

Contadores SNMP: Preguntas Frecuentes

Contenido

[Introducción](#)

[Preguntas de los contadores SNMP](#)

[Contadores SNMP y preguntas del equivalente del comando show](#)

[Información Relacionada](#)

Introducción

Este documento proporciona respuestas a preguntas frecuentes y suministra una guía para que los usuarios puedan encontrar recursos útiles y temas sobre SNMP asociados al equipo Cisco.

Preguntas de los contadores SNMP

Q. ¿Qué MIB debo utilizar para los contadores de la interfaz?

A. Basan a la administración de la interfaz sobre el SNMP en dos tablas: [ifTable](#) ([clientes registrados solamente](#)) y su extensión, [ifXTable](#) ([clientes registrados solamente](#)) descrita en el RFC1213/RFC2233. Las interfaces pueden tener varias capas, que depende de los media, y cada substrato es representado por una fila separada en la tabla. La relación entre la capa superior y las capas inferiores se describe en el [ifStackTable](#) ([clientes registrados solamente](#)). El ifTable define los contadores de 32 bits para entrante y los octetos de salida ([ifInOctets](#) ([clientes registrados solamente](#)) /[ifOutOctets](#)), los paquetes ([ifInUcastPkts](#) ([clientes registrados solamente](#)) /[ifOutUcastPkts](#) ([clientes registrados solamente](#)), [ifInNUcastPkts](#) ([clientes registrados solamente](#)) /[ifOutNUcastPkts](#) ([clientes registrados solamente](#))), los errores, y los descartes. El ifXTable proporciona los contadores 64-bit similares, también llamados los contadores de la alta capacidad (HC): [ifHCInOctets](#) ([clientes registrados solamente](#)) /[ifHCOutOctets](#) ([clientes registrados solamente](#)), e [ifHCInUcastPkts](#) ([clientes registrados solamente](#)) /[ifHCOutUcastPkts](#) ([clientes registrados solamente](#)).

Q. ¿Cuándo deben los contadores 64-bit ser utilizados?

A. [El RFC 2233](#) adoptado amplió los contadores 64-bit para las interfaces de la alta capacidad en las cuales los contadores de 32 bits no proporcionan bastante capacidad y envuelven demasiado rápido.

Como la velocidad de los medios de red aumenta, el tiempo mínimo en el cual un contador de 32 bits envuelve las disminuciones. Por ejemplo, una secuencia de 10 Mbps de paquetes adosados de tamaño completo hace que ifInOctets se repliegue en un poco más de 57 minutos. A 100Mbps, el tiempo mínimo de cobertura es de 5.7 minutos, y a 1 Gbps, el mínimo es de 34 segundos.

Note: El abrigo de los contadores SNMP, los contadores del comando line interface(cli) no hace.

Para las interfaces que operan con 20.000.000 (20 millones) de bits por segundo o menos, deberá utilizar contadores de paquetes y byte de 32 bits. Para las interfaces que actúan más rápidamente de 20 millones de bits por segundo, y más lentamente de 650,000,000 bits por segundo, usted debe utilizar los contadores de paquetes de 32 bits y a los contadores de octeto 64-bit. Para las interfaces que actúan en los dígitos por segundo 650,000,000 o un paquete y contadores de octeto más rápidos, 64-bit debe ser utilizado.

Correspondientemente, el software de Cisco IOS® no soporta los contadores 64-bit para las velocidades de la interfaz menos que el 20 Mbps. Esto significa que los contadores 64-bit no están soportados en los accesos de Ethernet del 10 Mb, sólo las normas Ethernet rápidas del 100 Mb y otros puertos de alta velocidad soportan los contadores 64-bit.

Q. ¿Qué versión del SNMP se requiere para preguntar los contadores 64-bit?

A. El SNMPv2C o el SNMPv3 se requiere para preguntar los contadores 64-bit. El SNMPv1 no soporta los contadores 64-bit. Sea consciente que el ifInOctets = .1.3.6.1.2.1.2.2.1.10 es un contador de 32 bits mientras que la versión 64-bit es ifHCInOctets = .1.3.6.1.2.1.31.1.1.1.6.

Por ejemplo:

El Catalyst 5000 utiliza el snmpget del HP OpenView, que omite el SNMPv1

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

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

Interrogación con el SNMPv1, valor por defecto para el snmpget del HP OpenView

```
# 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.
```

La misma interrogación con el SNMPv2C

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

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

Q. ¿Qué dispositivos de Cisco implementan a los contadores SNMP 64-bit, especialmente para el IF-MIB?

A. Estos dispositivos de Cisco implementan a los contadores SNMP 64-bit:

Note: Usted debe ser usuario registrado y usted debe ser abierto una sesión para visitar los links del Id. de bug Cisco y ver la información de bug detallada.

- Cisco2500, 2600 Router — Este Routers no soporta los contadores 64-bit.
- Catalyst 2950 y 3550 — El soporte comienza en el Cisco IOS Software Release 12.1(11)EA1 desde el Id. de bug Cisco [CSCdx67611](#) ([clientes registrados solamente](#)) y el Id. de bug Cisco [CSCdw52807](#) ([clientes registrados solamente](#)).
- Catalyst 2900XL y 3500XL — El soporte comienza en el Cisco IOS Software Release 12.0(5)WC3 desde el Id. de bug Cisco [CSCds45300](#) ([clientes registrados solamente](#)).
- Catalyst 5000 Series — Desde el Cisco IOS Software Release 3.x. En el RS /RSFC, el soporte comienza en el Cisco IOS Software Release 12.1(6)E1 desde el Id. de bug Cisco [CSCds50549](#) ([clientes registrados solamente](#)).
- Módulos ATM del Catalyst 5000/6000 — Desde el Cisco IOS Software Release 12.0(14)W05(20), refiera al Id. de bug Cisco [CSCds07238](#) ([clientes registrados solamente](#)).
- Catalyst 6000 Gigabit Ethernet WAN OS — Desde el Cisco IOS Software Release 12.1.12E, refiera al Id. de bug Cisco [CSCdw64849](#) ([clientes registrados solamente](#)).
- Catalyst 6000 Series — Todas las versiones de Cisco IOS Software. El soporte WS-F6K-MSFC y MS comienza en el Cisco IOS Software Release 12.1(8a)E4.
- Catalyst 8500 Series — El soporte comienza a partir del Cisco IOS Software Release 12.0(5)W5(13).
- Los routers Cisco 3600, 4000, y plataformas mayores — a partir del Cisco IOS Software 12.0(1) y del Cisco IOS Software Release 12.0(1)T, refieren al Id. de bug Cisco [CSCdj93712](#) ([clientes registrados solamente](#)) y al Id. de bug Cisco [CSCdt58029](#) ([clientes registrados solamente](#)).
- Interfaces de Frame Relay — A partir del Cisco IOS Software Release 12.0(17)S y del Cisco IOS Software Release 12.2(4)T3, refiera al [Frame Relay 64-bit Counters](#).
- Interfaces ATM del OC3 — A partir del Cisco IOS Software Release 12.0(6)T, refiera al Id. de bug Cisco [CSCdm45357](#) ([clientes registrados solamente](#)).
- Interfaces del túnel — A partir del Cisco IOS Software Release 12.0(16)S, refiera al Id. de bug Cisco [CSCdt58029](#) ([clientes registrados solamente](#)).

Note: El Cisco IOS Software no soporta los contadores 64-bit para las velocidades de la interfaz menos que el 20 Mbps. Esto significa que los contadores 64-bit no están soportados en los accesos de Ethernet del 10 Mb. Solamente las normas Ethernet rápidas del 100 Mb y otros puertos de alta velocidad soportan los contadores 64-bit.

Q. ¿Son el ifInOctets y los contadores SNMP del ifOutOctets lo mismo que la demostración interconecta los contadores del in/out?

A. Sí, pero solamente cuando el SNMP se habilita a partir del tiempo del inicio. Si usted acciona encendido un dispositivo de Cisco, después habilite el SNMP, los contadores SNMP salen de la 0. No cogen automáticamente sus valores de la salida CLI.

Q. ¿El ifInOctets y los contadores del ifOutOctets incluyen las tareas generales de alineación tramas (protocolo Point-to-Point, High-Level Data Link Control)?

A. Yes.

Q. ¿En una interfaz del Asynchronous Transfer Mode, los contadores incluyen el encabezamiento de la célula?

A. Los contadores del Asynchronous Transfer Mode (ATM) no incluyen la overhead de ATM (los

encabezamientos de la célula y relleno de AAL5).

Q. ¿Por qué los contadores SNMP no vuelven el mismo número que los comandos cli show?

A. Un objeto SNMP definido como contador debe seguir el [RFC1155](#) :

"3.2.3.3. Contador

Este tipo global de aplicación representa un número entero no negativo que monótono los aumentos hasta que alcance un valor máximo, cuando envuelve alrededor y comienza a aumentar otra vez a partir de la cero. Esta nota especifica un valor máximo de $2^{32}-1$ (el decimal 4294967295) para los contadores."

No hay métodos para reajustar un SNMP en dirección contraria cero sin la necesidad de recargar el dispositivo.

La salida de contador de un comando cli show puede ser reajustada en las interfaces puesto que las restricciones de SNMP no están presentes.

Los contadores de la interfaz originales definidos en el MIB-2 son contadores de 32 bits. Para una interfaz del 10 Mbps, un contador de 32 bits podía envolver teóricamente en 57 minutos. Es fácil evitar las discontinuidades con tal período prolongado. Pero para el 100 Mbps, el tiempo de repliegue del teórico mínimo es 5.7 minutos. Para las interfaces del 1 Gbps, baja a 34 segundos. Concedió estas épocas están para la transmisión de los paquetes de tamaño completo continuos, un ideal teórico. Sin embargo, cuanto más alta es la velocidad de la interfaz, más duro llega a ser para evitar faltar un abrigo contrario. Como solución a este problema, SNMPv2 S I definió un nuevo tipo de objeto, counter64, para los contadores 64-bit. Por lo tanto, hay varios nuevos contadores 64-bit definidos en la tabla de la interfaz de la extensión (ifxTable) definida en el [RFC 1573](#) (reemplazado más adelante por el [RFC 2233](#)). [Éstos son del IF-MIB-V1SMI.my \(clientes registrados solamente\)](#).

ifHCInOctets (.1.3.6.1.2.1.31.1.1.1.6)	ifHCOctets (1.3.6.1.2.1.31.1.1.1.10)
ifHCInUcastPkts (.1.3.6.1.2.1.31.1.1.1.7)	ifHCOUcastPkts (.1.3.6.1.2.1.31.1.1.1.11)
ifHCInMulticastPkts (.1.3.6.1.2.1.31.1.1.1.8)	ifHCOMulticastPkts (.1.3.6.1.2.1.31.1.1.1.12)
ifHCInBroadcastPkts (.1.3.6.1.2.1.31.1.1.1.9)	ifHCOBroadcastPkts (.1.3.6.1.2.1.31.1.1.1.13)

Aunque el soporte básico para los contadores 64-bit fuera escrito en el Cisco IOS Software Release 11.3, que empieza con el Cisco IOS Software Release 12.0, sólo los ifHCInOctets (.1.3.6.1.2.1.31.1.1.1.6) y los ifHCOctets (1.3.6.1.2.1.31.1.1.1.10) se han implementado para los subinterfaces LANE ATM LEC solamente. Para el Catalyst workgroup switches, el soporte de contador 64-bit se ha implementado en la versión 3.1.

Note: Usted debe utilizar el protocolo SNMPv2C o del SNMPv3 para extraer cualquier objeto del contador 64.

Contadores SNMP y preguntas del equivalente del comando show

Q. Qué hacen los routers Cisco hacen para las variables siguientes del SNMP MIB: ¿el ifInOctets, los ifInUcastPkts, los ifInNUcastPkts, los ifInDiscards, ifInErrors, los ifInUnknownProtos, ifOutOctets, los ifOutUcastPkts, los ifOutNUcastPkts, los ifOutDiscards, los ifOutErrors, e ifOutQLen?

A. Vea esta tabla para los detalles. Éstos son del [RFC1213-MIB](#) (clientes registrados solamente).

ifInNUcastPkts (.1.3.6.1.2.1.2.2.1.12)	Éstas son cuentas de broadcast entrante y de los paquetes de multidifusión.
ifInDiscards (.1.3.6.1.2.1.2.2.1.13)	Éstos se cuentan como ningunos buffers como se refleja en el comando show interfaces .
ifInErrors (.1.3.6.1.2.1.2.2.1.14)	Éstas son cuentas de todos los errores de entrada como se refleja en el comando show interfaces .
ifInUnknownProtos (.1.3.6.1.2.1.2.2.1.15)	Éstos se cuentan como errores sin clasificar.
ifOutOctets (.1.3.6.1.2.1.2.2.1.16)	Éstas son cuentas de la salida de la cantidad de bytes al lado de la interfaz tal y como se muestra en del comando show interfaces .
ifOutUcastPkts (.1.3.6.1.2.1.2.2.1.17)	Éstas son cuentas de los broadcastes salientes y paquetes de multicast.
ifOutDiscards (.1.3.6.1.2.1.2.2.1.19)	Éstos se cuentan como caídas de resultados tal y como se muestra en del comando show interfaces .
ifOutErrors (.1.3.6.1.2.1.2.2.1.20)	Éstos se cuentan como errores de salida tal y como se muestra en del comando show interfaces .
ifOutQLen (.1.3.6.1.2.1.2.2.1.21)	Éste es el número de paquetes permitidos estar en la cola de salida tal y como se muestra en del comando show interfaces .

Las variables enumeraron previamente que no las dicen aparecen en las **interfaces de la demostración** no están disponibles dondequiera con excepción del SNMP.

Ejemplos

Este ejemplo utiliza 3640 que se ejecuta con el Cisco IOS Software Release 12.2(2)T1. La cadena de comunidad solo lectura (RO) usada es pública y la cadena de comunidad de la lectura-Write (RW) usada es privada. Refiérase a [cómo configurar las cadenas de comunidad SNMP](#) para más información sobre cómo configurar las cadenas de comunidad SNMP en los dispositivos.

Esta salida es típica del comando `show ip interface brief` ejecutado en el 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
```

Esta salida es el objeto de MIB si **Descr (.1.3.6.1.2.1.2.2.1.2)** para el router anterior, que es una cadena de texto que contiene la información sobre la interfaz. Esto da el nombre y la descripción de la interfaz según lo obtenido, que utiliza la salida anterior del comando CLI. **el ifName (.1.3.6.1.2.1.31.1.1.1.1)** se puede también utilizar solamente **ifDescr** da la descripción de la interfaz junto con el nombre, donde el **ifName** da solamente el nombre de la interfaz.

```
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
```

1. 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
```

Los ifInDiscards son cero para todas las interfaces de este router. Si usted compara esto con el resultado CLI del comando `show interfaces fastEthernet 0/0`, éste confirma el resultado:

```
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
```

2. 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

```

Esta salida muestra que hay solamente un error de entrada para la interfaz **interfaces.ifTable.ifEntry.ifInErrors.7 = Counter32: 1**. Para determinar que interconectan esto está, lo compara con la salida del **ifDescr** arriba, que muestra que ésta es de **interfaces.ifTable.ifEntry.ifDescr.7 = Serial0/1**. Ahora ejecute el comando **show interfaces serial 0/1** en el enable mode para verificar el resultado anterior:

```

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

```

3. 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

```

Si usted compara el resultado anterior con la salida del `ifDescr`, esto indica: `interfaces.ifTable.ifEntry.ifOutOctets.1 = Counter32: 98` corresponde con `interfaces.ifTable.ifEntry.ifDescr.1 = ATM1/0` `interfaces.ifTable.ifEntry.ifOutOctets.3 = Counter32: 765470674` corresponde con `interfaces.ifTable.ifEntry.ifDescr.3 = FastEthernet0/0` `interfaces.ifTable.ifEntry.ifOutOctets.21 = Counter32: 98` corresponde con la capa `interfaces.ifTable.ifEntry.ifDescr.21 = ATM1/0.109-aal5` Ésta es la salida del comando `show interfaces` CLI para cada uno de las interfaces anteriores ejecutadas en el `enable` mode:

```

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 lwld, 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

```

La salida del **ifOutOctets** no hace juego el CLI hecho salir para el comando **show interfaces fastEthernet 0/0**, sino que es similar. Esto es porque pudo haber un retardo cuando se sondean las interfaces y cuando ejecutan al comando CLI.

```

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

```

Los ifOutDiscards son cero para todas las interfaces. Con el comando show interfaces fastEthernet 0/0 como un ejemplo, este comando produce este resultado:

```

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

```

5. 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

```

Los ifOutErrors son cero para todas las interfaces. Con el comando show interfaces fastEthernet 0/0 como un ejemplo, este comando produce este resultado:

```

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

```

6. 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
```

El ifOutQLen es cero para todas las interfaces. Con el comando show interfaces fastEthernet 0/0 como un ejemplo:

```
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. ¿Cuál no es la relación entre las declaraciones de las interfaces de la demostración ningunos buffers y caída de entradas en la cola? ¿Por qué los inDiscards del SNMP dan a ningunos buffers las cuentas y no las caídas de entradas en la cola, mientras que los

outDiscards del SNMP dan las pérdidas de la cola de salida?

A. El loclInputQueueDrops/los iflnDiscards trabaja diferentemente que loclOutputQueueDrops/ifOutDiscards. Los iflnDiscards cuentan el número de paquetes que se lancen lejos a falta de un recurso del sistema tal como un buffer. Éste es generalmente un subconjunto del loclInputQueueDrops. Usted ve a menudo el loclInputQueueDrops = los iflnDiscards. Pero, el loclInputQueueDrops también cuenta el número de paquetes caídos porque golpean el límite de la cola de entrada. Tan generalmente, usted ve el loclInputQueueDrops > los iflnDiscards.

Resumen

el loclInputQueueDrops = el límite de cola cae + ningunos iflnDiscards de las caídas del búfer = ningunas caídas del búfer (y es un subconjunto de loclInputQueueDrops)

Los loclOutputQueueDrops y los ifOutDiscards son siempre iguales cuando cuentan los mismos eventos. Esos eventos golpean el límite de cola de resultados, y no tienen un buffer del tx del hardware cuando un paquete fastswitched a partir de una interfaz a otra. Los OID de los objetos de MIB anteriores son éstos:

De OLD-CISCO-INTERFACES-MIB (clientes registrados solamente)	De RFC1213-MIB (clientes registrados solamente)
loclInputQueueDrops = .1.3.6.1.4.1.9.2.2.1.1.26	iflnDiscards = .1.3.6.1.2.1.2.2.1.13
loclOutputQueueDrops = .1.3.6.1.4.1.9.2.2.1.1.27	ifOutDiscards = .1.3.6.1.2.1.2.2.1.19

Q. ¿No puedo sondear ningún buffer en un router?

A. Yes. Usted puede sondear para los iflnDiscards para no sondear ningún buffer.

Q. ¿Cómo sondeo los descensos del límite de cola en un router?

A. Con el uso del SNMP, no hay manera para que el comando **show interfaces** explote los elementos individuales que entran las caídas de resultados.

Considere esta nueva información sobre qué entra las caídas de resultados al revés:

El Input Drops = el límite de cola caen + estrangulando caen + los descensos completos de la cola de SMT + RSRB caen + ningunas caídas del búfer

Además, nunca borran a los contadores SNMP, incluso si se borran las interfaces.

[Información Relacionada](#)

- [Consejos técnicos de los Servicios de aplicación IP](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)