Troubleshoot High CPU Caused by the BGP Scanner or Router Process

Contents

Introduction

This document describes how to troubleshoot the cause of high CPU readings when the BGP Scanner or Router are used.

Prerequisites

Requirements

This document requires a knowledge of how to interpret the **show process cpu** command.

Components Used

The information in this document is based on Cisco IOS® Software Release 12.0.

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.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Background Information

This document describes situations in which a Cisco IOS router can experience high CPU utilization due to either the Border Gateway Protocol (BGP) router process or the BGP scanner process, as shown in the output of the **show process cpu** command. The duration of the high CPU condition varies based on a number of conditions, in particular the size of the Internet routing table and the number of routes that a particular router holds in its routing and BGP tables. The **show process cpu** command shows CPU utilization averaged over the past five seconds, one minute, and five minutes. CPU utilization numbers do not provide a true linear indication of the utilization with respect to the offered load.

These are some of the major reasons:

- In a real world network, the CPU has to handle various system maintenance functions, such as network management.
- The CPU has to process periodic and event-triggered routing updates.
- There are other internal system overhead operations, such as polling for resource availability, which are not proportional to traffic load.

You can also use the **show processes cpu** command in order to obtain some indication of CPU activity.

> **Note**: For more information about the show commands, see the Cisco IOS Configuration Fundamentals Command Reference

Understand the BGP Processes

A Cisco IOS process, in general, consists of the individual threads and associated data that perform tasks, such as system maintenance, switching packets, and implementation of routing protocols. Several Cisco IOS processes executed on the router enable BGP to run. Use the **show process cpu | include BGP** command to see the amount of CPU utilization due to BGP processes.

This table lists the function of the BGP processes and shows that each process runs at different times, and those times depend on the tasks that are handled Because the BGP scanner and BGP router processes are responsible for a large amount of calculations, you can see high CPU due to either one of these processes. The next sections discuss these processes in greater detail.

Process Name	**Description**	**Interval**
BGP Open	Performs BGP peer establishment.	At initialization, when a TCP connection with a BGP peer is established.
BGP I/O	Handles BGP packets that are in the queue and are processed, such as UPDATES and KEEPALIVES.	As BGP control packets are received.
BGP Scanner	Walks the BGP table and confirms reachability of the next hops. BGP scanner also checks conditional-	Once a minute.

	advertisement to determine whether BGP advertises condition prefixes and performs route reduction. In an MPLS VPN environment, BGP scanner imports and exports routes into a particular VPN routing and forwarding instance (VRF).	
BGP Router	Calculates the best BGP path and processes any route **churn.** It also sends and receives routes, establishes peers, and interacts with the routing information base (RIB).	Once per second and when a peer is added, removed, or sof configured.

High CPU due to BGP Scanner

High CPU due to the BGP scanner process can be expected for short durations on a router that carries a large Internet routing table. Once a minute, BGP scanner walks the BGP RIB table and performs important maintenance tasks. These tasks include examination of the next-hop referenced in the BGP table of the router and verifies that the next-hop devices can be reached. So, a large BGP table takes an equivalently large amount of time to be walked and validated.

Because the BGP Scanner process runs through the entire BGP table, the duration of the high CPU condition varies with the number of neighbors and the number of routes learned per neighbor. Use the **show ip bgp summary** and **show ip route summary** commands to capture this information. The BGP scanner process walks the BGP table to update any data structures and walks the routing table for route redistribution purposes. (In this context, the routing table is also known as the routing information base (RIB), which the router outputs when you execute the ;show ip route command). Both tables are stored separately in the memory of the router and can be large and consume CPU cycles.

The next sample output of the **debug ip bgp updates** command captures an execution of BGP scanner:

```
router#
2d17h: BGP: scanning routing tables
2d17h: BGP: 10.0.0.0 computing updates, neighbor version 8,
     table version 9, starting at 0.0.0.0
2d17h: BGP: 10.0.0.0 update run completed, ran for 0ms, neighbor
     version 8, start version 9, throttled to 9, check point net 0.0.0.0
2d17h: BGP: 10.1.0.0 computing updates, neighbor version 8,
     table version 9, starting at 0.0.0.0
2d17h: BGP: 10.1.0.0 update run completed, ran for 4ms, neighbor
     version 8, start version 9, throttled to 9, check point net 0.0.0.0
router#
```

While BGP scanner runs, low priority processes need to wait a longer time to access the CPU. One low priority process controls Internet Control Message Protocol (ICMP) packets such as pings. Packets destined to or originated from the router can experience higher than expected latency since the ICMP process must wait behind the BGP scanner. The cycle is that BGP scanner runs for some time and suspends itself, and then ICMP runs. In contrast, pings sent through a router must be switched via Cisco Express Forwarding (CEF) and must not experience any additional latency. When you troubleshoot periodic spikes in latency, compare forwarding times for packets forwarded through a router against packets processed directly by the CPU on the router.

> **Note**: Ping commands that specify IP options, such as record route, also require that the CPU processes them directly and this can cause longer forwarding delays.

Use the **show process | include bgp scanner** command to see the CPU priority. The value of **Lsi** in the next sample output uses **L** to refer to a low priority process.

```
6513#show processes | include BGP Scanner
 172 Lsi 407A1BFC       29144     29130    1000 8384/9000  0 BGP Scanner
```

High CPU due to BGP Router Process

The BGP router process runs about once per second to check for work. BGP convergence defines the duration between the time when the first BGP peer is established and the point at which BGP is converged. In order to ensure the shortest possible convergence times, BGP router consumes all free CPU cycles. However, after it starts, it relinquishes (or suspends) the CPU intermittently.

Convergence time is a direct measurement of how long BGP router spends on the CPU, not the total time. This procedure shows a high CPU condition during BGP convergence and exchanges BGP prefixes with two external BGP (eBGP) peers.

1. Capture a baseline for normal CPU utilization before you start the test.
   ```
   router#show process cpu
     CPU utilization for five seconds: 0%/0%; one minute: 4%; five minutes: 5%
   ```
2. Once the test starts, the CPU reaches 100 percent utilization. The**show process cpu**command shows that the high CPU condition is caused by BGP router, denoted by 139 (the Cisco IOS process ID for BGP router) in the next output.
   ```
   router#show process cpu
     CPU utilization for five seconds: 100%/0%; one minute: 99%; five minutes: 81%

   !--- Output omitted. 139 6795740 1020252 6660 88.34% 91.63% 74.01% 0 BGP Router
   ```
3. You can monitor and capture multiple outputs of the **show ip bgp summary** and **show process cpu** commands at this time. The **show ip bgp summary** command captures the state of the BGP neighbors.
   ```
   router#show ip bgp summary
     Neighbor    V    AS  MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down State/PfxRcd
     10.0.0.0    4  64512  309453  157389    19981    0  253 22:06:44 111633
     10.1.0.0  4  65101  188934    1047    40081   41    0 00:07:51 58430
   ```
4. When the router completes prefix exchange with its BGP peers, the CPU utilization rates return to normal levels. The computed one minute and five minute averages can settle back down as well and can show higher than normal levels for a longer period than the five seconds rate.
   ```
   router#show process cpu
     CPU utilization for five seconds: 3%/0%; one minute: 82%; five minutes: 91%
   ```
5. Use the captured output from the previous show commands to calculate the BGP convergence time. In particular, use the **Up/Down< /strong> column of the show ip bgp summary command and compare the start and stop times of the high CPU condition. Typically, BGP convergence can take several minutes when a large Internet routing table. is swapped**

Note: The high CPU on the device could also be due to the instability of the BGP table. This happens if the router receives two copies of the routing table, one from the EBGP peering with the ISP and the other from the IBGP peering in the network. The root cause for this is the amount of memory on the device. Cisco recommends a minimum of 1 Gig of RAM for a single copy of the internet routing table. To circumvent this instability, increase the RAM on the device or filter the prefixes so that the BGP table and the memory occupied by it is relieved.

Performance Improvements

As the number of routes in the Internet routing table grows, so does the time it takes for BGP to converge. In general, convergence is defined as the process when all route tables are brought to a state of consistency. BGP is considered to be converged when these conditions are true:

- All routes have been accepted.
- All routes have been installed in the routing table.
- The table version for all peers equals the table version of the BGP table.
- The InQ and OutQ for all peers is zero.

This section describes some Cisco IOS performance improvements to reduce BGP convergence time, which result in a reduction of a high CPU condition due to a BGP process.

Queue to TCP Peer Connections

BGP now queues data aggressively from the BGP OutQ to the TCP socket for each peer until the OutQs have drained completely. Since BGP now sends at a faster rate, BGP converges more quickly.

BGP Peer Groups

While they help simplify BGP configuration, BGP peer groups also can enhance scalability. All peer group members must share a common outbound policy. Thus, the same update packets can be sent to each group member, and this reduces the number of CPU cycles that BGP requires to advertise routes to peers. In other words, with peer groups, BGP walks the BGP table only on the peer group leader, filters the prefixes through the outbound policies, and generates updates, which it sends to the peer group leader. In turn, the leader replicates the updates to group members with which it is synchronized. Without peer groups, BGP must walk the table for every peer, filter prefixes through outbound policies, and generate updates that are sent only to the one peer.

Path MTU and the ip tcp path-mtu-discovery Command

All TCP sessions are bounded by a limit on the number of bytes that can be transported in a single packet. This limit, known as the Maximum Segment Size (MSS), is 536 bytes by default. In other words, TCP breaks up packets in a transmit queue into 536 byte chunks before it passes packets down to the IP layer. Use the **show ip bgp neighbors | include max data** command to display the MSS of BGP peers:

```
Router#show ip bgp neighbors | include max data
Datagrams (max data segment is 536 bytes):
Datagrams (max data segment is 536 bytes):
```

```
Datagrams (max data segment is 536 bytes):
Datagrams (max data segment is 536 bytes):
```

The advantage of a 536 byte MSS is that packets are not likely to be fragmented at an IP device along the path to the destination since most links use an MTU of at least 1500 bytes. The disadvantage is that smaller packets increase the amount of bandwidth used to transport overhead. Since BGP builds a TCP connection to all peers, a 536 byte MSS affects BGP convergence times.

The solution is to enable the Path MTU (PMTU) feature with the **ip tcp path-mtu-discovery** command. You can use this feature to dynamically determine how large the MSS value can be and in the meantime, not create packets that need to be fragmented. PMTU allows TCP to determine the smallest MTU size among all links in a TCP session. TCP then uses this MTU value, minus room for the IP and TCP headers, as the MSS for the session. If a TCP session only traverses Ethernet segments, then the MSS is 1460 bytes. If it only traverses Packet over SONET (POS) segments, then the MSS is 4430 bytes. The increase in MSS from 536 to 1460 or 4430 bytes reduces TCP/IP overhead, which helps BGP converge faster.

After you enable PMTU, again use the **show ip bgp neighbors | include max data** command to see the MSS value per peer:

```
Router#show ip bgp neighbors | include max data
Datagrams (max data segment is 1460 bytes):
Datagrams (max data segment is 1460 bytes):
Datagrams (max data segment is 1460 bytes):
Datagrams (max data segment is 1460 bytes):
```

Increase Interface Input Queues

If BGP advertises thousands of routes to many peers, TCP must transmit thousands of packets in a short duration. The BGP peers receive these packets and send TCP acknowledgements to the BGP speaker that advertises, which causes the BGP speaker to receive a flood of TCP ACKs in a short period of time. If the ACKs arrive at a rate that is too high for the route processor, packets back up in inbound interface queues. By default, router interfaces use an input queue size of 75 packets. In addition, special control packets such as BGP UPDATES use a special queue with Selective Packet Discard (SPD). This special queue holds 100 packets. When BGP convergence, happens TCP ACKs can fill quickly the 175 spots of input buffering and the new packets that arrive must be dropped. On routers with 15 or more BGP peers and exchange of the full Internet routing table, over 10,000 drops per interface per minute can be seen. Here is example output from a router 15 minutes after reboot:

```
Router#show interface pos 8/0 | include input queue
Output queue 0/40, 0 drops; input queue 0/75, 278637 drops
Router#
```

If you increase the interface input queue depth (with the **hold-queue in** command) it helps reduce the number of dropped TCP ACKs, which reduces the amount of work BGP must do to converge. Normally, a value of 1000 resolves problems caused by input queue drops.

Note: Use this consciously as input queue increment can add some delay.

Additional Improvements in Cisco IOS

Cisco IOS includes several optimizations to the BGP peer group code to improve update packing and replication. Before you review these improvements, review the update packing and replication in more detail.

A BGP update consists of a combination of attributes (such as MED = 50 and LOCAL_PREF = 120) and a list of Network Layer Reachability Information (NLRI) prefixes which share that combination of attributes. The more NLRI prefixes that BGP can list in a single update, the faster BGP can converge since overhead (such as IP, TCP, and BGP headers) is reduced. **Update packing** refers to the packing of NLRI into BGP updates. For example, if a BGP table holds 100,000 routes with 15,000 unique attribute combinations, BGP needs to send only 15,000 updates if NLRI is packed with 100 percent efficiency.

> **Note**: Zero percent packing efficiency means that BGP would need to send 100,000 updates in this environment.

Use the **show ip bgp peer-group** command to view the efficiency of the BGP updates.

If a peer group member is in sync, a BGP router takes an update message formatted for the peer group leader and replicates it for the member. It is far more efficient to replicate an update for a peer group member than it is to reformat the update. For example, assume a peer group has 20 members and all members need to receive 100 BGP messages. One hundred percent replication means that a BGP router formats the 100 messages for the peer group leader and replicates those messages to the other 19 peer group members. To confirm replication improvements, compare the number of messages replicated to the number of messages formatted, as shown by the **show ip bgp peer-group** command. The improvements make a dramatic difference in convergence times and allow BGP to support many more peers.

For example, use the **show ip bgp peer-group** command to check the efficiency of update packing and update replication. The next output is from a convergence test with 6 peer groups, 20 peers in each of the first 5 peer groups (eBGP peers) and 100 peers in the sixth peer group (internal BGP (iBGP) peers). Also, the BGP table that was used had 36,250 attribute combinations.

The next sample output of the **show ip bgp peer-group | include replicated** command on a router that runs Cisco IOS 12.0(18)S displays this information:

```
Update messages formatted 836500, replicated 1668500
Update messages formatted 1050000, replicated 1455000
Update messages formatted 660500, replicated 1844500
Update messages formatted 656000, replicated 1849000
Update messages formatted 501250, replicated 2003750

!-- The first five lines are for eBGP peer groups. Update messages formatted 2476715,
replicated 12114785 !-- The last line is for an iBGP peer group.
```

In order to calculate the replication rate for each peer group, divide the number of updates replicated by the number of updates formatted:

1668500/836500 = 1.99 1455000/1050000 = 1.38 1844500/660500 = 2.79 1849000/656000 = 2.81 2003750/501250 = 3.99 12114785/2476715 = 4.89

- If BGP replicated perfectly, then the eBGP peer groups each have a replication rate of 19 because there are 20 peers in the peer group. The update is formatted for the peer group leader and then replicated to the other 19 peers. This provides an optimal replication rate of 19. The ideal replication rate for the iBGP peer group would be 99 since there are 100 peers.
- If BGP packed updates perfectly, then there are only formatted 36,250 updates. You only need to generate 36,250 updates for each peer group because that is the number of attribute combinations in the BGP table. The iBGP peer group alone formats almost 2,500,000 updates while the eBGP peer groups each generate anywhere from 500,000 to 1,000,000 updates.

On a router that runs Cisco IOS 12.0(19)S, the**show ip bgp peer-group | include replicated**command provides this information:

```
Update messages formatted 36250, replicated 688750
Update messages formatted 36250, replicated 688750
Update messages formatted 36250, replicated 688750
Update messages formatted 36250, replicated 688750
Update messages formatted 36250, replicated 688750
Update messages formatted 36250, replicated 3588750
```

Note: Update packing is optimal. Exactly 36,250 updates are formatted for each peer group. 688750/36250 = 19 688750/36250 = 19 688750/36250 = 19 688750/36250 = 19 688750/36250 = 19 3588750/36250 = 99

Note: Update replication is also perfect.

Troubleshoot Procedures

Use these procedures in order to troubleshoot high CPU due to BGP scanner or BGP router:

- Gather information on your BGP topology. Determine the number of BGP peers and the number of routes that are advertised by each peer. Is the duration of the high CPU condition reasonable based on your environment?
- Determine when the high CPU happens. Does it coincide with a regularly scheduled walk of the BGP table?
- Did the high CPU follow an interface flap? You can use command **show ip bgp dampening flap-statistics** command if dampening is enabled.
- Ping through the router and then ping from the router. ICMP echoes are handled as a low priority process. The documentUnderstanding the Ping and Traceroute Commandsexplains this in more detail. Ensure regular forwarding is not affected.
- You need to ensure that packets can follow a fast forwarding path when you check whether fast switching and/or CEF are enabled on the inbound and the outbound interfaces. Ensure that you do not see the **no ip route-cache cef** command on the interface or the **no ip cef** command on the global configuration. In order to enable CEF in global configuration mode, use the **ip cef** command.
- Check platform scale because, in most cases, it is due to an overloaded device that leads into such situations. Also, ensured that there is proper ternary content addressable memory (TCAM) space available on the router.

- Verify enough memory on the router. As per the recommendation, have a minimum of 1 GB of

DRAM assigned to Cisco IOS space per BGP peer that sends the full Internet routing table. The DRAM space mentioned here is just the memory required for BGP. Other features that run on the router can require additional space.

Related Information

- **IP Routing Support Page**
- **Technical Support - Cisco Systems**