

Troubleshoot High CPU on ASR1000 Series Router

Contents

[Introduction](#)

[Prerequisite](#)

[Requirements](#)

[Description](#)

[Troubleshoot Steps](#)

[Step 1 – Identify the module with high CPU](#)

[Step 2 – Analyze the module](#)

[Step 3 – IOS Processes](#)

[Step 4 – Linux Processes](#)

[Step 5 – FECF Processes](#)

[Step 6 – QFP Utilization](#)

[Step 7 – Determine the root cause and identify the fix](#)

[Troubleshoot Example](#)

[Additional Commands](#)

[Route Processor](#)

[Embedded Services Processor](#)

Introduction

This document describes how to troubleshoot high CPU issues on an ASR1000 series router.

Prerequisite

Requirements

Cisco recommends that you understand the [ASR1000 architecture](#) to interpret and utilize this document.

Description

High CPU on a Cisco router may be defined as the condition where the CPU utilization on the router is above the normal usage. In some scenarios the increased CPU usage is expected while in other scenarios it could indicate a problem. Transient high CPU utilization on the router due to network change or configuration change can be ignored and is expected behaviour.

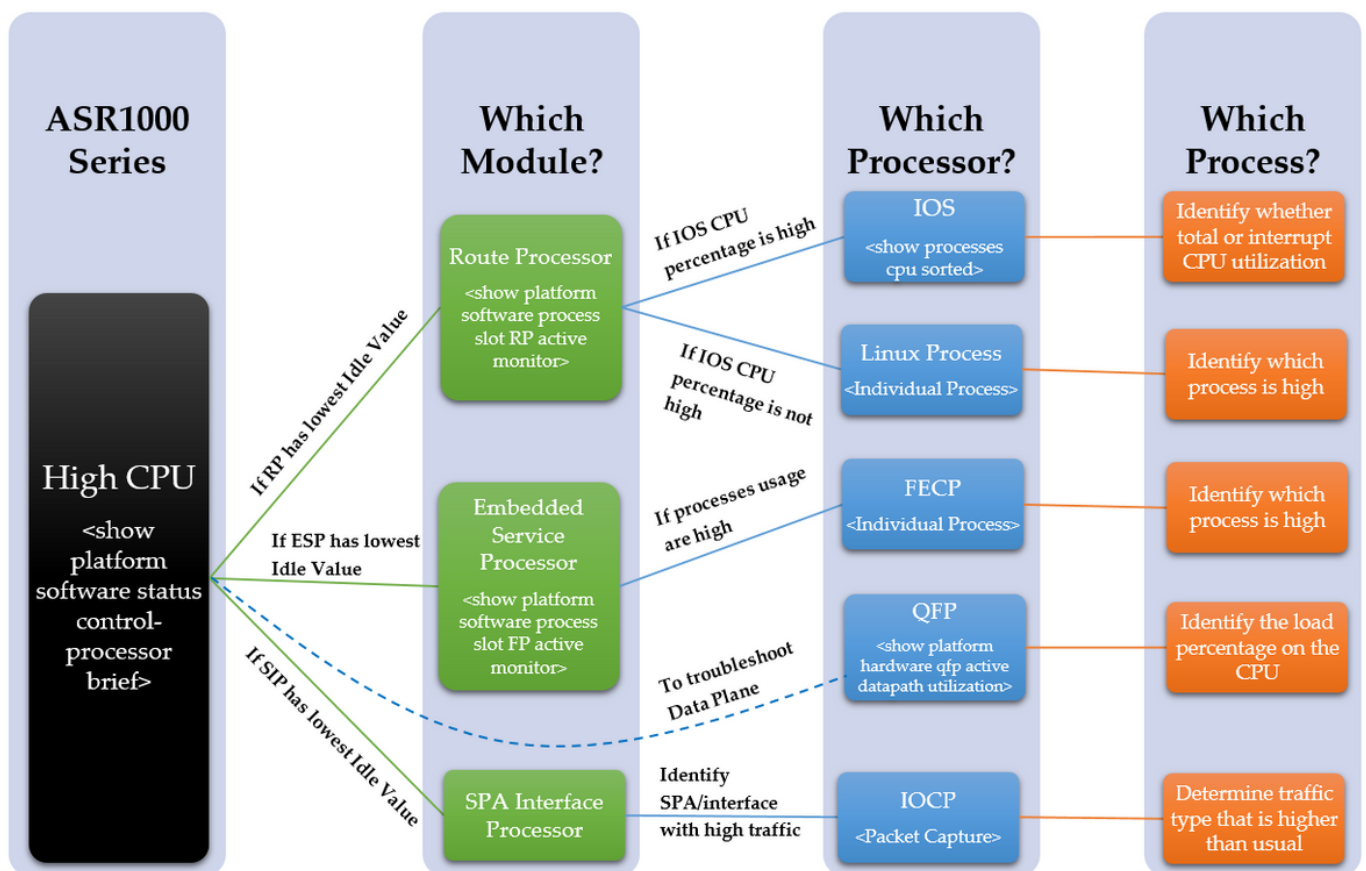
However, a router that experiences high CPU utilization for extended periods without any changes in the network or configuration is unusual and needs to be analyzed. Therefore when overused, the CPU is not able to actively service all other processes, which results in slow command line, control plane latency, packet drops, and the failure of services.

The causes of high CPU are:

1. Control Plane CPU receives too much punt traffic
2. A process that behaves unexpectedly and results in CPU over-utilization
3. Data Plane processor is over-utilized /oversubscribed
4. Too many processor interrupts

High CPU is not always a ASR1000 series router problem as router CPU utilization is directly proportional to the load on the router. For example if there is a network change, this will cause a large amount of control plane traffic as the network will re-converge. Therefore, we need to determine the root cause of the CPU over-utilization to determine if it is expected behaviour or an issue.

Below is a diagram that details a step by step process on how to troubleshoot a High CPU issue:



Troubleshoot Steps

Step 1 – Identify the module with high CPU

ASR1000 has several different CPU across the different modules. Therefore, we need to see which module shows greater than normal usage. This can be seen through the Idle value, as the lower the idle value, the higher the CPU utilization of that module. These different CPU all reflect the control plane of the modules.

Determine which module within the device is observed to experience high CPU. Is it the RP, ESP, or SIP with the below command

```
show platform software status control-processor brief
```

Refer to the below output to view the highlighted column

If the RP has a low Idle value, then proceed to Step 2 Point 1

If the ESP has a low Idle value, then proceed to Step 3 Point 2

If the SIP has a low Idle value, then proceed to Step 4 Point 3

```
Router#show platform software status control-processor brief
```

Load Average

Slot	Status	1-Min	5-Min	15-Min
RP0	Healthy	0.00	0.02	0.00
ESP0	Healthy	0.01	0.02	0.00
SIP0	Healthy	0.00	0.01	0.00

Memory (kB)

Slot	Status	Total	Used (Pct)	Free (Pct)	Committed
RP0	Healthy	2009376	1879196 (94%)	130180 (6%)	1432748
ESP0	Healthy	2009400	692100 (34%)	1317300 (66%)	472536
SIP0	Healthy	471804	284424 (60%)	187380 (40%)	193148

CPU Utilization

Slot	CPU	User	System	Nice	Idle	IRQ	SIRQ	IOwait
RP0	0	2.59	2.49	0.00	94.80	0.00	0.09	0.00
ESP0	0	2.30	17.90	0.00	79.80	0.00	0.00	0.00
SIP0	0	1.29	4.19	0.00	94.41	0.09	0.00	0.00

If the Idle values are all relatively high, it may not be a control plane issue. To troubleshoot the data plane the ESP's QFP needs to be observed. Symptoms of "high CPU" can still be observed due to an over-utilized QFP, which will not result in high CPU on the control plane processors. Proceed to STEP 6.

Step 2 – Analyze the module

• Route Processor

Confirm within the RP which processor is observed to have high CPU utilization with the below command. Is it the Linux process or the IOS?

```
show platform software process slot RP active monitor
```

If IOS CPU percentage is high (linux_iosd-imag), then is it is the RP IOS. Proceed to STEP 3

If the CPU percentage of other processes is high, then it is likely to be the Linux Process. Proceed to STEP 4

• Embedded Services Processor

Confirm within the ESP if the control plane processor is observed to have high CPU utilization. Is it

the FECP?

```
show platform software process slot FP active monitor
```

If processes are high then it is the FECP, then proceed to STEP 5

If it is not the FECP, it is not a control plane processes related issue within the ESP. If symptoms such as network latency or queue drops are still observed, the data plane may need to be reviewed for over-utilization. Proceed to STEP 6

- **SPA Interface Processor**

If the SIP is observed to have high CPU utilization then the IOCP will be observed to have high CPU. Determine which process or processes within the IOCP are observed to have high CPU utilization.

Perform a packet capture and identify which traffic is higher than usual and which processes are associated with this type of traffic. Proceed to STEP 7

Step 3 – IOS Processes

Refer to the below output, the first percentage is the total CPU utilization, and the second percentage is the interrupt CPU utilization, which is the amount of CPU used to process punted packets.

If the interrupt percentage is high then it signifies that a large amount of traffic is punted to the RP, (this can be confirmed with the command **show platform software infrastructure punt**)

If the interrupt percentage is low, but the total CPU is high, then there is a process or processes that will be observed to utilize the CPU for an extended period.

Confirm within the IOS which process or processes are observed to have high CPU utilization with the below command.

```
show processes cpu sorted
```

Identify which percentage is high (total CPU or interrupt CPU), and then if required identify the individual process/processes. Proceed to STEP 7

```
Router#show processes cpu sorted
```

```
CPU utilization for five seconds: 0%/0%; one minute: 1%; five minutes: 1%
```

PID	Runtime (ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY
Process							

PID	Runtime (ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY
Process							

188	8143	434758	18	0.15%	0.18%	0.19%	0
-----	------	--------	----	-------	-------	-------	---

Ethernet Msec Ti							
515	380	7050	53	0.07%	0.00%	0.00%	0
SBC main process							
3	2154	215	10018	0.07%	0.00%	0.19%	0
Exec							
380	1783	55002	32	0.07%	0.06%	0.06%	0
MMA DB TIMER							
63	3132	11143	281	0.07%	0.07%	0.07%	0
IOSD ipc task							
5	1	2	500	0.00%	0.00%	0.00%	0
IPC ISSU Dispatc							
6	19	12	1583	0.00%	0.00%	0.00%	0
RF Slave Main Th							
8	0	1	0	0.00%	0.00%	0.00%	0
RO Notify Timers							
7	0	1	0	0.00%	0.00%	0.00%	0
EDDRI_MAIN							
10	6	75	80	0.00%	0.00%	0.00%	0
Pool Manager							
9	5671	538	10540	0.00%	0.14%	0.12%	0
Check heaps							

Step 4 – Linux Processes

If the IOS is observed to have over-utilized the CPU, then we need to observe the CPU utilization for the individual linux process. These processes are the other processes listed from the **show platform software process slot RP active monitor**. Identify which process or processes are observed to experience high CPU then proceed to STEP 7.

Step 5 – FECF Processes

If a process or processes are high then it is likely those are the processes within the FECF that are responsible for the high CPU utilization, proceed to STEP 7

Step 6 – QFP Utilization

The Quantum Flow Processor is the forwarding ASIC. To determine the load on the forwarding engine, the QFP can be monitored. The below command lists the input and output packets (priority and non-priority) in packets per second, and bits per second. The final line displays the total amount of CPU load due to packet forwarding in a percentage.

```
show platform hardware qfp active datapath utilization
```

Identify if input or output are high, and view the process load and then proceed to STEP 7

```
Router#show platform hardware qfp active datapath utilization
```

CPP 0: Subdev 0	5 secs	1 min	5 min	60 min
Input: Priority (pps)	0	0	0	0
(bps)	208	176	176	176
Non-Priority (pps)	0	2	2	2
(bps)	64	784	784	784
Total (pps)	0	2	2	2
(bps)	272	960	960	960
Output: Priority (pps)	0	0	0	0
(bps)	192	160	160	160
Non-Priority (pps)	0	1	1	1
(bps)	0	6488	6496	6488
Total (pps)	0	1	1	1
(bps)	192	6648	6656	6648
Processing: Load (pct)	0	0	0	0

Step 7 – Determine the root cause and identify the fix

With the process or processes that are observed to have over-utilized the CPU identified, there is a clearer picture of why high CPU has occurred. To proceed, research the functions performed by the identified process. This will assist in to determine an action plan on how to approach the problem. For example - If the process is responsible for a particular protocol then you may want to look at the configuration related to this protocol.

If you still experience CPU related issues, it is recommended to contact TAC to allow an engineer to help you troubleshoot further. The above steps to troubleshoot will help the engineer isolate the issue more efficiently.

Troubleshoot Example

In this example we will run through the process to troubleshoot and try to best identify a possible root cause for the router's high CPU. To begin, determine which module is observed to experience the high CPU, we have the below output:

```
Router#show platform software status control-processor brief
Load Average
```

Slot	Status	1-Min	5-Min	15-Min
RP0	Healthy	0.66	0.15	0.05
ESP0	Healthy	0.00	0.00	0.00
SIP0	Healthy	0.00	0.00	0.00

Memory (kB)

Slot	Status	Total	Used (Pct)	Free (Pct)	Committed (Pct)
RP0	Healthy	2009376	1879196 (94%)	130180 (6%)	1432756 (71%)
ESP0	Healthy	2009400	692472 (34%)	1316928 (66%)	472668 (24%)
SIP0	Healthy	471804	284556 (60%)	187248 (40%)	193148 (41%)

CPU Utilization

Slot	CPU	User	System	Nice	Idle	IRQ	SIRQ	IOwait
RP0	0	57.11	14.42	0.00	0.00	28.25	0.19	0.00
ESP0	0	2.10	17.91	0.00	79.97	0.00	0.00	0.00
SIP0	0	1.20	6.00	0.00	92.80	0.00	0.00	0.00

As the Idle amount within RP0 is very low, it suggests a high CPU issue within the Route Processor. Therefore to troubleshoot further we will identify which processor within the RP is observed to experience high CPU.

Router#**show processes cpu sorted**

CPU utilization for five seconds: **84%/36%**; one minute: 34%; five minutes: 9%

PID	Runtime (ms)	Invoked	uSecs	5Sec	1Min	5Min	TTY
107	303230	50749	5975	46.69%	18.12%	4.45%	0
IOSXE-RP Punt Se							
63	105617	540091	195	0.23%	0.10%	0.08%	0 IOSD
ipc task							
159	74792	2645991	28	0.15%	0.06%	0.06%	0 VRRS
Main thread							
116	53685	169683	316	0.15%	0.05%	0.01%	0 Per-
Second Jobs							
9	305547	26511	11525	0.15%	0.28%	0.16%	0 Check
heaps							
188	362507	20979154	17	0.15%	0.15%	0.19%	0
Ethernet Msec Ti							
3	147	186	790	0.07%	0.08%	0.02%	0 Exec
2	32126	33935	946	0.07%	0.03%	0.00%	0 Load
Meter							
446	416	33932	12	0.07%	0.00%	0.00%	0 VDC
process							
164	59945	5261819	11	0.07%	0.04%	0.02%	0 IP
ARP Retry Age							
43	1703	16969	100	0.07%	0.00%	0.00%	0 IPC
Keep Alive M							

From this output, it can be observed that the total CPU percentage and the interrupt percentage are higher than expected. The top process that utilizes the CPU is the "IOSXE-RP Punt Se" which is the process that handles traffic for the RP CPU, therefore we can look further into this traffic that is punted to the RP.

```

Router#show platform software infrastructure punt
LSMPI interface internal stats:
enabled=0, disabled=0, throttled=0, unthrottled=0, state is ready
Input Buffers = 90100722
Output Buffers = 100439
rxdone count = 90100722
txdone count = 100436
Rx no particletype count = 0
Tx no particletype count = 0
Txbuf from shadow count = 0
No start of packet = 0
No end of packet = 0
Punt drop stats:
Bad version 0
Bad type 0
Had feature header 0
Had platform header 0
Feature header missing 0
Common header mismatch 0
Bad total length 0
Bad packet length 0
Bad network offset 0
Not punt header 0
Unknown link type 0
No swidb 1
Bad ESS feature header 0
No ESS feature 0
No SSLVPN feature 0
Punt For Us type unknown 0
Punt cause out of range 0
IOSXE-RP Punt packet causes:
    62210226 Layer2 control and legacy packets
        147 ARP request or response packets
    27801234 For-us data packets
        84426 RP<->QFP keepalive packets
            6 Glean adjacency packets
        1647 For-us control packets

FOR_US Control IPv4 protcol stats:
    1647 OSPF packets
Packet histogram(500 bytes/bin), avg size in 92, out 56:
Pak-Size      In-Count      Out-Count
    0+:        90097805      98790
    500+:          0              7

```

From this output, we can see there are a large amount of packets in the “For-us data packets” which indicates traffic directed towards the router, this counter was confirmed to be have incremented from observation of the command multiple times over several minutes. This confirms that the CPU is over-utilized by a large amount of punted traffic, which is often control plane traffic. Control plane traffic can include ARP, SSH, SNMP, Route Updates (BGP, EIGRP, OSPF) etc. From this information, we are able to identify the potential cause of the high CPU and this assists

to troubleshoot for the root cause. For example, a packet capture or a monitor of different traffic could be implemented to see the exact traffic punted to the RP which would allow the root cause to be identified and solved to prevent a similar issue in the future.

Once a packet capture is completed, some examples of potential punted traffic is:

- **ARP:** This could be due to an excessive number of ARP requests, which would occur if multiple IP addresses were to send ARP requests through the configuration of an IP route to a broadcast interface. This could also be due to flushed entries from the ARP table and will have to be relearned based on the MAC Address entries that age out, or interfaces that come up/down.
- **SSH:** This could cause high CPU due to a large show command (show tech-support) or when a lot of debug commands are enabled, which forces a lot of CLI to be sent over the SSH session.
- **SNMP:** This could be due to the SNMP agent that takes a long period of time to process a request, and therefore causes the high CPU. Often two probable causes are MIBs that are polled, or route and/or ARP tables that are polled by the NMS.
- **Route Updates:** Often an influx of route updates will be due to a network re-convergence, or link flaps. This could indicate routes that go down within the network, or entire devices that go down which forces the network to converge and recalculate the best routes, which depends on which routing protocol is in use.

This highlights how the root cause can be isolated through identification of the cause of the high CPU, when it comes down to an individual process level. From here, the individual process or protocol can be analysed in isolation to identify whether it is a configuration issue, software issue, network design, or intended practice.

Additional Commands

The below is a list of other additional useful commands to utilize and are sorted by which processor they relate to:

Route Processor

- **<show process cpu history>** Provides a graph of the CPU history over the last 60 seconds, minutes, and 72 hours
- **<show process *process_ID*>** Detailed information on individual process memory and CPU allocations
- **<show platform software infrastructure punt>** Provides information on all traffic that is punted to the RP
- **<show platform software status control-processor brief>** Details the load and 'health' of the CPU, as well as details the memory and module statistics
- **<show platform software process slot r0|r1 monitor>** Details the different processes and their CPU and Memory allocations on the selected module
- **<monitor platform software process r0|r1>** Provides a live feed that updates of the processes as they utilize the CPU. Requires the command "terminal terminal-type" to be entered in global configuration mode first to function correctly

Embedded Services Processor

- **<show platform software process list fp active summary>** Details the a summary of all the processes that are run on the CPU, as well as the average load
- **<show platform software process slot f0|f1 monitor>** Details the different processes and their CPU and Memory allocations on the selected module
- **<monitor platform software process f0|f1>** Provides a live feed that updates of the processes as they are they utilize the CPU Requires the command “terminal terminal-type” to be entered in global configuration mode first to function correctly