

Considere os contadores SNMP: perguntas frequentes

Contents

[Introdução](#)

[Perguntas sobre contadores SNMP](#)

- [P. Qual MIB devo usar para contadores de interface?](#)
- [P. Quando os contadores de 64 bits devem ser usados?](#)
- [P. Qual versão do SNMP é necessária para consultar contadores de 64 bits?](#)
- [P. Quais dispositivos Cisco implementam contadores SNMP de 64 bits, especialmente para o IF-MIB?](#)
- [P. Os contadores SNMP ifInOctets e ifOutOctets são os mesmos que os contadores show interfaces In/Out?](#)
- [P. Os contadores ifInOctets e ifOutOctets incluem sobrecarga de enquadramento \(Protocolo Ponto-a-Ponto, Controle de Enlace de Dados de Alto Nível\)?](#)
- [P. Em uma interface de Modo de transferência assíncrono, os contadores incluem o cabeçalho da célula?](#)
- [P. Por que os contadores SNMP não retornam o mesmo número que os comandos CLIshow?](#)

[Contadores SNMP e perguntas equivalentes do comando show](#)

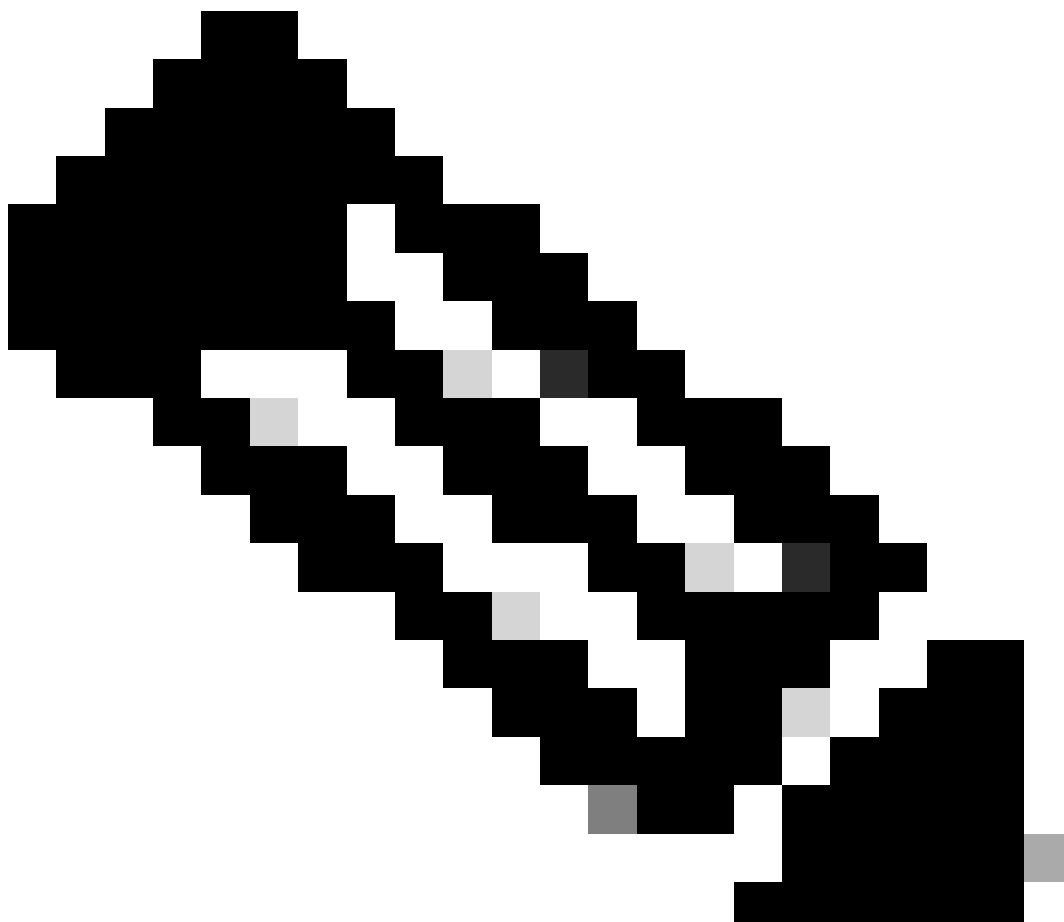
- [P. O que os roteadores Cisco fazem para essas variáveis MIB SNMP: ifInOctets, ifInUcastPkts, ifInNUcastPkts, ifInDiscards, ifInErrors, ifInUnknownProtos, ifOutOctets, ifOutUcastPkts, ifOutNUcastPkts, ifOutDiscards, ifOutErrors e ifOutQLen?](#)
 - [Examples](#)
- [P. Qual é a relação entre os show interfaces statements no buffers and input queue drops? Por que inDiscards de SNMP sem contagens de buffers e quedas de fila de não-entrada, enquanto outDiscards de SNMP fornecem quedas de fila de saída?](#)
- [P. Posso fazer poll sem buffers em um roteador?](#)
- [P. Como sondo as quedas de limite de fila em um roteador?](#)

[Informações Relacionadas](#)

Introdução

Este documento descreve as respostas às perguntas mais frequentes sobre os contadores SNMP relacionadas ao equipamento Cisco.

Perguntas sobre contadores SNMP



Observação: somente usuários registrados da Cisco podem acessar ferramentas e informações internas da Cisco.

P. Qual MIB devo usar para contadores de interface?

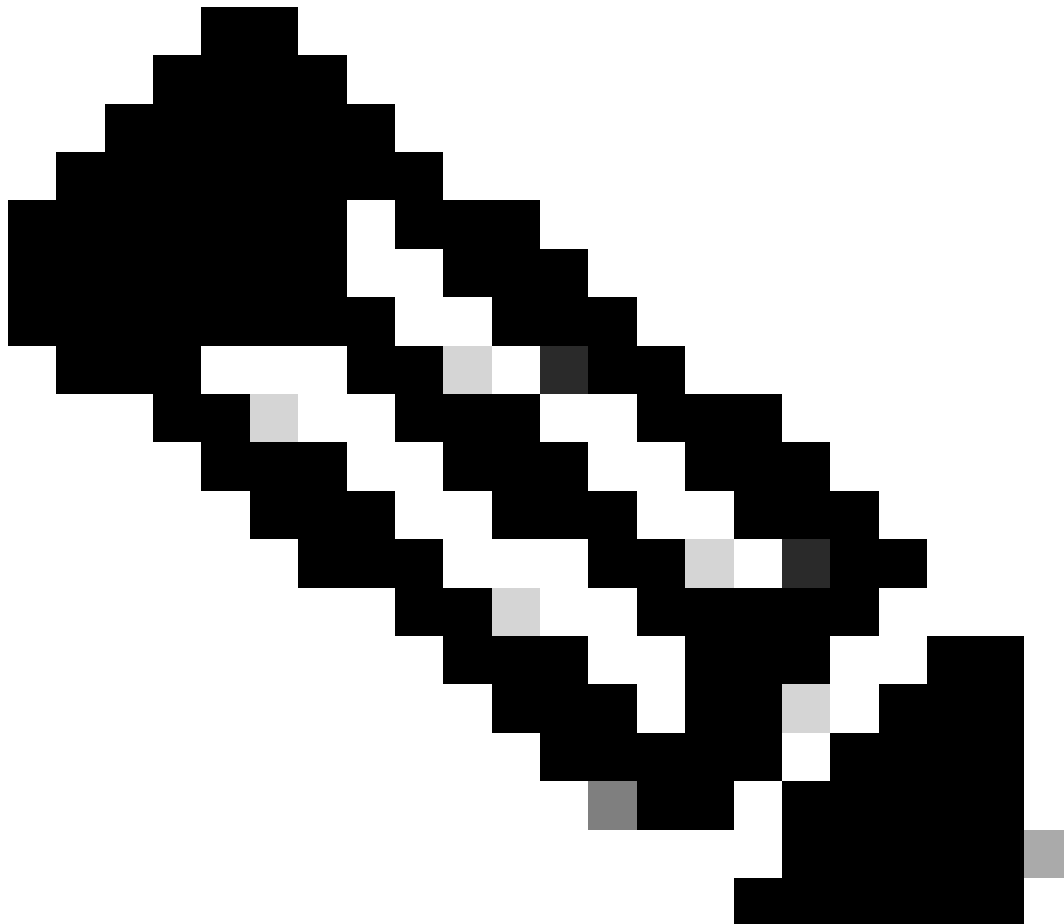
R. O gerenciamento de interface sobre SNMP é baseado em duas tabelas: [ifTable](#) e sua extensão, [ifXTable](#), descrita em RFC1213/RFC2233. As interfaces podem ter várias camadas, que dependem do meio físico, e cada subcamada é representada por uma linha separada na tabela. A relação entre as camadas superior e inferior é descrita na [ifStackTable](#). A ifTable define contadores de 32 bits para octetos de entrada e de saída (ifInOctets /ifOutOctets), pacotes ([ifInUcastPkts/ifOutUcastPkts](#)), ifInNUcastPkts /ifOutNUcastPkts, erros e descartes. O ifXTable fornece contadores de 64 bits semelhantes, também chamados de contadores de alta capacidade (HC): [ifHCInOctets/ ifHCOctets](#) e [ifHCInUcastPkts / ifHCOUcastPkts](#).

P. Quando os contadores de 64 bits devem ser usados?

R. O [RFC 2233](#) adotou contadores expandidos de 64 bits para interfaces de alta capacidade nas

quais os contadores de 32 bits não fornecem capacidade suficiente e são encapsulados muito rápido.

À medida que a velocidade dos meios de rede aumenta, o tempo mínimo em que um contador de 32 bits é encapsulado diminui. Por exemplo, um fluxo de 10 Mbps de pacotes back-to-back de tamanho completo faz com que o iflnOctets distribua em apenas 57 minutos. Em 100 Mbps, o tempo de empacotamento mínimo é de 5,7 e, em 1 Gbps, o mínimo é de 34 segundos.



Observação: os contadores SNMP são encapsulados, os contadores da interface de linha de comando (CLI) não.

Para interfaces que operam em até 20.000.000 (20 milhões) de bits por segundo, use bytes de 32 bits e contadores de pacote. Para interfaces que operam mais rápido que 20 milhões de bits por segundo e mais lento que 650.000.000 bits por segundo, você deve usar contadores de pacotes de 32 bits e contadores de octetos de 64 bits. Para interfaces que operam a 650.000.000 bits/segundo ou mais rápido, devem ser usados contadores de pacotes e octetos de 64 bits.

Da mesma forma, o software Cisco IOS® não suporta contadores de 64 bits para velocidades de interface inferiores a 20 Mbps. Isso significa que os contadores de 64 bits não são suportados em portas Ethernet de 10 Mb, somente portas Fast-Ethernet de 100 Mb e outras portas de alta velocidade suportam contadores de 64 bits.

P. Qual versão do SNMP é necessária para consultar contadores de 64 bits?

R.O SNMPv2C ou SNMPv3 é necessário para consultar contadores de 64 bits. O SNMPv1 não suporta contadores de 64 bits. Esteja ciente de que ifInOctets = .1.3.6.1.2.1.2.1.10 é um contador de 32 bits, enquanto a versão de 64 bits é ifHCInOctets = .1.3.6.1.2.1.31.1.1.1.6.

Por exemplo:

O Catalyst 5000 usa o snmpget do HP OpenView, que assume como padrão o SNMPv1

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

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

Consulta com SNMPv1, padrão para snmpget do HP OpenView

```
# 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.
```

Mesma consulta com SNMPv2C

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

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

P. Quais dispositivos Cisco implementam contadores SNMP de 64 bits, especialmente para o IF-MIB?

R.Esses dispositivos Cisco implementam contadores SNMP de 64 bits:

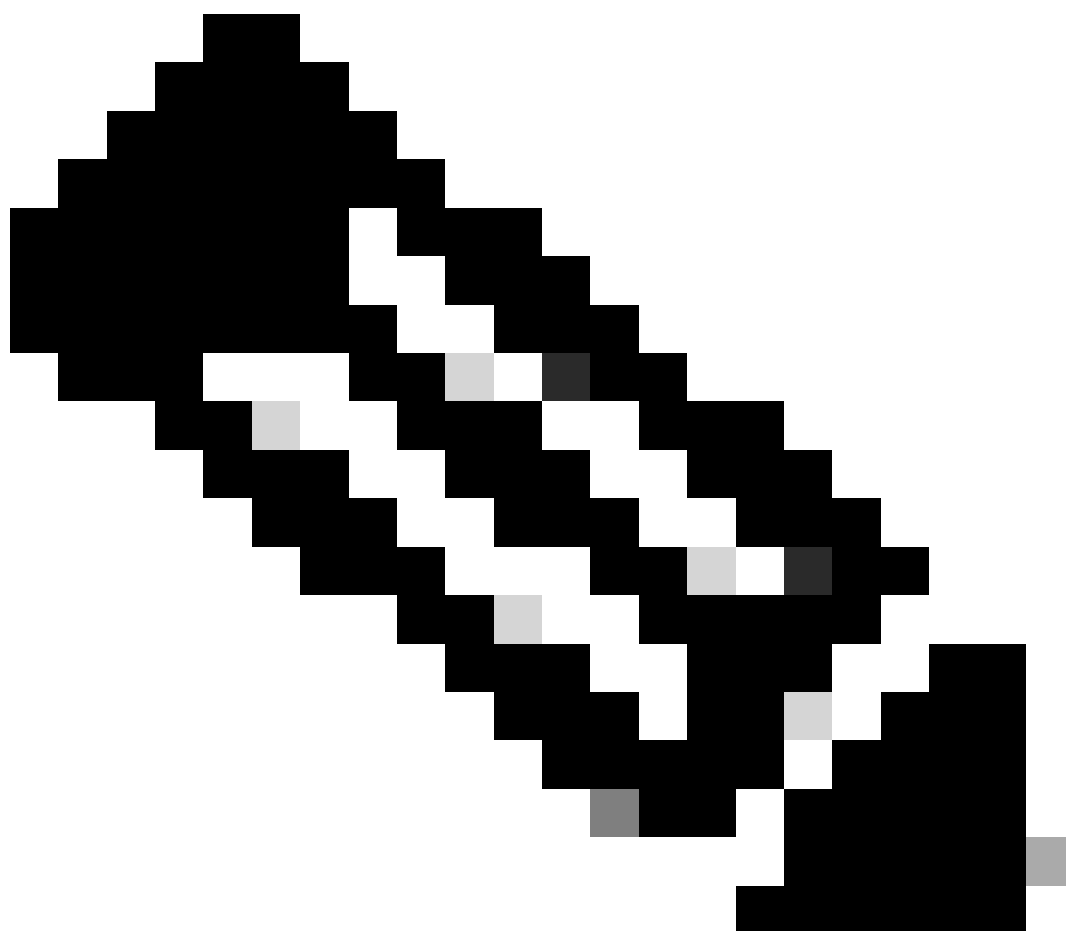


Observação: você deve ser um usuário registrado e deve estar conectado para visitar os links de ID de bug da Cisco e ver informações detalhadas sobre o bug.

-
- Roteadores Cisco 2500, 2600—Esses roteadores não suportam contadores de 64 bits.
 - Catalyst 2950 e 3550—O suporte começa no Cisco IOS Software Release 12.1(11)EA1 desde o bug da Cisco ID [CSCdx67611](#) e o bug da Cisco ID [CSCdw52807](#).
 - Catalyst 2900XL e 3500XL—O suporte começa no Cisco IOS Software Release 12.0(5)WC3 desde o bug da Cisco ID [CSCds45300](#).
 - Série Catalyst 5000—Desde o Cisco IOS Software Release 3.x. No RSM/RSFC, o suporte começa no Cisco IOS Software Release 12.1(6)E1 desde o bug da Cisco ID [CSCds50549](#).
 - Módulos ATM Catalyst 5000/6000—Desde o Cisco IOS Software Release 12.0(14)W05(20), consulte o bug da Cisco ID [CSCds07238](#).
 - Catalyst 6000 Gigabit Ethernet WAN OSM—Desde o Cisco IOS Software Release 12.1.12E,

consulte o bug da Cisco ID [CSCdw64849](#) .

- Série Catalyst 6000—Todas as versões do software Cisco IOS. O suporte a WS-F6K-MSFC e MSM começa no Cisco IOS Software Release 12.1(8a)E4.
 - Série Catalyst 8500—O suporte começa a partir do Cisco IOS Software Release 12.0(5)W5(13).
 - Interfaces de túnel—A partir do Cisco IOS Software Release 12.0(16)S, consulte o bug da Cisco ID [CSCdt58029](#).
-



Observação: o Cisco IOS Software não suporta contadores de 64 bits para velocidades de interface inferiores a 20 Mbps. Isso significa que os contadores de 64 bits não são suportados em portas Ethernet de 10 Mb. Somente portas Fast-Ethernet de 100 Mb e outras portas de alta velocidade suportam contadores de 64 bits.

P. Os contadores SNMP ifInOctets e ifOutOctets são os mesmos que os

contadores show interfaces In/Out?

R.Sim, mas somente quando o SNMP está ativado no momento da inicialização. Se você ligar um dispositivo Cisco, habilite o SNMP, os contadores SNMP começam em 0. Eles não captam automaticamente seus valores da saída CLI.

P. Os contadores ifInOctets e ifOutOctets incluem sobrecarga de enquadramento (Protocolo Ponto-a-Ponto, Controle de Enlace de Dados de Alto Nível)?

R.Sim.

P. Em uma interface de Modo de transferência assíncrono, os contadores incluem o cabeçalho da célula?

R.Os contadores ATM não incluem carga adicional de ATM (cabeçalhos de célula e preenchimento AAL5).

P. Por que os contadores SNMP não retornam o mesmo número que CLishowcommands?

R.Um objeto SNMP definido como um contador deve obedecer à [RFC1155](#) "3.2.3.3. Contador.

Este tipo de aplicação representa um inteiro não negativo que aumenta monotonicamente até atingir um valor máximo, quando ele se envolve e começa a aumentar novamente a partir de zero. Este memorando especifica um valor máximo de $2^{32}-1$ (4294967295 decimal) para contadores."

Não há métodos para zerar um contador SNMP sem a necessidade de recarregar o dispositivo.

A saída do contador de um show comando CLI pode ser redefinida nas interfaces, já que as restrições SNMP não estão presentes.

Os contadores de interface originais definidos em MIB-2 são contadores de 32 bits. Para uma interface de 10 Mbps, um contador de 32 bits poderia teoricamente ser empacotado em 57 minutos. É fácil evitar discontinuidades com um período tão longo. Mas para 100 Mbps, o tempo mínimo teórico de finalização é de 5,7 minutos. Para interfaces de 1 Gbps, cai para 34 segundos. Concedidos esses tempos são para a transmissão de pacotes back-to-back de tamanho completo, um ideal teórico. Mesmo assim, quanto maior a velocidade da interface, mais difícil fica evitar a perda de um contador. Como solução para esse problema, o SNMPv2 SMI definiu um novo tipo de objeto, counter64, para contadores de 64 bits. Portanto, há vários novos contadores de 64 bits definidos na tabela de interface de extensão (ifxTable) definida na [RFC 1573](#) (posteriormente substituída pela [RFC 2233](#)). Eles são do [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)



Observação: somente usuários registrados da Cisco podem acessar informações e ferramentas internas da Cisco

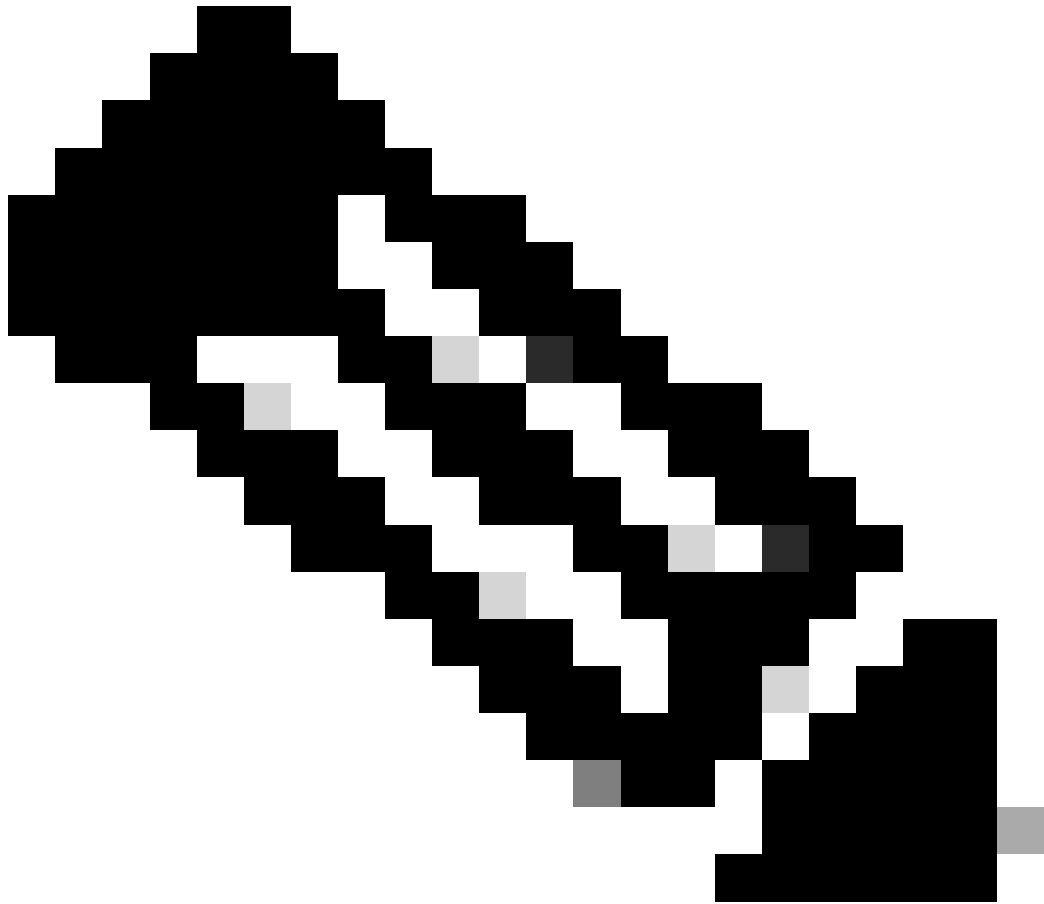
Embora o suporte básico para contadores de 64 bits tenha sido escrito no Cisco IOS Software Release 11.3, que começa no Cisco IOS Software Release 12.0, somente seHCInOctets (.1.3.6.1.2.1.31.1.1.6) e ifHCOctets (1.3.6.1.2.1.31.1.1.10) tiverem sido implementados somente para subinterfaces LEC LANE ATM. Para os switches de grupo de trabalho Catalyst, o suporte ao contador de 64 bits foi implementado na versão 3.1.



Observação: você deve usar o protocolo SNMPv2c ou SNMPv3 para recuperar qualquer objeto do contador 64.

Contadores SNMP e perguntas equivalentes do comando show

P. O que os roteadores Cisco fazem para essas variáveis MIB SNMP: ifInOctets, ifInUcastPkts, ifInNUcastPkts, ifInDiscards, ifInErrors, ifInUnknownProtos, ifOutOctets, ifOutUcastPkts, ifOutNUcastPkts, ifOutDiscards, ifOutErrors e ifOutQLen?



Observação: somente usuários registrados da Cisco podem acessar informações e ferramentas internas da Cisco

R.Consulte esta tabela para obter detalhes. Eles são do [RFC1213-MIB](#).

ifInNUcastPkts (.1.3.6.1.2.1.2.2.1.12)	Essas são contagens de pacotes de broadcast e multicast de entrada.
ifInDiscards (.1.3.6.1.2.1.2.2.1.13)	Eles são contados como nenhum buffer, conforme refletido no comando show interfaces.
erros de entrada (.1.3.6.1.2.1.2.2.1.14)	Essas são contagens de todos os erros de entrada, conforme refletido no comando show interfaces.
ifInUnknownProtos (.1.3.6.1.2.1.2.2.1.15)	São contados como erros não classificados.
ifOutOctets (.1.3.6.1.2.1.2.2.1.16)	Essas são contagens do número de bytes de saída pela interface conforme mostrado no comando show interfaces.
ifOutUcastPkts (.1.3.6.1.2.1.2.2.1.17)	Essas são contagens de pacotes de transmissão e multicast de saída.
ifOutDiscards (.1.3.6.1.2.1.2.2.1.19)	Eles são contados como descartes de saída conforme mostrado no comando show interfaces.
ifOutErrors (.1.3.6.1.2.1.2.2.1.20)	Eles são contados como erros de saída conforme mostrado no comando show interfaces.
ifOutQLen (.1.3.6.1.2.1.2.2.1.21)	Esse é o número de pacotes que podem estar na fila de saída conforme mostrado no comando show interfaces.

As variáveis listadas anteriormente que não dizem que aparecem em **show** interfaces não estão disponíveis em nenhum outro lugar além do SNMP.

Examples

Este exemplo usa um 3640 que é executado com o Cisco IOS Software Release 12.2(2)T1. A sequência de caracteres de comunidade somente leitura (RO) usada é pública e a sequência de caracteres de comunidade de leitura-gravação (RW) usada é privada. Consulte [Configurar séries de comunidade SNMP](#) para obter mais informações sobre como configurar séries de comunidade SNMP em dispositivos.

Esta saída é típica do comando **show ip interface brief** executado no modo enable:

```
<#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

Esta saída é o MIB Object **ifDescr** (.1.3.6.1.2.1.2.2.1.2) do roteador anterior, que é uma sequência de texto que contém informações sobre a interface. Isso fornece o nome e a descrição da interface obtidos, que usa a saída do comando CLI anterior. **ifName** (.1.3.6.1.2.1.31.1.1.1) também pode ser usado, mas **ifDescr** fornece a descrição da interface juntamente com o nome, onde **ifName** fornece apenas o nome da interface.

```
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
```

O ifInDiscards é zero para todas as interfaces deste roteador. Se você comparar isso com o resultado CLI do comando **show interfaces fastEthernet 0/0**, isso confirma o 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
```

Esta saída mostra que há apenas um erro de entrada para a interface **interfaces.ifTable.ifEntry.ifInErrors.7 = Counter32: 1**.

Para determinar qual interface é essa, compare-a com a saída de ifDescrabove, que mostra que ela é de **interfaces.ifTable.ifEntry.ifDescr.7 = Serial0/1**. Agora execute o comando **show interfaces serial 0/1** no modo enable para verificar o 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
```

Se você comparar o resultado anterior com a saída do **ifDescr**, isso indica:

•

interfaces.ifTable.ifEntry.ifOutOctets.1 = Counter32: 98 corresponde a **interfaces.ifTable.ifEntry.ifDescr.1 = ATM1/0**

•

interfaces.ifTable.ifEntry.ifOutOctets.3 = Counter32: 765470674 corresponde a **interfaces.ifTable.ifEntry.ifDescr.3 = FastEthernet0/0**

•

interfaces.ifTable.ifEntry.ifOutOctets.21 = Counter32: 98 corresponde a **interfaces.ifTable.ifEntry.ifDescr.21 = camada ATM1/0.109-aal5**

Esta é a saída do comando CLI **show interfaces** para cada uma das interfaces anteriores executadas no modo enable:

```
<#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
```

A saída do ifOutOctets não corresponde à saída CLI do comando **show interfaces FastEthernet 0/0**, mas é semelhante. Isso ocorre porque pode haver um atraso quando as interfaces são sondadas e quando o comando CLI é executado.

```
<#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 (.1.3.6.1.2.1.2.1.19):
```

```
<#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
```

O **ifOutDiscards** é zero para todas as interfaces. Com o comando **show interfaces fastEthernet 0/0** como exemplo, este comando produz 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.

erros de saída (.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
```

O ifOutErrors é zero para todas as interfaces. Com o comando **show interfaces fastEthernet 0/0** como exemplo, este comando produz 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 (.1.3.6.1.2.1.2.1.21):

<#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
```

O ifOutQLené zero para todas as interfaces. Com o comando **show interfaces fastEthernet 0/0** como exemplo:

```
<#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. Qual é a relação entre `show interfaces statements no buffers and input queue drops`? Por que o `inDiscards` de SNMP dado o `bufferCounts` e quedas de fila de entrada, enquanto o `outDiscards` de SNMP do `giveoutput queue drops`?

R. O `locIfInputQueueDrops/ifInDiscards` funciona de forma diferente do `locIfOutputQueueDrops/ifOutDiscards`. O `ifInDiscards` conta o número de pacotes que são descartados por falta de um recurso do sistema, como um buffer. Geralmente, é um subconjunto de `locIfInputQueueDrops`. Geralmente, você vê `locIfInputQueueDrops = ifInDiscards`. Mas `locIfInputQueueDrops` também conta o número de pacotes descartados porque atingem o limite da fila de entrada. Geralmente, você vê `locIfInputQueueDrops > ifInDiscards`.

Summary

$\text{locIfInputQueueDrops} = \text{Quedas de limite de fila} + \text{Sem quedas de buffer}$
 $\text{ifInDiscards} = \text{Sem quedas de buffer}$ (e é um subconjunto de `locIfInputQueueDrops`)

Os `locIfOutputQueueDrops` e `ifOutDiscards` são sempre iguais quando contam os mesmos eventos. Esses eventos atingem o limite da fila de saída e não têm um buffer tx de hardware quando um pacote é comutado rapidamente de uma interface para outra. Os OIDs dos Objetos MIB anteriores são:

FromOLD-CISCO-INTERFACES-MIB	DeRFC1213-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

P. Posso fazer poll sem buffers em um roteador?

R. Sim. Você pode pesquisar ifInDiscards para não pesquisar buffers.

P. Como faço o poll de quedas de limite de fila em um roteador?

R. Com o uso do SNMP, não há como o comando **show interfaces** separar os elementos individuais que entram nos descartes de saída.

Considere estas novas informações sobre o que entra no contador de quedas de saída:

Descartes de entrada = Descartes de limite de fila + Descartes de limitação + Descartes completos de fila SMT + Descartes RSRB + Sem descartes de buffer

Além disso, os contadores SNMP nunca são limpos, mesmo que as interfaces sejam limpas.

Informações Relacionadas

- [Suporte técnico e downloads da Cisco](#)

Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.