Troubleshooting Tips for the Cisco 7500 Router

This section provides information on how to troubleshoot common problems involving IOS on the Cisco 7500 platform. Some of these common scenarios include high CPU utilization, input drops, ignores, and output drops.

High CPU Utilization

Example 6-8 shows a portion of the output from the IOS **show process cpu** command on a router that is experiencing high CPU utilization.

Example 6-8. *show process cpu* Command Output Revealing High CPU Utilization

Router#**show process cpu** CPU utilization for five seconds: 90%/80%; one minute: 60%; five minutes: 40% PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process 1 1356 2991560 0 0.00% 0.00% 0.00% 0 BGP Router 2 100804 7374 13670 0.00% 0.02% 0.00% 0 Check heaps 3 0 1 0 0.00% 0.00% 0.00% 0 Pool Manager 4 0 2 0 0.00% 0.00% 0.00% 0 Timers 5 6044 41511000 0.00% 0.00% 0.00% 0 OIR Handler 6 0 1 0 0.00% 0.00% 0.00% 0 IPC Zone Manager 7 0 1 0 0.00% 0.00% 0.00% 0 IPC Realm Manager 6 7700 36331 211 8% 0.00% 0.00% 0 IP Input

Example 6-8 shows that for the last 5 seconds, the average CPU utilization was 90 percent of the total CPU capacity and 80 percent of the total CPU capacity (88.8 percent of the CPU busy time) was spent processing interrupts. Furthermore, the rolling averages for 1 minute and 5 minutes are 60 percent utilization and 40 percent utilization, respectively. Because the largest utilization percentage is contributed by interrupt processing and the long-term utilization indicators are more moderate, it's quite possible that this router is just receiving large bursts of packets to switch.

It is also worth noting, however, that on 7500 routers and others that use the MIPS CPU's, interrupt processing CPU utilization can also run higher than normal due to an error called an *alignment error*. Alignment errors occur when the program running on the CPU attempts to access a memory value at an address that violates the memory alignment requirements of the CPU. On MIPS CPUs, 16-bit values must begin at a memory address divisible by 2, 32-bit values must begin at a memory address divisible by 4, and so on. If IOS attempts to access data at an address that violates these restrictions, the CPU generates an *exception*(basically an error) and calls a special IOS function that retrieves the data in segments that don't violate the restrictions. This exception function adds many more instructions and more CPU time to an otherwise simple operation of accessing a data item. For this reason, alignment errors can have a significant negative impact on performance by consuming extra CPU cycles. All IOS alignment errors are logged and can be inspected by issuing the IOS **show align** CLI command.

Consistently high CPU utilization by processes should be investigated to find out which process or processes are consuming the CPU. From the **show process cpu** output, it's easy to find out which processes are the biggest CPU users. Process names are fairly intuitive. For example, the IP Input process is responsible for processing all IP packets at process-level. If you see high CPU by IP Input, this likely means many of the IP packets coming into the router are taking the process switched path. A useful command to determine if the IP packets coming into the router are indeed being process switched is the **show interface stat** command, as demonstrated in Example 6-9.

Example 6-9. *show interface stat* Command Output to Determine Packet Switching Path

Router#**sh int stat** POS8/1/0 Switching path Pkts In Chars In Pkts Out Chars Out Processor 2 4498 15586 23285531 Route cache 0 0 0 0 Distributed cache 60448 57737596 30772 1357152 Total 60450 57742094 46358 24642683

The following list highlights the columns in the output from Example 6-9 that you should monitor to determine packet traffic on a given router interface:

- **Pkts In—**

  The number of packets that came in the interface and the switching path they took. Example 6-9 shows that a total of 60,450 packets arrived into the interface POS8/1/0, two of those packets went to process-level, 0 took the fast/optimum/CEF path, and 60,448 packets that came in were distribute switched.

- **Chars In—**

  The number of bytes that arrived into the interface. Chars In/Pkts In can give an estimate of the average packet size of the packets that come into the interface.

- **Pkts Out—**

  The number of packets transmitted on this interface and the switching path they took. The packets transmitted out an interface might have arrived from many interfaces. Example 6-9 shows that a total of 15,586 packets that were transmitted out POS8/1/0 were process-switched and 30,772 packets took the distribute switched path.

The majority of packets received on any interface should be switched by the fastest switching method supported on the interface. Control packets (for example, routing updates or keepalives) and packets destined to the router are accounted as *process switched* inbound. When interpreting the output of the **show interface stat** command, remember packets received in an interface typically are transmitted out another interface and packets transmitted out an interface came in on another interface. Therefore, look at the **show interface stat** command for all interfaces to get an estimate of the overall traffic flow.

The following are some common reasons for high CPU (this list is by no means comprehensive):

- **Configuration—**

  Make sure all interfaces are configured for the fastest switching mode available.

- **Broadcast Storm—**

  All broadcast packets are sent to the process level for switching, so a broadcast storm can strain a router's CPU. A broadcast storm can occur due to a layer 2 issue (for example, Spanning Tree) or misconfigured hosts.

- **Denial of Service attack—**

  This is a very broad category. The *ping attack* is a common one where a malicious user can send a high stream of ping packets destined to the router. Make sure access lists are configured to prevent this. Committed access rate (CAR) is a rate-limiting feature that can also be used. Refer to the Cisco IOS documentation for more information on how to configure CAR.

- **Routing loop—**

  A routing loop can cause IP packets to terminate on the router because their TTL expires while looping. A constantly increasing *bad hop count* counter in the output of **show ip traffic** reflects this condition.

Input Drops

On the 75*xx* platform, the *input drop* counter in the **show interface** output is only incremented when packets destined to the process switched path are dropped. Some input drops are to be expected. However, the ratio of input drops to the total packets input is the important statistic. Although it is difficult to say when to start worrying about input drops, degrading network performance is usually a sign that it's time to begin looking at the number of input drops.

The default input hold queue size is 75. However, increasing it to a higher number might not be the solution. The root cause of why packets are going to process switching must be investigated. Some of the same reasons listed for high CPU utilization could also be responsible for input drops. If the *no buffer* counter in the **show interface**output is also incrementing along with input drops, it's an indication of buffer starvation—not enough buffers are available to accommodate the packets. The IOS **show buffers** command can be a useful tool to determine which buffer pool is being exhausted.

Ignores

An increasing *ignore* counter in the **show interface** output is indicative of a MEMD buffer shortage. Network traffic bursts can also contribute to ignores. Like input drops, the ratio of ignores to the total packet input count is the important metric, not the number of ignores alone.

Packets received on a high-speed interface and transmitted on a slow interface can also contribute to ignored packets. With this scenario, MEMD buffers from the input interface's pool can be tied up for long periods of time waiting on the transmit queue of the slow interface. This condition deprives other interfaces of buffers that use the same pool. One possible solution is to configure all the interfaces on the router for the same MTU size. With a common MTU size, there is one large MEMD buffer pool that all interfaces can draw upon, potentially providing more buffers. Configuring VIP-based interfaces for distributed switching, which allows receive side buffering, can also help alleviate ignores.

Output Drops

The *output drop* counter in the **show interface** output can be incremented for the following reasons:

- **Process switching—**

  A packet cannot be enqueued on the output hold queue because the hold queue is full.

- **Fast switching—**

  A packet cannot be enqueued on the interface's transmit queue because the transmit queue is full. The instantaneous value of an interface's transmit queue can be seen in the output of the **show controller cbus** command. A consistently low value indicates that the transmit queue is congested.

Increasing the output hold-queue limit won't help reduce output drops if the majority of packets are being dropped by fast switching. Like other drop counters, some output drops are to be expected. The absolute number of drops is not as important as the ratio of output drops to total packets output.

< BACK | Make Note | Bookmark | CONTINUE >

Index terms contained in this section