# PRACTICE REPORT, BGP CONFIGURATION GUIDE

ADVANCED ROUTING

daniel.juarez@iteso.mx, enrique.rios@iteso.mx, TEAM 5

2024-11-24

# Contents

# EXECUTIVE REPORT

## PROJECT OVERVIEW

This **BGP Configuration Guide** provides a comprehensive approach to implementing Border Gateway Protocol (BGP) in a multi-homed network, where multiple external connections improve both connectivity and resiliency. BGP, a protocol designed to exchange routing information between autonomous systems (AS), enables networks to make informed decisions about the best paths for data based on criteria beyond simple distance. In this guide, particular attention is given to two essential BGP attributes: the Multi-Exit Discriminator (MED) and Local Preference, both of which influence the path selection process based on policy rather than only shortest-path metrics.

The MED attribute is used between neighboring autonomous systems to indicate a preferred path for traffic entering from a specific external source. By setting MED values, administrators can direct inbound traffic through preferred entry points in cases where multiple paths exist. This offers control over how incoming traffic flows across the network, enabling load balancing and enhanced control of data traffic between providers and customers.

The Local Preference attribute, used within an AS, gives preference to specific exit points, effectively allowing internal routers to prioritize certain paths for outbound traffic. Higher Local Preference values are preferred, which allows for a highly customizable path selection process that can reflect specific network policies and requirements for redundancy and performance.

To ensure seamless communication in a multi-homed setup, this guide also emphasizes the importance of configuring both eBGP and iBGP sessions. eBGP, or external BGP, facilitates routing information exchange between different autonomous systems, while iBGP, or internal BGP, is configured between routers within the same AS to ensure that learned routes are disseminated internally. As BGP relies on a session-oriented approach using TCP connections between routers, the configuration of these neighbor relationships is vital for maintaining stable and predictable routing.

Additional techniques, such as using prefix-lists and route-maps, allow for granular control over routing policies. Prefix-lists serve as a filter mechanism to control which networks are advertised or accepted, while route-maps provide a way to set conditions on routing behavior, applying different preferences based on predefined criteria. For instance, route-maps can manipulate attributes like Local Preference

or MED for specific prefixes, enabling more nuanced routing decisions that better align with network objectives.

Throughout this guide, examples and scenarios illustrate practical applications of these BGP attributes and configuration options, highlighting common challenges and troubleshooting steps. These configurations ensure that internal and external routes are prioritized effectively, supporting reliable and optimized connectivity across complex network topologies. This guide serves as a reference for network administrators managing multi-homed environments, offering insights and best practices to maximize the stability and efficiency of BGP-enabled networks.

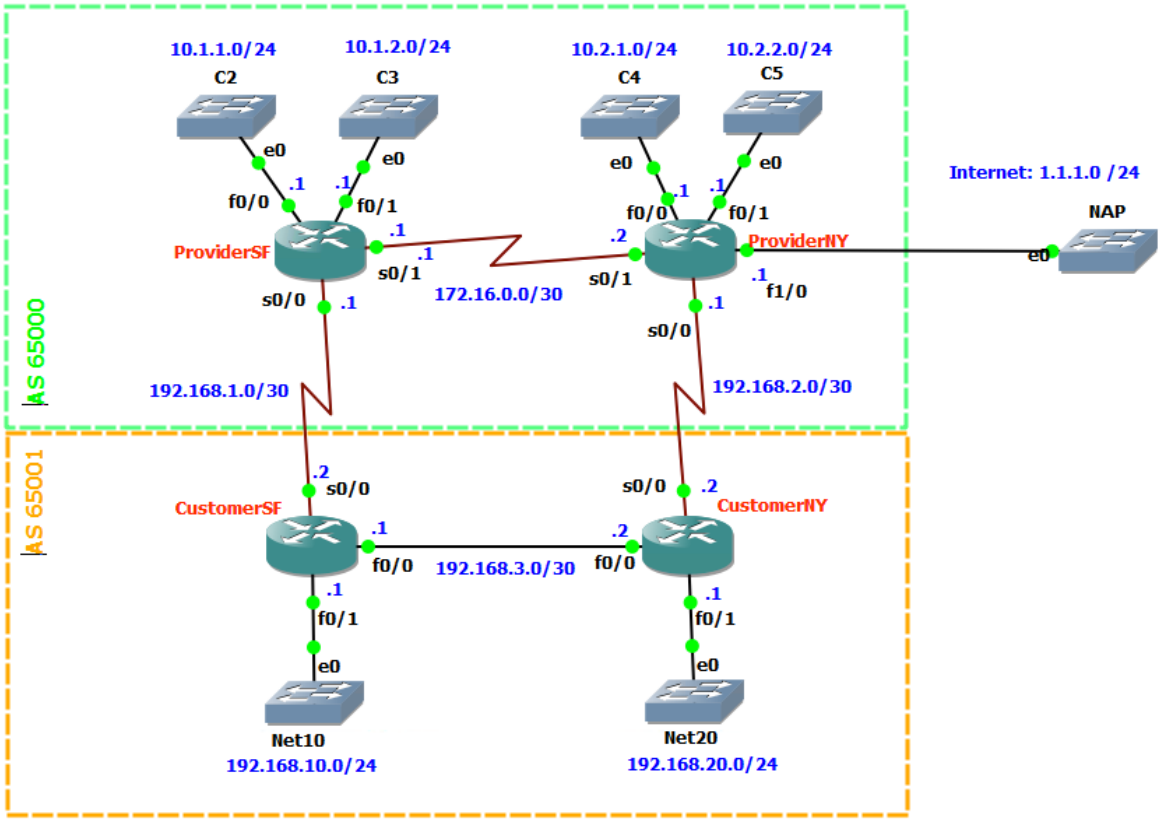
## GOALS

- To configure a BGP multi-homed network laveraging the **MED** and **Local Preference** attributes influding the BGP routing decisions.

# BGP CONFIGURATION GUIDE REPORT

## BGP CONFIGURATION GUIDE

### NETWORK DIAGRAM




### PROVIDER eBGP CONFIGURATIONS

Assumming the provider **WAN** has already been configured with **iBGP**, the next step is to configure **eBGP** to establish a TCP tunnel with the customer **WAN** to share network information:

- The **Provider_SF**

```
Provider_SF(config)#router bgp 65000
Provider_SF(config-router)#neighbor 192.168.1.2 remote-as 65001
```

- **ProviderNY**

```
Provider_NY(config)#router bgp 65000
Provider_NY(config-router)#neighbor 192.168.2.2 remote-as 65001
```

## CUSTOMER eBGP and iBGP CONFIGURATIONS

- All customer routers will be configured with **eBGP** and **iBGP** following the below reasoning:
  - **eBGP** so that the customer can learn prefixes from remote autonomous systems and its provider can advertise customer's network to the exterior.
  - **iBGP** since the customer **WAN** is implementing a multi-homed topology, meaning that external prefixes will be learned from different locations and local prefixes advertised from those locations, **iBGP** will enable a effective communication for this scenario.
- All customer routers will advertise **Net10** and **Net20** to their **BGP** peers by performing the following commands:

```
Cust1_SF(config)#router bgp 65001
Cust1_SF(config-router)#network 192.168.10.0

Cust1_NY(config)#router bgp 65001
Cust1_NY(config-router)#network 192.168.20.0
```

- As usual, a network administrator can input the below instructions to open a TCP tunnel with the defined **BGP** speakers, in this case, both routers will participate in both **eBGP** and **iBGP**:

```
Cust1_SF(config-router)#neighbor 192.168.1.1 remote-as 65000
Cust1_SF(config-router)#neighbor 192.168.3.2 remote-as 65001

Cust1_NY(config-router)#neighbor 192.168.2.1 remote-as 65000
Cust1_NY(config-router)#neighbor 192.168.3.1 remote-as 65001
```

The above configurations will generate reachability issues, these problems can be estated and explained with the following scenario:

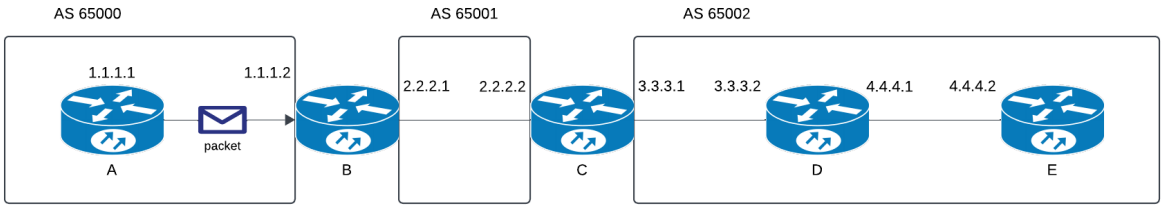



**Figure 1:** Figure 1

Suppose there exist a **BGP** network comformed by three autonomous systems. Router **A** begins to exchange network information with the rest of **BGP** speakers, one of the necessary parameters needed in **BGP** for routing decisions is the **next hop**, which in this case will be updated as follows:

- When router **B** is updated, its routing table will tell the router that the next hop to reach advertised blocks from router **A** will be **1.1.1.1**.
- When router **C** receives an update from router **B**, the **BGP** table of the router **C** will tell the router that any block advertised by **B** will be reached via **2.2.2.1**.
- When router **D** is updated, its routing table will say that any block advertised by **C** will be reacheble via **3.3.3.1**.
- Nonetheless, when router **D** updates router **E**, router **E** will contain the following information: any block advertised by router **D** is only reachable via **3.3.3.1**.

It can be noticed that **iBGP** did not change the **next-hop** parameter making communication between **E** and the exterior impossible. To avoid such issues in the practice topology, both provider and customer routers **must** implement the following command:

```
neighbor <neighbor> next-hop-self
```

To avoid reachability problems of this nature within *Figure 1*, a network administrator must specify on router **D** the command as follows:

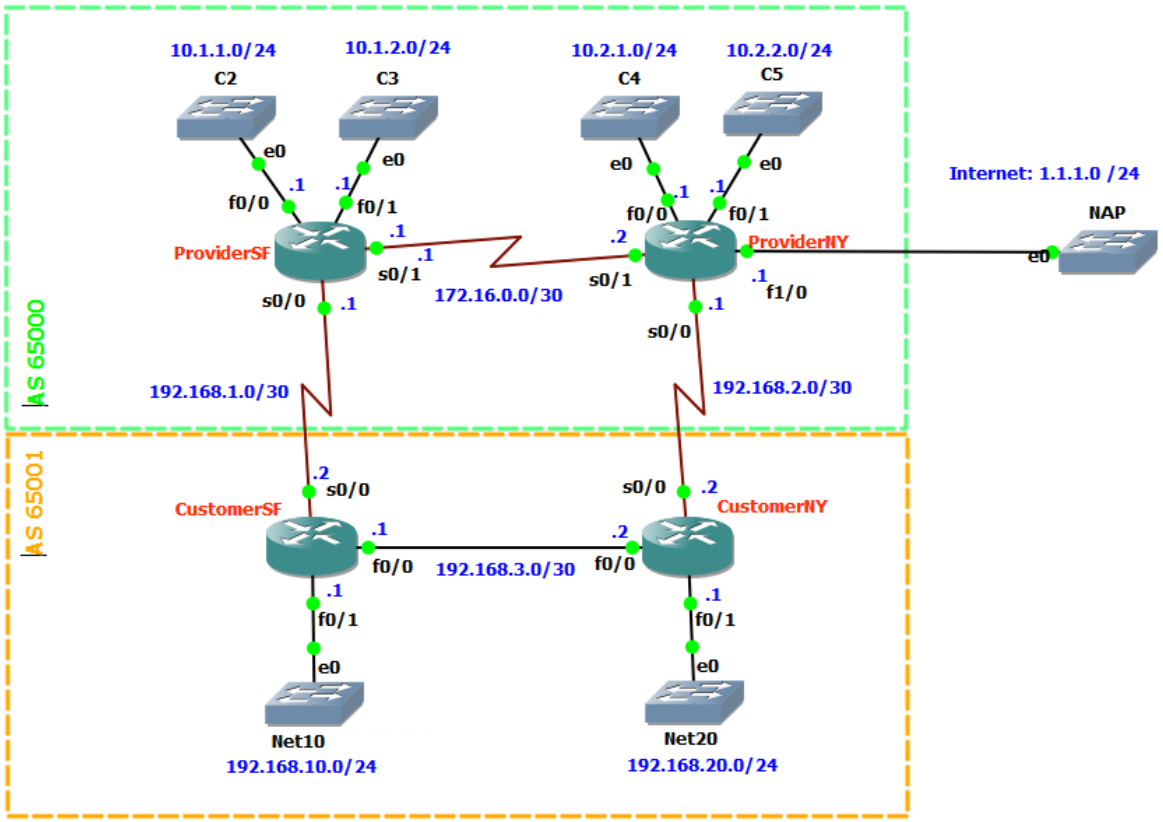
```
neighbor 4.4.4.2 next-hop-self
```

Forcing router **D** to modify the **next-hop** parameter to be set as itself, in the practice context, both provider and customer nodes have been configured as follows:

```
Cust1_SF(config-router)#neighbor 192.168.3.2 next-hop-self
Cust1_NY(config-router)#neighbor 192.168.3.1 next-hop-self
```

## CONFIGURATION GUIDE OF THE PREFIX-LIST, ROUTE-MAPS AND PREFERENCE LOCAL ATTRIBUTE

Another potential issue that must be denoted in the topology from this practice can be described using the following scenario:




In the topology case, **CustomerSF** has two options to route traffic from **Net10** to **C4**, either sending it via **ProviderSF** or via **CustomerNY**. It may not seem obvious in here but **CustomerNY** is indeed the best route for this case, let's state that reaching **C4** via **CustomerNY** requires less hops providing a faster connectivity. **CustomerSF** however, will not route packets to **CustomerNY** if trying to reach **C4** since **C4** route has been learned via **ProviderSF**, which is an external **BGP** peer and **any** route learned via external peers will have more preference over those learned from an internal peer. To modify this behaviour, the network administrator can use a *prefix-list* to assign rules to certain network prefixes, then, use a *route-map* to apply an *if-else* **BGP** condition and make a better routing decision for those packets where their network destiny correspond to the **ProviderNY** ones. This configurations can be done as follows:

- **Cust1_SF**

```
ip prefix-list C2 seq 5 permit 10.1.1.0/24
ip prefix-list C3 seq 5 permit 10.1.2.0/24
!
route-map EBGP-With-ProviderSF_IN permit 10
 match ip address prefix-list C2 C3
 set local-preference 300
route-map EBGP-With-ProviderSF_IN permit 20
 set local-preference 200
!
router bgp 65001
 neighbor 192.168.1.1 route-map EBGP-With-ProviderSF_IN in
```

The network administrator creates a *prefix-list* to match any destiny prefix that belongs to **C4** or **C5** prefixes, then a route map is applied to match **C2** and **C3** traffic setting a higher *local-preference* (higher ones are preferred) for the prefixes on the *prefix-list* and a lower *local-preference* for those networks that belongs to **ProviderNY**, finally, the access list is applied in the **BGP** process to tell **ProviderSF** to set a different local preference for certain prefixes that match so that we can have better routing decisions.

- **Cust1_NY** (same logic)

```
ip prefix-list C4 seq 5 permit 10.2.1.0/24
ip prefix-list C5 seq 5 permit 10.2.2.0/24
!
route-map EBGP-With-ProviderNY_IN permit 10
 match ip address prefix-list C4 C5
 set local-preference 300
route-map EBGP-With-ProviderNY_IN permit 20
 set local-preference 250
!
router bgp 65001
 neighbor 192.168.2.1 route-map EBGP-With-ProviderNY_IN in
```

Those configurations must be performed as well on the provider routers, nonetheless, the customer will not have access to them, however, still their routing decisions can be manipulated via the customer nodes, a network administrator can apply the very same logic but instead to be applied to learned routes, now it will be applied to advertised routes. The administrator can create another *prefix-list* to match certain blocks and assign then a lower or higher value to the **MED** attribute as follows:

- Example with **Cust1_SF:**

```
ip prefix-list Net10 seq 5 permit 192.168.10.0/24
!
route-map EBGP-With-ProviderSF_OUT permit 10
 match ip address prefix-list Net10
```

```
 set metric 200
route-map EBGP-With-ProviderSF_OUT permit 20
 set metric 300
!
router bgp 65001
 neighbor 192.168.1.1 route-map EBGP-With-ProviderSF_OUT out
```

Where the **MED** attribute will set a different metric to a certain path, manipulating how routing decisions should be made. The network administrator **must** be careful when configuring this as for any block that is not being contemplated will not have a metric, thus, not being reachable.

## TESTING CONFIGURATIONS

### BGP TABLES

```
CustomerSF#sh bgp
BGP table version is 10, local router ID is 192.168.10.1
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
* i1.1.1.0/24      192.168.3.2              0    200      0 65000 i
*>                 192.168.1.1                   200      0 65000 i
*> 10.1.1.0/24     192.168.1.1              0    300      0 65000 i
*> 10.1.2.0/24     192.168.1.1              0    300      0 65000 i
*>i10.2.1.0/24     192.168.3.2              0    300      0 65000 i
*                  192.168.1.1                   200      0 65000 i
*>i10.2.2.0/24     192.168.3.2              0    300      0 65000 i
*                  192.168.1.1                   200      0 65000 i
*> 192.168.10.0    0.0.0.0                  0         32768 i
*>i192.168.20.0    192.168.3.2              0    100      0 i
```

```
ProviderSF#sh bgp
BGP table version is 8, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i1.1.1.0/24      172.16.0.2               0    100      0 i
*> 10.1.1.0/24     0.0.0.0                  0         32768 i
*> 10.1.2.0/24     0.0.0.0                  0         32768 i
*>i10.2.1.0/24     172.16.0.2               0    100      0 i
*>i10.2.2.0/24     172.16.0.2               0    100      0 i
*> 192.168.10.0    192.168.1.2            200             0 65001 i
*  192.168.20.0    192.168.1.2            300             0 65001 i
*>i                172.16.0.2             200    100      0 65001 i
```

```
ProviderNY#sh bgp
BGP table version is 8, local router ID is 192.168.2.1
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
*> 1.1.1.0/24       0.0.0.0                  0         32768 i
*>i10.1.1.0/24      172.16.0.1               0    100      0 i
*>i10.1.2.0/24      172.16.0.1               0    100      0 i
*> 10.2.1.0/24      0.0.0.0                  0         32768 i
*> 10.2.2.0/24      0.0.0.0                  0         32768 i
*  192.168.10.0     192.168.2.2            250             0 65001 i
*>i                 172.16.0.1             200    100      0 65001 i
*> 192.168.20.0     192.168.2.2            200             0 65001 i
```

```
CustomerNY#sh bgp
BGP table version is 8, local router ID is 192.168.20.1
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
* i1.1.1.0/24       192.168.3.1              0    200      0 65000 i
*>                  192.168.2.1              0    200      0 65000 i
*  10.1.1.0/24      192.168.2.1                   200      0 65000 i
*>i                 192.168.3.1              0    300      0 65000 i
*  10.1.2.0/24      192.168.2.1                   200      0 65000 i
*>i                 192.168.3.1              0    300      0 65000 i
*> 10.2.1.0/24      192.168.2.1              0    300      0 65000 i
*> 10.2.2.0/24      192.168.2.1              0    300      0 65000 i
*>i192.168.10.0     192.168.3.1              0    100      0 i
*> 192.168.20.0     0.0.0.0                  0         32768 i
```

**PING TO NAP**

```
CustomerSF#ping 1.1.1.1 source f0/0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.10.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/30/40 ms
```

**Figure 2:** ping from Net10 interface

```
CustomerSF#ping 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

**Figure 3:** ping with no source parameter

## CONCLUSIONS

**Juarez Mota Daniel Alejandro:** This practice helped me to understand in deep how the **BGP** protocol uses certain attributes and parameters to influde in routing decisions and that by default, **BGP** does not consider certain situations where some routes might be better than others or some specific design problems. I also learned the practical difference between **iBGP** and **eBGP** and how this infludes when it comes to information exchange between **BGP** speakers. I consider **BGP** concepts to be very intuitive so I did not have any issues when understanding the practice configurations, I consider this practice to be quite useful as it helped me a lot to understand **BGP** better and its importance for external routing.

**Rios Gomez Jose Enrique:** Completing this BGP multi-homing project has been a rewarding experience that has deepened my understanding of advanced routing concepts. The hands-on nature of configuring BGP on various routers helped solidify my knowledge and provided me with valuable insights into the practical applications of these concepts. I enjoyed overcoming challenges like establishing proper routing relationships and optimizing route selections through Local Preference and MED adjustments. This experience has not only bolstered my technical skills but has also reinforced my passion for networking and the continuous learning that comes with it. I look forward to applying what I've learned in future projects and further expanding my knowledge in this field.

## REFERENCES


- "GNS3 Lab: BGP - Multihoming to a Single Provider with Partial Routing." 2009. Pierky's Blog. May 10, 2009. https://blog.pierky.com/gns3-lab-bgp-multihoming-to-a-single-provider-with-partial-routing/.
- "Understand BGP MED Attribute." 2024. Cisco. February 2024. https://www.cisco.com/c/en/us/support/docs/ip/bo gateway-protocol-bgp/217973-understand-bgp-med-attribute.html.