isco − Troubleshooting Input Queue Drops and Output Queue

Table of Contents

Troubleshooting Input Queue Drops and Output Queue Drops......1
Interactive: This document offers customized analysis of your Cisco device......1
Introduction......1
Before You Begin......1
Conventions......1
Prerequisites......1
Components Used......1
Processing and Switching......2
Input Queue Drops......2
Troubleshooting Input Queue Drops......3
Output Queue Drops......6
Troubleshooting Output Queue Drops......7
Commands for Obtaining More Information......7
show interfaces switching......8
show interfaces stats......9
ip accounting mac−address......9
show interfaces mac−accounting......9
Related Information......10

Troubleshooting Input Queue Drops and Output Queue Drops

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

Introduction
Before You Begin
Conventions
Prerequisites
Components Used
Processing and Switching
Input Queue Drops
Troubleshooting Input Queue Drops
Output Queue Drops
Troubleshooting Output Queue Drops
Commands for Obtaining More Information
show interfaces switching
show interfaces stats
ip accounting mac−address
show interfaces mac−accounting
Related Information

Introduction

The input and output queue drops referred to in this document are those that are taken from the router's **show interfaces** output. This document gives a brief description of what these drops mean, the type of problems they indicate, how to troubleshoot the source of these problems, and provides some tips on how to prevent them. Note that drops may often be useful as they trigger the flow control mechanisms of upper layer protocols (for example, decreasing the TCP window size).

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.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

Processing and Switching

In IP networks, forwarding decisions in routers are based on the contents of the routing table. When searching the routing table, the router is looking for the longest match for the destination IP address. This is done at process level, queued among other CPU processes so the lookup time is unpredictable and may be very long. Therefore, a number of switching methods based on exact−match−lookup have been introduced in Cisco IOS® Software.

The main benefit of exact−match−lookup is that the lookup time is deterministic and very short. This has significantly shortened the time a router takes for making a forwarding decision, so routines performing the search can be implemented at the interrupt level. This means that the arrival of a packet triggers an interrupt, which causes the CPU to postpone other tasks and handle the packet. The legacy method for forwarding packets, by looking for a best match in the routing table, cannot be implemented at interrupt level and must be performed at process level. For a number of reasons, some of which are mentioned below, the longest−match−lookup method cannot be completely abandoned, so these two lookup methods exist in parallel on Cisco routers. This strategy has been generalized and is now also applied to IPX and AppleTalk.

For more information on Cisco IOS Software switching paths, please refer to Performance Tuning Basics.

Input Queue Drops

When a packet enters the router, the router attempts to forward it at interrupt level. If a match can't be found in an appropriate cache table, the packet is queued in the incoming interface's input queue for processing. Some packets are always processed, but with the appropriate configuration and in stable networks, the rate of processed packets should never congest the input queue. If the input queue is full, the packet will be dropped.

```
router#show interfaces ethernet 0/0
...
Input queue: 30/75/187/0 (size/max/drops/flushes); Total output drops: 0
Output queue :0/40 (size/max)...
```

In the example above, there is no way to see exactly which packets have been dropped. Troubleshooting input queue drops is actually based on finding out which packets are filling the input queue. In our example, 30 packets were waiting in the input queue of interface ethernet0/0 when the **show interfaces ethernet 0/0** command was issued. The queue depth is 75 packets and there have been 187 drops since the last clearing of the interface counters.

Input queue drops are counted by the system if the number of packet buffers allocated to the interface is exhausted or reaches its maximum threshold. The maximum queue value can be increased by using the **hold−queue <*value*>** command for each interface (the queue length value can be 0 − 4096. The default value is 75).

Note: Shared−memory routers (1600, 2500, and 4000 series), also use the input queue for fast−switched traffic. If you get input queue drops on those platforms, make sure all traffic uses the best switching path available (see Performance Tuning Basics). Input queue drops generally occur when a packet is being process−switched. Process switching means that the router cannot use a preferable route−cache method, such as fast switching or Cisco Express Forwarding (CEF), to handle the forwarding decision. If input drops are still present, then there is simply too much traffic. Consider a hardware upgrade, or try to decrease the traffic load.

Troubleshooting Input Queue Drops

Troubleshooting input queue drops can only be successfully performed while there are packets constantly arriving in the input queue. You cannot troubleshoot any congestion that has taken place in the past. If more than one routed protocol is configured on the interface, the first step in troubleshooting is to determine the protocol which is congesting the input queue. The fastest way to do this is:

1. Determine the suspect protocol. Check the CPU utilization in **<*protocol*> Input** processes by running the **show processes cpu exec** command. If Cisco IOS Software version 12.1 or higher is running on the router, the output of the **show processes CPU** command can be shortened by using the output modifiers:

```
router#show processes CPU | i ^PID|Input
 PID  Runtime(ms)  Invoked  uSecs    5Sec   1Min   5Min TTY Process
  10        8503       1713   4963   0.00%  0.00%  0.00%   0 ARP Input
  24       69864      11429   6112   0.08%  0.11%  0.10%   0 Net Input
  28       55099       8942   6161  26.20% 20.07% 19.26%   0 IP Input
  37           4          2   2000   0.00%  0.00%  0.00%   0 SSCOP Input
  40           8          2   4000   0.00%  0.00%  0.00%   0 ILMI Input
  49           8          1   8000   0.00%  0.00%  0.00%   0 Probe Input
  50       28209       4637   6083   0.00%  0.03%  0.04%   0 RARP Input
  59           8          2   4000   0.00%  0.00%  0.00%   0 SPX Input
  61           8          2   4000   0.00%  0.00%  0.00%   0 Tag Input
  68       20803       3392   6132   0.00%  0.03%  0.00%   0 IPX Input
 104           4          1   4000   0.00%  0.00%  0.00%   0 IPXWAN Input
 107           8          1   8000   0.00%  0.00%  0.00%   0 AT Input
```

The following table lists the possible input processes and types of packets that may be responsible for congesting the input queue:

Input Process using CPU Cycles	Type of Packets
IP	IP
AT	AppleTalk
IPX, SPX, or IPXWAN	IPX
ARP	IP ARP

Other input processes are not likely to congest the input queue.

2. Find out if packets that congest the input queue are destined for the router or are forwarded through the router. Run the **show interfaces [*type number*] switching** command from exec mode (this is a hidden command − it does not show up if you use the "?" or TAB keys on the command line interface. Type the command in full on the router. This command is not documented in the Command Reference Guide).

```
router#show interfaces ethernet 0/0 switching
Ethernet0/0
    ...
    Protocol          Path    Pkts In   Chars In   Pkts Out  Chars Out
    ...
     IP            Process      12142    2211929         35       5169
              Cache misses      10212
    ...
```

Check if the number of received processed packets is followed by a high number of cache misses. If yes, this indicates that the packets congesting the input queue are forwarded through the router.

Otherwise, these packets are destined for the router.

3. If packets are destined for the router, find out which higher−layer protocol is congesting the input queue by running one of the **show traffic** exec commands below (this is applicable only if the input processes in the table above are suspected):

    - **show ip traffic**
    - **show ipx traffic**
    - **show appletalk traffic**
4. Try to get more information about the packets that are congesting the input queue. This step requires debugging the received packets. If you have followed the previous steps, you should have the information indicating which debugging commands should be turned on. This should make the task of examining the debugging output a lot easier.

    Note: This step can be applied directly, without going through the previous steps. However, debugging produces lots of messages which can be hard to read. By following all the troubleshooting steps above, you get an indication of what should be looked for in the debugging output.

    Warning: Debugging should be carried out with extreme caution; otherwise, CPU utilization may increase considerably. Debugging should be turned on for not more than 5−10 seconds. For more information on how to use the debugging commands, refer to Using Debug Commands. Console logging, terminal logging, and logging to a syslog server should be disabled. Buffer logging should be enabled and logging buffer size increased. A good value for the size of logging buffer would be 128000 bytes. The following commands should be used to accomplish this:

    - **no logging <host>**
    - **logging buffered 128000 debugging**

    The produced output should be sufficient to find the source of the problem. The debugging output can be checked by issuing the **show log** command after completing the debugging session. The following table shows you which **debug** command should be issued based on the type of packets congesting the input queue:

Type of Packets Congesting the Input Queue	Debug Command to be Used
IP	**debug ip packet**
AppleTalk	**debug apple packet**
IPX	**debug ipx packet**
ARP	**debug arp**

For more information on debugging, refer to the Cisco IOS Debug Command Reference.

Alternatively, you can use the **show buffers input−interface *[interface type*] [*interface number*] header** command to find out what types of packets are filling up the input queue.

Note: This is only useful if there are a lot of packets waiting in the input queue.

```
Router#show buffers input-interface serial 0/0
 Buffer information for Small buffer at 0x612EAF3C
   data_area 0x7896E84, refcount 1, next 0x0, flags 0x0
   linktype 7 (IP), enctype 0 (None), encsize 46, rxtype 0
   if_input 0x6159D340 (FastEthernet3/2), if_output 0x0 (None)
```

```
    inputtime 0x0, outputtime 0x0, oqnumber 65535
    datagramstart 0x7896ED8, datagramsize 728, maximum size 65436
    mac_start 0x7896ED8, addr_start 0x7896ED8, info_start 0x0
    network_start 0x7896ED8, transport_start 0x0
    source: 212.176.72.138, destination: 212.111.64.174, id: 0xAAB8,
    ttl: 118, prot: 1
  Buffer information for Small buffer at 0x612EB1D8
    data_area 0x78A6E64, refcount 1, next 0x0, flags 0x0
    linktype 7 (IP), enctype 0 (None), encsize 46, rxtype 0
    if_input 0x6159D340 (FastEthernet3/2), if_output 0x0 (None)
    inputtime 0x0, outputtime 0x0, oqnumber 65535
    datagramstart 0x78A6EB8, datagramsize 728, maximum size 65436
    mac_start 0x78A6EB8, addr_start 0x78A6EB8, info_start 0x0
    network_start 0x78A6EB8, transport_start 0x0
    source: 212.176.72.138, destination: 212.111.64.174, id: 0xA5B8,
    ttl: 118, prot: 1
```

Most of the time, you'll notice that one type of packet is present in large quantities. Here, for example, we are dealing with Internet Control Message Protocol (ICMP) packets (IP protocol 1).

If the problem is a router misconfiguration (for example, both fast switching and Cisco express forwarding (CEF) are disabled), there will probably be no particular pattern in the debugs, or in the **show buffers input−interface** command.

5. Once you have determined the type of packets that are congesting the input queue, the next step is to see if this congestion can be prevented.

   There are several reasons why packets have to be processed:

   - **Router misconfiguration**. Switching paths operating at interrupt level are disabled on relevant interfaces.

     To check which switching paths are configured on an interface, run the **show <*protocol*> interface [*type number*]** command.

     - To enable legacy fast switching, you have to configure it on output interfaces.
     - To enable netflow switching, you have to configure it on input interfaces.
     - To enable Cisco express forwarding (CEF), you have to enable CEF globally (on the entire router) and locally (on the incoming interface).

       Follow this URL for more information on Cisco IOS Switching Services Configuration Guide.
   - **Local destination**. Packets are destined for the router.

     - Excessive routing updates. In stable networks, the number of routing updates shouldn't be excessive. In unstable networks, too frequent updates of large routing tables could congest the input queue.
     - Excessive traffic directed to the router itself (using, for example, Simple Network Management Protocol (SNMP), telnet, Trivial File Transfer Protocol (TFTP), and ping). The source of these packets can be found by debugging the packets for the relevant protocol. Once found, the source can be eliminated.
   - **Reliable Open System Interconnection (OSI) layer 2 protocol is used for transport**. Packets going through serial interfaces with X.25 encapsulation must be processed because in the X.25 protocol suite, flow control is implemented on the second OSI layer.
   - **Software compression**. If the packet is coming in or has to be forwarded through an interface on which software compression is configured, the packet has to be processed.

- **Other features are unsupported at interrupt level**. This is highly dependent on the Cisco IOS Software release running on the router. Check the release notes to see which features are and are not supported at interrupt level. For example, in earlier Cisco IOS Software versions, multilink PPP packets had to be processed. In higher Cisco IOS Software versions, they can be fast−switched or even CEF−switched. Features such as encryption, local−area transport (LAT) translation, and data−link switching plus (DLSW+) are not yet fast−switched.
- **Excessive traffic through the router, intentionally using different information in each packet header**. Depending on the configured switching path, first packets to a destination or in a flow are always processed because there are not yet any entries in the cache that matches them. If a device sends packets at an extremely high rate and there is no match in the cache, those packets could congest the input queue.

The source of these packets are revealed after the debugging session. If the source address always appears to be different, troubleshooting should be continued on the upstream device, from which the packet has been received. If the interface on the router is connected to a broadcast medium, you can determine the Media Access Control (MAC) address of the source or the upstream device:

MAC accounting should be configured on the interface by issuing the **ip accounting mac−address input** interface configuration command. After that, the **show interfaces mac−accounting** exec command reveals the MAC address that has sent the packets at an excessive rate.

Output Queue Drops

Output drops are the result of a congested interface (for example, the traffic rate on the outgoing interface can't accept all packets that should be sent out). The ultimate solution to resolve the problem is to increase the line speed. However, there are ways to prevent, decrease, or control output drops without increasing the line speed. Preventing output drops is only possible if output drops are a consequence of short bursts of data. If output drops are caused by a constant high−rate flow, drops can't be prevented; they can only be controlled.

When packet processing is finished, the packet is sent to the outgoing interface's output queue. The size of the queue, the current number of packets in the queue, and the number of drops can be viewed by running the **show interfaces** exec command. Depending on the type of interface and the type of queueing configured, the number of output queue drops is not explicitly shown, because the output drops counter summarizes the output drops separately: at the processing level and at the interrupt level:

```
router#show interfaces serial 0/0
  ...
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
  ...
router#show interfaces serial 0/0
...
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue :0/40 (size/max)
...
```

However, since sending the packet from the output queue to the wire takes less time than processing the packet, it is highly unlikely that output queue drops (drops at processing level) can occur without drops at interrupt level. That is, output queue drops occur only if the interface is already congested at interrupt level, so that packets can't be pulled out of the output queue before it becomes full. Therefore, output drops at processing level (output queue drops) and output drops at interrupt level always occur together, and there is practically no need to distinguish between these two counters.

Note: There is one exception − if the output queue is constantly full and if no packets are being sent out of the interface at all, then it may be an indication of a hardware failure on the interface.

Troubleshooting Output Queue Drops

Adjusting the configuration of the following features may decrease, or even prevent, output drops:

- **Duplex mode**. If the interface is working in half−duplex mode, it should be configured (if possible) to work in full−duplex.
- **Layer 2 windowing mechanism**. If x.25 encapsulation is configured on the interface, the x.25 window size should be increased.
- **Distributed switching**. On Cisco 7500 routers, if there are Versatile Interface Protocol (VIP) cards installed in the chassis, distributed switching should be enabled. When distributed switching is enabled on the incoming VIP, if the outgoing interface is congested, the incoming VIP will buffer up to 1 second of traffic for that interface. This is called **rx−side buffering** .

Note: output drop prevention should never be attempted by increasing the output queue. If packets stay too long in the output queue, TCP timers might expire and trigger the retransmission. Retransmitted packets only congest the outgoing interface even more.

If output drops still occur after adjusting the configuration of the router as recommended above, then output drops can't be prevented or decreased. They can, however, be controlled, and this may be as effective as drops prevention. There are two approaches to controlling drops: congestion management and congestion avoidance. Both approaches are based on traffic classification, and can be used in parallel.

Congestion management ensures, with appropriate configuration, that important packets are always forwarded, while less important packets are dropped when the link is congested. Congestion management comprises fancy queueing mechanisms:

- priority queueing
- custom queueing
- class−based weighted fair queueing

Congestion avoidance is based on intentional packet drops. Since the window size in TCP connections depends on the round trip time, these intentional drops slow down the rate at which the source device is sending packets. Congestion avoidance uses weighted random early detection.

If unwanted output drops still occur after implementing these mechanisms, you need to increase the line speed.

Commands for Obtaining More Information

The following commands provide more information about queue drops:

- **show interfaces switching**
- **show interfaces stats**
- **ip accounting mac−address**
- **show interfaces mac−accounting**

If you have the output of a **show interfaces** command 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.

show interfaces switching

Description

This command shows the number of sent and received packets on an interface, classified per switching path. This is a hidden command.

Format

```
show interfaces [type number] switching
```

Sample Output

```
Ethernet0/0
               Throttle count          0
             Drops         RP          0      SP  0
       SPD Flushes       Fast          0     SSE  0
       SPD Aggress       Fast          0
      SPD Priority     Inputs         86   Drops  0
          Protocol       Path    Pkts In   Chars In   Pkts Out   Chars Out
             Other    Process         75       6728         79        4740
                 Cache misses          0
                         Fast          0          0          0           0
                    Auton/SSE          0          0          0           0
                IP    Process        142      11929         35        5169
                 Cache misses          0
                         Fast          0          0          0           0
                    Auton/SSE          0          0          0           0
         AppleTalk    Process          0          0         25        1635
                 Cache misses          0
                         Fast          0          0          0           0
                    Auton/SSE          0          0          0           0
           DEC MOP    Process          0          0          2         154
                 Cache misses          0
                         Fast          0          0          0           0
                    Auton/SSE          0          0          0           0
               ARP    Process         56       3580         13         780
                 Cache misses          0
                         Fast          0          0          0           0
                    Auton/SSE          0          0          0           0
               CDP    Process         90      26906         27        8900
                 Cache misses          0
                         Fast          0          0          0           0
                    Auton/SSE          0          0          0           0
```

Field	Definition
<protocol> **Process**	Number of processed packets. This includes packets destined for the router, and packets for which there was no entry in the appropriate switching cache table.
Cache misses	Packets that have been forwarded through process level (for which there was no entry in fast switching cache).
Fast	Packets forwarded at interrupt level.

show interfaces stats

Description

This command is similar to the **show interfaces switching** command, and provides information regarding the number of packets that are process−switched, fast−switched (any fast switching path), and distributed−switched (for VIP capable platforms). This is a hidden command.

Format

```
show interfaces [type number] stats
```

Sample Output

```
Router#show interfaces stats
FastEthernet8/0/0
          Switching path    Pkts In   Chars In   Pkts Out  Chars Out
               Processor         64      38646        323      32790
             Route cache     477985  611343050      14815   18948150
       Distributed cache          0          0       3564    4558356
                   Total     478049  611381696      18702   23539296
Serial12/0/0
          Switching path    Pkts In   Chars In   Pkts Out  Chars Out
               Processor         37       3783         36       2299
             Route cache      14815   18800000      45118   59862772
       Distributed cache       3450    4378520          0          0
                   Total      18302   23182303      45154   59865071
Interface Serial12/0/1 is disabled
...
```

ip accounting mac−address

Description

This is an interface configuration command. It provides accounting of received or transmitted packets, classified per source or destination MAC address.

Format

ip accounting mac−address {*input*|*output*}

show interfaces mac−accounting

Description

This is an exec command. It shows the number of sent and received packets classified per destination and source MAC address.

Format

show interfaces [*type number*] mac−accounting

Sample Output

```
router#show interfaces ethernet 0/0 mac−accounting
Ethernet0/0
  Input(494 free)
    0000.0c5d.92f9(58 ):  1 packets, 106 bytes, last: 4038ms ago
    0004.c059.c060(61 ):  0 packets, 0 bytes, last: 2493135ms ago
    00b0.64bc.4860(64 ):  1 packets, 106 bytes, last: 20165ms ago
    0090.f2c9.cc00(103):  12 packets, 720 bytes, last: 3117ms ago
                  Total:  14 packets, 932 bytes
  Output (511 free)
    0090.f2c9.cc00(103):  8 packets, 504 bytes, last: 4311ms ago
                  Total:  8 packets, 504 bytes
```

Related Information

- **Performance Tuning Basics**
- **Input Queue Overflow on an Interface**
- **Output Queue Overflow on an Interface**
- **Troubleshooting Input Drops on the Cisco 12000 Series Internet Router**
- **Troubleshooting Output Drops with Priority Queueing**
- **Technical Support − Cisco Systems**