

# Uso de SNMP para encontrar un número de puerto a partir de una dirección MAC en un switch Catalyst

## Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Antecedente](#)

[Detalles de las variables MIB, que incluye los identificadores de objeto \(los OID\)](#)

[Consiga el número del puerto en el cual se ha aprendido una dirección MAC](#)

[Instrucciones Paso a Paso](#)

[Información Relacionada](#)

## [Introducción](#)

Este documento describe cómo utilizar SNMP (Simple Network Management Protocol) para obtener el número de puerto en un switch Cisco Catalyst del cual ya se conoce la dirección MAC.

## [prerrequisitos](#)

### [Requisitos](#)

Quienes lean este documento deben tener conocimiento de los siguientes temas:

- Cómo conseguir los VLA N de un switch de Catalyst con el uso del SNMP
- Cómo utilizar la indexación de cadenas de comunidad con el SNMP
- Uso general del **comando get** y del **comando walk** SNMP

## [Componentes Utilizados](#)

Este documento se aplica a los switches de Catalyst que funcionan con el Catalyst regular OS (CatOS) o el software de Cisco IOS®. Software support el [BRIDGE-MIB](#) y el [IF-MIB](#).

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Catalyst 3524XL que funciona con el Cisco IOS Software Release 12.0(5)WC5a

- Versión 5.0.6 Red-SNMP **Nota:** Para obtener este software, refiera al Red-[SNMP](#) .

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando, asegúrese de haber comprendido el impacto que puede tener cualquier comando.

## [Convenciones](#)

Para obtener más información sobre las convenciones del documento, consulte las [Convenciones de Consejos Técnicos de Cisco](#).

## [Antecedente](#)

Para más información sobre cómo preguntar la tabla de la memoria de contenido direccionable (CAM), los VLA N, y todo el MIB relacionado, tal como el CISCO-VTP-MIB y el BRIDGE-MIB, refieren a la sección del [fondo del](#) documento [cómo conseguir las entradas de CAM dinámicas \(tabla CAM\) para los switches de Catalyst usando el SNMP](#).

## [Detalles de las variables MIB, que incluye los identificadores de objeto \(los OID\)](#)

```
.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 }
```

## [Consiga el número del puerto en el cual se ha aprendido una dirección MAC](#)

### [Instrucciones Paso a Paso](#)

Complete los pasos en esta sección para utilizar el SNMP para conseguir el número del puerto en el cual se ha aprendido una dirección MAC. Considere que el número del puerto está en el VLAN1.

**Nota:** En los comandos en esta sección:

- *el público* es la cadena de comunidad de lectura.
- *@1* es la parte del VLAN1 la cadena de comunidad de lectura.
- *crumpy* es el nombre del host del dispositivo. **Nota:** Usted puede también utilizar la dirección IP para este nombre del host.

**Nota:** La sección de la [conclusión](#) utiliza los valores que aparecen en los *itálicos* en la salida de comando.

1. Extraiga los VLA N. Utilice el **comando snmpwalk** en el objeto del 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) ...
```

**Nota:** Este comando utiliza la [indexación de cadenas de comunidad](#). El comando también utiliza el [vtpVlanState](#), que tiene OID

.1.3.6.1.4.1.9.9.46.1.3.1.1.2. Si usted ha cargado el MIB a su sistema de administración de la red (NMS), usted puede utilizar el nombre del objeto en vez del OID. Publique este comando en lugar de otro:

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

**Nota:** Usted puede también utilizar los nombres del objeto en los pasos 2 a 6.

2. Publique este comando para obtener la tabla de la dirección MAC considerando que el puerto pertenece al 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 ...
```

**Nota:** Proporcione el número VLAN apropiado después de la cadena de comunidad. En este ejemplo, es VLAN1. Las listas de comandos todas las direcciones MAC que se han aprendido en todos los puertos que pertenecen al VLAN1.

3. Publique este comando de determinar el número de puerto del Bridge para el VLAN1:

```
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 ...
```

**Nota:** El VLAN1 es [dot1dTpFdbPort](#), o .1.3.6.1.2.1.17.4.3.1.2.

4. Publique este comando de asociar el puerto de Bridge al [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
```

Este comando pregunta el [dot1dBasePortIfIndex](#), que tiene OID .1.3.6.1.2.1.17.1.4.1.2.

5. Utilice el **comando walk** con el [ifName](#) para correlacionar el valor del [ifIndex](#) con un nombre del puerto correcto. Ejecutar este comando: **Nota:** [El ifName](#) tiene 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. Conecte una dirección MAC al puerto en el cual el direccionamiento era docto. A partir del paso 1, la dirección MAC es: 17.4.3.1.1.0.0.12.7.172.8 = Hex: 00 00 0C 07 AC 08 Del paso 2, el puerto de Bridge dice que la dirección MAC pertenece al número de puerto del Bridge 13: 17.4.3.1.2.0.0.12.7.172.8 = 13 Del paso 3, el número de puerto del Bridge 13 tiene ifIndex número 2: 17.1.4.1.2.13 = 2 Desde el Paso 4, el ifIndex 2 corresponde al puerto Fast Ethernet 0/1: ifMIB.ifMIBObjects.ifXTable.ifXEntry.ifName.2 = Fa0/1

## [Conclusión](#)

La dirección MAC 00 00 0C 07 AC 08 se aprende en el Fa0/1 del puerto.

Compare esta conclusión con la salida de:

- El comando **show cam dynamic** para los switches CatOS
- El comando **show mac** para el Switches del Cisco IOS Software

Aquí está la salida de muestra:

```
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 ...
```

## [Información Relacionada](#)

- [Navegador de objeto SNMP](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)