

Considere los contadores SNMP: preguntas frecuentes

Contenido

[Introducción](#)

[Preguntas sobre contadores SNMP](#)

[P. ¿Qué MIB utilizo para los contadores de interfaz?](#)

[P. ¿Cuándo se utilizan los contadores de 64 bits?](#)

[P. ¿Qué versión de SNMP se requiere para consultar los contadores de 64 bits?](#)

[P. ¿Qué dispositivos Cisco implementan contadores SNMP de 64 bits, especialmente para IF-MIB?](#)

[P. ¿Son los contadores SNMP ifInOctets y ifOutOctets los mismos que los contadores show interfaces In/Out?](#)

[P. ¿Los contadores ifInOctets y ifOutOctets incluyen sobrecarga de tramas \(protocolo punto a punto, control de link de datos de alto nivel\)?](#)

[P. En una interfaz de Asynchronous Transfer Mode, ¿los contadores incluyen el encabezado de celda?](#)

[P. ¿Por qué los contadores SNMP no devuelven el mismo número que los comandos CLIshow?](#)

[Contadores SNMP y preguntas equivalentes al comando show](#)

[P. ¿Qué hacen los routers de Cisco para estas variables MIB SNMP: ifInOctets, ifInUcastPkts, ifInNUcastPkts, ifInDiscards, ifInErrors, ifInUnknownProtos, ifOutOctets, ifOutUcastPkts, ifOutNUcastPkts, ifOutDiscards, ifOutErrors y ifOutQLen?](#)

[Examples](#)

[P. ¿Cuál es la relación entre las declaraciones de interfaces de presentación en los búferes y las caídas de cola de entrada? ¿Por qué los inDiscards de SNMP no dan bufferscounts y notinput queue drops, mientras que los outDiscards de SNMP dan output queue drops?](#)

[P. ¿Puedo sondear los buffers en un router?](#)

[P. ¿Cómo sondeo las caídas de límite de cola en un router?](#)

[Información Relacionada](#)

Introducción

Este documento describe las respuestas a las preguntas más frecuentes sobre los contadores SNMP relacionadas con el equipo de Cisco.

Preguntas sobre contadores SNMP



Nota: solo los usuarios registrados de Cisco pueden acceder a la información y las herramientas internas de Cisco.

P. ¿Qué MIB utilizo para los contadores de interfaz?

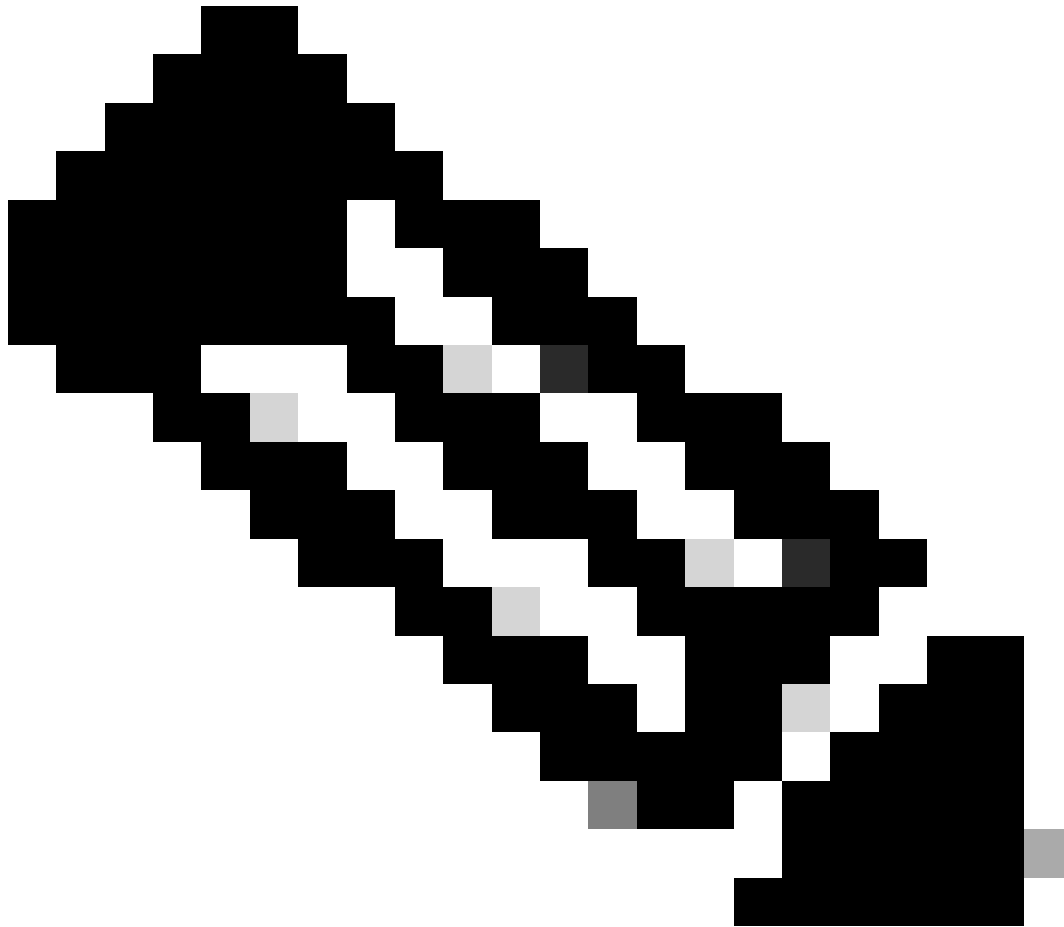
A. La administración de la interfaz sobre SNMP se basa en dos tablas: [ifTable](#) y su extensión, [ifXTable](#) que se describe en RFC1213/RFC2233. Las interfaces pueden tener varias capas, lo que depende de los medios, y cada subcapa está representada por una fila independiente en la tabla. La relación entre la capa superior y las capas inferiores se describe en [ifStackTable](#). La ifTable define contadores de 32 bits para octetos entrantes y salientes (ifInOctets /ifOutOctets), paquetes ([ifInUcastPkts/ifOutUcastPkts](#)) , [ifInNUcastPkts /ifOutNUcastPkts](#), errores y descartes. El ifXTable proporciona contadores similares de 64 bits, también llamados contadores de alta capacidad (HC): [ifHCInOctets/ ifHCOutOctets](#) y [ifHCInUcastPkts / ifHCOutUcastPkts](#).

P. ¿Cuándo se utilizan los contadores de 64 bits?

R. [RFC 2233](#) adoptó contadores de 64 bits ampliados para interfaces de alta capacidad en las

que los contadores de 32 bits no proporcionan suficiente capacidad y se ajustan demasiado rápido.

A medida que aumenta la velocidad de los medios de red, disminuye el tiempo mínimo durante el cual se ajusta un contador de 32 bits. Por ejemplo, una secuencia de 10 Mbps de paquetes adosados de tamaño completo hace que iflnOctets 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.



Nota: Los contadores SNMP se ajustan, pero los contadores de la interfaz de línea de comandos (CLI) no.

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 interfaces que funcionan más rápido que 20 millones de bits por segundo y más lento que 650.000.000 de bits por segundo, debe utilizar contadores de paquetes de 32 bits y contadores de octetos de 64 bits. Para interfaces que funcionan a 650.000.000 bits/segundo o más rápido, se deben utilizar contadores

de octetos y paquetes de 64 bits.

En consecuencia, el software Cisco IOS® no admite contadores de 64 bits para velocidades de interfaz inferiores a 20 Mbps. Esto significa que los contadores de 64 bits no son compatibles con los puertos Ethernet de 10 Mb, sólo los puertos Fast-Ethernet de 100 Mb y otros puertos de alta velocidad admiten contadores de 64 bits.

P. ¿Qué versión de SNMP se requiere para consultar los contadores de 64 bits?

A. SNMPv2C o SNMPv3 se requiere para consultar contadores de 64 bits. SNMPv1 no admite contadores de 64 bits. Tenga en cuenta que ifInOctets = .1.3.6.1.2.1.2.2.1.10 es un contador de 32 bits, mientras que la versión de 64 bits es ifHCInOctets = .1.3.6.1.2.1.31.1.1.1.6.

Por ejemplo:

Catalyst 5000 utiliza HP OpenView snmpget, cuyo valor predeterminado es SNMPv1

```
# snmpget -c public 10.32.5.18 ifName.1
```

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

Consulta con SNMPv1, valor predeterminado para HP OpenView snmpget

```
# snmpget -c public 10.32.5.18 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.
```

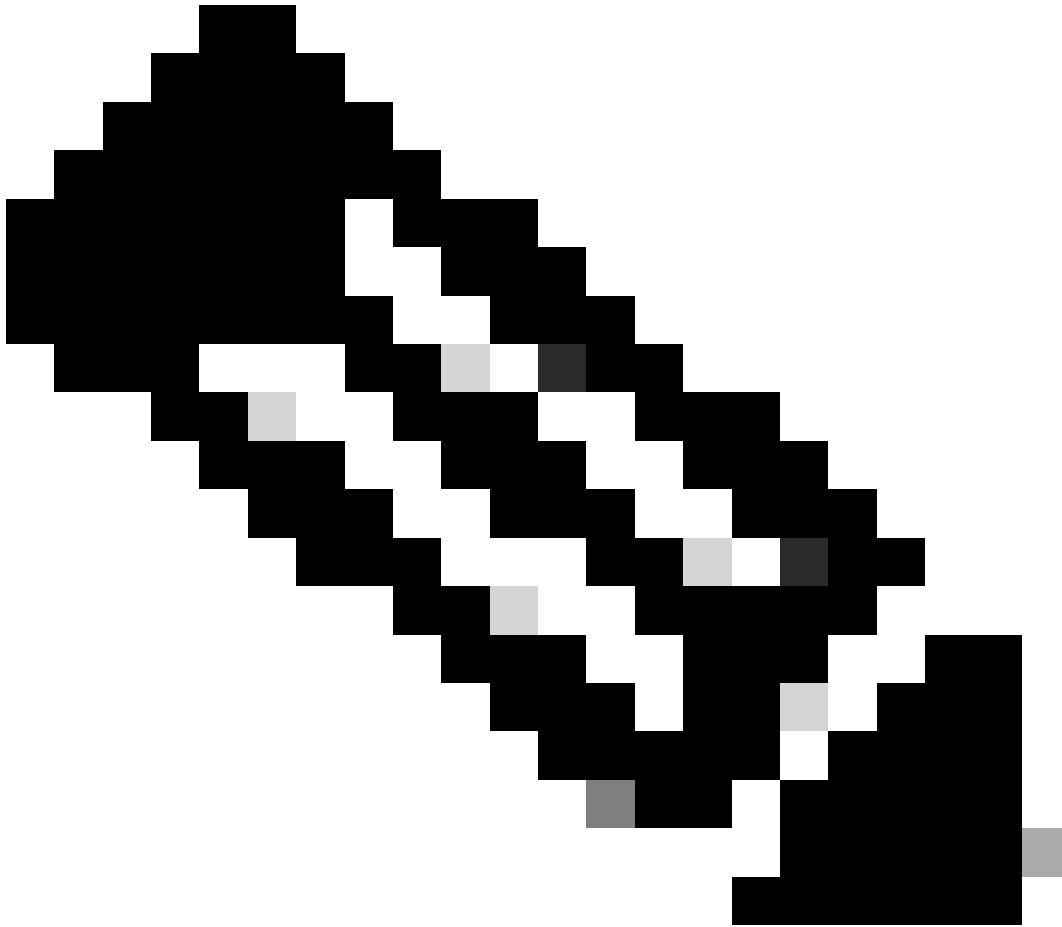
Misma consulta con SNMPv2C

```
# snmpget -v 2c -c public 10.32.5.18 ifHCInOctets.1
```

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

P. ¿Qué dispositivos Cisco implementan contadores SNMP de 64 bits, especialmente para IF-MIB?

R.Estos dispositivos de Cisco implementan contadores SNMP de 64 bits:

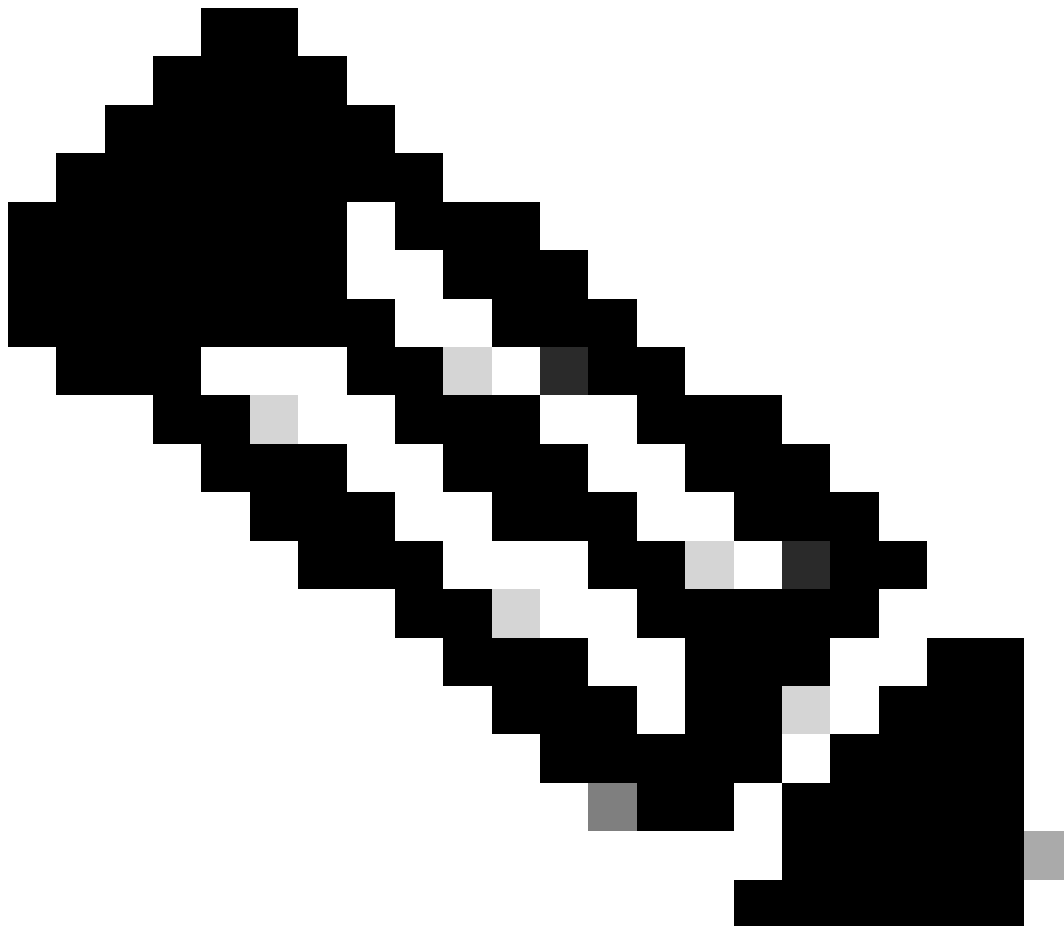


Nota: Debe ser un usuario registrado y haber iniciado sesión para visitar los enlaces de ID de bug de Cisco y ver información detallada de bug.

- Routers Cisco 2500 y 2600: estos routers no admiten contadores de 64 bits.
- Catalyst 2950 y 3550—El soporte comienza en la versión 12.1(11)EA1 del software del IOS de Cisco desde el ID de bug de Cisco [CSCdx67611](#) y el ID de bug de Cisco [CSCdw52807](#) .
- Catalyst 2900XL y 3500XL—El soporte comienza en la versión 12.0(5)WC3 del software del IOS de Cisco desde el ID de bug de Cisco [CSCds45300](#).
- Catalyst serie 5000: desde la versión 3.x del software del IOS de Cisco. En RSM/RSFC, el soporte comienza en la versión 12.1(6)E1 del software del IOS de Cisco desde el ID de bug Cisco [CSCds50549](#) .
- Módulos ATM Catalyst 5000/6000: desde la versión 12.0(14)W05(20) del software del IOS

de Cisco, consulte la identificación de error de Cisco [CSCds07238](#).

- Catalyst 6000 Gigabit Ethernet WAN OSM: desde la versión 12.1.12E del software del IOS de Cisco, consulte la identificación de error de Cisco [CSCdw64849](#).
 - Catalyst serie 6000: todas las versiones de software de Cisco IOS. La compatibilidad con WS-F6K-MSFC y MSM comienza en Cisco IOS Software Release 12.1(8a)E4.
 - Catalyst serie 8500: la compatibilidad comienza a partir de la versión 12.0(5)W5(13) del software Cisco IOS.
 - Interfaces de túnel: a partir de la versión 12.0(16)S del software del IOS de Cisco, consulte la identificación de error de Cisco [CSCdt58029](#).
-



Nota: El software Cisco IOS no admite contadores de 64 bits para velocidades de interfaz inferiores a 20 Mbps. Esto significa que los contadores de 64 bits no son compatibles con los puertos Ethernet de 10 Mb. Sólo los puertos Fast-Ethernet de 100 Mb y otros puertos de alta velocidad admiten contadores de 64 bits.

P. ¿Son los contadores SNMP ifInOctets y ifOutOctets los mismos que los contadores show interfaces In/Out?

A.Sí, pero sólo cuando SNMP está habilitado desde el inicio. Si enciende un dispositivo Cisco y luego activa SNMP, los contadores SNMP comienzan desde 0. No recogen automáticamente sus valores de la salida CLI.

P. ¿Incluyen los contadores ifInOctets e ifOutOctets la sobrecarga de tramas (protocolo punto a punto, control de vínculo de datos de alto nivel)?

R.Sí.

P. En una interfaz de Asynchronous Transfer Mode, ¿los contadores incluyen el encabezado de celda?

A.Los contadores del modo de transferencia asincrónica (ATM) no incluyen sobrecarga ATM (encabezados de celda y relleno AAL5).

P. ¿Por qué los contadores SNMP no devuelven el mismo número que los comandos CLishow?

A.Un objeto SNMP definido como un contador debe cumplir con [RFC1155](#) "3.2.3.3. Contador.

Este tipo de toda la aplicación representa un entero no negativo que aumenta monótonamente hasta que alcanza un valor máximo, cuando se ajusta y comienza a aumentar de nuevo desde cero. Este memorando especifica un valor máximo de $2^{32}-1$ (4294967295 decimales) para los contadores."

No hay métodos para restablecer un contador SNMP a cero sin necesidad de volver a cargar el dispositivo.

El resultado del contador de un show comando CLI se puede restablecer en las interfaces ya que no existen restricciones SNMP.

Los contadores de interfaz originales definidos en MIB-2 son contadores de 32 bits. Para una interfaz de 10 Mbps, un contador de 32 bits teóricamente podría ajustarse en 57 minutos. Es fácil evitar discontinuidades con un período tan largo. Sin embargo, para 100 Mbps, el tiempo mínimo teórico de cierre es de 5,7 minutos. Para interfaces de 1 Gbps, se reduce a 34 segundos. Es cierto que estos tiempos son para la transmisión de paquetes de tamaño completo adosados, un ideal teórico. Aun así, cuanto más alta sea la velocidad de la interfaz, más difícil será evitar perder una envoltura de contador. Como solución a este problema, SNMPv2 SMI definió un nuevo tipo de objeto, counter64, para los contadores de 64 bits. Por lo tanto, hay varios contadores nuevos de 64 bits definidos en la tabla de interfaz de extensión (ifxTable) definida en [RFC 1573](#) (reemplazado posteriormente por [RFC 2233](#)). Proviene de [IF-MIB-V1SMI.my](#).

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)



Nota: solo los usuarios registrados de Cisco pueden acceder a la información y las herramientas internas de Cisco

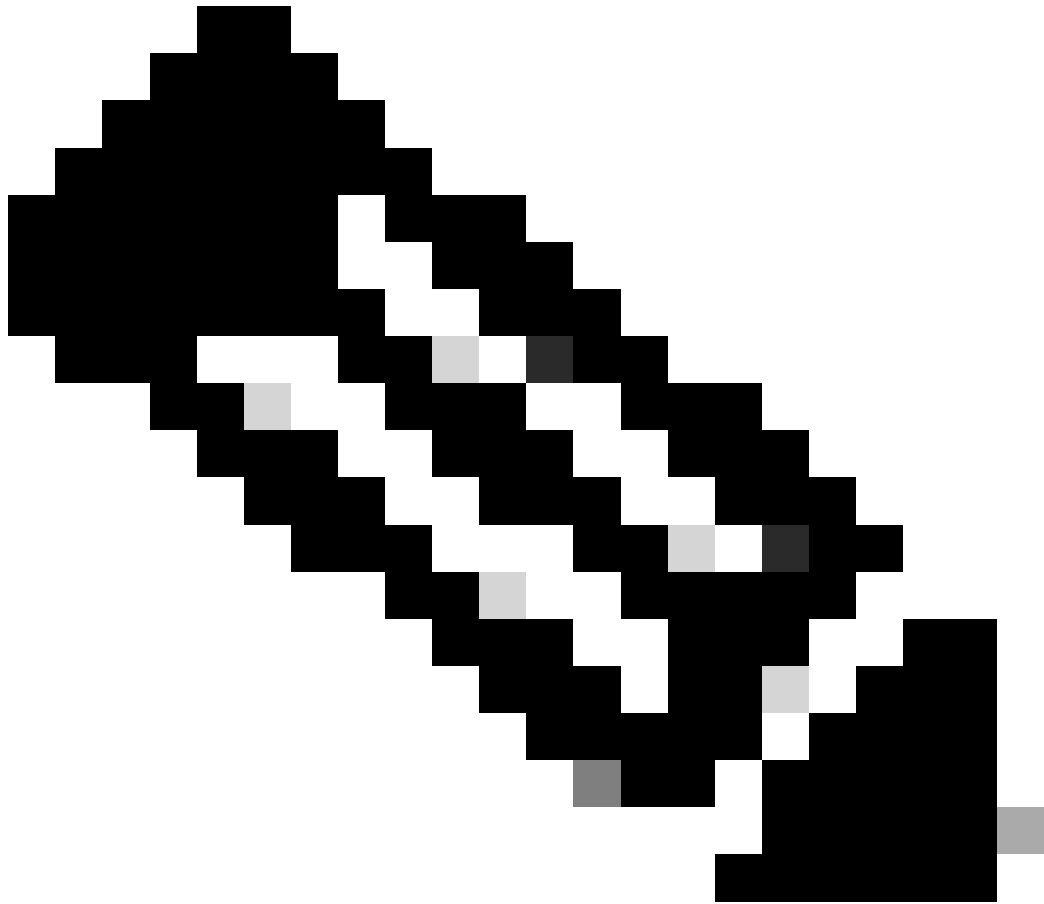
Aunque el soporte básico para los contadores de 64 bits se escribió en Cisco IOS Software Release 11.3, que comienza desde Cisco IOS Software Release 12.0, solo si `HCInOctets` (.1.3.6.1.2.1.31.1.1.1.6) y `ifHCOctets` (1.3.6.1.2.1.31.1.1.1.10) se han implementado para las subinterfaces LEC LANE ATM solamente. Para los switches de grupo de trabajo Catalyst, se ha implementado la compatibilidad con contadores de 64 bits en la versión 3.1.



Nota: Debe utilizar el protocolo SNMPv2c o SNMPv3 para recuperar cualquier objeto del contador 64.

Contadores SNMP y preguntas equivalentes al comando show

P. ¿Qué hacen los routers de Cisco para estas variables MIB SNMP: ifInOctets, ifInUcastPkts, ifInNUcastPkts, ifInDiscards, ifInErrors, ifInUnknownProtos, ifOutOctets, ifOutUcastPkts, ifOutNUcastPkts, ifOutDiscards, ifOutErrors y ifOutQLen?



Nota: solo los usuarios registrados de Cisco pueden acceder a la información y las herramientas internas de Cisco

R.Consulte esta tabla para obtener más información. Proviene de [RFC1213-MIB](#).

ifInNUcastPkts (.1.3.6.1.2.1.2.2.1.12)	Estos son recuentos de paquetes de difusión y multidifusión entrantes.
ifInDiscards (.1.3.6.1.2.1.2.2.1.13)	Éstos se cuentan como búferes no reflejados en el comando show interfaces.
ifInErrors (.1.3.6.1.2.1.2.2.1.14)	Estos son conteos 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)	Estos errores se cuentan como errores no clasificados.
ifOutOctets (.1.3.6.1.2.1.2.2.1.16)	Estos son conteos del número de bytes de salida por la interfaz como se muestra en el comando show interfaces.
ifOutUcastPkts (.1.3.6.1.2.1.2.2.1.17)	Estos son conteos de paquetes de broadcast y multicast salientes.
ifOutDiscards (.1.3.6.1.2.1.2.2.1.19)	Estos se cuentan como caídas de salida como se muestra en el comando show interfaces.
ifOutErrors (.1.3.6.1.2.1.2.2.1.20)	Estos se cuentan como errores de salida como se muestra en el comando show interfaces.
ifOutQLen (.1.3.6.1.2.1.2.2.1.21)	Este es el número de paquetes permitidos en la cola de salida como se muestra en el comando show interfaces.

Las variables previamente listadas que no dicen aparecer en **show interfaces** no están disponibles en ningún otro lugar que SNMP.

Examples

Este ejemplo utiliza un 3640 que se ejecuta con la versión 12.2(2)T1 del software del IOS de Cisco. La cadena de comunidad de sólo lectura (RO) utilizada es pública y la cadena de comunidad de lectura y escritura (RW) utilizada es privada. Consulte [Configuración de Cadenas de Comunidad SNMP](#) para obtener más información sobre cómo configurar las Cadenas de Comunidad SNMP en los dispositivos.

Este resultado es típico del comando **show ip interface brief** ejecutado en el modo de habilitación:

```
<#root>
```

```
3600#
```

```
show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Pro1
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	192.168.99.99	YES	NVRAM	down	dow
Loopback0	10.1.10.1	YES	NVRAM	up	up
Loopback1	unassigned	YES	NVRAM	up	up
Loopback101	10.3.3.3	YES	NVRAM	administratively down	dow
Loopback200	10.4.4.14	YES	NVRAM	administratively down	dow
Loopback201	10.4.4.18	YES	NVRAM	administratively down	dow

Este resultado es el objeto 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 información sobre la interfaz. Esto proporciona el nombre y la descripción de la interfaz según se obtuvo, que utiliza el resultado del comando CLI anterior.

ifName (.1.3.6.1.2.1.31.1.1.1.1) también se puede utilizar, pero **ifDescr** proporciona la descripción de la interfaz junto con el nombre, donde **ifName** sólo proporciona el nombre de la interfaz.

```
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-aa15 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
```

1.

```
ifInDiscards (.1.3.6.1.2.1.2.2.1.13):
```

```
<#root>
```

```
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
```

El valor de ifInDiscards es cero para todas las interfaces de este router. Si compara esto con el resultado CLI del comando **show interfaces fastEthernet 0/0**, esto confirma el resultado:

<#root>

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):

<#root>

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
```

Este resultado muestra que sólo hay un error de entrada para la interfaz **interfaces.ifTable.ifEntry.ifInErrors.7 = Counter32: 1**.

Para determinar qué interfaz es ésta, compárela con la salida de `ifDescr`, que muestra que ésta proviene de **interfaces.ifTable.ifEntry.ifDescr.7 = Serial0/1**. Ahora ejecute el comando **show interfaces serial 0/1** en el modo de habilitación para verificar el resultado anterior:

```
<#root>
```

```
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):

<#root>

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 compara el resultado anterior con el resultado de **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 **interfaces.ifTable.ifEntry.ifDescr.21 = capa ATM1/0.109-aal5**

Este es el resultado del comando CLI **show interfaces** para cada una de las interfaces anteriores ejecutadas en el modo de habilitación:

```
<#root>
```

```
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

La salida de ifOutOctets no coincide con la salida de CLI para el comando **show interfaces FastEthernet 0/0**, pero es similar. Esto se debe a que puede haber un retraso cuando se sondean las interfaces y cuando se ejecuta el comando CLI.

```
<#root>
```

```
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):
```

```
<#root>
```

```
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
```

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

```
<#root>
```

```
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):

<#root>

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
```

El valor de ifOutErrores cero para todas las interfaces. Con el comando **show interfaces fastEthernet 0/0** como ejemplo, este comando produce este resultado:

```
<#root>
```

```
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):

<#root>

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 valor de ifOutQLenes cero para todas las interfaces. Con el comando **show interfaces fastEthernet 0/0** como ejemplo:

```
<#root>
```

```
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
```

P. ¿Cuál es la relación entre el `show interfaces statements no buffer and input queue drops`? ¿Por qué los `inDiscards` de SNMP no dan `buffercounts` y `no input queue drops`, mientras que los `outDiscards` de SNMP hacen `give output queue drops`?

A. Los métodos `locIfInputQueueDrops/ifInDiscards` funcionan de forma diferente que `locIfOutputQueueDrops/ifOutDiscards`. El `ifInDiscards` cuenta el número de paquetes que se descartan por falta de un recurso del sistema como un buffer. Este suele ser un subconjunto de `locIfInputQueueDrops`. A menudo aparece `locIfInputQueueDrops = ifInDiscards`. Sin embargo, `locIfInputQueueDrops` también cuenta el número de paquetes descartados porque alcanzan el límite de la cola de entrada. Por lo general, verá `locIfInputQueueDrops > ifInDiscards`.

Summary

$\text{locIfInputQueueDrops} = \text{Caídas de límite de cola} + \text{Sin descartes de búfer}$
 $\text{ifInDiscards} = \text{Sin descartes de búfer (y es un subconjunto de locIfInputQueueDrops)}$

Los valores de `locIfOutputQueueDrops` y `ifOutDiscards` son siempre iguales cuando cuentan los mismos eventos. Esos eventos alcanzan el límite de la cola de salida y no tienen un búfer de transmisión de datos de hardware cuando un paquete se conmuta rápidamente de una interfaz a otra. Los OID de los objetos MIB anteriores son los siguientes:

FromOLD-CISCO-INTERFACES-MIB	Desde MIB de RFC1213
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

P. ¿Puedo sondear no buffers en un router?

R.Sí. Puede sondear ifInDiscards para sondear no buffers.

P. ¿Cómo sondeo las caídas de límite de cola en un router?

R.Con el uso de SNMP, no hay manera de que el comando **show interfaces** rompa los elementos individuales que entran en las caídas de salida.

Considere esta nueva información sobre lo que entra en el contador de caídas de salida:

Caídas de entrada = caídas de límite de cola + caídas de aceleración + caídas completas de cola SMT + caídas RSRB + sin caídas de búfer

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

Información Relacionada

- [Soporte técnico y descargas de Cisco](#)

Acerca de esta traducción

Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).