Cisco Nexus 1000V for VMware vSphere Layer 2 Switching Configuration Guide, Release 5.x

First Published: August 13, 2014

Last Modified: November 14, 2014

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

CHAPTER 1

New and Changed Information

This chapter contains the following sections:

New and Changed Information

This section lists new and changed content in this document by software release.

To find additional information about new features or command changes, see the *Cisco Nexus 1000V Release Notes* and *Cisco Nexus 1000V Command Reference*.

Table 1: New and Changed Features for the Cisco Nexus 1000V Layer 2 Switching Configuration Guide

Feature	Description	Changed in Release	Where Documented
Supporting Redundant Routing Protocols	Added support for redundant routing protocols and BPDU Guard.	4.2(1)SV1(5.1)	Supporting Redundant Routing Protocols, on page 57
Network Load Balancing	Added the ability to configure network load balancing for vEthernet.	4.2(1)SV1(5.1)	Configuring Network Load Balancing for vEthernet, on page 51

Feature	Description	Changed in Release	Where Documented
Layer 2 Configuration Limits	Increased configuration limits for active VLANs across all VEMs, MAC addresses over VLANs within a VEM, and MAC addresses per VLAN within a VEM.	4.2(1)SV1(4)	Layer 2 Switching Configuration Limits, on page 69
IGMP link-local group suppression	Added support to enable or disable link-local group suppression.	4.2(1)SV1(4)	Configuring IGMP Snooping, on page 43
clear mac address-table command	Removed address, interface, and port channel options.	4.2(1)SV1(4)	Configuring MAC Address Tables, on page 9
show mac address-table command	Updated **show** command output.	4.2(1)SV1(4)	Configuring MAC Address Tables, on page 9
feature private-vlan command	Added the ability to globally enable the private VLAN feature.	4.2(1)SV1(4)	Configuring Private VLANs, on page 23
Layer 2 Configuration Limits	Added configuration limits for active VLANs across all VEMs, MACs over VLANs within a VEM, PVLANs across all VEMs, and physical trunks per VSM.	4.0(4)SV1(2)	Layer 2 Switching Configuration Limits, on page 69

CHAPTER 2

Overview

This chapter contains the following sections:

Information about Layer 2 Switching

VEM Port Model

The Cisco Nexus 1000V differentiates the following Virtual Ethernet Module (VEM) ports:

- VEM Virtual Ports
- VEM Physical Ports

The following figure shows the VEM view of the network.

VEM Virtual Ports

The virtual side of the VEM maps together the following layers of ports:

Virtual NICs

There are two types of virtual NICs (vNICs). One vNIC represents a network interface on a Virtual Machine (VM), which emulates a physical port for the virtual host. The other vNIC is an internal port used by the hypervisor for management, iSCSI, and other network access. Each of these vNICs maps to a Virtual Ethernet port within the Cisco Nexus 1000V.

Virtual Ethernet Ports

A virtual Ethernet port (vEth) represents a port on the Cisco Nexus 1000V Distributed Virtual Switch. The Cisco Nexus 1000V has a flat space of vEth ports, 1...n. These vEth ports are what the virtual cable plugs into and are moved to the host that the VM is running on. Virtual Ethernet ports are assigned to port profiles.

VEM Physical Ports

The physical side of the VEM includes the following from top to bottom:

Uplink Ports

Each uplink port on the host represents a physical interface.

Ethernet Ports

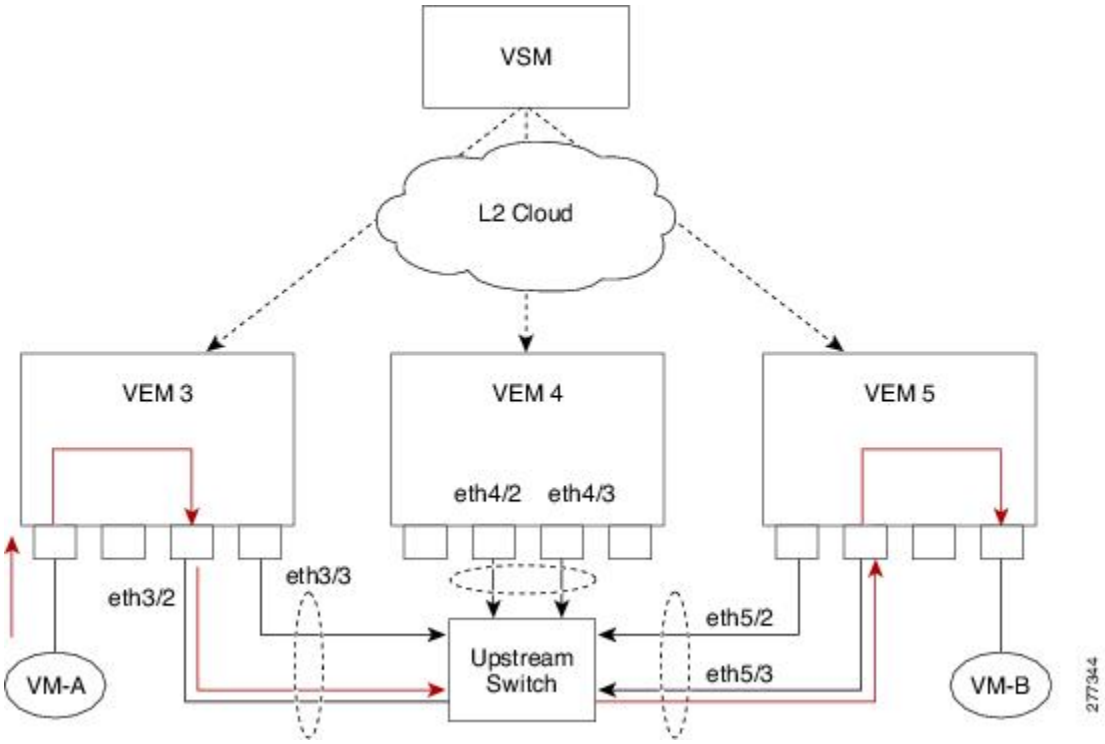
Each physical port that is added to the Cisco Nexus 1000V appears as a physical Ethernet port, just as it would on a hardware-based switch.

Note There is no initial relationship between a physical NIC assigned to the virtual distributed switch and the resulting ethernet port created on the VSM. However, after this is done, the PNIC and ethernet port mapping remains consistent as long at the port continues to be mapped to the virtual distributed switch, even when the physical host is rebooted.

VSM Port Model

The following figure shows the VSM view of the network.

Figure 1: VSM View



The Virtual Supervisor Module (VSM) has the following ports or interfaces:

Virtual Ethernet Interfaces

Virtual Ethernet interfaces (vEths) can be associated with any of the following:

- A virtual machine vNIC on the ESX host
- A virtual machine kernel NIC on the ESX host
- A virtual switch interface on an ESX CoS host

Physical Ethernet Interfaces

Physical Ethernet interfaces (Eths) correspond to the physical NICs on the ESX host.

Port Channel Interfaces

The physical NICs of an ESX host can be bundled into a logical interface called a port channel interface.

Switching Traffic Between VEMs

Each VEM that is attached to the VSM forwards traffic to and from the server as an independent and intelligent line card. Each VLAN uses its forwarding table to learn and store MAC addresses for ports that are connected to the VEM.

See the following figure to see how traffic flows between VEMs.

Layer 2 Ethernet Switching

The congestion related to high bandwidth and large numbers of users can be solved by assigning each device (for example, a server) to its own 10-, 100-, 1000-Mbps, or 10-Gigabit collision domain. Because each LAN port connects to a separate Ethernet collision domain, servers in a switched environment realize full bandwidth access.

Full duplex allows two stations to transmit and receive at the same time. 10/100-Mbps Ethernet usually operates in half-duplex mode, so that stations can either receive or transmit but not both. When packets can flow in both directions simultaneously, the effective Ethernet bandwidth doubles. 1/10-Gigabit Ethernet operates in full-duplex mode only.

Each LAN port can connect to a single workstation or server or to another device through which workstations or servers connect to the network.

To reduce signal degradation, each LAN port is considered to be an individual segment. When stations connected to different LAN ports need to communicate, frames are forwarded from one LAN port to the other at wire speed to ensure full bandwidth for each session.

MAC Address Tables

To switch frames between LAN ports efficiently, a MAC address table is maintained. The MAC address of the sending network is associated with the LAN port on which it was received.

VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes of physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any switchport can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a bridge or a router.

All ports, including the management port, are assigned to the default VLAN (VLAN1) when the device first comes up.

You can configure VLANs using the Cisco Nexus 1000V for KVM CLI, OpenStack Horizon Dashboard, or the OpenStack CLI. Although you can continue to configure VLANs using the Cisco Nexus 1000V for KVM CLI, Cisco recommends that you configure VLANs using OpenStack. For information about OpenStack, see the *Cisco Nexus 1000V for KVM Virtual Network Configuration Guide*.

Note You configure VLANs as VM subnets through OpenStack.

You must consistently use OpenStack for all VM network and subnet configuration. If you use *both* OpenStack and the VSM to configure VM networks and subnets, the OpenStack and the VSM configurations can become out-of-sync and result in faulty or inoperable network deployments.

Note Inter-Switch Link (ISL) trunking is not supported on the Cisco Nexus 1000V.

Control VLANs

A control VLAN is used for communication between the VSM and the VEMs within a switch domain. The control interface is the first interface on the VSM.

A control VLAN is used for the following:

- VSM configuration commands to each VEM and their responses.
- VEM notifications to the VSM. For example, a VEM notifies the VSM of the attachment or detachment of ports to the Distributed Virtual Switch (DVS).
- VEM NetFlow exports that are sent to the VSM, where they are forwarded to a NetFlow Collector.
- VSM active to standby synchronization for high availability.

Management VLANs

A management VLAN, which is used for system login and configuration, corresponds to the mgmt0 interface. The mgmt0 interface appears as the mgmt0 port on a Cisco switch and is assigned an IP address (IPv4 or IPv6). Although the management interface is not used to exchange data between the VSM and VEM, it is used to establish and maintain the connection between the VSM and VMware vCenter Server in Layer 2 mode. In Layer 3 mode (default), when the mgmt0 interface (default) is used for Layer 3 connectivity on the VSM, the management interface communicates with the VEMs and the VMware vCenter Server.

The management interface is the second interface on the VSM.

Packet VLANs

Note A packet VLAN is not a component of the Layer 3 control mode. If you are using Layer 3 control mode, you do not need a packet VLAN.

Similar to the control VLAN, a packet VLAN is used for communication between the VSM and the VEMs within a switch domain.

A packet VLAN is used to tunnel network protocol packets between the VSM and the VEMs such as the Cisco Discovery Protocol (CDP), Link Aggregation Control Protocol (LACP), and Internet Group Management Protocol (IGMP).

The packet interface is the third interface on the VSM.

Private VLANs

Private VLANs (PVLANs) are used to segregate Layer 2 ISP traffic and convey it to a single router interface. PVLANs achieve device isolation by applying Layer 2 forwarding constraints that allow end devices to share the same IP subnet while being Layer 2 isolated. The use of larger subnets reduces address management overhead.

IGMP Snooping

The Internet Group Management Protocol (IGMP) snooping software examines Layer 2 IP multicast traffic within a VLAN to discover the ports where interested receivers reside. Using the port information, IGMP snooping can reduce bandwidth consumption in a multi-access LAN environment to avoid flooding the entire VLAN. The IGMP snooping feature tracks which ports are attached to multicast-capable routers to help the routers forward IGMP membership reports. The IGMP snooping software responds to topology change notifications. By default, IGMP snooping is enabled on the device.

CHAPTER 3

Configuring MAC Address Tables

This chapter contains the following sections:

Information About MAC Address Tables

Layer 2 ports correlate the MAC address on a packet with the Layer 2 port information for that packet using the MAC address table. A MAC address table is built using the MAC source addresses of the frames received. When a frame is received for a MAC destination address not listed in the address table, the frame is flooded to all LAN ports of the same VLAN with the exception of the port that received the frame. When the destination station replies, the relevant MAC source addresses and port IDs are added to the address table. Subsequent frames are forwarded to a single LAN port without flooding all LAN ports.

You can configure MAC addresses, which are called static MAC addresses, to statically point to specified interfaces on the device. These static MAC addresses override any dynamically learned MAC addresses on those interfaces. You cannot configure broadcast or multicast addresses as static MAC addresses. The static MAC entries are retained across reboots if you copy the static MAC addresses configuration to the startup configuration by using the copy running-config startup-config command.

The address table per VEM can store up to 32,000 MAC entries. An aging timer triggers removal of addresses from the table when they remain inactive for the default time of 300 seconds. The aging timer can be configured on a global basis but not per VLAN.

You can configure the length of time an entry remains in the MAC address table, clear the table, and so forth.

Guidelines and Limitations

- The forwarding table for each VLAN in a VEM can store up to 4096 MAC addresses.
- You can configure only 32 static MAC addresses on a single interface and 1024 static MAC addresses on a DVS.
- The Cisco Nexus 1000V supports a maximum of 2000 private VLAN MAC addresses on a VSM.

Default Settings

Table 2: Default MAC Address Aging Time

Parameters	Default
Aging time	1800 seconds

Configuring the MAC Address Table

Configuring a Static MAC Address

You can configure a MAC address to statically point to a specific interface.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that you cannot configure broadcast or multicast addresses as static MAC addresses.
- Know that static MAC addresses override dynamically learned MAC addresses on an interface.

Note Be aware that the Cisco NX-OS commands may differ from those commands used in Cisco IOS.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac address-table static** *mac_address* **vlan** *vlan-id* {[**drop** | **interface** {**type***if_id*} | **port-channel number**]}	Adds a static MAC address in the Layer 2 MAC address table and saves it in the running configuration.

	Command or Action	Purpose
		The interface can be specified as either of the following: • ethernet *slot/port* • veth *number*
Step 3	switch(config)# **show mac address static interface** [**type** *if_id*]	(Optional) Displays static MAC addresses.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a static MAC address:

```
switch# configure terminal
switch(config)# mac address-table static
switch(config)# show mac address static interface12ab.47dd.ff89 vlan 3
interface ethernet 3/3

VLAN      MAC Address       Type    Age       Port                           Mod
---------+-----------------+-------+---------+------------------------------+---------
1         0002.3d11.5502    static  0         N1KV Internal Port             3
1         0002.3d21.5500    static  0         N1KV Internal Port             3
1         0002.3d21.5502    static  0         N1KV Internal Port             3
1         0002.3d31.5502    static  0         N1KV Internal Port             3
1         0002.3d41.5502    static  0         N1KV Internal Port             3
1         0002.3d61.5500    static  0         N1KV Internal Port             3
1         0002.3d61.5502    static  0         N1KV Internal Port             3
1         0002.3d81.5502    static  0         N1KV Internal Port             3
3         12ab.47dd.ff89    static  0         Eth3/3                         3
342       0002.3d41.5502    static  0         N1KV Internal Port             3
343       0002.3d21.5502    static  0         N1KV Internal Port             3
Total MAC Addresses: 11
n1000v(config)# show mac address static interface Ethernet 3/3
VLAN      MAC Address       Type    Age       Port                           Module
---------+-----------------+-------+---------+------------------------------+---------
3         12ab.47dd.ff89    static  0         Eth3/3                         3
Total MAC Addresses: 1
switch(config)#
```

Configuring the Aging Time

You can configure the amount of time that packet source MAC addresses, and the ports on which they are learned, remain in the MAC table.

Note The aging time is a global setting that cannot be configured per VLAN. Although it is a global setting, you can also configure the MAC aging time in interface configuration mode or VLAN configuration mode.

Before You Begin

Log in to the CLI in EXEC mode.

Note Be aware that the Cisco NX-OS commands may differ from those commands used in Cisco IOS.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch# **mac address-table aging-time** *seconds*	Specifies and saves in the running configuration the amount of time that will elapse before an entry in the Layer 2 MAC address table is discarded. Allowable entries are as follows: • 120 to 918000 seconds (default is 300) • If you specify zero (0), MAC aging is disabled.

This example shows how to configure the aging time:

```
switch# configure terminal
switch(config)# mac address-table aging-time 600
switch(config)# show mac address-table aging-time
Vlan  Aging Time
----- ----------
101   300
100   300
1     300
switch#
```

Clearing Dynamic Addresses from the MAC Address Table

Before You Begin

Log in to the CLI in EXEC mode.

Note Be aware that the Cisco NX-OS commands may differ from those commands used in Cisco IOS.

Procedure

	Command or Action	Purpose
Step 1	switch# **clear mac address-table dynamic** [**vlan** *vlan_id*]	Clears the dynamic address entries from the Layer 2 MAC address table.

	Command or Action	Purpose
Step 2	switch# **show mac address-table**	(Optional) Displays the MAC address table.

This example shows how to clear the entire MAC address table of all dynamic entries:

```
switch# clear mac address-table dynamic
switch#
```

This example shows how to clear the MAC address table of only those dynamic MAC addresses learned on VLAN 5:

```
switch# clear mac address-table dynamic vlan 5
switch#
```

Verifying the MAC Address Table Configuration

Use the following commands to verify the configuration:

Command	Purpose
show mac address-table	Displays the MAC address table.
show mac address-table module	Displays information about specific module a specific module.
show mac address-table static	Displays information about the MAC address table static entries.
show mac address-table static |inc veth	Displays the static MAC address of vEthernet interfaces in case a VEM physical port learns a dynamic MAC and the packet source is in another VEM on the same VSM.
show mac address static interface [**type** *if_id*]	Displays all static MAC addresses.
show mac address-table aging-time	Displays the aging time in the MAC address table.
show mac address-table count	Displays a count of MAC address entries.
show interface *interface_id* **mac**	Displays the MAC addresses and the burned-in MAC address for an interface.

Configuration Example for MAC Address Tables

This example shows how to add a static MAC address and establish a global aging time:

```
switch# configure terminal
switch(config)# mac address-table static 0000.0000.1234 vlan 10 interface ethernet 2/15
switch(config)# mac address-table aging-time 120
switch(config)#
```

Feature History for MAC Address Tables

Feature Name	Feature Name	Releases
MAC Address Tables	4.0(4)SV1(1)	This feature was introduced.

CHAPTER 4

Configuring VLANs

This chapter contains the following sections:

Information About VLANs

vEthernet interfaces that are assigned to specific VLANs are tagged with the VLAN when transmitted. A vEthernet interface that is not assigned to a specific VLAN, or assigned to VLAN 0, is transmitted as untagged on the physical NIC interfaces. When the VLAN is not specified, it is assumed to be 1.

The following table summarizes the actions taken on packets that are received by the Virtual Ethernet Module (VEM) based on VLAN tagging.

Table 3: VEM Action on VLAN Tagging

Port Type	Packet received	Action
Access	Tagged	The packet is dropped.
Access	Untagged	The VEM adds an access VLAN to the packet.
Trunk	Tagged	No action is taken on the packet.
Trunk	Untagged	The VEM adds a native VLAN tag to the packet.

Guidelines and Limitations

In accordance with the IEEE 802.1Q standard, up to 4094 VLANs (from 1 to 4094) are supported in the Cisco Nexus 1000V, and are listed in the following table.

Note For VLAN configuration limits, see Layer 2 Switching Configuration Limits, on page 69.

Table 4: Cisco Nexus 1000V VLAN Numbering

VLAN Numbers	Range	Usage
1	Normal	Cisco Nexus 1000V default. You can use this VLAN, but you cannot modify or delete it.
2 to1005	Normal	You can create, use, modify, or delete these VLANs.
1006 to 4094	Extended	You can create, name, or use these VLANs. You cannot change the following parameters: • The state is always active. • These VLANs are always enabled. You cannot shut down these VLANs. The extended system ID is always automatically enabled.
3968 to 4047 and 4094	Internally allocated	You cannot use, create, delete, or modify these VLANs. You can display these VLANs. The Cisco Nexus 1000V allocates these 80 VLANs, plus VLAN 4094, for features, like diagnostics, that use internal VLANs for their operation.

Default Settings

Table 5: Default VLAN Settings

Parameters	Default
VLAN assignment for all interfaces and all ports configured as switchports	VLAN 1
VLAN name	VLANxxxx where xxxx represent four numeric digits (including leading zeroes) equal to the VLAN ID number
Shut state	No shutdown
Operational state	Active
External Switch Tagging (EST)	Enabled
Physical ports	Trunk ports
IGMP snooping	Enabled

Configuring a VLAN

Creating a VLAN

You can do one of the following:

- Create a single VLAN that does not already exist.
- Create a range of VLANs that does not already exist.
- Delete an existing VLAN.

Note All interfaces and all ports configured as switchports are in VLAN 1 by default.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that VLAN characteristics are configured in the VLAN configuration mode. To configure a VLAN that is already created, see Configuring VLAN Characteristics, on page 19.
- Be familiar with the VLAN numbering in the Guidelines and Limitations, on page 16.

- Know that newly created VLANs remain unused until Layer 2 ports are assigned to them.
- Know that when you delete a specified VLAN, the ports associated to that VLAN are shut down and no traffic flows. When you delete a specified VLAN from a trunk port, only that VLAN is shut down and traffic continues to flow on all the other VLANs through the trunk port. However, the system retains all the VLAN-to-port mapping for that VLAN, and when you reenable, or re-create, that specified VLAN, the system automatically reinstates all the original ports to that VLAN. Note that the static MAC addresses and aging time for that VLAN are not restored when the VLAN is reenabled.

Note Be aware that the Cisco NX-OS commands may differ from those commands used in Cisco IOS.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **show vlan**	Displays the VLANs that already exist.
Step 3	switch(config)# [**no**] **vlan** {*vlan-id* | *vlan-range*}	Creates or deletes, and saves in the running configuration, a VLAN or a range or VLANs. To configure the VLAN, see Configuring VLAN Characteristics, on page 19. **Note** If you enter a VLAN ID that is assigned to an internally allocated VLAN, the system returns an error message. From the VLAN configuration mode, you can also create and delete VLANs. For information about Assigning Layer 2 interfaces to VLANs (access or trunk ports), see the *Cisco Nexus 1000V Interface Configuration Guide*. For information about configuring ports as VLAN access or trunk ports and assigning ports to VLANs, see the *Cisco Nexus 1000V Interface Configuration Guide*.
Step 4	switch(config-vlan)# **show vlan id** *vlan-id*	(Optional) Displays the VLAN configuration.
Step 5	switch(config-vlan)# **copy running-config startup-config**	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

In this example, VLAN 5 is created and you are automatically placed into the VLAN configuration mode for VLAN 5:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)#
```

This example shows the range, VLAN 15 to 20, being created. The VLANs in the range are activated, and you are automatically placed into VLAN configuration mode for VLANs 15 to 20.

Note If you create a range of VLANs that includes an unusable VLAN, all VLANs in the range are created except those that are unusable; and Cisco Nexus 1000V returns a message listing the failed VLANs.

```
switch# configure terminal
switch(config)# vlan 15-20
switch(config-vlan)#
```

This example shows how to delete VLAN 3967:

```
switch# configure terminal
switch(config)# no vlan 3967
switch(config)#
```

This example shows how to display the VLAN 5 configuration:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# show vlan id 5

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
5    VLAN0005                         active

VLAN Type
---- -----
5    enet

Remote SPAN VLAN
----------------
Disabled

Primary  Secondary  Type             Ports
-------  ---------  ---------------  ------------------------------------------

n1000v(config-vlan)# copy run start
[########################################] 100%
n1000v(config)#
```

Configuring VLAN Characteristics

You can do the following for a VLAN that has already been created:

Note Commands entered in the VLAN configuration mode are immediately saved to the running configuration.

- Name the VLAN.
- Configure the operational state (active or suspend) of the VLAN.
- Configure the VLAN media type (Ethernet).
- Shut down switching on the VLAN.

Before You Begin

Log in to the CLI in EXEC mode.

Note Some characteristics cannot be modified on some VLANs. For more information, see the VLAN numbering described in the Guidelines and Limitations, on page 16.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** {*vlan-id* | *vlan-range*}	Enters VLAN configuration mode for the specified VLAN. **Note** If the VLAN does not already exist, the system creates it and then enters the VLAN configuration mode for that VLAN.
Step 3	switch(config-vlan)# **name** *vlan-name*	Adds a name to the VLAN of up to 32 alphanumeric characters. • You cannot change the name of VLAN1 or the VLANs that are reserved for internal use. • The default name is VLANxxxx where xxxx represent four numeric digits (including leading zeroes) equal to the VLAN ID number.
Step 4	switch(config-vlan)# **state** {**active** | **suspend**}	Changes the operational state of the VLAN and saves it in the running configuration. Allowable entries are as follows: • **active** (default) • **suspend** While the VLAN state is suspended, the ports associated with this VLAN are shut down, and that VLAN does not pass any traffic. **Note** You cannot suspend the state for the default VLAN or VLANs 1006 to 4094.
Step 5	switch(config-vlan)# **no shutdown**	Enables VLAN switching in the running configuration. Allowable entries are as follows: • **no shutdown** (default) • **shutdown** **Note** You cannot shut down the default VLAN, VLAN1, or VLANs 1006 to 4094.
Step 6	switch(config-vlan)# **exit**	Exits VLAN configuration mode. **Note** You must exit VLAN configuration mode for the configurations to take effect.

	Command or Action	Purpose
Step 7	switch(config)# **show vlan** [**id** *vlan-id*]	(Optional) Displays the VLAN configuration.
Step 8	switch(config)# **copy running-config startup-config**	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

This example shows how to configure VLAN characteristics:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# name accounting
switch(config-vlan)# state active
switch(config-vlan)# no shutdown
switch(config-vlan)# exit
switch(config)# show vlan brief

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Eth2/1, Eth2/2, Eth2/3, Eth2/5
                                                Eth2/7, Eth2/8, Eth2/9, Eth2/10
                                                Eth2/15, Eth2/21, Eth2/22
                                                Eth2/23, Eth2/24, Eth2/25
                                                Eth2/46, Eth2/47, Eth2/48
5    accounting                       active
6    VLAN0006                         active
7    VLAN0007                         active
8    test                             active
9    VLAN0009                         active
10   VLAN0010                         active
50   VLAN0050                         active    Eth2/6
100  trunked                          active
200  VLAN0200                         active
201  VLAN0201                         active
202  VLAN0202                         active
3966 VLAN3966                         active
switch(config)#
```

Verifying the Configuration

Use the following commands to verify the configuration:

Command	Purpose
show running-config vlan *vlan-id*	Displays VLAN information in the running configuration.
show vlan [**all-ports** | **brief** | **id** *vlan-id* | **name** *name* | **dot1q tag native**]	Displays the specified VLAN information.
show vlan summary	Displays a summary of VLAN information.

Feature History for VLANs

Feature Name	Feature Name	Releases
VLANs	4.0(4)SV1(1)	This feature was introduced.

CHAPTER 5

Configuring Private VLANs

This chapter contains the following sections:

Information About Private VLANs

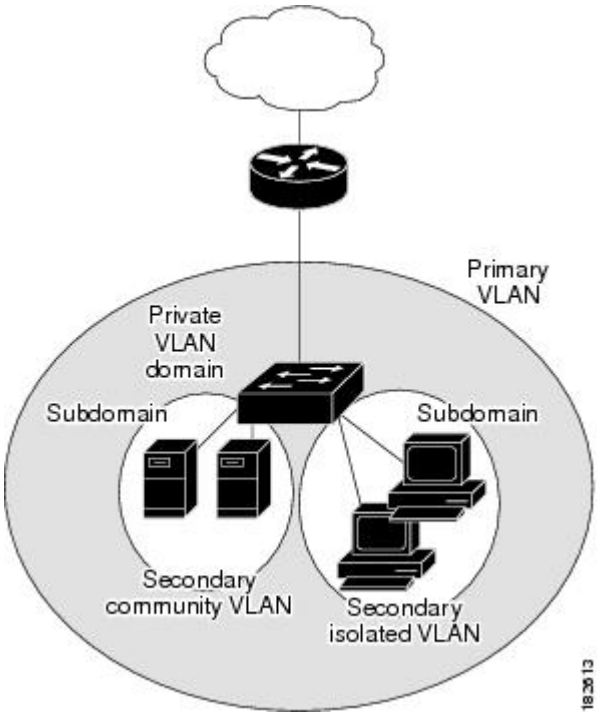
PVLANs achieve device isolation through the use of three separate port designations, each having its own unique set of rules that regulate each connected endpoint's ability to communicate with other connected endpoints within the same private VLAN domain.

Private VLAN Domains

A PVLAN domain consists of one or more pairs of VLANs. The primary VLAN makes up the domain; and each VLAN pair makes up a subdomain. The VLANs in a pair are called the primary VLAN and the secondary

VLAN. All VLAN pairs within a private VLAN have the same primary VLAN. The secondary VLAN ID is what differentiates one subdomain from another. See the following figure.

Figure 2: Private VLAN Domain



Spanning Multiple Switches

PVLANs can span multiple switches, just like regular VLANs. Inter-switch link ports do not need to be aware of the special VLAN type and carry frames tagged with these VLANs just like they do any other frames. PVLANs ensure that traffic from an isolated port in one switch does not reach another isolated or community port in a different switch even after traversing an inter-switch link. By embedding the isolation information at the VLAN level and by transporting it with the packet, it is possible to maintain consistent behavior throughout the network. The mechanism that restricts Layer 2 communication between two isolated ports in the same switch also restricts Layer 2 communication between two isolated ports in two different switches.

Private VLAN Ports

Within a PVLAN domain, there are three separate port designations. Each port designation has its own unique set of rules that regulate the ability of one endpoint to communicate with other connected endpoints within the same private VLAN domain. The three port designations are as follows:

- promiscuous
- isolated
- community

Primary VLANs and Promiscuous Ports

The primary VLAN encompasses the entire PVLAN domain. It is a part of each subdomain and provides the Layer 3 gateway out of the VLAN. A PVLAN domain has only one primary VLAN. Every port in a PVLAN domain is a member of the primary VLAN. The primary VLAN is the entire PVLAN domain.

A promiscuous port can talk to all other types of ports; it can talk to isolated ports as well as community ports and vice versa. Layer 3 gateways, DHCP servers, and other trusted devices that need to communicate with the customer endpoints are typically connected with a promiscuous port. A promiscuous port can be either an access port or a hybrid/trunk port according to the terminology presented in Annex D of the IEEE 802.1Q specification.

Secondary VLANs and Host Ports

Secondary VLANs provide Layer 2 isolation between ports in a PVLAN domain. A PVLAN domain can have one or more subdomains. A subdomain is made up of a VLAN pair that consists of the primary VLAN and a secondary VLAN. Because the primary VLAN is a part of every subdomain, secondary VLANs differentiate the VLAN subdomains.

To communicate to the Layer 3 interface, you must associate a secondary VLAN with at least one of the promiscuous ports in the primary VLAN. You can associate a secondary VLAN to more than one promiscuous port within the same PVLAN domain, for example, if needed for load balancing or redundancy. A secondary VLAN that is not associated with any promiscuous port cannot communicate with the Layer 3 interface.

A secondary VLAN can be one of the following types:

- Isolated VLANs—Isolated VLANs use isolated host ports. An isolated port cannot talk to any other port in that private VLAN domain except for promiscuous ports. If a device needs to have access only to a gateway router, it should be attached to an isolated port. An isolated port is typically an access port, but in certain applications, it can also be a hybrid or trunk port.

  An isolated VLAN allows all its ports to have the same degree of segregation that could be obtained from using one separate dedicated VLAN per port. Only two VLAN identifiers are used to provide this port isolation.

  Note While multiple community VLANs can be in a private VLAN domain, one isolated VLAN can serve multiple customers. All endpoints that are connected to its ports are isolated at Layer 2. Service providers can assign multiple customers to the same isolated VLAN and be assured that their Layer 2 traffic cannot be sniffed by other customers that share the same isolated VLAN.

- Community VLANs—Community VLANs use community host ports. A community port (c1 or c2 in the above figure) is part of a group of ports. The ports within a community can communicate at Layer 2 with one another and can also talk to any promiscuous port. For example, if an ISP customer has four devices and wants them isolated from those devices of other customers but still be able to communicate among themselves, community ports should be used.

  Note Because trunks can support a VLAN that carries traffic between its ports, VLAN traffic can enter or leave the device through a trunk interface.

Communication Between Private VLAN Ports

The following table shows how access is permitted or denied between PVLAN port types.

Table 6: Communication Between PVLAN Ports

	Isolated	Promiscuous	Community 1	Community 2	Interswitch Link Port[1]
Isolated	Deny	Permit	Deny	Deny	Permit
Promiscuous	Permit	Permit	Permit	Permit	Permit
Community 1	Deny	Permit	Permit	Deny	Permit
Community 2	Deny	Permit	Deny	Permit	Permit
Interswitch Link Port	Deny[2]	Permit	Permit	Permit	Permit

1 An interswitch link port is a regular port that connects two switches and that happens to carry two or more VLANs.

2 This behavior applies to traffic that traverses inter-switch link ports over an isolated VLAN only. Traffic from an inter-switch link port to an isolated port will be denied if it is in the isolated VLAN. Traffic from an inter-switch link port to an isolated port will be permitted if it is in the primary VLAN.

Guidelines and Limitations

PVLANs have the following configuration guidelines and limitations:

Control VLANs, packet VLANs, and management VLANs must be configured as regular VLANs and not as private VLANs.

The following are configuration limits:

- Private VLANs per DVS: 512 maximum
- Primary VLANs per promiscuous trunk port: 64 maximum
- Private VLAN associations: 511 maximum
- Private VLAN ports per DVS : 4096 maximum

Default Settings

Table 7: Default PVLAN Settings

Parameters	Default
PVLANs	Disabled

Configuring a Private VLAN

The following section guides you through the private VLAN configuration process. After completing each procedure, return to this section to make sure that you have completed all required procedures in the correct sequence.

Procedure

Step 1 Enable or disable the PVLAN feature globally. See Enabling or Disabling the Private VLAN Feature Globally, on page 27.

Step 2 Configure a VLAN as a primary VLAN. See Configuring a VLAN as a Primary VLAN, on page 28.

Step 3 Configure a VLAN as a secondary VLAN. See Configuring a VLAN as a Secondary VLAN.

Step 4 Associate the VLANs in a PVLAN. See Associating the VLANs in a PVLAN.

Step 5 Configure a PVLAN host port. See Configuring a Private VLAN Host Port.

Step 6 Associate a host port with a PVLAN. See Associating a vEthernet Port Profile with a Private VLAN.

Step 7 Verify a PVLAN configuration. See Verifying a Private VLAN Configuration.

Enabling or Disabling the Private VLAN Feature Globally

You can globally enable or disable the PVLAN feature.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **feature private-vlan**	Globally enables or disables the PVLAN feature.
Step 3	switch(config-vlan)# **show feature**	(Optional) Displays features available and whether they are enabled globally.
Step 4	switch(config-vlan)# **copy running-config startup-config**	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

This example shows how to enable or disable the PVLAN feature globally:

```
switch# configure terminal
switch(config)# feature private-vlan
```

```
switch(config-vlan)# show feature
Feature Name          Instance  State
--------------------  --------  --------
dhcp-snooping         1         enabled
http-server           1         enabled
ippool                1         enabled
lacp                  1         enabled
lisp                  1         enabled
lisphelper            1         enabled
netflow               1         disabled
port-profile-roles    1         enabled
private-vlan          1         enabled
sshServer             1         enabled
tacacs                1         enabled
telnetServer          1         enabled
switch(config-vlan)#
```

Configuring a VLAN as a Primary VLAN

You can configure a VLAN to function as the primary VLAN in a PVLAN.

Before You Begin

- Log in to the CLI in EXEC mode.
- You have already enabled the private VLAN feature using the Enabling or Disabling the Private VLAN Feature Globally, on page 27.
- Know that the VLAN that you are configuring as a primary VLAN already exists in the system as a normal VLAN, and you know the VLAN ID.

Note If the VLAN does not already exist, you are prompted to create it when you create the primary VLAN. For information about creating a VLAN, see Creating a VLAN.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** *primary-vlan-id*	Enters VLAN configuration mode for the specified VLAN and configures the primary VLAN ID in the running configuration.
Step 3	switch(config-vlan)# **private-vlan primary**	Designates the primary VLAN as a private VLAN in the running configuration.
Step 4	switch(config-vlan)# **exit**	Exits VLAN configuration mode. **Note** You must exit VLAN configuration mode for the configurations to take effect.
Step 5	switch(config)# **show vlan private-vlan**	(Optional) Displays the PVLAN configuration.

	Command or Action	Purpose
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

This example shows how to configure a VLAN as a primary VLAN:

```
switch# configure terminal
switch(config)# vlan 202
switch(config-vlan)# private-vlan primary
n1000v(config-vlan)# exit
switch(config)# show vlan private-vlan
Primary  Secondary  Type             Ports
-------  ---------  ---------------  ------------------------------------------
202                 primary

switch(config)#
```

Configuring a VLAN as a Secondary VLAN

You can configure a VLAN to function as the primary VLAN in a PVLAN.

Before You Begin

- Log in to the CLI in EXEC mode.
- You have already enabled the private VLAN feature using the Enabling or Disabling the Private VLAN Feature Globally.
- Know that the VLAN that you are configuring as a secondary VLAN already exists in the system as a normal VLAN, and you know the VLAN ID.

Note If the VLAN does not already exist, you are prompted to create it when you create the secondary VLAN. For information about creating a VLAN, see Creating a VLAN.

- Know whether you want the secondary VLANs to be community VLANs or isolated VLANs, and the VLAN IDs for each.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** *secondary-vlan-id*	Enters VLAN configuration mode for the specified VLAN and configures the secondary VLAN ID in the running configuration.

	Command or Action	Purpose
Step 3	switch(config-vlan)# **private-vlan** {**community** | **isolated**}	Designates the VLAN as either a community or isolated private VLAN in the running configuration.
Step 4	switch(config-vlan)# **exit**	Exits VLAN configuration mode. **Note** You must exit VLAN configuration mode for the configurations to take effect.
Step 5	switch(config)# **show vlan private-vlan**	(Optional) Displays the PVLAN configuration.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

This example shows how to configure a VLAN as a secondary VLAN:

```
switch# configure terminal
switch(config)# vlan 202
switch(config-vlan)# private-vlan community
switch(config-vlan)# exit
switch(config)# show vlan private-vlan
Primary  Secondary  Type             Ports
-------  ---------  ---------------  ------------------------------------------
202                 community

switch(config)#
```

Associating the VLANs in a PVLAN

You can associate the primary VLANs in a PVLAN with the secondary VLANs.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that the primary VLAN for this PVLAN is already configured as a PVLAN.
- Know that the secondary VLANs for this PVLAN are already configured as PVLANs.
- Know the VLAN IDs for each VLAN that is a part of the PVLAN.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **vlan** *primary-vlan-id*	Enters VLAN configuration mode and associates the VLANs to function as a PVLAN in the running configuration.
Step 3	switch(config-vlan)# **private-vlan association** {**add** | **remove**} *secondary vlan-id*	Associates a specified secondary VLAN with the primary VLAN to function as a PVLAN in the running configuration. To associate additional secondary VLANs, repeat this step.
Step 4	switch(config-vlan)# **exit**	Exits VLAN configuration mode.
Step 5	switch(config)# **show vlan private-vlan**	(Optional) Displays the PVLAN configuration.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

This example shows how to associate VLANs in a PVLAN:

```
switch# configure terminal
switch(config)# vlan 202
switch(config-vlan)# private-vlan association add 303
switch(config-vlan)# exit
switch(config)# show vlan private-vlan
Primary  Secondary  Type             Ports
-------  ---------  ---------------  ------------------------------------------
202      303        community        Veth1
switch(config)#
```

Configuring a Private VLAN Host Port

You can configure an interface as a host port to function with a PVLAN.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that the primary VLAN for this PVLAN is already configured as a PVLAN.
- Know that the secondary VLANs for this PVLAN are already configured as PVLANs.
- Know that the secondary VLANs are already associated with the primary VLAN.
- Know the name of the interface to be used with the PVLAN as a host port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface type** *if_id*	Enters interface configuration mode and creates a the named interface if it does not exist.
Step 3	switch(config-if)# **switchport mode private-vlan host**	Designates that the physical interface is to function as a PVLAN host port in the running configuration.
Step 4	switch(config-if)# **show interface type** *if_id*	(Optional) Displays the interface configuration.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a PVLAN host port:

```
switch# configure terminal
switch(config)# interface veth1
switch(config-if)# switchport mode private-vlan host
switch(config-if)# show interface veth1
Vethernet1 is up
    Hardware is Virtual, address is 0050.56b0.34c8
    Owner is VM "HAM61-RH5-32bit-ENVM-7.60.1.3"
    Active on module 2, host VISOR-HAM61.localdomain 0
    VMware DVS port 16777215
    Port-Profile is vlan631
    Port mode is Private-vlan host
    Rx
    48600 Input Packets 34419 Unicast Packets
    0 Multicast Packets 14181 Broadcast Packets
    4223732 Bytes
    Tx
    34381 Output Packets 34359 Unicast Packets
    22 Multicast Packets 0 Broadcast Packets 0 Flood Packets
    3368196 Bytes
    5 Input Packet Drops 11 Output Packet Drops

switch(config-if)#
```

Associating a vEthernet Port Profile with a Private VLAN

You can associate the vEthernet port profile with the primary and secondary VLANs in a PVLAN.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know the VLAN IDs of the primary and secondary VLANs in the PVLAN.
- Know that the primary VLAN for this PVLAN is already configured as a PVLAN.
- Know that the secondary VLANs for this PVLAN are already configured as PVLANs.

- Know the name of the interface functioning in the PVLAN as a host port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile type vethernet** *name*	Enters port profile configuration mode for the specified port profile.
Step 3	switch(config-port-profile)# **switchport mode private-vlan host**	Associates the vEthernet port with the PVLAN configuration. The port profile is associated with the VLANs in the PVLAN.
Step 4	switch(config-port-profile)# **switchport private-vlan host-association** *vlan_ids*	Assigns the primary and secondary VLAN IDs to the port profile and saves this association in the running configuration.
Step 5	switch(config-port-profile)# **no shut**	Enables the port profile.
Step 6	switch(config-port-profile)# **vmware port-group**	Designates the port profile as a VMware port group. The port profile is mapped to a VMware port group of the same name unless you specify a name here. When you connect the VSM to vCenter Server, the port group is distributed to the virtual switch on vCenter Server.
Step 7	switch(config-port-profile)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 8	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to associate a vEthernet port with a PVLAN:

```
switch # configure terminal
switch(config)# port-profile type vethernet vlan_private_isolated_127
switch(config-port-prof)# switchport mode private-vlan host
switch(config-port-prof)# switchport private-vlan host-association 126 127
switch(config-port-prof)# no shut
switch(config-port-prof)# vmware port-group
switch(config-port-prof)# state enabled
```

Configuring a Layer 2 Port Profile as a Promiscuous Trunk Port

You can configure a Layer 2 interface as a promiscuous trunk port that does the following:

- Combines multiple promiscuous ports into a single trunk port.

- Carries all normal VLANs.
- Carries multiple PVLAN primary VLANs each with selected secondary VLANs.

Note A promiscuous port can be either access or trunk. If you have one primary VLAN, you can use a promiscuous access port. If you have multiple primary VLANs, you can use a promiscuous trunk port.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know that the **private-vlan mapping trunk** command does not decide or override the trunk configuration of a port.
- Know that the port is already configured in a regular trunk mode before adding the PVLAN trunk configurations.
- Know that primary VLANs must be added to the list of allowed VLAN for the promiscuous trunk port.
- Know that secondary VLANs are not configured in the allowed VLAN list.
- Know that the trunk port can carry normal VLANs in addition to primary VLANs.
- Know that you can map up to 64 primary VLANs to their secondary VLANs in one promiscuous trunk port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile type ethernet** *name*	Places you in port-profile mode.
Step 3	switch(config-port-prof)# **switchport mode trunk**	Designates that the interfaces are to be used as trunking ports.
Step 4	switch(config-port-prof)# **switchport mode private-vlan trunk promiscuous**	In the running configuration, designates the interface as a promiscuous PVLAN trunk port.
Step 5	switch(config-port-prof)# **switchport private-vlan trunk allowed vlan** *vlan_range*	Sets the allowed VLANs and VLAN IDs when the interface is in PVLAN trunking mode.
Step 6	switch(config-port-prof)# **switchport private-vlan mapping trunk** *primary_vlan_ID* {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*}	Maps the PVLAN trunk port to a primary VLAN and to selected secondary VLANs in the running configuration. Multiple PVLAN pairs can be specified so that a promiscuous trunk port can carry multiple primary VLANs.

	Command or Action	Purpose
Step 7	switch(config-port-prof)# **no shut**	Enables the port profile.
Step 8	switch(config-port-profile)# **vmware port-group**	Designates the port profile as a VMware port group. The port profile is mapped to a VMware port group of the same name unless you specify a name here. When you connect the VSM to vCenter Server, the port group is distributed to the virtual switch on the vCenter Server.
Step 9	switch(config-port-profile)# **state enabled**	Enables the port profile and applies its configuration to the assigned ports.
Step 10	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a Layer 2 port profile as a promiscuous trunk port:

```
switch # configure terminal
switch(config)# port-profile type eth allaccess1
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport mode private-vlan trunk promiscuous
 switch(config-port-prof)# switchport private-vlan trunk allowed vlan 2,126-128,150-155
switch(config-port-prof)# switchport private-vlan mapping trunk 126 127,128
switch(config-port-prof)# no shut
switch(config-port-prof)# vmware port-group
switch(config-port-prof)# state enabled
```

Configuring a Private VLAN Promiscuous Access Port

You can configure a port to be used as a promiscuous access port in a PVLAN.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know the name of the interface that will function as a promiscuous access port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type* [*slot/port* | *number*]	Enters interface configuration mode for the specified interface.

	Command or Action	Purpose
Step 3	switch(config-if)# **switchport mode private-vlan promiscuous**	Designates that the interface is to function as a promiscuous access port for a PVLAN in the running configuration.
Step 4	switch(config-if)# **show interface** *type* [*slot/port* | *number*]	(Optional) Displays the interface configuration.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a PVLAN promiscuous access port:

```
switch# configure terminal
switch(config)# interface eth3/2
switch(config-if)# switchport mode private-vlan promiscuous
switch(config-if)# show interface eth3/2
Ethernet3/2 is up
    Hardware is Ethernet, address is 0050.5655.2e85 (bia 0050.5655.2e85)
    MTU 1500 bytes, BW -1942729464 Kbit, DLY 10 usec,
       reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation ARPA
    Port mode is promiscuous
    full-duplex, 1000 Mb/s
    Beacon is turned off
    Auto-Negotiation is turned on
    Input flow-control is off, output flow-control is off
    Rx
    276842 Input Packets 100419 Unicast Packets
    138567 Multicast Packets 37856 Broadcast Packets
    25812138 Bytes
    Tx
    128154 Output Packets 100586 Unicast Packets
    1023 Multicast Packets 26545 Broadcast Packets 26582 Flood Packets
    11630220 Bytes
    173005 Input Packet Drops 37 Output Packet Drops

switch(config-if)#
switch# configure terminal
switch(config)# interface vethernet1
n1000v(config-if)# switchport mode private-vlan promiscuous
switch(config-if)# show interface vethernet 1
Vethernet1 is up
  Port description is VM-1, Network Adapter 7
  Hardware: Virtual, address: 0050.569e.009f (bia 0050.569e.009f)
  Owner is VM "VM-1", adapter is Network Adapter 7
  Active on module 5
  VMware DVS port 5404
  Port-Profile is pri_25
  Port mode is Private-vlan promiscuous
  5 minute input rate 0 bits/second, 0 packets/second
  5 minute output rate 7048 bits/second, 2 packets/second
  Rx
    20276 Input Packets 379239 Unicast Packets
    24 Multicast Packets 1395 Broadcast Packets
    1428168 Bytes
  Tx
    256229 Output Packets 74946 Unicast Packets
    16247 Multicast Packets 2028117 Broadcast Packets 190123 Flood Packets
    44432239 Bytes
    162 Input Packet Drops 159 Output Packet Drops
```

```
switch(config-if)#
```

Associating a Promiscuous Access Port with a Private VLAN

You can associate the promiscuous access port with the primary and secondary VLANs in a PVLAN.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know the VLAN IDs of the primary and secondary VLANs in the PVLAN.
- Know the primary and secondary VLANs that are already configured as PVLAN.
- Know the name of the interface functioning in the PVLAN as a promiscuous access port.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type* [*slot/port* | *number*]	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **switchport private-vlan mapping** *primary_vlan_ID* {*secondary_vlan_list* | **add** *secondary_vlan_list* | **remove** *secondary_vlan_list*}	Associates the promiscuous access port with the VLAN IDs in the PVLAN in the running configuration.
Step 4	switch(config-if)# **show interface** *type* [*slot/port* | *number*]	(Optional) Displays the interface configuration.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to associate a promiscuous access port with a PVLAN:

```
switch# configure terminal
switch(config)# interface eth3/2
switch(config-if)# switchport private-vlan mapping 202 303
switch(config-if)# show vlan private-vlan
Primary  Secondary  Type             Ports
-------  ---------  ---------------  -------------------------------------------
202      303        community        Eth3/2, Veth1

switch(config-if)#
```

Removing a Private VLAN Configuration

You can remove a PVLAN configuration and return the VLAN to normal VLAN mode.

Before You Begin

- Log in to the CLI in EXEC mode.
- The VLAN is configured as a private VLAN, and you know the VLAN ID.
- When you remove a PVLAN configuration, the ports associated with it become inactive.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan** *private vlan-id*	Enters the VLAN configuration mode for the specified VLAN.
Step 3	switch(config-vlan)# **no private-vlan** {**community** | **isolated** | **primary**}	Removes the specified VLAN from a PVLAN in the running configuration. The private VLAN configuration is removed from the specified VLAN(s). The VLAN is returned to normal VLAN mode. The ports associated with the VLAN are inactive.
Step 4	switch(config-vlan)# **exit**	Exits VLAN configuration mode.
Step 5	switch(config)# **show vlan private-vlan**	(Optional) Displays the PVLAN configuration.
Step 6	switch(config)# **copy running-config startup-config**	(Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

This example shows how to remove a PVLAN configuration:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# no private-vlan primary
switch(config-vlan)# exit
switch(config)# show vlan private-vlan
Primary  Secondary  Type             Ports
-------  ---------  ---------------  --------------------------------------------

switch(config)#
```

Verifying a Private VLAN Configuration

Use the following commands to verify a private VLAN configuration:

Command	Purpose
show feature	Displays features available and whether they are enabled globally.
show running-config vlan *vlan-id*	Displays VLAN information.
show vlan private-vlan [*type*]	Displays information about PVLANs.
show interface switchport	Displays information about all interfaces configured as switchports.

Configuration Examples for Private VLANs

Example: PVLAN Trunk Port

This example shows how to configure interface Ethernet 2/6 as the following:

- PVLAN trunk port
- Mapped to primary PVLAN 202 which is associated with secondary VLANs 303 and 440
- Mapped to primary PVLAN 210 which is associated with secondary VLANs 310 and 450

```
switch# configure terminal
switch(config)# vlan 303,310
switch(config-vlan)# private-vlan community
switch(config-vlan)# exit
switch(config)# vlan 440,450
switch(config-vlan)# private-vlan isolated
switch(config-vlan)# exit
switch(config)# vlan 202
switch(config-vlan)# private-vlan primary
switch(config-vlan)# private-vlan association 303,440
switch(config-vlan)# exit
switch(config)# vlan 210
switch(config-vlan)# private-vlan primary
switch(config-vlan)# private-vlan association 310,450
switch(config-vlan)# exit

switch# configure terminal
switch(config)# int eth2/6
switch(config-if)# switchport mode private-vlan trunk promiscuous
switch(config-if)# switchport private-vlan trunk allowed vlan all
switch(config-if)# switchport private-vlan mapping trunk 202 303, 440
switch(config-if)# switchport private-vlan mapping trunk 210 310, 450
switch(config-if)# show interface switchport
Name: Ethernet2/6
  Switchport: Enabled
Operational Mode: Private-vlan trunk promiscuous
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
```

```
Trunking VLANs Enabled: 1-3967,4048-4093
Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: 1
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: 1-3967, 4048-4093
  Administrative private-vlan trunk private VLANs: (202,303) (202,440) (210,310) (210,450)
  Operational private-vlan: 202,210,303,310,440,450
switch(config-if)#
```

Example: PVLAN Using Port Profiles

This example configuration shows how to configure interface eth2/6 using port-profile, uppvlanpromtrunk156.

In this configuration, packets from secondary interfaces 153, 154, and 155 are translated into the PVLAN 156:

```
vlan 153-154
  private-vlan community
vlan 155
  private-vlan isolated
vlan 156
  private-vlan association 153-155
  private-vlan primary


switch# show run int eth2/6

version 4.0(1)
interface Ethernet2/6
switchport
inherit port-profile uppvlanpromtrunk156

switch# show port-profile name uppvlanpromtrunk156
port-profile uppvlanpromtrunk156
description:
status: enabled
capability privileged: no
capability uplink: yes
port-group: uppvlanpromtrunk156
config attributes:
switchport mode private-vlan trunk promiscuous
switchport private-vlan trunk allowed vlan all
switchport private-vlan mapping trunk 156 153-155
no shutdown
evaluated config attributes:
switchport mode trunk
switchport trunk allowed vlan all
switchport private-vlan mapping trunk 156 153-155
no shutdown
assigned interfaces:
Ethernet2/6

switch# show interface eth 2/6 switchport
Name: Ethernet2/6
  Switchport: Enabled
  Switchport Monitor: Not enabled
  Operational Mode: Private-vlan trunk promiscuous
  Access Mode VLAN: 1 (default)
  Trunking Native Mode VLAN: 1 (default)
  Trunking VLANs Enabled: 1-3967,4048-4093
  Administrative private-vlan primary host-association: none
  Administrative private-vlan secondary host-association: none
  Administrative private-vlan primary mapping: none
  Administrative private-vlan secondary mapping: none
  Administrative private-vlan trunk native VLAN: 1
  Administrative private-vlan trunk encapsulation: dot1q
  Administrative private-vlan trunk normal VLANs: 1-155,157-3967,4048-4093
  Administrative private-vlan trunk private VLANs: (156,153) (156,155)
```

```
 Operational private-vlan: 156,153,155 inherit port-profile uppvlanpromtrunk156
switch#
```

Feature History for Private VLANs

Feature Name	Feature Name	Releases
feature private-vlan command	4.2(1)SV1(4)	Added the ability to globally enable the PVLAN feature.
Private VLAN	4.0(4)SV1(1)	This feature was introduced.

CHAPTER 6

Configuring IGMP Snooping

This chapter contains the following sections:

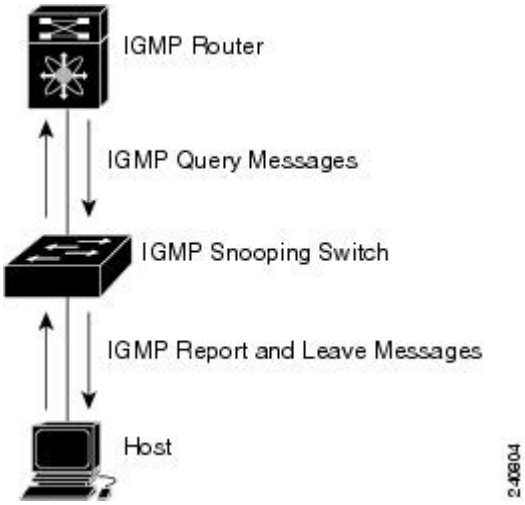
Information About IGMP Snooping

Introduction

The Internet Group Management Protocol (IGMP) snooping software examines Layer 2 IP multicast traffic within a VLAN to discover the ports where interested receivers reside. Using the port information, IGMP snooping can reduce bandwidth consumption in a multi-access LAN environment to avoid flooding the entire VLAN. The IGMP snooping feature tracks which ports are attached to multicast-capable routers to help the routers forward IGMP membership reports. The IGMP snooping software responds to topology change notifications. By default, IGMP snooping is enabled on the device.

The following figure shows an IGMP snooping switch that sits between the host and the IGMP router. The IGMP snooping switch snoops the IGMP membership reports and Leave messages and forwards them only when necessary to the connected IGMP routers.

Figure 3: IGMP Snooping Switch



The IGMP snooping software operates upon IGMPv1, IGMPv2, and IGMPv3 control plane packets where Layer 3 control plane packets are intercepted and influence the Layer 2 forwarding behavior.

The Cisco Nexus 1000V IGMP snooping implementation has the following proprietary features:

- Multicast forwarding based on an IP address rather than a MAC address.
- Optimized multicast flooding (OMF) that forwards unknown traffic to routers only and performs no data driven state creation.

For more information about IGMP snooping, see RFC 4541.

IGMPv1 and IGMPv2

IGMPv2 supports the fast leave feature. The fast leave feature does not send last member query messages to hosts. As soon as the software receives an IGMP leave message, the software stops forwarding multicast data to that port.

IGMPv1 does not provide an explicit IGMP leave message, so the software must rely on the membership message timeout to indicate that no hosts remain that want to receive multicast data for a particular group.

Report suppression is not supported and is disabled by default.

IGMPv3

IGMPv3 snooping provides constrained flooding based on the group IP information in the IGMPv3 reports. Report suppression is not supported and disabled by default. In addition, explicit tracking is not supported and disabled by default. Instead, the fast leave feature is used for handling leave messages.

Prerequisites for IGMP Snooping

IGMP snooping has the following prerequisites:

- You are logged in to the switch.
- A querier must be running on the uplink switches on the VLANs that contain multicast sources and receivers.

When the multicast traffic does not need to be routed, you must configure an external switch to query membership. On the external switch, define the query feature in a VLAN that contains multicast sources and receivers but no other active query feature. In the Cisco Nexus 1000V, report suppression is not supported and is disabled by default.

When an IGMP snooping query feature is enabled on an upstream switch, it sends out periodic IGMP queries that trigger IGMP report messages from hosts wanting to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to identify accurate forwarding.

Default Settings

Table 8: Default IGMP Snooping Settings

Parameters	Default
IGMP snooping	Enabled
IGMPv3 Explicit tracking	Enabled
IGMPv2 Fast leave	Disabled
Last member query interval	1 second
Link-local groups suppression	Enabled
Snooping querier	Disabled
IGMPv1/v2 Report suppression	Disabled
IGMPv3 Report suppression	Disabled

Configuring IGMP Snooping

Enabling or Disabling IGMP Snooping Globally for the VSM

You can enable or disable IGMP snooping globally for the VSM. IGMP snooping is enabled globally on the VSM (the default). If enabled globally, you can turn it on or off per VLAN.

Before You Begin

Log in to the CLI in EXEC mode.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **ip igmp snooping**	Enables or disables IGMP snooping in the running configuration for all VLANs. The default is enabled. If you have previously disabled the feature then you can enable it with this command.
Step 3	switch(config)# **show ip igmp snooping** [**vlan** *vlan-id*]	(Optional) Displays the configuration for verification. **Note** If disabled, IGMP snooping on all VLANs is disabled.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to disable IGMP snooping:

```
switch# configure terminal
switch(config)# no ip igmp snooping
switch(config)# show ip igmp snooping
Global IGMP Snooping Information:
  IGMP Snooping enabled
  IGMPv1/v2 Report Suppression disabled
  IGMPv3 Report Suppression disabled
  Link Local Groups Suppression enabled

IGMP Snooping information for vlan 1
  IGMP snooping enabled
  IGMP querier none
  Switch-querier disabled
  IGMPv3 Explicit tracking enabled
  IGMPv2 Fast leave disabled
  IGMPv1/v2 Report suppression disabled
  IGMPv3 Report suppression disabled
  Link Local Groups suppression enabled
  Router port detection using PIM Hellos, IGMP Queries
  Number of router-ports: 0
  Number of groups: 0
```

```
  Active ports:

--More--
switch(config)#
```

Configuring IGMP Snooping on a VLAN

You can configure IGMP snooping on a VLAN. IGMP snooping is enabled by default for all VLANs in the VSM.

Before You Begin

Log in to the CLI in EXEC mode.

Note If IGMP snooping is disabled globally, it takes precedence over the VLAN state.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vlan configuration** *vlan-id*	Enters configuration mode for the specified VLAN.
Step 3	switch(config-vlan-config)# [**no**] **ip igmp snooping**	Enables or disables IGMP snooping in the running configuration for the specific VLAN. If IGMP snooping is enabled for the VSM, IGMP snooping is enabled for the VLAN by default. **Note** IGMP snooping must be enabled globally (the default) in order to toggle it on or off per VLAN. If IGMP snooping is disabled globally, then it cannot be enabled per VLAN.
Step 4	switch(config-vlan-config)# [**no**] **ip igmp snooping explicit-tracking**	(Optional) Tracks IGMPv3 membership reports from individual hosts for each port on a per-VLAN basis in the running configuration. The default is enabled.
Step 5	switch(config-vlan-config)# [**no**] **ip igmp snooping fast-leave**	(Optional) Enables fast-leave for the specified VLAN in the running configuration. Fast-leave supports IGMPv2 hosts that cannot be explicitly tracked because of the host report suppression mechanism of the IGMPv2 protocol. Enables the software to remove the group state when it receives an IGMP Leave report without sending an IGMP query message. This parameter is used for IGMPv2 hosts when no more than one host is present on each VLAN port.

	Command or Action	Purpose
		When you enable fast leave, the IGMP software assumes that no more than one host is present on each VLAN port. The default is disabled.
Step 6	switch(config-vlan-config)# [**no**] **ip igmp snooping mrouter interface** type*if_id*	(Optional) Configures a static connection for the VLAN to a multicast router in the running configuration. The interface to the router must be in the specified VLAN. You can specify the interface by the type and the number, such as ethernet slot/port. vEths are not supported as router ports.
Step 7	switch(config-vlan-config)# [**no**] **ip igmp snooping static-group** *group-ip-addr* **interface** type*if_id*	(Optional) Configures a VLAN Layer 2 port as a static member of a multicast group in the running configuration. You can specify the interface by the type and the number, such as ethernet slot/port.
Step 8	switch(config-vlan-config)# [**no**] **ip igmp snooping link-local-groups-suppression**	(Optional) Configures link-local groups suppression. The default is enabled. **Note** You can apply link-local groups suppression to all interfaces in the VSM by entering this command in global configuration mode.
Step 9	switch(config-vlan-config)# **exit**	Exits VLAN configuration mode.
Step 10	switch(config)# **show ip igmp snooping** [**vlan** *vlan-id*]	(Optional) Displays the configuration for verification.
Step 11	switch(config)# **copy running-config startup-config**	(Optional) (Optional) Saves the running configuration persistently through reboots and restarts by copying it to the startup configuration.

This example shows how to configure IGMP snooping on a VLAN:

```
switch# configure terminal
switch(config)# vlan configuration 2
switch(config-vlan-config)# ip igmp snooping
switch(config-vlan-config)# ip igmp snooping explicit-tracking
switch(config-vlan-config)# ip igmp snooping fast-leave
switch(config-vlan-config)# ip igmp snooping mrouter interface type ethernet 2/1
switch(config-vlan-config)# ip igmp snooping static-group 230.0.0.1 interface type ethernet
 2/1
switch(config-vlan-config)# ip igmp snooping link-local-groups-suppression
switch(config-vlan-config)# exit
switch(config)# show ip igmp snooping vlan 2

IGMP Snooping information for vlan 5
  IGMP snooping enabled
  IGMP querier none
  Switch-querier disabled
```

```
  IGMPv3 Explicit tracking enabled
  IGMPv2 Fast leave enabled
  IGMPv1/v2 Report suppression disabled
  IGMPv3 Report suppression disabled
  Link Local Groups suppression enabled
  Router port detection using PIM Hellos, IGMP Queries
  Number of router-ports: 1
  Number of groups: 1
  Active ports:
switch(config)#
```

Verifying the IGMP Snooping Configuration

Use the following commands to verify the IGMP snooping configuration information.

Command	Purpose
show ip igmp snooping [**vlan** *vlan-id*]	Displays IGMP snooping configuration by VLAN.
show ip igmp snooping groups [**vlan** *vlan-id*] [**detail**]	Displays IGMP snooping information about groups by VLAN.
show ip igmp snooping querier [**vlan** *vlan-id*]	Displays IGMP snooping queriers by VLAN.
show ip igmp snooping mroute [**vlan** *vlan-id*]	Displays multicast router ports by VLAN.
show ip igmp snooping explicit-tracking [**vlan** *vlan-id*]	Displays IGMP snooping explicit tracking information by VLAN.

For detailed information about commands and their output, see the *Cisco Nexus 1000V Command Reference*.

Example Configuration IGMP Snooping

This example shows how to enable IP IGMP snooping for the VSM and make the following optional configurations for VLAN 2:

- Tracking of IGMPv3 membership reports from individual hosts for each port.
- A static connection to a multicast router through Ethernet 2/1.
- Static membership in multicast group 230.0.0.1.

```
switch# configure terminal
switch(config)# ip igmp snooping
switch(config)# vlan configuration 2
switch(config-vlan-config)# ip igmp snooping
switch(config-vlan-config)# ip igmp snooping explicit-tracking
switch(config-vlan-config)# ip igmp snooping mrouter interface ethernet 2/1
switch(config-vlan-config)# ip igmp snooping static-group 230.0.0.1 interface
ethernet 2/1
switch(config-vlan-config)# exit
switch(config)# show ip igmp snooping vlan 2
switch# copy running-config startup-config
switch#
```

Feature History for IGMP Snooping

Feature Name	Releases	Description
Link-local suppression	4.2(1)SV1(4)	Added support to enable or disable link-local group suppression.
Report suppression	4.0(4)SV1(3)	Removed support for report suppression.
IGMP Snooping	4.0(4)SV1(1)	This feature was introduced.

CHAPTER 7

Configuring Network Load Balancing for vEthernet

This chapter contains the following sections:

Information About Microsoft Network Load Balancing

Microsoft Network Load Balancing (NLB) is a clustering technology offered by Microsoft as part of the Windows server operating systems. Clustering enables a group of independent servers to be managed as a single system for higher availability, easier manageability, and greater scalability.

For more information about Microsoft NLB, see http://technet.microsoft.com/en-us/library/bb742455.aspx

Note Access to third-party websites identified in this document is provided solely as a courtesy to customers and others. Cisco Systems, Inc. and its affiliates are not in any way responsible or liable for the functioning of any third-party website, or the download, performance, quality, functioning or support of any software program or other item accessed through the website, or any damages, repairs, corrections or costs arising out of any use of the website or any software program or other item accessed through the website. Cisco's End User License Agreement does not apply to the terms and conditions of use of a third-party website or any software program or other item accessed through the website.

Guidelines and Limitations

NLB has the following configuration guidelines and limitations:

- The **no mac auto-static-learn** command is not supported on PVLAN ports.
- The **no mac auto-static-learn** command is not supported on the ports that are configured with **switchport port-security mac-address sticky**.
- Unknown unicast flood blocking (UUFB) does not block Microsoft-Network Load Balancing (MS-NLB) packets on MS-NLB vEthernet interfaces. UUFB can be used to limit flooding of MS-NLB packets to non-MS-NLB ports within a VLAN.

Configuring Microsoft Networking Load Balancing in Unicast Mode

You can configure Microsoft NLB in the interface or the port profile configuration mode.

Configuring Microsoft Network Load Balancing Support in Interface Configuration Mode

You can configure Microsoft NLB in the interface configuration mode.

Before You Begin

Note Make sure that the Cisco Nexus 1000V is configured before you configure Microsoft NLB on Windows virtual machines (VMs).

- Log in to the CLI in EXEC mode.
- Know that unicast is the default Microsoft Network Load Balancing mode of operation.
- Know that Microsoft NLB replaces the MAC address of each server in the cluster to a common Microsoft NLB MAC address.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **show running-config interface** *veth number*	Displays the vEthernet configuration to determine if no mac auto-static-learning is configured or not.
Step 3	switch(config)# **interface** *veth number*	Sets interface configuration mode on vEthernet modules.
Step 4	switch(config-if)# [**no**] **mac auto-static-learn**	Toggles auto-mac-learning on vEthernet modules.

	Command or Action	Purpose
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure Microsoft NLB directly on a vEthernet interface:

```
switch# configure terminal
switch(config-if)# show running-config interface vethernet 1
switch(config)# interface vethernet 1
switch(config-if)# no mac auto-static-learn
!Command: show running-config interface Vethernet1
!Time: Tue Nov 15 19:01:36 2011

version 4.2(1)SV1(5.1)

interface Vethernet1
  inherit port-profile vm59
  description stc3, Network Adapter 2
  no mac auto-static-learn
  vmware dvport 34 dvswitch uuid "ea 5c 3b 50 cd 00 9f 55-41 a3 2d 61 84 9e 0e c4"
  vmware vm mac 0050.56B3.0071

switch(config)#
```

This example shows how to unconfigure Microsoft NLB directly from a vEthernet interface:

```
switch# configure terminal
switch(config-if)# show running-config interface vethernet 1
switch(config)# interface vethernet 1
switch(config-if)# mac auto-static-learn
!Command: show running-config interface Vethernet1
!Time: Tue Nov 15 19:01:52 2011

version 4.2(1)SV1(5.1)

interface Vethernet1
  inherit port-profile vm59
  description stc3, Network Adapter 2
  mac auto-static-learn
  vmware dvport 34 dvswitch uuid "ea 5c 3b 50 cd 00 9f 55-41 a3 2d 61 84 9e 0e c4"
  vmware vm mac 0050.56B3.0071

switch(config)#
```

Configuring Microsoft Network Load Balancing in Port Profile Configuration Mode

You can configure Microsoft NLB in the port profile configuration mode.

Before You Begin

Note Make sure that the Cisco Nexus 1000V is configured before you configure Microsoft NLB on Windows Virtual Machines (VMs).

- Log in to the CLI in EXEC mode.
- Know that unicast is the default Microsoft Network Load Balancing mode of operation.
- Know that Microsoft NLB replaces the MAC address of each server in the cluster to a common Microsoft NLB MAC address.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **show running config port-profile** *profile name*	Displays the port profile configuration to determine if no mac auto-static-learning is configured or not.
Step 3	switch(config)# **port profile type vethernet ms-nlb**	Sets port profile configuration mode on vEthernet modules.
Step 4	switch(config-port-prof)# [**no**] **mac auto-static-learn**	Toggles auto-mac-learning on vEthernet modules.
Step 5	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure Microsoft NLB in port profile mode:

```
switch# configure terminal
switch(config-port-prof)# show running-config port-profile ms-nlb
!Command: show running-config port-profile ms-nlb
!Time: Tue Nov 15 19:00:40 2011

version 4.2(1)SV1(5.1)
port-profile type vethernet ms-nlb
  vmware port-group
  switchport mode access
  switchport access vlan 59
  no mac auto-static-learn
  no shutdown
  state enabled
switch(config-port-prof)#
```

This example shows how to unconfigure Microsoft NLB on a vEthernet interface in port profile mode:

```
switch# configure terminal
switch(config)# port-profile type vethernet ms-nlb
switch(config-port-prof)# mac auto-static-learn
switch(config-port-prof)# show running-config port-profile ms-nlb
!Command: show running-config port-profile ms-nlb
!Time: Tue Nov 15 19:01:05 2011

version 4.2(1)SV1(5.1)
port-profile type vethernet ms-nlb
  vmware port-group
  switchport mode access
  switchport access vlan 59
  mac auto-static-learn
  no shutdown
  state enabled
switch(config-port-prof)#
```

Configuring Microsoft Networking Load Balancing in Multicast Mode

You can configure Microsoft NLB in multicast mode.

Attention Microsoft NLB in multicast mode requires IGMP Snooping to be disabled on Cisco Nexus 1000V.When this is done, Microsoft NLB packets on that VLAN are flooded as unknown multicast.

In order to minimize flooding, it is suggested to use a dedicated VLAN for Microsoft NLB traffic.

Note The Multicast mode assigns the cluster unicast virtual IP address to a non-Internet Assigned Numbers Authority (IANA) multicast MAC address (03xx.xxxx.xxxx). IGMP snooping does not dynamically register this address, which results in flooding of the NLB traffic in the VLAN as an unknown multicast.

Before You Begin

Note Make sure that the Cisco Nexus 1000V is configured before you configure Microsoft NLB on Windows virtual machines (VMs).

- Know that unicast is the default Microsoft Network Load Balancing mode of operation.
- Make sure IGMP Snooping is disabled on Cisco Nexus 1000V. For information on IGMP Snooping, see the Configuring IGMP Snooping section in the *Cisco Nexus 1000V Layer 2 Switching Configuration Guide*.

Procedure

Step 1 On the Windows VM, navigate to **Start** > **Control Panel** > **Network Connections** > **Local Area Connections** > **Properties**.
Displays the **Local Area Connection Properties** dialog box.

Step 2 Select the **Network Load Balancing** checkbox, and then click **Properties**.
Displays the **Network Load Balancing Properties** dialog box.

Step 3 Click the **Cluster Parameters** tab and complete the following fields:

- IP Address
- Subnet mask
- Full internet name

Step 4 Check the **Multicast** check box.
Enables the multicast mode.
The NLB converts the cluster network address into a multicast address.

Note You must configure a static ARP entry on the upstream router/switches, such as **ip arp ipaddr mac_addr**. For example, **ip arp 10.3.4.5 03bf.0a03.0405**

Step 5 Check the **Allow remote control** check box.
Enables the remote control operations.

Step 6 Enter the password in the **Remote password** and **Confirm password** fields.
Configures the password.

Feature History for Microsoft Network Load Balancing for vEthernet

Feature Name	Feature Name	Releases
Network Load Balancing	4.2(1)SV1(5.1)	This feature was introduced.

CHAPTER 8

Supporting Redundant Routing Protocols

This chapter contains the following sections:

Information About Redundant Routing Protocols

The Cisco Nexus 1000V implements a loop detection mechanism that is based on source and destination MAC addresses and drops packets that are coming in on uplink ports if the source MAC address is already present on a local vEthernet interface. As a result, such protocols as the Virtual Router Redundancy Protocol (VRRP), the Common Address Redundancy Protocol (CARP), the Hot Standby Router Protocol (HSRP), and other similar protocols fail on Virtual Machines (VMs) that are associated to the Cisco Nexus 1000V.

Disabling loop detection provides a flexible way of supporting these protocols on VMs that are associated to the Cisco Nexus 1000V. By disabling the loop detection mechanism, you can configure any combination of the above mentioned protocols on a port profile or a vEthernet interface. As a result, you can run multiple protocols on the same VM.

Guidelines and Limitations

Supporting the redundant routing protocols feature has the following configuration guidelines and limitations:

- A disabled loop detection configuration is not supported on PVLAN ports.
- A disabled loop detection configuration is not supported on the port security ports.

Supporting Redundant Routing Protocols

Configuring a vEthernet Interface to Support Redundant Routing Protocols

You can configure a vEthernet interface to support redundant routing protocols.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know which redundant routing protocol that you want to disable.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *interface-number*	Enters interface configuration mode for the specified vEthernet interface (from 1 to 1048575).
Step 3	switch(config-if)# **disable-loop-detection** {**carp** | **hsrp** | **vrrp** |**custom-rp** [**src-mac-range** *s_mac end_mac*] [**dest-ip** *ip_address*] [**ip-proto** *no*] [**port** *port*]}	Enables or disables the loop detection mechanism to support a redundant routing protocol on a vEthernet interface. • **disable-loop-detection—**Disables the loop detection mechanism. • **no disable-loop-detection—**Enables the loop detection mechanism. This is the default configuration. The protocols supported on a vEthernet interface are as follows: • **carp—**Common Address Redundancy Protocol • **custom-rp—**User-defined protocol • **hsrp—**Hot Standby Router Protocol • **vrrp—**Virtual Router Redundancy Protocol The parameters for custom defined protocols are as follows: • **src-mac-range—**Source MAC address range for the user-defined protocol. • **dest-ip—**Destination IP address for the user-defined protocol. • **ip-proto—**IP protocol number for the user-defined protocol. • **port—**UDP or TCP destination port number for the user-defined protocol.

	Command or Action	Purpose
Step 4	switch(config-if)# **show running-config interface vethernet** *interface-number*	(Optional) Displays the interface status and information.
Step 5	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a vEthernet interface to support VRRP, CERP, HSRP, and user-defined protocols on a VM:

```
switch# configure terminal
switch# show running-config interface vethernet 5
switch(config)# interface veth5
switch(config-if)# disable-loop-detection carp
switch(config-if)# disable-loop-detection vrrp
switch(config-if)# disable-loop-detection hsrp
switch(config-if)# disable-loop-detection custom-rp dest-ip 224.0.0.12 port 2234
!Command: show running-config interface Vethernet5
!Time: Fri Nov 4 02:21:24 2011

version 4.2(1)SV1(5.1)

interface Vethernet5
inherit port-profile vm59
description Fedora117, Network Adapter 2
disable-loop-detection carp
disable-loop-detection custom-rp dest-ip 224.0.0.12 port 2234
disable-loop-detection hsrp
disable-loop-detection vrrp
vmware dvport 32 dvswitch uuid "ea 5c 3b 50 cd 00 9f 55-41 a3 2d 61 84 9e 0e c4"
vmware vm mac 0050.56B3.00B2

switch#
```

Configuring a Port Profile to Support Redundant Routing Protocols

You can configure a port profile to support redundant routing protocols. Use this procedure when the master in a master/slave relationship has lost connectivity, the slave has taken over the master role, or the original master is attempting to overtake the master role.

Note If you configure a vEthernet interface and a port profile to run multiple protocols on the same VM, the configuration on the vEthernet interface overrides the configuration on the port profile.

Before You Begin

- Log in to the CLI in EXEC mode.
- Know which redundant routing protocol that you want to disable.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *name*	Enters port profile configuration mode for the named port profile.
Step 3	switch(config-port-prof)# **switchport mode** {**access** | **trunk**}	Designates that the interface is to be used as a trunking port. A trunk port transmits untagged packets for the native VLAN and transmits encapsulated, tagged packets for all other VLANs.
Step 4	switch(config-port-prof)# **no shutdown**	Administratively enables all ports in the profile.
Step 5	switch(config-port-prof)# **disable-loop-detection** {**carp** | **hsrp** | **vrrp** |**custom-rp** [**src-mac-range** *s_mac end_mac*] [**dest-ip** *ip_address*] [**ip-proto** *no*] [**port** *port*]}	Enables or disables the loop detection mechanism to support a redundant routing protocol on vEthernet interface. • **disable-loop-detection—**Disables the loop detection mechanism. • **no disable-loop-detection—**Enables the loop detection mechanism. This is the default configuration. The protocols supported on a vEthernet interface are as follows: • **carp—**Common Address Redundancy Protocol • **custom-rp—**User defined protocol • **hsrp—**Hot Standby Router Protocol • **vrrp—**Virtual Router Redundancy Protocol The parameters for custom defined protocols are as follows: • **src-mac-range—**Source MAC address range for the user defined protocol. • **dest-ip—**Destination IP address for the user defined protocol. • **ip-proto—**IP protocol number for the user defined protocol. • **port—**UDP or TCP destination port number for the user defined protocol.
Step 6	switch(config-port-prof)# **show port-profile** [**brief** | **expand-interface** | **usage**] [**name** *profile-name*]	(Optional) Displays the configuration for verification.

	Command or Action	Purpose
Step 7	switch(config-port-prof)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to disable loop detection for the HSRP:

```
switch# configure terminal
switch(config)# port-profile hsrp-1
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# no shutdown
switch(config-port-prof)# disable-loop-detection hsrp
switch(config-port-prof)# show port-profile name hsrp-1
port-profile hsrp-1
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  switchport mode trunk
  disable-loop-detection hsrp
  no shutdown
 evaluated config attributes:
  switchport mode trunk
  disable-loop-detection hsrp
  no shutdown
 assigned interfaces:
 port-group: hsrp-1
 system vlans: none
 capability l3control: no
 capability iscsi-multipath: no
 capability vxlan: no
 capability l3-vservice: no
 port-profile role: none
 port-binding: static
```

This example shows how to disable loop detection for the VRRP:

```
n1000v# configure terminal
switch(config)# port-profile vrrp-1
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# no shutdown
switch(config-port-prof)# disable-loop-detection vrrp
switch(config-port-prof)# show port-profile name vrrp-1
port-profile vrrp-1
 type: Vethernet
 description:
 status: enabled
 max-ports: 32
 min-ports: 1
 inherit:
 config attributes:
  switchport mode trunk
  disable-loop-detection vrrp
  no shutdown
 evaluated config attributes:
  switchport mode trunk
  disable-loop-detection vrrp
  no shutdown
 assigned interfaces:
 port-group: vrrp-1
 system vlans: none
 capability l3control: no
```

```
capability iscsi-multipath: no
capability vxlan: no
capability l3-vservice: no
port-profile role: none
port-binding: static
```

Feature History for Supporting Redundant Routing Protocol

Feature Name	Feature Name	Releases
Supporting Redundant Routing Protocol	4.2(1)SV1(5.1)	This feature was introduced.

CHAPTER 9

Configuring BPDU Guard

This chapter contains the following sections:

Information About Bridge Protocol Data Unit Guard Feature

The Bridge Protocol Data Unit (BPDU) Guard feature is one of the Spanning Tree Protocol (STP) enhancements. This feature enhances switch network reliability, manageability, and security.

STP ensures a loop-free topology for any Ethernet LAN. STP prevents loops and broadcast radiation. We recommend that you enable BPDU Guard on access ports so that any end user devices on these ports that have BPDU Guard enabled cannot influence the topology. Any malfunctioning device that is connected to a vEthernet port can flood the Layer 2 network with unwanted BPDU that causes STP to break down. When you enable BPDU Guard feature on the access-ports, it shuts down the port that receives a BPDU. To bring up a port disabled by BDPU guard, you must remove the device and then restart the port by entering the **shut/no shut** command described later in this document.

Prerequisites for BPDU Guard

BPDU Gaurd has the following prerequisite:

- To configure BPDU Guard, you must install the Advanced Edition license on the Cisco Nexus 1000V switch.

Enabling or Disabling BPDU Guard Feature Globally

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] spanning-tree port type edge bpduguard default**	Globally enables or disables the BPDU Guard.
Step 3	switch(config)# **show spanning-tree bpduguard info**	(Optional) Displays the BPDU Guard state.
Step 4	switch(config)# **show switch edition**	(Optional) Displays the features that requires the Advanced Edition license on the Cisco Nexus 1000V switch.

This example shows how to enable BPDU Guard globally:

```
switch# configure terminal
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# show spanning-tree bpduguard info
Global spanning-tree bpduguard status: Enabled

switch(config)# show switch edition
Switch Edition: ADVANCED (3.0)

Feature Status
Name            State           Licensed    In version
-------------------------------------------------------
cts             disabled        Y           1.0
dhcp-snooping   disabled        Y           1.0
vxlan-gateway   disabled        Y           1.0
bgp             enabled         Y           3.0
bpduguard       enabled         Y           3.0

License Status
Edition      Available  In Use     Expiry Date
---------------------------------------------
Advanced     30         2          Never

Scale Support
Edition      Modules      Virtual Ports
-------------------------------------
Essential    128          4096
Advanced     256          12288
```

Enabling or Disabling BPDU Guard Mode on Port Profile

You can enable or disable BPDU Guard for a specific port profile. Configuring BPDU Gguard for a specific port profile will overwrite global configuration for the vEthernet ports that inherits the port profile. If you disable BPDU Guard globally, you can enable it for a specific port profile to overwrite the global configuration mode. The vEthernet ports under that port profile can receive BPDU packets without going to an error-disabled mode. Similarly, if you enable BPDU Guard is enabled globally, you can disable it for a specific port profile.

Note This port profile configuration overwrites the global configuration.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **port-profile** *profile_id*	Enters port-profile configuration mode.
Step 3	switch(config-port-prof)# **spanning-tree bpduguard{enable | disable}**	Enables or disables BPDU Guard for the particular vlan ID. **Note** You can remove the BPDU configuration from the port profile by using theno spanning-tree bpduguard command.
Step 4	switch(config-port-prof)# **end**	Exits port profile configuration mode.
Step 5	switch(config)# **show interface virtual spanning-tree bpduguard status**	(Optional) Displays the vEthernet ports and the BPDU Guard status for all interfaces. **Note** If a vEthernet port is inheriting global BPDU Guard settings, it does not display the status.
Step 6	switch(config)# **show interface virtual spanning-tree bpduguard status module** *module_no*	(Optional) Displays the vEthernet ports and BPDU Guard status for a specific module.

This example shows how to enable BPDU Guard on a VLAN port profile:

```
switch# configure terminal
switch(config)# port-profile VLAN-1238
switch(config-port-prof)# spanning-tree bpduguard enable
switch(config-port-prof)# end
switch(config)# show interface virtual spanning-tree bpduguard status
Veth77      Enabled
Veth770     -
Veth771     -
Veth772     -
Veth773     -
Veth774     Disabled
Veth775     -
Veth776     -
Veth777     Enabled
Veth778     -
Veth779     Enabled
```

Enabling or Disabling BPDU Guard on a vEthernet Port

You can enable or disable the BPDU Guard for a specific port. Configuring BPDU Guard for a specific port overrides global and port profile configurations. If you disable BPDU Guard globally or at a port profile level, you can enable it for a specific port to override you disable global or port profile configurations. The port can

receive BPDU packets without going to an error-disabled mode. Similarly, if you enable BPDU Guard globally or at a port profile level, you can disable it for a specific port.

Note This vEthernet port configuration overrides the global and port-profile level configuration.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *vethernet port*	Enters port \configuration mode.
Step 3	switch(config-if)# **spanning-tree bpduguard{enable | disable}**	Enables or disables BPDU Guard for the particular vEthernet port. **Note** You can remove the BPDU configuration from the port profile by using the no spanning-tree bpduguard command.
Step 4	switch(config-if)# **end**	Exits the port configuration mode.
Step 5	switch(config)# **show interface virtual spanning-tree bpduguard status**	(Optional) Displays the vEthernet ports and the BPDU Guard status for all interfaces. **Note** If a vEthernet port is inheriting global BPDU Guard settings, it does not display the status.

This example shows how to enable BPDU Guard on a VLAN port profile:

```
switch# configure terminal
switch(config)# interface vethernet 77
switch(config-if)# spanning-tree bpduguard enable
switch(config-port-prof)# end
switch(config)# show interface virtual spanning-tree bpduguard status
Veth77      Enabled
Veth770     -
Veth771     -
Veth772     -
Veth773     -
Veth774     Disabled
Veth775     -
Veth776     -
Veth777     Enabled
Veth778     -
Veth779     Enabled
```

Bringing up a vEthernet Port

Before You Begin

- You are getting the `Err_disable : BPDU guard violation ltl (port id) , ifindex(1c000030)` error.

- Ensure that the device that caused the port to shut down is removed from the network.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface vethernet** *vethernet port*	Enters port configuration mode.
Step 3	switch(config-if)# **shut**	Shuts down the vEthernet administratively.
Step 4	switch(config-if)# **no shut**	Starts the vEthernet port.
Step 5	switch(config-if)#**show interface vethernet** *port id*	(Optional) Displays the vEthernet port information.

This example shows how to bring up a vEthernet port:

```
switch# configure terminal
switch(config)# interface vethernet 4
switch(config-if)# shut
switch(config-if)# 2014 May 19 02:13:09 switch ethpm[2808]: %ETHPORT-5-IF_DOWN_ADMIN_DOWN:

Interface Vethernet4 is down (Administratively down)
no shut
2014 May 19 02:13:11 switch ethpm[2808]: %ETHPORT-5-IF_ADMIN_UP: Interface Vethernet4 is
admin up .
switch(config-if)# 2014 May 19 02:13:11 switch ethpm[2808]: %ETHPORT-5-IF_UP: Interface
Vethernet4 is up in mode access
end
switch#
switch# 2014 May 19 02:13:13 switch vshd[32105]: %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configured
from vty by admin on 7.1.4.25@pts/0
switch# show interface vethernet 4
Vethernet4 is up
  Port description is OST-SUSE-2-E100-1, Network Adapter 2
  Hardware: Virtual, address: 0050.5681.4a36 (bia 0050.5681.4a36)
  Owner is VM "OST-SUSE-2-E100-1", adapter is Network Adapter 2
  Active on module 8
  VMware DVS port 11906
  Port-Profile is VLAN-1238
  MTU 1500 bytes
  Port mode is access
  5 minute input rate 1240 bits/second, 2 packets/second
  5 minute output rate 312 bits/second, 0 packets/second
  Rx
    6715801 Input Packets 6714907 Unicast Packets
    836 Multicast Packets 58 Broadcast Packets
    0 Jumbo Packets
    6997031276 Bytes
  Tx
    8113 Output Packets 0 Unicast Packets
    3296 Multicast Packets 4817 Broadcast Packets 426 Flood Packets
    0 Jumbo Packets
    780299 Bytes
    0 Input Packet Drops 0 Output Packet Drops
```

Feature History for BPDU Guard

Feature Name	Release Name	Description
BPDU Guard	5.2(1)SV3(1.1)	This feature was introduced.

CHAPTER 10

Layer 2 Switching Configuration Limits

This chapter contains the following sections:

Layer 2 Switching Configuration Limits

The configuration limits are documented in the *Cisco Nexus 1000V Resource Availability Reference*.

Layer 2 Switching Configuration Limits