

Utilisation de SNMP pour trouver un numéro de port d'une adresse MAC sur un commutateur Catalyst

Contenu

[Introduction](#)

[Conditions préalables](#)

[Conditions requises](#)

[Composants utilisés](#)

[Conventions](#)

[Fond](#)

[Détails des variables MIB, qui inclut les identifiants d'objet \(les OID\)](#)

[Obtenez le numéro de port sur lequel une adresse MAC a été apprise](#)

[Instructions pas à pas](#)

[Informations connexes](#)

[Introduction](#)

Ce document décrit comment employer le protocole de gestion de réseau simple (SNMP) pour obtenir le numéro de port sur un commutateur Cisco Catalyst duquel vous connaissez l'adresse MAC.

[Conditions préalables](#)

[Conditions requises](#)

Les lecteurs de ce document devraient avoir connaissance des sujets suivants :

- Comment obtenir des VLAN d'un Catalyst commuté avec l'utilisation du SNMP
- Comment utiliser l'indexation de chaîne de la communauté avec le SNMP
- L'utilisation générale de la commande de snmp get et l'**inspection** commandent

[Composants utilisés](#)

Ce document applique aux Commutateurs de Catalyst qui exécutent le SYSTÈME D'EXPLOITATION régulier de Catalyst (CatOS) ou le logiciel de Cisco IOS®. Les supports logiciels le [MIB DE PONT](#) et l'[IF-MIB](#).

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Catalyst 3524XL qui exécute la version du logiciel Cisco IOS 12.0(5)WC5a
- Version 5.0.6 Net-SNMP **Note:** Pour obtenir ce logiciel, référez-vous au Net-[SNMP](#) .

Les informations contenues dans ce document ont été créées à partir des périphériques d'un environnement de laboratoire spécifique. Tous les périphériques utilisés dans ce document ont démarré avec une configuration effacée (par défaut). Si votre réseau est opérationnel, assurez-vous que vous comprenez l'effet potentiel de toute commande.

[Conventions](#)

Pour plus d'informations sur les conventions de documents, reportez-vous à [Conventions relatives aux conseils techniques Cisco](#).

[Fond](#)

Pour plus d'informations sur la façon de questionner la table de mémoire de contenu adressable (CAM), les VLAN, et tout le MIB relatif, tel que le CISCO-VTP-MIB et le MIB DE PONT, se rapportent à la section de [fond du document comment obtenir des entrées de CAM dynamique \(Tableau de CAM\) pour des Commutateurs de Catalyst utilisant le SNMP](#).

[Détails des variables MIB, qui inclut les identifiants d'objet \(les OID\)](#)

```
.1.3.6.1.2.1.17.4.3.1.1
dot1dTpFdbAddress OBJECT-TYPE
    -- FROM BRIDGE-MIB
    -- TEXTUAL CONVENTION MacAddress
    SYNTAX          OCTET STRING (6)
    MAX-ACCESS      read-only
    STATUS          Mandatory
    DESCRIPTION     "A unicast MAC address for which the bridge has forwarding
                    and/or filtering information."
 ::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) dot1dBridge(17) dot1dTp(4)
 dot1dTpFdbTable(3) dot1dTpFdbEntry(1) 1 }

.1.3.6.1.2.1.17.4.3.1.2
dot1dTpFdbPort OBJECT-TYPE
    -- FROM BRIDGE-MIB
    SYNTAX          Integer
    MAX-ACCESS      read-only
    STATUS          Mandatory
    DESCRIPTION     "Either the value "0", or the port number of the port on which
                    a frame having a source
                    address equal to the value of the corresponding instance of
                    dot1dTpFdbAddress has been seen.
                    A value of "0" indicates that the port number has not been learned,
                    but that the bridge does
                    have some forwarding/filtering information about this address (that is,
                    in the StaticTable).
                    Implementors are encouraged to assign the port value to this
                    object whenever it is
                    learned, even for addresses for which the corresponding value of
                    dot1dTpFdbStatus is not learned(3)."
 ::= { iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) dot1dBridge(17) dot1dTp(4)
 dot1dTpFdbTable(3) dot1dTpFdbEntry(1) 2 }

.1.3.6.1.2.1.2.2.1.1
```

```
ifIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION  "A unique value, greater than zero, for each interface.  It
                 is recommended that values are assigned contiguously
                 starting from 1.  The value for each interface sub-layer
                 must remain constant at least from one re-initialization of
                 the entity's network management system to the next re-
                 initialization."
    ::= { ifEntry 1 }
```

```
.1.3.6.1.2.1.17.1.4.1.2
dot1dBasePortIfIndex OBJECT-TYPE
    SYNTAX      INTEGER
    ACCESS      read-only
    STATUS      mandatory
    DESCRIPTION  "The value of the instance of the ifIndex object,
                 defined in MIB-II, for the interface corresponding
                 to this port."
    ::= { dot1dBasePortEntry 2 }
```

```
.1.3.6.1.2.1.31.1.1.1.1
ifName OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION  "The textual name of the interface.  The value of this
                 object should be the name of the interface as assigned by
                 the local device and should be suitable for use in commands
                 entered at the device's `console'.  This might be a text
                 name, such as `le0' or a simple port number, such as `1',
                 depending on the interface naming syntax of the device.  If
                 several entries in the ifTable together represent a single
                 interface as named by the device, then each will have the
                 same value of ifName.  Note that for an agent which responds
                 to SNMP queries concerning an interface on some other
                 (proxied) device, then the value of ifName for such an
                 interface is the proxied device's local name for it.
                 If there is no local name, or this object is otherwise not
                 applicable, then this object contains a zero-length string."
    ::= { ifXEntry 1 }
```

[Obtenez le numéro de port sur lequel une adresse MAC a été apprise](#)

[Instructions pas à pas](#)

Terminez-vous les étapes dans cette section afin d'employer le SNMP pour obtenir le numéro de port sur lequel une adresse MAC a été apprise. Considérez que le numéro de port est dans VLAN1.

Note: Dans les commandes dans cette section :

- `le public` est l'identifiant de communauté à accès en lecture.
- `@1` est la partie VLAN 1 de l'identifiant de communauté à accès en lecture.
- `crumpy` est le nom d'hôte de périphérique.**Note:** Vous pouvez également utiliser l'adresse IP pour ce nom d'hôte.

Note: La section de [conclusion](#) utilise les valeurs qui apparaissent en *italique* dans la sortie de commande.

1. Récupérez les VLAN. Utilisez la commande de **snmpwalk** sur l'objet de `vtpVlanState` (`.1.3.6.1.4.1.9.9.46.1.3.1.1.2`) :

```
%snmpwalk -c public crumpy .1.3.6.1.4.1.9.9.46.1.3.1.1.2
CISCO-VTP-MIB::vtpVlanState.1.1 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.3 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.7 = INTEGER: operational(1)
CISCO-VTP-MIB::vtpVlanState.1.10 = INTEGER: operational(1)
...
```

Note: Cette commande utilise l'[indexation de chaîne de la communauté](#). La commande utilise également le [vtpVlanState](#), qui a OID `.1.3.6.1.4.1.9.9.46.1.3.1.1.2`. Si vous avez chargé le MIB à votre système d'administration de réseaux (NMS), vous pouvez utiliser le nom d'objet au lieu de l'OID. Émettez cette commande à la place :

```
%snmpwalk -c public@1 crumpy vtpVlanState
```

Note: Vous pouvez également utiliser les noms d'objet dans les étapes 2 à 6.

2. Émettez cette commande afin d'obtenir la table d'adresse MAC en considérant que le port appartient à VLAN1 :

```
snmpwalk -c public@1 crumpy .1.3.6.1.2.1.17.4.3.1.1
17.4.3.1.1.0.0.12.7.172.8 = Hex: 00 00 0C 07 AC 08
17.4.3.1.1.0.1.2.27.80.145 = Hex: 00 01 02 1B 50 91
17.4.3.1.1.0.1.3.72.77.90 = Hex: 00 01 03 48 4D 5A
17.4.3.1.1.0.1.3.72.221.191 = Hex: 00 01 03 48 DD BF
...
```

Note: Fournissez le nombre approprié VLAN après que la chaîne de la communauté. Dans cet exemple, c'est VLAN1. Les listes de commandes toutes les adresses MAC qui ont été apprises sur tous les ports qui appartiennent au VLAN 1.

3. Émettez cette commande de déterminer le numéro de port de passerelle pour le VLAN 1 :

```
snmpwalk -c public@1 crumpy .1.3.6.1.2.1.17.4.3.1.2
17.4.3.1.2.0.0.12.7.172.8 = 13
17.4.3.1.2.0.1.2.27.80.128 = 13
17.4.3.1.2.0.1.2.27.80.145 = 13
17.4.3.1.2.0.1.2.163.145.225 = 13
...
```

Note: Le VLAN 1 est [dot1dTpFdbPort](#), ou `.1.3.6.1.2.1.17.4.3.1.2`.

4. Émettez cette commande de tracer le port de passerelle à l'[ifIndex](#), OID `.1.3.6.1.2.1.2.2.1.1` :

```
snmpwalk -c public@1 crumpy .1.3.6.1.2.1.17.1.4.1.2
17.1.4.1.2.13 = 2
17.1.4.1.2.14 = 3
17.1.4.1.2.15 = 4
17.1.4.1.2.16 = 5
```

Cette commande questionne le [dot1dBasePortIfIndex](#), qui a OID `.1.3.6.1.2.1.17.1.4.1.2`.

5. Employez la commande d'**inspection** avec l'[ifName](#) afin de corréler la valeur d'[ifIndex](#) avec un nom de port approprié. Émettez la commande suivante : **Note:** L'[ifName](#) a OID

```
.1.3.6.1.2.1.31.1.1.1.1.
```

```
snmpwalk -c public@1 crumpy .1.3.6.1.2.1.31.1.1.1.1
```

```
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.1 = VL1  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.2 = Fa0/1  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.3 = Fa0/2  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.4 = Fa0/3  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.5 = Fa0/4  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.6 = Fa0/5  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.7 = Fa0/6  
...
```

6. Joignez une adresse MAC au port sur lequel l'adresse a été apprise. De l'étape 1, l'adresse MAC est :

```
snmpwalk -c public@1 crumpy .1.3.6.1.2.1.31.1.1.1.1
```

```
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.1 = VL1  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.2 = Fa0/1  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.3 = Fa0/2  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.4 = Fa0/3  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.5 = Fa0/4  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.6 = Fa0/5  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.7 = Fa0/6  
...
```

De l'étape 2, le port de passerelle indique que l'adresse MAC appartient pour jeter un pont sur le numéro de port 13 :

```
snmpwalk -c public@1 crumpy .1.3.6.1.2.1.31.1.1.1.1
```

```
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.1 = VL1  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.2 = Fa0/1  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.3 = Fa0/2  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.4 = Fa0/3  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.5 = Fa0/4  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.6 = Fa0/5  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.7 = Fa0/6  
...
```

De l'étape 3, le numéro de port 13 de passerelle a l'ifIndex le numéro 2 :

```
snmpwalk -c public@1 crumpy .1.3.6.1.2.1.31.1.1.1.1
```

```
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.1 = VL1  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.2 = Fa0/1  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.3 = Fa0/2  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.4 = Fa0/3  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.5 = Fa0/4  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.6 = Fa0/5  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.7 = Fa0/6  
...
```

De l'étape 4, l'ifIndex 2 correspond pour mettre en communication Fast Ethernet 0/1 :

```
snmpwalk -c public@1 crumpy .1.3.6.1.2.1.31.1.1.1.1
```

```
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.1 = VL1  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.2 = Fa0/1  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.3 = Fa0/2  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.4 = Fa0/3  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.5 = Fa0/4  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.6 = Fa0/5  
ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.7 = Fa0/6  
...
```

Conclusion

L'adresse MAC 00 00 0C 07 le courant alternatif 08 est apprise sur le port Fa0/1.

Comparez cette conclusion à la sortie de :

- La commande **dynamique de show cam** pour des Commutateurs de CatOS
- La commande de **show mac** pour des Commutateurs de logiciel de Cisco IOS

Voici la sortie témoin :

```
crumpy# show mac
Dynamic Address Count:          58
Secure Address Count:          2
Static Address (User-defined) Count: 0
System Self Address Count:     51
Total MAC addresses:           111
Maximum MAC addresses:         8192
Non-static Address Table:
Destination Address  Address Type  VLAN  Destination Port
-----
0000.0c07.ac08 Dynamic 1 FastEthernet0/1

0001.021b.5091      Dynamic      1 FastEthernet0/1
0001.0348.4d5a      Dynamic      1 FastEthernet0/1
0001.0348.ddbf      Dynamic      1 FastEthernet0/1
0001.972d.dfae      Dynamic      1 FastEthernet0/1
0002.55c6.cfe7      Dynamic      1 FastEthernet0/1
0002.7d61.d400      Dynamic      1 FastEthernet0/1
...
```

Informations connexes

- [Navigateur d'objet SNMP](#)
- [Support et documentation techniques - Cisco Systems](#)