

Compteurs SNMP : Forum aux questions

Contenu

[Introduction](#)

[Questions sur les compteurs SNMP](#)

[Compteurs SNMP et show Command Equivalent Questions](#)

[Informations connexes](#)

Introduction

Ce document apporte des réponses aux questions fréquemment posées, et guide les utilisateurs afin qu'ils trouvent des ressources utiles sur le SNMP, et les problèmes SNMP qui sont liés au matériel de Cisco.

Questions sur les compteurs SNMP

Q. Quelle MIB dois-je utiliser pour les compteurs d'interface ?

A. La gestion des interfaces sur SNMP repose sur deux tables : [ifTable](#) (clients [enregistrés](#) uniquement) et son extension, [ifXTable](#) (clients [enregistrés](#) uniquement) décrits dans la norme RFC1213/RFC2233. Les interfaces peuvent avoir plusieurs couches, qui dépendent du support, et chaque sous-couche est représentée par une ligne distincte dans le tableau. La relation entre la couche supérieure et la couche inférieure est décrite dans [ifStackTable](#) (clients [enregistrés](#) uniquement). Le ifTable définit des compteurs 32 bits pour les octets entrants et sortants ([ifInOctets](#) (clients [enregistrés](#) uniquement) /[ifOutOctets](#)), les paquets ([ifInUcastPkts](#) ([clients enregistrés uniquement](#)) /[ifOutUcastPkts](#) ([clientsenregistrés](#) uniquement) , [InNU Pkts](#) ([clients enregistrés uniquement](#)) /[ifOutNUcastPkts](#) ([clients enregistrés uniquement](#))), [erreurs et rejets](#). Le ifXTable fournit des compteurs 64 bits similaires, également appelés compteurs de haute capacité (HC) : [ifHCInOctets](#) ([clients enregistrés uniquement](#)) /[ifHCOutOctets](#) (clients [enregistrés](#) uniquement) et [ifHCInUcastPkts](#) ([clients enregistrés uniquement](#)) /[ifHCOutUcastPkts](#) ([clients enregistrés](#)) .

Q. Quand faut-il utiliser des compteurs 64 bits ?

A. [La RFC 2233](#) a adopté des compteurs 64 bits étendus pour les interfaces à haute capacité dans lesquelles les compteurs 32 bits ne fournissent pas une capacité suffisante et un enchaînement trop rapide.

À mesure que la vitesse des supports réseau augmente, la durée minimale d'encapsulation d'un compteur 32 bits diminue. Par exemple, un flux de 10 Mbits/s de paquets back-to-back, de taille complète, provoque l'encapsulation de ifInOctets en un peu plus de 57 minutes. À 100 Mbits/s, la durée minimale de post-appel est de 5,7 minutes et à 1 Gbits/s, la durée minimale est de 34 secondes.

Remarque : Les compteurs SNMP s'enroulent, pas les compteurs de l'interface de ligne de commande (CLI).

Pour les interfaces qui fonctionnent à 20 000 000 (20 millions) bits par seconde ou moins, vous devez utiliser des compteurs d'octets et de paquets de 32 bits. Pour les interfaces qui fonctionnent plus rapidement que 20 millions de bits par seconde et plus lentement que 650 000 000 bits par seconde, vous devez utiliser des compteurs de paquets 32 bits et des compteurs d'octet 64 bits. Pour les interfaces qui fonctionnent à 650 000 000 bits/seconde ou plus, des compteurs de paquets et d'octet 64 bits doivent être utilisés.

Par conséquent, le logiciel Cisco IOS® ne prend pas en charge les compteurs 64 bits pour des débits d'interface inférieurs à 20 Mbits/s. Cela signifie que les compteurs 64 bits ne sont pas pris en charge sur les ports Ethernet 10 Mo, que les ports Fast Ethernet 100 Mo et les autres ports haut débit prennent en charge les compteurs 64 bits.

Q. Quelle version de SNMP est nécessaire pour interroger les compteurs 64 bits ?

A. SNMPv2C ou SNMPv3 est requis pour interroger les compteurs 64 bits. SNMPv1 ne prend pas en charge les compteurs 64 bits. Sachez que ifInOctets = .1.3.6.1.2.1.2.2.1.10 est un compteur 32 bits tandis que la version 64 bits est ifHCInOctets = .1.3.6.1.2.1.31.1.1.6.

Exemple :

Catalyst 5000 utilise HP OpenView snmpget, qui prend par défaut la valeur SNMPv1

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

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

Requête avec SNMPv1, valeur par défaut pour HP OpenView snmpget

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

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

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

Même requête avec SNMPv2C

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

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

Q. Quels périphériques Cisco implémentent des compteurs SNMP 64 bits, en particulier pour la base de données IF-MIB ?

A. Ces périphériques Cisco implémentent des compteurs SNMP 64 bits :

Q. Sur une interface en mode de transfert asynchrone, les compteurs incluent-ils l'en-tête de cellule ?

A. Les compteurs ATM (Asynchronous Transfer Mode) n'incluent pas la surcharge ATM (en-têtes de cellule et remplissage AAL5).

Q. Pourquoi les compteurs SNMP ne retournent-ils pas le même numéro que les commandes show CLI ?

A. Un objet SNMP défini comme compteur doit respecter [RFC1155](#) :

"3.2.3.3 . Compteur

Ce type à l'échelle de l'application représente un entier non négatif qui augmente de manière monotone jusqu'à atteindre une valeur maximale, lorsqu'il se retourne et commence à augmenter à nouveau à partir de zéro. Cette note spécifie une valeur maximale de $2^{32}-1$ (4294967295 décimale) pour les compteurs. »

Il n'existe aucune méthode pour réinitialiser un compteur SNMP à zéro sans avoir à recharger le périphérique.

La sortie du compteur d'une commande **show** CLI peut être réinitialisée sur les interfaces car les restrictions SNMP ne sont pas présentes.

Les compteurs d'interface d'origine définis dans MIB-2 sont des compteurs 32 bits. Pour une interface de 10 Mbits/s, un compteur de 32 bits peut théoriquement s'encapsuler en 57 minutes. Il est facile d'éviter les discontinuités avec une si longue période. Mais pour 100 Mbits/s, la durée minimale de l'encapsulation théorique est de 5,7 minutes. Pour les interfaces 1 Gbit/s, il tombe à 34 secondes. Il est admis que ces temps sont pour la transmission de paquets dos à dos pleine taille, un idéal théorique. Malgré cela, plus la vitesse de l'interface est élevée, plus il devient difficile d'éviter de manquer un recouvrement de compteur. Pour résoudre ce problème, SNMPv2 SMI a défini un nouveau type d'objet, counter64, pour les compteurs 64 bits. Par conséquent, il existe plusieurs nouveaux compteurs 64 bits définis dans la table d'interface d'extension (ifxTable) définie dans [RFC 1573](#) (remplacé ultérieurement par [RFC 2233](#)). Ceux-ci proviennent du [IF-MIB-V1SMI.my](#) (clients [enregistrés](#) uniquement).

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)	ifHCOctetsUcastPkts (.1.3.6.1.2.1.31.1.1.11)
ifHCInMulticastPkts (.1.3.6.1.2.1.31.1.1.8)	ifHCOctetsMulticastPkts (.1.3.6.1.2.1.31.1.1.12)
ifHCInBroadcastPkts (.1.3.6.1.2.1.31.1.1.9)	ifHCOctetsBroadcastPkts (.1.3.6.1.2.1.31.1.1.13)

Bien que la prise en charge de base des compteurs 64 bits ait été écrite dans le logiciel Cisco IOS Version 11.3, qui commence à partir du logiciel Cisco IOS Version 12.0, uniquement siHCInOctets (.1.3.6.1.2.1.31.1.1.6) et ifHCOctets (1.3.6.1.2.1.1.1 1.1.1.10) ont été implémentées pour les sous-interfaces LEC LANE ATM uniquement. Pour les commutateurs de groupe de travail Catalyst, la prise en charge des compteurs 64 bits a été implémentée dans la version 3.1.

Remarque : Vous devez utiliser le protocole SNMPv2c ou SNMPv3 afin de récupérer tout objet de compteur 64.

Compteurs SNMP et show Command Equivalent Questions

Q. Que font les routeurs Cisco pour les variables MIB SNMP suivantes : ifInOctets, ifInUcastPkts, ifInNUcastPkts, ifInDiscards, ifInErrors, ifInUnknownProtos, ifOutOctets, ifOutUcastPkts, ifOutNUcastPkts, ifOutDiscards, ifOutErrors et ifOutQLen ?

A. Voir ce tableau pour plus de détails. Celles-ci proviennent de la [RFC1213-MIB](#) (clients enregistrés uniquement).

ifInNUcastPkts (.1.3.6.1.2.1.2.2.1.12)	Il s'agit du nombre de paquets de diffusion et de multidiffusion entrants.
ifInDiscards (.1.3.6.1.2.1.2.2.1.13)	Celles-ci sont comptabilisées comme aucune mémoire tampon comme indiqué dans la commande show interfaces .
ifInErrors (.1.3.6.1.2.1.2.2.1.14)	Il s'agit du nombre de toutes les erreurs d'entrée, comme indiqué dans la commande show interfaces .
ifInUnknownProtos (.1.3.6.1.2.1.2.2.1.15)	Ces erreurs sont comptabilisées comme des erreurs non classifiées.
ifOutOctets (.1.3.6.1.2.1.2.2.1.16)	Il s'agit du nombre d'octets générés par l'interface, comme indiqué dans la commande show interfaces .
ifOutUcastPkts (.1.3.6.1.2.1.2.2.1.17)	Il s'agit du nombre de paquets de diffusion et de multidiffusion sortants.
ifOutDiscards (.1.3.6.1.2.1.2.2.1.19)	Celles-ci sont comptabilisées comme des pertes de sortie comme indiqué dans la commande show interfaces .
ifOutErrors (.1.3.6.1.2.1.2.2.1.20)	Ces erreurs sont comptabilisées comme des erreurs de sortie, comme indiqué dans la commande show interfaces .
ifOutQLen (.1.3.6.1.2.1.2.2.1.21)	Il s'agit du nombre de paquets autorisés à être sur la file d'attente de sortie, comme indiqué dans la commande show interfaces .

Les variables précédemment répertoriées qui ne disent pas qu'elles apparaissent dans **show interfaces** ne sont disponibles nulle part ailleurs que SNMP.

Exemples

Cet exemple utilise un 3640 qui s'exécute avec le logiciel Cisco IOS Version 12.2(2)T1. La chaîne de communauté en lecture seule (RO) utilisée est publique et la chaîne de communauté en lecture-écriture (RW) utilisée est privée. Référez-vous à [Configuration des chaînes de communauté SNMP](#) pour plus d'informations sur la configuration des chaînes de communauté SNMP sur les périphériques.

Cette sortie est typique de la commande **show ip interface brief** exécutée en mode enable :

```
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-Templatel 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
```

Ce résultat est l'objet MIB si **Descr (.1.3.6.1.2.1.2.2.1.2)** pour le routeur précédent, qui est une chaîne de texte qui contient des informations sur l'interface. Cela donne le nom et la description de l'interface tels qu'ils ont été obtenus, qui utilise la sortie de commande de l'interface de ligne de commande précédente. **ifName (.1.3.6.1.2.1.31.1.1.1.1)** peut également être utilisé mais **ifDescr** fournit la description de l'interface avec le nom, où **ifName** donne uniquement le nom de l'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-aal5 layer
interfaces.ifTable.ifEntry.ifDescr.22 = Virtual-Templatel
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
```

La valeur **ifInDiscards** est zéro pour toutes les interfaces de ce routeur. Si vous comparez ceci au résultat CLI de la commande **show interfaces fastEthernet 0/0**, ceci confirme le résultat :

```
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

```

Cette sortie montre qu'il n'y a qu'une seule erreur d'entrée pour

l'interface interfaces.ifTable.ifEntry.ifInErrors.7 = Compteur32 : 1. Afin de déterminer quelle interface il s'agit, comparez-la avec la sortie de **ifDescr** ci-dessus, qui montre que ceci

provient d'**interfaces.ifTable.ifEntry.ifDescr.7 = Serial0/1**. Exécutez maintenant la commande **show interfaces serial 0/1** en mode enable afin de vérifier le résultat précédent :

```

3600#show interfaces serial 0/1
Serial0/1 is administratively down, line protocol is down
  Hardware is DSCC4 Serial

```



```

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

```

3. ifOutOctets (.1.3.6.1.2.1.2.2.1.16) :

```
snmpwalk 172.16.99.20 public .1.3.6.1.2.1.2.2.1.16
```

```

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

```

Si vous comparez le résultat précédent avec le résultat du `ifDescr`, ceci indique :

- `interfaces.ifTable.ifEntry.ifOutOctets.1 = Compteur32 : 98` correspond à `interfaces.ifTable.ifEntry.ifDescr.1 = ATM1/0`
- `interfaces.ifTable.ifEntry.ifOutOctets.3 = Compteur32 : 765470674` correspond à `interfaces.ifTable.ifEntry.ifDescr.3 =`

FastEthernet0/0interfaces.ifTable.ifEntry.ifOutOctets.21 = Compteur32 : 98 correspond à **interfaces.ifTable.ifEntry.ifDescr.21 = couche ATM1/0.109-aal5** Ceci est le résultat de la commande **show interfaces** de l'interface de ligne de commande pour chacune des interfaces précédentes exécutées en mode enable :

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

```
3600#show interfaces fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0001.42b4.fe81 (bia 0001.42b4.fe81)
  Description: testme
  Internet address is 172.16.99.20/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 323 drops
  5 minute input rate 2000 bits/sec, 3 packets/sec
  5 minute output rate 1000 bits/sec, 1 packets/sec
    1772214 packets input, 178767841 bytes
    Received 1164210 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog
  0 input packets with dribble condition detected
  7149179 packets output, 765450524 bytes, 0 underruns(0/0/0)
  0 output errors, 0 collisions, 1 interface resets
  0 babbles, 0 late collision, 461 deferred
  0 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out
```

La sortie de **ifOutOctets** ne correspond pas à la sortie CLI de la commande **show interfaces FastEthernet 0/0**, mais elle est similaire. Ceci est dû au fait qu'il peut y avoir un délai lorsque les interfaces sont interrogées et lorsque la commande CLI est exécutée.

```
3600#show interfaces atm 1/0.109
ATM1/0.109 is down, line protocol is down
  Hardware is RS8234 ATMOC3
  Description: pvc
  Internet address is 10.164.0.46/30
```

```
MTU 4470 bytes, BW 2250 Kbit, DLY 80 usec,  
    reliability 5/255, txload 1/255, rxload 1/255  
Encapsulation ATM  
0 packets input, 0 bytes  
2 packets output, 98 bytes  
0 OAM cells input, 77093 OAM cells output  
AAL5 CRC errors : 0  
AAL5 SAR Timeouts : 0  
AAL5 Oversized SDUs : 0  
AAL5 length violation : 0  
AAL5 CPI Error : 0
```

4. ifOutDiscards (.1.3.6.1.2.1.2.2.1.19) :

```
snmpwalk 172.16.99.20 public .1.3.6.1.2.1.2.2.1.19
```

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

Le ifOutDiscards est zéro pour toutes les interfaces. Avec la commande show interfaces fastEthernet 0/0 comme exemple, cette commande produit le résultat suivant :

```
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
```

La valeur **ifOutErrors** est nulle pour toutes les interfaces. Avec la commande **show interfaces fastEthernet 0/0** comme exemple, cette commande produit le résultat suivant :

```
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

```

Le ifOutQLen est zéro pour toutes les interfaces. Avec la commande **show interfaces fastEthernet 0/0** comme exemple :

```
3600#show interfaces fastEthernet 0/0
```

```

FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0001.42b4.fe81 (bia 0001.42b4.fe81)
  Description: testme
  Internet address is 172.16.99.20/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255

```

```

Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, 100Mb/s, 100BaseTX/FX
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:00, output 00:00:00, output hang never
Last clearing of "show interface" counters never
Queueing strategy: fifo
Output queue 0/40, 0 drops; input queue 0/75, 323 drops
5 minute input rate 0 bits/sec, 1 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  1776912 packets input, 179225431 bytes
  Received 1167240 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 watchdog
  0 input packets with dribble condition detected
  7151102 packets output, 765796341 bytes, 0 underruns(0/0/0)
  0 output errors, 0 collisions, 1 interface resets
  0 babbles, 0 late collision, 461 deferred
  0 lost carrier, 0 no carrier
  0 output buffer failures, 0 output buffers swapped out

```

Q. Quelle est la relation entre les instructions `show interfaces no buffers` et les pertes de file d'attente d'entrée ? Pourquoi les inDisCARTS de SNMP ne donnent pas de nombre de tampons et non de pertes de file d'attente d'entrée, alors que les outDisCARTS de SNMP donnent des pertes de file d'attente de sortie ?

A. Le `locIfInputQueueDrops/ifInDiscards` fonctionne différemment de `locIfOutputQueueDrops/ifOutDiscards`. Le `ifInDiscards` compte le nombre de paquets qui sont jetés en raison de l'absence d'une ressource système telle qu'une mémoire tampon. Il s'agit généralement d'un sous-ensemble de `locIfInputQueueDrops`. Vous voyez souvent `locIfInputQueueDrops = ifInDiscards`. Mais `locIfInputQueueDrops` compte également le nombre de paquets abandonnés parce qu'ils atteignent la limite de file d'attente d'entrée. En règle générale, `locIfInputQueueDrops > ifInDiscards` s'affiche.

Résumé

`locIfInputQueueDrops` = Limite de file d'attente abandonnées + Pas de pertes de mémoire tampon
`silnDiscards` = Aucune perte de mémoire tampon (et est un sous-ensemble de `locIfInputQueueDrops`)

Les `locIfOutputQueueDrops` et `ifOutDiscards` sont toujours égaux lorsqu'ils comptent les mêmes événements. Ces événements atteignent la limite de la file d'attente de sortie, et n'ont pas de tampon de taxe matériel lorsqu'un paquet est commuté rapidement d'une interface à une autre. Les OID des objets MIB précédents sont les suivants :

À partir de VIEILLE-CISCO-INTERFACES-MIB (clients enregistrés uniquement)	À partir de RFC1213-MIB (clients enregistrés uniquement)
<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

Q. Puis-je interroger aucune mémoire tampon sur un routeur ?

A. Oui. Vous pouvez interroger iflnDiscards afin d'interroger aucune mémoire tampon.

Q. Comment puis-je interroger les pertes de limite de file d'attente sur un routeur ?

A. Avec l'utilisation de SNMP, il n'y a aucun moyen pour la commande **show interfaces** de décomposer les éléments individuels qui entrent dans les pertes de sortie.

Considérez ces nouvelles informations sur ce qui va dans le compteur des pertes de sortie :

Pertes d'entrée = Pertes de limite de file d'attente + Pertes de limitation + Pertes de file d'attente SMT complètes + Pertes RSRB + aucune perte de tampon

En outre, les compteurs SNMP ne sont jamais effacés, même si les interfaces sont effacées.

[Informations connexes](#)

- [Conseils techniques sur les services d'applications IP](#)
- [Support et documentation techniques - Cisco Systems](#)