Cisco – Troubleshooting High CPU Utilization on Cisco Route

Table of Contents

Troubleshooting High CPU Utilization on Cisco Routers

Interactive: This document offers customized analysis of your Cisco device.

Introduction

This document describes common symptoms and causes of, and solutions to, high CPU utilization on Cisco routers.

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

There are no specific prerequisites for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Symptoms of High CPU Utilization

The following list describes common symptoms of high CPU utilization. If you notice any of these symptoms, follow the troubleshooting steps in this document to alleviate the problem.

- High percentages in the **show processes cpu** command output

  If you have the output of a **show processes cpu** command or **show tech–support** (from enable mode), from your Cisco device, you can use to display potential issues and fixes. To use, you must be a registered customer, be logged in, and have JavaScript enabled.

  You can use Output Interpreter to display potential issues and fixes. To use Output Interpreter, you must be a registered customer, be logged in, and have JavaScript enabled.
- Input queue drops
- Slow performance
- Services on the router fail to respond, for instance:

  - ◆ Slow response in Telnet or unable to Telnet to the router
  - ◆ Slow response in Telnet or unable to Telnet to the router
  - ◆ Slow or no response to ping
  - ◆ Router doesn't send routing updates

Troubleshooting Strategy

Once you notice any of the above symptoms, determine the accessibility of the router:

- Are you able to issue **show** commands on the router? If so, start collecting more information immediately, using these show commands.
- Is the router inaccessible? Is this problem reproducible? If so, power–cycle the router and, before reproducing the problem, configure the **scheduler interval 500** command. This schedules low priority processes to run every 500 ms, which provides time for you to run some commands, even if CPU usage is at 100 percent. On Cisco 7200 and Cisco 7500 series routers, use the **scheduler allocate 3000 1000** command.
- Does the router show symptoms of high CPU utilization at brief and unpredictable intervals? If so, periodically collect the output of the **show processes cpu** command, which shows if the high CPU utilization is caused by interrupts or by a certain process. Use this UNIX script and, based on the first findings, modify the script to collect data needed for further investigation of the issue.

Determining Causes and Solving the Problem

Now it's time to determine what's loading the processor. Use the **show processes cpu** command to check if CPU utilization is high due to interrupts. If it isn't, check which process is loading the CPU.

Here's sample output:

```
router-5#show process cpu
CPU utilization for five seconds: 6%/4%; one minute: 5%; five minutes: 5%
 PID  Runtime(ms)  Invoked   uSecs    5Sec    1Min    5Min TTY Process
   1          104     3707      28   0.00%   0.00%   0.00%   0 Load Meter
   2        10208    15222     670   0.00%   0.01%   0.00%   0 OSPF Hello
   3        34620      579   59792   0.00%   0.20%   0.17%   0 Check heaps
   4            4        1    4000   0.00%   0.00%   0.00%   0 Pool Manager
```

Cisco – Troubleshooting High CPU Utilization on Cisco Routers

```
 5             0          2         0    0.00%   0.00%   0.00%   0 Timers
 6            48        315       152    0.00%   0.00%   0.00%   0 ARP Input
 7             0          1         0    0.00%   0.00%   0.00%   0 SERIAL A'detect
 8          3508       5588       627    0.00%   0.00%   0.00%   0 IP Input
 9          2604       2478      1050    0.08%   0.02%   0.00%   0 CDP Protocol
10          5992       3708      1615    0.00%   0.00%   0.00%   0 TCP Timer
11             0          1         0    0.00%   0.00%   0.00%   0 TCP Protocols
12             4          1      4000    0.00%   0.00%   0.00%   0 Probe Input
13             0          1         0    0.00%   0.00%   0.00%   0 RARP Input
14             8          5      1600    0.00%   0.00%   0.00%   0 BOOTP Server
15         74540      18845      3955    0.40%   0.38%   0.39%   0 IP Background
16            24        310        77    0.00%   0.00%   0.00%   0 IP Cache Ager
17             0          1         0    0.00%   0.00%   0.00%   0 Critical Bkgnd
18            24         10      2400    0.00%   0.00%   0.00%   0 Net Background
19             8         19       421    0.00%   0.00%   0.00%   0 Logger
20          1540      18519        83    0.00%   0.00%   0.00%   0 TTY Background
21           900      18526        48    0.00%   0.00%   0.00%   0 Per-Second Jobs
22          9124      18525       492    0.08%   0.04%   0.06%   0 Net Periodic
23           880       1855       474    0.00%   0.00%   0.00%   0 Net Input
24          3052       3709       822    0.08%   0.00%   0.00%   0 Compute load avgs
25         11036        310     35600    0.73%   0.08%   0.06%   0 Per-minute Jobs
26          3208       3705       865    0.00%   0.00%   0.00%   0 FR LMI
27           636        310      2051    0.00%   0.00%   0.00%   0 FR ARP
28           732     185219         3    0.00%   0.00%   0.00%   0 FR Broadcast Output
29          3632        523      6944    0.40%   0.14%   0.06%   0 Exec
30           936        315      2971    0.00%   0.00%   0.00%   0 IP-RT Background
31          1920        497      3863    0.00%   0.00%   0.00%   0 OSPF Router
```

High CPU Utilization due to Interrupts

CPU interrupts are primarily caused by fast switching of traffic. Interrupts are also generated any time a character is output from the console or auxiliary ports of a router. However, Universal Asynchronous Receiver/Transmitters (UARTs) are slow, compared to the processing speed of the router, so it's not likely that console or auxiliary interrupts can cause a high CPU utilization on the router.

There are several reasons for high CPU utilization due to interrupts:

- Voice ports are configured on the router. Even if there's no traffic, software continues to monitor channel associated signaling (CAS), which uses CPU resources.
- There are active Asynchronous Transfer Mode (ATM) interfaces on the router. Even there's no traffic, the ATM interfaces send out null cell (per ATM standards) and continue to use CPU resources.
- An inappropriate switching path is configured on the router. If you have a Cisco 7000 or Cisco 7500 series router, try improving its performance by using the **ip route−cache** *<path>* command, where *path* can be *cef*, *distributed*, or cbus, depending on the platform. If there are access lists linked to interfaces or if **ip accounting** is turned on, configure NetFlow switching using the **ip route−cache flow** command.
- The CPU is performing memory alignment corrections. If there are %ALIGN−3−CORRECT messages logged, then the high CPU utilization is caused by memory alignment corrections. Capture the output of the **show align** command, decode the tracebacks and search for a bug in your version of Cisco IOS®.
- The router is overloaded with traffic. The output of the **show interfaces** and **show interfaces switching** commands provide information about which interfaces are overloaded. To capture the output of these commands in a log file for later analysis, issue the **terminal length 0** command first. More detailed information follows:
- Check the output of **show interfaces**. Examine the load and number of throttles on interfaces. The load is an average value computed, by default, over five minutes. To change this interval, use the **load−interval** *seconds* command, where the seconds represent the length of time for which data is

used to compute load statistics. Use a value that's a multiple of 30.

Throttles are a better indication of an overloaded router. They show the number of times the receiver on the port has been disabled, possibly due to buffer or processor overload. Together with high CPU utilization on an interrupt level, throttles indicate that the router is overloaded with traffic.

Next, check the output of the **show interfaces switching** command to see what kind of traffic (protocol and switching path) is going through the overloaded interface. If some interfaces are too overloaded with traffic, consider redesigning the traffic flow in the network or upgrading the hardware.

If there is a possibilty that a single device is generating packets at an extremely high rate and thus overloading the router, there is a way to determine the Media Access Control (MAC) address of that device. The **ip accounting mac−address {input|output} interface** configuration command should be added to the configuration of the overloaded interface. The **show interfaces [ ] mac−accounting** command displays the collected information. Once the source device's MAC address is found, the corresponding IP address can be found by checking the output of the **show arp exec** command.

- There's a bug in the Cisco IOS Software version running on the router. Once you've performed all the previous steps in this list, check the Bug Navigator (registered customers only) for a bug that reports similar symptoms in a similar environment.

High CPU Utilization due to Processes

If a process is using a lot of CPU resources, check the log messages. Unusual activity related to a process results in an error message in the log. The following processes can cause high CPU utilization:

- IP Input
- HyBridge Input
- IP Simple Network Management Protocol (SNMP)
- Virtual EXEC
- TCP Timer
- VTEMPLATE Backgr
- Other Processes

IP Input

If the IP input process is using a lot of CPU resources, check the following issues:

- Fast switching is disabled on an interface (or interfaces) that has a lot of outgoing traffic. Examine the output of the **show interfaces switching** command to see which interface is burdened with traffic. Re−enable fast switching on that interface. Remember that regular fast switching is configured on output interfaces: if fast switching is configured on an interface, packets going out that interface are fast−switched. Optimum and NetFlow switching are configured on input interfaces. To create cache entries for a particular interface, configure optimum and NetFlow switching on all interfaces that are routing to that interface.
- Fast switching on the same interface is disabled. If an interface has lot of secondary addresses and there is a lot of traffic sourced from the same interface and destined for the address in the same interface, then it process−switches all of those packets. In this situation, you should enable **ip route−cache same−interface** on the interface.
- Traffic that can't be fast switched is arriving. This could be any of the following types of traffic. Click on linked items for more information.

♦ Packet for which there is no entry yet in the switching cache.

Even if fast, optimum, netflow, or Cisco express forwarding (CEF) switching(*) is configured, a packet for which there is no match in the cache will be processed. An entry in the cache is then created, and all subsequent packets that match the same criteria are fast, optimum, netflow, or CEF−switched. In normal circumstances, these processed packets don't cause high CPU utilization. However, if there is a device in the network which, 1) is generating lots of packets at an extremely high rate for devices reachable through the router, and 2) is using different source or destination ip addresses, there won't be a match for these packets in the switching cache, so they will be processed by the IP Input process (if netflow switching is configured, source and destination TCP ports are checked against entries in the cache as well). This source device can be a malfunctioning device or, more likely, a device attempting an attack.

(*)Only with glean adjacencies. See Cisco Express Forwarding documentation for more information about CEF adjacencies.

♦ Packets destined for the router
♦ IP packets with options
♦ Packets that require protocol translation
♦ Packets that require policy routing
♦ Packets going through serial interfaces with X.25 encapsulation
♦ Multilink PPP
♦ Compressed traffic
♦ Encrypted traffic

The following are examples of packets destined for the router:

◊ Routing updates arriving at an extremely high rate. If the router receives an enormous amount of routing updates that have to be processed, this task might overload the CPU. Normally, this can't happen in a stable network. The way you can gather more information depends on the routing protocol you have configured. However, you can start by periodically checking the output of the **show ip route** summary command. Rapidly changing values are a sign of an unstable network: Frequent routing table changes mean increased routing protocol processing, which results in increased CPU utilization. For further troubleshooting, consult the Troubleshooting TCP/IP section of the Internetwork Troubleshooting Guide.

◊ Any other kind of traffic destined for the router. Check who's logged on to the router and what the user is doing. If someone is logged on and is issuing commands that produce long output, the high CPU utilization by the IP input process will be followed by a much higher CPU utilization by the virtual EXEC process.

◊ Spoof attack. To identify the problem, check the amount of IP traffic by issuing the **show ip traffic** command. If there's a problem, the number of received packets with a local destination will be significant. Next, check through which interface the packets are coming in by examining the output of the **show interfaces** and **show interfaces switching** commands. Once you've identified the receiving interface, turn on ip accounting on the outgoing interface and see if there's a pattern. If it's an attack, the source address will almost always be different, but the destination address will be the same. An access list can be configured to solve the issue temporarily (preferably on the device closest to the source of the packets), but the real solution is to track down the source device and stop the attack.

♦ Packets that require policy routing. Prior to Cisco IOS version 11.3, policy−routed packets couldn't be fast switched. IOS version 11.3 and higher allows policy−routed packets to be fast

switched. In this case, use the interface configuration command **ip route–cache policy**.

♦ Packets going through serial interfaces with X.25 encapsulation. In the X.25 protocol suite, flow control is implemented on the second Open System Interconnection (OSI) layer.
♦ Compressed traffic. If there's no Compression Service Adapter (CSA) in the router, compressed packets must be process–switched.
♦ Encrypted traffic. If there's no Encryption Service Adapter (ESA) in the router, encrypted packets must be process–switched.

- A lot of packets, arriving at an extremely high rate, for a destination in a directly attached subnet, for which there is no entry in the Address Resolution Protocol (ARP) table. This shouldn't happen with TCP traffic, because of the windowing mechanism, but it can happen with User Datagram Protocol (UDP) traffic. To identify the problem, repeat the actions suggested for tracking down a spoof attack.
- A lot of multicast traffic going through the router. Unfortunately, there's no easy way to examine the amount of multicast traffic. The **show ip traffic** command only shows summary information. However, if you've configured multicast routing on the router, you can enable fast switching of multicast packets using the **ip mroute–cache** interface configuration command (fast switching of multicast packets is off by default).
- A lot of broadcast traffic. Check the number of broadcast packets in the **show interfaces** command output.
- Too much traffic is passing through the router. If the router is over–used and is incapable of handling this amount of traffic, try distributing the load among other routers or consider purchasing a high–end router.
- IP NAT (Network Address Translation) is configured on the router and there are lots of DNS (Domain Name System) packets going through the router. UDP or TCP packets with source and/or destination port 53 (DNS) are always punted to process level by NAT.

Whatever the reason for high CPU utilization in the **IP Input** process, the source of the problem can be tracked down by debugging IP packets. Since the CPU utilization is already high, the debugging has to be done with extreme caution. Debugging produces lots of messages, so only **logging buffered** should be configured. Logging to a console raises unnecessary interrupts to the CPU and thus increases the CPU utilization. Logging to a host (or monitor logging) generates additional traffic on interfaces. Debugging can be started by issuing the **debug ip packet detail exec** command. The debugging session shouldn't last longer than 3–5 seconds. Debugging messages are written in the logging buffer. A capture of a sample debugging session is given in the Sample IP Packet Debugging Session section of this document. Once the source device of unwanted IP packets is found, it can be disconnected from the network, or an access list can be created on the router to drop packets from that destination. Exact strategies for preventing different types of attacks are out of the scope of this document.

HyBridge Input

Routers with ATM interfaces that support a large number of permanent virtual circuits, configured to use bridged–format protocol data units with bridging and integrated routing and bridging (IRB), rely heavily on broadcasts for connectivity to remote users. This may cause high CPU utilization in the HyBridge Input process.

To troubleshoot this specific issue, see Troubleshooting High CPU Utilization Caused by the Hybridge Input Process on Routers With ATM Interfaces.

IP Simple Network Management Protocol (SNMP)

It's a known issue that the IP SNMP process can consume a lot of CPU resources if certain variables are polled. The most common situation is when a routing table is polled using SNMP. The routing table is stored in a tree–like structure in the main memory. The information has to be converted to a different structure in

order to be sent through SNMP. If you use SNMP, you can exclude certain variables using the **snmp−server view** command. See IP Simple Network Management Protocol (SNMP) Causes High CPU Utilization for more information.

Virtual EXEC

The virtual EXEC process handles virtual type terminal (vty) lines, such as Telnet sessions on the router. If you've issued a command that generates long output (such as **show tech−support**), or if the debug output has been redirected to the vty (using the **terminal monitor** or the **no logging console** command), the amount of CPU resources used by the virtual EXEC process increases.

TCP Timer

When the TCP timer process uses a lot of CPU resources, it indicates that there are too many TCP connection endpoints. This can happen in data−link switching (DLSw) or remote source−route bridging (RSRB) environments with many peers.

VTEMPLATE Backgr

A virtual template has to be cloned for each new virtual access interface whenever a new user gets connected to the router or access server. The CPU utilization in VTEMPLATE Backgr process can get extremely high if the number of users is large. This can be avoided by configuring pre−cloning of the virtual template. For further information, see the Session Scalability Enhancements document on CCO.

Other Processes

If any other process is consuming a lot of CPU resources, and there is no indication of any problem in logged messages, then the problem could possibly be caused by a bug in the IOS. Using the Bug Navigator, run a search for the specified process to see if any bugs have been reported.

Note: If you need a help from a Customer Support Engineer in the Cisco Technical Assistance Center, please capture the **show tech−support** command output (from enable mode) before contacting Cisco TAC. Also, if the high CPU utilization is caused by a process, please capture the **show stacks** *pid* command output (where pid is the process ID of the process causing the high CPU utilization). If the problem is caused by a bug in IOS, please relay the bug ID to the Cisco Customer Support Engineer handling the case.

Commands for Obtaining More Information

The following commands provide more information about the problem:

- **show processes cpu**
- **show interfaces**
- **show interfaces switching**
- **show interfaces stat**
- **show align**
- **show version**
- **show log**

If the router is completely inaccessible, first power−cycle it. Then, periodically collect the output of the commands above, except for the **show log** command, whose messages should be logged on a syslog server. The interval for collecting output should be five minutes. You can collect the data manually or automatically, using this UNIX shell script. You can also collect data using HTTP or SNMP. For details about configuring

HTTP and SNMP on a Cisco router, check the Cisco IOS Software Configuration document.

show processes cpu Command

The header of the **show processes cpu** command looks like the following:

```
CPU utilization for five seconds: X%/Y%; one minute: Z%; five minutes: W%
 PID  Runtime(ms)  Invoked  uSecs   5Sec   1Min   5Min TTY Process
```

The following table describes the fields in the header:

Field	Description
X	Average total utilization during last five seconds
Y	Average utilization due to interrupts,during last five seconds
X–Y	This represents percentage of traffic being process switched
Z	Average total utilization during last minute *
W	Average total utilization during last five minutes *
PID	Process ID
Runtime	CPU time the process has used (in milliseconds)
Invoked	Number of times a process has been called
uSecs	Microseconds of CPU time for each invocation
5Sec	CPU utilization by task in the last 5 seconds
1Min	CPU utilization by task in the last minute *
5Min	CPU utilization by task in the last 5 minutes *
TTY	Terminal that controls the process
Process	Name of process

* Values don't represent an arithmetical average, but an exponentially decayed average. Thus, more recent values have more influence on the calculated average.

Follow this link for a detailed explanation of **show processes cpu** command.

Note: Total CPU utilization shouldn't be used as a measure of the ability of the router to switch more packets. On Cisco 7500 routers, Versatile Interface Processors (VIP) and Route/Switch Processors (RSP) don't report linear CPU utilization. Close to half of the switching packet–per–second power comes after 90 – 95 percent CPU utilization.

show interfaces Command

The command reference contains a detailed explanation of the **show interfaces** command.

show interfaces switching Command

This command is used for determining active switching paths on interfaces. For more information about switching paths in Cisco IOS Software, see the Configuring Switching Paths document.

Cisco – Troubleshooting High CPU Utilization on Cisco Routers

Let's look at sample output of the **show interfaces switching** command for one interface:

```
RouterA#show interfaces switching
  Ethernet0
            Throttle count         0
              Drops        RP      0        SP         0
       SPD Flushes       Fast      0       SSE         0
       SPD Aggress       Fast      0
       SPD Priority     Inputs     0      Drops        0

            Protocol       Path   Pkts In   Chars In   Pkts Out   Chars Out
              Other      Process        0          0        595       35700
            Cache misses              0
                  Fast        0          0          0           0
             Auton/SSE        0          0          0           0
               IP     Process        4        456          4         456
            Cache misses              0
                  Fast        0          0          0           0
             Auton/SSE        0          0          0           0
              IPX     Process        0          0          2         120
            Cache misses              0
                  Fast        0          0          0           0
             Auton/SSE        0          0          0           0
      Trans. Bridge   Process        0          0          0           0
            Cache misses              0
                  Fast       11        660          0           0
             Auton/SSE        0          0          0           0
            DEC MOP    Process        0          0         10         770
            Cache misses              0
                  Fast        0          0          0           0
             Auton/SSE        0          0          0           0
              ARP     Process        1         60          2         120
            Cache misses              0
                  Fast        0          0          0           0
             Auton/SSE        0          0          0           0
              CDP     Process      200      63700        100       31183
            Cache misses              0
                  Fast        0          0          0           0
             Auton/SSE        0          0          0           0
```

The output lists the switching paths for all protocols configured on the interface, so you can easily see what kind and the amount of traffic going through the router. The following table explains the output fields.

Field	Definition
Process	Processed packets. These can be packets destined for the router, or packets for which there was no entry in the fast switching cache.
Cache misses	Packets for which there was no entry in fast switching cache. The first packet for this destination (or flow – depends on the type of fast switching configured) will be processed. All subsequent packets will be fast switched, unless fast switching is explicitly disabled on the outgoing interface.
Fast	Fast switched packets. Fast switching is enabled by default.
Auton/SSE	Autonomous switched, silicon switched or distributed switched packets. Available only on Cisco 7000 series routers with a Switch Processor or Silicon Switch Processor (for autonomous switching or silicon switching, respectively), or

	on Cisco 7500 series routers with a VIP (for distributed switching).

show interfaces stat Command

This command is a summarized version of the **show interfaces switching** command. Here's sample output for one interface:

```
RouterA#show interfaces stat
Ethernet0
          Switching path    Pkts In    Chars In    Pkts Out   Chars Out
              Processor       52077    12245489       24646     3170041
            Route cache           0           0           0           0
      Distributed cache           0           0           0           0
                  Total       52077    12245489       24646     3170041
```

The output of the **show interfaces stat** command is different for different platforms, depending on available and configured switching paths.

show align Command

This command is available only on reduced instruction set computing (RISC) processor−based platforms. On these platforms, the CPU can correct for mis−aligned memory reads or writes. Let's look at sample output:

```
Alignment data for:
4500 Software (C4500-DS40-M), Version mis-aligned RELEASE SOFTWARE (fc1)
Compiled Tue 31-Mar-98 15:05 by jdoe

Total Corrections 33911, Recorded 2, Reads 33911, Writes 0

Initial Initial
Address Count Access Type Traceback
40025F4D 15561 16bit read 0x606F4A7C 0x601C78F8 0x6012FE94 0x600102C0
40025F72 18350 32bit read 0x606FB260 0x6013113C 0x600102C0 0x60010988
```

show version Command

For the purpose of tracking high CPU utilization problems, the important part of this command output is the Cisco IOS Software version, platform, CPU type, and the uptime of the router. This command reference gives a detailed explanation of the **show version** command.

show log Command

This command shows the contents of buffered log messages. For more information about logging system messages, check the Log System Error Messages section of the Troubleshooting the Router Configuration guide.

UNIX Shell Script for Periodically Collecting Data

This appendix describes a simple script for periodically capturing data from the router. The core of the script is the following line:

```
(echo "show version") | telnet 192.168.1.1
```

The command in parentheses is executed in sub–shell and the output is sent to a Telnet session. Following is a sample script for capturing the output from the **show version** and **show processes cpu** commands:

```
#!/opt/local/bin/bash

###########################################
# Router's IP address
#
IP_ADDRESS='10.200.40.53'


# Directory where the log files will be stored
#
DIR=/var/log/router

######################################

if [ ! -e $DIR ]
then
  mkdir $DIR
fi

# Tag specification: mmddhhmm
DATE=`date +'%m%d'`
TIME=`date +'%H%M'`
TAG=$DATE$TIME

# Collect data from the router
(echo "foo";\
echo "bar";\
echo "term len 0";\
echo "show version";\
echo "show processes cpu";\
echo "term len 15";\
echo "show memory summary";\
echo "q";\
sleep 30)|telnet $IP_ADDRESS > $DIR/info.$TAG 2>$DIR/info.$TAG.msg
```

Note: In this script all data, including the password, are sent in a clear text format.

In the first section, you need to specify the IP address and the destination directory for log files. The second section contains the actual commands that are sent to the router. The first is the username, then the password, and so on. A trick for capturing only the first lines of output of certain commands is included. Terminal length is set to something short (15 in this case), and the "q" character is sent only by prompt.

If data is collected periodically, the output of **show version** shows if the problem has a periodical nature, for example, if it appears always at a certain time of day or on a particular day of the week. If you need to collect the output of more commands, they can be added to the script in the same manner as those shown in the example. If you need to truncate the output sent to the file, first increase the sleep period (the sleep command in parenthesis).

Run this script every five minutes if the high CPU utilization problem appears often and doesn't last long. Otherwise, you can run it every 15 or 30 minutes. For ease of use, save the script in a file such as */usr/bin/router–script*. Then, to run it every 5 minutes, add the following line to the /etc/crontab file:

```
*/5      *      *      *      *      /usr/bin/router-script
```

Restart the cron server. If you don't have the authority to change the */etc/crontab* file, run the script in a separate process as follows:

Cisco – Troubleshooting High CPU Utilization on Cisco Routers

```
                while [ 1 ]; do ./router-script ; sleep 300; done &
```

Sample IP Packet Debugging Session

Configured logging destinations should be checked first by issuing the **show logging** command:

```
grooverider#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level debugging, 52 messages logged
    Monitor logging: level debugging, 0 messages logged
    Buffer logging: level debugging, 148 messages logged
    Trap logging: level informational, 64 message lines logged
        Logging to 192.168.100.100, 3 message lines logged
        Logging to 192.168.200.200, 3 message lines logged
  --More--
```

Disable all logging destinations except logging buffer, and clear logging buffer:

```
grooverider#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
grooverider(config)#no logging console
grooverider(config)#no logging monitor
grooverider(config)#no logging 192.168.100.100
grooverider(config)#no logging 192.168.200.200
grooverider(config)#^Z
grooverider#clear logging
Clear logging buffer [confirm]
```

A debugging session can now be started:

```
grooverider#debug ip packet detail
IP packet debugging is on (detailed)
```

Debugging shouldn't last more then 3–5 seconds. It can be stopped by issuing the **undebug all** exec command:

```
grooverider#undebug all
All possible debugging has been turned off
```

Results can be checked by issuing the **show logging** exec command:

```
grooverider#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: disabled
    Monitor logging: disabled
    Buffer logging: level debugging, 145 messages logged
    Trap logging: level informational, 61 message lines logged

Log Buffer (64000 bytes):

*Mar  3 03:43:27.320: IP: s=192.168.40.53 (Ethernet0/1), d=144.254.2.204
   (Ethernet0/0), g=10.200.40.1, len 100, forward
*Mar  3 03:43:27.324: ICMP type=8, code=0
*Mar  3 03:43:27.324: IP: s=192.168.40.53 (Ethernet0/1), d=144.254.2.205
   (Ethernet0/0), g=10.200.40.1, len 100, forward
*Mar  3 03:43:27.324: ICMP type=8, code=0
*Mar  3 03:43:27.328: IP: s=192.168.40.53 (Ethernet0/1), d=144.254.2.206
   (Ethernet0/0), g=10.200.40.1, len 100, forward
*Mar  3 03:43:27.328: ICMP type=8, code=0
...
```

The log shows that:

- A packet has been received every four milliseconds
- The source ip address is 192.168.40.53
- The packets have come in on interface Ethernet0/1
- The packets have different destination IP addresses
- They have been sent out on interface Ethernet0/0
- The next–hop IP address is 10.200.40.1
- The packets were ICMP requests (type=8).

In this case, it can be seen that the high CPU utilization in **IP Input** process has been caused by a ping flood from IP address 192.168.40.53.

SYN floods can easily be detected this way as well, because SYN flag presence is indicated in the debugging output:

```
*Mar 3 03:54:40.436: IP: s=192.168.40.53 (Ethernet0/1), d=144.254.2.204
   (Ethernet0/0), g=10.200.40.1, len 44, forward
*Mar 3 03:54:40.440: TCP src=11004, dst=53,
   seq=280872555, ack=0, win=4128 SYN
```

Related Information

- **Technical Support – Cisco Systems**