Troubleshoot Border Gateway Protocol Basic Issues

Contents

Introduction

This document describes how to troubleshoot the most common issues with the Border Gateway Protocol (BGP) and provides basic solutions and guidelines.

Prerequisites

Requirements

There are no specific prerequisites for this document. Basic BGP protocol knowledge is useful, you can refer to the BGP Configuration Guide for more information.

Components used

This document is not restricted to specific software and hardware versions, but commands are applicable for Cisco IOS® and Cisco IOS® XE.

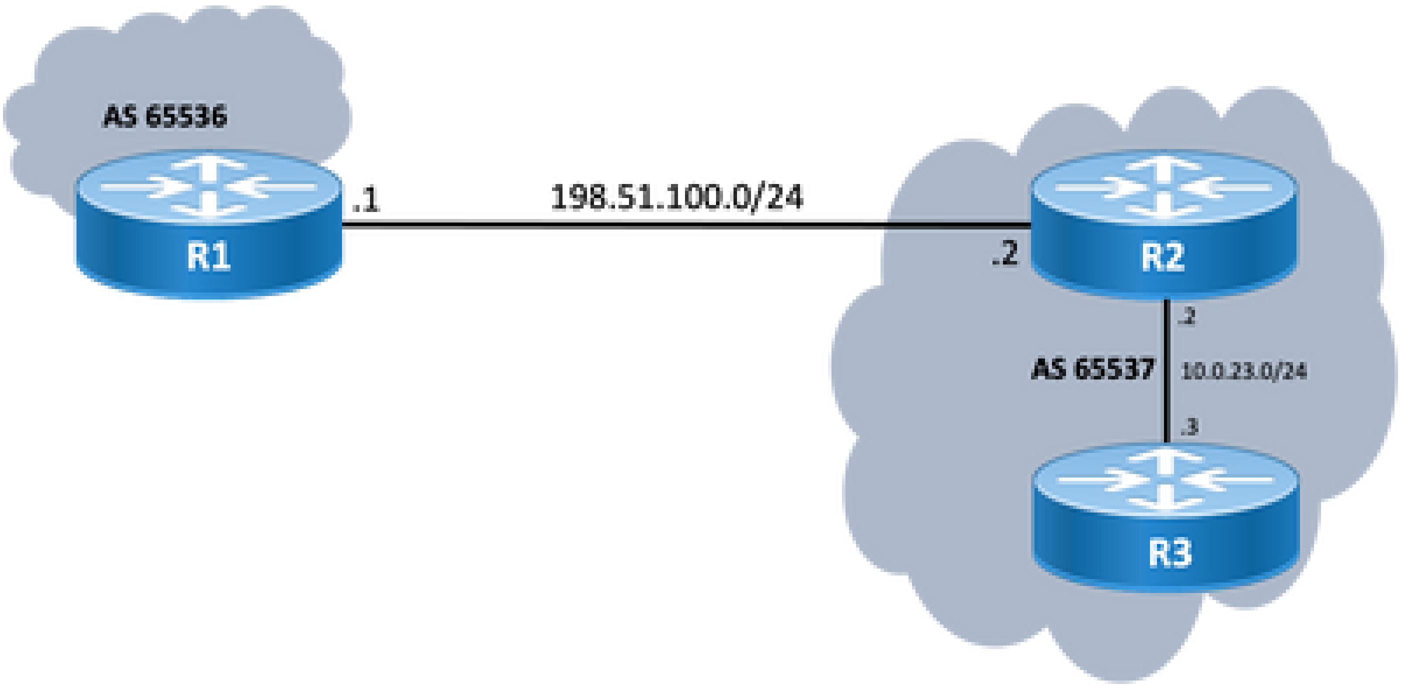
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

This document describes a basic guide to troubleshoot the most common issues in Border Gateway Protocol (BGP), gives corrective actions, useful commands/debugs to detect the root cause of the problems, and best practices to avoid potential issues. Keep in mind that all possible variables and scenarios cannot be considered and a deeper analysis could be required by Cisco TAC.

Topology

Use this topology diagram as a reference for the outputs provided in this document.



Scenarios and Problems

Adjacency Down

If a BGP session is down and does not come up, issue the show ip bgp all summary command. Here you can find the current status of the session:

- If the session is not up state, it can vary between IDLE and ACTIVE (depends on the Finite State Machine process).
- If session is up, you see the number of prefixes received.

```
<#root>

R2#

show ip bgp all summary
```

```
For address family: IPv4 Unicast
BGP router identifier 198.51.100.2, local AS number 65537
BGP table version is 19, main routing table version 19
18 network entries using 4464 bytes of memory
18 path entries using 2448 bytes of memory
1/1 BGP path/bestpath attribute entries using 296 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 7208 total bytes of memory
BGP activity 18/0 prefixes, 18/0 paths, scan interval 60 secs
18 networks peaked at 11:21:00 Jun 30 2022 CST (00:01:35.450 ago)

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.0.23.3       4        65537       6       5       19    0    0 00:01:34       18

198.51.100.1    4        65536       0       0        1    0    0 never    Idle
```

No Connectivity

The first requirement that has to be ensured is the connectivity between both peers so TCP session on port 179 can be established. Either they are directly connected or not. A simple ping is useful for this matter. If peering is established between loopback interfaces, a loopback to loopback ping must be done. If a ping test is performed without specific loopback as the source interface, the outgoing physical interface IP address is used as the packet's source IP address instead of the router's loopback IP address.

If ping is not successful, consider these causes:

- No connected route peer or no route at all: show ip route peer_IP_address can be used.
- Layer 1 issue: physical interface, SFP (connector), cable or external issue (transport and provider if applicable) needs to be considered.
- Check any firewall or access lists which can block connection.

If ping is successful, consider this:

Configuration Issues

- Wrong IP address or AS configured: For wrong IP address, there is no such message displayed but ensure proper configuration is done. For wrong AS, you must see a message like with the show logging command.

```
%BGP-3-NOTIFICATION: sent to neighbor 198.51.100.1 passive 2/2 (peer in wrong AS) 2 bytes 1B39
```

Check BGP configuration on both ends to correct AS numbers or peer IP address.

- Duplicate router ID:

```
%BGP-3-NOTIFICATION: sent to neighbor 198.51.100.1 passive 2/3 (BGP identifier wrong) 4 bytes 0A0A0A0A
```

Check BGP identifier on both ends via show ip bgp all summary and correct the duplicate issue. This can be achieved manually with global command bgp router-id X.X.X.X under bgp router configuration. As a best practice, ensure router ID is set manually to unique number.

- BGP source and TTL:

Most of the iBGP sessions are configured over the loopback interfaces reachable via an IGP. This loopback interface must be explicitly defined as the source, Do this with the command neighbor ip-address update-source interface-id .

For eBGP peer, directly connected interfaces are usually used for peering, and there is a check for Cisco IOS/Cisco IOS XE to fulfill this purpose, or it does not even try to establish session. If eBGP is tried from loopback to loopback on directly connected routers, this check can be disabled for a specific neighbor on both ends via neighbor ip-address disable-connected-check .

However, if there are multiple hops between the eBGP peers, a proper hop count is required, ensure the **neighbor ip-address ebgp-multihop [hop-count]**is configured with the correct hop count so session can be established.

If the hop-count is not specified, the default TTL value for iBGP sessions is 255, while the default TTL value for eBGP sessions is 1.

TCP Session Issues

A useful action to test port 179 is a manual telnet from one peer to the other:

```
<#root>

R1#

telnet 198.51.100.2 179

Trying 198.51.100.2, 179 ... Open

[Connection to 198.51.100.2 closed by foreign host]
```

Either open/connection closed, or connection refused by remote host indicates packets reach remote end, then, ensure there are no problems with control plane at far end. Otherwise, if there is a Destination unreachable, check any firewall or access lists which can block TCP port 179, or BGP packets, or any packet loss on the path.

In case of authentication problem, the messages you can see:

```
%TCP-6-BADAUTH: Invalid MD5 digest from 198.51.100.1(179) to 198.51.100.2(20062) tableid - 0
%TCP-6-BADAUTH: No MD5 digest from 198.51.100.1(179) to 198.51.100.2(20062) tableid - 0
```

Check authentication methods, password and related configuration, and to further troubleshoot refer to MD5 Authentication Between BGP Peers Configuration Example.

If the TCP session does not come up, you can use the next commands for isolation:

```
show tcp brief all
show control-plane host open-ports
debug ip tcp transactions
```

Adjacency Bounces

If session is up and down, look for show log and you can see some scenarios.

Interface Flap

```
%BGP-5-ADJCHANGE: neighbor 198.51.100.2 Down Interface flap
```

As message indicates, reason for this failure is the interface down situation, look for any physical issues on port/SFP, cable or disconnections.

Hold Timer Expired

```
%BGP-3-NOTIFICATION: sent to neighbor 198.51.100.2 4/0 (hold time expired) 0 bytes
```

It is a very common situation; it means that router did not receive or process a keepalive message or any update message before the hold timer expired. Device sends a notification message and closes the session. The most commons reasons for this issue are listed here:

- Interface issues: Look for any input errors, input queue drops or physical issues on both peers' connected interfaces; show interface can be used for this purpose.
- Packet loss in transit: Sometimes, Hello packets can be dropped in transit, the best way to ensure this is a packet capture at interface level.
    - You can use Embedded Packet Capture on Cisco IOS and Cisco IOS XE devices.
  - In case packets are seen at interface level, you need to be sure they reach control plane, EPC on control plane, or debug bgp [vrf name] ipv4 unicast keepalives is useful.
- High CPU: A high CPU condition can cause drops on the control plane, show processes cpu [sorted|history] is useful to identify problem. Based on the platform, you can find the next step to troubleshoot with the CPU Reference document
- CoPP policy issues: Troubleshoot methodology varies for each platform and is out of scope for this document.

- MTU mismatch: If there are MTU discrepancies in the path, and if ICMP messages are blocked in the path from source to destination, PMTUD does not function and can result in session flap. Updates are sent with the negotiated MSS value and a DF bit set. If a device in the path or even the destination is not able to accept the packets with higher MTU, it sends an ICMP error message back to BGP speaker. The destination router either waits for the BGP keepalive or BGP update packet to update its hold down timer.

  - You can check the MSS negotiated with show ip bgp neighbors ip_address.

A Ping test to a specific neighbor with df set can show you if such MTU is valid along the path:

<#root>

```
ping 198.51.100.2 size
 max_seg_size
df
```

If MTU issues are found, an accurate review of the configuration must be done to ensure that the MTU values are consistent throughout the network.

Note: For more information on MTU, refer to [BGP Neighbor Flaps with MTU Troubleshooting].

AFI/SAFI Issues

```
%BGP-5-ADJCHANGE: neighbor 198.51.100.2 passive Down AFI/SAFI not supported
%BGP-3-NOTIFICATION: received from neighbor 198.51.100.2 active 2/8 (no supported AFI/SAFI) 3 bytes 0000
```

Address-Family Identifier (AFI) is a capability extension added by Multi-Protocol BGP (MP-BGP). It correlates to a specific network protocol, such as IPv4, IPv6, and the like, and additional granularity through a Subsequent Address-Family Identifier (SAFI), such as unicast and multicast. MBGP achieves this separation by BGP path attributes (PAs) MP_REACH_NLRI and MP_UNREACH_NLRI. These attributes are carried inside BGP update messages and are used to carry network reachability information for different address families.

The message gives you the numbers of these AFI/SAFI registered by IANA:

- [IANA Address Family Numbers]
- [Subsequent Address Family Identifiers (SAFI) Parameters]

- Check BGP configuration for the address-families intended on both sides to correct any undesired address families.
- Use neighbor ip-address dont-capability-negotiate on both ends. For further information, refer to [Unsupported Capabilities Cause BGP Peer Malfunction.]

Path Install and Selection

For a better explanation about how BGP works, and to select best path, refer to [BGP Best Path Selection Algorithm.]

Next Hop

For a route to be installed into our routing table, next hop needs to be reachable, otherwise, even if prefix is on our Loc-RIB BGP table, it does not get into RIB. As a loop avoidance rule, on Cisco IOS/Cisco IOS XE, iBGP does not change next hop attribute and leaves AS_PATH alone while eBGP rewrites next hop and prepends its AS_PATH.

You can check next hop with show ip bgp [prefix]. It gives you the next hop and inaccessible word. In the example, this is a prefix announced by R1 via eBGP to R2 and learnt by R3 via iBGP connection from R2.

<#root>

```
R3#

show ip bgp 192.0.2.1

BGP routing table entry for 192.0.2.1/32, version 0

Paths: (1 available, no best path)

  Not advertised to any peer
  Refresh Epoch 1
  65536


198.51.100.1 (inaccessible)

 from 10.0.23.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal
      rx pathid: 0, tx pathid: 0
      Updated on Jul 1 2022 13:44:19 CST
```

On the output, next hop is the outgoing interface of R1 which is not known by R3. In order to fix this situation either you can advertise next-hop via IGP, static route or use the neighbor ip-address next-hop-self command on iBGP peer to modify the next-hop IP (which is directly connected). On diagram example, this configuration needs to be on R2; the neighbor towards R3 (neighbor 10.0.23.3 next-hop-self).

As a result, next hop changes (after a clear ip bgp 10.0.23.2 soft) to directly connected interface (reachable) and prefix is installed.

<#root>

```
R3#

show ip bgp 192.0.2.1

BGP routing table entry for 192.0.2.1/32, version 24

Paths: (1 available, best #1, table default)

  Not advertised to any peer
  Refresh Epoch 1
  65536


10.0.23.2
```

```
 from 10.0.23.2 (10.2.2.2)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      rx pathid: 0, tx pathid: 0x0
      Updated on Jul 1 2022 13:46:53 CST
```

RIB Failure

This happens when route cannot be installed into the Global RIB, which results in a RIB failure. Common reason is when same prefix is already on RIB for another routing protocol with lower administrative distance, but the exact reason for a RIB failure is seen with the command **show ip bgp rib-failure**. For deeper explanation, you can consult this link:

> **Note**: You can identify and correct such issue as explained in Understand BGP RIB-failure and The Command bgp suppress-inactive.

Race Condition

The most common issue seen is when IGP is preferred over eBGP on mutual redistribution scenario. When an IGP route is redistributed into BGP, it is considered locally generated by BGP and gets a weight of 32768 by default. All prefixes received from a BGP peer are assigned a local weight of 0 by default. Therefore, if the same prefix must be compared, the prefix with the higher weight is installed in the routing table based on the BGP best path selection process and this is why IGP route is installed on RIB.

The solution for this problem, is to set a higher weight for all routes received from the BGP peer under router bgp configuration:

```
<#root>

neighbor

ip-address

 weight 40000
```

> **Note**: For a detailed explanation, refer to Understand the Importance of BGP Weight Path Attribute in Network Failover Scenarios.

Other Issues

BGP Slow Peer

It is a peer that cannot keep up with the rate at which the sender generates update messages. There are many reasons for a peer to exhibit this problem; high CPU in one of the peers, excess traffic or traffic loss on a link, bandwidth resource, among others.

> **Note**: To help identify and correct slow peers issues, refer to Use the BGP "Slow Peer" Feature to Resolve Slow Peer Issues.

Memory Issues

BGP uses memory that is assigned to the Cisco IOS process to maintain network prefixes, best paths, polices and all related configuration to operate properly. The overall processes are seen with command show processes memory sorted:

<#root>

```
R1#

show processes memory sorted
Processor Pool Total: 2121414332 Used:  255911152 Free: 1865503180

reserve P Pool Total:     102404 Used:         88 Free:     102316
 lsmpi_io Pool Total:    3149400 Used:    3148568 Free:        832

 PID TTY  Allocated      Freed

Holding

    Getbufs    Retbufs Process
   0   0  266231616   81418808  160053760          0          0 *Init*
 662   0   34427640      51720   34751920          0          0 SBC main process
  85   0    9463568          0    8982224          0          0 IOSD ipc task
   0   0   34864888   25213216    8513400    8616279          0 *Dead*
 504   0     696632          0     738576          0          0 QOS_MODULE_MAIN
 518   0     940000       8616

    613760

       0          0

 BGP Router


 228   0     856064     345488     510080          0          0 mDNS
  82   0     547096     118360     417520          0          0 SAMsgThread
   0   0          0          0     395408          0          0 *MallocLite*
```

Processor pool is the memory used; around 2.1 GB in the example. Next, you must look at the Holding column to identify the sub-process holding most of it. Then, you need to check the BGP sessions you have, how many routes are received, and configuration used.

Common steps to reduce memory holding by BGP:

- BGP filtering: If it is not necessary to receive a full BGP table, use policies to filter routes and install only the prefixes you need.
- Soft reconfiguration: Look for **neighbor ip_address soft-reconfiguration inbound** under BGP configuration; this command allows you to see all prefixes received before any inbound policy (Adj-RIB-in). However, this table needs around half of the current BGP Local RIB table to store this information so you can avoid this configuration unless it is compulsorily required, or your current prefixes are few.

Note: For further information on how to optimize BGP refer to Configure BGP Routers for Optimal Performance and Reduced Memory Consumption.

High CPU

Routers use different processes for BGP to operate. To verify the BGP process is the cause of high CPU utilization, use the show process cpu sorted command.

<#root>

```
R3#

show processes cpu sorted


CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%
 PID Runtime(ms)     Invoked      uSecs    5Sec   1Min   5Min TTY Process
 PID Runtime(ms)     Invoked      uSecs    5Sec   1Min   5Min TTY Process
 163          36        1463         24  0.07%  0.00%  0.00%   0 ADJ background
  62          28         132        212  0.07%  0.00%  0.00%   0 Exec
   2          39         294        132  0.00%  0.00%  0.00%   0 Load Meter
   1           0           4          0  0.00%  0.00%  0.00%   0 Chunk Manager
   3          27        1429         18  0.00%  0.00%  0.00%   0

BGP Scheduler


   4           0           1          0  0.00%  0.00%  0.00%   0 RO Notify Timers
  63           4          61         65  0.00%  0.00%  0.00%   0

BGP I/O


  83         924          26      35538  0.00%  0.03%  0.04%   0

BGP Scanner


  96         142       11651         12  0.00%  0.00%  0.00%   0 Tunnel BGP
   7           0           1          0  0.00%  0.00%  0.00%   0 DiscardQ Backgro
```

Here are the common processes, causes, and general steps to overcome high CPU utilization due to BGP:

- BGP Router: Runs once per second to safeguard faster convergence. It is one of the most important threads. It reads the bgp update messages, validates the prefixes/networks and attributes, updates the per AFI/SAFI network/prefix table and attribute table, performs best-path calculation among many other tasks.
  Huge route churn is a very common scenario that leads to this situation.
- BGP Scanner: Low-priority process that runs every 60 seconds by default. This process checks the entire BGP table to verify the next-hop reachability and updates the BGP table accordingly, in case there is any change for a path. It runs through the Routing Information Base (RIB) for redistribution purposes.
  Check platform scale, as more prefixes and routes installed and TCAM used, more resources needed, and usually, an overloaded device leads into such situations.

> **Note**: For further information on how to troubleshoot these two processes, refer to Troubleshoot High CPU Caused by the BGP Scanner or Router Process.

- BGP I/O: Runs when BGP control packets are received and manages the queuing and processing of BGP packets. If there are excessive packets received in the BGP queue for a long period, or if there is a problem with TCP, the router shows symptoms of high CPU due to BGP I/O process. (Usually, BGP Router is also high in this situation. Look at the message counts to identify peer and capture packets to identify the source of these messages.)
- BGP Open: Process used on session establishment. Not a common high CPU issue unless session is stuck in Open State.
- BGP Event: Is responsible for next-hop processing. Look for next-hops flaps on prefixes received.

Related Information

- Technical Support & Documentation - Cisco Systems
- BGP configuration guide
- MD5 Authentication Between BGP Peers Configuration Example
- Embedded Packet Capture
- BGP Neighbor Flaps with MTU Troubleshooting

- IANA Address Family Numbers

- Subsequent Address Family Identifiers (SAFI) Parameters

- Unsupported Capabilities Cause BGP Peer Malfunction
- BGP Best Path Selection Algorithm
- Understand BGP RIB-failure and The Command bgp suppress-inactive
- Understand the Importance of BGP Weight Path Attribute in Network Failover Scenarios
- Use the BGP "Slow Peer" Feature to Resolve Slow Peer Issues
- Configure BGP Routers for Optimal Performance and Reduced Memory Consumption
- Troubleshoot High CPU Caused by the BGP Scanner or Router Process