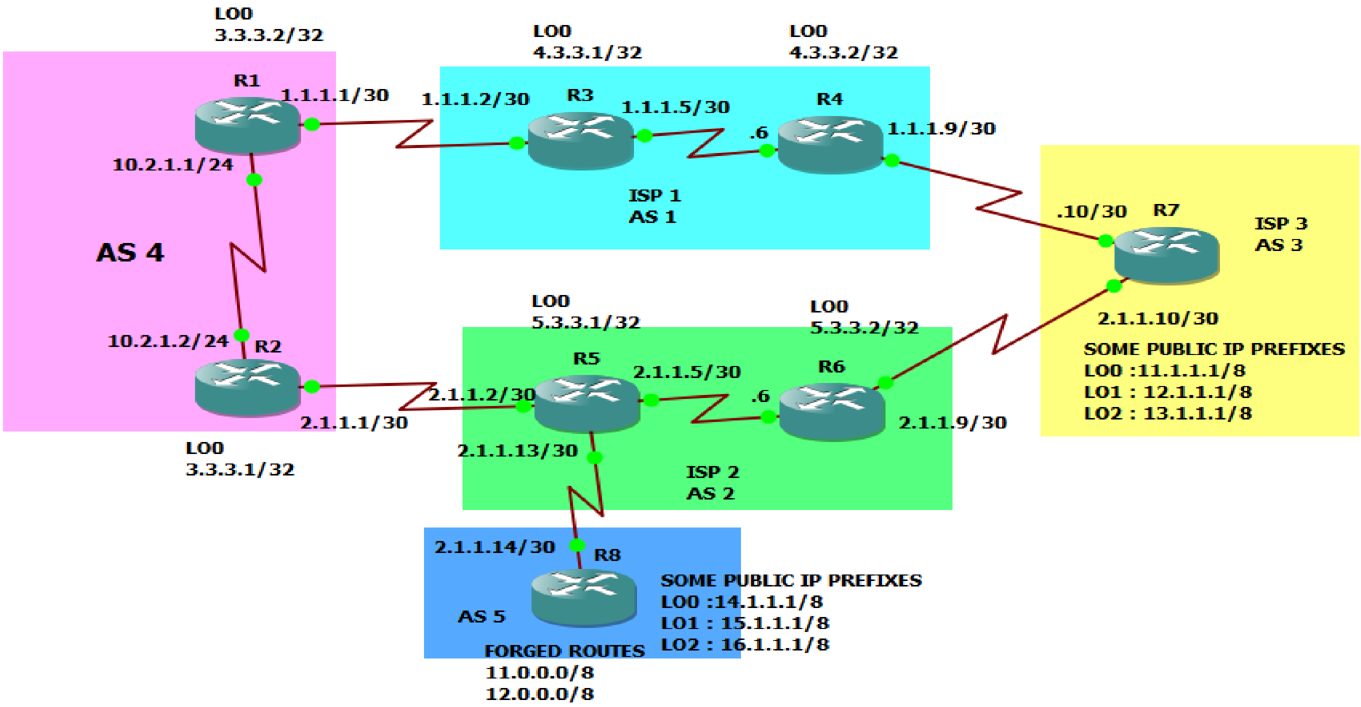
BGP Update message



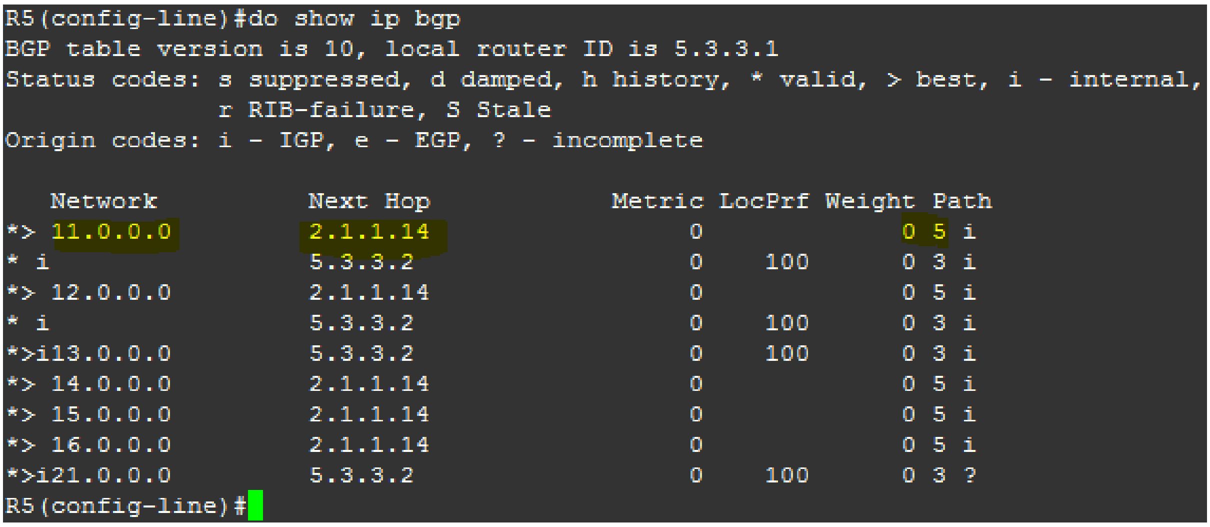
```
22 59.630412000 1.1.1.2 1.1.1.1 BGP 108 UPDATE Message
⊞ Frame 22: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface 0
⊞ Ethernet II, Src: c4:02:2f:e4:00:00 (c4:02:2f:e4:00:00), Dst: c4:01:22:08:00:00 (c4:01:22:08:00:00)
⊞ Internet Protocol Version 4, Src: 1.1.1.2 (1.1.1.2), Dst: 1.1.1.1 (1.1.1.1)
⊞ Transmission Control Protocol, Src Port: 179 (179), Dst Port: 16450 (16450), Seq: 65, Ack: 65, Len: 54
⊟ Border Gateway Protocol - UPDATE Message
    Marker: ffffffffffffffffffffffffffffffff
    Length: 54
    Type: UPDATE Message (2)
    Withdrawn Routes Length: 0
    Total Path Attribute Length: 25
  ⊟ Path attributes
    ⊞ Path Attribut - ORIGIN: IGP
    ⊞ Path Attribut - AS_PATH: 2
    ⊞ Path Attribut - NEXT_HOP: 1.1.1.2
    ⊞ Path Attribut - MULTI_EXIT_DISC: 0
  ⊟ Network Layer Reachability Information (NLRI)
    ⊞ 12.0.0.0/8
    ⊞ 11.0.0.0/8
    ⊞ 10.0.0.0/8
```

As we can see the update packet captured shows the routes exchanged between ebgp neighbors. These include the NLRI's, path attributes, withdrawn routes information which can be used by the attacker to introduce forged routes and see the traffic directed for these routes.

2)**Inserting Forged routes into BGP** : Forged routes can be inserted by an attacker either by compromising the router in some organization running BGP or taking connection from some ISP with no proper security measures and then making establishing sessions with that ISP's routers and directing traffic either to a blackhole or directing it to an area of interest where the attacker can exploit the information and use it for his personal benefits. Accidental insertion of routes can also happen which may lead all traffic to a blackhole. Also the attacker can randomly insert a large amount of routes so as to use the resources of the devices and ultimately crash the routing tables. Sometimes controlled insertion of the routes can also go wrong as it happened in pakistan when they blocked youtube in their country by sending the youtube traffic to blackhole, they accidentally leaked the route out of their country as they didn't filtered their outgoing updates properly which led to the whole worlds youtube traffic going to the blackhole in pakistan. As can example we will consider the below mentioned network diagram and insert a rogue route owned by some other ISP into the system and see the results.



Let's check the bgp table on router r5 or these forged routes. Earlier these routes were installed through router R6.

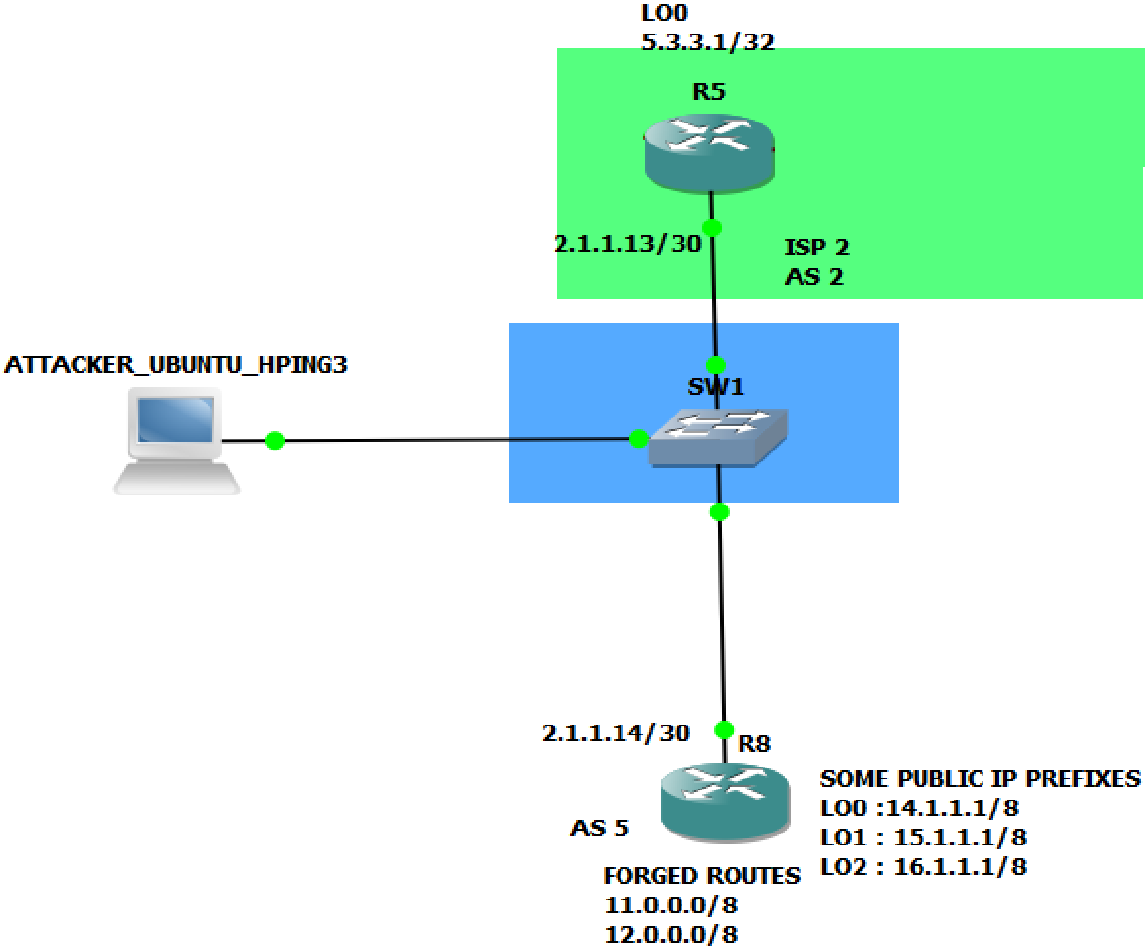

```
R5(config-line)#do show ip bgp
BGP table version is 10, local router ID is 5.3.3.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 11.0.0.0         2.1.1.14                 0             0 5 i
* i                 5.3.3.2                  0    100      0 3 i
*> 12.0.0.0         2.1.1.14                 0             0 5 i
* i                 5.3.3.2                  0    100      0 3 i
*>i13.0.0.0         5.3.3.2                  0    100      0 3 i
*> 14.0.0.0         2.1.1.14                 0             0 5 i
*> 15.0.0.0         2.1.1.14                 0             0 5 i
*> 16.0.0.0         2.1.1.14                 0             0 5 i
*>i21.0.0.0         5.3.3.2                  0    100      0 3 ?
R5(config-line)#
```

As we can see with the accidental insertion or insertion of forged routes by attacker on R8 router the traffic for these routes begin to flow towards a device which is not the original destination of that traffic. Now the attacker who has access to that router is getting all the information directed towards this network and exploiting that information.

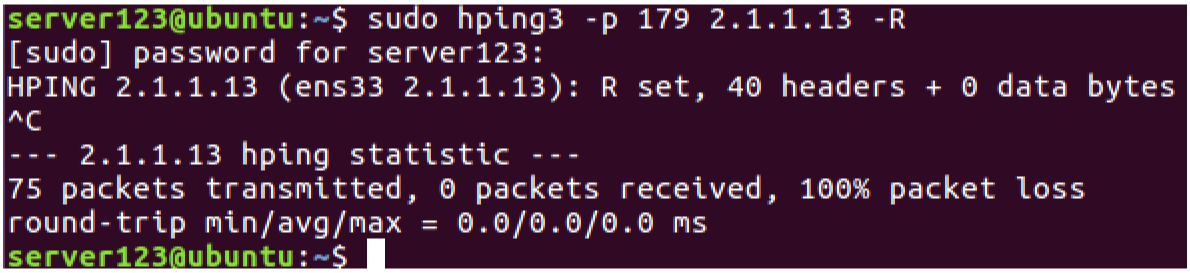
3) **Increasing the length of AS path :** Attacker can increase the length of AS path through a certain hop so that the traffic takes the path as intended by the attacker. It can be done by the attacker either to direct a traffic to a certain place so as to sniff that traffic and then resend that traffic or change that traffic as intended before resending it. The other reason the attacker can do this is due to some personal glitch with an organization or a country and he can intend all the traffic through particular region so as to make the resources over utilized and hence making that network useless or increasing the delay to such a point that the organization

incurs revenue loss. AS path length can be increased by using the tools such as path prepending as discussed in the bgp filtering section; filtering using AS numbers.

4) Resetting the BGP neighbor relationship and flooding bgp using synchronous attack : Attacker can perform a man in the middle attack and use tools such as wireshark to capture packets to see the ip addresses used to set the bgp sessions. Then attacker can spoof the ip address of the bgp speaker and send a tcp reset signal to the other bgp router resetting the tcp connection which will bring down the bgp neighborship. This will delete all the routes received from that neighbor. The routers can again form a bgp relationship and share the updates, but BGP has such a large amount of routes that the convergence time will be significantly high and no data can be sent through the routers when they are in a converging state. And if the reset signal is sent each time the neighborship is formed bgp will never converge which will lead to dropping of data due to lack of routes with each neighbor. Tools such as hping3 can be used to send the tcp reset signal to port 179 on bgp router which will reset the bgp relationship. Also the attacker can send a large number of tcp synchronization packets such that tcp service on the victim is exhausted. Since bgp uses tcp, there will be impact on bgp sessions as well and it will take a long time to establish a bgp session, thus increasing the convergence time. As an example we will perform a tcp reset attack using hping3 from an attacker machine which is ubuntu latest version. We will use a part of our main network diagram for this.



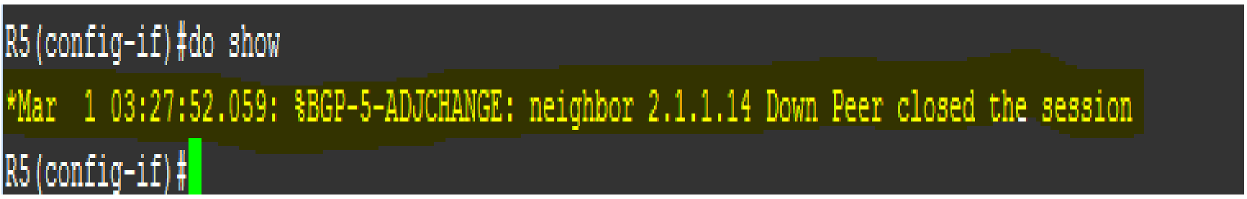
Attacker machine first utilizes packet capturing utilities to see the ip addresses involved in session building and then spoofs the ip address of one bgp peer to send a reset signal to other bgp peer.



```
server123@ubuntu:~$ sudo hping3 -p 179 2.1.1.13 -R
[sudo] password for server123:
HPING 2.1.1.13 (ens33 2.1.1.13): R set, 40 headers + 0 data bytes
^C
--- 2.1.1.13 hping statistic ---
75 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
server123@ubuntu:~$
```

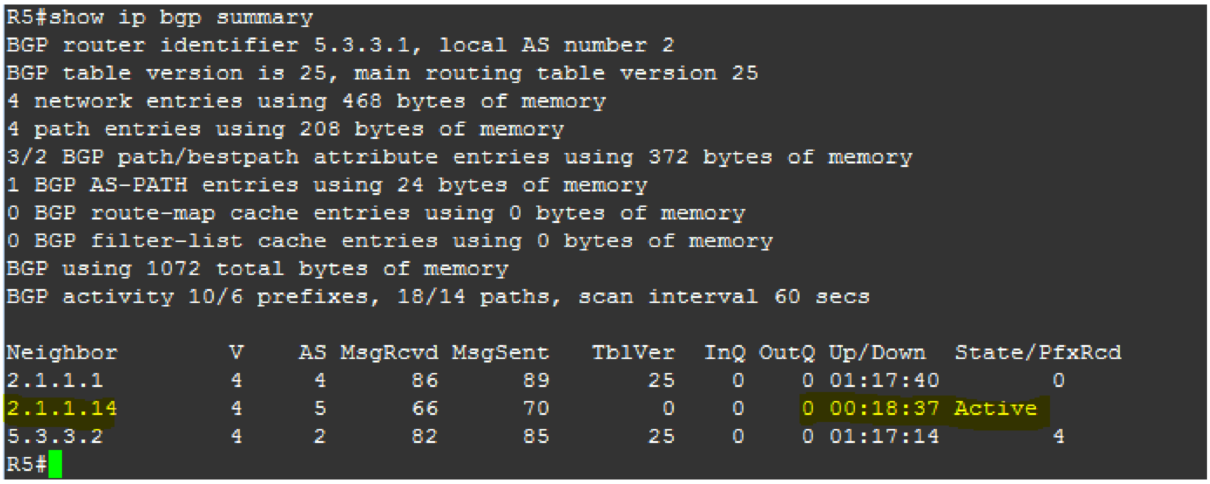
Hping3 attack to send the tcp reset to router R5.

Result on router R5



```
R5(config-if)#do show
*Mar  1 03:27:52.059: %BGP-5-ADJCHANGE: neighbor 2.1.1.14 Down Peer closed the session
R5(config-if)#
```

Notification received on router R5



```
R5#show ip bgp summary
BGP router identifier 5.3.3.1, local AS number 2
BGP table version is 25, main routing table version 25
4 network entries using 468 bytes of memory
4 path entries using 208 bytes of memory
3/2 BGP path/bestpath attribute entries using 372 bytes of memory
1 BGP AS-PATH entries using 24 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 1072 total bytes of memory
BGP activity 10/6 prefixes, 18/14 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.1.1.1         4     4      86      89       25    0    0 01:17:40        0
2.1.1.14        4     5      66      70        0    0    0 00:18:37 Active
5.3.3.2         4     2      82      85       25    0    0 01:17:14        4
R5#
```

The session goes in active state due to tcp reset signal

5) Session Hijacking : Session hijacking involves compromising the existing bgp session i.e., the attacker successfully masquerades as one of the peers in a BGP session. the motive may be to change routes used by the peer, in order to facilitate eavesdropping, sending the traffic to blackhole or traffic analysis.

6) **Route Flapping :** An attacker can change the routes very rapidly in a routing table thus inserting and deleting several routes per second from the BGP table. This can be very problematic and results in large bgp updates as the router has to propagate and withdraw a large number of routes which prevents convergence of valid routes and as long as the routing protocol is converging no traffic can flow resulting in slowdown and even loss of packets in a network. Mechanisms such as route dampening have been introduced which prevents the propagation of routes which are flapping too much in a given time thus preventing such kind of attacks but still bgp is vulnerable to such attacks.

7) **Dos and DDOS attacks:** Since routers have a finite storage and memory dos and ddos attacks can be used to criple the router resources and render them useless. Attacks such as

synchronization flooding can be performed which can leave the routers useless and as such bgp is also affected as it uses tcp for its communication.

8) Line cutting attacks : The attacker can cut certain links in a path so that the traffic is forced to take a path as intended by the attacker. Tools such as encryption can be used to render the traffic not readable by the attacker for this.

Note : The consequences of the above mentioned attacks can be very devastating. The BGP sessions can be brought down, the address space can be fragmented and misused and the Ases can be made unreachable or the path which a packet follows through different Ases to reach its destination can be altered as per the attackers convenience. Attacks can be further used to make bigger attacks and distributed attacks can also be performed making it difficult to detect the origin of attacks. Information can be compromised and traffic can be clogged and blackholed and the hijacked sessions can be further used to update incorrect routing information and can be used to reset the TCP neighbor relationships which can further lead to route recalculation which takes a bit of time as BGP has large number of routes in its routing table.

Section 3 : Previous researches and methods proposed to mitigate the security Threats

3.1 Best practises

1**)** Disable synchronization as synchronization requires the routes from bgp to be distributed into igp and vice versa. The amount of memory required to hold the same number of routes as in bgp when redistributed in igp is very high as compared to bgp which can result in memory overutilization and ultimately crashing of the router. So, its best practise not to rely on synchronization.

2) Enable logging for session establishments and peer downs or any fluctuating links.

3) Perform summarization in an effective way such that the routing table size is kept to minimal.

4) Use authentication mechanisms such as MD5 or ESP in IPsec. IP sec can also be used to encrypt BGP traffic but it has a drawback of increasing the delay due to encryption and decryption of several packets per message.

5)Block inbound announcements for bogon prefixes. These are the prefixes which are not yet allocated by IANA but are reserved. So these address should never be routed in the internet and if there are any packets coming or going to these address those packets are compromised. But these prefixes shall be carefully noted as they can be allocated by IANA at any point of time and such if they are not installed in the routing table the whole traffic can be lost. Routers can either link to BGP BOGON update server or can manually look for the updated Bogon addresses.

6)Tools such as maximum number of prefixes allowed in the routing table can be used in order to limit the max prefixes that can be installed in the routing table. This can prevent attacks such as router table overflow which can lead to crashing of routers due to overutilization of memory or can result in resetting the BGP neighbor relationships.

7) Since the BGP connects on the default port of 179 all other ports for bgp connection shall be blocked so that no other device can try connecting BGP at some other port.

8) Access control lists shall be properly implemented to deny any illegal traffic.

9) Reverse path source address validation can be performed so that bogus source address packets can be ignored.

10) Another method which was developed is TTL hack. TTL value of 255 is sent for the updated packets and we know that for BGP most of the times the peer is adjacent. As such the TTL value decrements by 1 most of the times and such if a decremented value for TTL received is less than 254 that update is a forged update.

12) Online Tools such as BGP mon can be used to monitor the health of an Autonomous system. This tool informs about the route hijack, policy violation or network instability and can also be integrated with the existing monitoring systems. It allows to monitor all the prefixes in an AS or some prefixes in an AS by monitoring them from over a hundred vantage points worldwide, allowing regional events to be detected that might otherwise not be detected by single vantage point monitoring systems.

3.2 Other BGP security architectures to counter BGP threats

1)SBGP

It was the first security solution targeted for BGP security. It uses the public key certificates to communicate the authentication data. Certificates use cryptography for identity and anyone in possession of the public key certificate can validate that information usng the private key associated with the public key. SBGP validates the data traffic between ASes using PKI infrastructure. SBGP uses a pair of PKI used to allocate address space and AS numbers. The first PKI is used to verify the address space allocated to a given organization while the second PKI is used to bind AS numbers to the organization. Hence all data including Address ownership, peer AS identity, path attributes, control messages are all signed using certificates. Therefore the receiver can easily authenticate the routing info and analyse and drop the forged data. Address attestations are performed which give an organization a right to originate the prefixes. Route attestations are distributed within SBGP in a modified BGP update message as a new attribute. Route attestations are digitally signed by each previous AS and as such the route attestation contains all the digital signature of previous ASes, hence path validation can be performed and it can be cross checked whether somebody has modified the packet in any way.

Drawbacks:

1) To much overhead due to cryptography will lead to a large delay as for every update generated it will take a noticeable time to create and verify the digital signatures which will slow down the network convergence.

2) If a certificate is compromised then it will be a lot of problem for every AS in between resulting in frequent queries which will ultimately result in overutilization of resources.
3) Requires change in the IOS of every router running an AS to update the BGP protocol which will require an extensive amount of effort and downtime to be accomplished which is practically not feasible in today's world.

2) Secure Origin BGP :

Its operation is similar to sbgp except that it avoids security and protocol overhead using protocol parameters. It also uses PKI for authenticating and authorizing purposes and uses three certificates kinds. The first certificate is used for BGP speaker authentication. Second one provides details on underlying topology and policies along with the protocol parameters. The third one is used for address allocation authorizations. All this information is transmitted using bgp secure message of Secure origin BGP. Topology database is used to validate received routes. Every AS generates a topology certificate containing the topology database and send it to other Ases to form a global topology database and this can be used to verify the authenticity of the received routes and any forged routes can be dropped. The computational cost of verifying signatures is overridden by performing these tasks before even BGP neighborship is established and authenticated data is stored and validated prior to establishing these sessions. The topology database is used to verify he received paths instead of inquiring each time the neighboring AS, thus reducing the delay.

Drawbacks:

1) This approach involves every router to contain the database of every other autonomous system. Every time a topology change is there in an AS it regenerates the topology database certificate adding to a large amount of overhead sent to the neighboring routers which have to recalculate the routes every time, thus increasing the convergence time practically not feasible.

2)The topology database itself requires routers to have a large memory in order to store every AS topology database which increases the costs to very high value.

3) It requires complex changes to the existing BGP architecture and all routers need to be updated with the new BGP written code which requires a lot of downtime which is very difficult to manage.

3) Pretty Secure BGP (psBGP):

psBGP uses a centralized model for authenticating AS numbers and a decentralized model for verifying IP prefix ownerships. The centralized model involves each AS obtaining a public key certificate and binding that certificate to AS. This provides authorization information for that particular AS which an attacker is not able to compromise. The IP prefix ownership validation involves each AS creating a prefix assertion list consisting of a number of bindings an AS number and prefixes belonging to that AS as well as the neighboring AS and its prefix numbers. This will fairly lead to validation of originating AS for prefixes.

Drawbacks:

1) Approach involves minimal overhead but the decentralized approach i vulnerable to hacking by an attacker and can lead to forged routes insertion.

4) Interdomain Route Validation(IRV) :

It is a receiver driven protocol and its operation involves the presence of an IRV server in every autonomous systems. Upon reception of an update by a BGP router it send that update to IRV server for verification of the received information. The local IRV server contacts the relevant AS for route validation and if validation from multiple Ases is needed it contacts the multiple IRV servers in different ASes for the verification process. A BGP speaker can then act on the received data and intall or reject the route as per the information received from the IRV server.

Drawbacks:

1) If a route traverses a large number of Ases it can be very problematic as the IRV server will query every IRV server in every AS in the path which can lead to a large time delay.
2) The IRV server involved in each AS can be difficult to trust as if there are too many IRV servers even if one IRV server is compromised it can be difficult to detect.

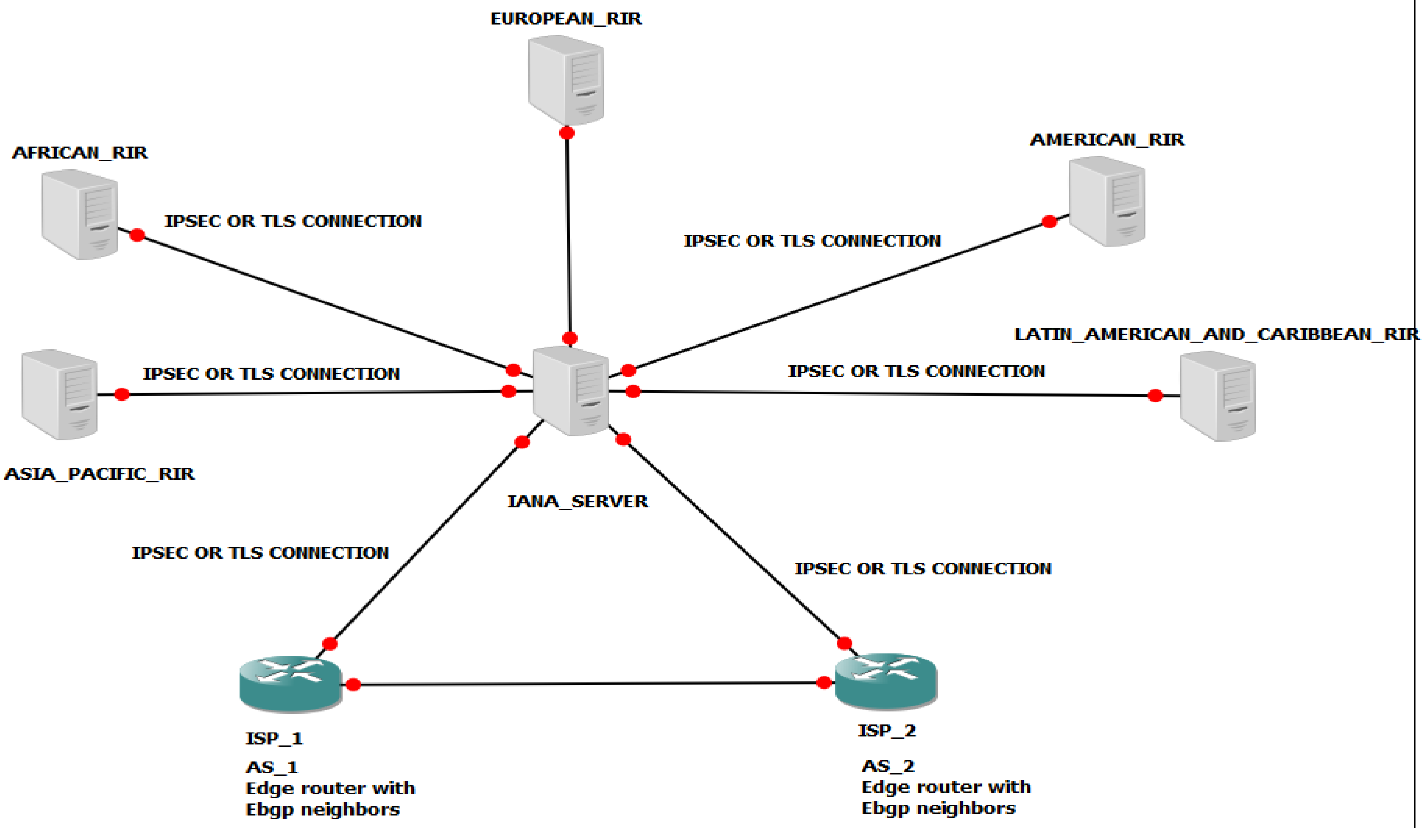
3.3 Problems with the above mentioned solutions Why these solutions are not implemented till now?

1) Securing BGP and planning to deploy requires a large coordination among different organizations both ISP's and the hardware and software providers which will cost them a huge amount of money in terms of upgradation and downtime .Neither these solutions guarantee that the problems poised in BGP can be solved perfectly with these solutions. Thus the lack of interest by these specifics have lead to lack of development and deployment of security solutions.
2) The sheer size of internet which has millions of devices makes this idea impractical as it will take at least some years to implement these solutions.
3)Too much cryptographic requirements will eventually impact the performance of the network

3.3.1 My ideas and solutions to solve the problems

After studying the above mentioned solutions and the drawbacks associated with the above solutions I have tried to bring forward some solutions which can solve the problem of BGP security to some extent without too much modification in the code of existing interoperating system by different vendors and with least use of cryptography. I would like to discuss the solutions below with the theory and I'll try to provide a proof of concept for the same.

3.3.1.1 Solution 1: Centralized approach for origin authentication and hashing function for route validation



As we know IANA and ICANN are the main organizations which control the distribution of PI prefixes and Autonomous systems numbers to various organizations through the specific regional information registries present in their regions. A such it has all the database containing information about which prefix is assigned to which organization and to which information about the IP prefix and its originating AS from IANA rather than contacting the ISP directly as given in the IRV server approach as the ISP itself is not a legal authority to decide whether a particular ip prefix is allocated to it or not. Additionally the IRV server can be compromised and incorrect information can be recorded in that and it is very difficult to manage security across every IRV server in every AS. Moreover the IANA server will have the list of neighboring AS to every AS which can be useful in cases of path validation or route validations. This approach doesn't involve too much overhead as there is no need to contact validate every AS number in the path. Just the first two AS numbers are verified the first one being the originating AS number and the second being the neighboring AS.
The edge routers of the organizations which are sharing the EBGP routes are the ones which have connection to IANA server as they are the first ones to receive the EBGP updates and they are responsible for further sending the update in their autonomous system from where it is propagated to other autonomous system. The edge routers are authenticated by IANA server by using digitally signed certificates(strong) or can be done using pre-shared keys(weak). The data between them can be encrypted using IPSEC or TLS or any third party encryption mechanism agreed by both sides can be used.
The IANA server itself is connected to RIR servers from where the information about prefix allocation and AS number allocation is collected and stored in the central database at the IANA server. Thus the IANA server database contains a list of ip prefixes corresponding to

their originating AS numbers as well as a list of the neighboring AS numbers to that AS. As an example the structure can be like shown below at IANA server.

AS 54

1) Prefix

5.0.0.0/8
6.1.1.0/24
7.1.0.0/18

2) Neighboring AS

57
58
59
60

As we can see the root AS is 54 which has the mentioned prefixes 5.0.0.0/8, 6.1.1.0/24, 7.1.0.0/18
allocated to it. It neighboring AS is 57,58,59,60. Now when the EBGP routers ISP_1 or ISP_2 obtain an EBGP update regarding the routes mentioned above they do the following:

1) Extract the prefix information and the first two AS path information(it will be the originating AS and the neighboring AS) from the update and send it to IANA server for verification.

2) The IANA server parses its database and looks for the mentioned prefix under the associated AS number. It also checks for the neighboring AS to that originating AS and verifies that with the information received from the EBGP router for that particular route.

3) If the information in the update matches the information in the IANA database a message is sent back to EBGP router that the information received is verified and can be trusted to be installed in the routing table and can be updated to others.

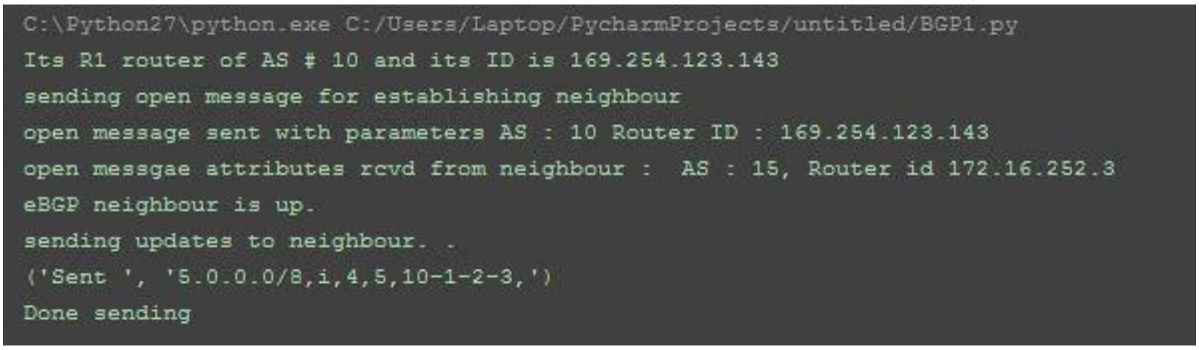
This way it can be verified that whether the originating AS is authorized to originate the prefixes and the verification of neighboring AS will provide a double check for the same. To solve the problem of route validation so that additional fields are not inserted by the attacker in the AS field to make that AS as atrffic AS where he can steal the information, there shall be a length field in the BGP messages which indicates the AS path length. This AS path length is the total number of Ases that the prefix has travelled before reaching this particular router as an update. Suppose if the prefix 5.0.0.0/8 has the update like listed below: (just the prefix and the transit Ases are shown for simplicity other parameters are ignored)

5.0.0.0/8 1 55 56 59 61 67 78 89 54

The AS length in this case is 9. This field is encrypted and is not visible to the attacker and it is calculated by using a hash function. This way if the attacker modifies any Autonomous system number in the AS path field the hash changes and it is not equal to the hash calculated by the receiving router. As such the receiver understand that the update has been modified in

between and it is not the right update to be installed and it ignores the update. This way both the originating AS and the route path are verified.
As a test I have established EBGP relationship between R1(AS10) and R2(AS15) router and send an update from the R2 router as shown in the screenshot for the NLRI

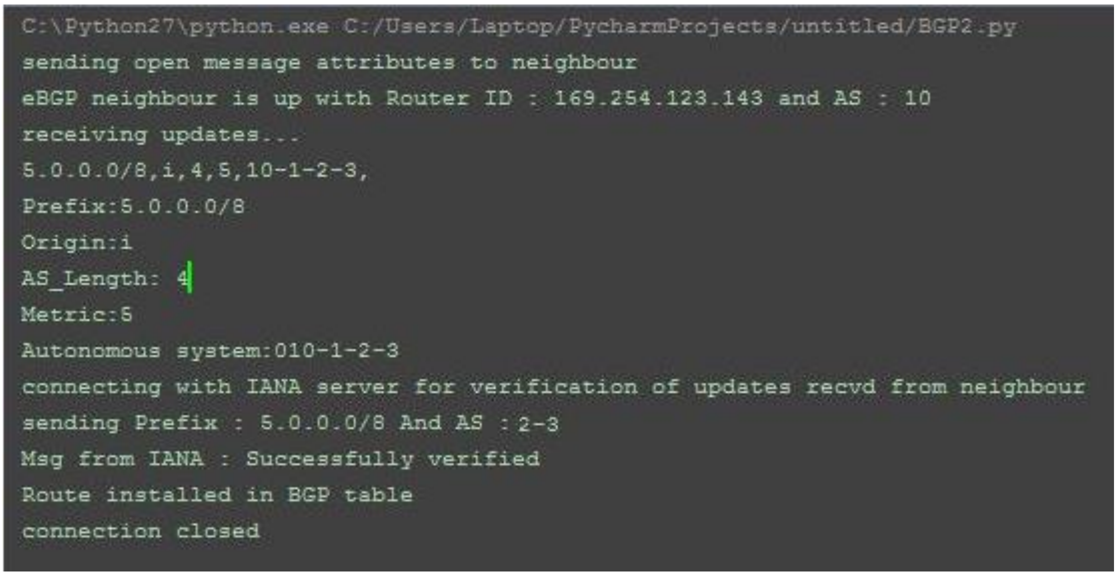
5.0.0.0/8 i 4 5 10-1-2-3



```
C:\Python27\python.exe C:/Users/Laptop/PycharmProjects/untitled/BGP1.py
Its R1 router of AS # 10 and its ID is 169.254.123.143
sending open message for establishing neighbour
open message sent with parameters AS : 10 Router ID : 169.254.123.143
open messgae attributes rcvd from neighbour :  AS : 15, Router id 172.16.252.3
eBGP neighbour is up.
sending updates to neighbour. .
('Sent ', '5.0.0.0/8,i,4,5,10-1-2-3,')
Done sending
```

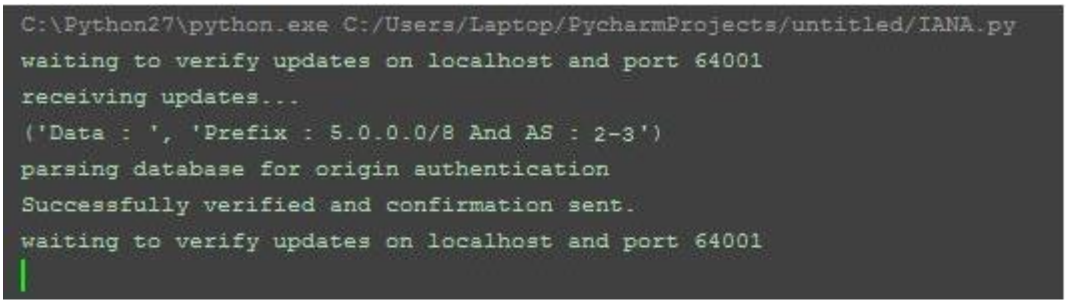
This update contains:
prefix = 5.0.0.0/8
origin code = i
AS path length = 4
Metric = 5
AS_Path = 10-1-2-3

R2 receives this update and verifies the AS path length which is 4 in this case by calculating the hash over the AS path field and considering this as an unmodified update the hash sent by sender and receiver is same. After that the prefix information and the originating AS and neighboring AS is being sent to IANA server for verification.



```
C:\Python27\python.exe C:/Users/Laptop/PycharmProjects/untitled/BGP2.py
sending open message attributes to neighbour
eBGP neighbour is up with Router ID : 169.254.123.143 and AS : 10
receiving updates...
5.0.0.0/8,i,4,5,10-1-2-3,
Prefix:5.0.0.0/8
Origin:i
AS_Length: 4
Metric:5
Autonomous system:010-1-2-3
connecting with IANA server for verification of updates recvd from neighbour
sending Prefix : 5.0.0.0/8 And AS :2-3
Msg from IANA : Successfully verified
Route installed in BGP table
connection closed
```

IANA server parses its database which is sql database and checks for the originating AS 3 and verifies the prefix information 5.0.0.0/8 and the neighboring AS field 2 under that AS.



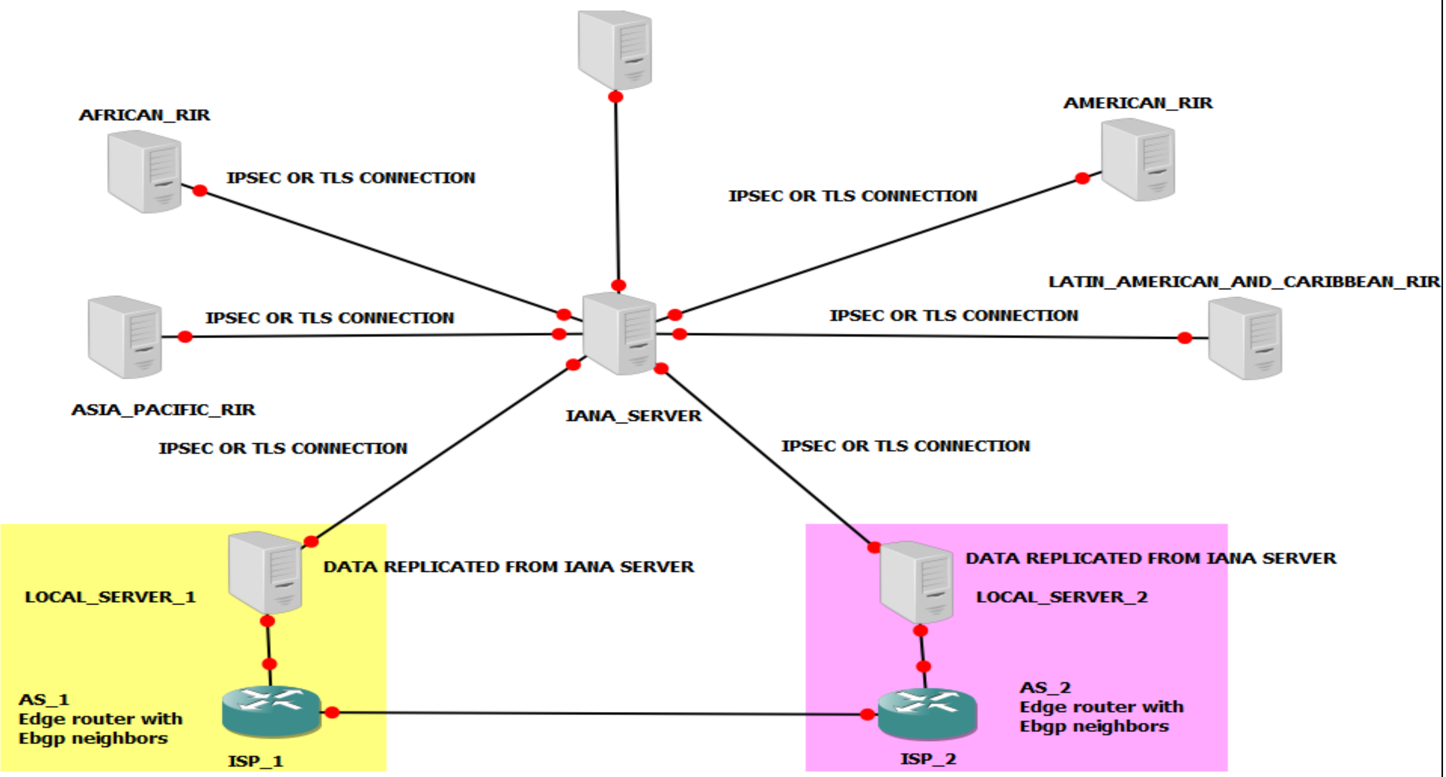
```
C:\Python27\python.exe C:/Users/Laptop/PycharmProjects/untitled/IANA.py
waiting to verify updates on localhost and port 64001
receiving updates...
('Data : ', 'Prefix : 5.0.0.0/8 And AS : 2-3')
parsing database for origin authentication
Successfully verified and confirmation sent.
waiting to verify updates on localhost and port 64001
```

As can be seen the entry is found in that database for the listed prefix and neighboring AS and is being sent to R2 that the update is ok and successful verification message is sent to R2. R2 then installs then route in its database.

But this approach will cause too much queries on the IANA server as if there are 100 Ases in the path lookup at IANA server will become complicated as it can receive multiple requests at a single point of time. So approach in solution 2 can make this process possible as it involves the concept of local server present with each organization which can have multiple servers to meet multiple requests from a single organization which will solve the problem of overloading and a single point of failure.

3.3.1.2 Solution 2 : Decentralized approach for origin authentication and hashing function for route validation

It is a modification of solution 1 and it involves a server or a cluster of servers in each AS system which is connected to IANA server. This approach takes off the load from the IANA server and address the problem of single point of failure in the system and speeds up the lookup process as every organization has now replicated the database of IANA server in order to fasten the lookup process which can become slow if a single IANA server or a cluster of IANA servers receives multiple requests from different ASes at a single point of time.



As can be seen the IANA database is now available at the local servers of the organization which can be used to verify the update requests sent by the EBGP neighbors before installing them in the routing table. The servers can be a group of servers which can load balance the requests received from the BGP routers. IANA server can authenticate these local servers by means of certificates and then can multicast the updates to all local servers which can then store the updated information in their local database. This way every organization can have a view of the prefixes and the AS to which they are related and the neighboring Autonomous system numbers as well. This way the originating AS number can be verified and the route validation can be done by the same hashing function as in solution 1 and the forged routes can be ignored.

Advantages of solution 1 and 2:

1) IOS of different vendors do not need to be modified extensively. It requires a little modification which requires that before the update is installed in the routing table it is verified prior to that.

2) The cost to implement this solution is minimal as the existing infra can be used to create this solution. It doesn't require any specialized hardware.

Disadvantages:

1) Although the modification required is very less but still the routers IOS has to be upgraded which will still require some downtime which can make these approaches not in interest of organizations.

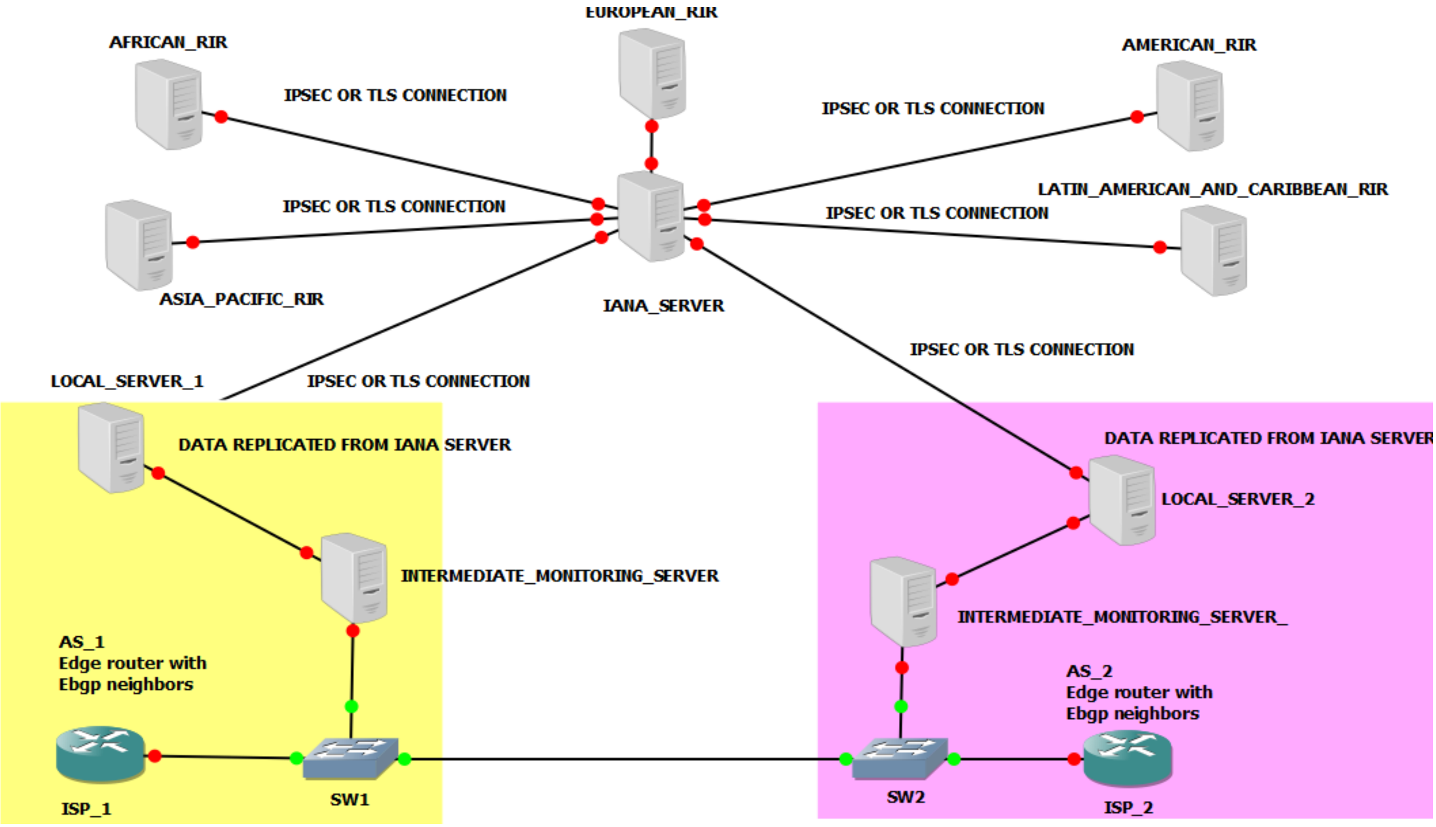
3.3.1.3 Solution 3 : Passive approach. Sniffing the EBGP updates and generating alerts.

This solution counters the above drawbacks and is in the interests of different organizations and vendors. It involves following a passive approach instead of active approach which requires no modification to their routers software hence a very little downtime or no downtime is required. This requires passive monitoring system which requires no changes to the active routers hardware and software's but still alerts the organization of the incoming threats. This solution requires manual intervention on the router to filter the routes by the administrator. It provides an alert to the administrator of the organization by monitoring the EBGP updates.

As can be seen in the below mentioned diagram now the updates received are sent to the monitoring server which checks the updates and verifies the updates with the local server which has the database in connection to IANA server. If any update is received which is compromised or is either generated by some other intermediate AS which is not the owner of that prefix an alert is generated. The administrator can manually intervene seeing the alert and can filter or allow that prefix. The rest network is same as mentioned in the above solutions. This approach is passive as there is no automated action to the problem. This solves a lot of problems with respect to vendors and its

Advantages are :

1) No modification is required to the existing software and hardware of the routing platforms.
2)Administrator can manually intervene and thus can look into any false alarms generated
3) Tuning can be done as per the organizations requirement for the monitoring software and there is no requirement of routing platform vendor.
4) It's a cheap solution to implement.



Summary and Future Work

The project covers the concepts of BGP protocol in detail with its configuration and different ways to configure BGP and tune its path attributes, security vulnerabilities and previous research works done in enhancing the security of the protocol and the drawbacks of the previous work done. Also it includes some new ideas to make BGP more secure which tries to eliminate the drawbacks of the previous research work done on securing the protocol.

Since BGP is the only inter domain routing protocol its security needs to be improved to meet the demands of today's complex networks. The above ideas can be implemented and tested in accordance with the different vendors such as Cisco, juniper, Alcatel who have a large number of devices working in the internet. As such future work can be done in coordination with these vendors to implement these ideas and get test results in live environment and hence improve the above ideas to get a concrete solution to the problem of BGP security.

References

1)CCNP_ROUTE_642_902_Implementing_Cisco_IP_Routing_ROUTE_Foundation_Learning_Guide
2) Routing TCP/IP, Volume II (CCIE Professional Development) By Jeff Doyle CCIE #1919, Jennifer DeHaven Carroll CCIE #1402
3) CCIE Routing and Switching Certification Guide, Fourth Edition
4) http://ix.cs.uoregon.edu/~butler/pubs/bgpsurvey.pdf
5) https://www.wired.com/images_blogs/threatlevel/files/nist_on_bgp_security.pdf
6) http://www.blackhat.com/presentations/bh-usa-03/bh-us-03-convery-franz-v3.pdf
7) http://people.scs.carleton.ca/~kranakis/Papers/TR-05-07.pdf
8) http://www.hping.org
9) https://tools.ietf.org/html/rfc4271
10) https://tools.ietf.org/html/rfc1771
11) https://tools.ietf.org/html/rfc1105
12) https://tools.ietf.org/html/rfc1163
13 https://tools.ietf.org/html/rfc7454
14) https://www.ietf.org/rfc/rfc4272.txt
15) https://tools.ietf.org/html/rfc5082
16) https://www.ietf.org/rfc/rfc4264.txt
17) https://tools.ietf.org/html/rfc6480
18) https://en.wikipedia.org/wiki/Autonomous_system_(Internet)
19) https://networklessons.com/bgp/bgp-private-and-public-as-range/
20) https://labs.apnic.net/?p=447
21) https://www.iana.org/assignments/as-numbers/as-numbers.xhtml
22) https://www.ietf.org/rfc/rfc1930.txt
23) https://en.wikipedia.org/wiki/Border_Gateway_Protocol