PRACTICE REPORT, BGP CONFIGURATION GUIDE

ADVANCED ROUTING

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

2024-10-24

Contents

EXECUTIVE REPORT

PROJECT OVERVIEW

Border Gateway Protocol (BGP) is an **Exterior Gateway Protocol (EGP)** that enables routing between different autonomous systems (ASes) in the broader Internet or large networks. To exchange routing information, BGP nodes establish a **TCP connection** on port 179 with each peer. This TCP connection, often referred to as a "BGP session," can either be **External BGP (eBGP)** or **Internal BGP (iBGP)**, depending on the relationship between the connected ASes. If the TCP session connects routers from **different autonomous systems**, the BGP session is considered **eBGP**. Conversely, if the routers belong to the **same autonomous system**, it's classified as **iBGP**. In this lab, **Team 5** configured a BGP topology involving **nine routers** across **three autonomous systems** to establish inter-AS connectivity. Through this setup, we gained practical experience in establishing BGP sessions, managing autonomous systems, and achieving route propagation across different ASes.

GOALS

- Deploy a basic **BGP** configuration to enable connectivity between three different Autonomous Systems

BGP CONFIGURATION GUIDE REPORT

BGP CONFIGURATION GUIDE

NETWORK DIAGRAM

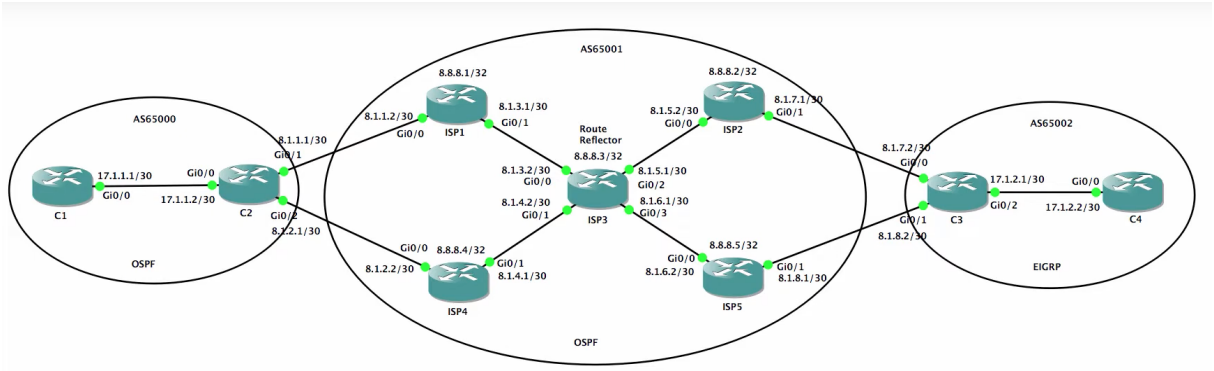

Figure 1: Network Diagram

ASSUMPTIONS

- IGP protocols (OSPF and EIGRP) have been already configured and converged successfully in their respective **AS**.
- In the **C2** router it has been declared the following command to advertise a default route:

```
default-information originate always
```

- in the **C3** it has been advertised as well a default route:

```
ip summary-address eigrp 1 0.0.0.0 0.0.0.0 1
```

CONFIGURATION GUIDE FOR eBGP SPEAKERS

- Enable the **BGP** process on the router:

```
router bgp {AS}
```

- Manually set-up connectivity between **BGP** peers. The following command will try to open the **TCP** connection with another **BGP** speaker. The `remote-as` parameter does not necessary need to be a different autonomous system:

```
neighbor {neighbor_ip} remote-as {neighbor_as}
```

- Once a peer has been declared and connected, **BGP** will only advertise whatever the network administrator specifies with the `network` command if and only if the advertised network is in the routing table. If the desired network to advertise is found in the routing table, then the router will add a new entry in its **BGP** table to start advertising it:

```
network {network_to_advertise} mask {mask}
```

The above was applied to our diagram as follows:

- **c2** (notice the **weight** attribute to prefer **ISP1** over **ISP4**)

```
router bgp 65000
network 17.1.1.0 mask 255.255.255.252
neighbor 8.1.1.2 remote-as 65001
neighbor 8.1.1.2 weight 100
neighbor 8.1.2.2 remote-as 65001
```

- **c3** (notice the **weight** attribute to prefer **ISP2** over **ISP5**)

```
router bgp 65002
network 17.1.2.0 mask 255.255.255.252
neighbor 8.1.7.1 remote-as 65001
neighbor 8.1.8.1 weight 100
neighbor 8.1.8.1 remote-as 65001
```

- **ISP1**

```
router bgp 65001
network 8.1.1.0 mask 255.255.255.252
network 8.1.3.0 mask 255.255.255.252
network 8.8.8.1 mask 255.255.255.255
neighbor 8.1.1.1 remote-as 65000
neighbor 8.8.8.3 remote-as 65001
```

- **ISP2**

```
outer bgp 65001
network 8.1.5.0 mask 255.255.255.252
network 8.1.7.0 mask 255.255.255.252
network 8.8.8.2 mask 255.255.255.255
neighbor 8.1.7.2 remote-as 65002
neighbor 8.8.8.3 remote-as 65001
```

- **ISP4**

```
router bgp 65001
network 8.1.2.0 mask 255.255.255.252
network 8.1.4.0 mask 255.255.255.252
network 8.8.8.4 mask 255.255.255.255
neighbor 8.1.2.1 remote-as 65000
neighbor 8.8.8.3 remote-as 65001
```

- **ISP5**

```
router bgp 65001
network 8.1.6.0 mask 255.255.255.252
network 8.1.8.0 mask 255.255.255.252
network 8.8.8.5 mask 255.255.255.255
neighbor 8.1.8.2 remote-as 65002
neighbor 8.8.8.3 remote-as 65001
neighbor 8.1.6.1 remote-as 65001
```

CONFIGURATION GUIDE FOR iBGP SPEAKERS WITH A LOOPBACK INTERFACE

To configure an **iBGP** speaker with a loopback interface it is required to do the same steps as above but adding the following command:

```
neighbor {neighbor_ip} update-source {loopack_interface}
```

The above command will enable the loopback interface to be the source of information exchange between the specified **BGP** peer, it was applied in this practice as follows:

- **ISP1**

```
neighbor 8.8.8.3 update-source loopback 1
```

- **ISP2**

```
neighbor 8.8.8.3 update-source loopback 1
```

- **ISP3**

```
neighbor 8.8.8.1 update-source loopback 1
neighbor 8.8.8.2 update-source loopback 1
neighbor 8.8.8.4 update-source loopback 1
neighbor 8.8.8.5 update-source loopback 1
```

- **ISP4**

```
neighbor 8.8.8.3 update-source loopback 1
```

- **ISP5**

```
neighbor 8.8.8.3 update-source loopback 1
```

CONFIGURATION GUIDE OF A ROUTE REFLECTOR

A router reflector is required to solve some issues **BGP** has with *fully-meshed* networks. Since **BGP** is a *loop-free* protocol, updates within *fully-meshed* networks will be hard to achieve, in such case, a route reflector is needed to act as a route forwarder and anaunce route to other **BGP** speakers as if they were on a *fully-meshed* network. To configure a route reflector, the network administrator has to define a peer and declare the *update-source* loopback interface, plus, specifying that certain peer will be a *route reflector* client with the following command:

```
neighbor {neighbor-ip} route-reflector-client
```

In the context of this practice, it was applied as follows:

```
router bgp 65001
neighbor 8.8.8.1 remote-as 65001
neighbor 8.8.8.1 update-source loopback 1
neighbor 8.8.8.1 route-reflector-client
neighbor 8.8.8.2 remote-as 65001
neighbor 8.8.8.2 update-source loopback 1
neighbor 8.8.8.2 route-reflector-client
neighbor 8.8.8.4 remote-as 65001
neighbor 8.8.8.4 update-source loopback 1
neighbor 8.8.8.4 route-reflector-client
neighbor 8.8.8.5 remote-as 65001
neighbor 8.8.8.5 update-source loopback 1
neighbor 8.8.8.5 route-reflector-client
```

TESTING CONFIGURATIONS

ROUTE TRACING BETWEEN CLIENTS C1 AND C4

```
C1#ping 17.1.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 17.1.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 52/73/88 ms
C1#trace 17.1.2.2

Type escape sequence to abort.
Tracing the route to 17.1.2.2

  1 17.1.1.2 28 msec 24 msec 12 msec
  2 8.1.1.2 40 msec 20 msec 32 msec
  3 8.1.3.2 28 msec 64 msec 52 msec
  4 8.1.5.2 56 msec 64 msec 56 msec
  5 8.1.7.2 32 msec 64 msec 88 msec
  6 17.1.2.2 84 msec 84 msec *
C1#
```

Figure 2: successfully connected with C4

```
C4#ping 17.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 17.1.1.1, timeout is 2 se
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max =
C4#trace 17.1.1.1

Type escape sequence to abort.
Tracing the route to 17.1.1.1

  1 17.1.2.1 56 msec 76 msec 72 msec
  2
*Mar  1 00:03:42.007: %CDP-4-DUPLEX_MISMATCH: duplex mismat
FastEthernet1/0 (full duplex).8.1.7.1 76 msec 76 msec 108 m
  3 8.1.5.1 144 msec 136 msec 76 msec
  4 8.1.3.1 76 msec 140 msec 140 msec
  5 8.1.1.1 156 msec 160 msec 148 msec
  6
*Mar  1 00:04:13.299: %CDP-4-DUPLEX_MISMATCH: duplex mismat
FastEthernet1/0 (full duplex).17.1.1.1 196 msec 204 msec *
C4#
```

Figure 3: successfully connected with C1

CLIENTS C2 AND C3 BGP ROUTING TABLES

```
   Network          Next Hop            Metric LocPrf Weight Path
r  8.1.1.0/30       8.1.2.2                                0 65001 i
r>                  8.1.1.2                  0           100 65001 i
r> 8.1.2.0/30       8.1.1.2                              100 65001 i
r                   8.1.2.2                  0             0 65001 i
*  8.1.3.0/30       8.1.2.2                                0 65001 i
*>                  8.1.1.2                  0           100 65001 i
*> 8.1.4.0/30       8.1.1.2                              100 65001 i
*                   8.1.2.2                  0             0 65001 i
*> 8.1.5.0/30       8.1.1.2                              100 65001 i
*                   8.1.2.2                                0 65001 i
*  8.1.6.0/30       8.1.2.2                                0 65001 i
*>                  8.1.1.2                              100 65001 i
*> 8.1.7.0/30       8.1.1.2                              100 65001 i
*                   8.1.2.2                                0 65001 i
*  8.1.8.0/30       8.1.2.2                                0 65001 i
*>                  8.1.1.2                              100 65001 i
*  8.8.8.1/32       8.1.2.2                                0 65001 i
   Network          Next Hop            Metric LocPrf Weight Path
*>                  8.1.1.2                  0           100 65001 i
*> 8.8.8.2/32       8.1.1.2                              100 65001 i
*                   8.1.2.2                                0 65001 i
*> 8.8.8.4/32       8.1.1.2                              100 65001 i
*                   8.1.2.2                  0             0 65001 i
*  8.8.8.5/32       8.1.2.2                                0 65001 i
*>                  8.1.1.2                              100 65001 i
*> 17.1.1.0/30      0.0.0.0                  0         32768 i
*  17.1.2.0/30      8.1.2.2                                0 65001 65002 i
*>                  8.1.1.2                              100 65001 65002 i
```

Figure 4: C2 BGP routing table

```
   Network          Next Hop            Metric LocPrf Weight Path
*  8.1.1.0/30       8.1.7.1                                0 65001 i
*>                  8.1.8.1                                0 65001 i
*  8.1.2.0/30       8.1.7.1                                0 65001 i
*>                  8.1.8.1                                0 65001 i
*  8.1.3.0/30       8.1.7.1                                0 65001 i
*>                  8.1.8.1                                0 65001 i
*  8.1.4.0/30       8.1.7.1                                0 65001 i
*>                  8.1.8.1                                0 65001 i
*  8.1.5.0/30       8.1.8.1                                0 65001 i
*>                  8.1.7.1                  0             0 65001 i
*  8.1.6.0/30       8.1.7.1                                0 65001 i
*>                  8.1.8.1                  0             0 65001 i
r  8.1.7.0/30       8.1.8.1                                0 65001 i
r>                  8.1.7.1                  0             0 65001 i
r  8.1.8.0/30       8.1.7.1                                0 65001 i
r>                  8.1.8.1                  0             0 65001 i
*  8.8.8.1/32       8.1.7.1                                0 65001 i
   Network          Next Hop            Metric LocPrf Weight Path
*>                  8.1.8.1                                0 65001 i
*  8.8.8.2/32       8.1.8.1                                0 65001 i
*>                  8.1.7.1                  0             0 65001 i
*  8.8.8.4/32       8.1.7.1                                0 65001 i
*>                  8.1.8.1                                0 65001 i
*  8.8.8.5/32       8.1.7.1                                0 65001 i
*>                  8.1.8.1                  0             0 65001 i
*  17.1.1.0/30      8.1.8.1                                0 65001 65000 i
*>                  8.1.7.1                                0 65001 65000 i
*> 17.1.2.0/30      0.0.0.0                  0         32768 i
```

Figure 5: C3 BGP routing table

CONCLUSIONS

Juarez Mota Daniel Alejandro: Trough this practice I gained knoweledge regarding to the **BGP** basics such as the need of this protocol, configurations and usage cases. I learned this protocol is implemented on the internet given its amount of configurations a network administrator can use on it. I learned how autonomous systems gain communication with each other and why **BGP** is significantly more efficient as an external protocol rather than being an internal one. I learned as well the importance of a route reflector and the issues presented on a *fully-meshed* network scenario when trying to implement **BGP**. In my case it was quite difficult to understand how **BGP** works since it differs a lot from **IGP** protocols such as **OSPF** or **EIGRP**.

Rios Gomez Jose Enrique: I found this practice highly engaging, as it provided foundational insights

into the internet's underlying structure and operations. However, it was somewhat challenging, given that BGP involves a complex set of 13 attributes to manage and is known for its slower convergence. Despite these aspects, it was fascinating to explore an alternative approach for integrating two routing protocols without relying on the redistribution command.

REFERENCES

- "Sample Configuration for IBGP and EBGP with or without a Loopback Address." n.d. Cisco. https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13751-23.html.
- "IP Routing: BGP Configuration Guide - Configuring Internal BGP Features [Cisco ASR 1000 Series Aggregation Services Routers]." n.d. Cisco. https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-16/irg-xe-16-book/configuring-internal-bgp-features.html.
- "Understand the Importance of BGP Weight Path Attribute." n.d. Cisco. https://www.cisco.com/c/en/us/support/do gateway-protocol-bgp/213285-understand-the-importance-of-bgp-weight.html.