

Contatori SNMP: Domande frequenti

Sommario

[Introduzione](#)

[Domande sui contatori SNMP](#)

[Contatori SNMP e show Command Equivalent Questions](#)

[Informazioni correlate](#)

Introduzione

Questo documento contiene le risposte alle domande frequenti e guida gli utenti a trovare risorse utili sui problemi relativi a SNMP e SNMP in relazione alle apparecchiature Cisco.

Domande sui contatori SNMP

D. Quale MIB utilizzare per i contatori di interfaccia?

R. La gestione dell'interfaccia su SNMP si basa su due tabelle: [ifTable](#) (solo clienti [registrati](#)) e la relativa estensione, [ifXTable](#) (solo clienti [registrati](#)) descritti nella RFC1213/RFC2233. Le interfacce possono avere più livelli, che dipendono dal supporto utilizzato, e ogni sottolivello è rappresentato da una riga separata nella tabella. La relazione tra il livello superiore e quello inferiore è descritta in [ifStackTable](#) (solo utenti [registrati](#)). IfTable definisce i contatori a 32 bit per ottetti in entrata e in uscita (ifInOctets (solo utenti registrati) /ifOutOctets), pacchetti ([ifInUcastPkts](#) (solo utenti registrati) /ifOutUcastPkts (solo clienti [registrati](#)), [ifInNUcastPkts](#) (solo clienti [registrati](#)) /ifOutNUcastPkts (solo clienti [registrati](#)), [errori e scarti](#)). IfXTable fornisce contatori simili a 64 bit, detti anche contatori ad alta capacità (HC): [ifHCInOctets](#) (solo clienti [registrati](#)) /ifHCOctets (solo clienti registrati) e [ifHCInUcastPkts](#) (solo clienti [registrati](#)) /ifHCOUcastPkts (solo clienti registrati).

D. Quando utilizzare i contatori a 64 bit?

R. [RFC 2233](#) ha adottato contatori espansi a 64 bit per interfacce ad alta capacità in cui i contatori a 32 bit non forniscono una capacità sufficiente e vanno a capo troppo velocemente.

Con l'aumento della velocità dei supporti di rete, diminuisce il tempo minimo di avvolgimento di un contatore a 32 bit. Ad esempio, un flusso di 10 Mbps di pacchetti back-to-back di dimensioni complete fa sì che IfInOctets esegua il wrapping in poco più di 57 minuti. A 100 Mbps, il tempo di avvolgimento minimo è di 5,7 minuti e a 1 Gbps, il tempo di avvolgimento minimo è di 34 secondi.

Nota: i contatori SNMP vanno a capo, a differenza dei contatori dell'interfaccia della riga di comando (CLI).

Per interfacce che funzionano a 20.000.000 (20 milioni) bit al secondo o meno, è necessario utilizzare contatori di byte e pacchetti a 32 bit. Per interfacce che funzionano a velocità superiori a

20 milioni di bit al secondo e a velocità inferiori a 650.000.000 bit al secondo, è necessario utilizzare contatori di pacchetti a 32 bit e contatori di ottetti a 64 bit. Per le interfacce che funzionano a 650.000.000 di bit/secondo o più velocemente, è necessario utilizzare contatori di pacchetti e ottetti a 64 bit.

Di conseguenza, il software Cisco IOS® non supporta i contatori a 64 bit per velocità di interfaccia inferiori a 20 Mbps. Ciò significa che i contatori a 64 bit non sono supportati sulle porte Ethernet a 10 MB, solo le porte Fast-Ethernet a 100 MB e altre porte ad alta velocità supportano i contatori a 64 bit.

D. Quale versione di SNMP è necessaria per eseguire query sui contatori a 64 bit?

R. SNMPv2C o SNMPv3 è necessario per eseguire query sui contatori a 64 bit. SNMPv1 non supporta i contatori a 64 bit. Tenere presente che ifInOctets = .1.3.6.1.2.1.2.2.1.10 è un contatore a 32 bit mentre la versione a 64 bit è ifHCInOctets = .1.3.6.1.2.1.31.1.1.1.6.

Ad esempio:

Catalyst 5000 utilizza lo snmpget HP OpenView, che per impostazione predefinita è SNMPv1

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

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

Query con SNMPv1, impostazione predefinita per snmpget OpenView HP

```
# 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.

Stessa query con SNMPv2C

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

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

D. Quali dispositivi Cisco implementano i contatori SNMP a 64 bit, in particolare per l'IF-MIB?

R. Questi dispositivi Cisco implementano contatori SNMP a 64 bit:

Nota: per visitare i collegamenti agli ID dei bug Cisco e visualizzare le informazioni dettagliate sui bug, è necessario essere un utente registrato e aver eseguito l'accesso.

- Cisco 2500 e 2600 router: questi router non supportano i contatori a 64 bit.
- Catalyst 2950 e 3550: il supporto inizia con il software Cisco IOS versione 12.1(11)EA1,

dall'ID bug Cisco CSCdx67611 (solo utenti [registrati](#)) e dall'ID bug Cisco [CSCdw52807](#) (solo utenti [registrati](#)).

- Catalyst 2900XL e 3500XL—Il supporto inizia con il software Cisco IOS versione 12.0(5)WC3 da quando l'ID bug Cisco è [CSCds45300](#) (solo utenti [registrati](#)).
- Catalyst serie 5000 - Dal software Cisco IOS versione 3.x. Sul modulo RSM/RSFC, il supporto inizia nel software Cisco IOS versione 12.1(6)E1 da quando è stato usato l'ID bug Cisco [CSCds50549](#) (solo utenti [registrati](#)).
- Moduli Catalyst 5000/6000 ATM - A partire dal software Cisco IOS versione 12.0(14)W05(20), fare riferimento all'ID bug Cisco [CSCds07238](#) (solo utenti [registrati](#)).
- Catalyst 6000 Gigabit Ethernet WAN OSM: dal software Cisco IOS versione 12.1.12E, fare riferimento all'ID bug Cisco [CSCdw64849](#) (solo utenti [registrati](#)).
- Catalyst serie 6000: tutte le versioni del software Cisco IOS. Il supporto di WS-F6K-MSFC e MSM inizia nel software Cisco IOS versione 12.1(8a)E4.
- Catalyst serie 8500 - Il supporto inizia a partire dal software Cisco IOS versione 12.0(5)W5(13).
- Router Cisco 3600, 4000 e piattaforme superiori - A partire dal software Cisco IOS versione 12.0(1) e dal software Cisco IOS versione 12.0(1)T, fare riferimento all'ID bug Cisco CSCdj93712 (solo utenti [registrati](#)) e all'ID bug Cisco [CSCdt58029](#) (solo utenti [registrati](#)).
- Interfacce Frame Relay: a partire dai software Cisco IOS versione 12.0(17)S e Cisco IOS versione 12.2(4)T3, fare riferimento ai [contatori Frame Relay a 64 bit](#).
- Interfacce OC3 ATM: a partire dal software Cisco IOS versione 12.0(6)T, fare riferimento all'ID bug Cisco [CSCdm45357](#) (solo utenti [registrati](#)).
- Interfacce tunnel: a partire dal software Cisco IOS versione 12.0(16)S, fare riferimento all'ID bug Cisco [CSCdt58029](#) (solo utenti [registrati](#)).

Nota: il software Cisco IOS non supporta contatori a 64 bit per velocità di interfaccia inferiori a 20 Mbps. Ciò significa che i contatori a 64 bit non sono supportati sulle porte Ethernet a 10 MB. Solo le porte Fast-Ethernet da 100 MB e altre porte ad alta velocità supportano contatori a 64 bit.

D. I contatori SNMP ifInOctets e ifOutOctets sono uguali ai contatori In/Out di show interfaces?

R. Sì, ma solo quando SNMP è abilitato dall'avvio. Se si accende un dispositivo Cisco, quindi si abilita il protocollo SNMP, i contatori SNMP iniziano da 0. Non rilevano automaticamente i valori dall'output della CLI.

D. I contatori ifInOctets e ifOutOctets includono il sovraccarico di framing (protocollo point-to-point, controllo di collegamento dati di alto livello)?

R. Sì.

D. In un'interfaccia della modalità di trasferimento asincrono i contatori includono l'intestazione della cella?

A. I contatori ATM (Asynchronous Transfer Mode) non includono il sovraccarico ATM (intestazioni di cella e spaziatura interna AAL5).

D. Perché i contatori SNMP non restituiscono lo stesso numero dei comandi show della CLI?

R. Un oggetto SNMP definito come contatore deve rispettare la [RFC1155](#) :

"3.2.3.3. Contatore

Questo tipo a livello di applicazione rappresenta un numero intero non negativo che aumenta in modo monotono fino a raggiungere un valore massimo, quando si avvolge e inizia a crescere di nuovo da zero. In questo promemoria viene specificato un valore massimo di $2^{32}-1$ (4294967295 decimali) per i contatori."

Non esistono metodi per reimpostare un contatore SNMP su zero senza dover ricaricare il dispositivo.

L'output del contatore da un comando **show** della CLI può essere reimpostato sulle interfacce perché le restrizioni SNMP non sono presenti.

I contatori di interfaccia originali definiti in MIB-2 sono contatori a 32 bit. Per un'interfaccia a 10 Mbps, un contatore a 32 bit potrebbe teoricamente andare a capo in 57 minuti. È facile evitare discontinuità con un periodo così lungo. Ma per 100 Mbps, il tempo di avvolgimento teorico minimo è di 5,7 minuti. Per interfacce a 1 Gbps, cade a 34 secondi. Questi tempi sono concessi per la trasmissione di pacchetti back-to-back di dimensioni complete, un ideale teorico. Tuttavia, maggiore è la velocità dell'interfaccia, più difficile sarà evitare di perdere una copertura del contatore. Per risolvere questo problema, SMI SNMPv2 ha definito un nuovo tipo di oggetto, counter64, per i contatori a 64 bit. Sono pertanto disponibili diversi nuovi contatori a 64 bit definiti nella tabella di interfaccia delle estensioni (ifxTable) definita nella [RFC 1573](#) (successivamente sostituita dalla [RFC 2233](#)). Questi sono tratti dal documento [IF-MIB-V1SMI.my](#) (solo utenti [registrati](#)).

ifHCInOctets (.1.3.6.1.2.1.31.1.1.6)	IfHCOctets (1.3.6.1.2.1.31.1.1.10)
IfHCInUcastPkts (.1.3.6.1.2.1.31.1.1.7)	IfHCOUcastPkts (.1.3.6.1.2.1.31.1.1.11)
ifHCInMulticastPkts (.1.3.6.1.2.1.31.1.1.8)	IfHCOMulticastPkts (.1.3.6.1.2.1.31.1.1.12)
ifHCInPacchettiTrasmissione (.1.3.6.1.2.1.31.1.1.9)	ifHCOBroadcastPkts (.1.3.6.1.2.1.31.1.1.13)

Anche se il supporto di base per i contatori a 64 bit è stato scritto nel software Cisco IOS versione 11.3, che inizia dal software Cisco IOS versione 12.0, solo se gli HCInOctets (.1.3.6.1.2.1.31.1.1.6) e gli HCOctets (1.3.6.1.2.1.31.1.1.10) sono stati implementati solo per le sottointerfacce LEC LANE ATM. Per gli switch Catalyst per gruppi di lavoro, il supporto di contatori a 64 bit è stato implementato nella versione 3.1.

Nota: è necessario utilizzare il protocollo SNMPv2c o SNMPv3 per recuperare qualsiasi oggetto contatore 64.

Contatori SNMP e show Command Equivalent Questions

D. Cosa fanno i router Cisco per le seguenti variabili MIB SNMP: ifInOctets, ifInUcastPkts, ifInNUcastPkts, ifInDiscards, ifInErrors, ifInUnknownProtos, ifOutOctets, ifOutUcastPkts, ifOutNUcastPkts, ifOutDiscards, ifOutErrors e

ifOutQLen?

R. Per i dettagli, vedere questa tabella. Questi sono tratti dalla [RFC1213-MIB](#) (solo utenti [registrati](#)).

ifInNUcastPkts (.1.3.6.1.2.1.2.2 .1.12)	Numero di pacchetti multicast e broadcast in ingresso.
ifInDiscard (.1.3.6.1.2.1.2.2 .1.13)	Questi vengono conteggiati come nessun buffer come indicato nel comando show interfaces .
ifInErrors (.1.3.6.1.2.1.2.2 .1.14)	Questi sono i conteggi di tutti gli errori di input riflessi nel comando show interfaces .
ifInProtocolli sconosciuti (.1.3.6.1.2.1.2.2 .1.15)	Questi sono conteggiati come errori non classificati.
ifOutOctets (.1.3.6.1.2.1.2.2 .1.16)	Questi sono conteggi del numero di byte restituiti dall'interfaccia come mostrato nel comando show interfaces .
ifOutUcastPkts (.1.3.6.1.2.1.2.2 .1.17)	Numero di pacchetti broadcast e multicast in uscita.
ifOutDiscard (.1.3.6.1.2.1.2.2 .1.19)	Questi vengono conteggiati come perdite di output come mostrato nel comando show interfaces .
ifOutErrors (.1.3.6.1.2.1.2.2 .1.20)	Questi vengono conteggiati come errori di output come mostrato nel comando show interfaces .
ifOutQLen (.1.3.6.1.2.1.2.2 .1.21)	Il numero di pacchetti che possono trovarsi nella coda di output, come mostrato nel comando show interfaces .

Le variabili elencate in precedenza che non dicono di essere visualizzate in **show interfaces** non sono disponibili in altri percorsi oltre a SNMP.

Esempi

In questo esempio viene utilizzato uno switch 3640 con software Cisco IOS versione 12.2(2)T1. La stringa della community di sola lettura (RO) utilizzata è pubblica e la stringa della community di lettura/scrittura (RW) utilizzata è privata. Per ulteriori informazioni su come configurare le stringhe della community SNMP sui dispositivi, consultare il documento sulla [configurazione delle stringhe della community SNMP](#).

Questo output è tipico del comando **show ip interface brief** eseguito in modalità di abilitazione:

```
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

Questo output è l'oggetto MIB se **Descr (.1.3.6.1.2.1.2.2.1.2)** per il router precedente, una stringa di testo che contiene informazioni sull'interfaccia. Fornisce il nome dell'interfaccia e la descrizione ottenuti, che usano l'output del comando CLI precedente. **ifName (.1.3.6.1.2.1.31.1.1.1.1)** può essere utilizzato anche se **ifDescr** fornisce la descrizione dell'interfaccia insieme al nome, mentre **ifName** fornisce solo il nome dell'interfaccia.

```
snmpwalk 172.16.99.20 public .1.3.6.1.2.1.2.2.1.2
interfaces.ifTable.ifEntry.ifDescr.1 = ATM1/0
interfaces.ifTable.ifEntry.ifDescr.2 = BRI0/0
interfaces.ifTable.ifEntry.ifDescr.3 = FastEthernet0/0
interfaces.ifTable.ifEntry.ifDescr.4 = Serial0/0
interfaces.ifTable.ifEntry.ifDescr.5 = BRI0/0:1
interfaces.ifTable.ifEntry.ifDescr.6 = BRI0/0:2
interfaces.ifTable.ifEntry.ifDescr.7 = Serial0/1
interfaces.ifTable.ifEntry.ifDescr.8 = Null0
interfaces.ifTable.ifEntry.ifDescr.10 = Foreign Exchange Office 2/0/0
interfaces.ifTable.ifEntry.ifDescr.11 = Foreign Exchange Office 2/0/1
interfaces.ifTable.ifEntry.ifDescr.12 = recEive And transMit 3/0/0
interfaces.ifTable.ifEntry.ifDescr.13 = recEive And transMit 3/0/1
interfaces.ifTable.ifEntry.ifDescr.14 = Loopback0
interfaces.ifTable.ifEntry.ifDescr.15 = Loopback1
interfaces.ifTable.ifEntry.ifDescr.16 = Loopback101
interfaces.ifTable.ifEntry.ifDescr.17 = Loopback200
interfaces.ifTable.ifEntry.ifDescr.18 = Loopback201
interfaces.ifTable.ifEntry.ifDescr.19 = Serial0/0.1
interfaces.ifTable.ifEntry.ifDescr.20 = ATM1/0.109-atm subif
interfaces.ifTable.ifEntry.ifDescr.21 = ATM1/0.109-aal5 layer
interfaces.ifTable.ifEntry.ifDescr.22 = Virtual-Template1
interfaces.ifTable.ifEntry.ifDescr.23 = Voice Encapsulation (POTS) Peer: 1
interfaces.ifTable.ifEntry.ifDescr.24 = Voice Over IP Peer: 2
interfaces.ifTable.ifEntry.ifDescr.25 = Voice Encapsulation (POTS) Peer: 111
interfaces.ifTable.ifEntry.ifDescr.26 = Voice Over IP Peer: 222
interfaces.ifTable.ifEntry.ifDescr.27 = Voice Over IP Peer: 1234
interfaces.ifTable.ifEntry.ifDescr.28 = Voice Over IP Peer: 300000
interfaces.ifTable.ifEntry.ifDescr.29 = Voice Over FR Peer: 3
interfaces.ifTable.ifEntry.ifDescr.30 = Voice Over IP Peer: 99
interfaces.ifTable.ifEntry.ifDescr.31 = Voice Encapsulation (POTS) Peer: 9
interfaces.ifTable.ifEntry.ifDescr.32 = BRI0/0-Physical
interfaces.ifTable.ifEntry.ifDescr.33 = BRI0/0-Signaling
interfaces.ifTable.ifEntry.ifDescr.34 = BRI0/0:1-Bearer Channel
interfaces.ifTable.ifEntry.ifDescr.35 = BRI0/0:2-Bearer Channel
```

1. ifInDiscard (.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

```

Il valore **ifInDiscards** è zero per tutte le interfacce del router. Se si confronta questo con il risultato CLI del comando **show interfaces fast Ethernet 0/0**, si conferma il risultato:

```

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
```

Questo output mostra che esiste un solo errore di input per

l'interfaccia interfaces.ifTable.ifEntry.ifInErrors.7 = Contatore32: 1. Per determinare di quale interfaccia si tratta, confrontarla con l'output di **ifDescr** riportato sopra, che indica che l'interfaccia proviene da **interfacce.ifTable.ifEntry.ifDescr.7 = Serial0/1**. Eseguire quindi il comando **show interfaces serial 0/1** in modalità abilitazione per verificare il risultato precedente:

```
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 lwld
  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
```

Se si confronta il risultato precedente con l'output di **ifDescr**, verrà indicato quanto segue:**interfaces.ifTable.ifEntry.ifOutOctets.1 = Contatore32: 98** corrisponde a **interfaces.ifTable.ifEntry.ifDescr.1 = ATM1/0****interfaces.ifTable.ifEntry.ifOutOctets.3 = Contatore32: 765470674** corrisponde a **interfacce.ifTable.ifEntry.ifDescr.3 = FastEthernet0/0****interfaces.ifTable.ifEntry.ifOutOctets.21 = Contatore32: 98** corrisponde a **interfaces.ifTable.ifEntry.ifDescr.21 = livello ATM1/0.109-aal5**Questo è l'output del comando **show interfaces** della CLI per ciascuna delle interfacce precedenti eseguite in modalità abilitazione:

```
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
```

L'output del comando **ifOutOctets** non corrisponde all'output CLI del comando **show interfaces Fast Ethernet 0/0**, ma è simile. Ciò perché potrebbe verificarsi un ritardo quando le interfacce vengono sottoposte a polling e quando viene eseguito il 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. ifOutDiscard (.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

```

Il valore di **ifOutDiscards** è zero per tutte le interfacce. Se si usa il comando **show interfaces fastEthernet 0/0** come esempio, questo comando restituisce il risultato:

```

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
```

Il valore di **ifOutErrors** è zero per tutte le interfacce. Se si usa il comando **show interfaces fastEthernet 0/0** come esempio, questo comando restituisce il risultato:

```
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
```

Il valore di ifOutQLen è zero per tutte le interfacce. Con il comando show interfaces fastEthernet 0/0 come esempio:

```
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
```

D. Qual è la relazione tra le istruzioni `show interfaces no buffer` e le perdite nella coda di input? Perché gli `inDiscard` di SNMP non forniscono conteggi dei buffer e non perdite della coda di input, mentre gli `outDiscard` di SNMP danno perdite della coda di output?

R. La funzione `locIfInputQueueDrops/ifInDiscards` funziona in modo diverso da `locIfOutputQueueDrops/ifOutDiscards`. `IfInDiscards` conteggia il numero di pacchetti eliminati per mancanza di una risorsa di sistema, ad esempio un buffer. Si tratta in genere di un sottoinsieme di `locIfInputQueueDrops`. È spesso visualizzato `locIfInputQueueDrops = ifInDiscards`. Tuttavia, `locIfInputQueueDrops` conta anche il numero di pacchetti scartati perché hanno raggiunto il limite della coda di input. In genere, quindi, viene visualizzato `locIfInputQueueDrops > ifInDiscards`.

Riepilogo

`locIfInputQueueDrops` = Interruzioni limite coda + Nessuna interruzione buffer
`ifInDiscards` = Nessuna interruzione buffer (ed è un sottoinsieme di `locIfInputQueueDrops`)

Le proprietà `locIfOutputQueueDrops` e `ifOutDiscards` sono sempre uguali quando contano gli stessi eventi. Questi eventi hanno raggiunto il limite della coda di output e non hanno un buffer di trasmissione hardware quando un pacchetto viene commutato rapidamente da un'interfaccia all'altra. Gli OID degli oggetti MIB precedenti sono i seguenti:

Da OLD-CISCO-INTERFACES-MIB (solo utenti registrati)	Da RFC1213-MIB (solo utenti registrati)
<code>locIfInputQueueDrops</code> = .1.3.6.1.4.1.9.2.2.1.1.26	<code>ifInDiscards</code> = .1.3.6.1.2.1.2.2.1.13
<code>locIfOutputQueueDrops</code> = .1.3.6.1.4.1.9.2.2.1.1.27	<code>ifOutDiscards</code> = .1.3.6.1.2.1.2.2.1.19

D. È possibile eseguire il polling dei buffer su un router?

R. Sì. È possibile eseguire il polling di `ifInDiscards` per non eseguire il polling dei buffer.

D. Come fare per eseguire il polling delle gocce limite della coda su un router?

R. Con l'uso del protocollo SNMP, il comando `show interfaces` non consente di suddividere i singoli elementi che vanno nelle derivazioni di output.

Considerare le nuove informazioni relative al contatore delle perdite di output:

Cadute di input = cadute limite coda + cadute limitazione + cadute complete coda SMT + cadute RSRB + nessuna perdita di buffer

Inoltre, i contatori SNMP non vengono mai azzerati, anche se le interfacce vengono azzerate.

[Informazioni correlate](#)

- [Suggerimenti tecnici per i servizi applicativi IP](#)
- [Documentazione e supporto tecnico – Cisco Systems](#)