Table of Contents

SNMP Counters: Frequently Asked Questions................1
Document ID: 26007................1
Questions................1
SNMP Counters Questions................1
SNMP Counters and show Command Equivalent Questions................1
SNMP Counters Questions................1
SNMP Counters and show Command Equivalent Questions................4
Examples................5
Related Information................14

SNMP Counters: Frequently Asked Questions

Document ID: 26007

This document provides answers to commonly asked questions, and guides users to find helpful resources on SNMP, and SNMP issues as they relate to Cisco equipment.

Questions

SNMP Counters Questions

- Which MIB should I use for interface counters?
- When should 64−bit counters be used?
- Which version of SNMP is required to query 64−bit counters?
- Which Cisco devices implement 64−bit SNMP counters, especially for the IF−MIB?
- Are ifInOctets and ifOutOctets SNMP counters the same as the **show interfaces** In/Out counters?
- Do the ifInOctets and ifOutOctets counters include framing overhead (Point−to−Point Protocol, High−Level Data Link Control)?
- On an Asynchronous Transfer Mode interface, do the counters include the cell header?
- Why don't SNMP counters return the same number as CLI **show** commands?

SNMP Counters and show Command Equivalent Questions

- What do Cisco routers do for the following SNMP MIB variables: ifInOctets, ifInUcastPkts, ifInNUcastPkts, ifInDiscards, ifInErrors, ifInUnknownProtos, ifOutOctets, ifOutUcastPkts, ifOutNUcastPkts, ifOutDiscards, ifOutErrors, and ifOutQLen?
- What is the relationship etween the show interfaces statements "no buffers" and "input queue drops?" Why do the inDiscards of SNMP give "no buffers" counts and not "input queue drops," while the outDiscards of SNMP do give "output queue drops?"
- Can I poll "no buffers" on a router?
- How do I poll "queue limit drops" on a router?

SNMP Counters Questions

Q. Which MIB should I use for interface counters?

A. Interface management over SNMP is based on two tables: ifTable and its extension, ifXTable described in RFC1213/RFC2233. Interfaces can have several layers, depending on the media, and each sub−layer is represented by a separate row in the table. The relationship between the higher layer and lower layers is described in the ifStackTable. The ifTable defines 32−bit counters for inbound and outbound octets (ifInOctets / ifOutOctets), packets (ifInUcastPkts / ifOutUcastPkts , ifInNUcastPkts / ifOutNUcastPkts), errors, and discards. The ifXTable provides similar 64−bit counters, also called high capacity (HC) counters: ifHCInOctets / ifHCOutOctets , and ifHCInUcastPkts / ifHCOutUcastPkts.

Q. When should 64−bit counters be used?

A. RFC 2233 adopted expanded 64−bit counters for high capacity interfaces in which 32−bit counters do not provide enough capacity and wrap too fast.

As the speed of network media increases, the minimum time in which a 32−bit counter wraps decreases. For example, a 10 Mbps stream of back−to−back, full−size packets causes ifInOctets to wrap in just over 57 minutes. At 100 Mbps, the minimum wrap time is 5.7 minutes, and at 1 Gbps, the minimum is 34 seconds.

Note: The SNMP counters wrap, the CLI counters don't.

For interfaces that operate at 20,000,000 (20 million) bits per second or less, you must use 32−bit byte and packet counters. For interfaces that operate faster than 20 million bits per second, and slower than 650,000,000 bits per second, you must use 32−bit packet counters and 64−bit octet counters. For interfaces that operate at 650,000,000 bits/second or faster, 64−bit packet and octet counters must be used.

Correspondingly, Cisco IOS Software does not support 64−bit counters for interface speeds of less than 20Mbps. This means that 64−bit counters are not supported on 10Mb Ethernet ports, only 100 Mb Fast−Ethernet and other high speed ports support 64−bit counters.

Q. Which version of SNMP is required to query 64−bit counters?

A. SNMPv2C or SNMPv3 is required to query 64−bit counters. SNMPv1 does not support 64−bit counters. Be aware that "ifInOctets = .1.3.6.1.2.1.2.2.1.10" is a 32−bit counter while the 64−bit version is "ifHCInOctets = .1.3.6.1.2.1.31.1.1.1.6".

For example:

Catalyst 5000 uses HP OpenView snmpget, which defaults to SNMPv1

```
# snmpget −c public 14.32.5.18 ifName.1


ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.1 DISPLAY STRING− (ascii) sc0
```

Query with SNMPv1, default for HP OpenView snmpget

```
# snmpget −c public 14.32.5.18 ifHCInOctets.1


snmpget Agent reported error with variable #1.
.iso.org.dod.internet.mgmt.mib−2.ifMIB.ifMIBObjects.ifXTable.ifXEntry.
   ifHCInOctets.1


SNMP Variable does not exist or access is denied.
```

Same query with SNMPv2C

```
# snmpget −v 2c −c public 14.32.5.18 ifHCInOctets.1


ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifHCInOctets.1 Counter64 622366215
```

Q. Which Cisco devices implement 64−bit SNMP counters, especially for the IF−MIB?

A. The following Cisco devices implement 64−bit SNMP counters:

Note: To follow the bug ID links below and see detailed bug information, you must be a registered user and you must be logged in.

- ◊ Cisco 2500, 2600 routers − do not support 64−bit counters.
- ◊ Catalyst 2950 and 3550 − support begins in 12.1(11)EA1 since CSCdx67611 and CSCdw52807
- ◊ Catalyst 2900XL and 3500XL − support begins in 12.0(5)WC3 since CSCds45300
- ◊ Catalyst 5000 series − since Cisco IOS Software release 3.x. On RSM/RSFC support begins in 12.1(6)E1 since CSCds50549
- ◊ Catalyst 5000/6000 ATM modules − since 12.0(14)W05(20), see CSCds07238
- ◊ Catalyst 6000 Gigabit Ethernet WAN OSM − since 12.1.12E, see CSCdw64849
- ◊ Catalyst 6000 series − all Cisco IOS Software releases. WS−F6K−MSFC and MSM support begins in 12.1(8a)E4.
- ◊ Catalyst 8500 series − as of Cisco IOS Software 12.0(5)W5(13)
- ◊ Cisco routers 3600, 4000, and higher platforms − as of Cisco IOS Software 12.0(1), 12.0(1)T CSCdj93712 CSCdt58029
- ◊ Frame Relay interfaces − as of 12.0(17)S, 12.2(4)T3, read Frame Relay 64−Bit Counters
- ◊ OC3 ATM interfaces − as of Cisco IOS Software 12.0(6)T CSCdm45357
- ◊ Tunnel interfaces − as of 12.0(16)S see CSCdt58029

Note: Cisco IOS Software does not support 64−bit counters for interface speeds of less than 20Mbps. This means that 64−bit counters are not supported on 10Mb Ethernet ports! Only 100 Mb Fast−Ethernet and other high speed ports support 64−bit counters.

Q. Are ifInOctets and ifOutOctets SNMP counters the same as the show interfaces In/Out counters?

A. Yes, but only when SNMP is enabled from boot time. If you power on a Cisco device, then enable SNMP, the SNMP counters start from 0. They do not automatically pick up their values from the CLI output.

Q. Do the ifInOctets and ifOutOctets counters include framing overhead (Point−to−Point Protocol, High−Level Data Link Control)?

A. Yes.

Q. On an Asynchronous Transfer Mode interface, do the counters include the cell header?

A. Asynchronous Transfer Mode (ATM) counters do not include ATM overhead (cell headers and AAL5 padding).

Q. Why don't SNMP counters return the same number as CLI `show` commands?

A. An SNMP object defined as a Counter must abide by RFC1155 :

> "3.2.3.3. Counter
> This application−wide type represents a non−negative integer which monotonically increases until it reaches a maximum value, when it wraps around and starts increasing again from zero. This memo specifies a maximum value of 2^32−1 (4294967295 decimal) for counters."

There are no methods to reset an SNMP counter to zero short of reloading the device.

The counter output from a CLI **show** command can be reset on interfaces since the SNMP restrictions are not present.

The original interface counters defined in MIB−2 are 32−bit counters. For a 10 Mbps interface, a 32−bit counter could theoretically wrap in 57 minutes. Avoiding discontinuities is easy with such a long period. But for 100 Mbps, the minimum theoretical wrap time is 5.7 minutes. For 1 Gbps interfaces, it falls to 34 seconds. Granted these times are for transmission of back−to−back full−sized packets, a theoretical ideal. Even so, the higher the interface speed, the harder it becomes to avoid missing a counter wrap. As a solution to this problem, SNMPv2 SMI defined a new object type −− counter64 −− for 64−bit counters. Therefore, there are several new 64−bit counters defined in the extension interface table (ifxTable) defined in RFC 1573 (later superceded by RFC 2233). These are from the IF−MIB−V1SMI.my.

ifHCInOctets (.1.3.6.1.2.1.31.1.1.1.6)	**ifHCOutOctets (1.3.6.1.2.1.31.1.1.1.10)**
ifHCInUcastPkts (.1.3.6.1.2.1.31.1.1.1.7)	**ifHCOutUcastPkts (.1.3.6.1.2.1.31.1.1.1.11)**
ifHCInMulticastPkts (.1.3.6.1.2.1.31.1.1.1.8)	**ifHCOutMulticastPkts (.1.3.6.1.2.1.31.1.1.1.12)**
ifHCInBroadcastPkts (.1.3.6.1.2.1.31.1.1.1.9)	**ifHCOutBroadcastPkts (.1.3.6.1.2.1.31.1.1.1.13)**

Although basic support for 64−bit counters was written into Cisco IOS software in version 11.3, starting from Cisco IOS software release 12.0, only ifHCInOctets (.1.3.6.1.2.1.31.1.1.1.6) and ifHCOutOctets (1.3.6.1.2.1.31.1.1.1.10) have been implemented for ATM LANE LEC sub−interfaces only. For Catalyst workgroup switches, 64−bit counter support has been implemented in version 3.1.

Note: You must use SNMPv2c or SNMPv3 protocol to retrieve any counter64 objects.

SNMP Counters and show Command Equivalent Questions

Q. What do Cisco routers do for the following SNMP MIB variables: ifInOctets, ifInUcastPkts, ifInNUcastPkts, ifInDiscards, ifInErrors, ifInUnknownProtos, ifOutOctets, ifOutUcastPkts, ifOutNUcastPkts, ifOutDiscards, ifOutErrors, and ifOutQLen?

A. See the following list for details. These are from the RFC1213−MIB.

ifInNUcastPkts (.1.3.6.1.2.1.2.2.1.12)	are counts of inbound broadcast and multicast packets
ifInDiscards (.1.3.6.1.2.1.2.2.1.13)	are counted as no buffers as reflected in the **show interfaces** command
ifInErrors (.1.3.6.1.2.1.2.2.1.14)	are counts of all input errors as reflected in the **show interfaces** command
ifInUnknownProtos (.1.3.6.1.2.1.2.2.1.15)	are counted as unclassified errors
ifOutOctets (.1.3.6.1.2.1.2.2.1.16)	are counts of the number of bytes output by the interface as shown in the **show interfaces** command
ifOutUcastPkts (.1.3.6.1.2.1.2.2.1.17)	are counts of outbound broadcast and multicast packets

ifOutDiscards (.1.3.6.1.2.1.2.2.1.19)	are counted as output drops as shown in the **show interfaces** command
ifOutErrors (.1.3.6.1.2.1.2.2.1.20)	are counted as output errors as shown in the **show interfaces** command
ifOutQLen (.1.3.6.1.2.1.2.2.1.21)	is the number of packets allowed to be on the output queue as shown in the **show interfaces** command

The variables listed above that do not say they appear in **show interfaces** are not available anywhere other than SNMP.

Examples

This example uses a 3640 running Cisco IOS® software release 12.2(2)T1. The Read−Only (RO) Community String used is public and the Read−Write(RW) Community String used is private. See Configuring SNMP Community Strings on Routers, RSMs, and Switches for details on how to configure SNMP Community Strings on devices.

The following output is typical of the **show ip interface brief** command executed in enable mode:

```
3600#show ip interface brief
Interface    IP−Address      OK? Method Status  Prol
BRI0/0       unassigned      YES NVRAM  administratively down dow
FastEthernet0/0     172.16.99.20    YES NVRAM  up      up
Serial0/0    unassigned      YES NVRAM  down    dow
Serial0/0.1  unassigned      YES unset  down    dow
BRI0/0:1     unassigned      YES unset  administratively down dow
BRI0/0:2     unassigned      YES unset  administratively down dow
Serial0/1    unassigned      YES NVRAM  administratively down dow
ATM1/0       unassigned      YES NVRAM  down    dow
ATM1/0.109   10.164.0.46     YES NVRAM  down    dow
Virtual−Template1   99.99.99.99     YES NVRAM  down    dow
Loopback0    10.1.10.1       YES NVRAM  up      up
Loopback1    unassigned      YES NVRAM  up      up
Loopback101  3.3.3.3  YES NVRAM  administratively down dow
Loopback200  4.4.4.14 YES NVRAM  administratively down dow
Loopback201  4.4.4.18 YES NVRAM  administratively down dow
```

The following output is the MIB Object **if Descr (.1.3.6.1.2.1.2.2.1.2)** for the above router, which is a text string containing information about the interface. This gives the interface name and description as obtained using the above command line interface (CLI) command output. **ifName (.1.3.6.1.2.1.31.1.1.1.1)** can also be used but **ifDescr** gives the interface description along with the name, where **ifName** only gives the interface name.

```
snmpwalk 172.16.99.20 public .1.3.6.1.2.1.2.2.1.2

interfaces.ifTable.ifEntry.ifDescr.1 = ATM1/0
interfaces.ifTable.ifEntry.ifDescr.2 = BRI0/0
interfaces.ifTable.ifEntry.ifDescr.3 = FastEthernet0/0
interfaces.ifTable.ifEntry.ifDescr.4 = Serial0/0
interfaces.ifTable.ifEntry.ifDescr.5 = BRI0/0:1
interfaces.ifTable.ifEntry.ifDescr.6 = BRI0/0:2
interfaces.ifTable.ifEntry.ifDescr.7 = Serial0/1
interfaces.ifTable.ifEntry.ifDescr.8 = Null0
interfaces.ifTable.ifEntry.ifDescr.10 = Foreign Exchange Office 2/0/0
interfaces.ifTable.ifEntry.ifDescr.11 = Foreign Exchange Office 2/0/1
interfaces.ifTable.ifEntry.ifDescr.12 = recEive And transMit 3/0/0
interfaces.ifTable.ifEntry.ifDescr.13 = recEive And transMit 3/0/1
interfaces.ifTable.ifEntry.ifDescr.14 = Loopback0
interfaces.ifTable.ifEntry.ifDescr.15 = Loopback1
interfaces.ifTable.ifEntry.ifDescr.16 = Loopback101
interfaces.ifTable.ifEntry.ifDescr.17 = Loopback200
interfaces.ifTable.ifEntry.ifDescr.18 = Loopback201
interfaces.ifTable.ifEntry.ifDescr.19 = Serial0/0.1
interfaces.ifTable.ifEntry.ifDescr.20 = ATM1/0.109−atm subif
interfaces.ifTable.ifEntry.ifDescr.21 = ATM1/0.109−aal5 layer
interfaces.ifTable.ifEntry.ifDescr.22 = Virtual−Template1
interfaces.ifTable.ifEntry.ifDescr.23 = Voice Encapsulation (POTS) Peer: 1
interfaces.ifTable.ifEntry.ifDescr.24 = Voice Over IP Peer: 2
interfaces.ifTable.ifEntry.ifDescr.25 = Voice Encapsulation (POTS) Peer: 111
interfaces.ifTable.ifEntry.ifDescr.26 = Voice Over IP Peer: 222
interfaces.ifTable.ifEntry.ifDescr.27 = Voice Over IP Peer: 1234
interfaces.ifTable.ifEntry.ifDescr.28 = Voice Over IP Peer: 300000
interfaces.ifTable.ifEntry.ifDescr.29 = Voice Over FR Peer: 3
interfaces.ifTable.ifEntry.ifDescr.30 = Voice Over IP Peer: 99
interfaces.ifTable.ifEntry.ifDescr.31 = Voice Encapsulation (POTS) Peer: 9
interfaces.ifTable.ifEntry.ifDescr.32 = BRI0/0−Physical
interfaces.ifTable.ifEntry.ifDescr.33 = BRI0/0−Signaling
interfaces.ifTable.ifEntry.ifDescr.34 = BRI0/0:1−Bearer Channel
interfaces.ifTable.ifEntry.ifDescr.35 = BRI0/0:2−Bearer Channel
```

6. ifInDiscards (.1.3.6.1.2.1.2.2.1.13):

```
snmpwalk 172.16.99.20 public .1.3.6.1.2.1.2.2.1.13

interfaces.ifTable.ifEntry.ifInDiscards.1 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.2 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.3 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.4 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.5 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.6 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.7 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.8 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.10 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.11 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.12 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.13 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.14 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.15 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.16 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.17 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.18 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.19 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.20 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.21 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.22 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.23 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.24 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.25 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.26 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.27 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.28 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.29 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.30 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.31 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.32 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.33 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.34 = Counter32: 0
interfaces.ifTable.ifEntry.ifInDiscards.35 = Counter32: 0
```

The **ifInDiscards** is zero for all the interfaces of this router. Comparing this with the CLI result of the **show interfaces fastEthernet 0/0** command confirms the result:

```
3600#show interfaces fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0001.42b4.fe81 (bia 0001.42b4.fe81)
  Description: testme
  Internet address is 172.16.99.20/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full−duplex, 100Mb/s, 100BaseTX/FX
  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
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 323 drops
  5 minute input rate 1000 bits/sec, 2 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1767411 packets input, 178272010 bytes
     Received 1161500 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
7146925 packets output, 765049281 bytes, 0 underruns(0/0/0)
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 461 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

5. ifInErrors (.1.3.6.1.2.1.2.2.1.14):

```
snmpwalk 172.16.99.20 public .1.3.6.1.2.1.2.2.1.14

interfaces.ifTable.ifEntry.ifInErrors.1 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.2 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.3 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.4 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.5 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.6 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.7 = Counter32: 1
interfaces.ifTable.ifEntry.ifInErrors.8 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.10 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.11 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.12 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.13 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.14 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.15 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.16 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.17 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.18 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.19 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.20 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.21 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.22 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.23 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.24 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.25 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.26 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.27 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.28 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.29 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.30 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.31 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.32 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.33 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.34 = Counter32: 0
interfaces.ifTable.ifEntry.ifInErrors.35 = Counter32: 0
```

This output shows that there is only one input error for the interface

interfaces.ifTable.ifEntry.ifInErrors.7 = Counter32: 1.

To determine which interface this is, compare it with the output of **ifDescr** above, which shows that this is from **interfaces.ifTable.ifEntry.ifDescr.7 = Serial0/1.** Now executing the CLI **show interfaces serial 0/1** command in enable mode verifies the above result:

```
3600#show interfaces serial 0/1
Serial0/1 is administratively down, line protocol is down
  Hardware is DSCC4 Serial
  Description: atm−dxi test
  MTU 1500 bytes, BW 2048 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM−DXI, loopback not set
  Keepalive not set
  Last input never, output never, output hang never
  Last clearing of "show interface" counters 1w1d
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     1 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 1 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
     DCD=down  DSR=down  DTR=down  RTS=down  CTS=down
```

1. ifOutOctets (.1.3.6.1.2.1.2.2.1.16):

```
snmpwalk 172.16.99.20 public .1.3.6.1.2.1.2.2.1.16

interfaces.ifTable.ifEntry.ifOutOctets.1 = Counter32: 98
interfaces.ifTable.ifEntry.ifOutOctets.2 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.3 = Counter32: 765470674
interfaces.ifTable.ifEntry.ifOutOctets.4 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.5 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.6 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.7 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.8 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.10 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.11 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.12 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.13 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.14 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.15 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.16 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.17 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.18 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.19 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.20 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.21 = Counter32: 98
interfaces.ifTable.ifEntry.ifOutOctets.22 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.23 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.24 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.25 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.26 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.27 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.28 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.29 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.30 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.31 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.32 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.33 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.34 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutOctets.35 = Counter32: 0
```

Comparing the above result with the output of the **ifDescr** indicates:

- **interfaces.ifTable.ifEntry.ifOutOctets.1 = Counter32: 98** corresponds with **interfaces.ifTable.ifEntry.ifDescr.1 = ATM1/0**
- **interfaces.ifTable.ifEntry.ifOutOctets.3 = Counter32: 765470674**corresponds with i**nterfaces.ifTable.ifEntry.ifDescr.3 = FastEthernet0/0** and
- **interfaces.ifTable.ifEntry.ifOutOctets.21 = Counter32: 98** corresponds with **interfaces.ifTable.ifEntry.ifDescr.21 = ATM1/0.109−aal5 layer**

The output of the CLI **show interfaces** command for each of the above interfaces executed in enable mode follows:

```
3600#show interfaces atM 1/0
   ATM1/0 is down, line protocol is down
  Hardware is RS8234 ATMOC3
  MTU 4470 bytes, sub MTU 4470, BW 155000 Kbit, DLY 80 usec,
     reliability 5/255, txload 1/255, rxload 1/255
  Encapsulation ATM, loopback not set
  Encapsulation(s): AAL5
  1024 maximum active VCs, 1 current VCCs
  VC idle disconnect time: 300 seconds
  Last input never, output 1w1d, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: None
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     2 packets output, 98 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

```
3600#show interfaces fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0001.42b4.fe81 (bia 0001.42b4.fe81)
  Description: testme
  Internet address is 172.16.99.20/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full−duplex, 100Mb/s, 100BaseTX/FX
  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
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 323 drops
  5 minute input rate 2000 bits/sec, 3 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
     1772214 packets input, 178767841 bytes
     Received 1164210 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog
     0 input packets with dribble condition detected
     7149179 packets output, 765450524 bytes, 0 underruns(0/0/0)
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 461 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

The output of the **ifOutOctets** does not match the CLI output for the **show interfaces FastEthernet 0/0** command, but it is similar. This is because there might be a delay when the interfaces are polled and when the CLI command is executed.

```
3600#show interfaces atM 1/0.109
ATM1/0.109 is down, line protocol is down
  Hardware is RS8234 ATMOC3
  Description: pvc
  Internet address is 10.164.0.46/30
  MTU 4470 bytes, BW 2250 Kbit, DLY 80 usec,
     reliability 5/255, txload 1/255, rxload 1/255
  Encapsulation ATM
  0 packets input, 0 bytes
  2 packets output,98 bytes
  0 OAM cells input, 77093 OAM cells output
  AAL5 CRC errors : 0
  AAL5 SAR Timeouts : 0
  AAL5 Oversized SDUs : 0
  AAL5 length violation : 0
  AAL5 CPI Error : 0
```

4. ifOutDiscards (.1.3.6.1.2.1.2.2.1.19):

```
snmpwalk 172.16.99.20 public .1.3.6.1.2.1.2.2.1.19

interfaces.ifTable.ifEntry.ifOutDiscards.1 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.2 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.3 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.4 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.5 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.6 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.7 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.8 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.10 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.11 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.12 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.13 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.14 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.15 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.16 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.17 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.18 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.19 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.20 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.21 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.22 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.23 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.24 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.25 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.26 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.27 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.28 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.29 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.30 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.31 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.32 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.33 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.34 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutDiscards.35 = Counter32: 0
```

The **ifOutDiscards** is zero for all the interfaces. Using the **show interfaces fastEthernet 0/0** command as an example, produces the following result:

```
3600#show interfaces fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0001.42b4.fe81 (bia 0001.42b4.fe81)
  Description: testme
  Internet address is 172.16.99.20/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full−duplex, 100Mb/s, 100BaseTX/FX
  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
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 323 drops
  5 minute input rate 1000 bits/sec, 2 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
     1774581 packets input, 179005552 bytes
     Received 1165620 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
     7150259 packets output, 765645035 bytes, 0 underruns(0/0/0)
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 461 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

3. ifOutErrors (.1.3.6.1.2.1.2.2.1.20):

```
snmpwalk 172.16.99.20 public .1.3.6.1.2.1.2.2.1.20

interfaces.ifTable.ifEntry.ifOutErrors.1 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.2 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.3 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.4 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.5 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.6 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.7 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.8 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.10 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.11 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.12 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.13 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.14 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.15 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.16 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.17 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.18 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.19 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.20 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.21 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.22 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.23 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.24 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.25 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.26 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.27 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.28 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.29 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.30 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.31 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.32 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.33 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.34 = Counter32: 0
interfaces.ifTable.ifEntry.ifOutErrors.35 = Counter32: 0
```

The **ifOutErrors** is zero for all the interfaces. Using the **show interfaces fastEthernet 0/0** command as an example, produces the following:

```
3600#show interfaces fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0001.42b4.fe81 (bia 0001.42b4.fe81)
  Description: testme
  Internet address is 172.16.99.20/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full−duplex, 100Mb/s, 100BaseTX/FX
  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
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 323 drops
  5 minute input rate 0 bits/sec, 1 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1776187 packets input, 179154616 bytes
     Received 1166778 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
     7150781 packets output, 765744231 bytes, 0 underruns(0/0/0)
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 461 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

2. ifOutQLen (.1.3.6.1.2.1.2.2.1.21):

```
snmpwalk 172.16.99.20 public .1.3.6.1.2.1.2.2.1.21

interfaces.ifTable.ifEntry.ifOutQLen.1 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.2 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.3 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.4 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.5 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.6 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.7 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.8 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.10 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.11 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.12 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.13 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.14 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.15 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.16 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.17 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.18 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.19 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.20 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.21 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.22 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.23 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.24 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.25 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.26 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.27 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.28 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.29 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.30 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.31 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.32 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.33 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.34 = Gauge32: 0
interfaces.ifTable.ifEntry.ifOutQLen.35 = Gauge32: 0
```

The **ifOutQLen** is zero for all the interfaces. Using the **show interfaces fastEthernet 0/0** command as an example:

```
3600#show interfaces fastEthernet 0/0
   FastEthernet0/0 is up, line protocol is up
     Hardware is AmdFE, address is 0001.42b4.fe81 (bia 0001.42b4.fe81)
     Description: testme
     Internet address is 172.16.99.20/24
     MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
        reliability 255/255, txload 1/255, rxload 1/255
     Encapsulation ARPA, loopback not set
     Keepalive set (10 sec)
     Full−duplex, 100Mb/s, 100BaseTX/FX
     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
     Queueing strategy: fifo
     Output queue 0/40, 0 drops; input queue 0/75, 323 drops
     5 minute input rate 0 bits/sec, 1 packets/sec
     5 minute output rate 0 bits/sec, 0 packets/sec
        1776912 packets input, 179225431 bytes
        Received 1167240 broadcasts, 0 runts, 0 giants, 0 throttles
        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
        0 watchdog
        0 input packets with dribble condition detected
        7151102 packets output, 765796341 bytes, 0 underruns(0/0/0)
        0 output errors, 0 collisions, 1 interface resets
        0 babbles, 0 late collision, 461 deferred
        0 lost carrier, 0 no carrier
        0 output buffer failures, 0 output buffers swapped out
```

Q. What is the relationship between the `show interfaces` statements "no buffers" and "input queue drops?" Why do the inDiscards of SNMP give "no buffers" counts and not "input queue drops," while the outDiscards of SNMP do give "output queue drops?"

A.The locIfInputQueueDrops/ifInDiscards work differently than locIfOutputQueueDrops/ifOutDiscards. The ifInDiscards counts the number of packets that were thrown away for lack of a system resource such as a buffer. This is generally a subset of the locIfInputQueueDrops. You often see locIfInputQueueDrops = ifInDiscards. However, locIfInputQueueDrops also counts the number of packets dropped because of hitting the input queue limit. So more generally, you see locIfInputQueueDrops > ifInDiscards.

Summary

locIfInputQueueDrops = Queue Limit Drops + No Buffer Drops
ifInDiscards = No Buffer Drops (and is a subset of locIfInputQueueDrops)

The locIfOutputQueueDrops and ifOutDiscards are always equal when they count the same events. Those events are hitting the output queue limit, and not having a hardware tx buffer when a packet is fastswitched from one interface to another. The OIDs of the above MIB Objects are as follows:

From OLD−CISCO−INTERFACES−MIB	From RFC1213−MIB
locIfInputQueueDrops = .1.3.6.1.4.1.9.2.2.1.1.26	ifInDiscards = .1.3.6.1.2.1.2.2.1.13
locIfOutputQueueDrops = .1.3.6.1.4.1.9.2.2.1.1.27	ifOutDiscards = .1.3.6.1.2.1.2.2.1.19

Q.Can I poll "no buffers" on a router?

A. Yes. You can poll no buffers by polling for ifInDiscards.

Q. How do I poll "queue limit drops" on a router?

A. Using SNMP, there is no way for the **show interfaces** command to break out the individual elements that go into the output drops.

Consider this new information about what goes into the output drops counter:

Input drops ==
Queue limit drops +
Throttling drops +
SMT queue full drops +
RSRB drops +
no buffer drops

In addition, SNMP counters are never cleared, even if the interfaces are cleared.

Related Information

- **IP Application Services Technical Tips**

Updated: Nov 19, 2004 Document ID: 26007