Catalyst 6500/6000 Switch High CPU Utilization

Document ID: 63992

Contents

Introduction

This document describes causes of high CPU utilization on Cisco Catalyst 6500/6000 Series Switches and Virtual Switching System (VSS) 1440 based systems. Like Cisco routers, switches use the **show processes cpu** command in order to show CPU utilization for the switch supervisor engine processor. However, due to the differences in architecture and forwarding mechanisms between Cisco routers and switches, the typical output of the **show processes cpu** command differs significantly. The meaning of the output differs as well. This document clarifies these differences and describes CPU utilization on the switches and how to interpret the **show processes cpu** command output.

Note: In this document, the words "switch" and "switches" refer to the Catalyst 6500/6000 Switches.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on the software and hardware versions for Catalyst 6500/6000 Switches and Virtual Switching System (VSS) 1440 based systems.

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, make sure that you understand the potential impact of any command.

Note: The supported software for Virtual Switching System (VSS) 1440 based systems is Cisco IOS® Software Release 12.2(33)SXH1 or later.

Conventions

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

Difference Between CatOS and Cisco IOS System Software

Catalyst OS (CatOS) on the Supervisor Engine and Cisco IOS® Software on the Multilayer Switch Feature Card (MSFC) (Hybrid): You can use a CatOS image as the system software to run the supervisor engine on Catalyst 6500/6000 Switches. If the optional MSFC is installed, a separate Cisco IOS Software image is used to run the MSFC.

Cisco IOS Software on both the Supervisor Engine and MSFC (Native): You can use a single Cisco IOS Software image as the system software to run both the supervisor engine and MSFC on Catalyst 6500/6000 Switches.

Note: Refer to Comparison of the Cisco Catalyst and Cisco IOS Operating Systems for the Cisco Catalyst 6500 Series Switch for more information.

Understand CPU Utilization on Catalyst 6500/6000 Switches

Cisco software−based routers use software in order to process and route packets. CPU utilization on a Cisco router tends to increase as the router performs more packet processing and routing. Therefore, the **show processes cpu** command can provide a fairly accurate indication of the traffic processing load on the router.

Catalyst 6500/6000 Switches do not use the CPU in the same way. These switches make forwarding decisions in hardware, not in software. Therefore, when the switches make the forwarding or switching decision for most frames that pass through the switch, the process does not involve the supervisor engine CPU.

On Catalyst 6500/6000 Switches, there are two CPUs. One CPU is the supervisor engine CPU, which is called the Network Management Processor (NMP) or the Switch Processor (SP). The other CPU is the Layer 3 routing engine CPU, which is called the MSFC or the Route Processor (RP).

The SP CPU performs functions that include:

- Assists in MAC address learning and aging

  Note: MAC address learning is also called path setup.
- Runs protocols and processes that provide network control

  Examples include Spanning Tree Protocol (STP), Cisco Discovery Protocol (CDP), VLAN Trunk Protocol (VTP), Dynamic Trunking Protocol (DTP), and Port Aggregation Protocol (PAgP).
- Handles network management traffic that is destined to the CPU of the switch

  Examples include Telnet, HTTP, and Simple Network Management Protocol (SNMP) traffic.

The RP CPU performs functions that include:

- Builds and updates the Layer 3 routing and Address Resolution Protocol (ARP) tables
- Generates the Cisco Express Forwarding (CEF) Forwarding Information Base (FIB) and adjacency tables, and downloads the tables into the Policy Feature Card (PFC)
- Handles network management traffic that is destined to the RP

  Examples include Telnet, HTTP, and SNMP traffic.

Situations and Features That Trigger Traffic to Go to Software

Packets That Are Destined to the Switch

Any packet that is destined to the switch goes to software. Such packets include:

- Control packets

  Control packets are received for STP, CDP, VTP, Hot Standby Router Protocol (HSRP), PAgP, Link Aggregation Control Protocol (LACP), and UniDirectional Link Detection (UDLD).
- Routing protocol updates

  Examples of these protocols are Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP), Border Gateway Protocol (BGP), and Open Shortest Path First Protocol (OSPF Protocol).
- SNMP traffic that is destined to the switch
- Telnet and Secure Shell Protocol (SSH) traffic to the switch.

High CPU ultilization due to SSH is seen as:

```
00:30:50.793 SGT Tue Mar 20 2012

CPU utilization for five seconds: 83%/11%; one minute: 15%; five minutes: 8%

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

   3        6468      8568        754 69.30%  7.90%  1.68%   1 SSH Process
```

Include these commands in the EEM script in order to verify the number of SSH sessions established when the CPU goes high:

  - show users
  - show line
- ARP responses to ARP requests

Packets and Conditions That Require Special Processing

This list provides specific packet types and conditions that force packets to be handled in software:

- Packets with IP options, an expired Time to Live (TTL), or non−Advanced Research Projects Agency (ARPA) encapsulation
- Packets with special handling, such as tunneling
- IP fragmentation
- Packets that require Internet Control Message Protocol (ICMP) messages from the RP or SP
- Maximum transmission unit (MTU) check failure
- Packets with IP errors, which include IP checksum and length errors
- If the input packets return a bit error (such as the single−bit error (SBE)), the packets are sent to the CPU for software processing and are corrected. The system allocates a buffer for them and uses the CPU resource to correct it.
- When PBR and reflexive access list are in the path of a traffic flow, the packet is software switched, which requires an additional CPU cycle.
- Adjacency same interface
- Packets that fail the Reverse Path Forwarding (RPF) check **rpf−failure**
- Glean/receive

  Glean refers to packets that require ARP resolution, and receive refers to packets that fall in the receive case.
- Internetwork Packet Exchange (IPX) traffic that is software−switched on the Supervisor Engine 720 in both Cisco IOS Software and CatOS

  IPX traffic is also software−switched on the Supervisor Engine 2/Cisco IOS Software, but the traffic is hardware−switched on the Supervisor Engine 2/CatOS. IPX traffic is hardware−switched on the Supervisor Engine 1A for both operating systems.
- AppleTalk traffic
- Hardware resources full conditions

  These resources include FIB, content−addressable memory (CAM), and ternary CAM (TCAM).

ACL−Based Features

- Access control list (ACL)−denied traffic with the ICMP unreachables feature turned on

  Note: This is the default.

Some ACL−denied packets are leaked to the MSFC if IP unreachables are enabled. Packets that require ICMP unreachables are leaked at a user−configurable rate. By default, the rate is 500 packets per second (pps).

- IPX filtering on the basis of unsupported parameters, such as source host

  On the Supervisor Engine 720, the process of Layer 3 IPX traffic is always in software.
- Access control entries (ACEs) that require logging, with the **log** keyword

  This applies to ACL log and VLAN ACL (VACL) log features. ACEs in the same ACL that do not require logging still process in hardware. The Supervisor Engine 720 with PFC3 supports the rate limit of packets that are redirected to the MSFC for ACL and VACL logging. The Supervisor Engine 2 supports the rate limit of packets that are redirected to the MSFC for VACL logging. Support for ACL logging on the Supervisor Engine 2 is scheduled for the Cisco IOS Software Release 12.2S branch.
- Policy−routed traffic, with use of **match length**, **set ip precedence**, or other unsupported parameters

  The **set interface** parameter has support in software. However, the **set interface null 0** parameter is an exception. This traffic is handled in hardware on the Supervisor Engine 2 with PFC2 and the Supervisor Engine 720 with PFC3.
- Non−IP and non−IPX router ACLs (RACLs)

  Non−IP RACLs apply to all supervisor engines. The non−IPX RACLs apply to the Supervisor Engine 1a with PFC and the Supervisor Engine 2 with PFC2 only.
- Broadcast traffic that is denied in an RACL
- Traffic that is denied in a unicast RPF (uRPF) check, ACL ACE

  This uRPF check applies to the Supervisor Engine 2 with PFC2 and Supervisor Engine 720 with PFC3.
- Authentication proxy

  Traffic that is subject to authentication proxy can be rate−limited on the Supervisor Engine 720.
- Cisco IOS Software IP Security (IPsec)

  Traffic that is subject to Cisco IOS encryption can be rate−limited on the Supervisor Engine 720.

NetFlow−Based Features

The NetFlow−based features that this section describes apply to the Supervisor Engine 2 and Supervisor Engine 720 only.

- NetFlow−based features always need to see the first packet of a flow in software. Once the first packet of the flow reaches software, subsequent packets for the same flow are hardware−switched.

  This flow arrangement applies to reflexive ACLs, Web Cache Communication Protocol (WCCP), and Cisco IOS Server Load Balancing (SLB).

  Note: On the Supervisor Engine 1, reflexive ACLs rely on dynamic TCAM entries to create hardware shortcuts for a particular flow. The principle is the same: the first packet of a flow goes to software. Subsequent packets for that flow are hardware−switched.
- With the TCP Intercept feature, the three−way handshake and session close are handled in software. The rest of the traffic is handled in hardware.

  Note: Synchronize (SYN), SYN acknowledge (SYN ACK), and ACK packets comprise the three−way handshake. Session close occurs with finish (FIN) or reset (RST).

- With Network Address Translation (NAT), traffic is handled in this way:
    - On the Supervisor Engine 720:

      Traffic that requires NAT is handled in hardware after the initial translation. Translation of the first packet of a flow occurs in software, and subsequent packets for that flow are hardware−switched. For TCP packets, a hardware shortcut is created in the NetFlow table after completion of the TCP three−way handshake.
    - On the Supervisor Engine 2 and Supervisor Engine 1:

      All traffic that requires NAT is software−switched.
- Context−based Access Control (CBAC) uses NetFlow shortcuts in order to classify traffic that requires inspection. Then, CBAC sends only this traffic to software. CBAC is a software−only feature; traffic that is subject to inspection is not hardware−switched.

  Note: Traffic that is subject to inspection can be rate−limited on the Supervisor Engine 720.

Multicast Traffic

- Protocol Independent Multicast (PIM) snooping
- Internet Group Management Protocol (IGMP) snooping (TTL = 1)

  This traffic is indeed destined to the router.
- Multicast Listener Discovery (MLD) snooping (TTL = 1)

  This traffic is indeed destined to the router.
- FIB miss
- Multicast packets for registration that have direct connection to the multicast source

  These multicast packets are tunneled to the rendezvous point.
- IP version 6 (IPv6) multicast

Other Features

- Network−Based Application Recognition (NBAR)
- ARP Inspection, with CatOS only
- Port Security, with CatOS only
- DHCP snooping

IPv6 Situations

- Packets with a hop−by−hop option header
- Packets with the same destination IPv6 address as that of routers
- Packets that fail the scope enforcement check
- Packets that exceed the MTU of the output link
- Packets with a TTL that is less than or equal to 1
- Packets with an input VLAN that equals the output VLAN
- IPv6 uRPF

  Software performs this uRPF for all packets.
- IPv6 reflexive ACLs

  Software handles these reflexive ACLs.

- 6to4 prefixes for IPv6 Intra−Site Automatic Tunnel Addressing Protocol (ISATAP) tunnels

  Software handles this tunneling. All other traffic that enters an ISATAP tunnel is hardware−switched.

LCP Schedular and DFC Module

In a Distributed Forwarding Card (DFC), the `lcp schedular` process that runs on a high CPU is not an issue and does not pose any problem to the operation. The LCP schedular is part of the firmware code. On all modules that do not require a DFC, the firmware runs on a specific processor called the Line Card Processor (LCP). This processor is used to program the ASIC hardware and to communicate to the central supervisor module.

When the `lcp schedular` is initiated, it makes use of all the process time available. But when a new process needs processor time, `lcp schedular` frees up process time for the new process. There is no impact to the performance of the system with regard to this high CPU utilization. The process simply grabs all unused CPU cycles, as long as no higher priority process requires them.

```
DFC#show process cpu

PID  Runtime(ms)  Invoked  uSecs    5Sec   1Min   5Min TTY Process
  22           0         1      0   0.00%  0.00%  0.00%   0 SCP ChilisLC Lis
  23           0         1      0   0.00%  0.00%  0.00%   0 IPC RTTYC Messag
  24           0         9      0   0.00%  0.00%  0.00%   0 ICC Slave LC Req
  25           0         1      0   0.00%  0.00%  0.00%   0 ICC Async mcast
  26           0         2      0   0.00%  0.00%  0.00%   0 RPC Sync
  27           0         1      0   0.00%  0.00%  0.00%   0 RPC rpc−master
  28           0         1      0   0.00%  0.00%  0.00%   0 Net Input
  29           0         2      0   0.00%  0.00%  0.00%   0 Protocol Filteri
  30           8       105     76   0.00%  0.00%  0.00%   0 Remote Console P
  31          40      1530     26   0.00%  0.00%  0.00%   0 L2 Control Task
  32          72       986     73   0.00%  0.02%  0.00%   0 L2 Aging Task
  33           4        21    190   0.00%  0.00%  0.00%   0 L3 Control Task
  34          12       652     18   0.00%  0.00%  0.00%   0 FIB Control Task
  35        9148       165  55442   1.22%  1.22%  1.15%   0 Statistics Task
  36           4       413      9   0.00%  0.00%  0.00%   0 PFIB Table Manag
  37      655016  64690036     10  75.33% 77.87% 71.10%   0 lcp schedular
  38           0       762      0   0.00%  0.00%  0.00%   0 Constellation SP
```

Common Causes and Solutions for High CPU Utilization Issues

IP Unreachables

When an access group denies a packet, the MSFC sends ICMP unreachable messages. This action occurs by default.

With the default enablement of the **ip unreachables** command, the supervisor engine drops most of the denied packets in hardware. Then, the supervisor engine sends only a small number of packets, a maximum of 10 pps, to the MSFC for drop. This action generates ICMP−unreachable messages.

The drop of denied packets and generation of ICMP−unreachable messages imposes a load on the MSFC CPU. In order to eliminate the load, you can issue the **no ip unreachables** interface configuration command. This command disables ICMP−unreachable messages, which allows the drop in hardware of all access group−denied packets.

ICMP−unreachable messages are not sent if a VACL denies a packet.

NAT Translations

NAT utilizes both hardware and software forwarding. The initial establishment of the NAT tranlsations must be done in software and further forwarding is done with hardware. NAT also utilizes the Netflow table (128 KB maximum). Therefore, if the Netflow table is full, the switch will also start to apply NAT forwarding via software. This normally happens with high traffic bursts and will cause an increase on the CPU of 6500.

Use of CEF FIB Table Space in the Flow Cache Table

The Supervisor Engine 1 has a Flow Cache Table that supports 128,000 entries. However, on the basis of the efficiency of the hashing algorithm, these entries range from 32,000 to 120,000. On the Supervisor Engine 2, the FIB table is generated and programmed into the PFC. The table holds as many as 256,000 entries. The Supervisor Engine 720 with PFC3−BXL supports up to 1,000,000 entries. Once this space is exceeded, the packets become switched in software. This can cause high CPU utilization on the RP. In order to check the number of routes in the CEF FIB table, use these commands:

```
Router#show processes cpu
CPU utilization for five seconds:  99.26%
                      one minute: 100.00%
                    five minutes: 100.00%

PID Runtime(ms) Invoked    uSecs    5Sec    1Min    5Min    TTY Process
--- ----------- ---------- -------- ------- ------- ------- --- ---------------
1   0           0          0           0.74%   0.00%   0.00% -2  Kernel and Idle
2   2           245        1000        0.00%   0.00%   0.00% -2  Flash MIB Updat
3   0           1          0           0.00%   0.00%   0.00% -2  L2L3IntHdlr
4   0           1          0           0.00%   0.00%   0.00% -2  L2L3PatchRev
5   653         11737      1000        0.00%   0.00%   0.00% -2  SynDi

!--- Output is suppressed.

26  10576       615970     1000        0.00%   0.00%   0.00% 0   L3Aging
27  47432       51696      8000        0.02%   0.00%   0.00% 0   NetFlow
28  6758259     1060831    501000     96.62%  96.00%  96.00% 0   Fib
29  0           1          0           0.00%   0.00%   0.00% -2  Fib_bg_task

!--- Output is suppressed.


CATOS% show mls cef
Total L3 packets switched:           124893998234
Total L3 octets switched:          53019378962495
Total route entries:                       112579
  IP route entries:                        112578
  IPX route entries:                            1
  IPM route entries:                            0
IP load sharing entries:                      295
IPX load sharing entries:                       0
Forwarding entries:                        112521
Bridge entries:                                56
Drop entries:                                   2


IOS% show ip cef summary
IP Distributed CEF with switching (Table Version 86771423), flags=0x0
  112564 routes, 1 reresolve, 0 unresolved (0 old, 0 new)
  112567 leaves, 6888 nodes, 21156688 bytes, 86771426
inserts, 86658859
invalidations
  295 load sharing elements, 96760 bytes, 112359 references
  universal per-destination load sharing algorithm, id 8ADDA64A
  2 CEF resets, 2306608 revisions of existing leaves
  refcounts:  1981829 leaf, 1763584 node

!--- You see these messages if the TCAM space is exceeded:

%MLSCEF-SP-7-FIB_EXCEPTION: FIB TCAM exception, Some entries will be software switched
%MLSCEF-SP-7-END_FIB_EXCEPTION: FIB TCAM exception cleared, all CEF entries will be
  hardware switched
```

On the Supervisor Engine 2, the number of FIB entries reduces to half if you have configured RPF check on the interfaces. This configuration can lead to the software switch of more packets and, consequently, high CPU utilization.

In order to resolve the high CPU utilization issue, enable route summarization. Route summarization can minimize the latency in a complex network by reducing processor workloads, memory requirements, and bandwidth demand.

Refer to Understanding ACL on Catalyst 6500 Series Switches for additional information about TCAM utilization and optimization.

Optimized ACL Logging

Optimized ACL Logging (OAL) provides hardware support for ACL logging. Unless you configure OAL, the process of packets that require logging takes place completely in software on the MSFC3. OAL permits or drops packets in hardware on the PFC3. OAL uses an optimized routine to send information to the MSFC3 in order to generate the logging messages.

Note: For information about OAL, refer to the *Optimized ACL Logging with a PFC3* section of Understanding Cisco IOS ACL Support.

Rate Limit of Packets to the CPU

On the Supervisor Engine 720, rate limiters can control the rate at which packets can go to software. This rate control helps prevent denial−of−service attacks. You can also use a few of these rate limiters on the Supervisor Engine 2:

```
Router#show mls rate-limit
   Rate Limiter Type       Status     Packets/s   Burst
---------------------   ----------    ---------   -----
         MCAST NON RPF   Off                  -       -
        MCAST DFLT ADJ   On              100000     100
      MCAST DIRECT CON   Off                  -       -
        ACL BRIDGED IN   Off                  -       -
       ACL BRIDGED OUT   Off                  -       -
           IP FEATURES   Off                  -       -
          ACL VACL LOG   On                2000       1
           CEF RECEIVE   Off                  -       -
             CEF GLEAN   Off                  -       -
      MCAST PARTIAL SC   On              100000     100
        IP RPF FAILURE   On                 500      10
           TTL FAILURE   Off                  -       -
ICMP UNREAC. NO-ROUTE   On                 500      10
ICMP UNREAC. ACL-DROP   On                 500      10
         ICMP REDIRECT   Off                  -       -
           MTU FAILURE   Off                  -       -
           LAYER_2 PDU   Off                  -       -
            LAYER_2 PT   Off                  -       -
             IP ERRORS   On                 500      10
           CAPTURE PKT   Off                  -       -
            MCAST IGMP   Off                  -       -
```

```
Router(config)#mls rate-limit ?
  all        Rate Limiting for both Unicast and Multicast packets
  layer2     layer2 protocol cases
  multicast  Rate limiting for Multicast packets
  unicast    Rate limiting for Unicast packets
```

Here is an example:

```
Router(config)#mls rate-limit layer2 l2pt 3000
```

In order to rate−limit all CEF−punted packets to the MSFC, issue the command that is in this example:

```
Router(config)#mls ip cef rate-limit 50000
```

In order to reduce the number of packets punted to the CPU due to TTL=1, issue this command:

```
Router(config)#mls rate-limit all ttl-failure 15


!--- where 15 is the number of packets per second with TTL=1.
!--- The valid range is from 10 to 1000000 pps.
```

For example, this is the output of the **netdr capture**, which shows that the IPv4 TTL is 1:

```
Source mac    00.00.50.02.10.01  3644
Dest mac      AC.A0.16.0A.B0.C0  4092
Protocol      0800               4094
Interface     Gi1/8              3644
Source vlan   0x3FD(1021)        3644
Source index  0x7(7)             3644
Dest index    0x380(896)         3654

L3

ipv4 source   211.204.66.117      762
ipv4 dest     223.175.252.49     3815
ipv4 ttl                   1     3656
ipv6 source                -        0
ipv6 dest                  -        0
ipv6 hoplt                 -        0
ipv6 flow                  -        0
ipv6 nexthdr               -        0
```

High CPU can also be due to packets with TTL=1 which are leaked to the CPU. In order to limit the number of packets that are leaked to the CPU, configure a hardware rate limiter. Rate limiters can rate−limit packets that are leaked from the hardware data path up to the software data path. Rate limiters protect the software control path from congestion by dropping the traffic that exceeds the configured rate. The rate limit is configured using the **mls rate−limit all ttl−failure** command.

Physical Merger of VLANs Due to Incorrect Cabling

High CPU utilization also can result from the merge together of two or more VLANs due to improper cabling. Also, if STP is disabled on those ports where the VLAN merger happens, high CPU utilization can occur.

In order to resolve this problem, identify the cabling errors and correct them. If your requirement allows, you can also enable STP on those ports.

Broadcast Storm

A LAN broadcast storm occurs when broadcast or multicast packets flood the LAN, which creates excessive traffic and degrades network performance. Errors in the protocol−stack implementation or in the network configuration can cause a broadcast storm.

Due to the architectural design of the Catalyst 6500 series platform, the broadcast packets are only and always dropped at the software level.

Broadcast suppression prevents the disruption of LAN interfaces by a broadcast storm. Broadcast suppression uses filtering that measures broadcast activity on a LAN over a 1−second time period and compares the measurement with a predefined threshold. If the threshold is reached, further broadcast activity is suppressed for the duration of a specified time period. Broadcast suppression is disabled by default.

Note: VRRP flapping from backup to master caused by broadcast storms might cause high CPU utilization.

In order to understand how broadcast suppression works and to enable the feature, refer to:

- Configuring Broadcast Suppression (Cisco IOS system software)
- Configuring Broadcast Suppression (CatOS system software)

BGP Next−Hop Address Tracking (BGP Scanner Process)

The BGP Scanner process walks the BGP table and confirms reachability of the next hops. This process also checks conditional advertisement in order to determine whether BGP should advertise condition prefixes and/or perform route dampening. By default, the process scans every 60 seconds.

You can expect high CPU utilization for short durations because of the BGP Scanner process on a router that carries a large Internet routing table. Once per minute, the BGP Scanner walks the BGP Routing Information Base (RIB) table and performs important maintenance tasks. These tasks include:

- A check of the next hop that is referenced in the router BGP table
- Verification that the next−hop devices can be reached

Thus, a large BGP table takes an equivalently large amount of time to be walked and validated. The BGP Scanner process walks the BGP table in order to update any data structures and walks the routing table for route redistribution purposes. Both tables are stored separately in the router memory. Both tables can be very large and, thus, consume CPU cycles.

For more information on CPU utilization by the BGP Scanner process, refer to the *High CPU due to BGP Scanner* section of Troubleshooting High CPU Caused by the BGP Scanner or BGP Router Process.

For more information on the BGP Next−Hop Address Tracking feature and the procedure to enable/disable or adjust the scan interval, refer to BGP Support for Next−Hop Address Tracking.

Non−RPF Multicast Traffic

Multicast routing (unlike unicast routing) is only concerned with the source of a given multicast data stream. That is, the IP address of the device that originates the multicast traffic. The basic principle is that the source device "pushes" the stream out to an undefined number of receivers (within its multicast group). All multicast routers create distribution trees, which control the path that multicast traffic takes through the network in order to deliver traffic to all receivers. The two basic types of multicast distribution trees are source trees and shared trees. RPF is a key concept in multicast forwarding. It enables routers to correctly forward multicast traffic down the distribution tree. RPF makes use of the existing unicast routing table to determine the upstream and downstream neighbors. A router forwards a multicast packet only if it is received on the upstream interface. This RPF check helps to guarantee that the distribution tree is loop−free.

Multicast traffic is always visible by every router on a bridged (Layer 2) LAN, according to the IEEE 802.3 CSMA/CD specification. In the 802.3 standard, bit 0 of the first octet is used to indicate a broadcast and/or multicast frame, and any Layer 2 frame with this address is flooded. This is also the case even if CGMP or IGMP Snooping are configured. This is because multicast routers must see the multicast traffic, if they are expected to make a proper forwarding decision. If multiple multicast routers each have interfaces onto a common LAN, then only one router forwards the data (chosen by an election process). Due to the flooding nature of LANs, the redundant router (router that does not forward the multicast traffic) receives this data on the outbound interface for that LAN. The redundant router normally drops this traffic, because it has arrived on the wrong interface and therefore fails the RPF check. This traffic that fails the RPF check is called non−RPF traffic or RPF failure packets, because they have been transmitted backwards against the flow from the source.

The Catalyst 6500 with an MSFC installed, can be configured to act as full−fledged multicast router. Utilizing Multicast Multi−Layer Switching (MMLS), RPF traffic is usually forwarded by the hardware within the switch. The ASICs are given information from the multicast routing state (for example, (*,G) and (S,G)), so that a hardware shortcut can be programmed into the Netflow and/or FIB table. This non−RPF traffic is still necessary in some cases, and is required by the MSFC CPU (at process level) for the PIM Assert mechanism. Otherwise, it is then dropped by the software fast−switching path (assumption is that software fast−switching is not disabled on the RPF interface).

The Catalyst 6500 that uses redundancy might not handle non−RPF traffic efficiently in certain topologies. For non−RPF traffic, there is usually no (*,G) or (S,G) state in the redundant router, and therefore no hardware or software shortcuts can be created to drop the packet. Each multicast packet must be examined by the MSFC route processor individually, and this is often referred to as CPU Interrupt traffic. With Layer 3 hardware switching and multiple interfaces/VLANs that connect the same set of routers, the non−RPF traffic that hits the CPU of the redundant MSFC is amplified "N" times the original source rate (where "N" is the number of LANs to which the router is redundantly connected). If the rate of the non−RPF traffic exceeds the packet dropping capacity of the system, then it might cause high CPU utilization, buffer overflows and overall network instability.

With the Catalyst 6500, there is an access list engine that enables filtering to take place at wire rate. This feature can be used to handle non−RPF traffic for Sparse Mode groups efficiently, in certain situations. You can only use the ACL−based method within sparse−mode 'stub networks', where there are no downstream multicast routers (and corresponding receivers). Additionally, because of the packet forwarding design of the Catalyst 6500, internally redundant MSFCs cannot use this implementation. This is outlined within the Cisco bug ID CSCdr74908 (registered customers only) . For dense−mode groups, non−RPF packets must be seen on the router for the PIM Assert mechanism to function properly. Different solutions, such as CEF or Netflow based rate−limiting and QoS are used to control RPF failures in dense−mode networks and sparse−mode transit networks.

On the Catalyst 6500 there is an access list engine that enables filtering to take place at wire rate. This feature can be used to handle non−RPF traffic for Sparse Mode groups efficiently. In order to implement this solution, place an access list on the incoming interface of the `stub network' to filter multicast traffic that did not originate from the `stub network'. The access list is pushed down to the hardware in the switch. This access list prevents the CPU from ever seeing the packet and allows the hardware to drop the non−RPF traffic.

Note: Do not place this access list on a transit interface. It is only intended for stub networks (networks with hosts only).

Refer to these documents for more information:

- Redundant Router Issues with IP Multicast in Stub Networks
- Non−RPF Traffic Processing

show Commands

The CPU utilization when you issue a **show** command is always almost 100%. It is normal to have high CPU utilization when you issue a **show** command and normally remains for only a few seconds.

For example, it is normal for the Virtual Exec process to go high when you issue a **show tech−support** command as this output is an interrupt driven output. Your only concern having high CPU in other processes other than **show** commands.

The **show cef not−cef−switched** command shows why packets are punted to the MSFC (receive, ip option, no adjacency, etc) and how many. For example:

```
Switch#show cef not−cef−switched
CEF Packets passed on to next switching layer
Slot  No_adj No_encap Unsupp'ted Redirect  Receive  Options   Access     Frag
RP      6222        0        136        0    60122        0        0        0
5          0        0          0        0        0        0        0        0
IPv6 CEF Packets passed on to next switching layer
Slot  No_adj No_encap Unsupp'ted Redirect  Receive  Options   Access      MTU
RP         0        0          0        0        0        0        0        0
```

The **show ibc** and **show ibc brief** commands show the CPU queue and can be used when you are monitoring CPU status.

Exec Processes

The Exec process in Cisco IOS Software is responsible for communication on the TTY lines (console, auxiliary, asynchronous) of the router. The Virtual Exec process is responsible for the VTY lines (Telnet sessions). The Exec and Virtual Exec processes are medium priority processes, so if there are other processes that have a higher priority (High or Critical), the higher priority processes get the CPU resources.

If there is a lot of data transferred through these sessions, the CPU utilization for the Exec process increases. This is because when the router wants to send a simple character through these lines, the router uses some CPU resources:

- For the console (Exec), the router uses one interrupt per character.
- For the VTY line (Virtual Exec), the Telnet session has to build one TCP packet per character.

This list details some of the possible reasons for high CPU utilization in the Exec process:

- **There is too much data sent through the console port.**

    1. Check to see if any debugs have been started on the router with the **show debugging** command.
    2. Disable console logging on the router with the **no** form of the **logging console** command.
    3. Verify if a long output is printed on the console. For example, a **show tech−support** or a **show memory** command.
- **The exec command is configured for asynchronous and auxiliary lines.**

    1. If a line has only outgoing traffic, disable the Exec process for this line. This is because if the device (for example, a modem) attached to this line sends some unsolicited data, the Exec process starts on this line.

2. If the router is used as a terminal server (for reverse Telnet to other device consoles), it is recommended that you configure the **no exec** command on the lines that are connected to the console of the other devices. Data that comes back from the console might otherwise start an Exec process, which uses CPU resources.

A possible reason for high CPU utilization in the Virtual Exec process is:

- **There is too much data sent across the Telnet sessions.**

  The most common reason for high CPU utilization in the Virtual Exec process is that too much data is transferred from the router to the Telnet session. This can happen when commands with long outputs such as **show tech−support**, **show memory**, and so on, are executed from the Telnet session. The amount of data transferred through each VTY session can be verified with the **show tcp vty** *<line number>* command.

L3 Aging Process

When the L3 aging process exports a large number of *ifindex* values using NetFlow Data Export (NDE), the CPU usage might hit 100%.

If you encounter this problem, check whether these two commands are enabled:

```
set mls nde destination−ifindex enable

set mls nde source−ifindex enable
```

If you enable these commands, the process must export all destination and source ifindex values using NDE. The L3 aging process utilization goes high since it must perform FIB lookup for all destination and source *ifindex* values. Because of this, the table becomes full, the L3 aging process goes high, and the CPU usage hits 100%.

In order to resolve this issue, disable these commands:

```
set mls nde destination−ifindex disable

set mls nde source−ifindex disable
```

Use these commands to verify the values:

- **show mls cef summary**
- **show mls cef maximum−routes**

BPDU Storm

Spanning tree maintains a loop−free Layer 2 environment in redundant switched and bridges networks. Without STP, frames loop and/or multiply indefinitely. This occurrence causes a network meltdown because high traffic interrupts all devices in the broadcast domain.

In some respects, STP is an early protocol that was initially developed for slow software−based bridge specifications (IEEE 802.1D), but STP can be complicated in order to implement it successfully in large switched networks that have these features:

- Many VLANs
- Many switches in a STP domain

- Multi−vendor support
- Newer IEEE enhancements

If the network faces frequent spanning tree calculations or the switch has to process more BPDUs, it can result in high CPU, as well as BPDU drops.

In order to work around these issues, perform any or all of these steps:

1. Prune off the VLANs from the switches.
2. Use an enhanced version of STP, such as MST.
3. Upgrade the hardware of the switch.

Also refer to best practices to implement Spanning Tree Protocol in the network.

- Best Practices for Catalyst 4500/4000, 5500/5000, and 6500/6000 Series Switches Running CatOS Configuration and Management
- Best Practices for Catalyst 6500/6000 Series and Catalyst 4500/4000 Series Switches Running Cisco IOS Software

SPAN Sessions

Based on the architecture of Catalyst 6000/6500 Series Switches, SPAN sessions do not affect the performance of the switch, but, if the SPAN session includes a high traffic / uplink port or an EtherChannel, it can increase the load on the processor. If it then singles out a specific VLAN, it increases the workload even more. If there is bad traffic on the link, that can further increase the workload.

In some scenarios, the RSPAN feature can cause loops, and the load on the processor shoots up. For more information, refer to Why Does the SPAN Session Create a Bridging Loop?

The switch can pass traffic as usual since everything is in hardware, but the CPU can take a beating if it tries to figure out which traffic to send through. It is recommended that you configure SPAN sessions only when it is required.

%CFIB−SP−STBY−7−CFIB_EXCEPTION : FIB TCAM exception, Some entries will be software switched

```
%CFIB−SP−7−CFIB_EXCEPTION : FIB TCAM exception, Some entries will be software switched
%CFIB−SP−STBY−7−CFIB_EXCEPTION : FIB TCAM exception, Some entries will be software
switched
```

This error message is received when the amount of available space in the TCAM is exceeded. This results in high CPU. This is a FIB TCAM limitation. Once TCAM is full, a flag will be set and FIB TCAM exception is received. This stops from adding new routes to the TCAM. Therefore, everything will be software switched. The removal of routes does not help resume hardware switching. Once the TCAM enters the exception state, the system must be reloaded to get out of that state. The maximum routes that can be installed in TCAM is increased by the **mls cef maximum−routes** command.

Catalyst 6500/6000 running with high CPU has an IPv6 ACL with L4 ports

Enable **mls ipv6 acl compress address unicast** . This command is needed if the IPv6 ACL is matching on L4 protocol port numbers. If this command is not enabled, IPv6 traffic will be punted to the CPU for software processing. This command is not configured by default.

Copper SPFs

In Cisco ME 6500 series Ethernet switches, the copper SFPs require more firmware interaction than other types of SFPs, which increases the CPU utilization.

The software algorithms that manage copper SFPs have been improved in the Cisco IOS SXH releases.

Modular IOS

In Cisco Catalyst 6500 series switches that run modular IOS software, the normal CPU utilization is a little greater than non−modular IOS software.

Modular IOS software pays a price per activity more than it pays a price per packet. Modular IOS software maintains the processes by consuming certain CPU even if there is no much packets, so the CPU consumption is not based on the actual traffic. However, when packets been processed go high rate, the CPU consumed in Modular IOS software should not be more than that in non−modular IOS software.

Check CPU Utilization

If the CPU utilization is high, issue the **show processes cpu** command first. The output shows you the CPU utilization on the switch as well as the CPU consumption by each process.

```
Router#show processes cpu
CPU utilization for five seconds: 57%/48%; one minute: 56%; five minutes: 48%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
   1           0         5          0  0.00%  0.00%  0.00%   0 Chunk Manager
   2          12     18062          0  0.00%  0.00%  0.00%   0 Load Meter
   4      164532     13717      11994  0.00%  0.21%  0.17%   0 Check heaps
   5           0         1          0  0.00%  0.00%  0.00%   0 Pool Manager

!−−− Output is suppressed.

 172           0         9          0  0.00%  0.00%  0.00%   0 RPC aapi_rp
 173      243912   2171455        112  9.25%  8.11%  7.39%   0 SNMP ENGINE
 174          68       463        146  0.00%  0.00%  0.00%   0 RPC pm−mp

!−−− Output is suppressed.
```

In this output, the total CPU utilization is 57 percent and the interrupt CPU utilization is 48 percent. Here, these percentages appear in boldface text. The interrupt switch of traffic by the CPU causes the interrupt CPU utilization. The command output lists the processes that cause the difference between the two utilizations. In this case, the cause is the SNMP process.

On the supervisor engine that runs CatOS, the output looks like this:

```
Switch> (enable) show processes cpu


CPU utilization for five seconds:  99.72%
                      one minute: 100.00%
                    five minutes: 100.00%

PID Runtime(ms) Invoked    uSecs     5Sec    1Min    5Min    TTY Process
--- ----------- ---------- --------- ------- ------- ------- --- ---------------
1   0           0          0           0.28%   0.00%   0.00% -2  Kernel and Idle
2   2           261        1000        0.00%   0.00%   0.00% -2  Flash MIB Updat
3   0           1          0           0.00%   0.00%   0.00% -2  L2L3IntHdlr
```

```
4   0           1          0          0.00%   0.00%   0.00% -2  L2L3PatchRev

!--- Output is suppressed.

61  727295      172025     18000      0.82%   0.00%   0.00% -2  SptTimer
62  18185410    3712736    106000    22.22%  21.84%  21.96% -2  SptBpduRx
63  845683      91691      105000     0.92%   0.00%   0.00% -2  SptBpduTx
```

In this output, the first process is `Kernel and Idle`, which shows idle CPU utilization. This process is normally high, unless some other processes consume CPU cycles. In this example, the `SptBpduRx` process causes high CPU utilization.

If the CPU utilization is high due to one of these processes, you can troubleshoot and determine why this process runs high. But, if the CPU is high due to traffic being punted to the CPU, you need to determine why the traffic is being punted. This determination can help you identify what the traffic is.

For troubleshooting, use this EEM script example in order to collect the output from the switch when you experience high CPU utilization:

```
event manager applet cpu_stats

event snmp oid "1.3.6.1.4.1.9.9.109.1.1.1.1.3.1" get-type exact entry-op gt entry-val "70"

exit-op lt exit-val "50" poll-interval 5

action 1.01 syslog msg "------HIGH CPU DETECTED----, CPU:$_snmp_oid_val%"

action 1.02 cli command "enable"

action 1.03 cli command "show clock | append disk0:cpu_stats"

action 1.04 cli command "show proc cpu sort | append disk0:cpu_stats"

action 1.05 cli command "Show proc cpu | exc 0.00% | append disk0:cpu_stats"

action 1.06 cli command "Show proc cpu history | append disk0:cpu_stats"

action 1.07 cli command "show logging | append disk0:cpu_stats "

action 1.08 cli command "show spanning-tree detail | in ieee|occurr|from|is exec | append
disk0:cpu_stats"

action 1.09 cli command "debug netdr cap rx | append disk0:cpu_stats"

action 1.10 cli command "show netdr cap | append disk0:cpu_stats"

action 1.11 cli command "undebug all"
!
```

Note: The **debug netdr capture rx** command is helpful when the CPU is high due to process switching of packets instead of hardware. It captures 4096 packets incoming to the CPU when the command is run. The command is completely safe and is the most convenient tool for high CPU issues on the 6500. It does not cause extra load to the CPU.

Utilities and Tools to Determine the Traffic That Is Punted to the CPU

This section identifies some utilities and tools that can help you look at this traffic.

Cisco IOS System Software

In Cisco IOS Software, the switch processor on the supervisor engine is referred to as the SP, and the MSFC is called the RP.

The **show interface** command gives basic information on the state of the interface and the traffic rate on the interface. The command also provides error counters.

```
Router#show interface gigabitethernet 4/1
GigabitEthernet4/1 is up, line protocol is up (connected)
  Hardware is C6k 1000Mb 802.3, address is 000a.42d1.7580 (bia 000a.42d1.7580)
  Internet address is 100.100.100.2/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half−duplex, 100Mb/s
  input flow−control is off, output flow−control is off
  Clock mode is auto
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 5/75/1/24075 (size/max/drops/flushes); Total output drops: 2
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 7609000 bits/sec, 14859 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
  L2 Switched: ucast: 0 pkt, 184954624 bytes − mcast: 1 pkt, 500 bytes
  L3 in Switched: ucast: 2889916 pkt, 0 bytes − mcast: 0 pkt, 0 bytes mcast
  L3 out Switched: ucast: 0 pkt, 0 bytes mcast: 0 pkt, 0 bytes
     2982871 packets input, 190904816 bytes, 0 no buffer
     Received 9 broadcasts, 0 runts, 0 giants, 0 throttles
     1 input errors, 1 CRC, 0 frame, 28 overrun, 0 ignored
     0 input packets with dribble condition detected
     1256 packets output, 124317 bytes, 0 underruns
     2 output errors, 1 collisions, 2 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

In this output, you can see that the incoming traffic is Layer 3−switched instead of Layer 2−switched. This indicates that the traffic is being punted to the CPU.

The **show processes cpu** command tells you whether these packets are regular traffic packets or control packets.

```
Router#show processes cpu | exclude 0.00
CPU utilization for five seconds: 91%/50%; one minute: 89%; five minutes: 47%
 PID Runtime(ms)   Invoked      uSecs   5Sec   1Min   5Min TTY Process
   5      881160     79142      11133  0.49%  0.19%  0.16%   0 Check heaps
  98      121064   3020704         40 40.53% 38.67% 20.59%   0 IP Input
 245      209336    894828        233  0.08%  0.05%  0.02%   0 IFCOM Msg Hdlr
```

If the packets are process−switched, you see that the `IP Input` process runs high. Issue this command in order to see these packets:

show buffers input−interface

```
Router#show buffers input−interface gigabitethernet 4/1 packet

Buffer information for Small buffer at 0x437874D4
```

```
  data_area 0x8060F04, refcount 1, next 0x5006D400, flags 0x280
  linktype 7 (IP), enctype 1 (ARPA), encsize 14, rxtype 1
  if_input 0x505BC20C (GigabitEthernet4/1), if_output 0x0 (None)
  inputtime 00:00:00.000 (elapsed never)
  outputtime 00:00:00.000 (elapsed never), oqnumber 65535
  datagramstart 0x8060F7A, datagramsize 60, maximum size 308
  mac_start 0x8060F7A, addr_start 0x8060F7A, info_start 0x0
  network_start 0x8060F88, transport_start 0x8060F9C, caller_pc 0x403519B4

  source: 100.100.100.1, destination: 100.100.100.2, id: 0x0000, ttl: 63,
  TOS: 0 prot: 17, source port 63, destination port 63

08060F70:                       000A 42D17580            ..BQu.
08060F80: 00000000 11110800 4500002E 00000000  ........E.......
08060F90: 3F11EAF3 64646401 64646402 003F003F  ?.jsddd.ddd..?.?
08060FA0: 001A261F 00010203 04050607 08090A0B  ..&.............
08060FB0: 0C0D0E0F 101164                      ......d
```

If the traffic is **interrupt switched**, you cannot see those packets with the **show buffers input−interface** command. In order to see the packets that are punted to the RP for interrupt switching, you can perform a Switched Port Analyzer (SPAN) capture of the RP port.

Note: Refer to this document for additional information about interrupt−switched versus process−switched CPU utilization:

- *High CPU Utilization due to Interrupts* section of Troubleshooting High CPU Utilization on Cisco Routers

SPAN RP−Inband and SP−Inband

A SPAN for the RP or SP port in Cisco IOS Software is available in Cisco IOS Software Release 12.1(19)E and later.

This is the command syntax:

```
test monitor session 1−66 add {rp−inband | sp−inband} [rx | tx | both]
```

Use this syntax for the Cisco IOS Software 12.2 SX releases:

```
test monitor add {1..66} {rp−inband | sp−inband} {rx | tx | both}
```

Note: For the SXH release, you must use the **monitor session** command in order to configure a local SPAN session, and then use this command to associate the SPAN session with the CPU:

```
source {cpu {rp | sp}} | single_interface | interface_list |
     interface_range | mixed_interface_list | single_vlan |
     vlan_list | vlan_range | mixed_vlan_list} [rx | tx | both]
```

Note: For more information on these commands, refer to Configuring Local SPAN (SPAN Configuration Mode) in the *Catalyst 6500 Release 12.2SX Software Configuration Guide*.

Here is an example on an RP console:

```
Router#monitor session 1 source interface fast 3/3

!−−− Use any interface that is administratively shut down.

Router#monitor session 1 destination interface 3/2
```

Now, go to the SP console. Here is an example:

```
Router-sp#test monitor session 1 add rp-inband rx
```

Note: In Cisco IOS 12.2 SX releases, the command has been changed to **test monitor add 1 rp-inband rx**.

```
Router#show monitor
Session 1
---------
Type : Local Session
Source Ports :
Both : Fa3/3
Destination Ports : Fa3/2
SP console:
Router-sp#test monitor session 1 show
Ingress Source Ports: 3/3 15/1
Egress Source Ports: 3/3
Ingress Source Vlans: <empty>
Egress Source Vlans: <empty>
Filter Vlans: <empty>
Destination Ports: 3/2
```

Note: In Cisco IOS 12.2 SX releases, the command has been changed to **test monitor show 1**.

Here is an example on an SP console:

```
Router-sp#test monitor session 1 show
Ingress Source Ports: 3/3 15/1
Egress Source Ports: 3/3
Ingress Source Vlans: <empty>
Egress Source Vlans: <empty>
Filter Vlans: <empty>
Destination Ports: 3/2
```

CatOS System Software

For switches that run CatOS system software, the supervisor engine runs CatOS and the MSFC runs Cisco IOS Software.

If you issue the **show mac** command, you can see the number of frames that are punted to the MSFC. Port 15/1 is the supervisor engine connection to the MSFC.

Note: The port is 16/1 for supervisor engines in slot 2.

```
Console> (enable) show mac 15/1

Port     Rcv-Unicast          Rcv-Multicast        Rcv-Broadcast
-------- -------------------- -------------------- --------------------
15/1                   193576                    0                    1

Port     Xmit-Unicast         Xmit-Multicast       Xmit-Broadcast
-------- -------------------- -------------------- --------------------
15/1                        3                    0                    0

Port     Rcv-Octet            Xmit-Octet
-------- -------------------- --------------------
15/1                 18583370                    0

MAC      Dely-Exced MTU-Exced  In-Discard Out-Discard
-------- ---------- ---------- ---------- -----------
15/1              0          -          0           0
```

A quick increase in this number indicates that packets are punted to the MSFC, which causes high CPU utilization. You can then look at the packets in these ways:

- SPAN MSFC port 15/1 or 16/1
- SPAN sc0

SPAN MSFC Port 15/1 or 16/1

Set up a SPAN session in which the source is the MSFC port 15/1 (or 16/1) and the destination is an Ethernet port.

Here is an example:

```
Console> (enable) set span 15/1 5/10
Console> (enable) show span

Destination     : Port 5/10
Admin Source    : Port 15/1
Oper Source     : None
Direction       : transmit/receive
Incoming Packets: disabled
Learning        : enabled
Multicast       : enabled
Filter          : −
Status          : active
```

If you collect a sniffer trace on port 5/10, the sniffer trace shows packets that transmit to and from the MSFC. Configure the SPAN session as **tx** in order to capture packets that are only destined to the MSFC, and not from the MSFC.

SPAN sc0

Set up a SPAN session with the **sc0** interface as the source in order to capture frames that go to the supervisor engine CPU.

```
Console> (enable) set span ?
  disable                    Disable port monitoring
  sc0                        Set span on interface sc0
  <mod/port>                 Source module and port numbers
  <vlan>                     Source VLAN numbers
```

Note: For Optical Services Modules (OSMs), you cannot perform a SPAN capture of traffic.

Recommendations

The supervisor engine CPU utilization does not reflect the hardware forwarding performance of the switch. Still, you must baseline and monitor the supervisor engine CPU utilization.

1. Baseline the supervisor engine CPU utilization for the switch in a steady−state network with normal traffic patterns and load.

   Note which processes generate the highest CPU utilization.
2. When you troubleshoot CPU utilization, consider these questions:

   - Which processes generate the highest utilization? Are these processes different from your baseline?

- Is the CPU consistently elevated, over the baseline? Or are there spikes of high utilization, and then a return to the baseline levels?
- Are there Topology Change Notifications (TCNs) in the network?

  Note: Flapping ports or host ports with STP PortFast disabled cause TCNs.
- Is there excessive broadcast or multicast traffic in the management subnets/VLAN?
- Is there excessive management traffic, such as SNMP polling, on the switch?

3. During the high CPU time (when the CPU is 75% or above), collect the output from these commands:

   - **show clock**
   - **show version**
   - **show processes cpu sorted**
   - **show proc cpu history**
   - **show log**
4. If possible, isolate the management VLAN from the VLANs with user data traffic, particularly heavy broadcast traffic.

   Examples of this type of traffic include IPX RIP/Service Advertising Protocol (SAP), AppleTalk, and other broadcast traffic. Such traffic can impact the supervisor engine CPU utilization and, in extreme cases, can interfere with the normal operation of the switch.
5. If the CPU runs high due to the punt of traffic to the RP, determine what that traffic is and why the traffic is punted.

   In order to make this determination, use the utilities that the Utilities and Tools to Determine the Traffic That Is Punted to the CPU section describes.

Related Information

- **Useful commands for troubleshooting high CPU on catalyst 6500's with Sup720**
- **Common CatOS Error Messages on Catalyst 6000/6500 Series Switches**
- **Common Error Messages on Catalyst 6500/6000 Series Switches Running Cisco IOS Software**
- **Troubleshooting Hardware and Common Issues on Catalyst 6500/6000 Series Switches Running Cisco IOS System Software**
- **Unicast Flooding in Switched Campus Networks**
- **Cisco Catalyst 6500 Series Switches Product Support**
- **EEM Script for collecting data during Intermittent High CPU issue**
- **LAN Product Support**
- **LAN Switching Technology Support**
- **Technical Support & Documentation − Cisco Systems**

Updated: Aug 27, 2012 Document ID: 63992